Machine Learning

DECAP737

Edited by
Dr. V Devenderan

LOVELY PROFESSIONAL UNIVERSITY

Machine Learning

Edited By:
Dr. V Devenderan

Content

Unit 1:	***Introduction to Machine Learning*** *Dr. VDevendran, Lovely Professional University*	**1**
Unit 2:	***Python Basics*** *Dr. VDevendran, Lovely Professional University*	**11**
Unit 3:	***Data Pre-Processing*** *Dr. VDevendran, Lovely Professional University*	**31**
Unit 4:	***Implementation ofPre-processing*** *Dr. VDevendran, Lovely Professional University*	**42**
Unit 5:	***Physical Layer*** *Dr. Rajni Bhalla, Lovely Professional University*	**56**
Unit 6:	***Introduction to Numpy*** *Dr. Rajni Bhalla, Lovely Professional University*	**73**
Unit 7:	***Classification*** *Dr. VDevendran, Lovely Professional University*	**87**
Unit 8:	***Classification Algorithms*** *Dr. Rajni Bhalla, Lovely Professional University*	**99**
Unit 9:	***Classification Implementation*** *Dr. VDevendran, Lovely Professional University*	**111**
Unit 10:	***Clustering*** *Dr. VDevendran, Lovely Professional University*	**125**
Unit 11:	***Ensemble Methods*** *Dr. VDevendran, Lovely Professional University*	**134**
Unit 12:	***Data Visualization*** *Dr. Rajni Bhalla, Lovely Professional University*	**145**
Unit 13:	***Neural Networks*** *Dr. VDevendran, Lovely Professional University*	**160**
Unit 14:	***Neural Network Implementation*** *Dr. Rajni Bhalla, Lovely Professional University*	**171**

Unit 01: Introduction to Machine Learning

CONTENTS

Objectives

Introduction

1.1 Introduction to Machine Learning

1.2 Data set

1.3 Supervised Learning

1.4 Unsupervised Learning

1.5 Reinforcement Learning

1.6 Applications of Machine Learning

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further readings

Objectives

- Understanding the concepts of machine learning.
- Understanding the difference between variousmachine learning approaches.
- Understanding various basic data types.
- Understanding the major tasks in preprocessing.
- Understanding the real time applications of machine learning.

Introduction

A computer program is said to learn from experience with respect to some class of tasks and performance measure, if the performance at the tasks, as measured by performance measure, improves with the experience. In this unit, the concepts of machine learning are discussed in detail. The different types of machine learning approaches are discussed using examples. The preparation of the data sets is discussed along with basic data types and data cleaning operations. There is a important part in machine learning, known as preprocessing and feature engineering are also highlighted.

1.1 Introduction to Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. This is now widely used in across all the domains, starting from classification problems to humanoid robot.

The output of an algorithm represents the learned knowledge. The given model as in figure 1, is deployed by the user for decision-making; it gives the prediction with respect to the assigned task for measurements / observations not in task experience; a good model will generalize well to

observations unseen by the machine during training. Huge data collection and storage technologies have altered the landscape of scientific data analysis, which includes natural resources, prediction of floods, astronomy, biology and etc. Machine learning is present in all those examples.

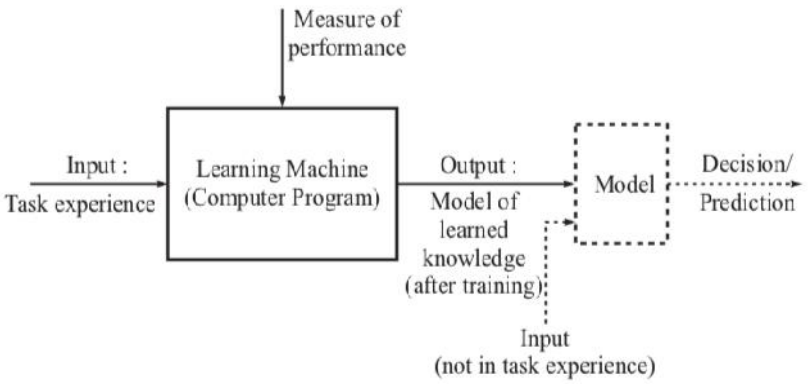


Figure1: Block Diagram of a learning machine

1.2 Data set

Data set is the collection of data used for machine learning. Basically, the dataset is divdided into three categories. They are training data, testing Data and validation Data. Here, the training data is considered for initial training purpose. Testing data is used for checking the trained machine. Validation data is used for tuning the trained machine with the help of important parameters.Training data is the data used to train an algorithm or machine learning model, is depicted in the given figure 1.

Table 1 The data table

Features x_j / Instances $s^{(i)}$	x_1	x_2	…	x_n	Decision y
$s^{(1)}$			…		
$s^{(2)}$			…		
⋮			…		⋮
$s^{(N)}$			…		

The above representation as given in Table 1, depicts the input as N instances, s(1), s(2), … S(N), each is an example of the concept to be learned. Each instances provides the input to the machine learning algorithm, and is categorized by its values mentioned as y, as in the last column. The data can be understood in more elaborated, as given below.

Four types of data are explained here, as it is often be handled in the process of dataset preparation or preprocessing. The data types are as given below.

- Numerical Data
- Categorical Data
- Time Series Data
- Text Data

Numerical Data

Numerical data is a datatype expressed in numbers. This further classified as continuous and discontinuous data as in Figure 2.

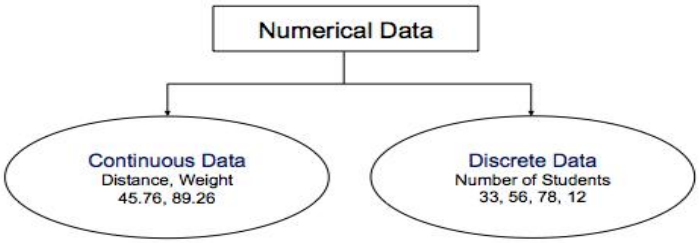


Figure 2 Numerical Data

Categorical Data

Categorical data is a collection of information that is divided into groups. They are further divided into two types such as ordinal and nominal.

Ordinal Data

Ordinal data has ranking / ordering. Ordinal features are sorted or ordered as in the figure 3.

Size of T-Shirt - S, M, L, XL.

Convert string values into integer as per order like XL > L > M > S.

Size	Encoded
S	0
XL	3
M	1
L	2

Figure 3 Ordinal Data

Nominal Data

- Nominal features are not ordered as in figure 4. Nominal data has No ranking / order.
- Colourof T-Shirt: Red, Green, Blue.
- Assign numeric value to each feature.
- 0 -> Red, 1 -> Green, 2 -> Blue

Figure 4Nominal Data

Time Series Data

A Time Series is a sequence of data points that occur in successive order over some period of time as in Figure 5.

Figure 5Time Series Data

Text Data

Text data usually consists of documents, which can represent words, sentences or even paragraphs.

Usually, digital information can be categorized into two classes. They are structured and unstructured. Studies have recently revealed that more than 70 percent of all the data available for corporations today is unstructured. But, structured data fits into a fixed format or data table, what we discussed in Table 1 above.

Preprocessing is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicated, or incomplete data within a dataset. Depicted in Figure 6. This is mainly focusing on dealing with missing data and handling categorical data.

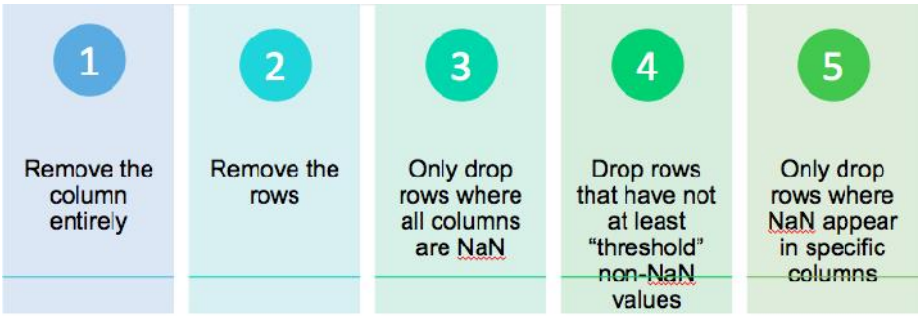


Figure 6Steps of Data Cleaning

Missing values can be handled through the simplest method called Imputation method. And, we should note that Removal of entire row/column may lose too much valuable data. Hence, we should avoid at maximum effort for the removal of rows and columns. Also, different interpolation techniques can be used to estimate the missing values like mean / median / mode imputation. This will increase the efficiency of data set. Sample model is shown in Figure 7.

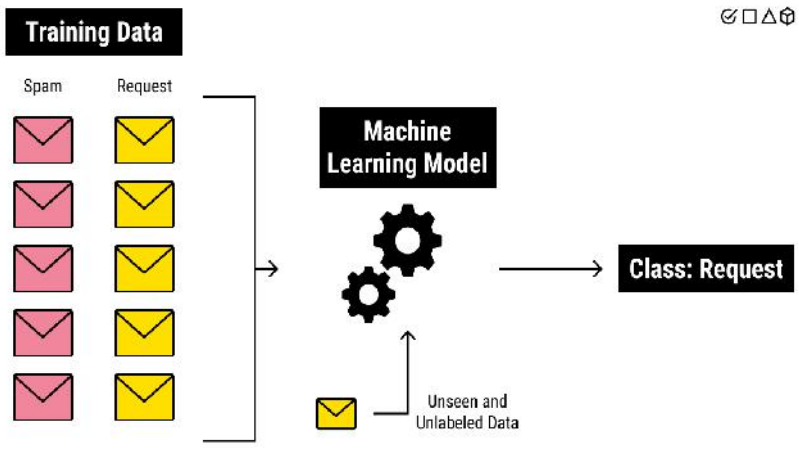


Figure 7 Machine Learning Model

Machine learning can be done in one of three approaches, that is, Supervised, Unsupervised and Reinforcement. These approaches are discussed below. Any method that incorporates information from experience in the design of a machine employs learning.

1.3 Supervised Learning

Machines are trained using well-labelled training data as in Figure 8. The data provided for training should be very much correct without any false data as the machine solely depending on the given data for its training. Means, the machine is designed by exploiting the a priori known information in the form of "direct" training examples consisting of observed values of system states / input vectors and the response / output to each states.

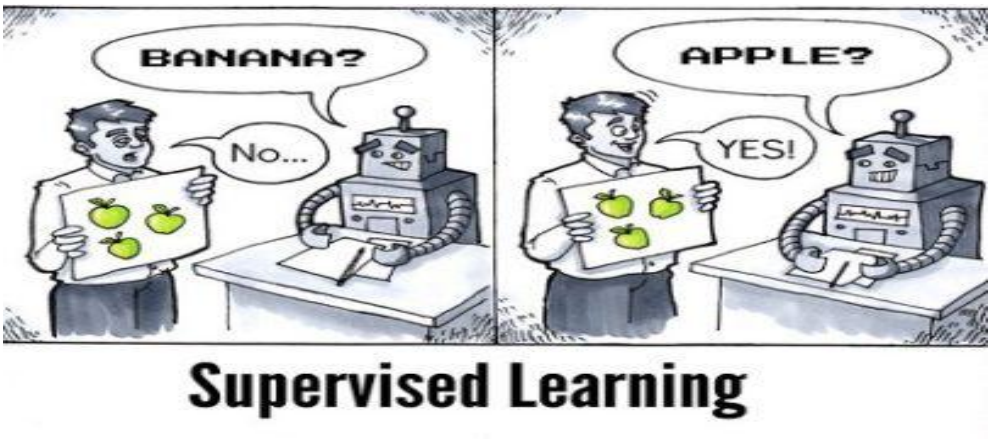


Figure 8Supervised Machine Learning

The supervisor has given in the form of given output / response. There are two types of tasks for which this supervisor can be used. They are classification / pattern recognition problems and the other one is regression / numeric prediction problems.

1.4 Unsupervised Learning

This approach of leaning will be used when the output / response is not available in the training data. This type of problem, uses only the set of feature vectors. This unsupervised learning has to unravel the underlying similarities in the training data for learning.

Two different types of learning tasks frequently appear in the real world applications of unsupervised learning. They are, cluster analysis and association analysis. The cluster analysis is employed to create groups or clusters of similar records on the basis of many measurements made for these records / feature vectors. A primary issue in clustering is that of defining the similarity between the feature vectors. It has many applications which includes big data analytics, remote sensing, image segmentation, image and speech processing and many more. Association analysis used unsupervised to discover patterns in the data where no target is specified earlier. It is up to human interpretation to make sense of the patterns. The common area of application is known as market basket analysis, which studies customer's purchase patterns for products that are bought together. This application is widely encountered in online recommender systems, where customers considering buying a product are shown other products that are often bought along with the desired products.

Figure 9Unsupervised Machine Learning

1.5 Reinforcement Learning

Reinforcement learning is founded on the concept tha if an action is followed by a satisfactory state of affairs, or by an improved state of affairs according to some properly defined way, then the inclination to produce that action becomes stronger. That is known as reinforced. This idea can be extended to permit the action choices to be dependent on state information, which then brings in the aspect of feedback.

A reinforcement learning system, is a system that via interaction with its environment enhances its performance by obtaining feedback in the form of a scalar reward or penalty (a reinforcement signal), that is indicative of the suitability of the response. The learning is not instructed with regard to what action has to be taken. Instead it is expected to find out which actions produce the maximum reward by trying them.

The actions may influence not only the immediate reward but also the next situation, and through that all subsequent rewards. The two aspects – trial and error search – and cumulative reward are the two significant distinguishing attributes of reinforcement learning. Even though the early performance may fail to be up to the mark, with adequate interaction with the environment, it will ultimately learn an effective strategy for maximizing cumulative reward.

The reinforcement learning problem covers tasks such as learning to control a mobile robot, learning to optimize operations in factories and learning to play board games. Reinforcement algorithms are related to dynamic programming algorithms frequently used to solve optimization problems.

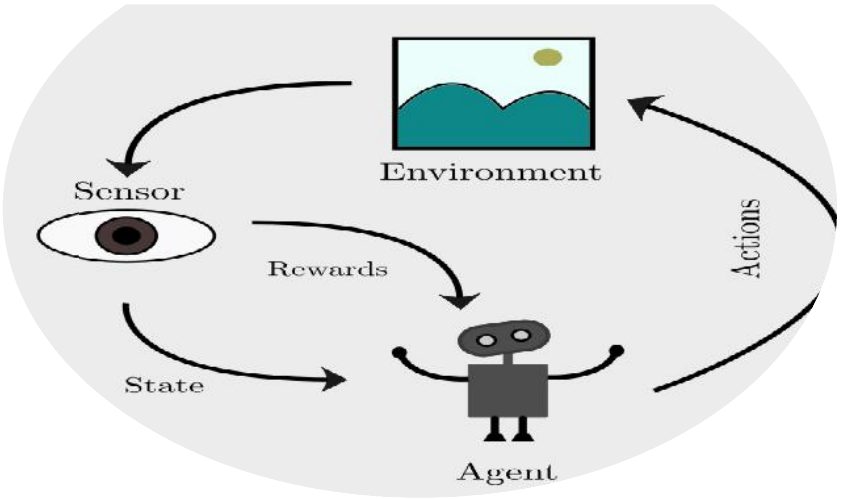


Figure 10Reinforcement Machine Learning

1.6 Applications of Machine Learning

- Machine vision: in this field where pattern recognition has been applied with major successes. A machine vision system captures images through a camera and analyzes these to be able to describe the image. A machine vision system is applicable in the manufacturing industry, for automating the assembly line or for automated visual inspection. Therefore, the images have to be analyzed online, and a pattern recognition system has to categorize the objects into the defect or non-defect category. A robot arm can then put the objects in the right place.
- Biometric Recognition: It has been made clear by decades of research in pattern recognition that the level of visual understanding and recognition that human exhibit cannot be matched by computer algorithms. Biometric recognition includes finger print identification, face recognition, gesture recognition and etc.
- Handwriting Recognition: It is the area with major consequences in automation and information handling. This problem focuses on handwritten documents. We know, the commercially available detector i.e., Optical Character Recognition (OCR). The typical application of handwritten recognition is processing the handwritten cheques given the bank customers. Another application can be an automatic postal mail sorting machines for postal code identification in post offices.
- Medical Diagnosis: This also uses pattern recognition. Doctors make use of it, while making diagnostic decisions. The doctor, of course, makes the ultimate diagnosis. Computer aided diagnosis has been applied to and is of interest for, a range of medical data. They may be, like X-Rays, computer tomographic images, ultrasound images, electro cardiograms (ECGs) and Electro encephalograms (EEGs).
- Drug Design: This is usually based on a long and expensive process involving complex chemical experiments to check whether or not a particular chemical compound could be a good candidature for a specific drug, which would be a positive result involving further chemical experiments. For several years, a new scheme based on computational simulations has been emerging. The general idea is to access the feasibility of a chemical compound for the synthesis of the drug with a predictive model based on a database of previous experiments.
- Speech Recognition: Speech is the most natural means by which humans share, convey and exchange information. This area has been well researched so far. Intelligent machines that recognize spoken information that can be used in numerous applications. For example, to help control machines by talking to them, entering data into a computer via a microphone. Speech recognition can enhance our ability to communicate with deaf and dumb.
- Text Mining: This concerns identification of patterns in text. The procedure involves analysis of text for extraction of useful information for specific purposes.The way, information available on the web and on corporate intranets, digital libraries and news wires is spread or propagated, is overwhelming. Integration of this information into the decision making process, at a fast pace, is essential in order to help business stay competitive in today's market. Text mining has reached the industrial world and is helping to exploit knowledge that is often beyond human consumption. Typical jobs for mining text databases are classification of documents into predefined classes, grouping together of similar documents and identifying documents that fulfill the criteria / specifications of a query.
- Natural Language Processing: Language is obviously a critical component of how people communicate and how information is stored in the business world and beyond. The goal of Natural language processing is to analyze, understand, and generate language that humans use naturally, so that eventually a computer will naturally be able to interpret what the other

person is saying. Spell checking, grammar checking, translation are the other applications of NLP and etc.,

- Fault diagnostic: Preventive upkeep of motors and generators and other electro-mechanical devices, can delay malfunctions. Otherwise the deviceswill interrupt industrial procedures.Hence, typical defects or flaws include misalignment of shaft, mechanical slackening, defective bearings, and unbalanced pumps. Diagnostic of faults are performed using machine learning algorithms, which is extremely helpful in this field.
- Business intelligence: Business intelligence technologies offer not only historical and current information but also predictive views of business applications. It is essential for businesses to be able to comprehend the commercial control of their organization, in term of customer base, market, supply and resources, and competition. In the absence of data mining, many businesses may be unable to effectively perform market analysis, compare customer feedback on similar products, find the strength and weaknesses of their competitors, retain extremely valuable customers, and arrive at intelligent business decisions.

Summary

In this unit, the concepts of machine learningare discussed along with the different approaches of machine learning. Each approach is discussed in detail with examples. The differences in each of the approaches would be better understood. Data set is very important for machine learning. Hence, it is necessary to understand about the basic data types, which is also explored thoroughly. This will help to convert or process the obtained data. But, there was also lot of challenges in processing the data set. This also covered in the name of preprocessing and data cleaning. The major tasks of preprocessing and the possible ways of data cleaning were also discussed. The terminology – feature engineering was highlighted as it was related to data cleaning.

Keywords

- Dataset
- Preprocessing
- Datacleaning
- Supervised learning
- Unsupervised learning
- Reinforcement learning

Self Assessment

1. Machine learning approach, which build a model based on sample data, is known as _____.

A. Supervised
B. Unsupervised
C. Reinforcement
D. None of the above

2. _____ approach uses the rewarding method for machine learning.

A. Supervised
B. Unsupervised
C. Reinforcement

D. None of the above

3. Which of the following dataset is used for supervised machine learning?
A. Training dataset
B. Testing dataset
C. Validation dataset
D. All the above

4. _____ machine learning approach uses unlabelled data for learning.
A. Supervised
B. Unsupervised
C. Reinforcement
D. None of the above

5. The two class problems are otherwise called _____.
A. Clustering
B. Binary Classification
C. Multiclass Classification
D. None of the above

6. Justify the statement. "Preprocessing is the process of converting raw data into data which will be suitable for machine learning".
A. True
B. False

7. Preprocessing is actually the combination of data cleaning and _____.
A. Data integration
B. Data transformation
C. Data reduction
D. All of the above

8. Supervised machine learning approachuses the _____ dataset.
A. Labelled Dataset
B. Unlabelled Dataset
C. Both Labelled and Unlabelled
D. None of the above

9. Imputation method is used for _____.
A. Removing rows
B. Removing columns
C. Filling the missing values
D. None of the above

10. _____ is the process of changing the format, structure or values of data.
A. Data integration
B. Data cleaning

C. Data transformation

D. Data Preprocessing

Answers for Self Assessment

1.	A	2.	C	3.	D	4.	B	5.	B
6.	A	7.	D	8.	A	9.	C	10.	C

Review Questions

1. Explain the different types of data.
2. Differentiatenominal and ordinal data types.
3. Give examples for categorical data.
4. List out the methods used for filling the missing values.
5. Identify the machine learning algorithms for each machine learning approaches.

Further Readings

- MadanGopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://www.javatpoint.com/types-of-machine-learning
- https://www.geeksforgeeks.org/ml-types-learning-supervised-learning/
- https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114

Unit 02: Python Basics

CONTENTS

Objectives
Introduction
2.1 What is Python?
2.2 Basics of Programming
2.3 IF Statement
2.4 IF – ELSE Statement
2.5 For Loop
2.6 While Loop
2.7 Unconditional Statements
2.8 Functions
2.9 Recursive Function
2.10 Other Packages
Summary
Keywords
Self Assessment
Answers for Self Assessment
Review Questions
Further readings

Objectives

1. To understand the online tools used for python such as JupyterLab and Google Colab.
2. To understandthe fundamentals of programming such as Variables, keywords, Data types, Expression, Statements, Operator and Operator Precedence.
3. To differentiatethe conditional and unconditional statements from simple if, if-else, nested if, for loop, while loop, break and continue.
4. To understand the use of function and recursion which will bediscussed with examples.
5. To know the packages in python along with their purposes.

Introduction

In this unit, we try to introduce you the very popular programming language called Python. We know that there are many programming languages such as C, C++, which were already existed and used for decades. Here, we will try to understand the merits of python language over others. Moreover, we will be writing a simple python programand try to execute using an online tool. Programs can be experimented to understandthe conditional, unconditional statements along with functions and recursion. Function declaration, calling of functions, parameters can be well understood from the given examples. Let us begin with what is python.

2.1 What is Python?

The python language is founded and written by Guido van Rossum, who was born in the Netherlands. His First version (0.9.0) of python was released on February 20, 1991. Presently the

current version of python is 3.9.7. There are many reasons for its popularity. It is readable like an English statement having simple syntaxes. Python is a general-purpose open source language. Python is portable language, so that it runs on many Unix variants including Linux and mac OS, and on Windows. Python is also an interpreted language, interactive language and object-oriented programming language. Python codes are executed comparatively little faster.

Figure 1 Founder of Python Language

The language offers multiple ready made libraries such as NumPy, SciPy, MatplotLib, Scikit-Learn and frameworks which will support the initial phase of development. These are all the reasons made Python very popular among programming community.

Who are using python?

There are many MNC companies were started using python for their development. Here are the few such companies and their works are given for better understanding of python language. We will start with Google company. Google uses Python for web search systems, YouTube video sharing service is largely written in Python. Dropbox company's storage service codes both its server and desktop client software primarily in Python.EVE Online, a massive multiplayer online game(MMOG) by CCP Games, uses Python. The widespread BitTorrent peer-to-peer file sharing system began its life as a Python program. iRobot uses Python to develop commercial and military robotic devices.Netflix have documented the role of Python in their software infrastructures. Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM use Python for hardware testing.JPMorgan Chase, UBS, Getco, and Citadel apply Python to financial market forecasting.NASA, Los Alamos, Fermilab, JPL, and others use Python for scientific programming tasks. Now, let us understand, why we are importance to the python language.

Online Tools

Jupyter Notebook and Google Colab are the popular online tools for programming in python. Let us first discuss the Jupyter. You will get the access in this link: https://jupyter.org/try. Once you have visited the page, it will look like the figure 2 given below.

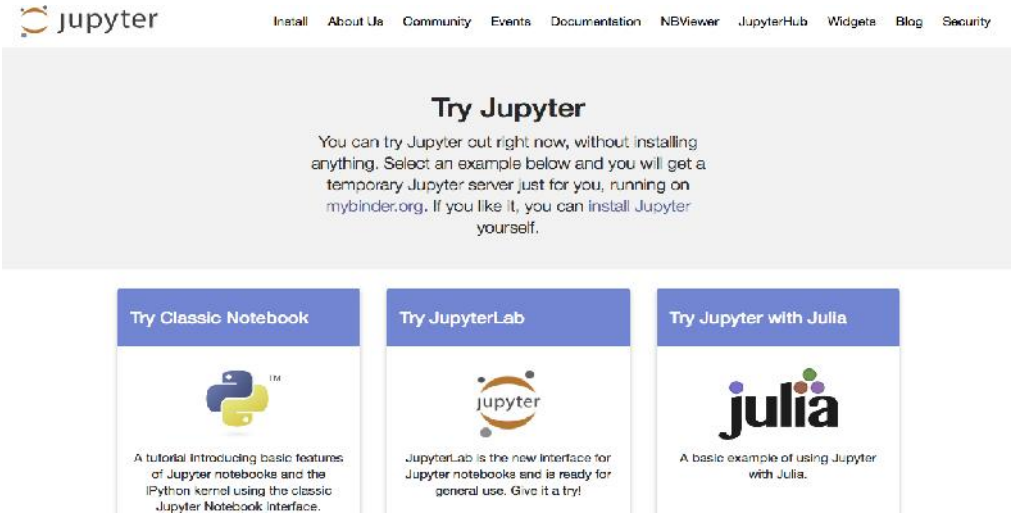


Figure 2 Home page of Jupyter

Now please select the first blue color button "Try Classic Notebook". The page will be looking like like the figure 3 given below.

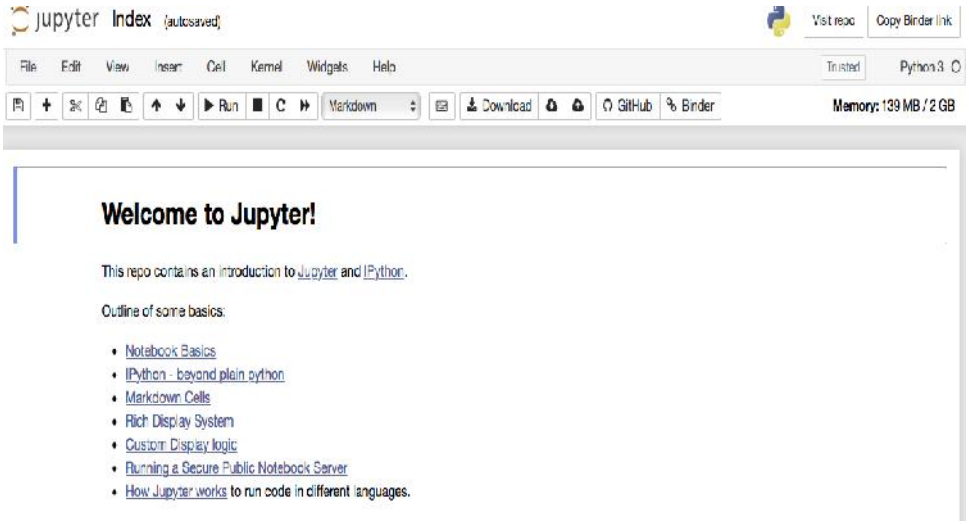


Figure 3 The page of Jupyter Classic Notebook

Then, you need to create a new python notebook from the file menu. The page will be looking like the following figure 4.

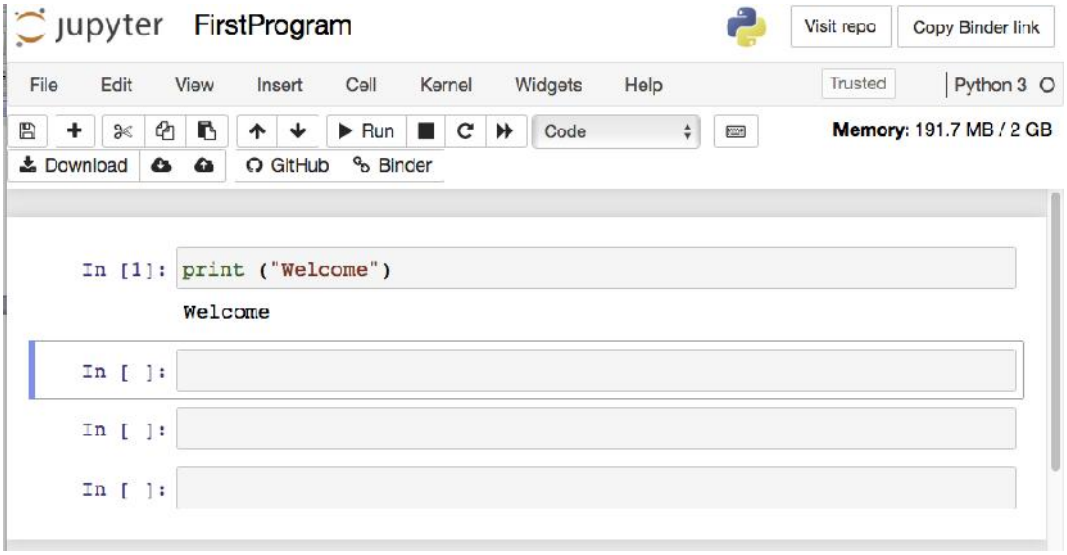


Figure 4 New Python Notebook

Now, we are ready to go with our first program. Simple print statements is tested in the above notebook, just outputting the message "Welcome".

Python Installation

Python softwares and the installation manual can be downloaded from the link https://www.python.org/downloads/. You will be shown the opt version with respect to your system configuration and operating system. Latest version will be Python 3.9.7 as shown in the figure.

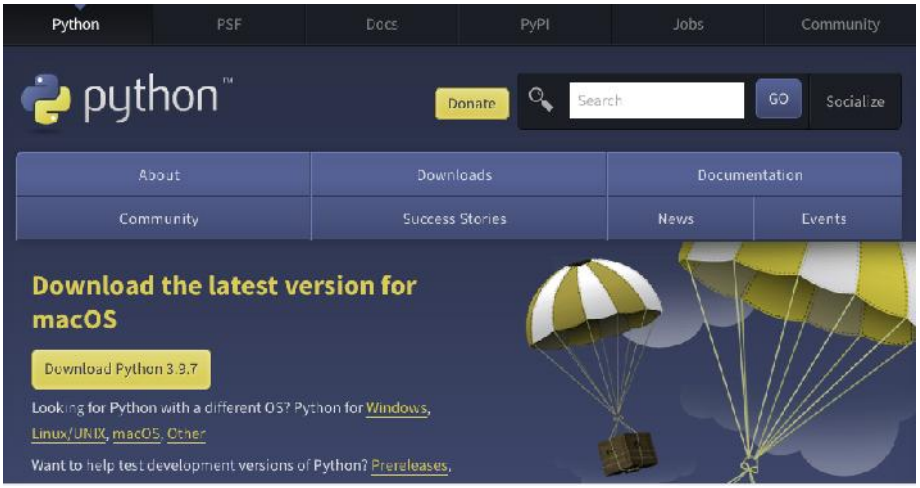


Figure 5 Python Downloads

Just a click is enough to download. And the procedures are simple for the installation.

2.2 Basics of Programming

The basics start with the simple detail that how to store some information in the program, how to do some mathematical operations and how to display them in the monitor. The following are very important for programming. Let us see one by one.

Variables

Variable is a name that refers to a value that may be changed in the program.There is no command to declare a variable in python.

```
In [10]: a = 100

In [11]: print (a)
         100

In [12]: b = "Mangolia"

In [13]: print (b)
         Mangolia
```

Figure 6 Use of Variables

In the above figure, two variables were declared namely a and b having the values 100 (numeric value) and “Mangolia” (string value) respectively.The data type of the variable will be same asof the data type of the assigned value. Those values are also displayed using the print statement.

Datatype

Variables can store different types of data. They are Numeric Data Types, Boolean Data Type, Set Data Type, Dictionary and Sequence Data Types as shown in the figure. Let us start with Numeric Data Types, where only the numbers are involved, whichis again divided into three categories such as integer (without decimals), float (with decimals)and complex numbers.Boolean data will be storing True or False values.Others will be studied in the coming units in detail.

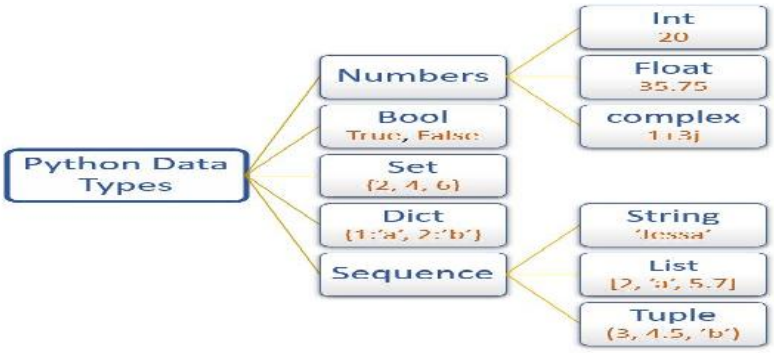


Figure 7Python Data Types

Keywords

There are some predefined and reserved words have special meaning to python. Those words are called Keywords. These keywords can not be used as name for the variable / identifier, not to be used as function names. These are otherwise called as system defined variable. There are more 30 keywords used in python. Few keywords are like, and, or, not, if, elif, else, for, while, break, return, True, False, continue, in, is, import and etc.

Expression

An expression is a combination of values, variables and operators. This can be understood from an example given in the following figure. There are three variables a, b and c. The variables are known as operands. Operators are used in between to perform some operations using the operands. In this case, plus (+) operator, multiplication (*) operators are used. Final result is stored in the variable result.

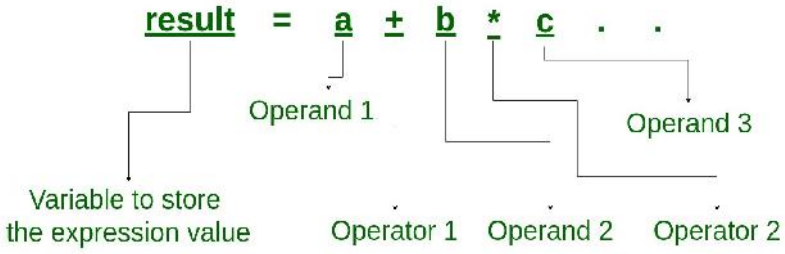


Figure 8Statement of Expression

Let us have an example for an expression. Let a = 10, b=5, c = 3. What will be the output of the above expression?The output is 25 as shown in the figure.

```
In [1]: a=10
In [2]: b=5
In [3]: c=3
In [4]: result = a + b * c
In [5]: print(result)
        25
```

Figure 9Example of Expression

Statements

Statements are the instructions given in the source code for execution.The outcome of the program is depending upon how all the statements are arranged for execution. The statements are executed in a sequential order starting from the first statement in program. There are three types of statements in python. They are Assignment statements, Conditional Statements and Looping Statements, which are discussed below.

Assignment statements

The statements that are used to copy a value into the variable is called assignment statements. The equal sign (=) is used for copying the value. Hence, the operator (=) is called assignment operator. The target of an assignment statement is written on the left side of the equal sign (=). The value what is to be assigned will be in the right side of the equal sign (=).

For example, a = 100 is the assignment statement. Here the value 100 is assigned to the variable a. And, we have one more example like this. x, y = 50, 100 is also the assignment statement, where the value 50 is assigned to variable x and the value 100 is assigned to variable y.

Conditional statements

Any statementthat outputs the Boolean value (True / False) is called conditional statement as given in the figure. Framing of conditions is the key element in controlling the flow of execution. Let us have an example of conditional statement.

Example 1 : (a < b)

Example 2 : (marks >= 75)

Above two examples will be giving either True or False as the output. Hence, they are called conditional statements.

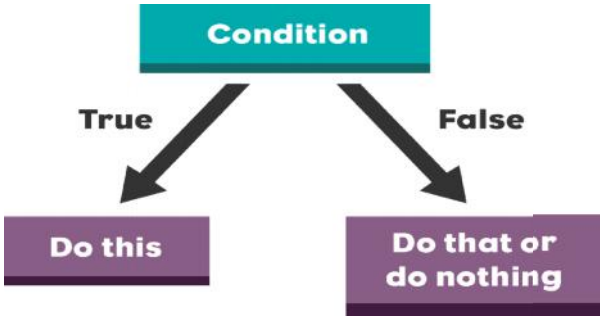


Figure 10 Flow of Control

Looping statements

The looping statement(s) are a statementor a block of statements that are used to execute repeatedly until a specified condition is satisfied. When the condition is True, it executes and when the condition is False, it stops the execution.If the execution is not getting stopped, then the looping statement will become infinite loop.There are different looping statements are available such as while loop and for loop.

Operator

The operators are used to perform some mathematical operations on the values and the variables. There are few standard symbols available in python. Let us have a look on list of operators and their usages. According to their usages, all the operators are grouped in different categories such as arithmetic operators, Relational Operators and logical operators.

Arithmetic Operators

Operator	Description	Syntax
+	Addition	x + y
-	Subtraction	x - y

/	Division (float)	x / y
//	Division (floor)	x // y
%	Modulus	x % y
**	Power	x ** y

Table 1 Arithmetic Operators

Relational Operators

These operators are used to compare the values or variables. The output of relational operators will be either True or False.

Operator	Description	Syntax
>	Greater than	x > y
<	Less than	x < y
==	Equal to	x = y
!=	Not equal to	x != y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Table 2 Relational Operators

Logical operators

Logical operators are used to combine two or more conditional statements. The operators are Logical AND, Logical OR and Logical NOT.

Operator	Description	Syntax
and	Logical AND	x and y
or	Logical OR	x or y
not	Logical NOT	not x

Table 3 Logical Operators

The logical AND operator outputs the value True if all the conditions are True. But, Logical OR operator outputs the value True if anyone the condition has the value True. Logical NOT operator reverse the actual output of the condition such as converts as False if the actual value is True and vice versa.

Assignment operators

These are used to assign the values to the variable. This operator was already discussed in the topic of Assignment Statements. Here, let us have the list of other operators used for assignments operation.

Operator	Description	Syntax
=	Assign value to left side operand	x = 100
+=	This is simplification of a = a + b	a += b

-=	This is simplification of a = a - b	a -= b
*=	This is simplification of a = a * b	a *= b
/=	This is simplification of a = a / b	a /= b
%=	This is simplification of a = a % b	a %= b
//=	This is simplification of a = a // b	a //= b

Table 4 Assignment Operators

Operator Precedence

It is understood that an expression is having one or more operators and operands having simple or complex mathematical operations. Two operands are needed for an operator to perform the specified operation. Hence, some order of preference or priority is required to select the operators to compute the expression. This is called as operator precedence. It can be understood from a simple expression as shown in the figure.

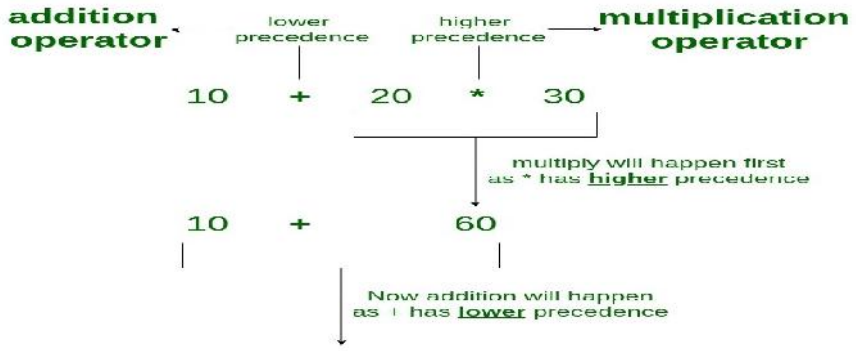


Figure 11Example of Operator Precedence

In the above figure, the expression first computes the multiplication operator and then the result of the multiplication is used for the next computation of addition operator. Here, multiplication operator is having the higher priority than the addition operator. Similarly, there are many other operators are available as we know. It is important to know their precedence so that we can use them correctly as per our requirements in the expression. Following figure is trying to give you the clear picture of precedence; the higher priority starts from top and reaching the lowest priority in the bottom.

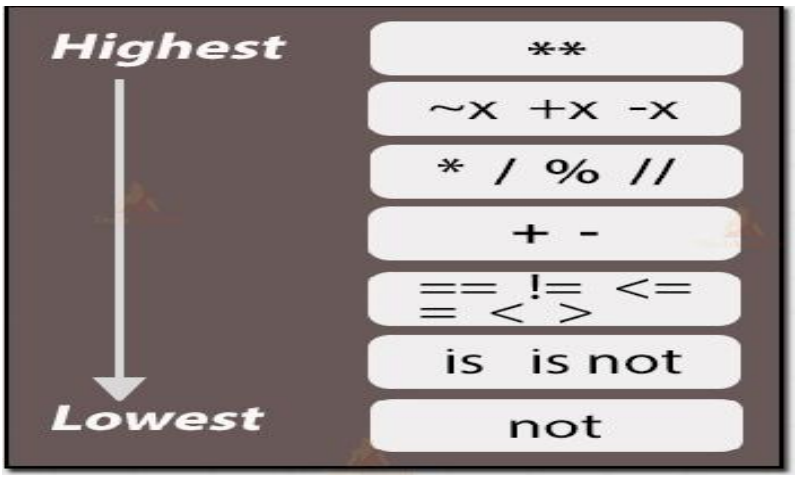


Figure 12 Priority Structure of Operators Precedence

2.3 IF Statement

The flow of execution in a program can be controlled using the proper conditions. Here, is the simple conditional statement called if statement. This is used to execute a block of statements if the condition is True and to execute the next statements if the condition is False. Hence, the flow of execution lies in the decision making as in the figure.

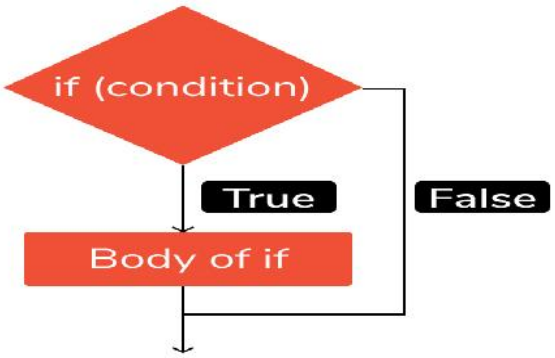


Figure 13 Working with simple IF Statement

Here is an example depicting the simple if statement. The example says it gives the output as "Eligible for final examination" if the attendance is greater than equal to 75 percentages, otherwise do nothing and proceeds with the further instructions.

```
attendance = 90
if (attendance>= 75):
        print ("Eligible for final examination")
```

Here the variable attendance is having 90. Hence, condition is satisfied and the print statement is executed successfully. Let us have the same example with the attendance value as 65.

```
attendance = 65
if (attendance>= 75):
        print ("Eligible for final examination")
```

What will be the output? Of course, it doesn't print anything as there are no statements given when the condition becomes False. This can be solved in the next type of conditional statement.

2.4 IF – ELSE Statement

Previously, the simple if statement is discussed. That is having only one part means that we are giving the instructions where the condition is True. Here in the if-else statement, we are going to give the statements for both the sections. One block of statements will be executed when the condition is true and another block of statements will be executed when the condition is False. Anyway, at a time, only one block is executed for sure.The figure shown below is explaining the concepts clearly.

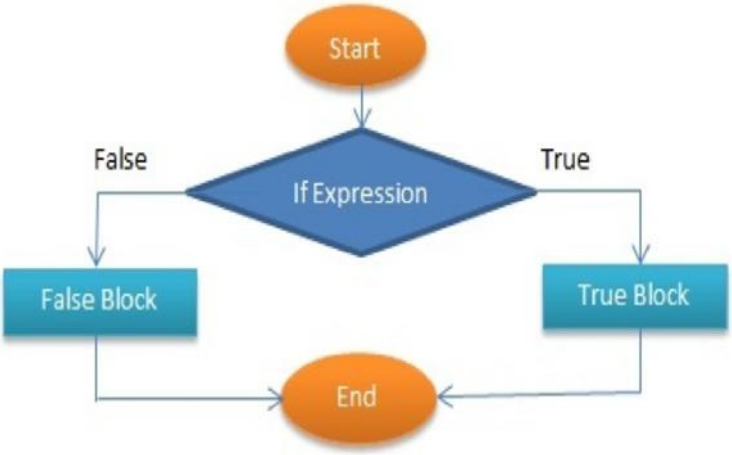


Figure 14 Working with IF Else Statement

If the condition (if expression) is False, the control is going to false block and reaches the end. Similarly, if the condition is false, the control is going to true block and reaches the end. Let us discuss this using an example.

x = 6

y = 8

if (x>y):

print ("x is greater than y")

else:

print ("y is greater than x")

Here, let the variables x and y has the values 6 and 8 respectively. Now, concentrate onthe if statement and find out the value of condition. (x > y) ➔ (6 > 8) ➔False. The control goes to the False Part and outputs "y is greater than x". In case, if we change the values as x= 8 and y=6. What will be the output?

x = 86

y = 70

if (x>y):

print (" x is greater than y")

else:

print ("y is greater than x")

yes, we will get the output as "x is greater than y" as the condition becomes True and True Part has been executed.

2.5 For Loop

We know that this is one of the looping statements. For loop helps us to execute a particular block of statements repeatedly for a certain number of times. The figure clearly picturize the style of execution. Number of times can be controlled through the variable. Starting values and Ending Values are the keys for the control. Use of the variable, can be effectively managed by incrementing or decrementing the values according to our requirements.

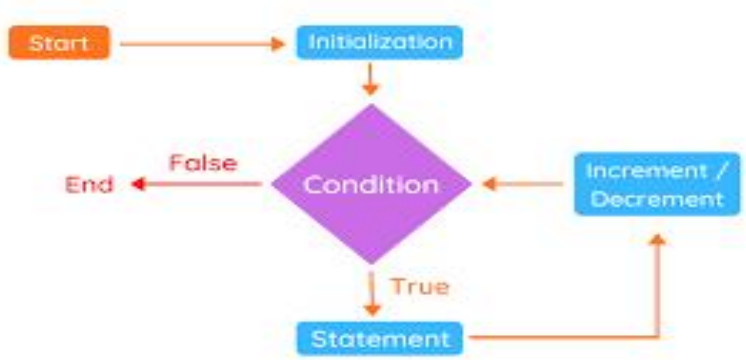


Figure 15 Working with for Loop

On every successful execution of the statement, it goes back to the condition after increments/decrement the variable and checks the condition still satisfied or not. If satisfied, it continues to execute the statement again and repeat the same till the condition is failed. Let us have an example.

for t in range (5):

print (t)

The above example is having the variable t. The initial value will be 0 in this case. The last value will be 4. Before explaining this, let us take care of what is range function.

This function will create the sequence of numbers in a given range starting from 0 by default. In this, having range (5) will give values from 0 to 4.This is making the sequence of numbers easy.

Now, let us focus on the output of the for loop statement given above. Here, print (t) is executed five times.

Output:

0

1

2

3

4

2.6 While Loop

This is also the control statement, whichis used to execute a set of statements till the given condition is True. On the completion of every execution, it checks for the condition is True or Not. If the condition fails or the condition is False, the looping is terminated and goes to the successive statements. The following figure explains the flow of execution.

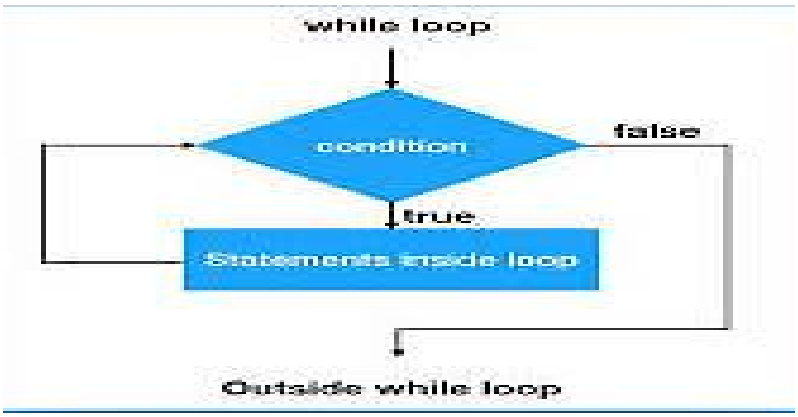


Figure 16Flow chart of While Loop

Let me explain you from a simple example below.The variable used is number having initial value as 0. Here, the program aims to execute a block of statement till the variable number is not equal to 8, It means it never knows how many times it is going to execute the block of statement. We need to manage this condition as per our requirements. Here, the task is simple.

```
number=0
while number != 8:
    number = number + 1
    print ("the number is", number)
```

Figure 17Example of While Loop

Initially, the value of number is 0. The condition (0 not equal to 8) is True. It executes the block of statements first time. Now, the value of number is increased by 1 and it becomes number = 1. Now, it checks the condition again. The condition (1 not equal to 8) is True. It executes the block of statements second time and it goes on till the condition is True. Stops otherwise.

2.7 Unconditional Statements

The statements, which are not using any condition to controlthe flow of execution, are called as unconditional statements. There are two unconditional statements used in python, i.e., break and continue. We discuss these in detail.

Break statement

This statement is used to stop the current execution. There is no need to give any condition for this break statement. This statement is used whenever you need to stop or whenever you find any exceptions during the execution. The usage of break statement is understood from an example given below.

```
x = 100
while ( x < 600) :
print (x)
if ( x == 300) :
  break
x = x + 100
```

There is a variable x in the example having the initial value as 100.While-loop executes till the value of x is less than 600. On every execution, the value of x is incremented by 100. But, as there is a break statement, planned to be executed exactly when x = 300. Hence, it stops the current execution at that point and never continues the loop further.

Continue Statement

This statement is used to continue to the next iteration (loop) without executing further statements in the current iteration (loop). There is no need to give any condition for these continue statement. This statement is used whenever you need to avoid the further statements and want to execute the next iteration. The usages of continue statement is understood from an example given below.

```
x = 100
while (x< 600):
print (x)
if (x == 300):
continue
```

x = x + 100

There is a variable x in the example having the initial value as 100. While-loop executes till the value of x is less than 600. On every execution, the value of x is incremented by 100. But, as there is a continue statement, planned to be executed exactly when x = 300. Hence, it stops the current execution at that point and continues to the next iteration(loop). Means, once the continue statement is executed, further statementsi.e., increment will not be executed and the value of x remains as 300. So, the loop will further become infinite loop as the condition will not become False at any case, as there is no increment of value of x.

2.8 Functions

Function is a block of statements, whichis executed only when it is called. Function is given a specific name. We can use that name whenever we want to call that function. Function can be divided into twotypes. They are system-defined function and user-defined function. Let us focus more on the user-defined function. Declaration of function and calling of functions are discussed here. The following figure represents the usage of passing a value (x) to the function (f) and getting the output. Function is called by sending the value of x. Function is using the value of x and performing the computation. The result is sent back as the final output.

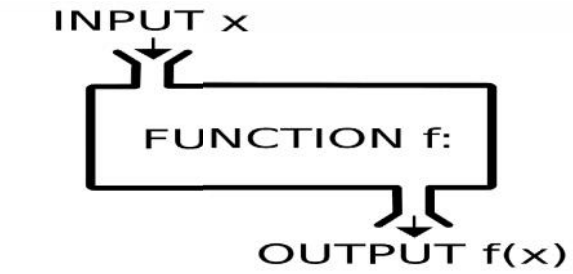


Figure 18Block diagram of a function

We know that a function that you define yourself in a program is known as user defined function. The function definition and declaration is well understood from the example given below. We are defining a function using the keyword "def". The name of the function is "fahr_to_celsius". This function accepts only one parameter i.e., temp. The computed value is returned from the function using the keyword "return" as given in the figure. This function does the converting the temperature into Fahrenheit.

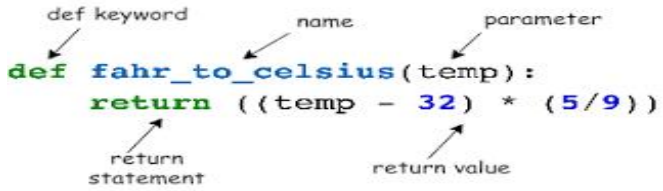


Figure 19 Example of Function definition

Here is one more example, which accepts two parameters a and b. Addition of giventwo numbers is performed in the function. The function name is my_fun. We are not returning anything from the function.Sothere is noreturn statementexist in the function.

```
def my_fun(a, b):
    c = a+b
    print("Sum of a and b is: ", c)
```

Figure 20 Example of Function with two parameters

The values or variables given in the function definition are called as parameters. At the same time, the values or variables given in the calling function is called and arguments. The number of parameters should be matching with the number of arguments that are passed to the function. The order of passing the values is very important in passing into the parameters.

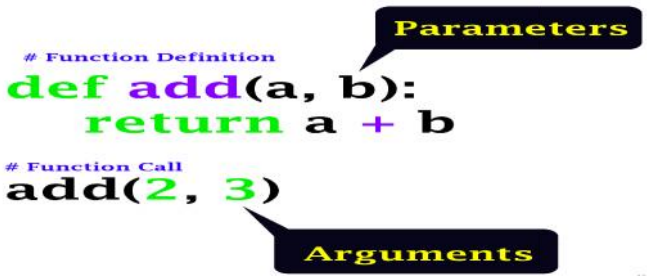


Figure 21 Parameters and Arguments

Once it is defined, the function cannot be executed until it is called by its name. That's why the statement, which is used to call any function, is known as calling function. From the figure given below, we can understand that the function is called by the statement myfun(3, 4). Parameters are passed during calling the function. It is assumed as 3 is going to variable x and 4 is going to variable y. Multiplicated value (3 x 4 = 12) is returned as the final output.

```
>>> def myfun(x, y):
        return x * y
>>> myfun(3, 4)
12
```

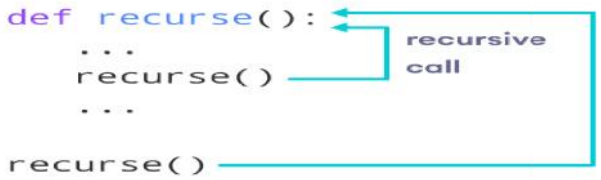
System Defined Function

Python is having lot many built-in functions whose functionality is already defined by the system is known as system defined function. Some of the system-defined functions are math functions such as math.pow( ), math.exp( ) and also the print( ) are the system-defined functions.This is otherwise called as library function or standard function or pre-defined function.

2.9 Recursive Function

A recursive function is a type of function that calls itself during its execution until the condition is satisfied, which is given within the function. We can give our own condition as per the requirement to finish the task. The flow of execution can be understood from the figure given below.

In this figure, the function name is recurse ( ). The function may be called using the calling function externally. Once the function is called, the subsequent calling of function is done within the function internally.



Here, we can see an example where we are going to calculate the factorial of a given number say n. The function name is fac ( ) . The recursive function is the one which is getting called by itself. Hence, you are using the same function name wherever you need to call the function. The recursive will go until you reach n = 0.

```
def fac(n):
    if n == 0:
        return 1
    else:
        return fac(n - 1) * n
```

The output of the above recursion function is understood in the following manner.

```
= 5 * factorial(4)
= 5 * 4 * factorial(3)
= 5 * 4 * 3 * factorial(2)
= 5 * 4 * 3 * 2 * factorial(1)
= 5 * 4 * 3 * 2 * 1
= 120
```

2.10 Other Packages

In python, thereare four important packages available for the effective programming and data science in particular. The packages are NumPy, SciPy, Matplotlib and Pandas. NumPy is used for all the common numerical calculations, SciPy will be used for scientific calculations such as sin(theta), cos(theta), exp(x) and so on. Pandas are used for reading the files or dataset and manipulating them such as updating the data or removing the unnecessary data row-wise and column-wise. Matplotlib is used for graphical representation of the given data for better understanding before working on them.

Summary

- The fundamentals of python programming such as Variables, keywords, Datatypes, Expression, Statements, Operator and Operator Precedence were discussed.
- Understood how to write a simple python program in the online tools such as JupyterLab and Google Colab.
- We could able to differentiate the conditional and unconditional statements. Illustrated with examples.
- The usage of simple functions and recursion functions were discussed with examples.
- Few real time applications also elaborated here to understand the popularity of python.

Keywords

- Python
- Jupyter
- Colab
- Operators
- Functions
- Packages

Self Assessment

1. What is the output of the following code?

   x = 10 // 3

```
print(x)
```

A. 0
B. 1
C. 2
D. 3

2. What is the output of the following code?

```
x = 12 % 3
print(x)
```

A. 0
B. 1
C. 2
D. 3

3. What is the output of the following expression?

```
x = 100 + 50 - 20 / 2 * 5
print(x)
```

A. 325
B. 148
C. 0
D. 100

4. What type of loop exist in the givencode?

```
for i in [2,5,6,7]:
print (i)
```

A. Finite Loop
B. Infinite Loop

5. What is the outcome of the following for loop statement?

```
sum = 0
for i in range(5):
  i = i + 5
  sum = sum + i
print(sum)
```

A. 15
B. 0
C. 35
D. 45

6. What is the output of the following for loop?

```
for i in range (0, 5, 2):
print(i)
```

A. 0, 2, 4, 6

B. 0, 1, 2, 3, 4
C. 0, 2, 4
D. 1,2,3,4,5

7. How many times the print statement is executed in the givencode?

```
x = 5
while (x <= 5):
  print(x)
```

A. 1 time
B. 5 times
C. Infinite Loop
D. None of the above

8. How many times the print statement is executed in the given code?

```
x = 10
while (x <= 10):
  print(x)
  x = x - 2
```

A. 5 times
B. 10 time
C. Infinite Loop
D. None of the above

9. How many times the print statement is executed in the given code?

```
x = 10
while (x <= 10):
  print(x)
  continue
  x = x - 5
  break
```

A. 10 times
B. 2 times
C. Infinite Loop
D. None of the above

10. How many times the print statement is executed in the given code?

```
x = 10
while (x <= 10):
  print(x)
  break
  x = x - 5
  continue
```

A. 1 time
B. 5 times
C. 10 times
D. Infinite Loop

11. What is the output of the followingcode?

```
x = 15
y = 20
if ( not(x > y) and (y > x) ):
   print("All good")
else:
   print("All bad")
```

A. All good
B. All bad
C. Both (A) and (B)
D. None of the above

12. What is the output of the followingif-elif codes?

```
x=75
if (x >= 90):
   print("Grade O")
elif (x >= 80):
   print("Grade A")
elif (x >= 70):
   print("Grade B")
elif (x >= 60):
   print("Grade C")
```

A. Grade O
B. Grade A
C. Grade B
D. Grade C

13. What is the output of the following nested-if code? Ans: Grade B

```
x=90
if (x >= 90):
   print("Grade O")
else:
   if (x >= 80):
      print("Grade A")
   else:
      if (x >= 70):
         print("Grade B")
```

```
else:
    if (x >= 60):
        print("Grade C")
```

A. Grade O
B. Grade A
C. Grade B
D. Grade C

14. Assume the function already defined as given below.

```
def MyFunc(n):
    sum = 0
    for i in range(n):
        sum = sum + i
    return (sum)
```

What is the value of x if you execute the following code?

```
x = MyFunc(10)
```

A. 0
B. 45
C. 55
D. None of the above

15. Assume the function already defined as given below.

```
def MyFunct(x, y):
    x = x + 10
y = y + 10
    return(x,y)
```

What is the value of t if you execute the following code?

```
t = MyFunct(10,20)
```

A. (10, 20)
B. (20, 30)
C. Nothing is printed.
D. None of the above

Answers for Self Assessment

1.	D	2.	A	3	D	4.	B	5.	C
6.	C	7.	C	8.	C	9.	C	10.	A
11.	A	12.	C	13.	C	14.	B	15.	B

Review Questions

1. Explain the Datatypes and their functionalities.
2. Differentiate conditional and unconditional statements. Give the respective name of the statements.
3. Illustrate finite and infinite loop. Give reasons for getting infinite loop.
4. How do you receive the output from the function? Explain with an example.
5. Why do you need Recursive Function? How it stops the recursive operation.

Further readings

- John Zelle, "Python Programming: An Introduction to Computer Science", Second Edition, Franklin, Beedle and Associates Inc, 2009.

Web Links

- https://docs.python.org/3/faq/general.html
- https://docs.python.org/3/tutorial/index.html
- https://www.python.org/downloads/
- https://jupyter.org/try
- https://colab.research.google.com

Unit 03: Data Pre-Processing

CONTENTS

Objectives
Introduction
3.1 Introduction to Data Analysis
3.2 Importing the data
3.3 Summarizing the Dataset
3.4 Data Visualization
3.5 Exporting the data
3.6 Data Wrangling
3.7 Exploratory Data Analysis (EDA)
Summary
Keywords
Self Assessment
Answers for Self Assessment
Review Questions
Further readings

Objectives

- To understandthe concepts of Data Preprocessing and Data Analysis.
- To understandthe fundamentals of datasetand downloading from the website.
- To understand the python code for preprocessing of data.
- To understand the process of data wrangling with examples.
- To know the different aspects of exploratory data analysis.

Introduction

Data preprocessing is a process of preparing the raw data and making it suitable for a machine-learning model. It is the first and crucial step while creating a machine-learning model because the real world data generally contains noises, missing values and may be in an unusable format, which cannot be directly used for machine learning model. Hence, the data preprocessing is required tasks for cleaning the data and making it suitable for a machine-learning model, which also increases the accuracy and efficiency of a machine-learning model.In this unit, we will discuss and understand the fundamentals of data preprocessing and the necessary steps and approaches in doing the preprocessing. Also, we explore the concept of data analysis and we try to understand how the data wrangling and exploratory data analysis helps for effective data preprocessing.

3.1 Introduction to Data Analysis

Data analysis plays a crucial role in processing data in making them as useful information. Data analysis is the process, which includescleaning the data, changing the data and processing raw data and extracting relevant information that helps for machine learning. We cannot preprocess effectively unless we understand better about the data given to us. Hence, the data analysis became

important in the approach of data preprocessing. The process of data analysis consists of the following steps.

- Gathering the Requirement for Data: This helps you to decide the need for the data, what type of data you want to use, and what data you plan to analyze.
- Data Collection:It's time to collect the data from your sources. Data collection will be done from your identified requirements.
- Data Cleaning: It's time to clean up the collected data. Assume that some of your collected data is useful and some of data is not useful. The cleaning techniques are given in Fig 1. This process is where you remove white spaces, duplicate records, and basic errors. Data cleaning is mandatory before sending the information for data analysis.

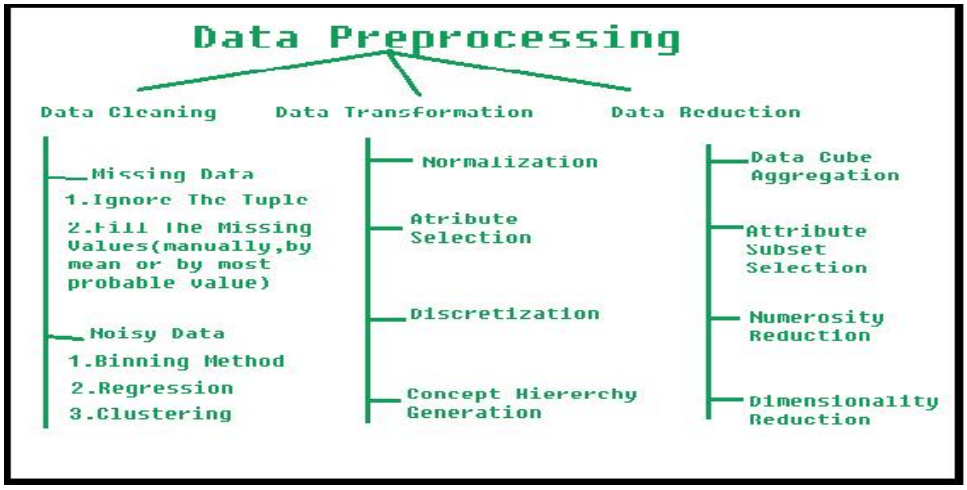


Fig 1. The approaches of data preprocessing

Data Analysis: You will understand the data better and better in all the possible ways. You can also use tools, which includes Excel, Python, R, Rapid Miner and etc. You will create many graphs / diagrams as the outcome of your data analysis.

Data Interpretation: Assume that you have your data analysis results. Now, you need to interpret them and come up with the best courses of action based on your findings.

Data Visualization: This is a fancy way of saying like "graphically show your information in a way that people can read and understand it." You can use charts, graphs, maps, bullet points, or other methods.

There are few more types, which are commonly used in the worlds of technology and business and the same is given below.

Diagnostic Analysis: This answers the question, "Why did this happen?" Ideally, the analysts find similar patterns that existed in the past, and consequently, use those solutions to resolve the present challenges hopefully.

Predictive Analysis: This answers the question, "What is most likely to happen?" By using patterns found in older data as well as current events, analysts predict future events.

Prescriptive Analysis: Mix all the insights gained from the other data analysis types, and you have prescriptive analysis. Sometimes, an issue can't be solved solely with one analysis type, and instead requires multiple insights.

Statistical Analysis: This answers the question, "What happened?" This analysis covers data collection, analysis, modeling, interpretation, and presentation using dashboards.

Text Analysis: Also called "data mining," text analysis uses databases and data mining tools to discover patterns residing in large datasets. It transforms raw data into useful business information. Text analysis is arguably the most straightforward and the most direct method of data analysis.

Although there are many data analysis methods available, they all fall into one of two primary types of data analysis. They are qualitative analysis and quantitative analysis. The qualitative data

analysis method derives data via words, symbols, pictures, and observations. This method doesn't use statistics. But, the Quantitative Data Analysisproduces different numbers as the result of data analysis with the help of statistical methods. Statistical data analysis methods collect raw data and process it into numerical data.

3.2 Importing the data

For our practice, we can load the data directly from the UCI Machine Learning repository (https://archive.ics.uci.edu/ml/index.php). This data /dataset can be imported into python using pandas as shown below. The dataset downloaded is known as "Pima Indian Dataset" using the following steps.

Step 1 Declaring the Pandas Library

Step 2 File is assigned to a variable name

Step 3 Assigning the Columns Names or Column Headings.

Step 4 Importing the PIMA Dataset (File Name is pima_indians.csv)

Observe the following code for importing the dataset using python.

```
import pandas
data = 'pima_indians.csv'
names = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'Outcome']
dataset = pandas.read_csv(data, names=names)
```

3.3 Summarizing the Dataset

The dataset may be understood by the following observations. It is also known as summary of the dataset.Observations are given in the bulletins.

- Basic Information about the dataset is obtained from the following code.

  ```
  print(dataset.info ())
  ```

- Dimensions of Dataset can be obtained using the following code.

  ```
  print(dataset.shape)
  ```

- Listing all top 10 data, the following code helps.

  ```
  print(dataset.head(10))
  ```

- Listing all bottom 10 data, the following code helps.

  ```
  print(dataset.tail(10))
  ```

- View the Statistical Summary from this code.

  ```
  print(dataset.describe())
  ```

3.4 Data Visualization

Data visualization is the process of representing data using visual elements like charts, graphs, etc. The data can be better understood if we provide and summarize using the beautiful diagrams, which is known as data visualization. The sample for the visualization is given in Fig 2. Generally, there are two types of plots exists and used for data visualization. They are univariate and multivariate.

Let us explore the first type of visualization i.e., univariate plots.

Fig 2 A Sample Data Visualization

Univariate Plots

Here, 'uni' means one and 'variate' indicates a variable. Therefore, univariate plot is a form of diagram / graph that only involves single variable.Example is given in Fig 3.

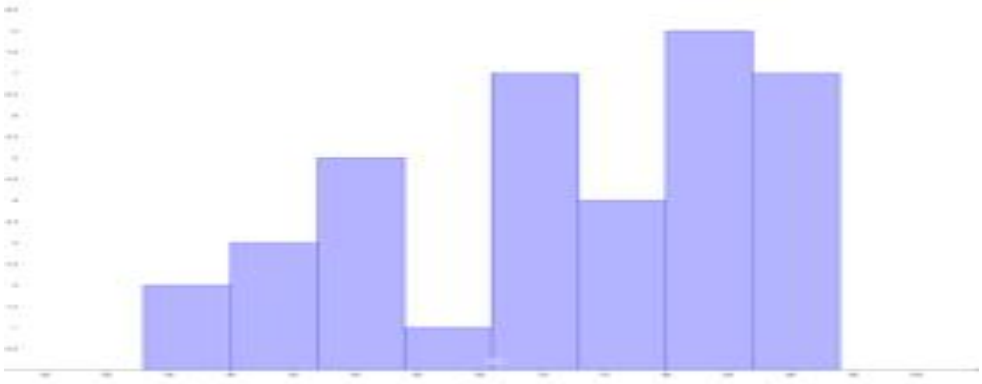


Fig. 3 An example for univariate plots

You can use the following code for this purpose.

```
import pandas
import matplotlib.pyplot as plt
data = 'iris_df.csv'
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
dataset = pandas.read_csv(data, names=names)
dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
plt.show()
```

You can create a histogram of each input variable to get an idea of the distribution using the commands shown below:

```
dataset.hist()
plt.show()
```

Multivariate Plots

Multivariate plots help us to understand the interactions between the variables. For example, we look at different variables (or factors) and how they might impact certain situations or outcomes.

Consider the marketingscenario, you might look at how the variable_1i.e., "money spent on advertising" impacts the variable_2 i.e., "number of sales". Here, we are considering two variables for the analysis and the same is put up in the visualization just like Fig 4.

Fig. 4 A sample for multivariate plots

3.5 Exporting the data

After the data preprocessing is completed, we need to store the updated / modified data into the hard drive permanently. For example, the data should move from the python Jupyter to Hard Disk. The most common format is a csv file or excel file.The built in functions to_csv() and to_excel() of pandas can be used in order to export data.The syntax can be understood from the following code.

- Exporting data as a csv file

  df.to_csv('C:/Deva/MyDataset1.csv')

- Exporting data as a excel file

  df.to_excel('C:/Deva/MyDataset1.csv ')

Consider the code for creating a dataframe and exporting the contents into permanent file.

```
import pandas as pd
data = {'product': ['computer', 'tablet', 'printer', 'laptop'], 'price': [850, 200, 150, 1300]}
df = pd.DataFrame(data)
df.to_csv('C:\MyFolder\MyFile1.csv', index=False, header=True)
print(df)
```

3.6 Data Wrangling

Data wrangling is one of the most important tasks in Machine Learning and also in data science. The process of gathering, collecting and transforming the original / raw data into another format is called data wrangling. This is made for better understanding, better decision-making, better accessing and better analysis in less time. There are few concepts that can help for effective data wrangling.

Data exploration: In this process, the data is studied, analyzed and understood by visualizing representations of data.

```
# Import pandas package
import pandas as pd
# Assign data
data = {'Name': ['Jai', 'Princi', 'Gaurav','Anuj', 'Ravi', 'Natasha', 'Riya'],'Age': [17, 17, 18, 17, 18, 17, 17],'Gender': ['M', 'F', 'M', 'M', 'M', 'F', 'F'],'Marks': [90, 76, 'NaN', 74, 65, 'NaN', 71]}
# Convert into DataFrame
```

```
df = pd.DataFrame(data)
# Display data
df
```

Dealing with missing values:

```
# Compute average
c = avg = 0
for ele in df['Marks']:
  if str(ele).isnumeric():
    c += 1
    avg += ele
avg /= c
# Replace missing values
df = df.replace(to_replace="NaN",
          value=avg)
```

Reshaping data:

```
# Categorize gender
df['Gender'] = df['Gender'].map({'M': 0,'F': 1, }).astype(float)
```

Filtering data:

```
# Filter top scoring students
df = df[df['Marks'] >= 75]

# Remove age row
df = df.drop(['Age'], axis=1)
```

Merge operation is used to merge raw data and into the desired format as follows. Here the field is the name of the column, which is similar on both data-frame.

```
pd.merge( data_frame1,data_frame2, on="field ")
```

WRANGLING DATA USING MERGE OPERATION

```
# Import module
import pandas as pd
# Creating Dataframe
details = pd.DataFrame({ 'ID': [101, 102, 103, 104, 105,106, 107, 108, 109, 110],
'NAME': ['Jagroop', 'Praveen', 'Harjot','Pooja', 'Rahul', 'Nikita', 'Saurabh', 'Ayush', 'Dolly', "Mohit"],'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE', 'CSE','CSE', 'CSE', 'CSE', 'CSE', 'CSE']})
 # Creating Dataframe
fees_status = pd.DataFrame( {'ID': [101, 102, 103, 104, 105,106, 107, 108, 109, 110],'PENDING': ['5000', '250', 'NIL','9000', '15000', 'NIL','4500', '1800', '250', 'NIL']})
 # Merging Dataframe
print(pd.merge(details, fees_status, on='ID'))
```

DETAILS STUDENTS DATA WHO WANT TO PARTICIPATE IN THE EVENT:

```
# Import module
import pandas as pd
# Initializing Data
```

```
student_data = {'Name': ['Amit', 'Praveen', 'Jagroop','Rahul', 'Vishal', 'Suraj','Rishab',
'Satyapal', 'Amit', 'Rahul', 'Praveen', 'Amit'],'Roll_no': [23, 54, 29, 36, 59, 38,12, 45, 34, 36, 54,
23],'Email': ['xxxx@gmail.com', 'xxxxxx@gmail.com','xxxxxx@gmail.com',
'xx@gmail.com','xxxx@gmail.com', 'xxxxx@gmail.com','xxxxx@gmail.com',
'xxxxx@gmail.com','xxxxx@gmail.com', 'xxxxxx@gmail.com','xxxxxxxxxx@gmail.com',
'xxxxxxxxxx@gmail.com']}

 # Creating Dataframe of Data

df = pd.DataFrame(student_data)

 # Printing Dataframe

print(df)
```

DATA WRANGLED BY REMOVING DUPLICATE ENTRIES:

```
# import module

import pandas as pd

# initializing Data

student_data = {'Name': ['Amit', 'Praveen', 'Jagroop', 'Rahul', 'Vishal', 'Suraj', 'Rishab',
'Satyapal', 'Amit','Rahul', 'Praveen', 'Amit'],'Roll_no': [23, 54, 29, 36, 59, 38,12, 45, 34, 36, 54,
23],'Email': ['xxxx@gmail.com', 'xxxxxx@gmail.com', 'xxxxxx@gmail.com', 'xx@gmail.com',
'xxxx@gmail.com', 'xxxxx@gmail.com','xxxxx@gmail.com',
'xxxxx@gmail.com','xxxxx@gmail.com', 'xxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com',
'xxxxxxxxxx@gmail.com']}

# creating dataframe

df = pd.DataFrame(student_data)

# Here df.duplicated() function displays duplicate entries in Rollno column.

non_duplicate = df[~df.duplicated('Roll_no')]

# printing non-duplicate values

print(non_duplicate)
```

3.7 Exploratory Data Analysis (EDA)

Thisconcept isused by data scientists to analyze and investigate the datasets. They will summarize the important characteristics of the dataset using data visualization methods. This analysis will make data scientists very easy and comfortable to discover some new patterns, spot anomalies in the existing patterns, test their hypothesis and etc.,The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. Analysis also helps stakeholders by confirming they are asking the right questions. Analysis can help answer questions about standard deviations, categorical variables, and confidence intervals. Once Analysis is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modeling, including machine learning.There are four primary types of Exploratory Data Analysis, which are given below.

Univariate non-graphical:

The data that come from making a particular measurement on all of the subjects in a sample represent our observations for a single characteristic such as age, gender, speed at a task, or response to a stimulus.

Univariate graphical:

Graphical methods are required to provide a full picture of the data. The few commonly used methods are as follows. Stem-and-leaf plots, which show all data values and the shape of the distribution. Histograms, a barplot in which each bar represents the frequency (count) or

proportion (count/total count) of cases for a range of values.Box plot, which graphically depicts the five-number summary of minimum, first quartile, median, third quartile, and maximum.

Multivariate non-graphical:

Multivariate data arises from more than one variable. Multivariate non-graphical EDA techniques generally show the relationship between two or more variables of the data through cross-tabulation or statistics.

Multivariate graphical:

Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable.Other common types of multivariate graphics include, Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.Multivariate chart, which is a graphical representation of the relationships between factors and a response.Run chart, which is a line graph of data plotted over time.Bubble chart, which is a data visualization that displays multiple circles (bubbles) in a two-dimensional plot.Heat map, which is a graphical representation of data where values are depicted by color.

Summary

- The concepts of Data Analysis areintroduced.
- We understood the fundamentals of dataset and downloading from the website.
- The process of data wrangling is discussed with examples.
- We came to know about different aspects of exploratory data analysis and their types.
- We have seen the necessary python code for preprocessing, data visualization and others.
- Necessary Python code is given to explain the data preprocessing and other relevant concepts.

Keywords

- Data Analysis
- Import and Export
- Data Preprocessing
- Data Wrangling
- Exploratory Data Analysis

Self Assessment

1. Data Analysis is a process of?

A. Inspecting data
B. Cleaning data
C. Transforming data
D. All of the above

2. How many main statistical methodologies are used in data analysis?

A. 2
B. 3
C. 4
D. 5

3. Data Analytics uses _____ to get insights from data.

A. Statistical figures
B. Numerical aspects
C. Statistical methods
D. None of the mentioned above

4. Text Analytics, also referred to as Text Mining?
A. True
B. False

5. Correlation is the relationship between _____ variables.
A. One
B. Two
C. Zero
D. All of the mentioned above

6. Which of the following is the correct extension of the Python file?
A. .python
B. .pl
C. .py
D. .p

7. Which keyword is used for function in Python language?
A. Function
B. def
C. Fun
D. Define

8. What is a hypothesis?
A. A statement that the researcher wants to test through the data collected in a study
B. A research question the results will answer
C. A theory that underpins the study
D. A statistical method for calculating the extent to which the results could have happened by chance

9. Customer analytics refers to _____.
A. Customer Relationship Management: churn analysis and prevention
B. Marketing: cross-sell, up-sell
C. Pricing: leakage monitoring, promotional effects tracking, competitive price responses
D. All of the mentioned above

10. Which of the following graph can be used for simple summarization of data?
A. Scatter plot
B. Overlaying
C. Bar plot
D. All of the mentioned

11. Result analysis are relatively easy to replicate or reproduce.

A. True

B. False

12. Which of the following gave rise to need of graphs in data analysis?

A. Data visualization

B. Communicating results

C. Decision making

D. All of the mentioned

13. What will be output of given code?

```
df = pd.DataFrame( { 'c1' : [ 12, 34, 45], 'c2' : [32, 21, 44], 'c3' : [74, 41, 20] } )
print(df.index)
```

A. Index([0,1,2], dtype = 'int')

B. RangeIndex(start=0, stop=3, step=1)

C. 0 1 2

D. None of the above

14. The plot method on Series and Data Frame is just a simple wrapper around _____.

A. gplt.plot()

B. plt.plot()

C. plt.plotgraph()

D. None of the mentioned

15. Which of the following is not true about series and data frames?

A. Both are size mutable.

B. Both can be derived from pandas.

C. Both can be reshaped into different forms.

D. Both can be created by passing data in form of list, dictionaries and ndarray.

Answers for Self Assessment

1.	D	2.	A	3	C	4.	A	5.	B
6.	C	7.	B	8.	A	9.	D	10.	C
11.	B	12.	D	13.	B	14.	B	15.	A

Review Questions

1. Explain the importance of data analysis.
2. Give the different approaches for data cleaning.
3. Give the python code for importing the data from UCI repository.
4. Differenciateunivariate and multivariate analysis with examples.
5. Whydata wrangling is used?Give the various steps involved in this.

6. How to remove the duplicate entries from the dataset?
7. Illustrate the fundamentals of exploratory data analysis.
8. Give the types of exploratory data analysis.

Further Readings

- John Zelle, "Python Programming: An Introduction to Computer Science", Second Edition, Franklin, Beedle and Associates Inc, 2009.
- Applied Machine Learning by MadanGopal, McGraw Hill Education, India, 2018.
- Machine Learning by Tom Mitchell, McGraw Hill Education, India, 2017.
- Principles of Soft Computing by S. N. Sivanandam and S. N. Deepa, Wiley, India, 2018.

Web Links

- https://www.simplilearn.com/data-analysis-methods-process-types-article
- https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/
- https://learnpython.com/blog/how-to-summarize-data-in-python/
- https://www.tutorialspoint.com/python_data_science/python_data_wrangling.htm
- https://www.analyticsvidhya.com/blog/2021/08/exploratory-data-analysis-and-visualization-techniques-in-data-science/

Unit 04 : Implementation ofPre-processing

CONTENTS

Objectives

Introduction

4.1 Importing the Data

4.2 Summarizing the Dataset

4.3 Data Visualization

4.4 Exporting the Data

4.5 Data Wrangling

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further Readings

Objectives

- To implementthe concepts of Data Preprocessing and Data Analysis.
- To implementthe importing and exporting of datasets.
- To understand the python code for preprocessing of data.
- To draw different types of graphs using matplotlib and pandas packages.
- To understand the process of data wrangling with examples.

Introduction

Data preprocessing is a process of preparing the raw data and making it suitable for a machine-learning model. It is the first and crucial step while creating a machine-learning model because the real world data generally contains noises, missing values and may be in an unusable format, which cannot be directly used for machine learning model. Hence, the data preprocessing is required tasks for cleaning the data and making it suitable for a machine-learning model, which also increases the accuracy and efficiency of a machine-learning model.In this unit, we will discuss and understand the fundamentals of data preprocessing and the necessary steps and approaches in doing the preprocessing. Also, we explore the concept of data analysis and we try to understand how the data wrangling and exploratory data analysis helps for effective data preprocessing.

4.1 Importing the Data

For our practice, we can load the data directly from the UCI Machine Learning repository (https://archive.ics.uci.edu/ml/index.php). We have downloaded the Iris dataset from this and stored in our Desktop. The actual path is mentioned to read the file for importing into python as in Fig 1.

```
In [1]: import pandas as pd
        df = pd.read_csv('/Users/jeyaramvr/Desktop/iris.data', header=None)
        print(df)
```

```
       0    1    2    3               4
0    5.1  3.5  1.4  0.2     Iris-setosa
1    4.9  3.0  1.4  0.2     Iris-setosa
2    4.7  3.2  1.3  0.2     Iris-setosa
3    4.6  3.1  1.5  0.2     Iris-setosa
4    5.0  3.6  1.4  0.2     Iris-setosa
..   ...  ...  ...  ...             ...
145  6.7  3.0  5.2  2.3  Iris-virginica
146  6.3  2.5  5.0  1.9  Iris-virginica
147  6.5  3.0  5.2  2.0  Iris-virginica
148  6.2  3.4  5.4  2.3  Iris-virginica
149  5.9  3.0  5.1  1.8  Iris-virginica

[150 rows x 5 columns]
```

Fig 1. Python code for reading dataset

4.2 Summarizing the Dataset

The dataset may be understood from the abovedetails.Basic Information about the dataset is obtained from the following code as in Fig 2.

```
In [2]: print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       150 non-null    float64
 1   1       150 non-null    float64
 2   2       150 non-null    float64
 3   3       150 non-null    float64
 4   4       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

Fig 2. Python code for reading basic information

Dimensions of Dataset can be obtained using the following code as in Fig 3.

```
[3]: print(df.shape)
```

```
(150, 5)
```

Fig 3. Python code for shape of the dataset

Listing all top 10 data, the following code helps as in Fig 4.

```
[4]: print(df.head(10))
```

```
     0    1    2    3            4
0  5.1  3.5  1.4  0.2  Iris-setosa
1  4.9  3.0  1.4  0.2  Iris-setosa
2  4.7  3.2  1.3  0.2  Iris-setosa
3  4.6  3.1  1.5  0.2  Iris-setosa
4  5.0  3.6  1.4  0.2  Iris-setosa
5  5.4  3.9  1.7  0.4  Iris-setosa
6  4.6  3.4  1.4  0.3  Iris-setosa
7  5.0  3.4  1.5  0.2  Iris-setosa
8  4.4  2.9  1.4  0.2  Iris-setosa
9  4.9  3.1  1.5  0.1  Iris-setosa
```

Fig 4. Python code for getting top 10 rows of dataset

Listing all bottom 10 data, the following code helps as in Fig 5.

```
[5]: print(df.tail(10))
```

```
       0    1    2    3               4
140  6.7  3.1  5.6  2.4  Iris-virginica
141  6.9  3.1  5.1  2.3  Iris-virginica
142  5.8  2.7  5.1  1.9  Iris-virginica
143  6.8  3.2  5.9  2.3  Iris-virginica
144  6.7  3.3  5.7  2.5  Iris-virginica
145  6.7  3.0  5.2  2.3  Iris-virginica
146  6.3  2.5  5.0  1.9  Iris-virginica
147  6.5  3.0  5.2  2.0  Iris-virginica
148  6.2  3.4  5.4  2.3  Iris-virginica
149  5.9  3.0  5.1  1.8  Iris-virginica
```

Fig 5. Python code for getting bottom 10 rows of dataset

View the Statistical Summary from this code as in Fig 6.

```
[6]: print(df.describe())
```

```
                0           1           2           3
count  150.000000  150.000000  150.000000  150.000000
mean     5.843333    3.054000    3.758667    1.198667
std      0.828066    0.433594    1.764420    0.763161
min      4.300000    2.000000    1.000000    0.100000
25%      5.100000    2.800000    1.600000    0.300000
50%      5.800000    3.000000    4.350000    1.300000
75%      6.400000    3.300000    5.100000    1.800000
max      7.900000    4.400000    6.900000    2.500000
```

Fig 6. Python code for getting statistical data of a dataset

4.3 Data Visualization

Data visualization is the process of representing data using visual elements like charts, graphs, etc.The data can be better understood if we provide and summarize using the beautiful diagrams, which is known as data visualization.Generally, there are two types of plots exists and used for data visualization. They are univariate and multivariate. We can select first four columns from the iris data set for the visualization using iloc function as in Fig 7.

```
[10]: x = df.iloc[0:150, [0,3]].values
      print(x)
      [4.9 0.1]
      [5.4 0.2]
      [4.8 0.2]
      [4.8 0.1]
      [4.3 0.1]
      [5.8 0.2]
      [5.7 0.4]
      [5.4 0.4]
      [5.1 0.3]
      [5.7 0.3]
      [5.1 0.3]
      [5.4 0.2]
      [5.1 0.4]
      [4.6 0.2]
      [5.1 0.5]
      [4.8 0.2]
      [5.  0.2]
      [5.  0.4]
      [5.2 0.2]
      [5.2 0.2]
```

Fig 7. Getting First four columns from the iris data set

The last column is the target / labels used for classification purposes. The following code will help us if we want to know what it is. We can use it if it is necessary. Fig 8 depicts the python code.

```
[11]: y = df.iloc[0:150,4].values
      print(y)

      ['Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
       'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
```

Fig 8. Getting Last column from the iris data set

Univariate Plots

Let us explore the first type of visualization i.e., univariate plots. Here, 'uni' means one and'variate' indicates a variable. Therefore, univariate plot is a form of diagram / graph that only involves single variable as in Fig 9, 10 and 11.

```
In [17]: A = x[:,0]
         B = x[:,1]
         plt.bar(A, B)
```

```
Out[17]: <BarContainer object of 150 artists>
```

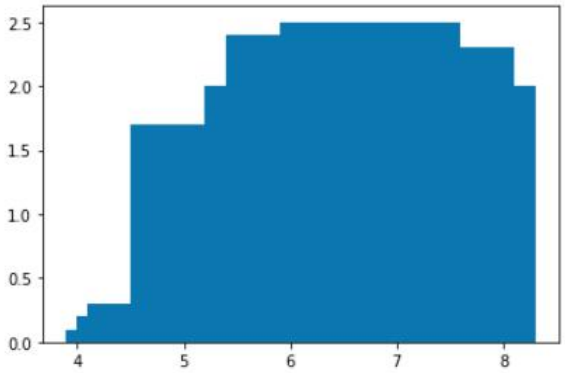


Fig 9 Bar Chart

```
[ ]: #Box and Whisker Plots
```

```
[6]: dataset.hist()
     plt.show()
```

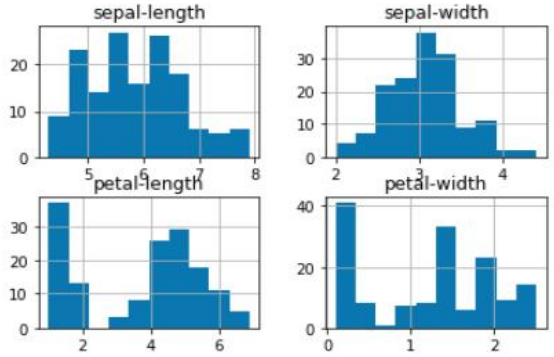


Fig 10 Histogram Graphs

```
[1]: import pandas
     import matplotlib.pyplot as plt
```

```
[2]: data = 'iris.data'
     names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
```

```
[3]: dataset = pandas.read_csv(data, names=names)
```

```
[4]: dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
     plt.show()
```

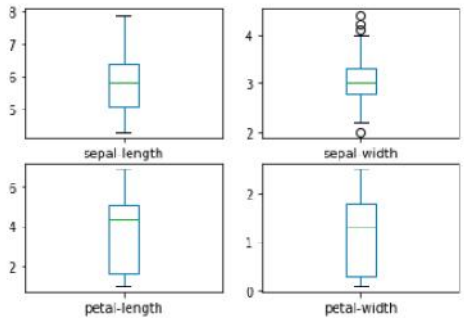


Fig 11 Box Charts

Multivariate Plots

Multivariate plots help us to understand the interactions between the variables. Here, we are considering two variables for the analysis and the same is put up in the visualization using matplotlib package as in Fig 12.

```
[12]: ### matplotlib

[13]: import matplotlib.pyplot as plt

[14]: plt.scatter(x[0:50,0], x[0:50,1],color='red',marker='*',label='iris-setosa')
      plt.scatter(x[50:100,0] ,x[50:100,1],color='green',marker='+',label='iris-versicolor')
      plt.scatter(x[100:150,0] ,x[100:150,1],color='blue',marker='<',label='iris-virginica')
      plt.legend()
      plt.show()
```

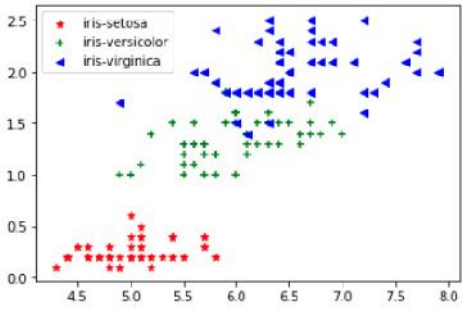


Fig 12 Scatter plots

4.4 Exporting the Data

After the data preprocessing is completed, we need to store the updated / modified data into the hard drive permanently. For example, the data should move from the python Jupyter to Hard Disk. The most common format is a csv file or excel file.The built in functions to_csv() and to_excel() of pandas can be used in order to export data.The syntax can be understood from the following code.

Exporting data as a csv fileas in Fig 13.

```
[14]: df.to_csv('/Users/jeyaramvr/Desktop/irisCSVcopy.csv')

[15]: df2 = pd.read_csv('/Users/jeyaramvr/Desktop/irisCSVcopy.csv', header=None)

[16]: df2.head(5)
```

:[16]:

	0	1	2	3	4	5
0	NaN	0.0	1.0	2.0	3.0	4
1	0.0	5.1	3.5	1.4	0.2	Iris-setosa
2	1.0	4.9	3.0	1.4	0.2	Iris-setosa
3	2.0	4.7	3.2	1.3	0.2	Iris-setosa
4	3.0	4.6	3.1	1.5	0.2	Iris-setosa

Fig 13 Exporting the data as csv file

Exporting data as a excel file as in Fig 14.

```
df.to_excel('/Users/jeyaramvr/Desktop/irisEXCELcopy.xls')
```

```
df3 = pd.read_excel('/Users/jeyaramvr/Desktop/irisEXCELcopy.xls', header=None)
```

```
df3.describe()
```

	0	1	2	3	4
count	150.000000	151.000000	151.000000	151.000000	151.000000
mean	74.500000	5.804636	3.040397	3.747020	1.210596
std	43.445368	0.952494	0.463347	1.764343	0.774610
min	0.000000	0.000000	1.000000	1.000000	0.100000
25%	37.250000	5.100000	2.800000	1.600000	0.300000
50%	74.500000	5.800000	3.000000	4.300000	1.300000
75%	111.750000	6.400000	3.300000	5.100000	1.800000
max	149.000000	7.900000	4.400000	6.900000	3.000000

Fig 14 Exporting data as Excel File.

4.5 Data Wrangling

Data wrangling is one of the most important tasks in Machine Learning and also in data science. The process of gathering, collecting and transforming the original / raw data into another format is called data wrangling. This is made for better understanding, better decision-making, better accessing and better analysis in less time. There are few concepts that can help for effective data wrangling. Let us consider this below for converting dictionary data type into dataframe data type. This helps to explore more on data using available python libraries as in Fig 15.

```
[1]: #Import pandas package
     import pandas as pd

     # Assign data
     data = {'Name': ['Jai', 'Princi', 'Gaurav', 'Anuj', 'Ravi', 'Natasha', 'Riya'],
             'Age': [17, 17, 18, 17, 18, 17, 17],
             'Gender': ['M', 'F', 'M', 'M', 'M', 'F', 'F'],
             'Marks': [90, 76, 'NaN', 74, 65, 'NaN', 71]}
```

```
[2]: # Convert into DataFrame
     df = pd.DataFrame(data)

     # Display data
     print(df)
```

```
      Name  Age Gender Marks
0      Jai   17      M    90
1   Princi   17      F    76
2   Gaurav   18      M   NaN
3     Anuj   17      M    74
4     Ravi   18      M    65
5  Natasha   17      F   NaN
6     Riya   17      F    71
```

Fig 15 Conversion into DataFrame

Dealing with missing values:

```
[3]: # Compute average
     c = avg = 0
     for ele in df['Marks']:
         if str(ele).isnumeric():
             c += 1
             avg += ele
     avg /= c
```

```
[4]: print(avg)
```

```
75.2
```

Fig 16 Calculating Average of Marks

```
[5]: df = df.replace(to_replace="NaN", value=avg)
```

```
[6]: print(df)
```

```
      Name  Age Gender  Marks
0      Jai   17      M   90.0
1   Princi   17      F   76.0
2   Gaurav   18      M   75.2
3     Anuj   17      M   74.0
4     Ravi   18      M   65.0
5  Natasha   17      F   75.2
6     Riya   17      F   71.0
```

Fig 17 Replacing Nan with Average Value

```
[7]: # Categorize gender
df['Gender'] = df['Gender'].map({'M': 0, 'F': 1, }).astype(float)
```

```
[8]: print(df)
```

```
      Name  Age  Gender  Marks
0      Jai   17     0.0   90.0
1   Princi   17     1.0   76.0
2   Gaurav   18     0.0   75.2
3     Anuj   17     0.0   74.0
4     Ravi   18     0.0   65.0
5  Natasha   17     1.0   75.2
6     Riya   17     1.0   71.0
```

Fig 18 Giving 0 or 1 for Gender Column

```
[9]: # Filter top scoring students
df = df[df['Marks'] >= 75]
print(df)
```

```
      Name  Age  Gender  Marks
0      Jai   17     0.0   90.0
1   Princi   17     1.0   76.0
2   Gaurav   18     0.0   75.2
5  Natasha   17     1.0   75.2
```

Fig 19 Filtering the Data

WRANGLING DATA USING MERGE OPERATION

```
[1]: #Import pandas package
import pandas as pd

# Assign data
data = {'Name': ['Jai', 'Princi', 'Gaurav', 'Anuj', 'Ravi', 'Natasha', 'Riya'],
        'Age': [17, 17, 18, 17, 18, 17, 17],
        'Gender': ['M', 'F', 'M', 'M', 'M', 'F', 'F'],
        'Marks': [90, 76, 'NaN', 74, 65, 'NaN', 71]}
```

```
[2]: # Convert into DataFrame
df = pd.DataFrame(data)

# Display data
print(df)
```

```
      Name  Age Gender Marks
0      Jai   17      M    90
1   Princi   17      F    76
2   Gaurav   18      M   NaN
3     Anuj   17      M    74
4     Ravi   18      M    65
5  Natasha   17      F   NaN
6     Riya   17      F    71
```

Fig 20 Sample preparation of dataframe from "data"

```
[4]: # Import module
import pandas as pd

# Creating Dataframe
details = pd.DataFrame({ 'ID': [101, 102, 103, 104, 105, 106, 107, 108, 109, 110],
'NAME': ['Jagroop', 'Praveen', 'Harjot', 'Pooja', 'Rahul', 'Nikita', 'Saurabh',
         'Ayush', 'Dolly', 'Mohit"], 'BRANCH': ['CSE', 'CSE', 'CSE', 'CSE', 'CSE',
        'CSE', 'CSE', 'CSE', 'CSE', 'CSE']})
print(details)
```

```
    ID     NAME BRANCH
0  101  Jagroop    CSE
1  102  Praveen    CSE
2  103   Harjot    CSE
3  104    Pooja    CSE
4  105    Rahul    CSE
5  106   Nikita    CSE
6  107  Saurabh    CSE
7  108    Ayush    CSE
8  109    Dolly    CSE
9  110    Mohit    CSE
```

Fig 20 Sample preparation of dataframe from "details"

```
[5]: # Creating Dataframe
fees_status = pd.DataFrame ( {'ID': [101, 102, 103, 104, 105, 106, 107, 108,
109, 110], 'PENDING': ['5000', '250', 'NIL', '9000', '15000', 'NIL', '4500', '1800',
'250', 'NIL']})
print(fees_status)
```

```
    ID PENDING
0  101    5000
1  102     250
2  103     NIL
3  104    9000
4  105   15000
5  106     NIL
6  107    4500
7  108    1800
8  109     250
9  110     NIL
```

Fig 21 Sample preparation of dataframe from "fees_status"

```
[6]: # Merging Dataframe
print(pd.merge(details, fees_status, on='ID'))
```

```
    ID     NAME BRANCH PENDING
0  101  Jagroop    CSE    5000
1  102  Praveen    CSE     250
2  103   Harjot    CSE     NIL
3  104    Pooja    CSE    9000
4  105    Rahul    CSE   15000
5  106   Nikita    CSE     NIL
6  107  Saurabh    CSE    4500
7  108    Ayush    CSE    1800
8  109    Dolly    CSE     250
9  110    Mohit    CSE     NIL
```

Fig 22 Merging of dataframe from "details" and "fees_details"

DETAILS STUDENTS DATA

```
[7]: # Import module
import pandas as pd
# Initializing Data
student_data = {'Name': ['Amit', 'Praveen', 'Jagroop', 'Rahul', 'Vishal', 'Suraj',
'Rishab', 'Satyapal', 'Amit', 'Rahul', 'Praveen', 'Amit'],
'Roll_no': [23, 54, 29, 36, 59, 38, 12, 45, 34, 36, 54, 23],
'Email': ['xxxx@gmail.com', 'xxxxxx@gmail.com', 'xxxxxx@gmail.com', 'xx@gmail.com',
'xxxx@gmail.com', 'xxxxx@gmail.com', 'xxxxx@gmail.com', 'xxxxx@gmail.com',
'xxxxx@gmail.com', 'xxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com', 'xxxxxxxxxx@gmail.com']}

# Creating Dataframe of Data
df = pd.DataFrame(student_data)

# Printing Dataframe
print(df)
```

```
        Name  Roll_no              Email
0       Amit       23       xxxx@gmail.com
1    Praveen       54     xxxxxx@gmail.com
2    Jagroop       29     xxxxxx@gmail.com
3      Rahul       36        xx@gmail.com
4     Vishal       59       xxxx@gmail.com
5      Suraj       38      xxxxx@gmail.com
6     Rishab       12      xxxxx@gmail.com
7   Satyapal       45      xxxxx@gmail.com
8       Amit       34      xxxxx@gmail.com
9      Rahul       36     xxxxxx@gmail.com
10   Praveen       54  xxxxxxxxxx@gmail.com
11      Amit       23  xxxxxxxxxx@gmail.com
```

Fig 23 Student Data before duplicate checking

DATA WRANGLED BY REMOVING DUPLICATE ENTRIES:

```
[8]: non_duplicate = df[~df.duplicated('Roll_no')]
print(non_duplicate)
```

```
        Name  Roll_no              Email
0       Amit       23       xxxx@gmail.com
1    Praveen       54     xxxxxx@gmail.com
2    Jagroop       29     xxxxxx@gmail.com
3      Rahul       36        xx@gmail.com
4     Vishal       59       xxxx@gmail.com
5      Suraj       38      xxxxx@gmail.com
6     Rishab       12      xxxxx@gmail.com
7   Satyapal       45      xxxxx@gmail.com
8       Amit       34      xxxxx@gmail.com
```

Fig 24 Student Data without duplicates

Summary

- Implemented the concepts of Data Preprocessing and Data Analysis.
- Implemented the importing and exporting of datasets.
- Understood the python code for preprocessing of data.
- We understood how to draw different types of graphs using matplotlib and pandas packages.
- Understood the process of data wrangling with examples.

Keywords

- Import and Export
- Data Preprocessing
- Pandas
- Matplotlib
- Data Wrangling

Self Assessment

Q1) Data Analysis is a process of?

A. Inspecting data

B. Cleaning data

C. Transforming data

D. All of the above

Q2) How many main statistical methodologies are used in data analysis?

A. 2

B. 3

C. 4

D. 5

Q3) Data Analytics uses _____ to get insights from data.

A. Statistical figures

B. Numerical aspects

C. Statistical methods

D. None of the mentioned above

Q4) Text Analytics, also referred to as Text Mining?

A. True

B. False

Q5) Correlation is the relationship between _____ variables.

A. One

B. Two

C. Zero

D. All of the mentioned above

Q6) Which of the following is the correct extension of the Python file?

A .python

B .pl

C .py

D .p

Q7) Which keyword is used for function in Python language?

A Function

B def

C Fun

D Define

Q8) What is a hypothesis?

A. A statement that the researcher wants to test through the data collected in a study

B. A research question the results will answer

C. A theory that underpins the study

D. A statistical method for calculating the extent to which the results could have happened by chance

Q9) Customer analytics refers to _____.

A. Customer Relationship Management: churn analysis and prevention

B. Marketing: cross-sell, up-sell

C. Pricing: leakage monitoring, promotional effects tracking, competitive price responses

D. All of the mentioned above

Q10) Which of the following graph can be used for simple summarization of data?

A Scatter plot

B Overlaying

C Bar plot

D All of the mentioned

Q11) Result analysis are relatively easy to replicate or reproduce.

A True

B False

Q12) Which of the following gave rise to need of graphs in data analysis?

A Data visualization

B Communicating results

C Decision making

D All of the mentioned

Q13)What will be output of given code?

```
df = pd.DataFrame( { 'c1' : [ 12, 34, 45], 'c2' : [32, 21, 44], 'c3' : [74, 41, 20] } )
print(df.index)
```

A Index([0,1,2], dtype = 'int')

B RangeIndex(start=0, stop=3, step=1)

C 0 1 2

D None of the above

Q14)The plot method on Series and DataFrame is just a simple wrapper around _____.

A gplt.plot()

B plt.plot()

C plt.plotgraph()

D None of the mentioned

Q15)Which of the following is not true about series and dataframes?

A Both are size mutable.

B Both can be derived from pandas.

C Both can be reshaped into different forms.

D Both can be created by passing data in form of list, dictionaries and ndarray.

Answers for Self Assessment

1.	D	2.	A	3	C	4.	A	5.	B
6.	C	7.	B	8.	A	9.	D	10.	C
11.	B	12.	D	13.	B	14.	B	15.	A

Review Questions

1. Explain the importance of data analysis.
2. Give the different approaches for data cleaning.
3. Give the python code for importing the data from UCI repository.
4. Differentiateunivariate and multivariate analysis with examples.
5. Whyis data wrangling used?Give the various steps involved in this.
6. How to remove the duplicate entries from the dataset?
7. Illustrate the fundamentals of exploratory data analysis.
8. Give the types of exploratory data analysis.

Further Readings

John Zelle, "Python Programming: An Introduction to Computer Science", Second Edition, Franklin, Beedle and Associates Inc, 2009.

Applied Machine Learning by MadanGopal, McGraw Hill Education, India, 2018.

Machine Learning by Tom Mitchell, McGraw Hill Education, India, 2017.

Principles of Soft Computing by S. N. Sivanandam and S. N. Deepa, Wiley, India, 2018.

Web Links

- https://www.simplilearn.com/data-analysis-methods-process-types-article
- https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/
- https://learnpython.com/blog/how-to-summarize-data-in-python/
- https://www.tutorialspoint.com/python_data_science/python_data_wrangling.htm
- https://www.analyticsvidhya.com/blog/2021/08/exploratory-data-analysis-and-visualization-techniques-in-data-science/

Unit 05: Physical Layer

CONTENTS

Objectives
Introduction
5.1 What is the Purpose of a Regression Model?
5.2 Types of Regression Analysis
5.3 Multiple Linear Regression
5.4 Assumptions for Multiple Linear Regression
Summary
Keywords
Self Assessment
Answers for Self Assessment
Review Questions
Further Readings

Objectives

- learn what is regression analysis.
- understand the purpose of regression analysis.
- learn different types of regression analysis

Introduction

Regression analysis is the method that is most frequently used to address regression issues in machine learning. It is based on data modelling and comprises choosing the line that fits the data the best and travels the least distance between each data point while passing through all of the data points. Although there are other regression analysis methods, logistic and linear regression are the most frequently employed. In the end, the nature of the data will dictate the kind of regression analysis model we use.

5.1 What is the Purpose of a Regression Model?

When knowledge of the independent variables is available, regression analysis is used to either predict the value of the dependent variable or to determine how an independent variable will affect the dependent variable.

5.2 Types of Regression Analysis

There are several regression analysis prediction methods accessible. The number of independent variables, the shape of the regression line, and the kind of dependent variable are other factors that influence the approach choice.

Figure 1 Types of regression Analysis

1. Linear Regression

Linear regression, which presumes a linear relationship between a dependent variable (Y) and an independent variable (X), is the modelling technique that is most frequently utilised. It uses a best-fit line, commonly referred to as a regression line. Y = c+m*X + e, where 'c' stands for the intercept,'m' for the line's slope, and 'e' for the error term, is the formula for the linear relationship.

One dependent variable and more than one independent variable can be used in a complex linear regression model, which can be simple (just one dependent variable and one independent variable).

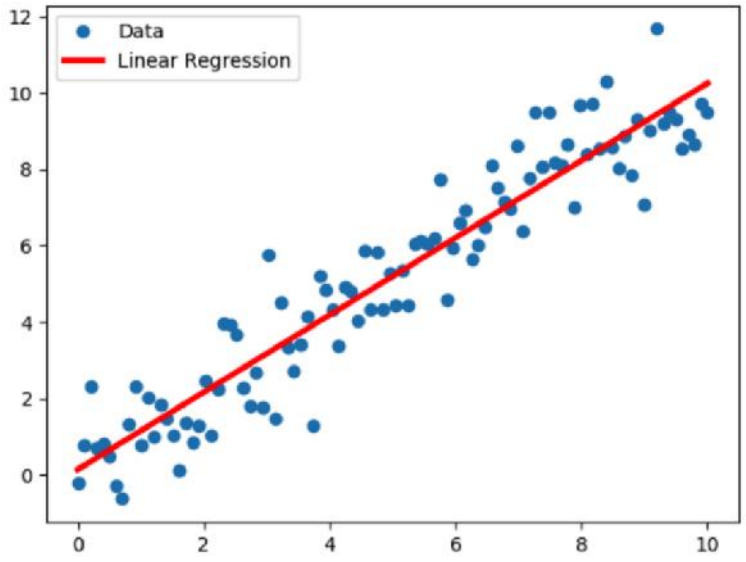


Figure 2 Linear Regression

2. Logistic Regression

The logistic regression method is appropriate when the dependent variable is discrete. In other words, this method is used to determine the likelihood of events that are mutually exclusive, such as pass/fail, true/false, 0/1, and so on. Thus, the probability has a value

between 0 and 1, the target variable has a range of two possible values, and its relationship to the independent variable is depicted by a sigmoid curve.

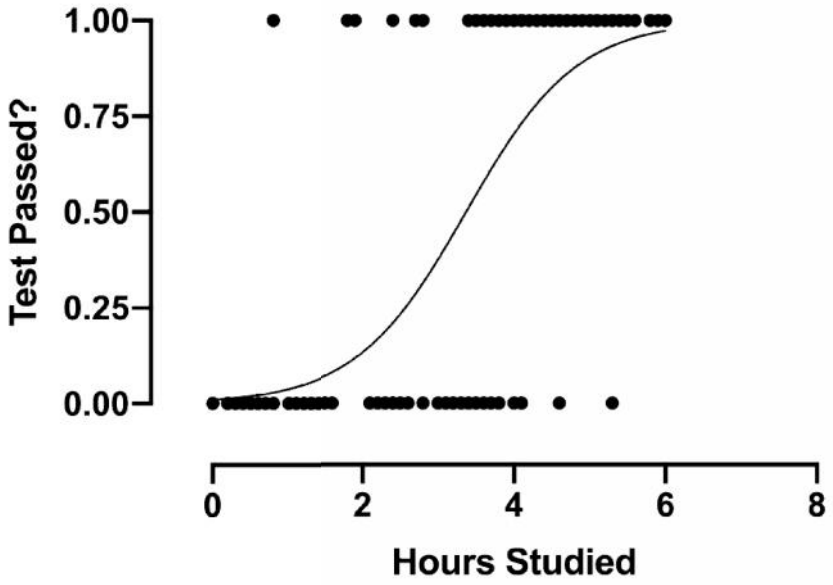


Figure 3 Logistic Regression

3. *Polynomial regression*

In order to depict a non-linear relationship between dependent and independent variables, polynomial regression analysis is performed. The best fit line is curved instead of straight in this variation of the multiple linear regression model.

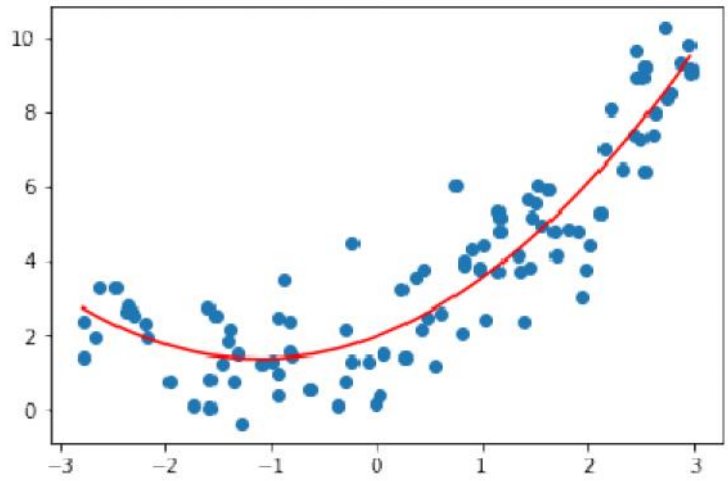


Figure 4 Polynomial Regression

4. *Ridge Regression*

The ridge regression technique is used when the independent variables are highly correlated and the data shows multicollinearity. Even if least squares estimates are impartial in multicollinearity, their variances are high enough to induce a difference between the observed value and the true value. By inflating the regression estimates, ridge regression lowers standard errors.

The multicollinearity issue in the ridge regression equation is solved by the lambda () variable.

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

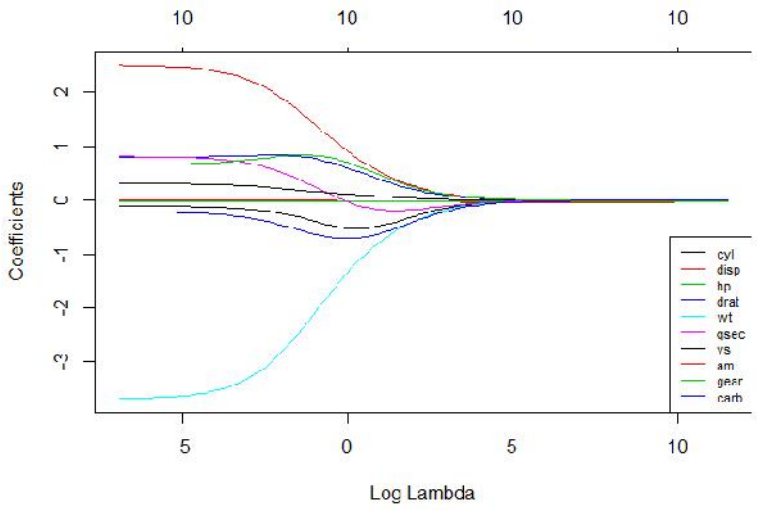
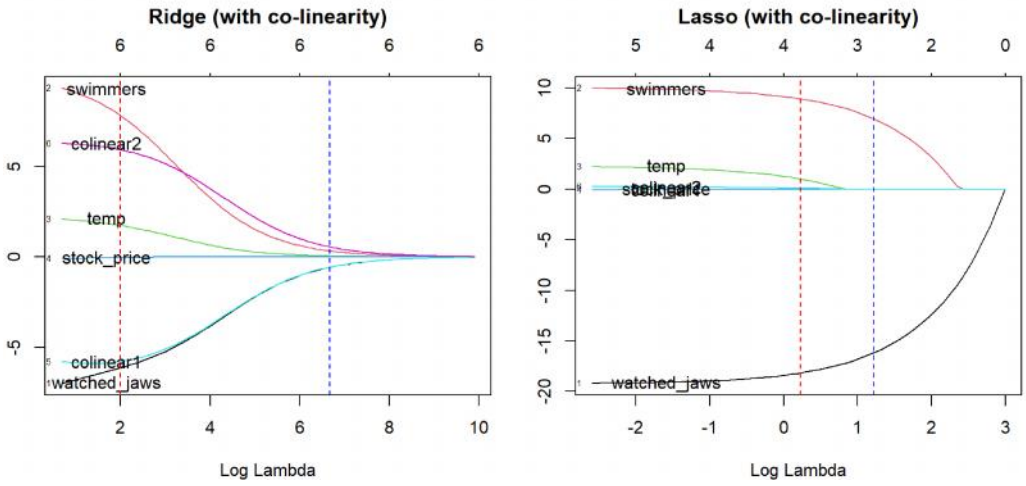


Figure 5 Ridge Regression

5. *Lasso Regression*

The lasso (Least Absolute Shrinkage and Selection Operator) method penalises the absolute magnitude of the regression coefficient, just like ridge regression does. The lasso regression method also uses variable selection, which causes the coefficient values to zero off completely.



6. *Quantile Regression*

A part of the linear regression method is the quantile regression methodology. When the conditions for linear regression are not met or when there are outliers in the data, it is used. Quantile regression is used in statistics and econometrics.

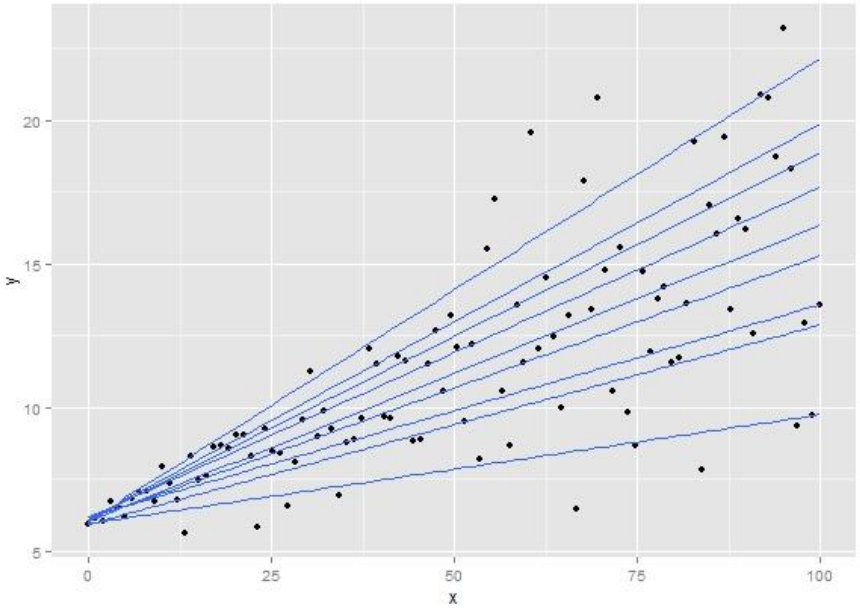


Figure 6 Quantile Regression

7. Bayesian Linear Regression

The Bayes theorem is utilised in Bayesian linear regression, a type of regression analysis method used in machine learning to determine the values of the regression coefficients. This method calculates the posterior distribution of the features rather than the least-squares. As a result, the method performs better in terms of stability than standard linear regression.

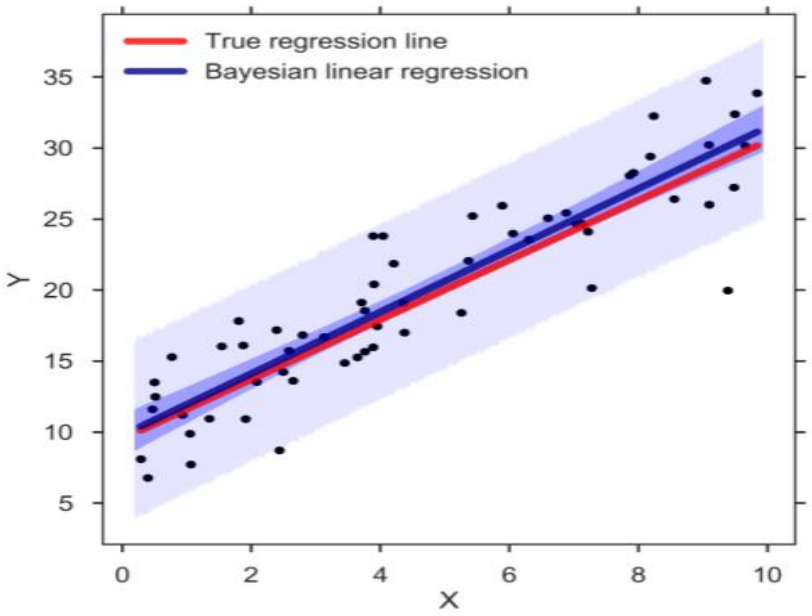


Figure 7 Bayesian Linear Regression

8. Principal Component Regression

The principlecomponents regression approach is frequently used to evaluate multicollinear regression data. By biassing the regression estimates, the significant components regression approach, like ridge regression, lowers standard errors. The training data are first modified using principal component analysis (PCA), and the changed samples are then utilised to train the regressors.

9. Partial Least Squares Regression

A quick and effective method for covariance-based regression analysis is partial least squares regression. It is beneficial for regression issues where there is a high likelihood of multicollinearity between the variables. Regression is used once the procedure reduces the number of variables to a reasonable number of predictors.

10. Elastic Net Regression

When working with highly correlated data, elastic net regression combines the ridge and lasso regression techniques. By utilizing the penalties connected to the ridge and lasso regression techniques, it regularizes regression models.

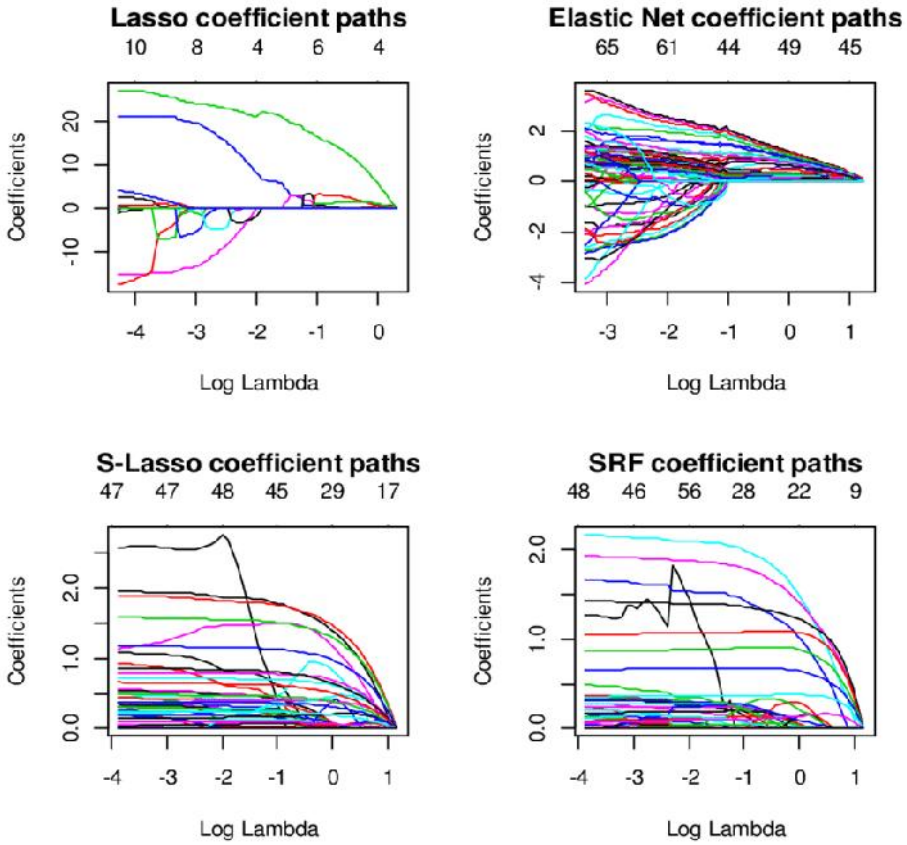


Figure 8 Elastic Net Regression

5.3 Multiple Linear Regression

In simple linear regression, which models the response variable (Y) using just one Independent/Predictor(X) variable. However, there may be a number of situations when more than one predictor variable has an impact on the response variable; in these situations, the Multiple Linear Regression technique is applied.

As more than one predictor variable is required to predict the response variable, Multiple Linear Regression is a development of Simple Linear Regression. It can be described as:

One of the key regression techniques, multiple linear regression simulates the linear relationship between a single continuous dependent variable and a number of independent variables.

Example

Prediction of CO_2 emission based on engine size and number of cylinders in a car.

Some key points about MLR:

The independent or predictor variable may be continuous or categorical for MLR, but the dependent or target variable (Y) must be continuous/real.

Each feature variable needs to simulate the dependent variable's linear relationship.

A regression line is attempted to be fitted using MLR through a multidimensional space of data points.

MLR Equation

The target variable (Y) in multiple linear regression is a linear mixture of several predictor variables (x1, x2, x3,..., xn). The equation for multiple linear regression is as follows since it is an improvement over simple linear regression:

Y= b₀+b₁x₁+ b₂x₂+ b₃x₃+..... bnxn ..... (a)

Where,

Y= Output/Response variable

b_0, b_1, b_2, b_3 , b_n....= Coefficients of the model.

x_1, x_2, x_3, x_4,...= Various Independent/feature variable

5.4 Assumptions for Multiple Linear Regression

There should be a linear link between the predictor and target variables.

The residuals from the regression must be evenly distributed.

MLR makes very few, if any, assumptions for multicollinearity (correlation between the independent variables).

Implementation of Multiple Linear Regression model using Python

To implement MLR using Python, we have below problem:

A dataset of 50 startup enterprises is available. R&D spending, office expenses, marketing expenses, state information, and profit for a fiscal year are the five primary pieces of information in this dataset. Our objective is to develop a model that can quickly identify which firm has the highest profit and which element has the most impact on a company's profit.

Profit is the dependent variable since we need to determine it; the other four variables are independent variables. The main steps for implementing the MLR model are listed below:

1. Data Pre-processing Steps
2. Fitting the MLR model to the training set
3. Predicting the result of the test set

Step 1: Data Pre-processing

Data pre-processing, which we just covered in this course, is the first step. The steps in this technique are as follows:

Bringing in libraries In order to build the model, we will first import the library. The code is as follows:

- ***# importing libraries***

1. import numpy as nm
2. **import** matplotlib.pyplot as mtp
3. **import** pandas as pd

Bringing in a dataset Thedataset (50_CompList), which contains all the variables, will now be imported. The code is as follows:

- ***#importing datasets***

1. data_set= pd.read_csv('50_CompList.csv')

Output: We will get the dataset as:

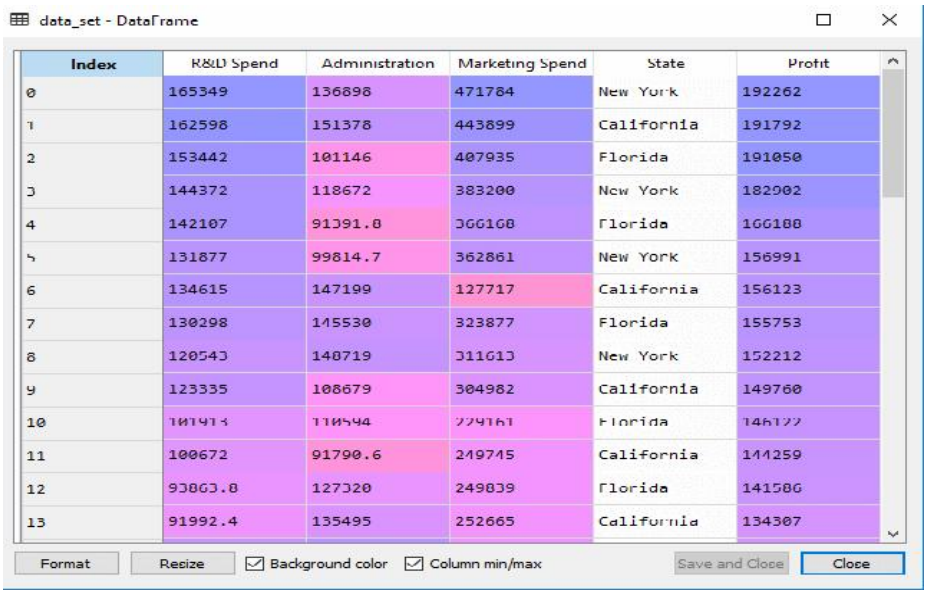


Figure 9 Dataset-dataframe

We can see from the output above that there are five variables, four of which are continuous and one of which is a categorical variable.

- Extracting dependent and independent Variables:

x= data_set.iloc[:, :-1].values

y= data_set.iloc[:, 4].values

Output

Encoding Dummy Variables

We shall encrypt our single categorical variable (State), as it cannot be used in the model directly. We'll use the LabelEncoder class to convert the categorical variable to numbers. However, it falls short because there is still some relational order, which could lead to an incorrect model. Therefore, we will utilise OneHotEncoder to produce the dummy variables in order to solve this issue. The code is as follows:

#Catgorical data

from sklearn.preprocessing **import** LabelEncoder, OneHotEncoder

labelencoder_x= LabelEncoder()

x[:, 3]= labelencoder_x.fit_transform(x[:,3])

onehotencoder= OneHotEncoder(categorical_features= [3])

x= onehotencoder.fit_transform(x).toarray()

Since the other variables are continuous, we are simply encoding the independent variable state in this situation.

Output

Figure 10 Output

The state column has been changed into dummy variables (0 and 1), as seen in the result shown above. Each dummy variable column in this case corresponds to a single State. By contrasting it with the original dataset, we can make sure. The first column represents the state of California, the second column represents the state of Florida, and the third column represents the state of New York.

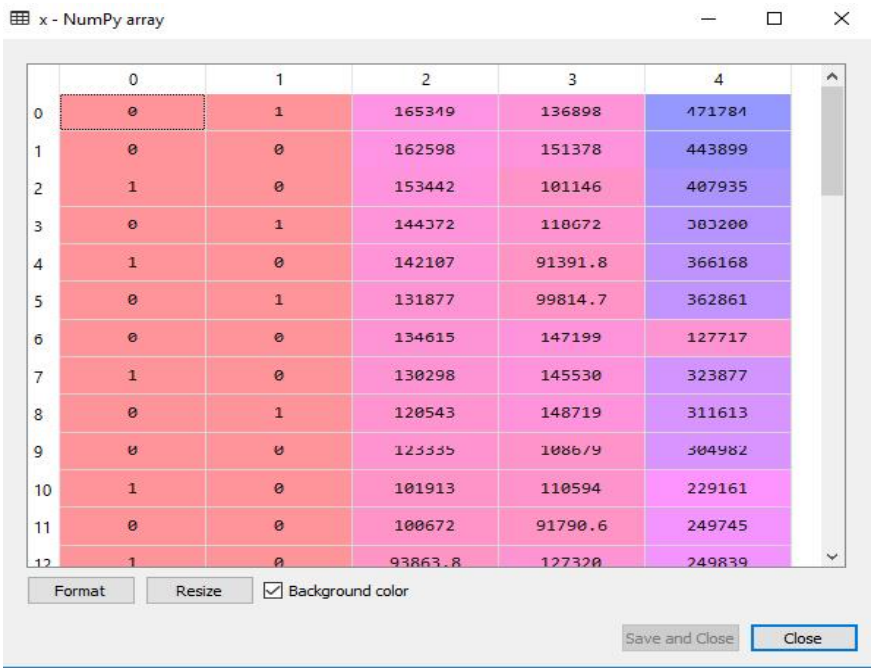
Noted: It must be one less than the total number of dummy variables because we shouldn't use all of them at once or else a dummy variable trap would be created.

The only piece of code we are writing right now is to avoid the dummy variable trap:

#avoiding the dummy variable trap:

x = x[:, 1:]

The model may become multicollinear if the initial dummy variable is not eliminated.



As shown in the output image up top, the first column has been eliminated.

The dataset will now be divided into a training set and a test set. Below is the code for this:

```
# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
```

Our dataset will be divided into a training set and a test set by the code above.

Output:The dataset will be divided into a training set and a test set by the aforementioned code. By selecting the variable explorer option in Spyder IDE, you may view the output. The training set and test set will resemble the illustration below:

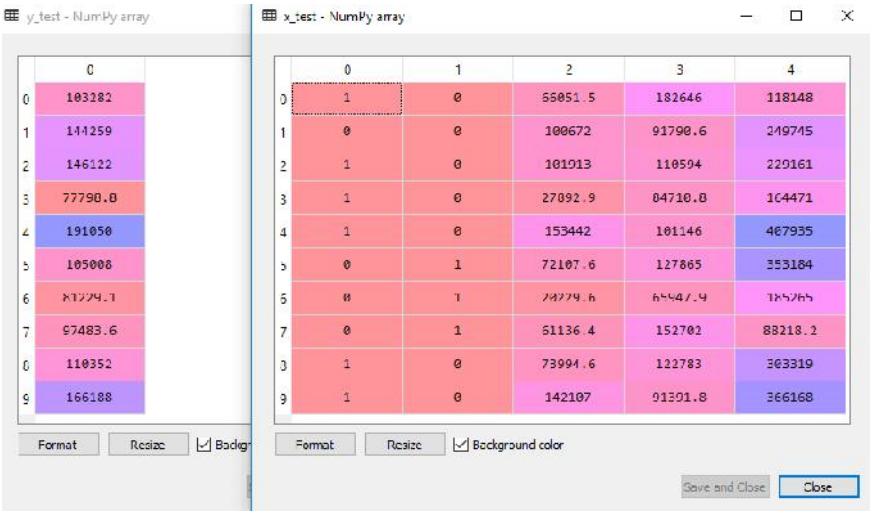


Figure 11 Test set

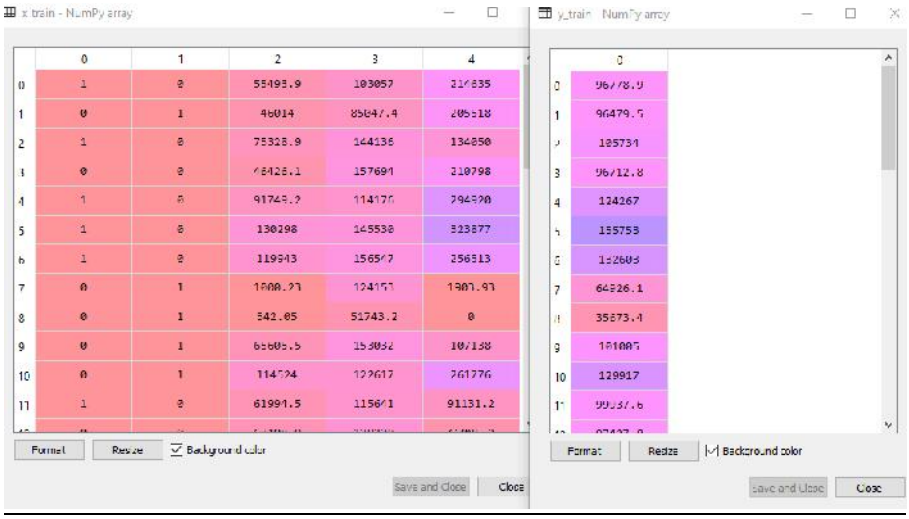


Figure 12 Training set

We tomatically, saving us the time.

Step: 2- Fitting our MLR model to the Training set:

Now that our dataset has been properly prepared for training, we will fit our regression model to the training set. The process will resemble what we accomplished with the Simple Linear Regression model. This will be coded as follows:

```
#Fitting the MLR model to the training set:
from sklearn.linear_model import LinearRegression
```

```
regressor= LinearRegression()
regressor.fit(x_train, y_train)
```

Output:

```
Out[9]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

We have now successfully used the training dataset to train our model. The performance of the model will be evaluated using the test dataset in the following phase.

Step: 3- Prediction of Test set results

Checking the model's performance is the final stage for our model. We'll do it by projecting the outcome of the test set. We will generate a y_pred vector for prediction. The code is as follows:

```
1. #Predicting the Test set result;
y_pred= regressor.predict(x_test)
```

The lines of code above when run will create a new vector under the variable explorer option. By contrasting the projected values with test set values, we may evaluate our model.

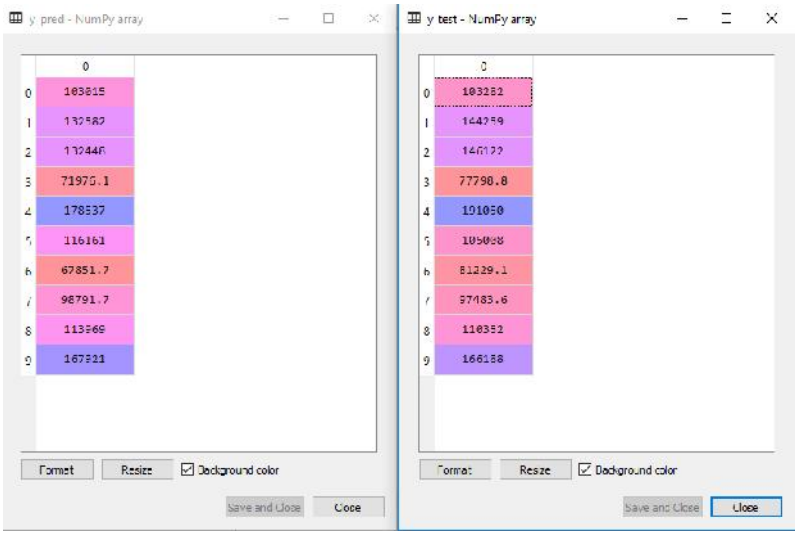


Figure 13 Output

We have the test set and the anticipated result set in the output shown above. By contrasting these two values index by index, we may evaluate the performance of the model. As an illustration, the first index has a predicted profit value of 103015 dollars and a test/real profit value of 103282 dollars. Since the difference is only 267 dollars, as predicted, our model is fully complete.

Additionally, we can look both the test and training dataset scores. The code is as follows:

```
print('Train Score: ', regressor.score(x_train, y_train))
print('Test Score: ', regressor.score(x_test, y_test))
```

Output: The score is:

Train Score: 0.9501847627493607

Test Score: 0.9347068473282446

According to the aforementioned score, our model is 93% accurate on the test dataset and 95% correct on the training dataset.

Notably, we'll examine how applying the Backward Elimination procedure can help the model perform better in the topic that follows.

- ***Applications of Multiple Linear Regression:***

There are mainly two applications of Multiple Linear Regression:

- Effectiveness of Independent variable on prediction.

- Predicting the impact of changes:

Summary

- Regression is a statistical analysis technique used to model the relationship between a dependent variable and one or more independent variables
- This type of regression models the relationship between a single independent variable and a continuous dependent variable.
- It involves modeling the relationship between multiple independent variables and a continuous dependent variable.
- It extends linear regression by introducing polynomial terms to capture nonlinear relationships between variables.
- Unlike linear regression, logistic regression is used when the dependent variable is categorical or binary. It models the probability of an event occurring.
- It is a regularization technique that adds a penalty term to linear regression to mitigate overfitting and handle multicollinearity.
- Similar to ridge regression, lasso regression also introduces a penalty term but uses L1 regularization. It can perform variable selection by shrinking some coefficients to zero.
- Elastic net regression combines both L1 and L2 regularization (ridge and lasso) to address multicollinearity and perform feature selection.
- This type of regression is used when the data is collected over time and involves modeling the relationship between variables with a temporal component.
- Nonlinear regression models the relationship between variables using nonlinear functions. It is useful when the data does not fit a linear model well.
- It applies Bayesian statistical techniques to regression analysis, incorporating prior knowledge and updating beliefs about the relationship between variables.
- GLMs extend linear regression to handle different types of dependent variables, including binary, count, and categorical data. Examples include Poisson regression and logistic regression.
- Robust regression techniques are designed to handle outliers and influential observations that can significantly impact traditional regression models.

Keywords

- ***Regression analysis:*** It is a statistical technique used to model the relationship between a dependent variable and one or more independent variables.
- ***Linear regression:*** It is a type of regression analysis where the relationship between the dependent variable and independent variable(s) is assumed to be linear. It aims to find the best-fit line that minimizes the differences between the observed data points and the predicted values.
- ***Multiple regression:*** It extends linear regression by considering multiple independent variables to model the relationship with a dependent variable. It helps analyze how multiple factors collectively influence the dependent variable.
- ***Polynomial regression***: It expands linear regression by introducing polynomial terms (e.g., quadratic, cubic) to capture nonlinear relationships between variables.
- ***Logistic regression***: Unlike linear regression, logistic regression is used when the dependent variable is categorical or binary. It models the probability of an event occurring based on the independent variables.

- ***Ridge regression***: It is a regularization technique that adds a penalty term (L2 regularization) to linear regression to prevent overfitting and handle multicollinearity (high correlation between independent variables).
- ***Lasso regression***: Similar to ridge regression, lasso regression adds a penalty term (L1 regularization) to linear regression. It can perform variable selection by shrinking some coefficients to zero, effectively excluding them from the model.
- ***Elastic net regression***: Elastic net regression combines both L1 and L2 regularization (ridge and lasso) to address multicollinearity and perform feature selection.
- ***Time series regression***: It is used when the data is collected over time, allowing for the modeling of relationships between variables with a temporal component.
- ***Nonlinear regression***: Nonlinear regression models the relationship between variables using nonlinear functions instead of assuming a linear relationship.
- ***Bayesian regression***: Bayesian regression applies Bayesian statistical techniques to regression analysis, incorporating prior knowledge and updating beliefs about the relationship between variables.
- ***Generalized linear models (GLM):*** GLMs extend linear regression to handle different types of dependent variables, including binary, count, and categorical data. Examples include Poisson regression and logistic regression.
- ***Robust regression***: Robust regression techniques are designed to handle outliers and influential observations that can significantly impact traditional regression models.
- ***Dependent variable***: Also known as the response variable, it is the variable being predicted or explained by the independent variables in the regression model.
- ***Independent variable***: Also known as the predictor variable, it is the variable(s) used to explain or predict the value of the dependent variable.
- ***Coefficient:*** In regression analysis, coefficients represent the weights or slopes assigned to the independent variables, indicating the strength and direction of their relationship with the dependent variable.
- ***Prediction***: Regression models can be used to make predictions about the value of the dependent variable based on the values of the independent variables.
- ***Residuals***: Residuals are the differences between the observed values of the dependent variable and the predicted values by the regression model. They provide information about the model's accuracy.
- ***Overfitting:*** Overfitting occurs when a regression model fits the training data too closely, capturing noise and random variations rather than the true underlying relationship. It may result in poor performance when applied to new, unseen data.
- ***Underfitting***: Underfitting happens when a regression model is too simple and fails to capture the underlying patterns and relationships in the data.
- ***Multicollinearity***: Multicollinearity refers to high correlation between independent variables in a regression model. It can lead to instability in the coefficient estimates and makes it challenging to interpret the individual effects of the variables.
- ***Model selection***: Model selection involves choosing the most appropriate regression model among several candidate models based on various criteria such as goodness of fit, simplicity, and interpretability.
- ***Goodness of fit***: Goodness of fit measures how well a regression model fits the observed data. Common metrics include R-squared, which quantifies the proportion of variance explained by the model.
- ***R-squared***: R-squared (coefficient of determination) is a statistical measure that represents the proportion of variance in the dependent variable that can be explained by the independent variables in the regression model. It ranges from 0 to 1, with higher values indicating a better fit.

- ***Adjusted R-squared***: Adjusted R-squared is a modified version of R-squared that takes into account the number of independent variables and adjusts for the degrees of freedom.
- ***Cross-validation***: Cross-validation is a technique used to assess the performance and generalization ability of a regression model by splitting the data into training and testing subsets.
- ***Outliers:*** Outliers are data points that deviate significantly from the overall pattern in the data. They can have a strong influence on the regression model and affect its results.
- ***Homoscedasticity***: Homoscedasticity refers to the assumption in regression analysis that the variability of the residuals is constant across different levels of the independent variables.
- ***Heteroscedasticity***: Heteroscedasticity occurs when the variability of the residuals is not constant across different levels of the independent variables. It can violate the assumptions of linear regression.
- ***Assumptions of regression***: Regression analysis relies on certain assumptions, including linearity, independence of errors, constant variance, absence of multicollinearity, and normal distribution of residuals. These assumptions should be checked and met for reliable regression results.

Self Assessment

1. Which type of regression analysis is suitable for predicting a continuous dependent variable based on one or more independent variables?

A. Logistic regression

B. Polynomial regression

C. Multiple linear regression

D. Ridge

2. Which technique is used to address multicollinearity in regression analysis?

A. Principal Component Analysis (PCA)

B. Ridge regression

C. Lasso regression

D. Logistic

3. What does R-squared measure in regression analysis?

A. The proportion of variance in the dependent variable explained by the independent variables

B. The correlation coefficient between the dependent and independent variables

C. The significance of the regression coefficients

D. The residuals of the regression model

4. Which type of regression analysis is used when the dependent variable is categorical or binary?

A. Polynomial regression

B. Logistic regression

C. Ridge regression

D. Simple linear regression

5. What is the purpose of regularization techniques, such as ridge regression and lasso regression?

A. To handle outliers in the data

B. To reduce overfitting and improve model generalization

C. To select the most important independent variables

D. To address heteroscedasticity in the residuals

6. What is the main difference between homoscedasticity and heteroscedasticity?

A. Homoscedasticity refers to equal variances of residuals, while heteroscedasticity refers to unequal variances.

B. Homoscedasticity refers to the absence of multicollinearity, while heteroscedasticity refers to its presence.

C. Homoscedasticity refers to normally distributed residuals, while heteroscedasticity refers to non-normal residuals.

D. Homoscedasticity refers to linear relationships between variables, while heteroscedasticity refers to nonlinear relationships.

7. Which assumption in linear regression states that the residuals should follow a normal distribution?

A. Linearity assumption

B. Independence assumption

C. Homoscedasticity assumption

D. Normality assumption

8. What is the primary purpose of feature selection in regression analysis?

A. To improve model interpretability

B. To reduce overfitting and simplify the model

C. To handle multicollinearity

D. To address heteroscedasticity in the residuals

9. Which technique is used to evaluate the performance of a regression model on unseen data?

A. Cross-validation

B. Residual analysis

C. Outlier detection

D. Goodness-of-fit test

10. Linear Regression is used for:

A. Classification problems

B. Clustering problems

C. Regression problems

D. Dimensionality reduction

11. The objective of Linear Regression is to find the best-fit line that minimizes:

A. The mean squared error

B. The mean absolute error

C. The sum of squared residuals

D. The sum of absolute residuals

12. The equation of a simple linear regression line is given by:

A. y = mx + b

B. y = ax^2 + bx + c

C. y = e^(mx + b)

D. y = log(mx + b)

13. Logistic Regression is used for:

A. Regression problems

B. Classification problems

C. Clustering problems

D. Dimensionality reduction

14. The output of Logistic Regression is:

A. A continuous value

B. A probability score

C. A binary class label

D. A multi-class label

15. Which evaluation metric is commonly used to assess the performance of Linear Regression models?

A. Accuracy

B. F1-score

C. R-squared (R^2)

D. Area Under the Curve (AUC)

Answers for Self Assessment

1.	C	2.	B	3.	A	4.	B	5.	B
6.	A	7.	D	8.	B	9.	A	10.	C
11.	C	12.	A	13.	B	14.	C	15.	C

Review Questions

1. What is regression analysis, and what is its primary purpose?
2. Explain the difference between simple linear regression and multiple linear regression.
3. How does polynomial regression differ from linear regression? When is it useful?

4. What is logistic regression, and what types of problems is it suitable for?
5. What are the purposes of regularization techniques such as ridge regression and lasso regression?
6. Describe the concept of overfitting in regression analysis. How can it be addressed?
7. What is the difference between homoscedasticity and heteroscedasticity in the context of regression analysis?
8. How does time series regression differ from cross-sectional regression?
9. Explain the concept of multicollinearity in regression analysis and its potential impact on the model.
10. What are the key assumptions of linear regression, and why are they important to consider

Further Readings

- https://www.javatpoint.com/regression-analysis-in-machine-learning
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- https://www.utstat.toronto.edu/~brunner/books/LinearModelsInStatistics.pdf
- https://www.analyticsvidhya.com/blog/2022/01/different-types-of-regression-models/

Unit 06: Introduction to Numpy

CONTENTS

Objectives

Introduction

6.1 Implementation and Performance Analysis of Linear Regression

6.2 Multiple Regression

6.3 How does it function?

6.4 Non-Linear Regression

6.5 How does a Non-Linear Regression work?

6.6 What are the Applications of Non-Linear Regression?

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further Readings

Objectives

- learn basic concepts about arrays and lists.
- learn to differentiate between array and list.
- learn several array creation routines in Numpy which are used to create Ndarray objects.

Introduction

When attempting to determine the relationship between two variables, the term regression is utilised.That link is employed in statistical modelling and machine learning to forecast how future events will turn out.

Every data scientist should start by mastering linear regression, one of the original machine learning methods. This straightforward model aids in our understanding of fundamental machine learning ideas, such as identifying dependent and independent variables, developing models, and comprehending the mathematics and statistics underlying models.

Using the statsmodel and sklearn libraries are the two most popular methods for creating linear regression in Python. Both are excellent choices with advantages and disadvantages.

6.1 Implementation and Performance Analysis of Linear Regression

Data Preparation:Start by collecting and preparing your dataset. Ensure that you have a suitable dataset with a continuous dependent variable and one or more independent variables.

Data Split: Split the dataset into training and testing subsets. Typically, the training set is used to train the model, and the testing set is used to evaluate its performance.

Model Training: Fit the linear regression model to the training data. Most programming languages and libraries provide built-in functions or classes for linear regression. Train the model by providing the independent variables and the corresponding dependent variable.

Model Evaluation: Evaluate the trained model's performance using various evaluation metrics. Common metrics for linear regression include R-squared, mean squared error (MSE), and mean absolute error (MAE). Calculate these metrics by comparing the predicted values to the actual values from the testing dataset.

Interpretation of Coefficients: Analyse the coefficients of the linear regression model to understand the relationship between the independent variables and the dependent variable. Positive coefficients indicate a positive correlation, while negative coefficients indicate a negative correlation. The magnitude of the coefficients represents the strength of the relationship.

Residual Analysis: Examine the residuals (the differences between the predicted and actual values) to assess the model's goodness of fit. Plotting the residuals against the predicted values can help identify patterns, such as heteroscedasticity or outliers.

Performance Visualization: Visualize the performance of the linear regression model using appropriate graphs and plots. For example, you can create scatter plots to visualize the relationship between the independent and dependent variables, or plot the predicted values against the actual values.

Further Analysis: If necessary, you can explore additional aspects such as feature selection, regularization techniques (e.g., ridge regression, lasso regression), or cross-validation to enhance the model's performance or address specific requirements.

Performance Comparison: To assess the linear regression model's performance, you can compare it with other regression techniques or variations of linear regression (e.g., polynomial regression). This comparison can help determine the effectiveness and suitability of the linear regression approach for your specific dataset.

Remember to interpret the results with caution and consider the assumptions of linear regression, such as linearity, independence of errors, constant variance, and normal distribution of residuals. Additionally, always cross-validate your findings and consider potential limitations or sources of bias in the data.

Overall, implementing and analysing the performance of linear regression involves data preparation, model training, evaluation, interpretation, and visualization. It is crucial to follow a systematic approach and use appropriate evaluation metrics and visualization techniques to draw meaningful conclusions from the analysis.

6.2 Multiple Regression

Similar to linear regression, multiple regression attempts to predict a value based on two or more factors, but with more than one independent value.

Look at the data set below; it includes some car-related information.

Table 1 data.csv

Car	Model	Volume	Weight	CO2
Toyota	Aygo	1000	790	99
Mitsubishi	Space Star	1200	1160	95
Skoda	Citigo	1000	929	95
Fiat	500	900	865	90
Mini	Cooper	1500	1140	105
VW	Up!	1000	929	105
Skoda	Fabia	1400	1109	90

Mercedes	A-Class	1500	1365	92
Ford	Fiesta	1500	1112	98
Audi	A1	1600	1150	99
Hyundai	I20	1100	980	99
Suzuki	Swift	1300	990	101
Ford	Fiesta	1000	1112	99
Honda	Civic	1600	1252	94
Hundai	I30	1600	1326	97
Opel	Astra	1600	1330	97
BMW	1	1600	1365	99
Mazda	3	2200	1280	104
Skoda	Rapid	1600	1119	104
Ford	Focus	2000	1328	105
Ford	Mondeo	1600	1584	94
Opel	Insignia	2000	1428	99
Mercedes	C-Class	2100	1365	99
Skoda	Octavia	1600	1415	99
Volvo	S60	2000	1415	99
Mercedes	CLA	1500	1465	102
Audi	A4	2000	1490	104
Audi	A6	2000	1725	114
Volvo	V70	1600	1523	109
BMW	5	2000	1705	114
Mercedes	E-Class	2100	1605	115
Volvo	XC70	2000	1746	117
Ford	B-Max	1600	1235	104
BMW	2	1600	1390	108

Opel	Zafira	1600	1405	109
Mercedes	SLK	2500	1395	120

The size of the engine can be used to estimate a car's CO2 emissions, but multiple regression allows us to include additional variables, such as the car's weight, to improve the prediction's accuracy.

6.3 How does it function?

We have modules in Python that will carry out the work for us. Import the Pandas module first.

import pandas

We can read CSV files using the Pandas module and get a DataFrame object in response.

The name of file is data.csv. Data is defined above in the table.

Input: df = pandas.read_csv("data.csv").

Create a list of the independent values, and then designate this list as variable X.

Add the dependent values to the y variable.

```
X = df[['Weight', 'Volume']]
y = df['CO2']
```

X is typically used to denote the list of independent values, and Y is typically used to denote the list of dependent values. We will import the sklearn module as well because we will use some of its methods:

from sklearn import linear_model

We'll generate a linear regression object from the sklearn module using the LinearRegression() method.

Fit(), a method on this object, fills the regression object with information about the relationship between the independent and dependent variables as parameters:

regr = linear_model.LinearRegression()
regr.fit(X, y)

Now we have a regression object that are ready to predict CO2 values based on a car's weight and volume:

#predict the CO2 emission of a car where the weight is 2300kg, and the volume is 1300cm3:

predictedCO2 = regr.predict([[2300, 1300]])

```
import pandas
from sklearn import linear_model

df = pandas.read_csv("data.csv")

X = df[['Weight', 'Volume']]
y = df['CO2']

regr = linear_model.LinearRegression()
regr.fit(X, y)

#predict the CO2 emission of a car where the weight is 2300kg, and the volume is 1300cm3:
predictedCO2 = regr.predict([[2300, 1300]])

print(predictedCO2)
```

Result

[107.2087328]

According to our calculations, a car with a 1.3-liter engine and a weight of 2300 kg emits about 107 grammes of carbon dioxide for every kilometre driven.

Coefficient

A quantity used to describe a relationship with an unknowable variable is called a coefficient.

For instance, if x is a variable, then 2x is x multiplied by 2. The coefficient is 2, and the unknown variable is x.

We can request the weight and volume coefficient values against CO2 in this situation. The response(s) we receive explain what would happen if one of the independent values was raised or lowered.

Print the coefficient values of the regression object

```
import pandas
from sklearn import linear_model
df = pandas.read_csv("data.csv")
X = df[['Weight', 'Volume']]
y = df['CO2']
regr = linear_model.LinearRegression()
regr.fit(X, y)
print(regr.coef_)
```

Result

```
[0.00755095 0.00780526]
```

Result Explained

The weight and volume coefficient values are shown in the result array.

Volume: 0.00780526 Pounds Weight: 0.00755095

These numbers indicate that a weight gain of 1 kg will result in an increase in CO2 emissions of 0.00755095g.

Additionally, the CO2 emission rises by 0.00780526 g for every 1 cm3 increase in engine volume.

That's a good assumption, but let's put it to the test!

We have already estimated that the CO2 emission will be around 107g for a car with a 1300cm3 engine weighing 2300kg.

What happens if we add 1000 kg to the weight?

Copy the example from before, but change the weight from 2300 to 3300

```
import pandas
from sklearn import linear_model

df = pandas.read_csv("data.csv")

X = df[['Weight', 'Volume']]
y = df['CO2']
regr = linear_model.LinearRegression()
regr.fit(X, y)
predictedCO2 = regr.predict([[3300, 1300]])
print(predictedCO2)
```

Result:

```
[114.75968007]
```

According to our calculations, a 3300 kg automobile with a 1.3 litre engine will emit about 115 grammes of carbon dioxide for every kilometre it travels.

Which demonstrates that the 0.00755095 coefficient is accurate:

107.2087328 + (1000 * 0.00755095) = 114.75968

6.4 Non-Linear Regression

Polynomial regression includes non-linear regression. A non-linear relationship between the dependent and independent variables is modelled using this technique. It is employed when the data has a curved trend and linear regression, when compared to non-linear regression, would not yield particularly precise answers. This is so because it is expected that the data in linear regression is linear.

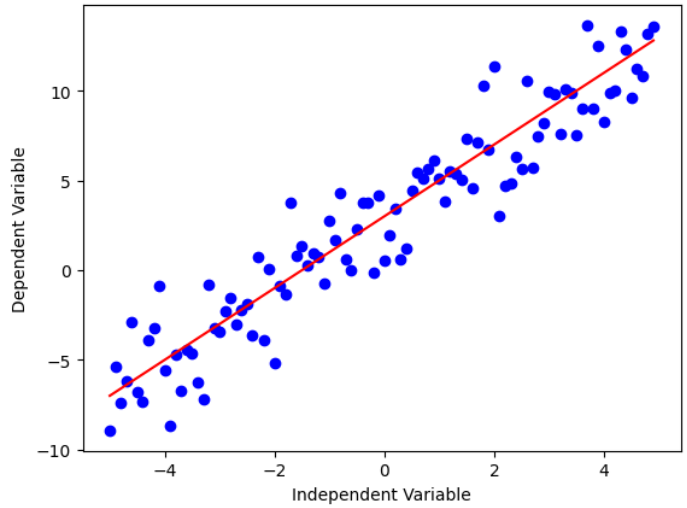
According to our needs, there are a wide variety of regressions available that may be used to match any dataset, including quadratic, cubic, and so on, to an infinite degree.

6.5 How does a Non-Linear Regression work?

If we pay close attention, we will notice that the transition from linear regression to non-linear regression has already occurred. Simply adding the higher-order terms of the dependent features to the feature space is what we are intended to do. Sometimes, but not exactly, this is referred to as feature engineering.

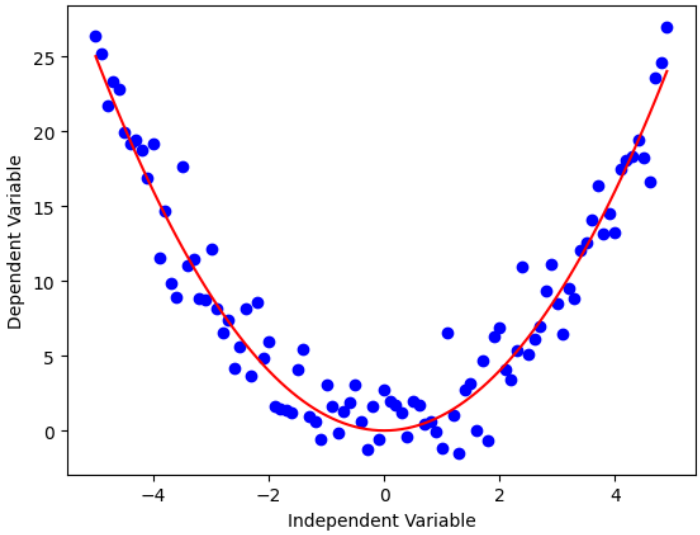
We can fit a curvilinear model to the available data by including non-linear components. Even though non-linear regression and linear regression are comparable, there are several types of difficulties that a machine learning expert must overcome while training a model for such a task. Therefore, a number of well-known techniques, like Levenberg-Marquardt and Gauss-Newton, are utilised to create nonlinear models.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
 x = np.arange(-5.0, 5.0, 0.1)
 # You can adjust the slope and intercept
# to verify the changes in the graph
y = 2*(x) + 3
y_noise = 2 * np.random.normal(size=x.size)
ydata = y + y_noise
# plt.figure(figsize =(8, 6))
plt.plot(x, ydata,  'bo')
plt.plot(x, y, 'r')
plt.ylabel('Dependent Variable')
plt.xlabel('Independent Variable')
plt.show()
```



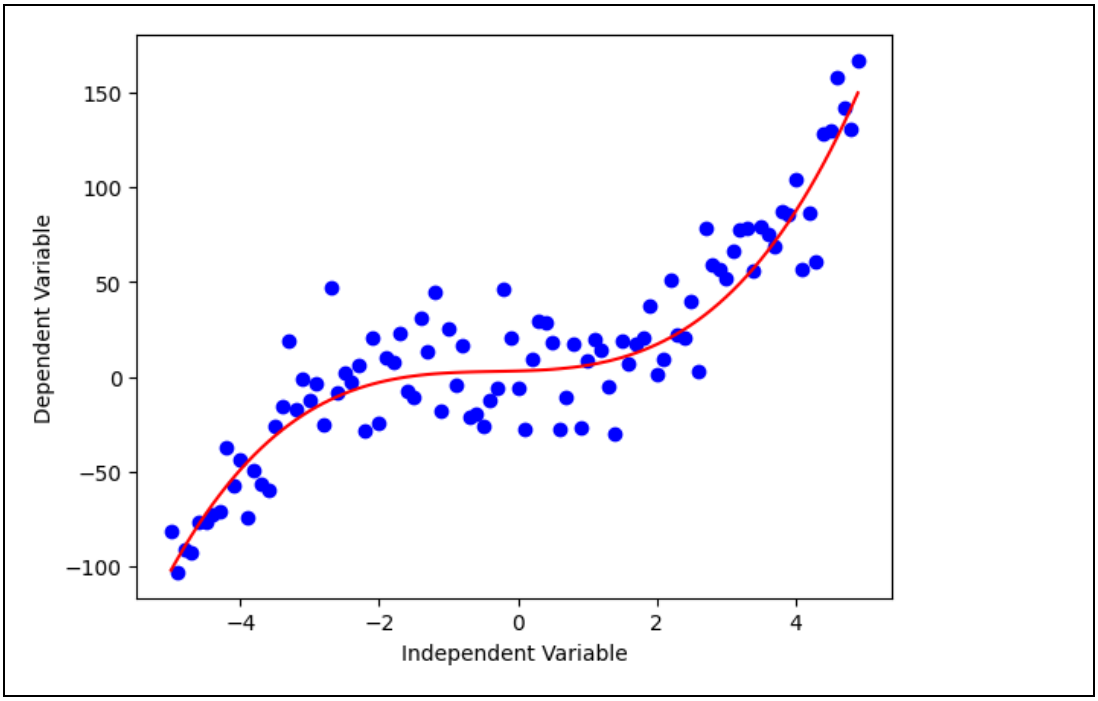
We can now evaluate the non-linearity of the datasets and regressions after seeing an example of linear regression. Create some data for quadratic regression, for example.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
 x = np.arange(-5.0, 5.0, 0.1)
 # You can adjust the slope and intercept
# to verify the changes in the graph
y = np.power(x, 2)
y_noise = 2 * np.random.normal(size=x.size)
ydata = y + y_noise
plt.plot(x, ydata,  'bo')
plt.plot(x, y, 'r')
plt.ylabel('Dependent Variable')
plt.xlabel('Independent Variable')
plt.show()
```



Let's now attempt to give our polynomial one more degree. This will give us a better idea of how non-linear data actually seem in the real world.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
 x = np.arange(-5.0, 5.0, 0.1)
 # You can adjust the slope and intercept
# to verify the changes in the graph
y = 1*(x**3) + 1*(x**2) + 1 * x + 3
y_noise = 20 * np.random.normal(size=x.size)
ydata = y + y_noise
plt.plot(x, ydata,  'bo')
plt.plot(x, y, 'r')
plt.ylabel('Dependent Variable')
plt.xlabel('Independent Variable')
plt.show()
```



A model must have a nonlinear function of the parameters Theta, not necessarily the features X, in order for it to be deemed nonlinear. The non-linear equation can take many different forms, including exponential, logarithmic, logistic, and many others.

$$\hat{y} = \theta_0 + \theta_2{}^2 x$$

$$\hat{y} = \theta_0 + \theta_1 \theta_2{}^x$$

$$\hat{y} = \log(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3)$$

$$\hat{y} = \frac{\theta_0}{1 + \theta_1{}^{(x-\theta_2)}}$$

Figure 1 Non-Linear Regression Equations

As you can see, the change in Y depends on changes in the parameter Theta rather than necessarily on X alone in any of these equations. In other words, a model in non-linear regression has non-linear parameters.

6.6 What are the Applications of Non-Linear Regression?

Since most real-world data is non-linear, non-linear regression approaches outperform linear regression techniques. Using non-linear regression techniques, it is possible to create a solid model with predictions that are both accurate and consistent with the historical pattern of the data. The non-linear regression techniques were successful in completing tasks linked to population growth or decay that exhibit exponential growth or decay, financial forecasting, and logistic pricing models.

Summary

- The regression chapter provides an overview of regression analysis, a statistical technique used to model the relationship between a dependent variable and one or more independent variables. The chapter covers various types of regression models and their applications in different scenarios.

- The chapter begins with an introduction to regression analysis, emphasizing its purpose in understanding and predicting relationships between variables. It explains the key concepts of dependent and independent variables and introduces the idea of fitting a regression model to data.
- Different types of regression models are discussed in the chapter. Simple linear regression is presented as the basic form, where a single independent variable is used to predict a continuous dependent variable. Multiple linear regression extends this concept by incorporating multiple independent variables.
- The chapter also explores polynomial regression, which allows for nonlinear relationships by introducing polynomial terms. Logistic regression is introduced as a technique for modeling categorical or binary dependent variables.
- Regularization techniques, such as ridge regression and lasso regression, are covered to address issues like multicollinearity and overfitting. The concept of feature selection and its importance in regression analysis are explained.
- Assumptions of linear regression are discussed, including linearity, independence of errors, constant variance, and normal distribution of residuals. Violations of these assumptions can affect the accuracy and reliability of regression models.
- The chapter emphasizes the importance of model evaluation and interpretation. Evaluation metrics, such as R-squared, mean squared error (MSE), and mean absolute error (MAE), are introduced to assess the model's performance. Residual analysis and visualizations are discussed as tools for understanding the model's fit to the data.
- Throughout the chapter, practical implementation aspects, such as data preparation, training the model, and interpreting the coefficients, are highlighted. Considerations for addressing outliers, heteroscedasticity, and multicollinearity are also covered.
- The chapter concludes by emphasizing the need for careful interpretation, cross-validation, and consideration of potential limitations and biases in the data. It highlights the importance of comparing regression models and exploring additional techniques to enhance performance and meet specific requirements.
- Overall, the regression chapter provides a comprehensive overview of regression analysis, from the basic concepts to advanced techniques, highlighting their applications and considerations for implementation and interpretation.

Keywords

- ***Regression analysis***: A statistical technique used to model the relationship between a dependent variable and one or more independent variables. It aims to understand and predict the behavior of the dependent variable based on the independent variables.
- ***Linear regression***: A type of regression analysis where the relationship between the dependent variable and independent variable(s) is assumed to be linear. It finds the best-fit line that minimizes the differences between the observed data points and the predicted values.
- ***Multiple regression:*** A regression analysis technique that involves modeling the relationship between a dependent variable and multiple independent variables. It helps analyze how multiple factors collectively influence the dependent variable.
- ***Polynomial regression***: A regression model that extends linear regression by introducing polynomial terms (e.g., quadratic, cubic) to capture nonlinear relationships between variables. It can better fit data that doesn't follow a linear pattern.
- ***Logistic regression:*** A type of regression used when the dependent variable is categorical or binary. It models the probability of an event occurring based on the independent variables. It is commonly used for classification problems.
- ***Ridge regression***: A regularization technique that adds a penalty term (L2 regularization) to linear regression. It helps mitigate overfitting and handles multicollinearity, which occurs when there is high correlation between independent variables.

- ***Lasso regression:*** A regularization technique that adds a penalty term (L1 regularization) to linear regression. It performs variable selection by shrinking some coefficients to zero, effectively excluding them from the model. It can help with feature selection.
- ***Elastic net regression:*** A regression technique that combines both L1 and L2 regularization (ridge and lasso) to handle multicollinearity and perform feature selection. It offers a balance between the two regularization approaches.
- ***R-squared***: A statistical measure that represents the proportion of variance in the dependent variable that can be explained by the independent variables in a regression model. It ranges from 0 to 1, with higher values indicating a better fit.
- ***Cross-validation:*** A technique used to evaluate the performance and generalization ability of a regression model by splitting the data into training and testing subsets. It helps assess how well the model performs on unseen data.
- ***Homoscedasticity:*** An assumption in regression analysis that refers to equal variances of the residuals (differences between predicted and actual values) across different levels of the independent variables. It implies a consistent spread of errors.
- ***Assumptions of regression:*** These are the underlying assumptions in regression analysis that need to be met for reliable results. They include linearity, independence of errors, constant variance of residuals, absence of multicollinearity, and normal distribution of residuals.

Self Assessment

1. What type of relationship does linear regression model between the dependent variable (Y) and the independent variable (X)?

A. Linear

B. Quadratic

C. Exponential

D. Logarithmic

2. In a simple linear regression equation Y = 2X + 3, what is the slope of the line?

A. 2

B. 3

C. 5

D. -2

3. Which method is commonly used to find the best-fitted line in linear regression?

A. Gradient Descent

B. K-Means

C. Hierarchical Clustering

D. Mean Shift

4. In non-linear regression, the relationship between the dependent variable (Y) and the independent variable(s) (X) can be modeled as:

A. Straight line

B. Curve

C. Hyperplane

D. Exponential function

5. Which of the following is a common approach to finding the best-fitted curve in non-linear regression?

A. Ordinary Least Squares (OLS)

B. Gradient Descent

C. Logistic Regression

D. Akaike Information Criterion (AIC)

6. In multiple regression, what is the dependent variable?

A. X

B. Y

C. Z

D. Both a and b

7. What is the key difference between linear regression and multiple regression?

A. Linear regression has one dependent variable, and multiple regressions have multiple dependent variables.

B. Linear regression can handle non-linear relationships, while multiple regression cannot.

C. Multiple regression has multiple independent variables, while linear regression has only one.

D. There is no difference; they are the same.

8. When would you choose non-linear regression over linear regression?

A. When there is a linear relationship between variables.

B. When you have multiple dependent variables.

C. When the data exhibits a non-linear pattern.

D. When the data is not numerical.

9. What is the primary goal of regression analysis?

A. Classification of data points into categories

B. Finding correlations between variables

C. Predicting a continuous dependent variable

D. Summarizing data in a tabular form

10. Which of the following types of regression is suitable for predicting a binary outcome?

A. Linear regression

B. Multiple regression

C. Logistic regression

D. Polynomial regression

11. In simple linear regression, the relationship between the dependent variable (Y) and the independent variable (X) is modeled as:

A. $Y = a + bX$

B. Y = aX^2 + bX + c

C. Y = aX + b

D. Y = a^X + b

12. What is the main objective of the Ordinary Least Squares (OLS) method in regression?

A. Minimize the sum of absolute errors

B. Minimize the sum of squared errors

C. Maximize the correlation coefficient

D. Maximize the R-squared value

13. When evaluating a regression model, what does the R-squared value represent?

A. The accuracy of the model's predictions

B. The percentage of variance in the dependent variable explained by the independent variable

C. The p-value of the regression coefficients

D. The number of data points in the dataset

14. Which technique can be used to handle multicollinearity in multiple linear regression?

A. Drop one of the independent variables causing multicollinearity

B. Combine the correlated independent variables into one variable

C. Perform feature scaling on the independent variables

D. Use regularization techniques like Ridge or Lasso regression

15. What is the key assumption of linear regression?

A. The relationship between variables is nonlinear.

B. The residuals are normally distributed.

C. The dependent variable is categorical.

D. The number of observations is greater than the number of independent variables.

Answers for Self Assessment

1.	A	2.	A	3.	A	4.	B	5.	B
6.	B	7.	C	8.	C	9.	C	10.	C
11.	C	12.	B	13.	B	14.	D	15.	B

Review Questions

1. Discuss the significance of evaluating the performance of a linear regression model. What are some commonly used evaluation metrics for assessing its performance?
2. Explain the concept of multicollinearity in the context of multiple regression. How does multicollinearity affect the interpretation of the regression coefficients?

3. Compare and contrast the performance evaluation process for linear regression and multiple regression models. What additional factors need to be considered in multi-regression analysis?
4. What are the main limitations of linear regression when dealing with non-linear relationships between variables? How can non-linear regression models address these limitations?
5. Describe the process of assessing the goodness of fit for a non-linear regression model. What specific evaluation metrics and techniques can be used for non-linear regression performance analysis?
6. Discuss the importance of examining residual plots in the performance analysis of regression models. How can these plots help identify potential issues or violations of regression assumptions?
7. Explain the concept of overfitting in the context of regression analysis. How does overfitting affect the performance of a regression model, and what techniques can be used to mitigate it?
8. Describe the steps involved in comparing the performance of different regression models. What are some criteria and techniques that can be used to select the best model?
9. Discuss the assumptions underlying the performance analysis of linear regression models. Why is it important to assess and meet these assumptions before drawing conclusions from the analysis?
10. Explain the role of cross-validation in the performance analysis of regression models. How can cross-validation help in assessing a model's ability to generalize to new, unseen data?

Further Readings

- Madan Gopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Unit 07:Classification

CONTENTS

Objectives

Introduction

7.1 Introduction to Classification Problems

7.2 Decision Boundaries

7.3 Dataset

7.4 K-Nearest Neighbours (k-NN)

7.5 Decision Tree

7.6 Building Decision Tree

7.7 Training and visualizing a Decision Tree

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further readings

Objectives

- Understanding the classification problems and types of classification.
- Understanding the different parameters for building decision trees.
- Understanding the concepts of k-Nearest neighbours algorithm.
- Understanding the difference between decision tree and random forest.
- Understanding the fundamentals of decision boundaries.

Introduction

In this unit, we will study k-Nearest Neighbors algorithm and Decision Tree Algorithmfor the classification problems. We will understand how the classification problems will be handled along with types of classification. Dataset preparation is one of the very important aspects for machine learning, which is already covered in the previous units, is also highlighted here.In this unit, we focus more on decision boundaries and building the decision trees including training and testing. The concepts of linearly separable data and non-linearly separable data is discussed in a simplest wary. Examples are given whenever it is needed for the explanation. Similarly, K-Nearest Neighbors algorithm is also discussed thoroughly. Let us see one by one.

7.1 Introduction to Classification Problems

Let us start with classification directly. Classification is the process of segregating the given data into specific category.Name of category is known as class labels.For example, if you consider the vehicle data, then you can consider the class labels as Two Wheelers and Four Wheelers. Two Wheelers will refer all the type of bikes having two wheels and Four Wheelers will refer all type of cars and other vehicles having four wheels.Classification can be performed on the given data such as images; numbers, text or sometimes the data can be the mixed type. The preprocessing concepts

and feature engineering can be helpful to convert the data into the format that machine learning algorithm understands the data.Assume that the data is ready for classification and proceed further.

Classification is categorized into two types based on the number of ways the data can be classified, which are binary classification and multiclass classification.

Binary Classification

This has only two class labels.The input data is classified into two parts.For example, as per our previous discussion, let us assume the class labels 'two-wheelers' and 'not two-wheelers'. The machine learning algorithm has knowledge about only one thing that is 'two-wheelers'. The outcome will be either YES or NO.Machine gives YES if the given data is about two-wheelers and gives NO if the given data is not about two-wheelers. Training Data and testing data are used to train the machine and used to check the efficiency of the trained machine respectively.

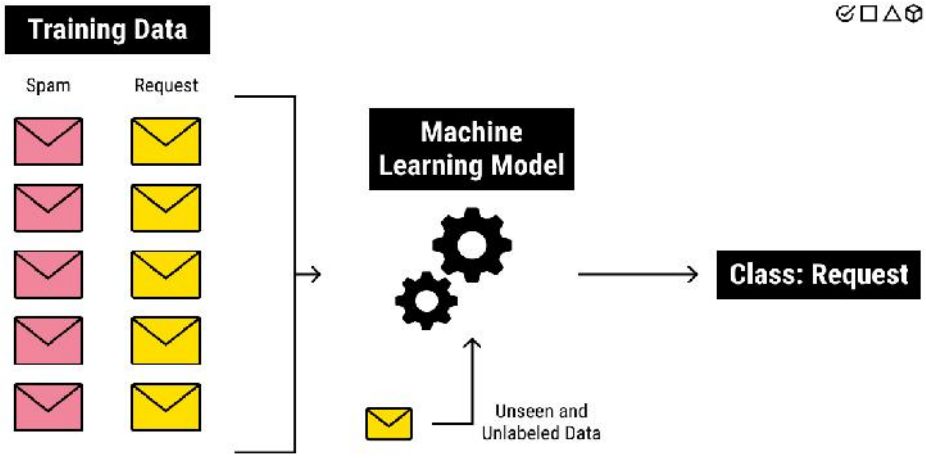


Figure 1 Spam filter

The above example is depicting the binary classification processed for classifying spam mails and non-spam mails as shown in figure 1.

Multiclass Classification

This type of classification is having more than two class labels as shown in figure 2.The training data should be prepared accordingly. The machine will be having knowledge about multiple things. Figure 2 and 3 is explaining the concepts and an example to demonstrate the multiclass classification.

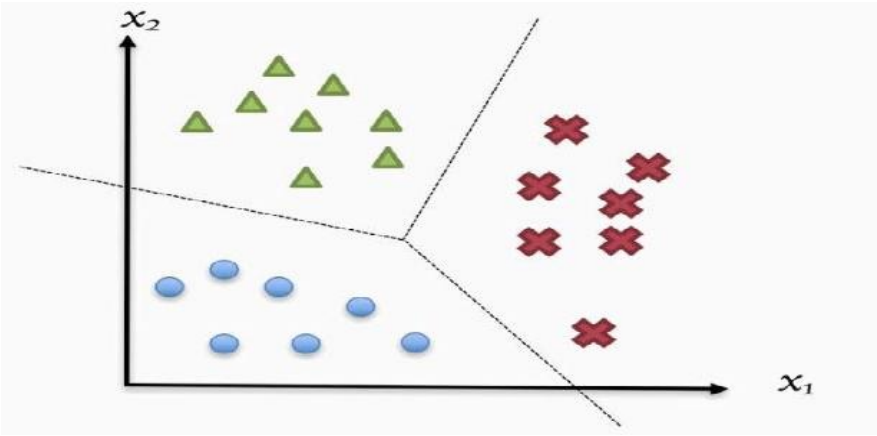


Figure 2 Multi-class classifications

The above figure 2 depicting the multi-class with different colors as blue, green and red for example. Hence, it is three class classification problems.

Figure 3Handwritten Digits Classification

The above figure 3 is a very important classification problem that is handwritten digits classification. The sample data presented here, collected from 15 people for the digits from 0 to 9. The machine learning algorithms should detect the given number correctly. This is understood as multi-class classification.

7.2 Decision Boundaries

The data is said to be linearly separable if a line is enough to classify the given data as shown in the Figure 5. The black colored line is said to be the decision boundary. The dots of blue colored and red colored are the data of different categories. So, the decision boundary is dividing exactly the different categories of data. Hence, we can say that data are successfully classified.

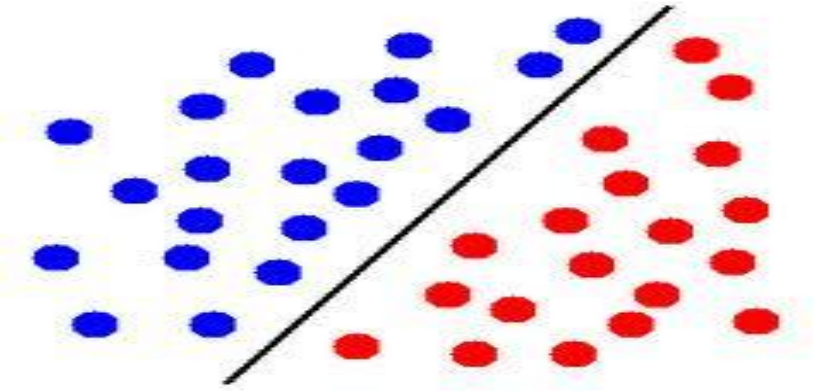


Figure 5 Linearly separable data

The data is said to be non-linearly separable if the non-linear shape can only classify the given data as shown in Figure 6. The problem of classification is successfully attempted here too as discussed earlier.

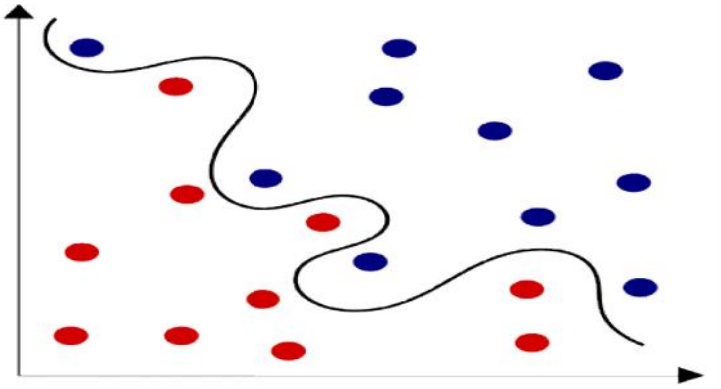


Figure 6 Non-linearly separable data

Many machine learning algorithms are available for performing classification tasks. Few are, Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machines, Naïve Bayes, Random forestand Artificial Neural Networks. We should understand each of the algorithms with their working model, but we here highlight few in this unit such as decision tree and K-Nearest Neighbours.One of the important challenges in Machine Learning is difficulty in judging how much data is enough for learning. The issues can be known as underfitting and overfitting as shown in Figure 4.

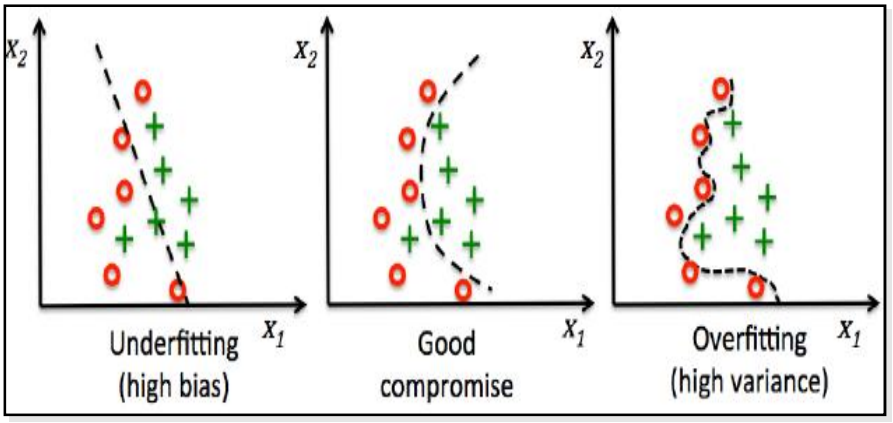


Figure 4 Machine learning challenges

When the size of the training data is very less, then Machine learning algorithm will learn too much and it can't be able to understand the new data. Hence, it will fail and it is known as overfitting issues. Also note here that variance is high.

Similarly, when the size of the training data is too much, then Machine learning algorithm will not able to learn properly. This will lead to the failure of machine learning algorithm and it is known as underfitting issues. Note that bias is high here.

Always the balancing between the variance and the bias is little tricky and require experience. It badly needs the concepts of advanced machine learning algorithm to deal with.

7.3 Dataset

We use the readymade dataset from the website "UC Irvine Machine Learning Repository". This is used for the education purpose only. Majority of the machine learning tasks can be performed with the available datasets from the given repository. The link is given at the end of this document. And, the dataset from Kaggle platform is also another option for you for practice. Image processing subject knowledge is needed to handle image data. Video processing subject knowledge is needed to handle video data. Signal processing subject knowledge is needed to handle voice data. Natural Language Processing subject knowledge is needed to handle text data. The term 'data' is a generic thing. The domain knowledge and the knowledge to handle the given data are very much required to perform any classification task.

7.4 K-Nearest Neighbours (k-NN)

This is a simple and intuitive method of classification. This does not assume anything about the data structure and it is therefore known as data-driven. This k-NN algorithm is a classification technique that does not make assumptions about the model between the class membership (y) and the features (x1, x2, x3, … xn). Hence It is not model-driven.This model free algorithm finds similar past patterns or instances from the training set with the use of a suitable distance measure and interpolates from them the correct output. In machine learning, these are also known as instance based or memory based learning algorithms.

k-NN approach of classification will emerge from the concept of non parametric estimation of probability density function of a pattern distribution. We first determine the k-points that are closest neighbors of 'x' with the help of a specific distance metric. The categorization of 'x' is, then given by the class label found in most of the k-neighbors. All neighbors have equal vote and the

class with the most number of votes among the k neighbors is selected. If k=1, then it is usually not sufficient for determining the class of 'x' due to noise and outliers in the data. A set of nearest neighbors is needed to accurately decide the class. The category computation of a new data point is shown in Figure 5. It depends on how many data points are near to the new data point. The majority of the category will be the category of this new point as shown in Figure 5.

The key issue of k-NN algorithm is the distance function or similarity function, which is selected on the basis of applications and nature of the data. The cautious selection of an appropriate distance function is a crucial step in the use of k-NN. Another crucial thing is to select the number of neighbors. This relies heavily on the problem being solved, as the number is dependent on the data distribution. This needs investigation of varying numbers of neighbors and end up with an optimal number with the help of validation set. The training set also should be given more focus by ensuring whether data set has enough examples of all possible categories. It should be a balanced training set.Overall, it is recommended to have an odd number for k to avoid ties in classification, and cross-validation tactics can help you choose the optimal k for your dataset.

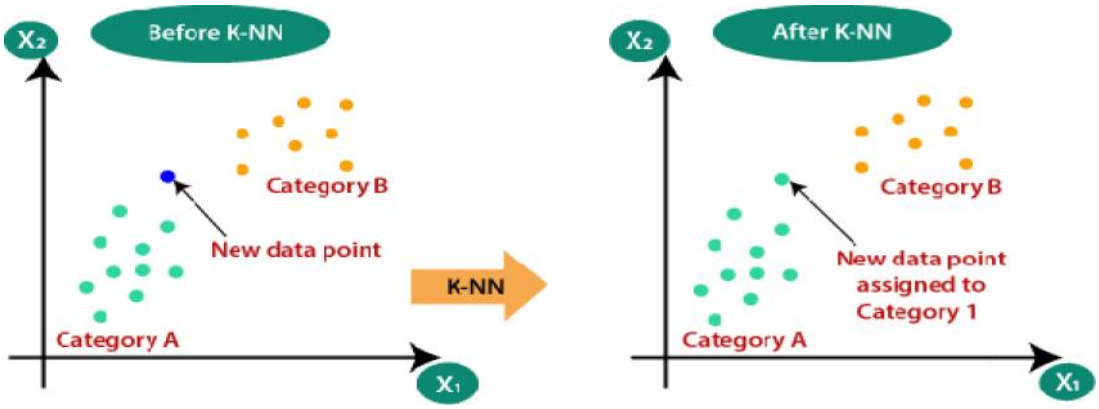


Figure 5 k-NN with a sample point

Python implementation is given below for k-Nearest Neighbor algorithm:

```
from sklearn.neighbors import KNeighborsClassifier

model_name = 'K-Nearest Neighbor Classifier'

knnClassifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p=2)

knn_model = Pipeline(steps=[('preprocessor', preprocessorForFeatures), ('classifier' , knnClassifier)])

knn_model.fit(X_train, y_train)

y_pred = knn_model.predict(X_test)
```

7.5 Decision Tree

Decision tree can be said to be a map of reasoning process. It is a hierarchical set of rules explaining the way in which a large set of data can be divided into smaller data partitions. The structure of the decision tree is understood from the given figure 6. When each time a split takes place, the components of the resulting partitions become increasingly similar to one another with regard to the target.And also, Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression problems.

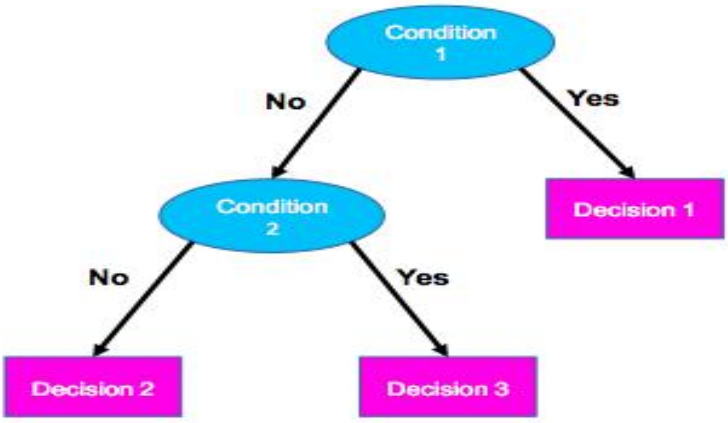


Figure 6 Decision Tree

Components of Decision Tree

- Nodes
- Edges / branches
- Leaves

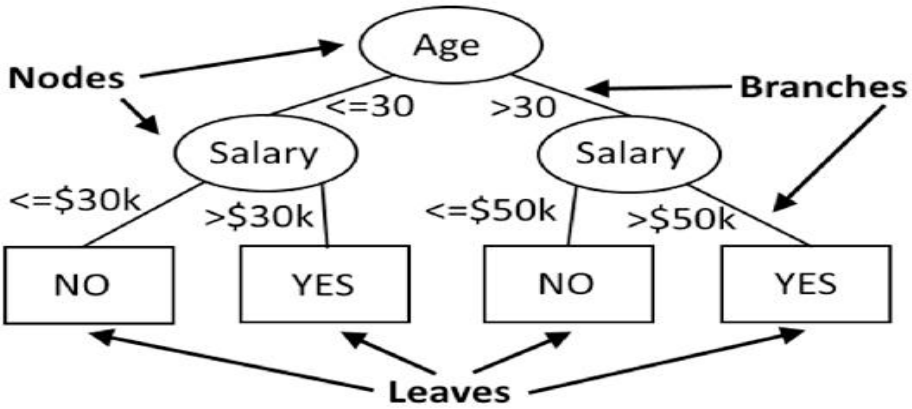


Figure 7 Decision Tree components

We can remember the hypothesis of Decision Tree that Tree size should be minimum and one who gets the required answer with less number of questions is assumed as efficient.

Rule based system can be understood with an example as given below. According to the given rule, the applications of customers who have been employed for a maximum of two years will be eligible for credit if their income is greater than or equal to fifty thousand rupees. Otherwise, it is understood that customer becomes not eligible for the credit.

If (years_employed>= 2) and (income >= 50000) then credit = Approved.

An algorithm starts with a learning set of instances or patterns and their associated class labels. The training set is portioned into smaller subsets in a sequence of recursive splits as the tree is being built. The tree building follows a top down hierarchical approach. The tree learning algorithms are said to be greedy, because at every stage, beginning at the root with complete dataset, this will search for the best split with non-backtracking. Let us have an example for a classification problem using decision tree. Assume the dataset as given in Table 1.

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Table 1 Dataset for Classification Problem

The input variables are x1 = Weather / Outlook, x2 = Temperature, x3 = Humidity and x4 = Wind. The target variable y = PlayTennis.The task is to predict the value of PlayTennis for an arbitrary day based on the values of its attributes. The decision tree what is built out of the above dataset (Table 1) is given below in Figure 8. The classification of an instance begins at the root node of the tree, where the specified attribute is tested and then it moves down the tree branch corresponding to the value of the attribute in the example stated below. The procedure is iterated for the subtree rooted at the new node.

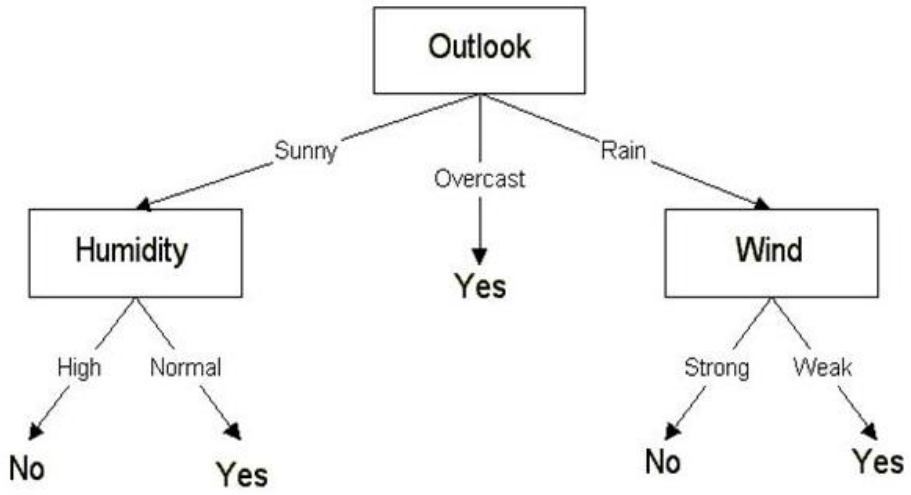


Figure 8 Decision Tree for (Outlook data) Classification problem

The only thing left to think is how to determine which attribute to split on, given a set of samples with different classes. Out of four attributes, which is the best choice to make it as root node?. The measure to evaluate a potential split is impurity of the target variable in the daughter nodes. High impurity means that the distribution of the target in the daughter nodes is similar to that of parent node, whereas low impurity means that members of a single class predominate. The best split is the one that decreases impurity in the daughter nodes by the greatest amount.There are three popular impuritymeasures as given below.

- Entropy
- Information Gain
- Gini Index

You can refer the textbook [Madan Gopal, Applied Machine Learning, McGraw Hill Education, India, 2018] for the computation.

7.6 Building Decision Tree

The dataset used for training the decision tree and the working model can be seen in the link. The final trained decision tree is shown in Figure 9.

https://sites.ualberta.ca/~hadavand/DataAnalysis/notebooks/DecisionTree.html

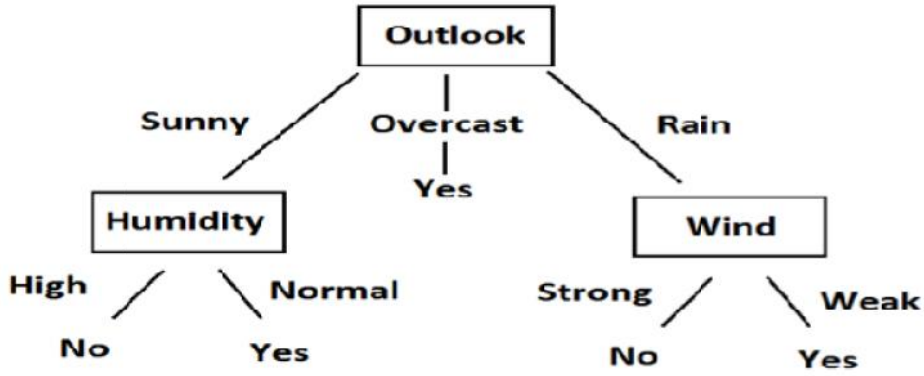


Figure 9 Decision Tree Model

The algorithm ID3 (Iterative Dichotomiser 3)is using the entropy and information gain as metric to select the attributes. At the same time, the algorithm CART (Classification and Regression Trees) is using the Gini Impurity as metric to select the attributes.For example, assume the case as given below.

		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

Table 2 Data summary of Outlook from table 1.

Calculate the entropy from the Table 1.

E(S) = -[(9/14)log(9/14) + (5/14)log(5/14)]

= 0.94

Calculate this entropy w.r.t outllokfrom the Table 2.

E(S, outlook) = (5/14)*E(3,2) + (4/14)*E(4,0) + (5/14)*E(2,3)

=(5/14)(-(3/5)log(3/5)-(2/5)log(2/5))+ (4/14)(0)

+ (5/14)((2/5)log(2/5)-(3/5)log(3/5))

= 0.693

Information gain for the Outlook Column is below.

IG(S, outlook) = 0.94 - 0.693

= 0.247

Similarly, we should calculate the information gain of other columns. The results are given below for your reference.

IG(S, Temperature) = 0.940 - 0.911 = 0.029

IG(S, Humidity) = 0.940 - 0.788 = 0.152

IG(S, Wind) = 0.940 - 0.8932 = 0.048

Now, we select the column / feature having the largest entropy gain. Here it is Outlook at the first place. The same kind of steps need to be carried out after every step, after every division of dataset in particular.

7.7 Training and visualizing a Decision Tree

The following python code will be used to train the decision tree and see the output.

- from sklearn.tree import DecisionTreeClassifier
- Model = DecisionTreeClassifier (criterion = 'entropy')
- Model.fit(x_train, y_train)
- Output = Model.predict (x_test)

Visualizing a decision tree is already discussed in the previous section 7.5 onwards. Decision tree algorithm is a display algorithm. You must be knowing by this time all the components of a decision tree, how to design it and how to program it using the above code.

Summary

- Understood the classification problems and types of classification.
- Understood the different parameters for building decision trees.
- Understood the concepts of k-Nearest neighbours algorithm.
- Understood the difference between decision tree and random forest.
- Understood the fundamentals of decision boundaries.

Keywords

- Classification
- k-Nearest Neighbours
- Decision Tree
- Distance Metrics

Self Assessment

1. Which of the following will be Euclidean Distance between the two data point A(1,3) and B(2,3)?

A. 8
B. 4
C. 2
D. 1

2. Machine learning algorithms build a model based on the sample data known as _____.

A. Training Data
B. Transfer Data
C. Validation Data
D. Test Data

3. In k-NN, what will happen when you decrease / increase the value of k?

A. Smoothness of boundary doesn't dependent on value of K

B. The boundary becomes smoother with decreasing value of K
C. The boundary becomes smoother with increasing value of K
D. None of the above

4. High entropy means that the partitions in classification are _____.
A. Pure
B. Notpure
C. Useful
D. Useless

5. Decision tree is the most powerful for _____
A. Classification
B. Prediction
C. Both (a) and (b)
D. None of these

6. Decision trees can handle_____.
A. High dimensional data
B. Low dimensional data
C. Mediumdimensional data
D. Noneof these

7. Decision-tree algorithm falls under the category of _____
A. Unsupervisedlearning algorithms
B. Reinforcementlearning algorithm
C. Supervisedlearning algorithms
D. Proneto errors in classification problems with many class

8. _____is the measure of uncertainty of a random variable and it characterizes the impurity of an arbitrary collection of examples.
A. Information Gain
B. Entropy
C. Gini Index
D. Noneof these

9. Bootstrap and Aggregation, commonly known as_____
A. Bagging
B. Information Gain
C. Entropy
D. Noneof these

10. k-NN algorithm does more computation on test time rather than train time.
A. True
B. False

11. Decision Tree is a display of an algorithm.

A. True

B. False

12. Decision Tree Nodes are represented by _____.

A. Circles

B. Squares / Rectangles

C. Triangles

D. All of the above

13. _____ refers to a model that can neither models the training data nor generalizes to new data.

A. Goodfitting

B. Overfitting

C. Underfitting

D. Allof the above

14. "Decision Tree can be used for classification and regression problems". Justify the given statement.

A. True

B. False

15. Which of the following is a disadvantage of decision trees?

A. Factor analysis

B. Decision trees are robust to outliers

C. Decision trees are prone to be overfit

D. None of the above

Answers for Self Assessment

1.	D	2.	A	3	C	4.	B	5.	C
6.	A	7.	C	8.	B	9.	A	10.	A
11.	A	12.	B	13.	C	14.	A	15.	C

Review Questions

1. Explain the different types of classification with examples.
2. List the various distance metrics used in k-NN.
3. Explain the process of designing a decision tree with an example.
4. Explain in detail about the selection of best node.
5. Highlight the important things about Entropy, Information Gain and Gini Index.

Further readings

- MadanGopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://archive.ics.uci.edu/ml/index.php
- https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html
- https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
- https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

Unit 08:Classification Algorithms

CONTENTS

Objectives

Introduction

8.1 Introduction to Classification Algorithms

8.2 Dataset

8.3 Logistic Regression

8.4 Support Vector Machine

8.5 Types of Kernels

8.6 Margin and Hyperplane

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further Readings

Objectives

1. Understanding the classification problems and types of classification.
2. Understanding how the SVM is used for classification.
3. Understanding the concepts of higher dimensional space used in SVM.
4. Understanding the different types of kernels in detail.
5. Understanding the logistic regression and its implementation.

Introduction

In this unit, we will study two important algorithms especially on the classification perceptive. They are Logistic regression algorithm and support vector machine algorithm. We will understand how the classification problems will be handled along with types of classification. Also, Dataset preparation is one of the very important aspects for machine learning, which is already covered in the previous units, is also highlighted.How the data is provided to logistic regression and support vector machine is given in more elaborately.The fundamental concepts of SVM, in particular, the concepts of hyperplane and margin are discussed here along with different types of kernels in SVM. Examples are given whenever it is needed for the explanation. Let us explore one by one.

8.1 Introduction to Classification Algorithms

Many machine learning algorithms are available for performing classification tasks not only binary classification but also multiclass classification. Few are, Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machines, Naïve Bayes, Random forest and Artificial Neural Networks. We should understand all of the algorithms with their working model, but we here highlight only two in this unit such aslogistic regression and support vector machines.One of the important challenges in Machine Learning is difficulty in judging how much data is enough for learning. The issues can be known as under fitting and overfitting as shown in Figure 1. When the

size of the training data is very less, then Machine learning algorithm will learn too much and it can't be able to understand the new data. Hence, it will fail which is known as overfitting issues. Also note here that variance is high. Similarly, when the size of the training data is too much, then Machine learning algorithm will not able to learn properly. This will lead to the failure of machine learning algorithm, whichis known as under fitting issues. Note that bias is high here.

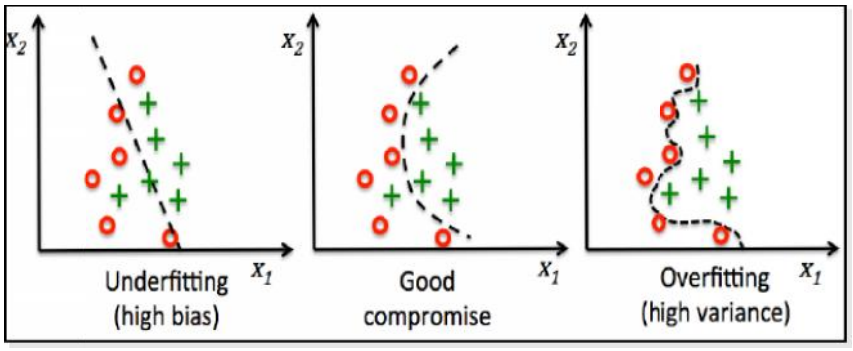


Figure 1 Machine learning challenges

Always the balancing between the variance and the bias is little tricky and require experience. It badly needs the concepts of advanced machine learning algorithm to deal with.First of all, we need proper dataset for the classification task, which is given in the next section.

8.2 Dataset

We use the ready-made dataset from the website "UC Irvine Machine Learning Repository". This is suggested for the education purpose only. Majority of the machine learning tasks can be performed with the available datasets from the given repository. The link is given at the end of this document. Home page of the site is given in Figure 2. And, the dataset from Kaggle platform is also another option for you for practice. Image processing subject knowledge is needed to handle image data. Video processing subject knowledge is needed to handle video data. Signal processing subject knowledge is needed to handle voice data. Natural Language Processing subject knowledge is needed to handle text data. The term 'data' is a generic thing. The domain knowledge and the knowledge to handle the given data are very much required to perform any classification task.

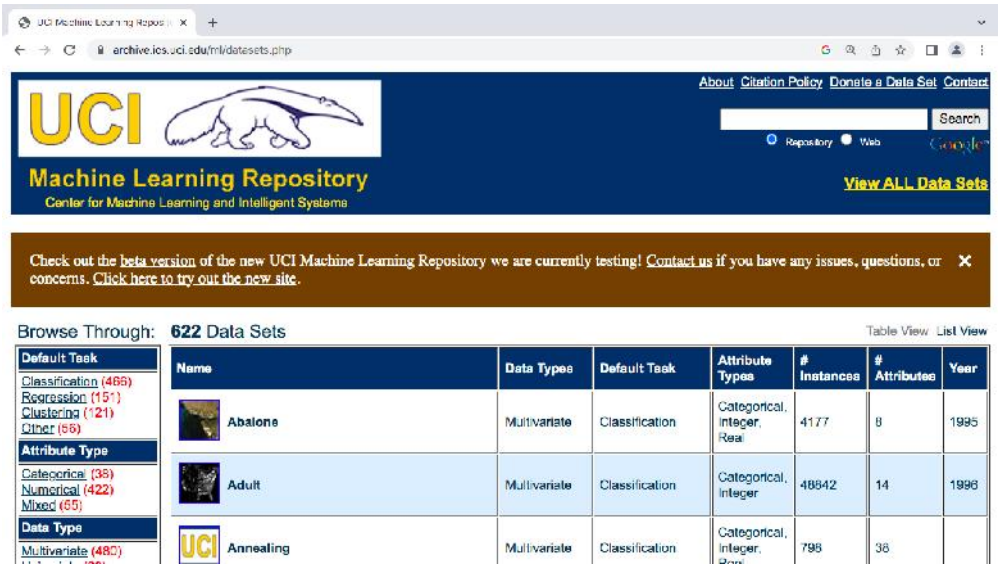


Figure 2 UCI Repository Home Page

8.3 Logistic Regression

Let us start with a question first – what is regression? Regression is a statistical measurement that attempts to determine the strength of the relationship between dependent variable and independent variable. This has two algorithms as given below.

- Linear Regression
- Logistic Regression

The following diagram will give you an idea on the working model how it looks like. Linear regression is usually used for regression problems. For example, we can try to predict the gold price after 5 years based on the previous 25 years of data. This problem is the popular example for regression. Similarly, prediction may be done in agricultural yield prediction, population prediction, average income per person and etc.The said prediction problemsare possible in the regression, with the help of the previous data. If previous data is correct then the prediction will be correct. In the sense that, we should take at most care in data collection. Logistic regression algorithm is used for the classification problems. Both are of supervised algorithms. Both the algorithms are using two terminologies, i.e., dependent variable and independent variable as in Figure 3. Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.There are three types of logistic regression models, which are defined based on categorical response. They are discussed below.

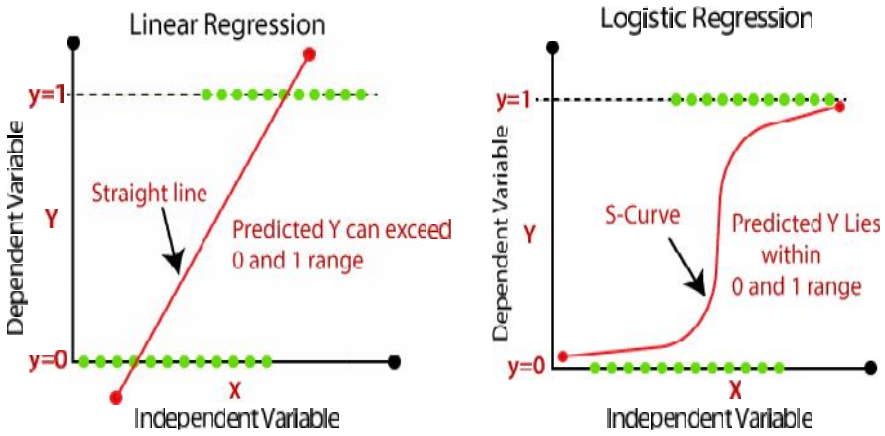


Figure 3 Linear and Logistic Regression Models

Binary logistic regression

In this approach, the dependent variable has only two possible outcomes i.e., the value will be either 0 or 1.

Multinomial logistic regression

In this approach, the dependent variable has three or more possible outcomes; however, these values have no specified order.

Ordinal logistic regression

This type of model is leveraged when the response variable has three or more possible outcome, but these values do have a defined order. Examples of ordinal responses include grading scales from A to F or rating scales from 1 to 5.

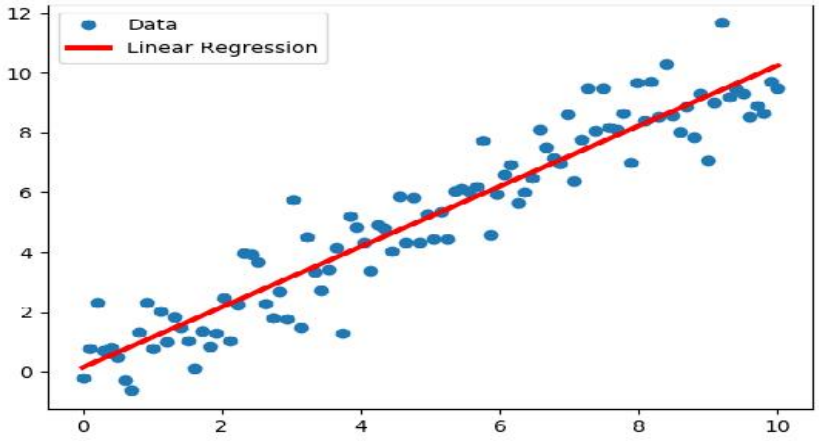


Figure 4 Regression Model – AnExample

As we already mentioned, the linear regression model is used for regression problems. These regression concepts are understood by this time. We will look at the different types of regression models as below.

Line regression

Look at the figure 4. Let us assume that all the blue colored points are previously collected data on the specific domainuptothe year 2022. I want to know, how the data will be for the year 2030. Simply, forecasting the values based on the presented or given data. The goal is to predict the value of a dependent variable based on independent variables.Least Squares Method is used as Metric to select the good regression line for the given data as shown in Figure 5 and Figure 6. Figure 5 is looking for the optimal regression line, which will best fit the given data and the orientation. We need to check multiple options or multiple lines to conclude the better one.

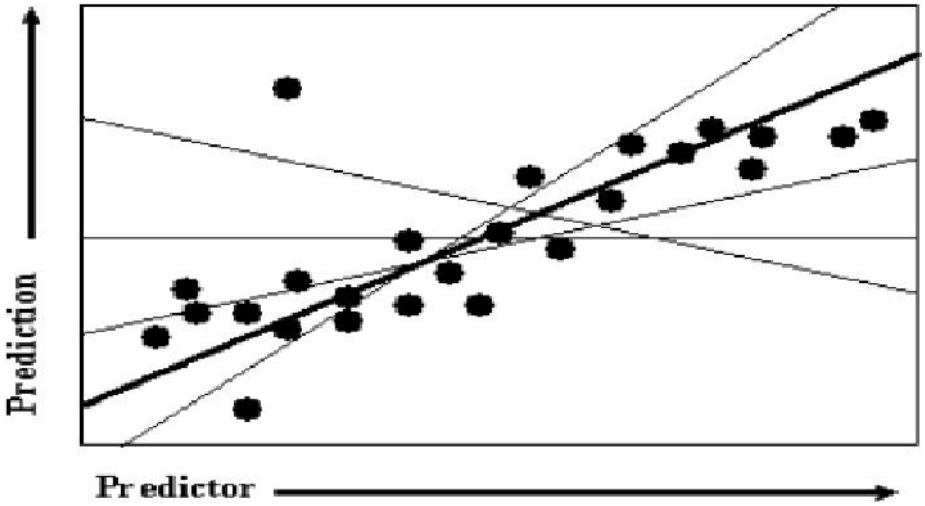


Figure 5 Searching for the optimal regression line

Figure 6 depicting the process of least square error calculation. The line, which will be having less error value comparatively, that, will be the optimal regression line. The distance between the line and the point is considered as 'e' as shown in figure 6. Summation of all the errors is used in the formula, which is discussed in Figure 7.

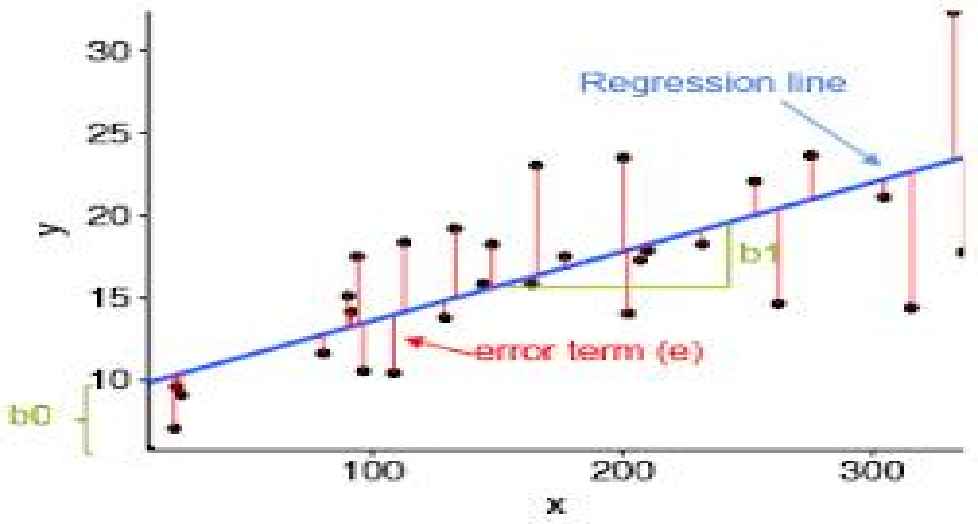


Figure 6 Least square error calculation

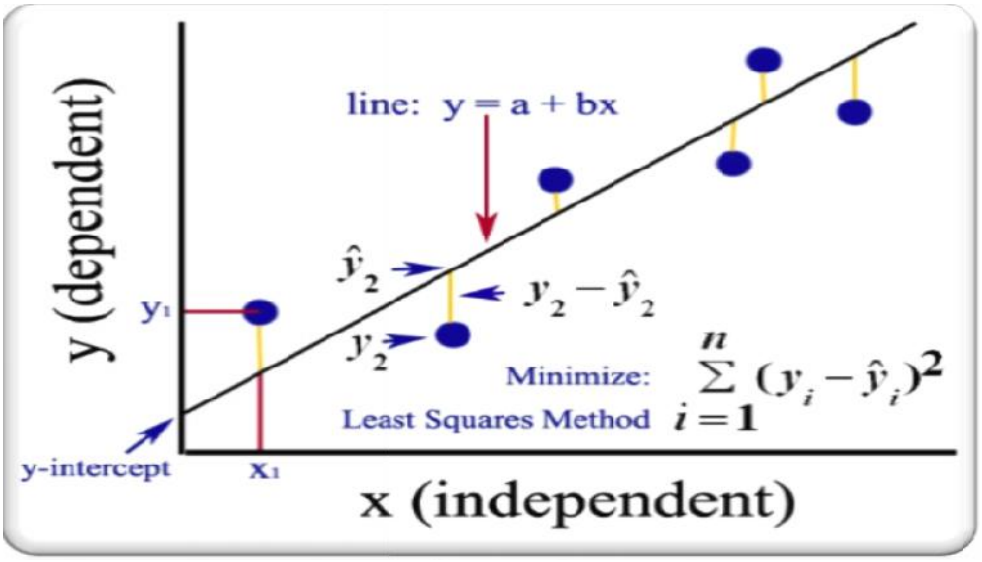


Figure 7 Formula and the terms in Least square error

Polynomial Regression

Polynomial Regression is a regression algorithm that models the relationship between a dependent and independent variables as nth degree polynomial. The model is given in the Figure 8.

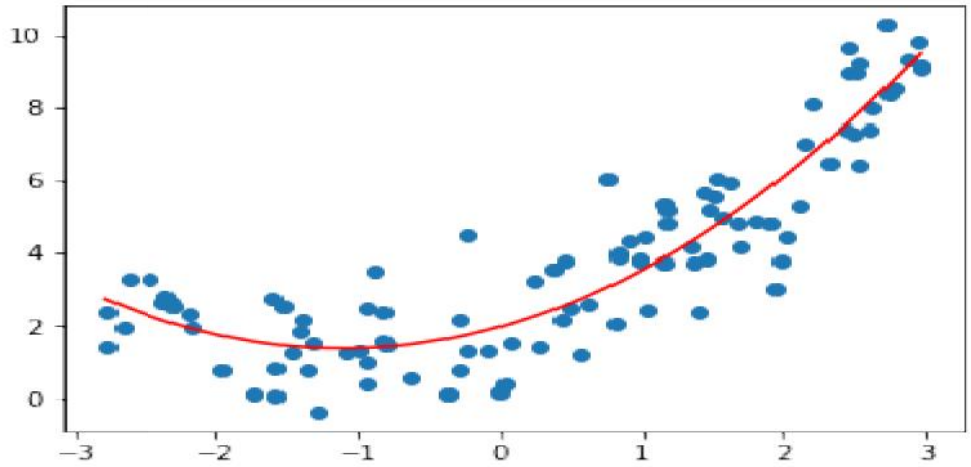


Figure 8 Polynomial regression model

Logistic regression is using the sigmoid function. Sigmoid is a mathematical function that takes any real number and maps it to a probability between 1 and 0. The function is given in Figure 9. Recall the figure 3 for more clarity. The goal is to find the logistic regression function $p(\mathbf{x})$ such that the predicted responses $p(\mathbf{x}_i)$ are as close as possible to the actual response y_i for each observation $i = 1, 2, 3, 4, 5,\ldots, n$. Once you have the logistic regression function $p(\mathbf{x})$, you can use it to predict the outputs for new and unseen inputs, assuming that the underlying mathematical dependence is unchanged.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Figure 9 Sigmoid function

8.4 Support Vector Machine

Support Vector Machine is a supervised machine-learning algorithm. This is popularly known as SVM. This algorithm uses kernel functions for classification as shown in Figure 10. SVM basically developed for binary classification. This Figure 10 depicts the classification separating by a hyperplane shown in red color line.

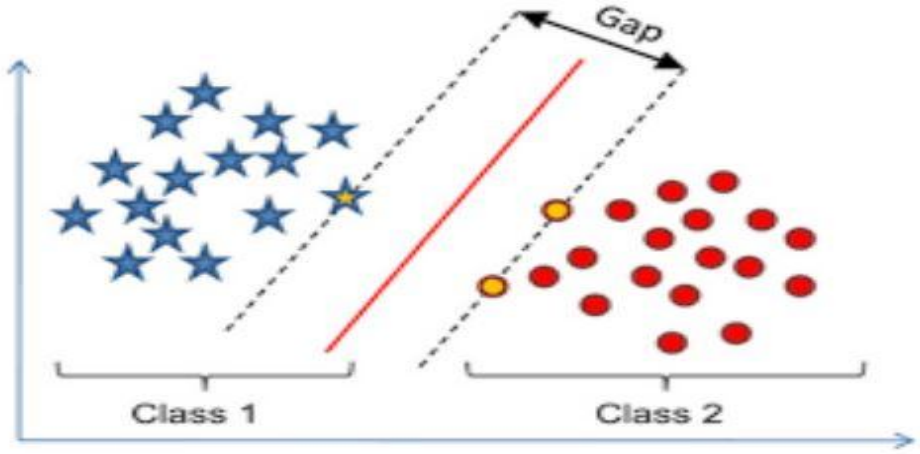


Figure 10Classification using SVM

Kernels are using higher dimensional space for effective classification as shown in Figure 11.

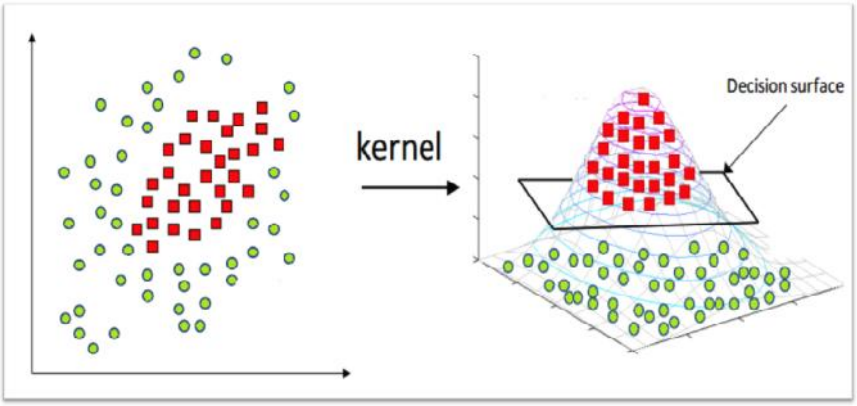


Figure 11 SVM Kernel Trick

The process of finding the higher dimensional / n-Dimensional space is very trickywhere the classification will be done efficiently. This is the actual task of the kernels as shown in Figure 12 for an example. The data given in the left side is non-linearly separable in the 2-dimensional space, but kernels find a higher dimensional space where the given data is linearly separable. There are three types of kernels function commonly used such as, linear kernel function, polynomial kernel function and radial basis kernel functions.

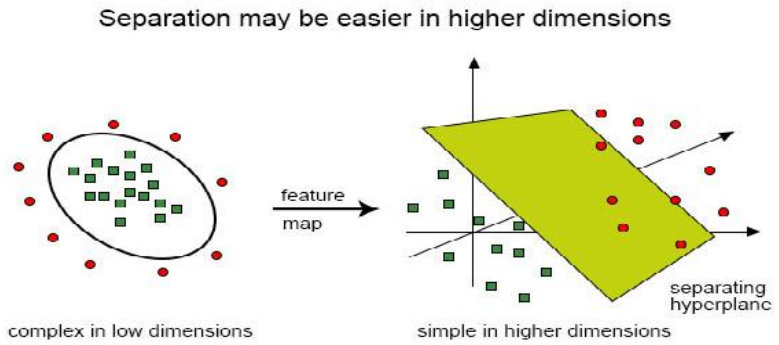


Figure 12 SVM Kernel Trick

Python Libraries for classification

- from sklearn.svm import SVC
- classifier=SVC(kernel = 'linear')
- classifier.fit(X_train, y_train)
- classifier.predict (X_test)

Python Libraries for regression

- from sklearn.svm import SVR
- regres = SVR( )
- X = [[0, 0], [2, 2]]
- y = [0.5, 2.5]
- regres.fit(X, y)
- regres.predict([[1, 1]])

8.5 Types of Kernels

Kernel Function is a method used to take data as input and transform it into the required form of processing data. The formula is given in Table 1.

Linear Kernel

Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used.

```
classifier = SVC(kernel ='linear')
```

Polynomial Kernel

This function can be used as per the syntax given below using Python.

```
classifier = SVC(kernel ='poly', degree = 4)
```

Radial Basis Kernel

This function can be used as per the syntax given below using Python.

```
classifier = SVC(kernel ='rbf', random_state = 0)
```

Kernels	Formula
linear	$k(x, y) = x.y$
sigmoid	$k(x, y) = \tanh(ax.y + b)$
polynomial	$k(x, y) = (1 + x.y)^d$
RBF	$k(x, y) = \exp(-a\|x - y\|^2)$
exponential RBF	$k(x, y) = \exp(-a\|x - y\|)$

Table 1 Different Kernel Functions

8.6 Margin and Hyperplane

The objective of SVM is to maximize the margin. Margin is the distance between the hyperplane and the support vectors, the data, which are closest to the hyperplane. In SVM, large margin is considered a good margin. There are two types of margins hard margin and soft margin. The support vectors are clearly understood from the Figure 13. The middle line (dotted line) is known as the hyperplane. The left margin is the distance between hyperplane and the support vectors (red color) and the right margin is the distance between the hyperplane and the support vectors (green color). The summation of left and right margin should be higher, which is the objective of SVM.

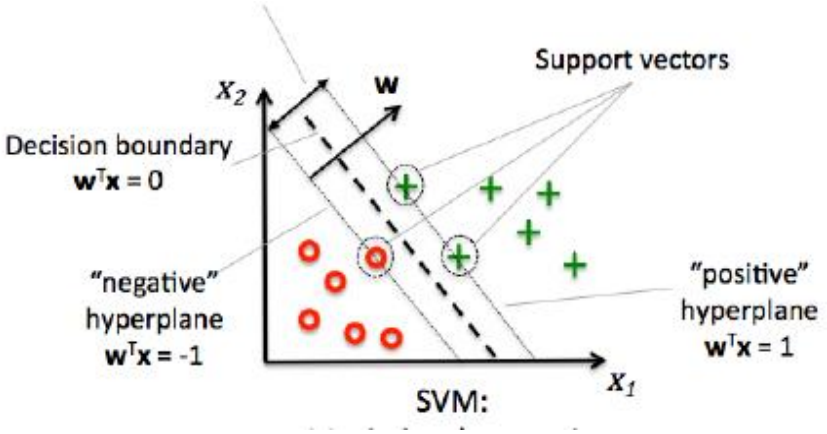


Figure 13 SVM Margin and Hyperplane

The equation of each negative and positive hyperplane is clearly visible in the Figure 14 along with the hyperplane, which is depicted in red color line. The maximum gap is also given in the Figure 14. Hence, it is understood by this time that we need to adjust or tune the weights until we obtain the higher margin, which is used to separate the positive and negative class data. Analysis on the higher dimensional space must be carried out simultaneously in this process of finding maximum margin.

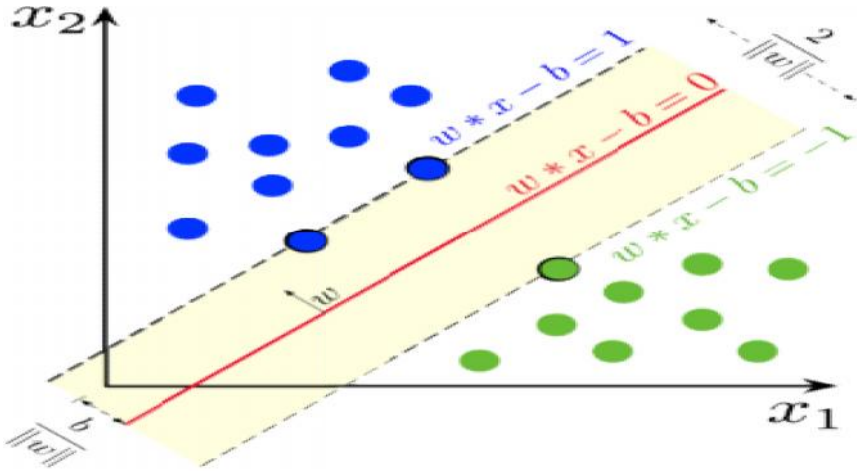


Figure 14 SVM Margin and Hyperplane

Summary

- Understood the classification problems and different types of classification.
- Discussed the difference between regression and classification.
- The basic concepts of logistic algorithm with an example are discussed.
- The fundamentals of support vector machinealgorithm along with their margin, hyperplane, support vectors are explained with examples.

Keywords

- Classification
- Kernel
- Support Vector Machines
- Logistic Regression
- Hyperplane
- Margin

Self Assessment

1. Logistic regression is used when you want to _____.

A. Predict a continuous variable from dichotomous variables.

B. Predict any categorical variable from several other categorical variables.

C. Predict a continuous variable from dichotomous or continuous variables.

D. Predict a dichotomous variable from continuous or dichotomous variables.

2. Machine learning algorithms build a model based on the sample data known as _____.

A. Training Data

B. Transfer Data

C. Validation Data

D. Test Data

3. Support vector machines is _____ algorithm.

A. Clustering
B. Unsupervised Machine Learning
C. Supervised Machine Learning
D. Reinforcement learning

4. SVM stands for_____.
A. System Vector Machines
B. Support Vector Machines
C. Support Vector Migrations
D. System Virus Mitigation

5. How many different types of Logistic Regression?
A. 1
B. 2
C. 3
D. 4

6. _____ the target variable can have three or more possible values without any order.
A. Multinomial Logistic Regression
B. Binary Logistic Regression
C. Ordinal Logistic Regression
D. All of the above

7. Logistic regression is basically_____.
A. Classification
B. Reinforcement
C. Supervised
D. Unsupervised
8. Linear regression assumes that the data follows a linear function; Logistic regression models the data using the _____.
A. Linearfunction
B. Sigmoidfunction
C. Continues function
D. Samplefunction

9. Using Linear Regression, all predictions are _____.
A. less than 0.5
B. greater than 0.5
C. less than 1
D. greater than 1

10. What do you mean by a hard margin?
A. The SVM allows very low error in classification
B. The SVM allows high amount of error in classification
C. Both (A) and (B)
D. None of the above

11. Support vectors are the data points that lie closest to the decision surface.

A. TRUE

B. FALSE

12. Which of the following are real world applications of the SVM?

A. Text and Hypertext Categorization

B. Image Classification

C. Clustering of News Articles

D. All of the above

13. _____ refers to a model that can neither models the training data nor generalizes to new data.

A. Goodfitting

B. Overfitting

C. Underfitting

D. Allof the above

14. Maximizing the distances between nearest data point and hyper plane will help us to decide the right hyper-plane is related to _____.

A. Margin

B. Mercer's Theorem

C. Regression

D. None of these

15. Which of the following can only be used when training data are linearly separable?

A. The centroid method

B. Linear Logistic Regression

C. Linear Soft margin SVM

D. Linear hard-margin SVM

Answers for Self Assessment

1.	D	2.	A	3	C	4.	B	5.	C
6.	A	7.	C	8.	B	9.	A	10.	A
11.	A	12.	D	13.	C	14.	A	15.	D

Review Questions

1. Explain the different types of classification with examples.
2. What do you understand by the concept of hyperplane and margin?
3. Describe and explain the process of kernels in SVM.
4. Explain in detail about the decision tree classifier.
5. Highlight the important things about random forest classifier.

Further Readings

- MadanGopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://archive.ics.uci.edu/ml/index.php
- https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html
- https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

Unit 09: Classification Implementation

CONTENTS

Objectives

Introduction

9.1 Datasets

9.2 K-Nearest Neighbour using Iris Dataset

9.3 Support Vector Machine using Iris Dataset

9.4 Logistic Regression

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further Readings

Objectives

- To understand the classification problems.
- To implement the K-Nearest Neighbours Algorithm for classification problems.
- To implement the Support Vector Machine Algorithm for classification problems.
- To understand the usages of different kernels in Python Code.
- To implement the logistic regression algorithm for classification problems.

Introduction

Many machine-learning algorithms are available for performing classification tasks not only binary classification but also multiclass classification. Few are, Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machines, Naïve Bayes, Random forest and Artificial Neural Networks. In this unit, we will try to understand k-nearest neighbor algorithm, support vector machine algorithm and logistic regression algorithm in detail. The above-mentioned algorithms will be implemented using python language. The programs are also given in this section one by one with outcome of each execution. Let us explore one by one.

9.1 Datasets

We use the ready-made dataset from the website "UC Irvine Machine Learning Repository" as in Fig 1.We are going to use Iris Dataset as in the Fig 2 for our implementation. At the same time, you can also try with some other datasets as in Fig 3 and Fig 4. You can download those datasets and use it in the python program. Otherwise, you can refer the web link directly in the python code.It will read the data directly.

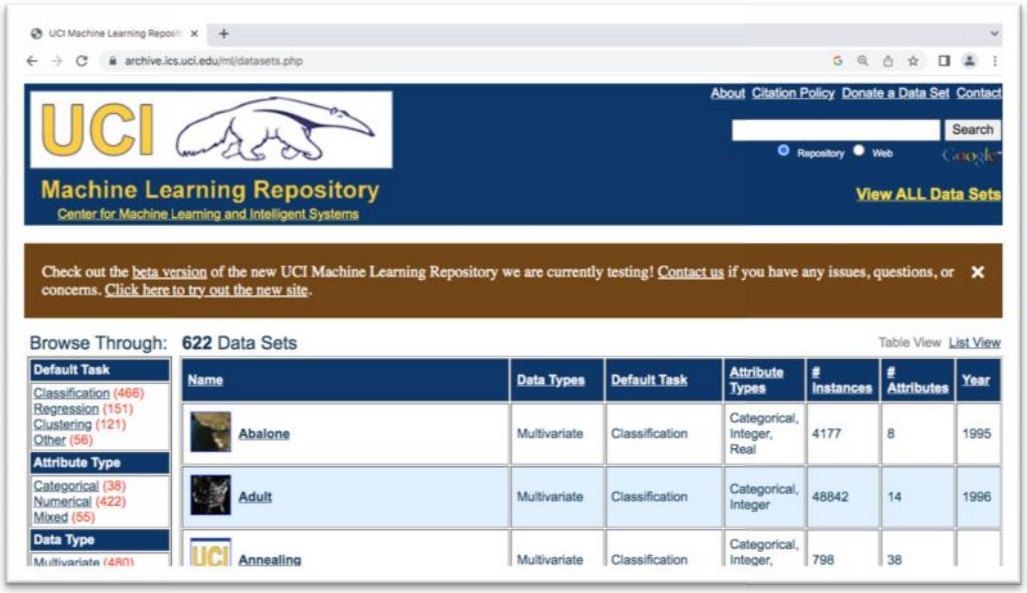


Figure 1 UCI Repository Home Page

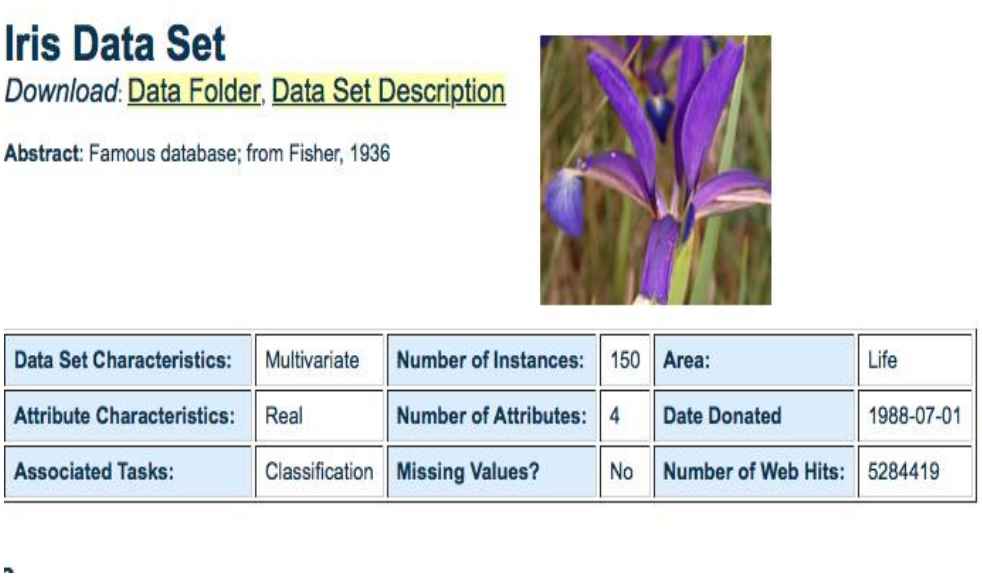
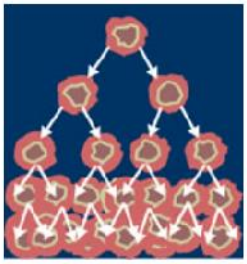


Fig 2 Iris Dataset

Breast Cancer Data Set

Download: Data Folder, Data Set Description

Abstract: Breast Cancer Data (Restricted Access)



Data Set Characteristics:	Multivariate	**Number of Instances:**	286	**Area:**	Life
Attribute Characteristics:	Categorical	**Number of Attributes:**	9	**Date Donated**	1988-07-11
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	701529

Fig 3 Breast Cancer Dataset

Algerian Forest Fires Dataset Data Set

Download: Data Folder, Data Set Description

Abstract: The dataset includes 244 instances that regroup a data of two regions of Algeria.

Data Set Characteristics:	Multivariate	**Number of Instances:**	244	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	12	**Date Donated**	2019-10-22
Associated Tasks:	Classification, Regression	**Missing Values?**	N/A	**Number of Web Hits:**	75966

Fig 4 Algerian Forest Fire Dataset

9.2 K-Nearest Neighbour using Iris Dataset

We have to install scikit learn package from the below given command in order to use the libraries of all the machine learning algorithms including standard datasets.

pip install scikit-learn

Here is the code for K-Nearest Neighbor implementation using Iris Dataset.

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

In [2]: # Using IRIS DATASET

In [3]: df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', hea

In [4]: #https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv
        #https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv
        #https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer/breast-cancer.data

In [5]: #X = df.iloc[:, [0,1,2,3]].values
        X = df.iloc[:, :-1].values
        y = df.iloc[:, 4].values
```

```
[6]: X

t[6]: array([[5.1, 3.5, 1.4, 0.2],
             [4.9, 3. , 1.4, 0.2],
             [4.7, 3.2, 1.3, 0.2],
             [4.6, 3.1, 1.5, 0.2],
             [5. , 3.6, 1.4, 0.2],
             [5.4, 3.9, 1.7, 0.4],
             [4.6, 3.4, 1.4, 0.3],
             [5. , 3.4, 1.5, 0.2],
             [4.4, 2.9, 1.4, 0.2],
             [4.9, 3.1, 1.5, 0.1],
             [5.4, 3.7, 1.5, 0.2],
             [4.8, 3.4, 1.6, 0.2],
             [4.8, 3. , 1.4, 0.1],
             [4.3, 3. , 1.1, 0.1],
             [5.8, 4. , 1.2, 0.2],
             [5.7, 4.4, 1.5, 0.4],
             [5.4, 3.9, 1.3, 0.4],
             [5.1, 3.5, 1.4, 0.3],
             [5.7, 3.8, 1.7, 0.3],
```

```
[7]: y

:[7]: array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
```

```
[8]: #Converting the labels (y) into numbers ( 0 for Iris-setosa, 1 for Iris-versicolor, 2 for Iris-virginica )

[9]: from sklearn import preprocessing
     le = preprocessing.LabelEncoder()
     y = le.fit_transform(y)

10]: y

10]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
[11]: from sklearn.model_selection import train_test_split

[12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

[13]: X_train.shape

t[13]: (105, 4)

[14]: X_test.shape

t[14]: (45, 4)

[15]: y_train.shape

t[15]: (105,)

[16]: y_test.shape

t[16]: (45,)
```

```
[17]: from sklearn.preprocessing import StandardScaler
      sc = StandardScaler()
      sc.fit(X_train)
      sc.fit(X_test)

[17]: StandardScaler(copy=True, with_mean=True, with_std=True)

[18]: X_train_std = sc.transform(X_train)
      X_test_std = sc.transform(X_test)

[19]: from sklearn.neighbors import  KNeighborsClassifier
      knn = KNeighborsClassifier(n_neighbors = 5, p = 2, metric = 'minkowski')

[20]: knn.fit(X_train_std, y_train)

[20]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                 metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                 weights='uniform')

[21]: y_pred = knn.predict(X_test_std)
```

```
[22]: from sklearn.metrics import accuracy_score

[23]: print('misclassified samples: %d'%(y_test!=y_pred).sum())

      misclassified samples: 4

[24]: print('Accuracy:%.2f'%accuracy_score(y_test,y_pred))

      Accuracy:0.91

[25]: from sklearn.metrics import classification_report
      print(classification_report(y_test, y_pred))

                   precision    recall  f1-score   support

                0       1.00      1.00      1.00        12
                1       0.82      0.93      0.87        15
                2       0.94      0.83      0.88        18

      avg / total       0.92      0.91      0.91        45
```

9.3 Support Vector Machine using Iris Dataset

Support Vector Machine is a supervised machine-learning algorithm.This is popularly known as SVM.The same is implemented with the help of Iris Dataset using Python Language. All the codes are shown here along with their outputs.

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

In [2]: # Using IRIS DATASET

In [3]: df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', hea

In [4]: #https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv
        #https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv
        #https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer/breast-cancer.data

In [5]: #X = df.iloc[:, [0,1,2,3]].values
        X = df.iloc[:, :-1].values
        y = df.iloc[:, 4].values
```

```
 [6]: X

t[6]: array([[5.1, 3.5, 1.4, 0.2],
             [4.9, 3. , 1.4, 0.2],
             [4.7, 3.2, 1.3, 0.2],
             [4.6, 3.1, 1.5, 0.2],
             [5. , 3.6, 1.4, 0.2],
             [5.4, 3.9, 1.7, 0.4],
             [4.6, 3.4, 1.4, 0.3],
             [5. , 3.4, 1.5, 0.2],
             [4.4, 2.9, 1.4, 0.2],
             [4.9, 3.1, 1.5, 0.1],
             [5.4, 3.7, 1.5, 0.2],
             [4.8, 3.4, 1.6, 0.2],
             [4.8, 3. , 1.4, 0.1],
             [4.3, 3. , 1.1, 0.1],
             [5.8, 4. , 1.2, 0.2],
             [5.7, 4.4, 1.5, 0.4],
             [5.4, 3.9, 1.3, 0.4],
             [5.1, 3.5, 1.4, 0.3],
             [5.7, 3.8, 1.7, 0.3],
```

```
 [7]: y

:[7]: array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
```

[8]:
```
#Converting the labels (y) into numbers ( 0 for Iris-setosa, 1 for Iris-versicolor, 2 for Iris-virginica )
```

[9]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
y = le.fit_transform(y)
```

[10]:
```
y
```

[10]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

[11]:
```
from sklearn.model_selection import train_test_split
```

[12]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

[13]:
```
X_train.shape
```

[13]:
```
(105, 4)
```

[14]:
```
X_test.shape
```

[14]:
```
(45, 4)
```

[15]:
```
y_train.shape
```

[15]:
```
(105,)
```

[16]:
```
y_test.shape
```

[16]:
```
(45,)
```

[17]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(X_train)
sc.fit(X_test)
```

[17]:
```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

[18]:
```
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
```

[19]:
```
from sklearn.svm import SVC
svc = SVC(kernel='rbf')
svc.fit(X_train_std, y_train)
```

[19]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

[20]:
```
y_pred = svc.predict(X_test_std)
```

[21]:
```
print('Misclassified samples: %d' %(y_test != y_pred).sum())
```

```
Misclassified samples: 2
```

```
[22]: from sklearn.metrics import accuracy_score
      print('Accuracy:%.2f'%accuracy_score(y_test, y_pred))
```

```
Accuracy:0.96
```

```
[24]: #print('Coefficients or w = ',svc.coef_)
```

```
[25]: print('Intercept or b = ',svc.intercept_)
```

```
Intercept or b =  [-0.02218956 -0.15617086  0.01977951]
```

```
[26]: print('Index in X_train - Acted as support vectors = ', svc.support_)
```

```
Index in X_train - Acted as support vectors =  [  8  65  76  81  86  88  89   0  13  17  18  25  26  29  37  55  59
61
  66  72  73  82  92  93 100   3  15  24  27  34  36  39  47  53  54  69
  70  74  78  87 101]
```

```
[27]: print('Support vectors = ', svc.support_vectors_)
```

```
Support vectors =  [[-8.00057777e-01  5.85390655e-01 -1.12576284e+00 -8.80146763e-01]
 [-1.12110007e-01  1.69222172e+00 -1.12576284e+00 -1.15235710e+00]
 [-9.14715739e-01 -7.87079872e-02 -1.18100747e+00 -1.28846227e+00]
 [-1.14403166e+00 -7.87079872e-02 -1.29149673e+00 -1.15235710e+00]
 [-9.14715739e-01  1.02812308e+00 -1.18100747e+00 -7.44041593e-01]
 [-1.71732147e+00 -7.87079872e-02 -1.45723063e+00 -1.42456744e+00]
 [-1.12110007e-01  3.02041901e+00 -1.23625210e+00 -1.01625193e+00]
 [-1.02937370e+00 -1.40690527e+00 -2.41848723e-01 -1.99620915e-01]
 [ 5.75837763e-01 -1.18553906e+00  6.42065392e-01  4.80904932e-01]
 [ 1.14912757e+00 -5.21440415e-01  5.86820760e-01  3.44799763e-01]
 [-8.00057777e-01 -1.18553906e+00 -4.07582620e-01 -6.35157458e-02]
 [ 1.03446961e+00 -7.87079872e-02  6.97310024e-01  7.53115271e-01]
 [ 2.31863878e-01 -7.42806629e-01  7.52554656e-01  6.17010102e-01]
 [-9.14715739e-01 -2.29237013e+00 -1.31359459e-01 -1.99620915e-01]
 [-3.41425930e-01 -1.62827148e+00  1.44863702e-01  2.08694593e-01]
 [ 1.37844350e+00  3.64024441e-01  5.31576128e-01  3.44799763e-01]
 [ 1.26378553e+00  1.42658227e-01  6.42065392e-01  4.80904932e-01]
 [ 8.05153687e-01 -5.21440415e-01  4.76331495e-01  4.80904932e-01]
 [-2.26757969e-01 -7.87079872e-02  4.21086863e-01  4.80904932e-01]
 [-4.56083892e-01 -7.87079872e-02  4.21086863e-01  4.80904932e-01]
 [ 1.17205916e-01  3.64024441e-01  5.86820760e-01  8.89220441e-01]
 [ 1.03446961e+00  1.42658227e-01  5.31576128e-01  4.80904932e-01]
```

```
[28]: print('Number of support vectors for each class = ', svc.n_support_)
```

```
Number of support vectors for each class =  [ 7 18 16]
```

```
[29]: from sklearn.metrics import classification_report
      print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	0.86	1.00	0.92	12
2	1.00	0.88	0.94	17
avg / total	0.96	0.96	0.96	45

9.4 Logistic Regression

Let us try the classification problem using the logistic regression algorithm. The python code is given here in detail along with the proper execution and the relevant outputs. You can execute the same code in Jupyter Notebook as shown here.

```
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt

[2]: # Using IRIS DATASET

[3]: df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', hea

[4]: #https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv
     #https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv
     #https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer/breast-cancer.data

[5]: X = df.iloc[:, :-1].values    #X = df.iloc[:, [0,1,2,3]].values
     y = df.iloc[:, 4].values
```

```
[6]: X

:[6]: array([[5.1, 3.5, 1.4, 0.2],
             [4.9, 3. , 1.4, 0.2],
             [4.7, 3.2, 1.3, 0.2],
             [4.6, 3.1, 1.5, 0.2],
             [5. , 3.6, 1.4, 0.2],
             [5.4, 3.9, 1.7, 0.4],
             [4.6, 3.4, 1.4, 0.3],
             [5. , 3.4, 1.5, 0.2],
             [4.4, 2.9, 1.4, 0.2],
             [4.9, 3.1, 1.5, 0.1],
             [5.4, 3.7, 1.5, 0.2],
             [4.8, 3.4, 1.6, 0.2],
             [4.8, 3. , 1.4, 0.1],
             [4.3, 3. , 1.1, 0.1],
             [5.8, 4. , 1.2, 0.2],
             [5.7, 4.4, 1.5, 0.4],
             [5.4, 3.9, 1.3, 0.4],
             [5.1, 3.5, 1.4, 0.3],
             [5.7, 3.8, 1.7, 0.3],
```

```
[7]: y

t[7]: array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
             'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolo
      r',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
             'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
```

```
[8]: #Converting the labels (y) into numbers [ 0 for Iris-setosa, 1 for Iris-ver
```

```
[9]: from sklearn import preprocessing
     le = preprocessing.LabelEncoder()
     y = le.fit_transform(y)
```

```
[10]: y
```

```
[10]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
[11]: from sklearn.model_selection import train_test_split
```

```
[12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
[13]: X_train.shape
```

```
[13]: (105, 4)
```

```
[14]: X_test.shape
```

```
[14]: (45, 4)
```

```
[15]: y_train.shape
```

```
[15]: (105,)
```

```
[16]: y_test.shape
```

```
[16]: (45,)
```

```
[17]: from sklearn.preprocessing import StandardScaler
      sc = StandardScaler()
```

```
[18]: sc.fit(X_train)
      sc.fit(X_test)
```

```
[18]: StandardScaler(copy=True, with_mean=True, with_std=True)
```

```
[19]: X_train_std = sc.transform(X_train)
      X_test_std = sc.transform(X_test)
```

```
[20]: from sklearn.linear_model import LogisticRegression
      lr = LogisticRegression(C=1000.0, random_state=0)
```

```
[21]: lr.fit(X_train_std, y_train)
```

```
[21]: LogisticRegression(C=1000.0, class_weight=None, dual=False,
                fit_intercept=True, intercept_scaling=1, max_iter=100,
                multi_class='ovr', n_jobs=1, penalty='l2', random_state=0,
                solver='liblinear', tol=0.0001, verbose=0, warm_start=False)
```

```
[22]: y_pred=lr.predict(X_test_std)

[23]: print('misclassified samples: %d'%(y_test!= y_pred).sum())
      misclassified samples: 2

[24]: from sklearn.metrics import accuracy_score
      print('Accuracy:%.2f'%accuracy_score(y_test,y_pred))
      Accuracy:0.96

[25]: from sklearn.metrics import classification_report
      print(classification_report(y_test, y_pred))
                   precision    recall  f1-score   support

                0       1.00      1.00      1.00        13
                1       1.00      0.90      0.95        20
                2       0.86      1.00      0.92        12

      avg / total       0.96      0.96      0.96        45
```

Summary

- Understood how to read the iris dataset directly using web-link.
- We have implemented K-Nearest Neighbour Algorithm and the performance was 91%.
- Support Vector Machine algorithm is implemented and the accuracy was 96%.
- Radial basis function was used in SVM as the kernel function for the implementation.
- Thelogistic regression algorithm gave the performance of 96%.
- Thedataset is preprocessed with Standard Scaler function and the same is used for training and testing.

Keywords

- **Classification**
- **Kernel**
- **Support Vector Machines**
- **Logistic Regression**
- **Hyperplane**
- **Margin**

Self Assessment

Q1) Logistic regression is used when you want to _____ .

A Predict a continuous variable from dichotomous variables.

B Predict any categorical variable from several other categorical variables.

C Predict a continuous variable from dichotomous or continuous variables.

D Predict a dichotomous variable from continuous or dichotomous variables.

Q2) Machine learning algorithms build a model based on the sample data known as _____.

A. Training Data

B. Transfer Data
C. Validation Data
D. Test Data

Q3) Support vector machines is _____ algorithm.

A. Clustering
B. Unsupervised Machine Learning
C. Supervised Machine Learning
D. Reinforcement learning

Q4) SVM stands for_____.

A. System Vector Machines
B. Support Vector Machines
C. Support Vector Migrations
D. System Virus Mitigation

Q5) How many different types of Logistic Regression?

A. 1
B. 2
C. 3
D. 4

Q6) _____ the target variable can have three or more possible values without any order.

A. Multinomial Logistic Regression
B. Binary Logistic Regression
C. Ordinal Logistic Regression
D. All of the above

Q7) Logistic regression is basically_____.

A Classification
B Reinforcement
C Supervised
D Unsupervised

Q8) Linear regression assumes that the data follows a linear function; Logistic regression models the data using the _____.

A Linearfunction
B Sigmoidfunction
C Continues function
D Samplefunction

Q9) Using Linear Regression, all predictions are _____.

A. less than 0.5
B. greater than 0.5
C. less than 1
D. greater than 1

Q10) What do you mean by a hard margin?

A. The SVM allows very low error in classification
B. The SVM allows high amount of error in classification
C. Both (A) and (B)
D. None of the above

Q11) Support vectors are the data points that lie closest to the decision surface.

A. True
B. False

Q12) Which of the following are real world applications of the SVM?

A. Text and Hypertext Categorization
B. Image Classification
C. Clustering of News Articles
D. All of the above

Q13) _____ refers to a model that can neither models the training data nor generalizes to new data.

A. Goodfitting
B. Overfitting
C. Underfitting
D. Allof the above

Q14) Maximizing the distances between nearest data point and hyper plane will help us to decide the right hyper-plane is related to _____.

A. Margin
B. Mercer's Theorem
C. Regression
D. None of these

Q15) Which of the following can only be used when training data are linearly separable?

A. The centroid method
B. Linear Logistic Regression
C. Linear Soft margin SVM
D. Linear hard-margin SVM

Answers for Self Assessment

1.	D	2.	A	3	C	4.	B	5.	C
6.	A	7.	C	8.	B	9.	A	10.	A
11.	A	12.	D	13.	C	14.	A	15.	D

Review Questions

1. Whatis binary classification and multi-class classification? Giveexamples.
2. How do you access the standard datasets directly from sklearn library?
3. Describe the outputs of SVM algorithm when you use different kernels such as linear or polynomial.
4. Explain the preprocessing techniques required while using Breast Cancer Dataset.
5. Comment on the challenges faced when you use Algerian Forest Fires Datasets with respect to KNN, SVM and Logistic Regression algorithm.

Further Readings

- MadanGopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://archive.ics.uci.edu/ml/index.php
- https://scikit-learn.org/stable/modules/svm.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Unit 10:Clustering

CONTENTS

Objectives
Introduction
10.1 Introduction to Clustering
10.2 K-Means Algorithm
10.3 Mathematical Model of K-Means
10.4 Hierarchical Clustering
10.5 Types of Hierarchical Clustering
10.6 Linkage Methods
Summary
Keywords
Self Assessment
Answers for Self Assessment
Review Questions
Further Readings

Objectives

1. Understanding the fundamental concepts of clustering.
2. Understanding the working style ofK-Means Algorithms.
3. Understanding thelinkage methods used for hierarchical clustering.
4. Understanding in detail about types ofHierarchical algorithm.
5. Understanding the mathematical model of the clustering algorithms.

Introduction

In this unit, the basic concepts of clustering are discussed with necessary examples. This section introduces the popular clustering algorithm known as K-Means Clustering Algorithm in detail. The mathematical model of the algorithm is discussed along with different distance metrics used for computation of distances between the points. In the same way, the different types of clustering is also considered and discussed with examples. In particular, hierarchical clustering is focused at the best along with the linkage methods, which is very important for hierarchical clustering mechanisms.

10.1 Introduction to Clustering

Clustering is the unsupervised learning technique. Clustering is the process of grouping the similar data items.In the training datasets for clustering problems, outputs / labels are not specified. We know only the features / data with no idea how to group them. The groups are called as clusters. The data points within each cluster should be having high similarity but should be having high dissimilarity between the clusters. It also be mentionedasClusters should be homogeneous within and clusters should be heterogeneity between clusters. Clusters detection provides a way to learn about the structure of complex data. A natural way to make sense of complex data is to break the data into smaller clusters of data, and then it is easy to find patterns within each cluster. Group of

data are called clusters. The example is given in figure 1, where the data is grouped in three different clusters. All the clusters are encircled for a kind of representation.

Clustering is very much important as it determines the intrinsic grouping among the unlabeleddata. There is no specific criteria or condition to perform good clustering. It depends on the user and their requirements. In the next section, we introduce the K-Means clustering algorithm, which is commonly used by the academic and research community.

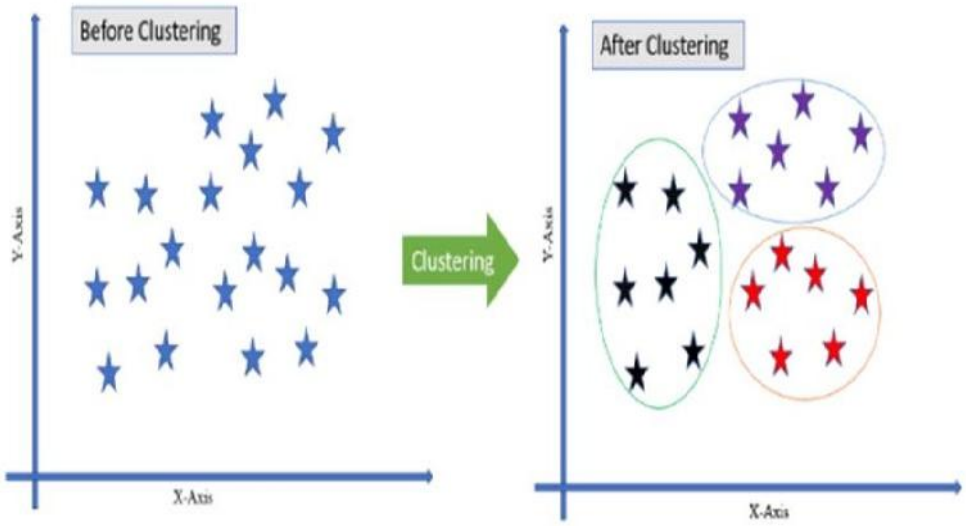


Figure1. Clustering with 3 - Clusters of data

10.2 K-Means Algorithm

K-means is a clustering algorithm. 'K' represents number of possible (pre-calculated) clusters. If k=2, then it means there are only two clusters. Algorithm then it performs grouping of data with respect to two clusters. Clusters are the distinct non-overlapping subgroups. Each data point belongs to only one group.As you know, that this algorithm does not need any kind of training as it is handling unlabeled data. This algorithm is performed based on the centroid point. The centroid will be the center point of a cluster. Each cluster will be having the centroid point. Our algorithm has to find the region of similar data and their centroid and the same will be carried out for other remaining clusters as shown in Figure 2.

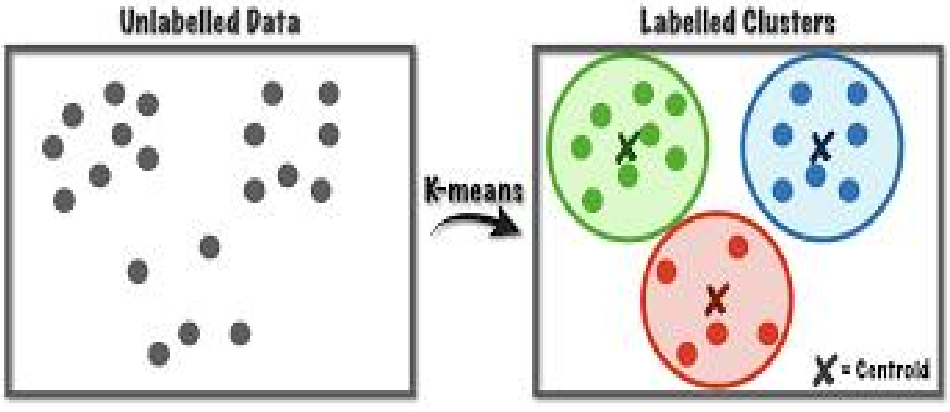


Figure 2 Clusters of data using K-Means

The algorithm steps are given below:

Step 1. Assume k-Clusters (eg. k=2).

Step 2. Choose two centroid pointsrandomly, say p_1 and p_4.

Step 3. Let p_1 belongs to $Cluster_1$ and p_4 belongs to $Cluster_2$.

Step 4. Calculate the Euclidean distances from centroid (p_1) to all the data points.

Step 5. Calculate the Euclidean distances from centroid (p_4) to all the data points.

Step 6. Arrange and group the data points according to their distances.

Step 7. Data points will be joining to their nearest clusters.

Step 8. Calculate the mean value of the data points in each clusters.

Step 9. Replace the old centroids with new values respectively.

Step 10. Repeat steps 2 to 5 with new centroids.

Step 11. Otherwise, stop the iterations.

This algorithm needs multiple iterations.On every iteration, we will have some change in the centroid points on each cluster as the data points are moving between one cluster to an another cluster and so on. But, there will be a state where there won't be any change in the centroid points. Let us assume it as stopping condition. We will get our final clusters. All the details can be understood from the above given steps of K-Means algorithm.

Python Library for K-Means

- from sklearn.cluster import KMeans
- kmeans = KMeans(n_clusters=4)
- kmeans.fit(X)
- y_kmeans = kmeans.predict(X)

10.3 Mathematical Model of K-Means

The similarity of data depends on the kind of data to be clustered. As the data usually describes features of objects in a numerical form, the ideal measure of similarity is by measuring the distance between the data points (vectors). The commonly employed measurement technique is Euclidean norm. We discuss here how the mathematical model of eucliean distance. Also, we discuss one more distance metric known as Manhattan distance. They are listed below.

- Euclidean Distance
- Manhattan Distance

The above two methods are commonly used to calculate distance. The formula is shown and explained in the following figure 3.

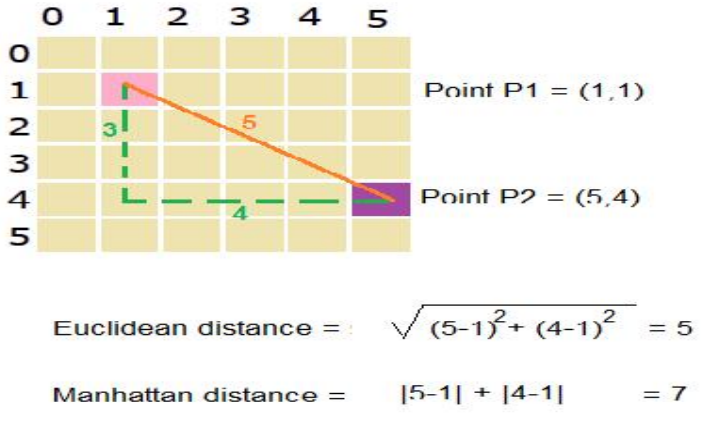


Figure 3 Mathematical Model of Distance Metrics

Measuring the Cluster Quality

The simplest and commonly used criterion for clustering is the sum-of-squared-error criterion. Let N_k be the number of samples (data) in cluster k and μ_k be the mean of those samples (data) is calculated as given in (1).

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{x}^{(i)} \quad (1)$$

and then the sum of squared errors is defined by as given in (2) where k stands for number of clusters.

$$J_e = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \left\| \mathbf{x}^{(i)} - \boldsymbol{\mu}_k \right\|^2 \quad (2)$$

10.4 Hierarchical Clustering

Hierarchical clustering is an unsupervised machine learning algorithm, which is used to group the unlabeled data in the form of clusters.The process starts from dividing the original data into clusters and the each clusters again divided into subclusters and so on until there are no more clustering is possible. This is known as hierarchical cluster analysis (HCA).Hierarchy is explained in figure 4. The figure can be understood in terms of power. The power is in the top, which is complete. Then the power is divided into three people. And then it is subsequently getting divided / distributed further until there is no division is possible. This is known as hierarchical approach used for dividing the data again and again that leads to sensible clusters.

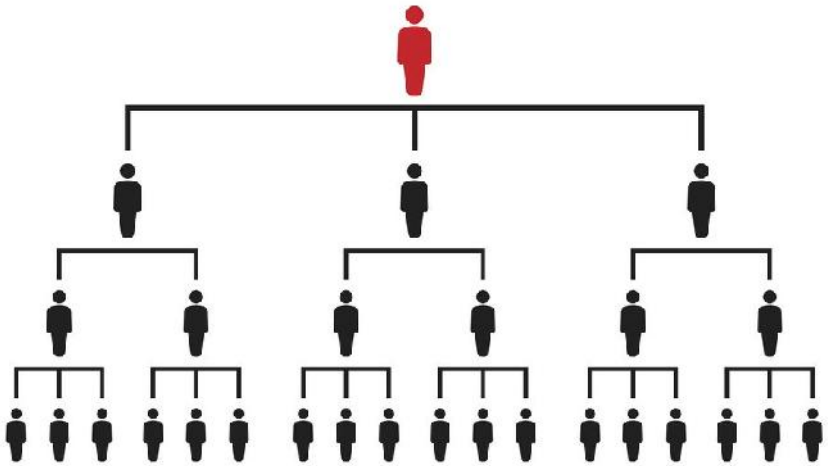


Figure 4 Sample for a hierarchy structure

A tree, which is used to represent the hierarchical clustering, is known as dendrogram. The figure 5 and figure 6 shows a dendrogram with six samples and large data respectively.

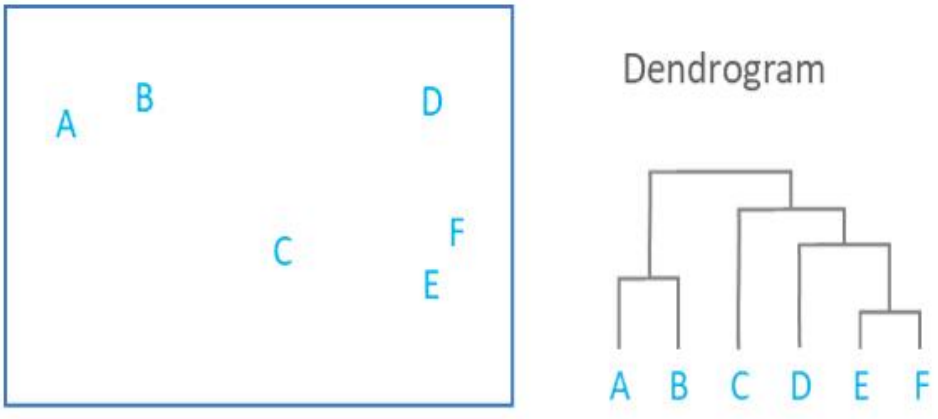


Figure 5 Dendrogram for small dataset

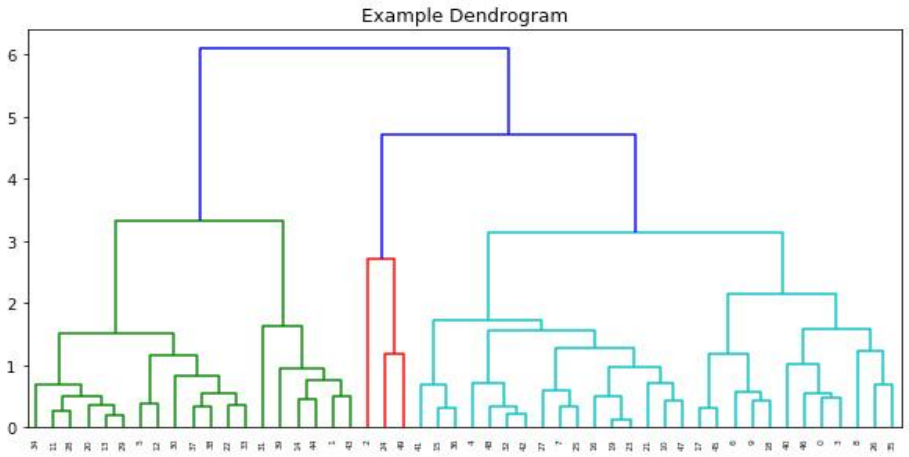


Figure 6 Dendrogram for large dataset

10.5 Types of Hierarchical Clustering

The process of hierarchical clustering can be categorized into distinct models. They are agglomerative and divisive models.Agglomerative model processes from the bottom and go in bottom-up approach, it starts from singleton (cluster having only one data) and followed by further merging of clusters until some stopping criteria is satisfied. The divisible model begins with single cluster at the top. Assume that we have now only one cluster that consists of all the points. It divides consecutively and separating the data into different clusters.These models are depicted in the figure 7.

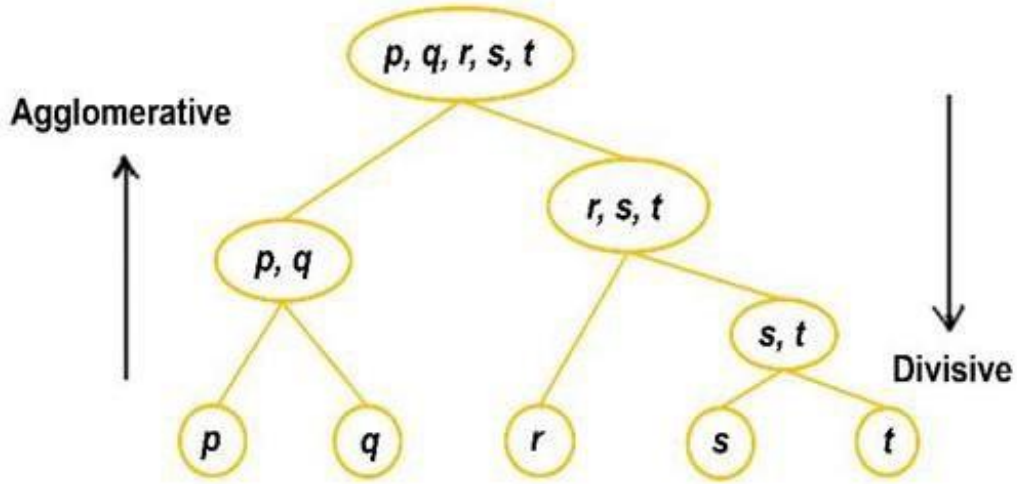


Figure 7 Types of hierarchical clustering

Agglomerative

- Initially each data point is considered as an individual cluster as shown in Figure 8.
- Clustering process starts then.
- Known as bottom up approach.

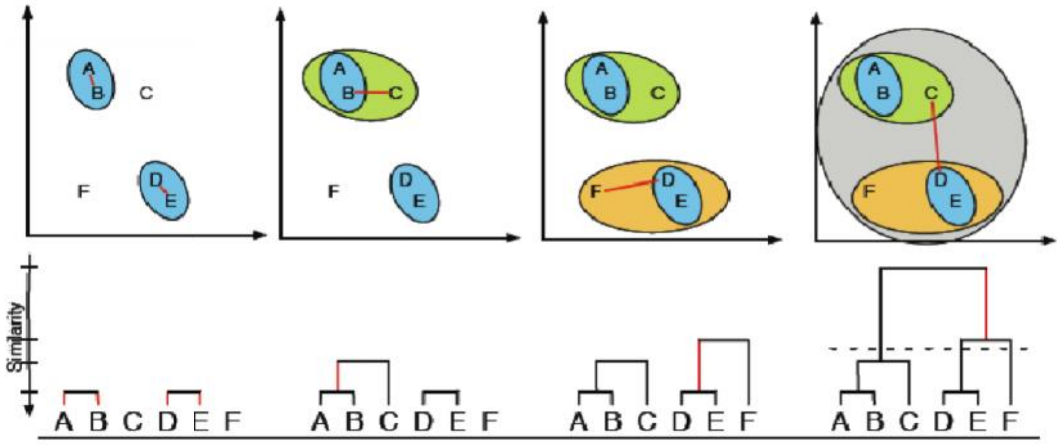


Figure 8 Agglomerative Clustering

Divisive

- Initially all the data points are considered as one cluster.
- Then, starts splitting into sub-clusters recursively.
- Known as top-down approach.
- To select which cluster to split, the SSE (sum of squared error) is used.
- The divisive may be understood from the above Figure 8 by looking from the directionright to left.

Python Library for Agglomerative Clustering

- from sklearn.cluster import AgglomerativeClustering
- clust = AgglomerativeClustering( )
- clust.fit(X)
- clust.labels_
- clust.n_clusters_
- clust.n_leaves_

10.6 Linkage Methods

On the above discussion, the primary requirement is to measure the distance between two clusters. Linkage Methods are used to find the distance between the two clusters.There are four commonly used measures which are listed below. They are single linkage, complete linkage, centroid linkage and centroid linkage methods. Euclidean Distance and Manhattan Distance are used to find the distance between two points.

- Single Linkage
- Complete Linkage
- Centroid Linkage
- Average Linkage

Single linkage calculates the minimum distance between clusters. This measure takes into consideration of only the two nearest members of a pair of clusters. This procedure or process is sensitive to outliers. Complete linkage calculates the maximum distance between the clusters. The centroid linkage calculates the distance between the centroids of each of two clusters. The average linkage calculates the average distance of all the possible distances between the clusters.

Summary

- Understood the fundamental concepts of clustering.

- Understood the working style of K-Means Algorithms.
- Understood the linkage methods used for hierarchical clustering.
- Understood in detail about types of Hierarchical algorithm.
- Understood the mathematical model of the clustering algorithms.

Keywords

- Clustering
- Euclidean distance
- Manhattan distance
- Hierarchical clustering
- Agglomerative model
- Divisive model
- Linkage methods

Self Assessment

1. The independent variable is also called _____.

A Regressor
B Regressand
C Predictand
D Estimated

2. The dependent variable is also called _____.

A Regressand Variable
B Predictand Variable
C Explained Variable
D All of the above

3. All data points falling along a straight line is called _____.

A Linear Relationship
B Non-linear Relationship
C Residual
D Scatter Diagram

4. Which of the following is finally produced by Hierarchical Clustering?

A Finalestimate of cluster centroids
B Treeshowing how close things are to each other
C Assignmentof each point to clusters
D All of the above

5. Identify the best method that is used for finding optimal clusters in k-means algorithm.

A. Euclidean method
B. Manhattan method
C. Elbow method
D. Silhouette method

6. _____clusters formed in this method forms a tree-type structure based on the hierarchy.

A. Density-Based
B. Hierarchical Based
C. Grid-based
D. None of these

7. Which of the following clustering requires merging approach?

A Partitional
B Hierarchical
C Naive Bayes
D None of the above

8. Hierarchical clustering should be primarily used for exploration.

A True
B False

9. Which of the following is required by K-means clustering?

A Defineddistance metric
B Numberof clusters
C Initialguess as to cluster centroids
D Allof the above

10. Justify the statement. "Divisive Clustering is not the hierarchical clustering methods".

A True
B False

11. Which of the following combination is incorrect?

A Continuous – euclidean distance
B Continuous – correlation similarity
C Binary – manhattan distance
D None of the above

12. Divisive has _____ approach.

A Topdown
B Bottomup
C Downup
D None of these

13. Which of the following clustering algorithm requires the number of clusters to be pre-specified?

A hierarchical clustering
B k-means clustering
C DBSCAN
D Markov clustering algorithm

14. Agglomerative has _____ approach.

A. Top down
B. Bottom up
C. Down up
D. None of these

15. _____methods have good accuracy and ability to merge two clusters.
A. Density-Based
B. Hierarchical Based
C. Grid-based
D. None of these

Answers for Self Assessment

1.	A	2.	D	3	A	4.	B	5.	C
6.	B	7.	B	8.	A	9.	D	10.	B
11.	D	12.	A	13.	B	14.	B	15.	A

Review Questions

1. Explain the computation of various distance metrics.
2. What do you understand by the concept of dendrogram?
3. Differenciate agglomerative and divisive hierarchical clustering.
4. Mention any two applications of clustering algorithms.
5. Explain the different linkage methods with examples.

Further Readings

- MadanGopal,Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://www.displayr.com/what-is-hierarchical-clustering/
- https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning
- https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/
- https://www.statisticshowto.com/probability-and-statistics/regression-analysis/
- https://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/

Unit 11: Ensemble Methods

CONTENTS

Objectives

Introduction

11.1 Ensemble Learning

11.2 Bagging

11.3 Boosting

11.4 Random Forests

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further Readings

Objectives

- Understanding the basics of ensemble methods.
- Understanding the concept of bagging.
- Understanding the structure of random forests.
- Understanding the difference between the decision tree and random forests.
- Understanding the working style of random forests.

Introduction

Ensemble learning methods are made up of a set of classifiers. All the output of each classifier will be taken and are aggregated to produce a final result. The most well-known ensemble methods are bagging or bootstrap aggregation and boosting. These methods are commonly used to reduce the variance within a noisy dataset. They are explained in detail with necessary examples. Finally, the concepts of random forest are explained with the working procedure. Also, we can understand the difference between the random forest and decision trees.These units will emphasis the importance of ensemble learning over machine learning algorithms.

11.1 Ensemble Learning

Ensemble learning helps to improve machine-learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. For the given amount of data and given quality of training data, the output of one hypothesis function may be inappropriate for the problem at hand. The ideal model to make more reliable decisions is to create a combination of outputs of many different hypotheses. Many machine-learning algorithms do this by learning an ensemble of hypothesis and employing them in a combined form. Bagging and boosting are the most frequently used among these schemes. The general methods may be applied to classification or categorization and regression problems. They frequently increase predictive performance over a single hypothesis.

By combining the decisions of various hypotheses, we amalgamate the different outputs into a single prediction. For classification problems, it is done through voting or weighted vote where asin

the case of regression problems, the average or weighted average is computed. We will be having the difference in parameter values in each ML algorithm is because of the fact that their training patterns differ and each one handles a certain percentage of data accuracy. In the bagging technique, individual approaches are constructed separately. But, in boosting technique, each new model is impacted by the performance of those built earlier. In boosting, we first make a model with accuracy on the training set greater than average. And then add new component classifiers to make an ensemble whose joint decision rule possesses a high level of accuracy on the training set. AdaBoost or Adaptive Boosting is a widely used algorithm for boosting. Ensemble structure is shown in Figure 1 for better understanding.

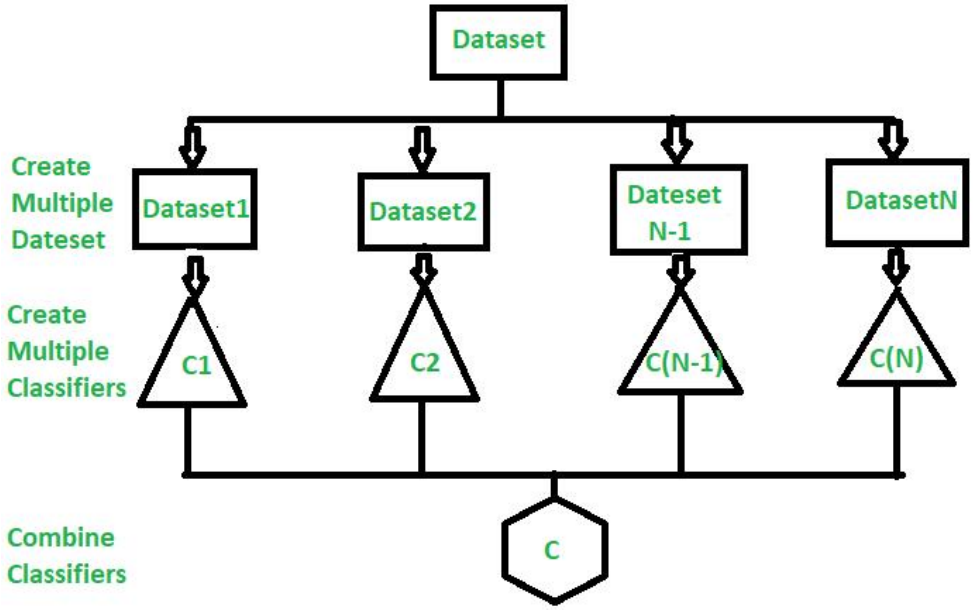


Figure 1 A Model of an Ensemble Classifier

11.2 Bagging

Basically, creating a different training subset from sample training data with replacement is called Bagging. The final output is based on majority voting.This is also known as Bootstrap Aggregation. The distribution of data is given in the Figure 2 below.

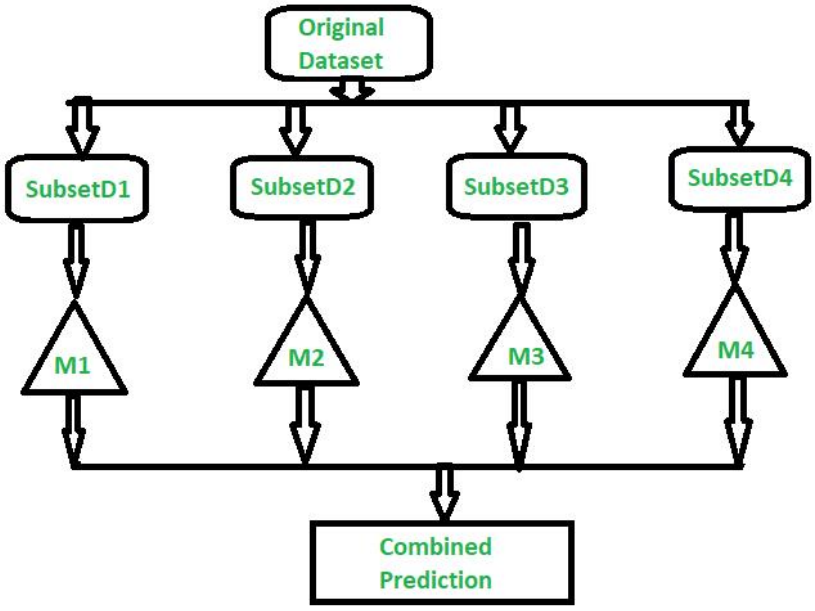


Figure 2 Ensemble Classifier with Bagging

Replacement means that if a row is selected, it is returned to the training dataset for potential reselection in the same training dataset. This means that a row of data may be selected zero, one, or multiple times for a given training dataset.

We can understand from the above figure that, the original dataset is divided into multiple small dataset and given to different machine learning algorithms for learning. The individual results are consolidated and final result will be received.

11.3 Boosting

Boosting is a method used to reduce the predictive errors in machine learning. Boosting actually tries to overcome this issue by training multiple machine learning models sequentially to improve the accuracy of the overall system. Performance of the machine learning model improves since the model is created using multiple weak learners into a single learning model. We can also say that "The final model is created by combing weak learners into strong learners by creating sequential models and it has the highest accuracy is called Boosting. Example: ADA BOOST, XG BOOST.

Let us discuss what is weak learners and strong learners. Weak learners have low prediction accuracy and it is similar to random guessing. They will have overfitting issues. So, they can't classify the data, which varies too much from the original dataset. If we train the model to identify the "Cat" as animal, which has pointed ears, but it will fail to recognize a "Cat" whose ears are curled. Strong learners have higher prediction accuracy. This converts a system of weak learners into a strong learning system. Let us consider an example, to identify the "Cat" image, it combines a weak learner that guesses for pointy ears and another learner that guesses for cat-shaped eyes. The overall accuracy of the system improves since the learners analyze all the properties individually.

Usually, the decision trees are used for boosting implementations. Decision trees are data structures in machine learning that works by dividing the dataset into smaller and smaller datasets based on their features. We know that bagging and boosting are the two common ensemble methods used to improve the prediction accuracy.But, the main difference lies in the method of training.The difference between the bagging and boosting is understood from the figure 3 given below.

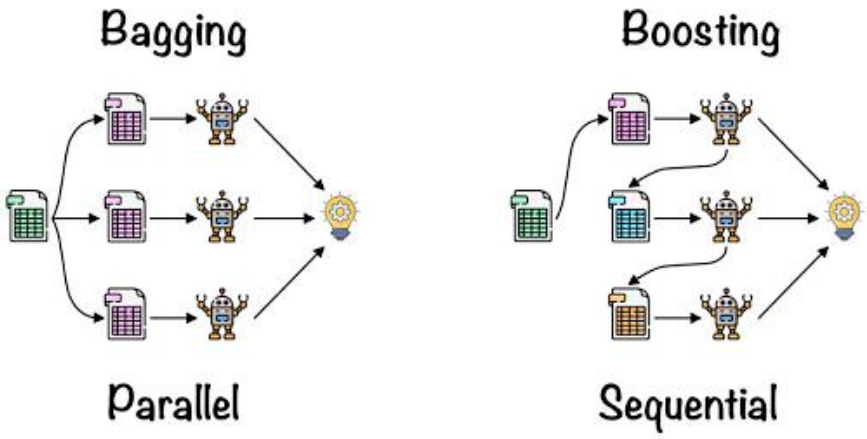


Figure 3 Comparison of Bagging and Boosting

The process of training used in boosting, can be understood from the simple steps given below.

Step 1

The boosting algorithm assigns equal weights to each data sample. Then it feeds the data to the first machine learning model known as base algorithm. The base algorithm makes the prediction for each data sample.

Step 2

The boosting algorithm assesses the model predictions and increases the weight of samples with a more significant error. It also assigns a weight based on model performance. A model that outputs excellent predictions will have a high amount of influence over the final decision.

Step 3

The algorithm passes the weighted data to the next decision tree.

Step 4

The algorithm repeats the steps from 2 to 3 until the instances of training errors are below a certain threshold.

Also, the boosting has few types of it. They are adaptive boosting, gradient boosting and extreme gradient boosting. Let us see what are theymean.

Adaptive boosting

It is one of the earliest boosting models. It adapts and tries to self-correct in every iteration of the boosting process. Initially adaptive boosting gives the same weights to each dataset. Then it automatically adjusts the weights of the data points after every decision tree. It gives more weights to incorrectly classified items to correct them for the next iteration. It repeats the process until the residual error or the difference between actual and predicted values, falls below an acceptable threshold. We can also use AdaBoost with many predictors and it is typically not as sensitive as other boosting algorithms. This approach does not work well when there is a correction among features or high data dimensionality. AdaBoost is a suitable type of boosting for classification problems.

Gradient Boosting

It is similar to AdaBoost but it has sequential training technique. The difference between the AdaBoost and gradient boosting is that gradient boosting does not give incorrectly classified items the more weight. Instead, gradient boosting optimizes the loss function by generating base learners sequentially so that the present base learner is always more effective than the previous one. This method attempts to generate accurate results initially instead of correcting errors throughout the process, like AdaBoost. Gradient boosting can help with both classification and regression based problems.

Extreme gradient boosting

This is having the more computational speed and scale in several ways over gradient boosting. Extreme boosting uses multiple cores on the CPU so that learning can occur in parallel during training. It is a boosting algorithm that can handle extensive datasets, making it attractive for big data applications. The key features of extreme gradient are parallelization, distributed computing, cache optimization and out-of-core processing.

Boosting has easy-to-understand and easy-to-interpret algorithms that learn from their mistakes. These algorithms don't require any data preprocessing, and they have built-in routines to handle missing data. In addition, most languages have built-in libraries to implement boosting algorithms with many parameters that can fine-tune performance.

Boosting models are vulnerable to outliers or data values that are different from the rest of the dataset. Because each model attempts to correct the faults of its predecessor, outliers can skew results significantly. Also, we might find it challenging to use boosting for real-time implementation because the algorithm is more complex than other processes. Boosting methods have high adaptability so you can use wide variety of model parameters that immediately affect the model's performance.

11.4 Random Forests

Random Forest is an extension over bagging. Random forest is a commonly used machine-learning algorithm trademarked by Leo Breiman and Adele Cutler, which combine the output of multiple decision trees to reach a single result. It is understood that random forest model is made up of multiple decision trees as shown in Figure 4.

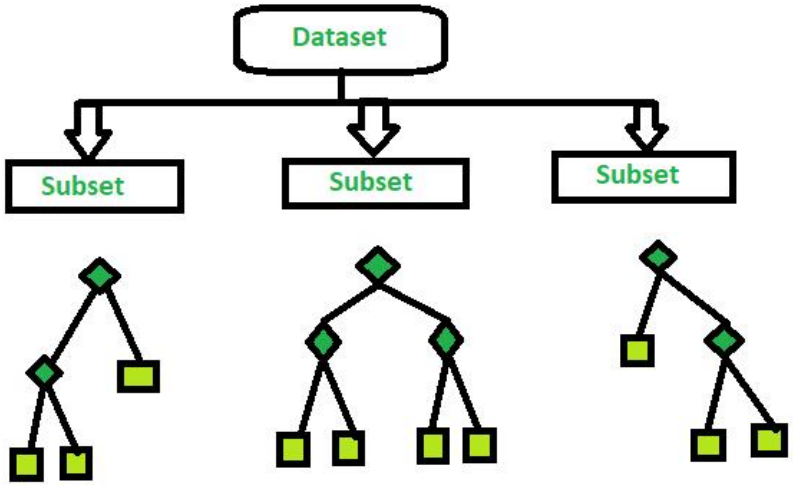


Figure 4 Structure of Random Forest

We have already discussed the making of decision trees in the Unit-7. Kindly recall it before preceding this. The following figure 5 will be used for recalling the decision tree concepts. Also, the metrics such as gini impurity, information gain and entropy can be used to evaluate the quality of the eachsplit.

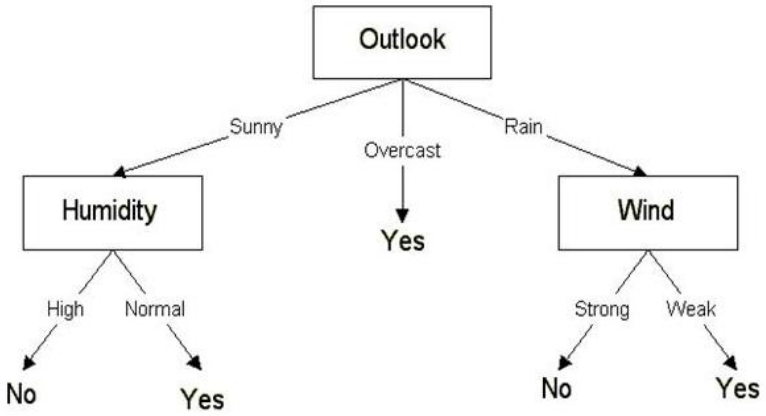


Figure 5 Structure of Decision Tree

Decision trees arealso having the bias and overfitting problems since it is also supervised learning. These problems are overcome by making multiple decision trees and form an ensemble of decision trees in the random forest algorithm. This ensemble is giving comparatively good results. Random forest algorithm has three main hyper parameters, which need to be set before training them. These include node size, number of trees and the number of features sampled. Then the random forest classifier may be used for either classification problems or regression problems.

The following are the simple steps for random forest:

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

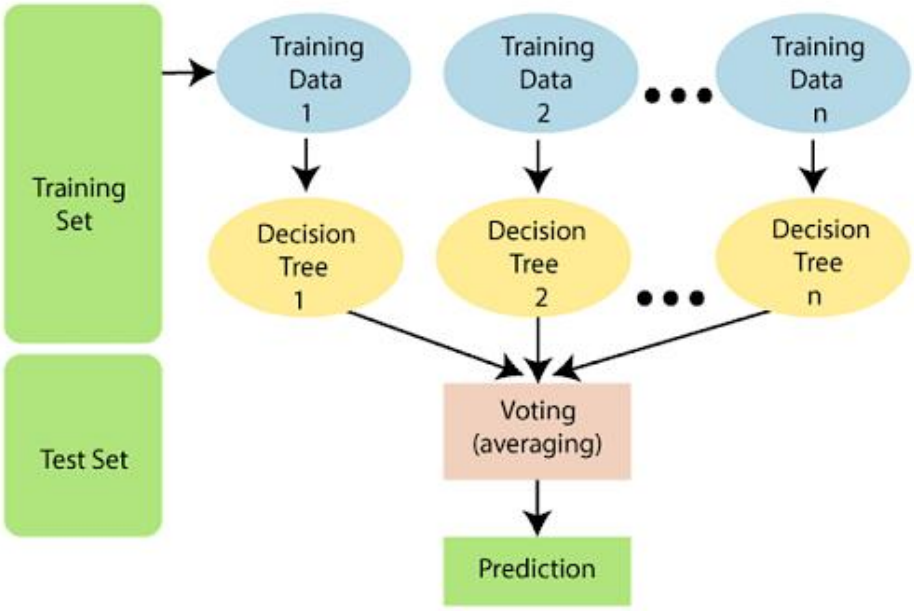


Figure 6Structure of Random Forest

Implementation of Random Forest Algorithm using Python Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

path = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
headernames = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']
dataset = pd.read_csv(path, names = headernames)
dataset.head()

X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30)

from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 50)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
result = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
```

```
print(result)
result1 = classification_report(y_test, y_pred)
print("Classification Report:",)
print (result1)
result2 = accuracy_score(y_test,y_pred)
print("Accuracy:",result2)
```

There are key benefits and challenges from random forest algorithm as follows.

Few Benefits:

- Reduced risk of overfitting:
  - Decision trees run the risk of overfitting, as they tend to tightly fit all the samples within training data. However, when there are a robust number of decision trees in a random forest, the classifier would not overfit the model since the averaging of uncorrelated trees lowers the overall variance and prediction error.
- Provides flexibility
  - Since random forest can handle both regression and classification tasks with a high degree of accuracy, it is a popular method among data scientists. Feature bagging also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing.
- Easy to determine feature importance:
  - Random forest makes it easy to evaluate variable importance, or contribution, to the model. There are a few ways to evaluate feature importance. Gini importance and mean decrease in impurity are usually used to measure how much the model's accuracy decreases when a given variable is excluded. However, permutation importance, also known as mean decrease accuracy is another importance measure. It also identifies the average decrease in accuracy by randomly permutating the feature values in out of bag samples.

Few Challenges:

- Time consuming process: Since random forest algorithms can handle large data sets, they can be providing more accurate predictions, but can be slow to process data as they are computing data for each individual decision tree.
- Requires more resources: Since random forests process larger data sets, they will require more resources to store the data.
- Complexity: The prediction of a single decision tree is easier to interpret when compared to a forest of them.

Summary

This unit discussed about ensemble learning methods that were made up of a set of classifiers. All the output of each classifier were taken and were aggregated to produce the final result. The most well-known ensemble methods were bagging or bootstrap aggregation and boosting. These methods were commonly used to reduce the variance within a noisy dataset. They were explained in detail with necessary examples. The types of boosting were also highlighted in this unit. Finally, the concepts of random forest were explained with the working procedure. Also, we understood the difference between the random forest and decision trees. These units emphasized the importance of ensemble learning over machine learning algorithms in all the possible ways.

Keywords

- Bagging
- Random Forest
- Decision Tree
- Boosting

Self Assessment

1. What is true about an ensemble classifier? 1. Classifiers that are more "sure" can vote with more conviction. 2. Classifiers can be more "sure" about a particular part of the space. 3. Most of the times, it performs better than a single classifier.

A. 1 and 2
B. 1 and 3
C. 2 and 3
D. All of the above

2. Bootstrap and Aggregation, commonly known as_____

A. Information Gain
B. Bagging
C. Entropy
D. none of these

3. Random Forest has _____ as base learning models

A. Multiple decision trees
B. Bagging
C. Entropy
D. None of these

4. _____helps improve machine learning results by combining several models.

A. Machine Learning
B. bagging
C. Entropy
D. Ensemble learning

5. In _____the output class is the prediction based on the average of probability given to that class.

A. Hard voting
B. Soft voting
C. Both A and B
D. None of these

6. In voting classifier which of the following does not exist?

A. Hard voting
B. Soft voting
C. Both A and B
D. None of these

7. Decision tree is the most powerful for _____
A. Classification
B. Prediction
C. Both a and b
D. None of these

8. _____is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples.
A. Information Gain
B. Entropy
C. Gini Index
D. none of these

9. Decision-tree algorithm falls under the category of _____.
A. Unsupervised learning algorithms
B. Reinforcement learning algorithm
C. Supervised learning algorithms
D. Prone to errors in classification problems with many class

10. Decision trees can handle_____.
A. High dimensional data
B. Low dimensional data
C. Medium dimensional data
D. None of these

11. In random forest or gradient boosting algorithms, features can be of any type. For example, it can be a continuous feature or a categorical feature. Which of the following option is true when you consider these types of features?
A. Only Random forest algorithm handles real valued attributes by discretizing them
B. Only Gradient boosting algorithm handles real valued attributes by discretizing them
C. Both algorithms can handle real valued attributes by discretizing them
D. None of these

12. Suppose you are using a bagging based algorithm say a Random Forest in model building. Which of the following can be true?
    1. Number of tree should be as large as possible.
    2. You will have interpretability after using Random Forest
A. 1
B. 2
C. 1 and 2
D. None of these

13. Which of the following algorithm doesn't uses learning Rate as of one of its hyperparameter?

    1. Gradient Boosting, 2. Extra Trees, 3. AdaBoost and 4. Random Forest.

A. 1 and 3
B. 1 and 4
C. 2 and 3
D. 2 and 4

14. When you use the boosting algorithm you always consider the weak learners. Which of the following is the main reason for having weak learners?
    1. To prevent overfitting
    2. To prevent under fitting

A. 1
B. 2
C. 1 and 2
D. None of these

15. Which of the following is true about the Gradient Boosting trees?
    1. In each stage, introduce a new regression tree to compensate the shortcomings of existing model
    2. We can use gradient decent method for minimize the loss function

A. 1
B. 2
C. 1 and 2
D. None of these

Answers for SelfAssessment

1.	D	2.	B	3	A	4.	D	5.	B
6.	D	7.	C	8.	B	9.	C	10.	A
11.	C	12.	A	13.	D	14.	A	15.	C

Review Questions

1. Explain the architecture of Random Forest.
2. List the various types of Boosting.
3. Give thepython library functions used to implement ensemble learning?
4. Differenciate weak learner and strong learner.
5. How the final decision is taken in bagging and boosting methods?

Further Readings

- MadanGopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.

- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm
- https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm
- https://www.ibm.com/topics/random-forest
- https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/

Unit 12: Data Visualization

CONTENTS

Objectives

Introduction

12.1 K Means Algorithm

12.2 Applications

12.3 Hierarchical Clustering

12.4 Hierarchical Clustering Algorithms

12.5 What is Ensemble Learning

12.6 Ensemble Techniques

12.7 Maximum Voting

12.8 Averaging

12.9 Weighted Average

Summary

Keywords

Self Assessment

Answers for Self Assessment

Review Questions

Further Readings

Objectives

After this unit, student would be able to

- Understand basic concepts of seaborn
- Learn basic difference between seaborn and matplotlib
- Know how data visualization perform using seaborn

Introduction

One of the most popular methods for gaining a general understanding of the data's structure is clustering. It can be summed up as the process of finding data subgroups where data points in the same subgroup (cluster) are extremely similar and other data points in other clusters are very dissimilar. To put it another way, we look for homogeneous subgroups within the data so that the data points in each cluster are as comparable as feasible based on a similarity metric like the Euclidean-based distance or the correlation-based distance. Choosing the similarity metric to employ depends on the application.

Clustering analysis can be carried out either on the basis of samples or on the basis of characteristics, where we attempt to identify subgroups of samples based on features. Here, we'll talk about feature-based clustering. We utilize clustering in picture segmentation/compression to group comparable regions together, market segmentation to discover clients that are similar to one another in terms of behaviours or traits, document clustering based on subjects, etc.

Because we lack the ground truth to compare the output of the clustering algorithm to the true labels in order to assess its effectiveness, clustering is regarded as an unsupervised learning method

in contrast to supervised learning. By dividing the data points into discrete subgroups, we simply wish to try to study the data's structure.

Only Kmeans, one of the most popular clustering algorithms because of its simplicity, will be covered in this chapter.

12.1 K Means Algorithm

The iterative Kmeans algorithm attempts to divide the dataset into K unique, non-overlapping subgroups (clusters), each of which contains a single data point. While keeping the clusters as distinct (far) apart as possible, it aims to make the intra-cluster data points as comparable as possible. It distributes data points to clusters in a way that minimises the sum of the squared distances between the data points and the cluster centroid, which is the average value of all the data points in the cluster. The homogeneity (similarity) of the data points within a cluster increases as the amount of variance within the cluster decreases.

The following is how the kmeans algorithm functions:

- Specify number of clusters to be K.
- Set up the centroids by randomly choosing K data points, without replacement, and then shuffling the dataset.
- Up till the centroids don't change, keep iterating. i.e., the clustering of data points remains constant.
- Calculate the sum of the squared distances between all of the centroids and the data points.
- Assign each data point to the centroid of the nearest cluster.
- By averaging all of the data points that make up each cluster, get the centroids for the clusters.

Expectation-Maximization is the method that kmeans uses to tackle the issue. The data points are assigned to the closest cluster in the E-step. The centroid of each cluster is calculated in the M-step. Here is a breakdown of the mathematical steps we can take to solve it.

The objective function is:

$$J - \sum_{i-1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2 \qquad (1)$$

where wik=1 for data point xi if it belongs to cluster *k*; otherwise, wik=0. Also, μk is the centroid of xi's cluster.

It is a two-part minimization issue. First, we fix k and minimize J w.r.t.wik. Then we treat wik fixed and minimize J w.r.t. Technically speaking, we update cluster assignments (E-step) after differentiating J w.r.t.wik. After recalculating the centroids based on the cluster assignments from the previous step (M-step), we distinguish J w.r.t. k. Consequently, E-step is:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \|x^i - \mu_k\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k - argmin_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

To put it another way, choose the cluster to which the data point xi belongs based on its sum of squared distances to the cluster's centroid.

And M-step is:

$$\frac{\partial J}{\partial \mu_k} \quad 2 \sum_{i=1}^{m} w_{ik}(x^i - \mu_k) \quad 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}} \qquad (3)$$

Which equates to recalculating each cluster's centroid to account for the new assignments.

Here are a few things to consider:

Since almost always the features in any dataset would have different units of measurements, such as age vs income, it is advised to standardise the data to have a mean of zero and a standard deviation of one. This is because clustering algorithms, including kmeans, use distance-based measurements to determine the similarity between data points.

Different initializations may result in different clusters since the kmeans method may get trapped in a local optimum and not converge to a global optimum due to its iterative nature and the random initialization of centroids at the beginning of the algorithm. Therefore, it is advised to execute the method with several centroids' initializations and select the results of the run that produced the smallest sum of squared distance.

The same holds true for the assignment of examples as it does for the within-cluster variation:

$$\frac{1}{m_k}\sum_{i=1}^{m_k}\|x^i - \mu_{c^k}\|^2 \qquad (4)$$

Implementation

Here, we'll utilize a straightforward application of kmeans to just show a few ideas. The more effective sklearn implementation will then take care of a lot of things for us.

```
import numpy as np
from numpy.linalg import norm


class Kmeans:
    '''Implementing Kmeans algorithm.'''


    def __init__(self, n_clusters, max_iter=100, random_state=123):
self.n_clusters = n_clusters
self.max_iter = max_iter
self.random_state = random_state


    def initializ_centroids(self, X):
np.random.RandomState(self.random_state)
random_idx = np.random.permutation(X.shape[0])
        centroids = X[random_idx[:self.n_clusters]]
        return centroids


    def compute_centroids(self, X, labels):
        centroids = np.zeros((self.n_clusters, X.shape[1]))
for k in range(self.n_clusters):
centroids[k, :] = np.mean(X[labels == k, :], axis=0)
        return centroids


    def compute_distance(self, X, centroids):
        distance = np.zeros((X.shape[0], self.n_clusters))
```

```
for k in range(self.n_clusters):
row_norm = norm(X - centroids[k, :], axis=1)
            distance[:, k] = np.square(row_norm)
        return distance

    def find_closest_cluster(self, distance):
        return np.argmin(distance, axis=1)

    def compute_sse(self, X, labels, centroids):
        distance = np.zeros(X.shape[0])
for k in range(self.n_clusters):
distance[labels == k] = norm(X[labels == k] - centroids[k], axis=1)
        return np.sum(np.square(distance))

    def fit(self, X):
self.centroids = self.initializ_centroids(X)
        for i in range(self.max_iter):
old_centroids = self.centroids
            distance = self.compute_distance(X, old_centroids)
self.labels = self.find_closest_cluster(distance)
self.centroids = self.compute_centroids(X, self.labels)
            if np.all(old_centroids == self.centroids):
                break
self.error = self.compute_sse(X, self.labels, self.centroids)

    def predict(self, X):
        distance = self.compute_distance(X, self.centroids)
        return self.find_closest_cluster(distance)
```

12.2 Applications

In many different applications, including market segmentation, document clustering, image segmentation, and image compression, the kmeans technique is particularly well-liked and widely employed. Usually, our aim when performing a cluster analysis is one of the following:

1. Obtain a meaningful understanding of the data's structure before proceeding.
2. If we think there is a large variance in the behaviors of distinct subgroups, we will cluster-then-predict, where different models will be generated for different subgroups. Clustering patients into several subgroups and developing a model for each subgroup to forecast the likelihood of suffering a heart attack are two examples of that.

12.3 Hierarchical Clustering

Hierarchical clustering is an alternative approach to *k*-means clustering for identifying groups in a data set. In contrast to *k*-means, hierarchical clustering will create a hierarchy of clusters and therefore does not require us to pre-specify the number of clusters. Furthermore,

hierarchical clustering has an added advantage over *k*-means clustering in that its results can be easily visualized using an attractive tree-based representation called a *dendrogram*.

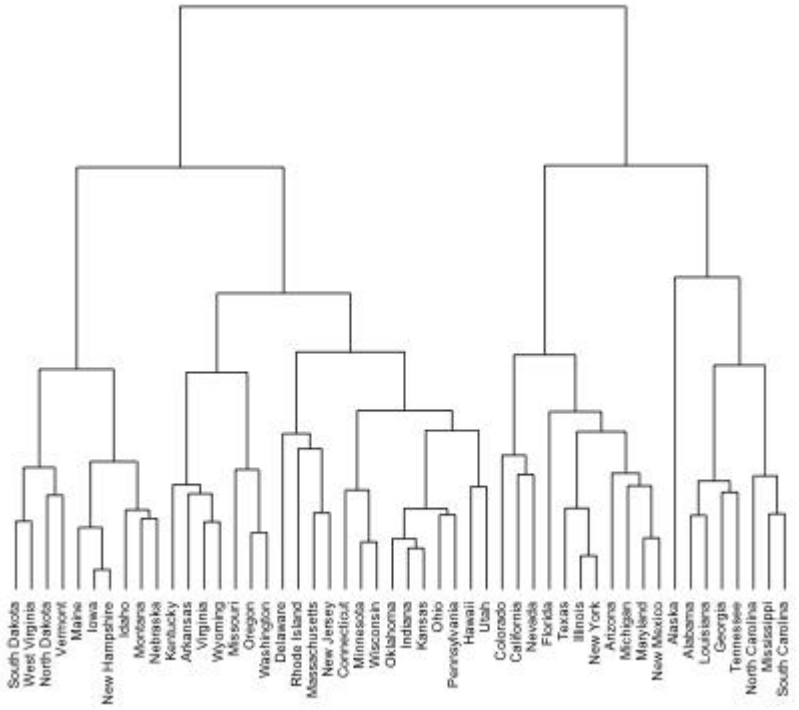


Figure 1 Hierarchical Clustering

Prerequisites

```
# Helper packages
library(dplyr)       # for data manipulation
library(ggplot2)     # for data visualization
# Modeling packages
library(cluster)     # for general clustering algorithms
library(factoextra)  # for visualizing cluster results
```

The Ames housing data will be used to explain the main ideas of hierarchical clustering. For the sake of simplicity, we'll only utilize the 34 numeric features, but if you'd like to reproduce this study using the entire set of features, please see our discussion in Section 20.7. We first standardise the data because these features are measured at dramatically different magnitudes:

```
ames_scale<- AmesHousing::make_ames() %>%
select_if(is.numeric) %>%  # select numeric columns
  select(-Sale_Price) %>%    # remove target column
mutate_all(as.double) %>%  # coerce to double type
scale()                    # center & scale the resulting columns
```

12.4 Hierarchical Clustering Algorithms

Two major types of hierarchical clustering can be distinguished:

Agglomerative clustering, also known as AGNES (Agglomerative Nesting), operates from the bottom up. In other words, each observation is first viewed as a leaf cluster with a single element. The two clusters that are most similar to one another are joined into new, larger clusters (referred to as nodes) at each stage of the process. This process is repeated until every single point is a part of the same large cluster (root). The outcome is a tree that can be shown on a dendrogram.

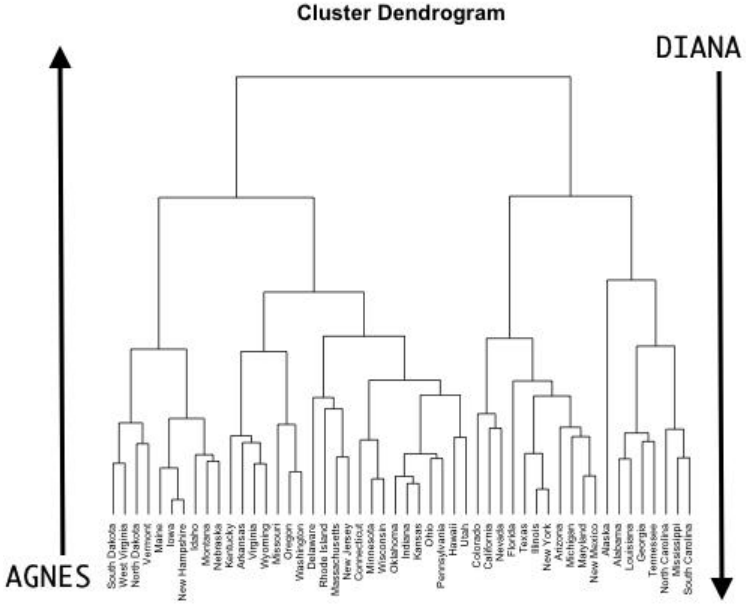


Figure 2Agglomerative Clustering

Divisive hierarchical clustering:

Commonly known as DIANA (DIviseANAlysis), this method operates top-down. The opposite of Agnes is DIANA. It starts with the root, where every observation is a part of a single cluster. The current cluster is divided into the two clusters that are thought to be the most diverse at each phase of the procedure. Up till all observations are in their own cluster, the process is iterated.

You should be aware that agglomerative clustering is effective in locating small groups. On the other side, divisive hierarchical clustering is more effective at locating substantial clusters.

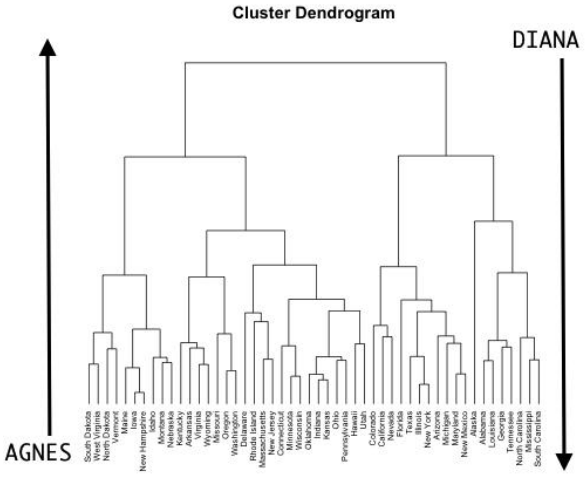


Figure 3 Cluster Dendogram

The most common methods are:

Maximum or complete linkage clustering: The biggest value of the pairwise dissimilarities between elements in clusters 1 and 2 is taken into account as the distance between the two clusters

when using maximum or complete linkage clustering. It usually results in more tightly packed clusters.

Minimum or single linkage clustering: Calculates all pairwise differences between items in clusters 1 and 2, then takes the difference with the least value as the linkage criterion. It frequently results in lengthy, "loose" clusters.

Mean or average linkage clustering: The distance between the two clusters is calculated by adding up all pairwise differences between the items in clusters 1 and 2, then taking the average of those differences into account. The clusters it produces can differ in how compact they are.

Centroid linkage clustering:calculates the difference between the centroid for cluster 1 (a mean vector with p elements, one for each variable), and the centroid for cluster 2.

Ward's minimum variance method: Reduces all within-cluster variance to a minimum. The cluster pairs with the shortest between-cluster distance are combined at each phase. tends to generate clusters that are denser.

When conducting a hierarchical cluster analysis, there are several agglomeration methods that can be used to define clusters; however, complete linkage and Ward's method are frequently favoured for AGNES clustering. The maximum average dissimilarity is used by DIANA to separate clusters, which is very similar to the mean or average linkage clustering method described before.

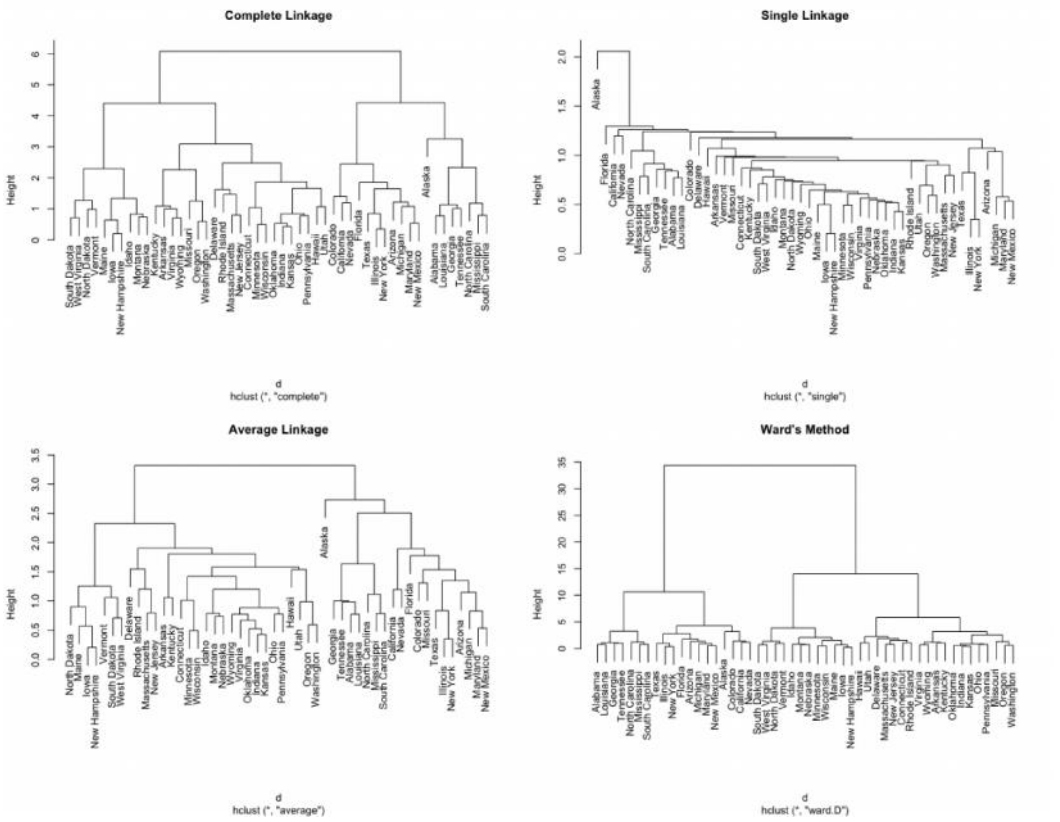


Figure 4Differing hierarchical clustering outputs based on similarity measures.

12.5 What is Ensemble Learning

By combining predictions from various models, the machine learning technique known as ensemble learning improves forecasting accuracy and robustness. By utilising the collective intelligence of the ensemble, it seeks to reduce any inaccuracies or biases that may be present in individual models.

Combining the results of various models to produce a more accurate forecast is the basic idea behind ensemble learning. Ensemble learning boosts the effectiveness of the learning system as a

whole by taking into account many viewpoints and making use of the advantages of several models. This method not only improves accuracy but also offers resistance to data uncertainty. Ensemble learning has proven to be a strong tool in many domains, providing more robust and reliable forecasts by successfully combining predictions from numerous models.

12.6 Ensemble Techniques

In this section, we'll examine a few straightforward yet effective strategies, specifically:

- Maximum Voting
- Average
- By weighing each item

12.7 Maximum Voting

For categorization issues, the max voting approach is typically utilised. With this method, predictions are made for each data point using a variety of models. Each model's predictions are regarded as a "vote." The majority of the models' forecasts serve as the basis for the final projection.

For instance, if you asked five of your coworkers to give your movie a rating (out of 5), we'll suppose that three of them gave it a 4, and two gave it a 5. Since the majority of respondents rated it a 4, that will be the final rating. This can be viewed as taking the average of all predictions.

The result of max voting would be something like this:

Colleague 1	Colleague 1	Colleague 1	Colleague 1	Colleague 1	Final Rating
5	4	5	4	4	4

Here, the goal variable for training data is y_train, and the independent variables in the training data are called x_train. The independent variables x_test and the target variable y_test make up the validation set.

Alternatively, you can use "VotingClassifier" module in *sklearn* as follows:

```
from sklearn.ensemble import VotingClassifier

model1 = LogisticRegression(random_state=1)

model2 = tree.DecisionTreeClassifier(random_state=1)

model = VotingClassifier(estimators=[('lr', model1), ('dt', model2)], voting='hard')

model.fit(x_train,y_train)

model.score(x_test,y_test)
```

12.8 Averaging

Multiple forecasts are made for each data point when averaging, similar to the max voting method. In this approach, the final prediction is made by averaging the results of all the models. When computing probabilities for classification problems or making predictions in regression problems, averaging can be applied.

For example, in the below case, the averaging method would take the average of all the values.i.e. (5+4+5+4+4)/5 = 4.4

Colleague 1	Colleague 1	Colleague 1	Colleague 1	Colleague 1	Final Rating
5	4	5	4	4	4.4

Sample Code

```
model1 = tree.DecisionTreeClassifier()
model2 = KNeighborsClassifier()
model3= LogisticRegression()

model1.fit(x_train,y_train)
model2.fit(x_train,y_train)
model3.fit(x_train,y_train)

pred1=model1.predict_proba(x_test)
pred2=model2.predict_proba(x_test)
pred3=model3.predict_proba(x_test)

finalpred=(pred1+pred2+pred3)/3
```

12.9 Weighted Average

This is a development of the averaging approach. Different weights are assigned to each model, indicating the significance of each model for prediction. For instance, if two of your coworkers are critics but the rest lack experience in this area, the responses from these two buddies will be given more weight than those of the other participants.

The result is calculated as [(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = 4.41.

Colleague1	Colleague2	Colleague3	Colleague4	Colleague5	Final rating	
weight	0.23	0.23	0.18	0.18	0.18	
rating	5	4	5	4	4	4.41

Sample Code:

```
model1 = tree.DecisionTreeClassifier()
model2 = KNeighborsClassifier()
model3= LogisticRegression()

model1.fit(x_train,y_train)
model2.fit(x_train,y_train)
model3.fit(x_train,y_train)

pred1=model1.predict_proba(x_test)
pred2=model2.predict_proba(x_test)
pred3=model3.predict_proba(x_test)

finalpred=(pred1*0.3+pred2*0.3+pred3*0.4)
```

Summary

- In summary, the end of chapter for the topics of k-means and hierarchical clustering involves a comprehensive review of the key concepts techniques covered in the chapters. This includes

an understanding of how k-means is used for unsupervised, as well as the approaches for hierarchical clustering such as dendrograms and agglomerative clustering.

- k-Means algorithm a partitioning method that divides the dataset into k-overlapping clusters, where each point belongs to only one cluster. The algorithm is iteratively until the optimal centroid positions are found.
- Hierarchical clustering, on the other hand, creates a hierarchy of clusters using either the agglomer or divisive approach.
- Agglomerative clustering, each data point starts as its own cluster, and then clusters are merged until one cluster remains.
- Divisive clustering begins with the whole data set as one cluster, which is recursively subdivided into smaller clusters Both techniques have different applications and advantages, and the choice of algorithm depends on the nature of the data and the problem at hand.
- Clustering refers to the process of grouping similar objects or data points together based on their similarities or differences. It is a technique in data mining and machine learning to identify patterns in large datasets.
- A dendrogram is a tree-like diagram that represents the hierarchy of clusters produced by hierarchical clustering. It is a useful tool forizing the relationships between clusters and identifying potential subgroups within the data.
- K-Means Clustering is a popular unsupervised clustering algorithm that aims to partition given dataset into k distinct,-overlapping clusters
- K-means clustering algorithm is the most popular unsupervised learning technique utilized for clustering analysis. It has been widely used across industries such as agriculture, healthcare marketing, and more due to its simplicity and efficiency.
- A dendrogram is a tree-like diagram that represents the hierarchy of clusters produced by hierarchical clustering. It is a useful tool forizing the relationships between clusters and identifying potential subgroups within the data. The calculation of Euclidean distance plays a crucial role in data analysis, particularly in clustering and classification tasks. It is a fundamental in distance-based algorithms such as-means and hierarchical clustering.

Keywords

- **k-Means Clustering** is a popular partition-based algorithm that groups data points into 'k' clusters by minimizing the sum of squared distances between data points and their cluster centroids. It is widely used for tasks like customer segmentation and image compression.
- The **Average Method**, also known as Mean Method, is a linkage criterion used in Hierarchical Clustering. It calculates the distance between two clusters based on the average distance of all data point pairs from each cluster, resulting in a balanced approach to cluster merging.
- **Clustering with Weights** is a technique that assigns varying importance to individual data points during the clustering process. By incorporating weights, the algorithm considers certain points more influential than others, leading to more nuanced and context-aware clustering outcomes.
- **Comparative Analysis** involves evaluating the performance, strengths, and weaknesses of different clustering algorithms, such as k-Means, Hierarchical Clustering, DBSCAN, and others. This analysis helps researchers and practitioners choose the most suitable method for specific datasets and applications.
- **Clustering** finds applications in diverse fields, including marketing, biology, finance, and image analysis. For instance, in marketing, clustering aids in customer segmentation for targeted marketing strategies, while in biology, it assists in classifying genes based on expression patterns.
- **Hierarchical Clustering** is a powerful unsupervised learning technique that groups data into a tree-like hierarchy of clusters. It iteratively merges or divides clusters based on proximity, creating a dendrogram that visually represents the cluster relationships.

- **Bagging**, short for Bootstrap Aggregating, is an ensemble technique that builds multiple models independently and combines their predictions through voting or averaging. By training each model on bootstrapped subsets of the data, bagging reduces variance and enhances model robustness, making it popular for improving decision tree-based models like Random Forest.
- Boosting is an ensemble approach that focuses on sequentially improving the performance of weak learners by giving more weight to misclassified instances. Algorithms like AdaBoost and Gradient Boosting Machines (GBM) are commonly used in boosting, creating a strong learner from multiple weak learners and achieving high accuracy on challenging tasks.
- Stacking, also known as Stacked Generalization, combines predictions from diverse base models using a meta-model. It leverages the diverse strengths of individual models to make more accurate predictions and can be applied to various machine learning tasks, enabling effective model combination and performance enhancement.
- Voting ensembles combine the predictions of multiple models by either majority voting or weighted voting. This ensemble approach is simple yet effective, and it can be employed with a variety of machine learning algorithms, including classifiers and regression models, to produce more confident and reliable predictions.
- Ensemble pruning and trimming involve reducing the size of an ensemble by removing weak or redundant models. This process aims to improve efficiency, reduce memory requirements, and prevent overfitting, ensuring that the ensemble maintains a good balance between accuracy and complexity.
- Evaluating ensemble models requires specific metrics like ensemble accuracy, area under the ROC curve (AUC-ROC), or F1 score. Cross-validation and bootstrapping are employed to assess ensemble performance and provide unbiased estimates of model effectiveness.
- Ensemble learning in the context of deep learning involves combining multiple neural networks or deep learning architectures to improve predictive performance and generalization. Techniques like model averaging, stacking, and bagging can be applied to neural networks, leveraging the power of ensembles in complex tasks and boosting overall performance.

SelfAssessment

1. Which of the following is the primary objective of k-Means clustering?

A. Maximizing inter-cluster variance
B. Minimizing intra-cluster variance
C. Minimizing inter-cluster variance
D. Maximizing intra-cluster variance

2. The k-Means algorithm is susceptible to:

A. Overfitting
B. Underfitting
C. Initial centroid sensitivity
D. Outliers

3. The value of k in k-Means clustering represents:

A. The number of clusters desired.
B. The maximum number of iterations.
C. The learning rates.
D. The number of features in the dataset.

4. Which of the following methods is used to calculate the distance between two clusters in Hierarchical Clustering?

A. Single Linkage
B. Average Linkage
C. Complete Linkage
D. All of the above

5. Hierarchical Clustering always results in:

A. Balanced binary trees
B. Unbalanced binary trees
C. Balanced ternary trees
D. Unbalanced ternary trees

6. The main purpose of ensemble learning is to:

A. Reduce the complexity of models
B. Combine multiple models to improve performance
C. Increase the interpretability of models
D. Enhance the training speed of models

7. Which ensemble method focuses on sequentially improving the performance of weak learners?

A. Bagging
B. Boosting
C. Stacking
D. Voting

8. Ensemble pruning and trimming involve:

A. Increasing the size of the ensemble for better performance
B. Reducing the diversity of models in the ensemble
C. Removing weak or redundant models from the ensemble
D. Eliminating all models except the best one

9. What is the main objective of clustering in machine learning?

A. Maximizing inter-cluster variance
B. Minimizing intra-cluster variance
C. Maximizing intra-cluster variance
D. Minimizing inter-cluster variance

10. Which of the following clustering algorithms is a partition-based method?

A. K-Means
B. DBSCAN

C. Agglomerative Hierarchical Clustering
D. Mean Shift

11. In K-Means clustering, how are initial cluster centroids typically chosen?

A. Randomly from the data points
B. Centroids of the first few clusters
C. Centroids of the last few clusters
D. Manually by the user

12. Which ensemble clustering technique assigns data points to clusters based on majority voting or averaging of cluster assignments from different clustering algorithms?

A. Bagging
B. Boosting
C. Stacking
D. Voting

13. In ensemble clustering, diversity among individual clustering algorithms is essential to:

A. Reduce computational cost
B. Avoid overfitting
C. Improve clustering accuracy
D. Speed up convergence

14. Clustering evaluation metrics, such as Silhouette Score and Davies-Bouldin Index, are used to assess:

A. The interpretability of clusters
B. The computational complexity of clustering algorithms
C. The quality and coherence of clustering results
D. The number of clusters required for optimal results

15. Hierarchical Clustering is a technique used for:

A. Data classification
B. Data regression
C. Data visualization
D. Data clustering

Answers for Self Assessment

1.	B	2.	C	3.	A	4.	D	5.	B
6.	B	7.	B	8.	C	9.	D	10.	A
11.	A	12.	D	13.	C	14.	C	15.	D

Review Questions

1. Explain the k-Means algorithm in detail, including its steps and convergence criteria. Discuss the impact of the initial centroids' selection on the clustering results.
2. Compare and contrast k-Means clustering and Hierarchical clustering in terms of their working principles, advantages, and limitations. Provide real-world examples where each algorithm would be suitable.
3. Illustrate the process of hierarchical clustering using a dendrogram. Explain how different linkage methods (Single, Complete, and Average) influence the clustering results.
4. Discuss the concept of ensemble learning and its significance in improving predictive performance. Explain two popular ensemble techniques and their applications in clustering tasks.
5. Evaluate the effectiveness of ensemble pruning and trimming methods in reducing the complexity of an ensemble while maintaining performance. Provide examples and discuss the trade-offs in ensemble size reduction.
6. Explain how ensemble-based methods can address the limitations of k-Means clustering. Provide a step-by-step guide on how to build an ensemble of k-Means models to improve clustering accuracy and stability.
7. Discuss the role of diversity in ensemble learning and its impact on ensemble performance. Describe three strategies to induce diversity among individual models within an ensemble.
8. Compare the performance of k-Means clustering and hierarchical clustering on a given dataset. Use appropriate evaluation metrics to measure the clustering quality, and analyze the strengths and weaknesses of each algorithm's results.
9. Examine the challenges of using ensemble learning in deep learning models. Discuss how ensembling can mitigate common issues like overfitting and improve the robustness of deep learning predictions.
10. Analyze a real-world clustering problem and propose an ensemble-based solution. Describe the choice of base clustering algorithms, the method of combining their results, and the justification for using ensemble learning in this specific scenario.

Further Readings

- MadanGopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://www.geeksforgeeks.org/difference-between-k-means-and-hierarchical-clustering/
- https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a
- https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-

ensemble-models/

Unit 13: Neural Networks

CONTENTS

Objectives
Introduction
13.1 Biological Structure of a Neuron
13.2 Artificial Neuron and its Structure
13.3 Perceptron
13.4 Multi-layer Networks
13.5 Introduction to Deep Neural Networks (DNN)
13.6 Evaluation Metrics of Machine Learning Models
Summary
Keywords
Self Assessment
Answers for Self Assessment
Review Questions
Further readings

Objectives

- Understanding the processing of a neuron.
- Understanding the Structure of Artificial Neural Networks.
- Understanding the structure of a perceptron model and multilayer perceptron model.
- Understanding the various evaluation metrics of ML models.
- Understanding the basic concepts of deep neural networks.

Introduction

Artificial Neural Networks (ANNs) are relatively crude electronic models based on the neural structure of the brain. The brain learns from experience and stores in the cells as in Figure 1. Artificial neural networks try to mimic the functioning of brain.

Figure 1 Human Brain and Cells

Even simple animal brains are capable of functions that are currently impossible for computers. Computers do the things well, but they have trouble recognizing even simple patterns. This unit explores the technical aspects and makes you to understand the processing of neuron too along with ANN Architecture.

13.1 Biological Structure of a Neuron

Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result. Figure2 shows the relationship of these four parts.

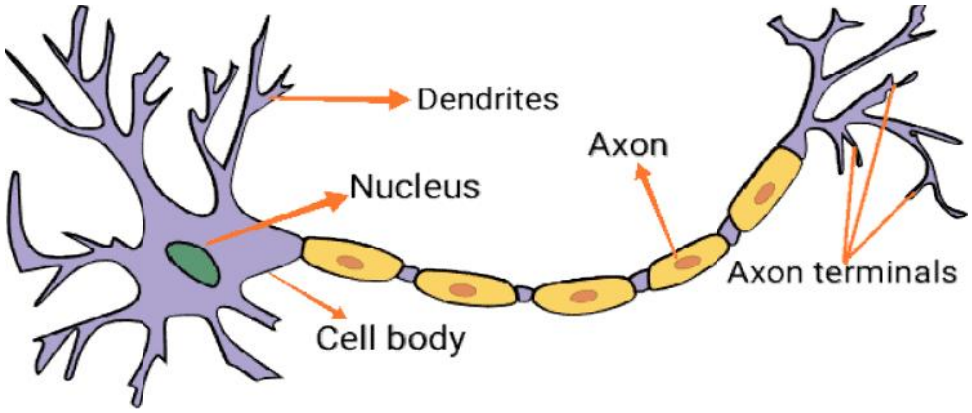


Figure 2Structure of Biological Neuron

Within humans there are many variations on basic type of neuron, yet, all biological neurons have the same four basic components. They are known by their biological names – cell body (soma), dendrites, axon, and synapses.

Cell body (Soma)

The body of neuron cell contains the nucleus and carries out biochemical transformation necessary to the life of neurons.

Dendrite

Each neuron has fine, hair like tubular structures (extensions) around it. They branch out into tree around the cell body. They accept incoming signals.

Axon

It is a long, thin, tubular structure which works like a transmission line. Synapse: Neurons are connected to one another in complex spatial arrangement. When axon reaches its final destination it branches again called as terminal arborization. At the end of axon are highly complex and specialized structures called synapses. Connection between two neurons takes place at these synapses.

Dendrites receive the input through the synapses of other neurons. The soma processes these incoming signals over time and converts that processed value into an output, which is sent out to other neurons through the axon and the synapses.

13.2 Artificial Neuron and its Structure

An artificial neuron is a mathematical function conceived as a simple model of a real (biological) neuron. The artificial neuron simulates four basic functions of a biological neuron. Figure 3 shows basic representation of an artificial neuron.

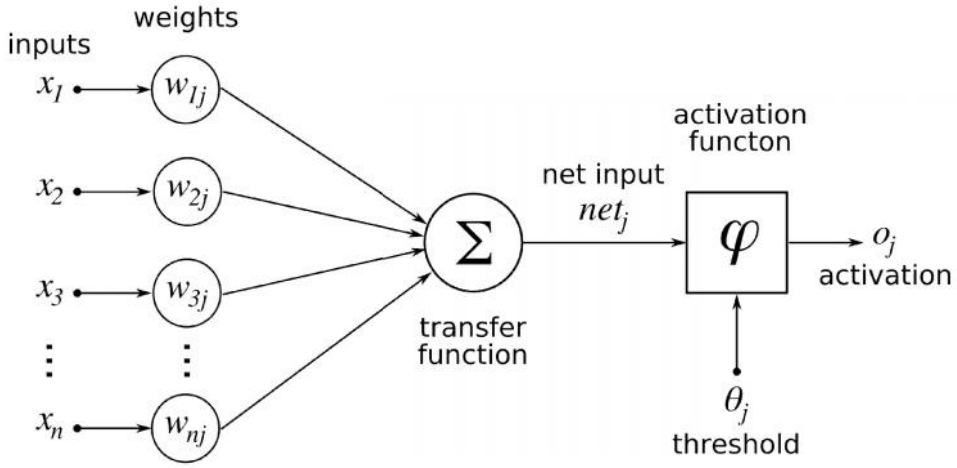


Figure 3 Structure of Artificial Neuron

In Figure 3, various inputs to the network are represented by the mathematical symbol, x(n). Each of these inputs is multiplied by a connection weight. The weights are represented by w(n). In the simplest case, these products are summed, fed to a transfer function (activation function) to generate a result, and this result is sent as output. This is also possible with other network structures, which utilize different summing functions as well as different transfer functions. Some summation functions have an additional activation function' applied to the result before it is passed on to the transfer function for the purpose of allowing the summation output to vary with respect to time.

An Activation Function decides whether a neuron should be activated or not. This means that it will decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations.The role of the Activation Function is to derive output from a set of input values fed to a node (or a layer).

There are three types of activation functions available. They are, binary step activation function, linear activation function and non-linear activation function.

Binary step activation function

Binary step activation function as in Figure 4 depends on a threshold value that decides whether a neuron should be activated or not. The input fed to the activation function is compared to a certain threshold; if the input is greater than it, then the neuron is activated, else it is deactivated, meaning that its output is not passed on to the next hidden layer.

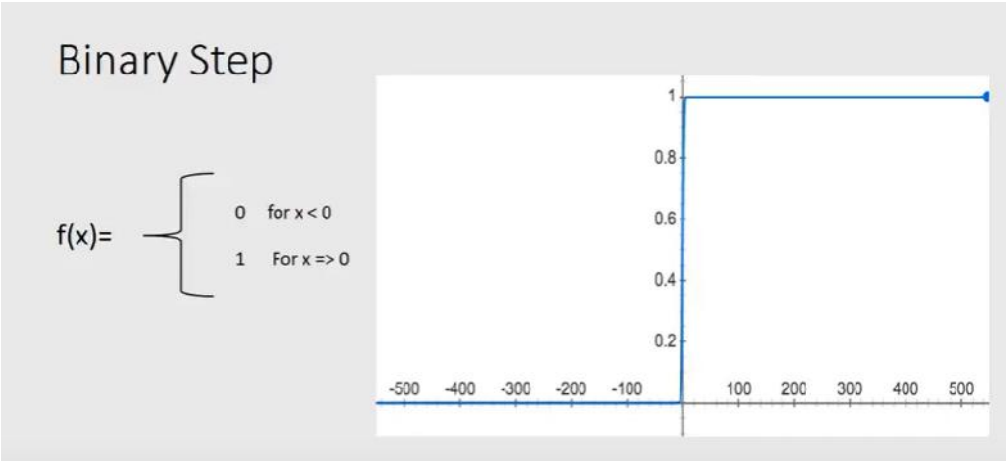


Figure 4 Binary step activation function

Linear activation function

The linear activation function as in Figure 5, also known as "no activation," or "identity function" (multiplied x1. 0), is where the activation is proportional to the input. The function doesn't do anything to the weighted sum of the input, it simply spits out the value it was given.

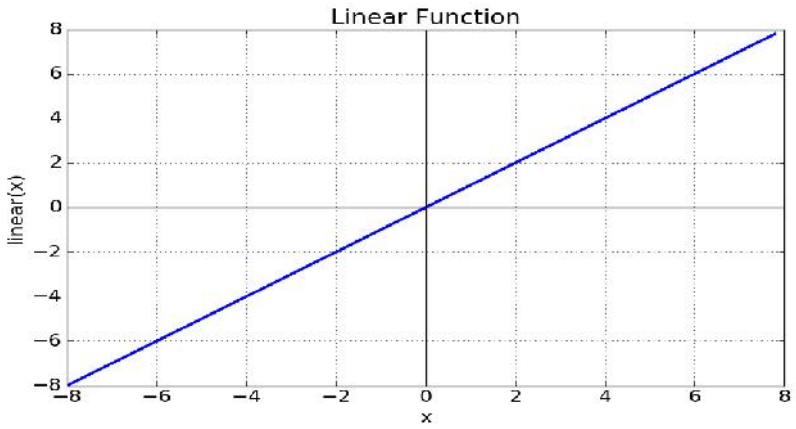


Figure 5 Linear Activation Function

Non-linear activation functions

The non-linear functions are known to be the most used activation functions. It makes it easy for a neural network model to adapt with a variety of data and to differentiate between the outcomes. These functions are mainly divided basis on their range or curves:

- Sigmoid Activation Functions
- Tanh Activation Functions
- ReLU Activation Functions
- Leaky Relu
- Softmax Activation Function

13.3 Perceptron

Perceptron was developed in 1958 by Frank Rosenblatt,a researcher in neuro-physiology to perform a kind of pattern recognition tasks. It resulted from the solution of classification problem.As shown in figure 6, the perceptron model takes a vector of real-valued inputs, calculates a linear combination of these inputs, then outputs +1 if the result is greater than the threshold and -1 if the result is not greater than the threshold.Perceptron was developed as simplest yet powerful classifier providing the linear separability of class patterns or examples. Perceptron can't handle tasks, which are not linearly separable. We can say that, A set of points in 2-dimensional space is linearly separable if this sets of points can be separated by a straight line. The perceptron criterion function is based on misclassification error, which means the number of samples misclassified. The perceptron architecture is as given below.

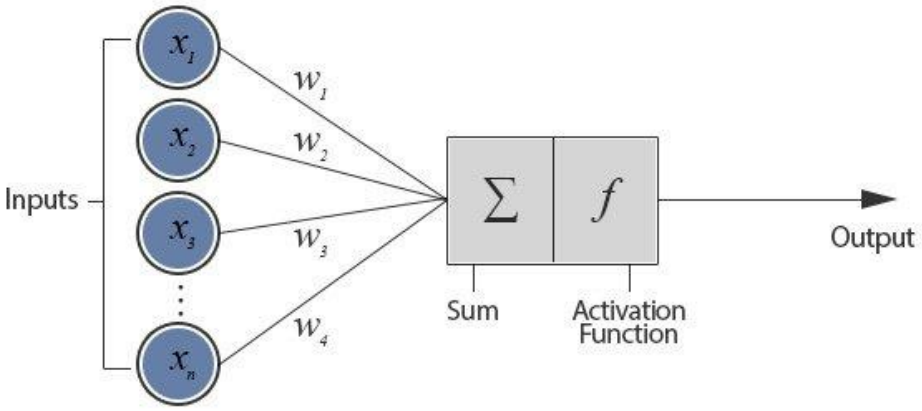


Figure 6 Structure of a Perceptron Model

Perceptron is otherwise known as single layer neural networks having the input layer and a single neuron in the output layer.It's not having any hidden layers. Moreover, the output neuron is the

only processing unit where the threshold activation function is used and gives the final output as +1 or -1.

The history has proved that the neural networks can overcome the limitations of Rosenblatt's perceptron. The neural networks primarily solve the regression problems based on minimum squared error criterion. This employed the gradient procedures for minimization. The perceptron was superseded by more sophisticated and powerful neuron and neural network structures. A popular network used today is the multilayer network, which is discussed in the next section.

13.4 Multi-layer Networks

The structure of artificial neural networks is given below, for example, as in Figure 7, which is having one input layer, two hidden layers and one output layer. Inputs enter into the processing element from the upper left.

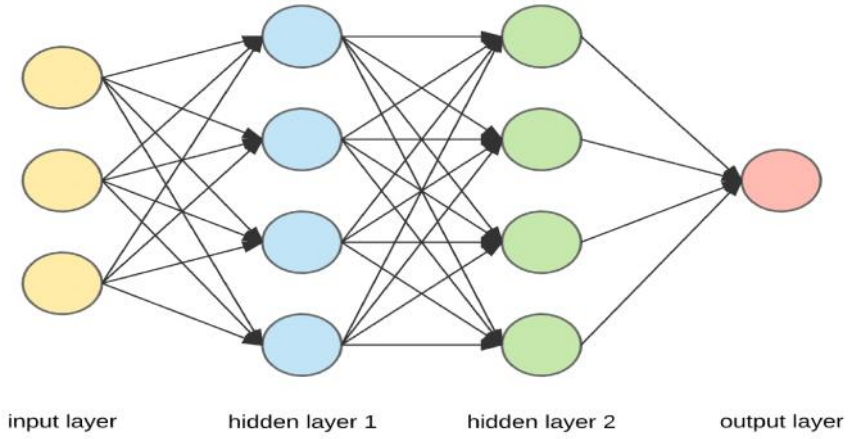


Figure 7 Artificial Neural Networks Architecture

The first step is to multiply each of these inputs by their respective weighting factor [w(n)]. These modified inputs are then fed into the summing function, which usually sums these products, however, many different types of operations can be selected. These operations can produce a number of different values, which are then propagated forward; values such as the average, the largest, the smallest, the ORed values, the ANDed values, etc. Other types of summing functions can also be created and sometimes they may be further complicated by the addition of an activation function which enables the summing function to operate in a time sensitive way.

The output of the summing function is then sent into a transfer function, which turns this number into a real output (a 0 or a 1, -1 or +1 or some other number) via some algorithm. The transfer function can also scale the output or control its value via thresholds. This output is then sent to other processing elements or an outside connection, as dictated by the structure of the network.

Binary-Class Classification

The Figure 4 is describing the concept of binary class model. The output layer is having only one neuron illustrating the yes / no response. Output = 1 means Yes and the output = 0 means No. Hence, this model is known as binary-class classification.

Any data, which can be classified into two categories, is also known as binary-class classification as shown in Figure 8.

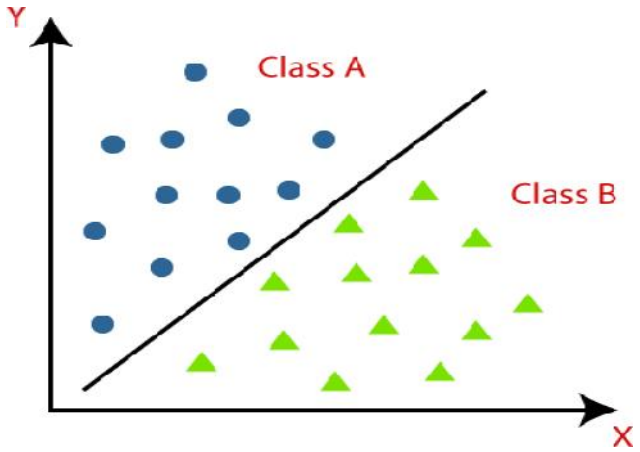


Figure 8 Binary-Class Classification

Multi-Class Classification Model

The Figure 9 is describing the concept of multi-class model. The output layer is having more than one neuron illustrating the yes / no response on each output neuron. Hence, this model is known as multi-class classification.

Any data, which can be classified into more than twocategories, is also known as multi-class classification as shown in Figure 10.

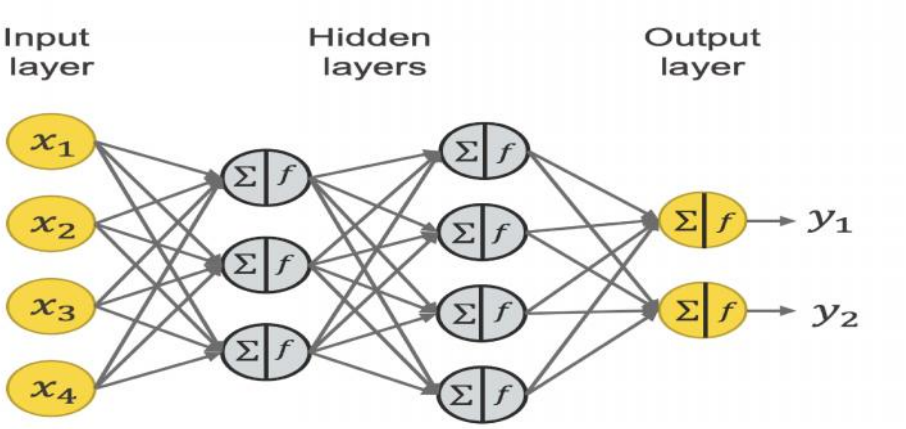


Figure 9 Structure of Multi-Class ANN Classifier

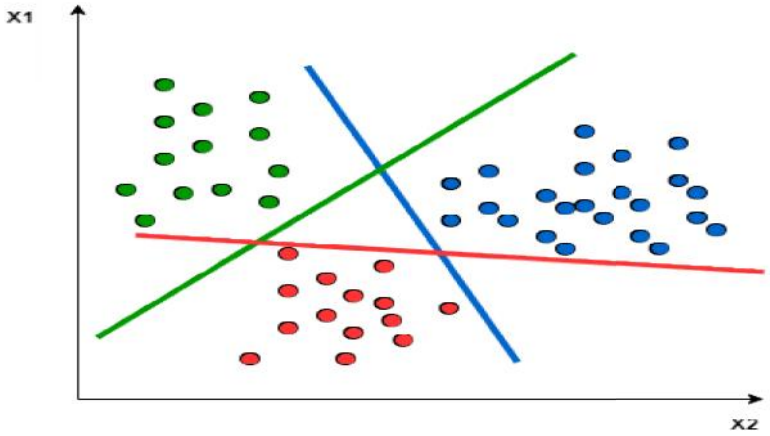


Figure 10Multi-Class Classification

Gradient descent serves as the basis for learning algorithms that search the hypothesis space of possible weight vectors to find the weights that best fit the training examples. The gradient-descent training rule for a single neuron is important because it provides the basis for the Backpropagation algorithm, which can learn networks with many interconnected units.

When it is interpreted as a vector in weight space, the gradient specifies the direction that produces the steepest increase in E, the derivative. The negative of this vector, therefore, gives the direction of steepest decrease. Here, eta is a positive constant (should be less than 1), called as the learning rate, which determines the step size in the gradient descent search.

Backpropagation Algorithm

MLP networks trained by the Backpropagation algorithm are capable of expressing a rich variety of nonlinear decision surfaces or approximating nonlinear functions. This also uses the gradient descent algorithms.The Backpropagation algorithm learns the weights for an MLP network, given a network with a fixed set of units and interconnections. It employs gradient descent to attempt to minimize the squared error between the network outputs and the target values for these outputs.

A typical feed-forward neural network is made up of a hierarchy of layers, and the neurons in the network are arranged along these layers as discussed in the previous section. The external environment is connected to the network through input layer and the output layer. The multi-layer perceptron network is a feed-forward neural network with one or more hidden layers. Each hidden layer has its own specific function. The working style of Backpropagation algorithm is divided into two parts. First is in forward direction and the second is in backward direction as shown in Figure 11 and figure 12.

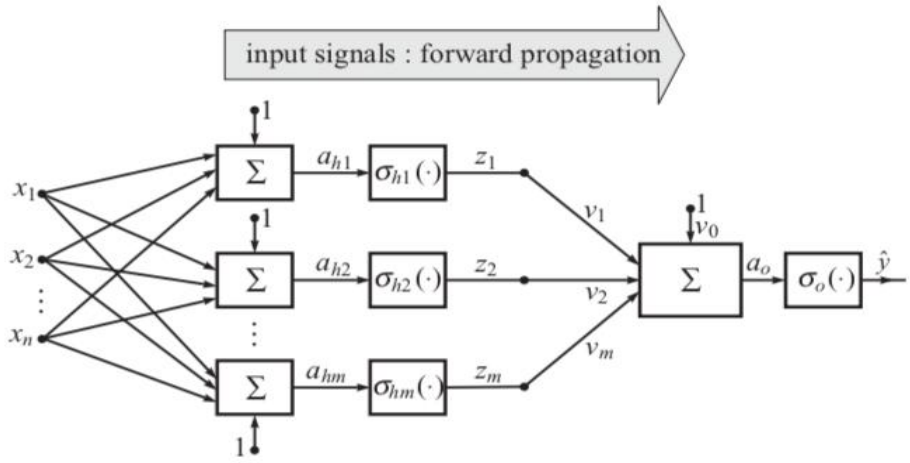


Figure 11 Forward Propagation in MLP

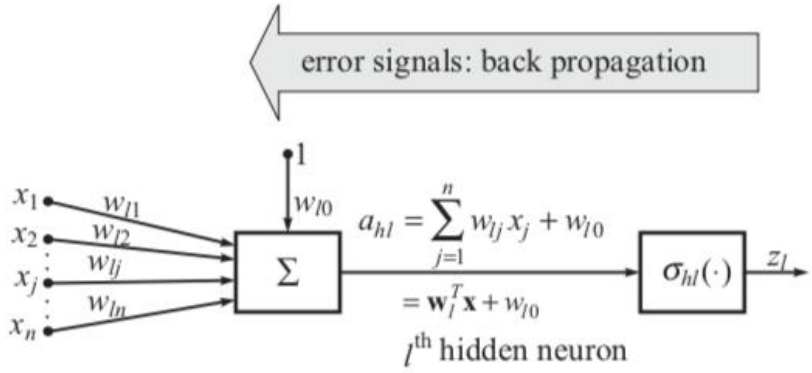


Figure 12 Backward Propagation in MLP

13.5 Introduction to Deep Neural Networks (DNN)

A deep neural network is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships. The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks.We have an input, an output, and a flow of sequential data in a deep network. Deep nets process data in complex ways by employing sophisticated math

modeling. The learning portion of creating models spawned the development of artificial neural networks.

13.6 Evaluation Metrics of Machine Learning Models

We evaluated the effectiveness of the categorization model using a number of assessment metrics, including the followings.

1. Accuracy: It is the proportion of precise predictions to all predictions.

Accuracy =(TP+TN)/(TP+FP+TN+FN)

2. Sensitivity: It is the proportion of true positives to all other positives found in the data.

Sensitivity =TP/(TP+FN)

3. Precision: The ratio is the total predicted positives divided by the sum of true positive and

False positive.

Precision = TP/(TP+FP)

4. Specificity: It is ratio of true negatives to total of true negative and false positive.

Specificity= TN/(TN+FP)

5. F_Score: It's the harmonic mean of the precision and recall.

F_Score=2*(Recall*Precision)/(Recall+Precision)

6. True Positive Rate (TPR) is a synonym for recall, hence the following definition applies:

TPR=TP/(TP+FN)

7. False Positive Rate (FPR) is characterised as follows:

FPR=FP/(FP+TN)

Summary

This unit explored the basic concepts of Artificial Neural Networks, stating from what is biological neuron. Understood that the imitation of biological neuron became artificial neuron. The processing of a artificial neuron was explained clearly using a diagram. The Structure of Artificial Neural Networks was discussed in detail.Understood the difference between Biological Neuron and Artificial Neuron.Importance of Activation Functions and different types of activation function was explained in this unit.In addition to this, the structure of perceptron model and multilayer perceptron model or feed-forward neural network was discussed along with back-propagation. An introduction to deep networks is also highlighted in this unit.

Keywords

- Biological Neuron
- Artificial Neuron
- Artificial Neural Networks
- Activation Function
- Binary classification
- Multi-class classification
- Perceptron
- Backpropagation
- Deep neural networks

Self Assessment

1. How many hidden layers can be present in a multi layer neural network?

A. 0
B. 1
C. 2
D. 'N'

2. The fundamental unit of network is _____.

A. Brain
B. Nucleus
C. Neuron
D. Axon

3. What are dendrites?

A. Fibers of nerves
B. Nuclear projections
C. Other name for nucleus
D. None of the above

4. What is an activation value?

A. Threshold value
B. Weighted sum of inputs
C. Main input to neuron
D. None of the above

5. The process of adjusting the weight is known as _____.

A. Activation
B. Synchronization
C. Learning
D. None of the above

6. Which is true for artificial neural networks?

A. It has set of nodes and connections
B. Each node computes it's weighted input
C. Node could be in excited state or non-excited state
D. All of the above

7. Artificial Neural Networks can be used in _____ fields.

A. Classification
B. Data Processing
C. Compression
D. All of the above

8. What is called if the output layer is having only one neuron?

A. Zero Classification
B. Binary Classification

C. Multi-class Classification

D. All the above

9. _____ is the range of output from Sigmoid Activation Function.

A. 0 to 1

B. -1 to 0

C. -1 to +1

D. None of the above

10. The first learning rule is developed by _____.

A. McCulloh

B. Pitss

C. Hebb

D. Minsky and Papert

11. _____ layer receives the input from the user.

A. Input Layer

B. Hidden Layer

C. Output Layer

D. All of the above

12. Justify the given statement.

"All the neurons exist in Artificial Neural Networks is processing the input and transfers the output to next layer."

A. Yes

B. No

13. Linear activation function is also known as _____.

A. Identity function

B. No activation function

C. Option (A) and (B)

D. None of These

14. Tangent Hyperbolic Activation function values ranging from _____ to _____.

A. 0 to 1

B. -1 to +1

C. -1 to 0

D. None of the above

15. Artificial Neural Network architecture is optimized using _____ after completing the training of the model.

A. Training Dataset

B. Testing Dataset

C. Validation Dataset

D. Computer vision

Answers for SelfAssessment

1.	D	2.	C	3	A	4.	B	5.	C
6.	D	7.	D	8.	B	9.	A	10.	C
11.	A	12.	B	13.	C	14.	B	15.	C

Review Questions

1. Explain the architecture of Artificial Neural Networks.
2. List the various tools used to implement ANN.
3. What are all the activation functions used for training ANN?
4. Givean example how the weights are adjusted.
5. Differentiate biological neuron and artificial neuron.

Further readings

- Madan Gopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://www.javatpoint.com/artificial-neural-network
- https://www.analyticsvidhya.com/blog/2021/05/beginners-guide-to-artificial-neural-network/
- https://www.techopedia.com/definition/5967/artificial-neural-network-ann
- https://towardsdatascience.com/an-introduction-to-deep-learning-af63448c122c

Unit 14: Neural Network Implementation

CONTENTS

Objectives
Introduction
14.1 What is Artificial Neural Network?
14.2 The Architecture of an Artificial Neural Network
14.3 Advantages of Artificial Neural Network (ANN)
14.4 Disadvantages of Artificial Neural Network
14.5 How do Artificial Neural Networks Work?
14.6 Types of Artificial Neural Network
14.7 Implementation of Machine Learning Algorithms
Summary
Keywords
Self Assessment
Answers for Self Assessment
Review Questions
Further Readings

Objectives

- understand basic of network security.
- understand network security issues
- learn security goals
- understand security services
- approaches of network security

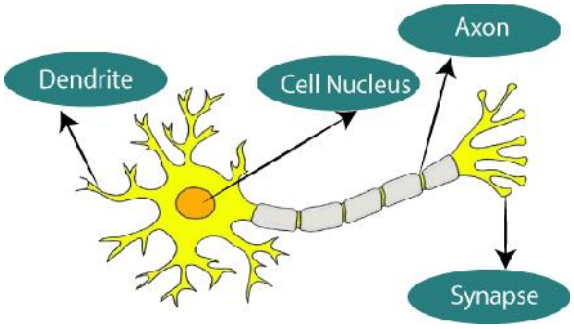
Introduction

Basic and advanced ideas of ANNs are provided via the Artificial Neural Network Tutorial. Our Artificial Neural Network tutorial was created for both professionals and beginners.

The phrase "artificial neural network" refers to a branch of artificial intelligence that was inspired by biology and is based on the brain. A computational network based on biological neural networks, which create the structure of the human brain, is typically referred to as an artificial neural network. Artificial neural networks also feature neurons that are linked to each other in different layers of the networks, just as neurons in a real brain. Nodes are the name for these neurons.

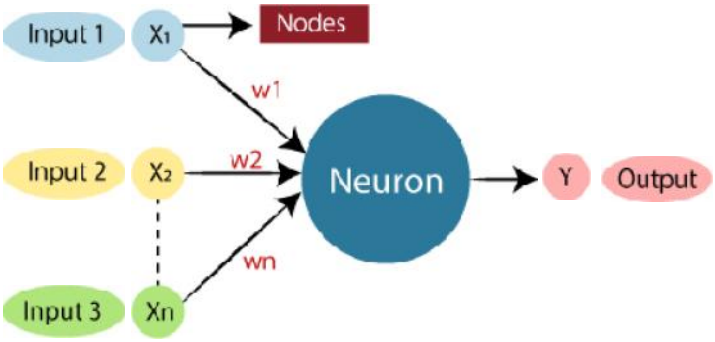
14.1 What is Artificial Neural Network?

The biological neural networks that shape the structure of the human brain are where the phrase "artificial neural network" originates. Artificial neural networks also feature neurons that are interconnected to one another in different levels of the networks, much like the human brain, which has neurons that are coupled to one another. Nodes are the name for these neurons.



The given figure illustrates the typical diagram of Biological Neural Network.

The typical Artificial Neural Network looks something like the given figure.



In artificial neural networks, dendrites from biological neural networks serve as inputs, cell nuclei serve as nodes, synapses serve as weights, and axons serve as outputs.

Biological neural networks and artificial neural networks are related:

Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

Artificial neural networks are used in artificial intelligence to simulate the network of neurons that make up the human brain, giving computers the ability to comprehend information and make decisions in a manner similar to that of a person. Computers are programmed to behave exactly like interconnected brain cells to create an artificial neural network.

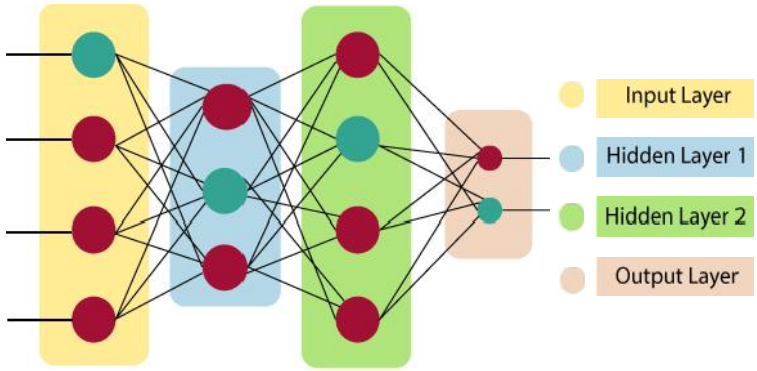
The human brain contains about 1000 billion neurons. Between 1,000 to 100,000 association points are present in each neuron. Data is distributedly stored in the human brain, allowing us to simultaneously access many pieces of information from memory as needed. The human brain is said to contain a staggering number of incredible parallel processors.

Consider an example of a digital logic gate that accepts input and outputs so that we may better grasp the artificial neural network. Two inputs are required for the "OR" gate. If either one or both of the inputs are "On," the output will also be "On". If both inputs are "Off," the output will also be "Off." In this case, output is dependent on input. Our brains do not carry out the same function. Because our brain's neurons are constantly "learning," the relationship between outputs and inputs is constantly changing.

14.2 The Architecture of an Artificial Neural Network

Understanding the components of a neural network is necessary to comprehend the idea of the architecture of an artificial neural network. A vast number of artificial neurons, also known as units, are placed in a hierarchy of layers to form what is known as a neural network. Let's examine the many layers that can be found in an artificial neural network.

Artificial Neural Network primarily consists of three layers:



Input Layer

As the name implies, it accepts inputs in a variety of programming-provided formats.

Hidden Layers

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layers

The hidden layer is used to transform the input into a variety of outputs, which are then communicated through this layer.

When given input, the artificial neural network computes the weighted total of the inputs and incorporates a bias. A transfer function is used to visualise this computation.

$$\sum_{i=1}^{n} Wi * Xi + b$$

In order to produce the output, it passes the weighted total as an input to an activation function. A node's activation functions determine whether or not it should fire. The output layer is only accessible to individuals who are fired. Depending on the type of task we are completing, there are many activation functions that can be used.

14.3 Advantages of Artificial Neural Network (ANN)

Processing in parallel capability:

Artificial neural networks have a numerical value that allows them to carry out multiple tasks at once.

archiving data over the network:

Traditional programming does not employ a database; instead, it stores data on the entire network. The network continues to function even if some data disappears from one location temporarily.

ability to work with limited information:

Following ANN training, the data may still produce output even with insufficient data. The relevance of the missing data in this situation is what causes the performance loss.

Having a spread of memories

Determining the instances and motivating the network in accordance with the intended output by showing it these examples is crucial for ANN to be able to adapt. The network's output can be false if the event can't be represented by the network in all of its characteristics because the network's succession is directly proportional to the selected occurrences.

a fault-tolerant attitude

The network is fault-tolerant since expropriation of one or more ANN cells does not prevent the network from producing output.

14.4 Disadvantages of Artificial Neural Network

Guarantee of appropriate network architecture:

The construction of artificialneural networks is not determined by any specific rules. Through experience, trial, and error, the right network structure is achieved.

Unrecognized networkbehavior:

It is the most important ANN issue. When an ANN generates a testing solution, it doesn't explain why or how. It erodes network confidence.

Hardware reliance

According to their structure, artificial neural networks require processors with parallel processing power. As a result, the equipment's realization is dependent.

Having trouble getting the network to see the problem:

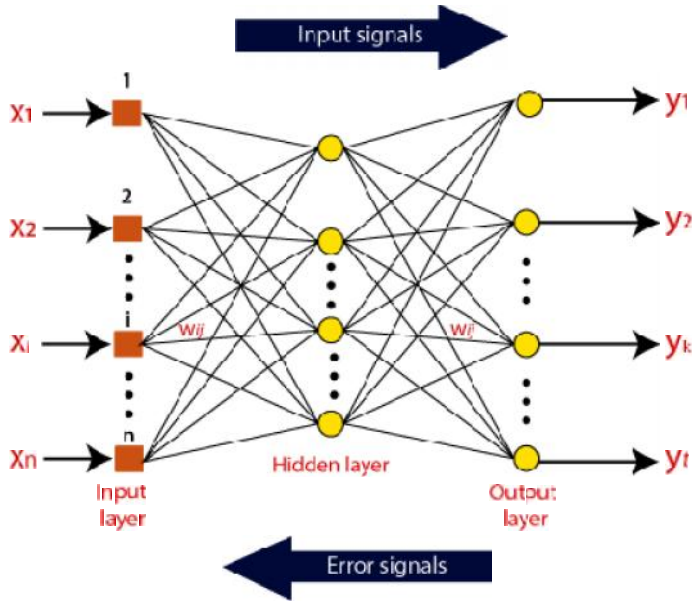
ANNs can process data that is numerical. Before using ANN, problems must be transformed into numerical values. The network's performance will be directly impacted by the presentation mechanism that must be decided here. It is dependent on the user's skills.

Unknown is the network's lifespan

The network is reduced to a particular error value, and this error value does not produce the best outcomes for us.

14.5 How do Artificial Neural Networks Work?

The ideal way to visualize an artificial neural network is as a weighted directed graph, where the nodes are the artificial neurons. The directed edges with weights represent the relationship between the neuron inputs and outputs. The input signal for the artificial neural network comes from an external source as a pattern and an image as a vector. Then, for each n-th input, these inputs are mathematically assigned using the notation x(n).



Each input is then multiplied by the weights that correspond to it (these weights are the information that the artificial neural networks use to solve a particular problem). In the artificial neural network, these weights often indicate how well neurons are connected to one another. Inside the computer unit, a summary of each weighted input is created.

The output is made non-zero by adding bias if the weighted total is equal to zero, or else something else is added to scale up the output to the system's reaction. The input for bias is the same, and the weight is 1. The sum of the weighted inputs in this case can range from 0 to positive infinity. Here, a certain maximum value is benchmarked to maintain the response within the bounds of the intended value, and the sum of the weighted inputs is fed through the activation function.

The set of transfer functions utilised to produce the desired output is referred to as the activation function. A variety of activation functions exist, although they are mainly either linear or non-linear sets of functions. The Binary, linear, and Tan hyperbolic sigmoidal activation function sets are a few of the often employed sets of activation functions. Let's examine each of these in more detail:

Binary

The output of a binary activation function is either a one or a zero. Here, a threshold value has been established in order to achieve this. The final output of the activation function is returned as one or 0 depending on whether the net weighted input of neurons is greater than 1.

Sigmoidal Hyperbolic:

The Sigmoidal Hyperbola function is generally seen as an "**S**" shaped curve. Here the tan hyperbolic function is used to approximate output from the actual net input. The function is defined as:

F(x) = (1/1 + exp(-????x))

Where ???? is considered the Steepness parameter.

14.6 Types of Artificial Neural Network

Artificial neural networks (ANN) come in a variety of forms, and they all carry out tasks in a way that is comparable to how human brain neuron and network functions. Most artificial neural networks will share certain characteristics with a biological counterpart that is more complicated, and they are quite good at what they are meant to do. segmentation or categorization, as examples.

Feedback ANN

The output of a feedback ANN is fed back into the network to achieve the best internally evolved results. based on the Centre for Atmospheric Research at the University of Massachusetts, Lowell.

The feedback networks are excellent for addressing optimization problems since they feed information back into themselves. Utilizing feedback ANNs, the internal system error repairs.

Feed-Forward ANN

A feed-forward network is a type of neural network that consists of at least one layer of neurons as well as input and output layers. The network's intensity can be observed based on the collective behavior of the connected neurons, and the output is chosen by evaluating the network's output in the context of its input. The main benefit of this network is that it learns to assess and identify input patterns.

14.7 Implementation of Machine Learning Algorithms

Certainly! I'll demonstrate the implementation and performance comparison of three popular machine learning classification models on different datasets. For this example, let's use the following models:

Logistic Regression

Random Forest

Support Vector Machine (SVM)

We'll use the scikit-learn library in Python for implementing these models and evaluating their performance. Additionally, we'll use three different datasets to showcase their performance across diverse scenarios. Let's proceed with the implementation.

Step 1: Import the Necessary libraries and load the datasets

```
import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn import datasets


# Load the datasets

dataset_1 = datasets.load_iris()  # Iris dataset

dataset_2 = datasets.load_digits()  # Digits dataset

dataset_3 = datasets.load_breast_cancer()  # Breast cancer dataset
```

Step 2: Prepare the datasets for training and testing

```
# Prepare dataset 1 (Iris)

X1 = dataset_1.data

y1 = dataset_1.target

X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(X1, y1, test_size=0.2, random_state=42)


# Prepare dataset 2 (Digits)

X2 = dataset_2.data

y2 = dataset_2.target

X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(X2, y2, test_size=0.2, random_state=42)


# Prepare dataset 3 (Breast Cancer)
```

```
X3 = dataset_3.data
y3 = dataset_3.target
X_train_3, X_test_3, y_train_3, y_test_3 = train_test_split(X3, y3, test_size=0.2, random_state=42)
```

Step 3: Implement and train the models

```
# Implement Logistic Regression
lr_model = LogisticRegression()
lr_model.fit(X_train_1, y_train_1)

# Implement Random Forest
rf_model = RandomForestClassifier()
rf_model.fit(X_train_2, y_train_2)

# Implement SVM
svm_model = SVC()
svm_model.fit(X_train_3, y_train_3)
```

Step 4: Evaluate the models and compare their performances

```
from sklearn.metrics import accuracy_score

# Evaluate Logistic Regression
lr_predictions = lr_model.predict(X_test_1)
lr_accuracy = accuracy_score(y_test_1, lr_predictions)
print("Logistic Regression Accuracy:", lr_accuracy)

# Evaluate Random Forest
rf_predictions = rf_model.predict(X_test_2)
rf_accuracy = accuracy_score(y_test_2, rf_predictions)
print("Random Forest Accuracy:", rf_accuracy)

# Evaluate SVM
svm_predictions = svm_model.predict(X_test_3)
svm_accuracy = accuracy_score(y_test_3, svm_predictions)
print("Support Vector Machine Accuracy:", svm_accuracy)
```

By executing the code, you will obtain the accuracy scores for each model on their respective datasets. This allows for a performance comparison between the three models. Keep in mind that accuracy alone may not provide a comprehensive evaluation, and it's advisable to consider additional metrics based on the specific requirements and characteristics of your datasets.

Summary

- The phrase "artificial neural network" refers to a branch of artificial intelligence that was inspired by biology and is based on the brain. A computational network based on biological neural networks, which create the structure of the human brain, is typically referred to as an artificial neural network.

- The biological neural networks that shape the structure of the human brain are where the phrase "artificial neural network" originates
- The components of a neural network is necessary to comprehend the idea of the architecture of an artificial neural network. A vast number of artificial neurons, also known as units, are placed in a hierarchy of layers to form what is known as a neural network.
- The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.
- Artificial neural networks have a numerical value that allows them to carry out multiple tasks at once.
- Traditional programming does not employ a database; instead, it stores data on the entire network. The network continues to function even if some data disappears from one location temporarily.
- The structures and operations of human neurons serve as the basis for artificial neural networks. It is also known as neural networks or neural nets.
- Synapses are the connections that allow impulses to be sent from dendrites to the cell body of biological neurons. In artificial neurons, synapse weights connect the one-layer nodes to the next-layer nodes.
- Learning takes place in the cell body nucleus or soma of biological neurons, which possesses a nucleus that aids in impulse processing. If the impulses are strong enough to pass the threshold, an action potential is created and moves through the axons.
- The pace at which a biological neuron fires when an impulse is potent enough to cross the threshold is known as activation.

Keywords

- ***Artificial Neural Networks***: Computational models inspired by the structure and functioning of the human brain, used in machine learning for solving complex problems by simulating interconnected artificial neurons.
- ***Perceptron***: The fundamental building block of an artificial neural network, comprising a weighted input sum, an activation function, and an output. It processes input data and produces a binary output based on the weighted sum.
- ***Activation Function***: A mathematical function applied to the output of a perceptron or a neuron in a neural network. It introduces non-linearity, allowing the network to model complex relationships and make predictions based on the input.
- ***Feedforward Neural Networks***: Neural networks composed of interconnected layers of perceptrons or neurons, where information flows only in one direction, from the input layer through hidden layers to the output layer. They are used for tasks such as classification and regression.
- ***Backpropagation:*** An algorithm used to train neural networks by adjusting the weights based on the calculated errors between the predicted outputs and the desired outputs. It uses gradient descent to iteratively minimize the error and improve the network's performance.
- ***Gradient Descent:*** An optimization algorithm used in backpropagation to update the weights of a neural network. It calculates the gradient of the error function with respect to the weights and adjusts the weights in the direction of steepest descent to minimize the error.
- ***Multilayer Perceptron***: A type of feedforward neural network with multiple hidden layers between the input and output layers. It is a versatile architecture capable of learning complex relationships and widely used for various tasks.
- ***Convolutional Neural Networks***: Neural networks specifically designed for processing grid-like data, such as images. They utilize convolutional layers to extract features hierarchically and are effective in tasks like image classification and object detection.
- ***Recurrent Neural Networks***: Neural networks designed for processing sequential data with temporal dependencies. They have feedback connections that enable them to store and utilize

information from previous time steps, making them suitable for tasks like natural language processing and time series analysis.

- ***Image Classification***: The task of assigning labels or categories to images based on their content. Artificial neural networks, particularly convolutional neural networks, have shown remarkable performance in image classification tasks.
- ***Object Detection:*** The process of identifying and locating objects within images or videos. Convolutional neural networks are widely used in object detection algorithms to accurately detect and classify objects.
- ***Natural Language Processing***: A field of study focused on enabling computers to understand, interpret, and generate human language. Artificial neural networks, including recurrent neural networks, are used in tasks such as language translation, sentiment analysis, and text generation.
- ***Time Series Analysis***: The analysis of data points collected over time to identify patterns, trends, and make predictions. Recurrent neural networks are effective in time series analysis as they can model temporal dependencies and capture long-term patterns.
- ***Recommendation Systems***: Systems that provide personalized recommendations to users based on their preferences and behaviors. Neural networks, particularly collaborative filtering techniques, are commonly used to build recommendation systems and improve the accuracy of recommendations.
- ***Hyperparameter Tuning***: The process of selecting the optimal values for the hyperparameters of a neural network. It involves techniques such as grid search and random search to find the hyperparameter configuration that maximizes the model's performance.
- ***Regularization Techniques***: Techniques used to prevent overfitting in neural networks by introducing additional constraints or penalties on the model's parameters. Common regularization techniques include L1 and L2 regularization, dropout, and early stopping.
- ***Model Evaluation***: The process of assessing the performance of a trained neural network. It involves using various metrics such as accuracy, precision, recall, and F1 score to measure how well the model predicts the desired outputs.
- ***Accuracy:*** A metric used to measure the performance of a classification model, representing the ratio of correctly predicted instances to the total number of instances in the dataset.
- ***Precision***: A metric that measures the proportion of true positive predictions out of all positive predictions made by the model. It is used to evaluate the correctness of positive predictions.
- ***Recall***: A metric that measures the proportion of true positive predictions out of all actual positive instances in the dataset. It is used to evaluate the model's ability to find all positive instances.
- ***F1 Score***: A metric that combines precision and recall into a single value, providing a balanced measure of a model's performance. It is the harmonic mean of precision and recall, and it is useful when the class distribution is imbalanced.

SelfAssessment

1. Why are biological neural networks necessary?

A. To tackle problems involving machine vision and natural language

B. processing, as well as to use heuristic search techniques to locate solutions.

C. to create a clever, user-friendly, interactive human system.

D. all the aforementioned

2. What is the current fad in software?

A. to solve complex issues,

B. to be task-specific,
C. to bring the computer closer to the user,
D. and to be versatile.

3. What distinguishes human intelligence from that of machines?
A. Humans see everything as a pattern, whereas machines just see it as data.
B. Humans are emotional.
C. Humans possess higher IQ and intellect
D. human beings have sensory organs.

4. What does the neural network's auto-association task entail?
A. Determine the relationship between two successive inputs
B. a task involving storage and recall
C. anticipating future inputs
D. None of the previously listed

5. Unsupervised learning: What is it?
A. Clearly stated group characteristics
B. There could be a lot of groups.
C. Neither the feature nor the total number of groups are known.
D. None of the previously listed

6. An illustration of an unsupervised feature map?
A. Text recognition
B. first voice recognition
C. Image recognition
D. None of the previously listed

7. What does neural network plasticity entail?
A. The input pattern is ever-changing
B. The input pattern has stopped changing.
C. Constantly shifting output patterns
D. Static output

8. Describe categorization.
A. Choosing the features to include in a pattern recognition challenge.
B. establishing the class to which an input pattern belongs.
C. selecting the neural network type to be used
D. None of the previously listed

9. What do neural network models consist of?

A. A mathematical illustration of our comprehension
B. Biological neural network depiction.
C. Both ways
D. None of the previously listed

10. What kind of dynamics are immediately impacted by changing inputs?
A. Synaptic
B. Neural
C. Activation
D. both neuronal and synaptic

11. Which use of neural networks is the most direct?
A. Vector quantization is a.
B. Pattern mapping
C. pattern recognition
D. Control software

12. Why do neural networks have advantages over computers?
A. They are able to learn from b examples because
B. They have extremely fast real-time computing rates.
C. They are more tolerant .
D. all the aforementioned

13. Artificial Neural Networks are inspired by:
A. Biological neurons and neural networks in the brain
B. Genetic algorithms and evolutionary processes
C. Decision trees and rule-based systems
D. Statistical methods like linear regression

14. The basic building blocks of an Artificial Neural Network are:
A. Neurons
B. Genes
C. Decision trees
D. Data points

15. The weights and biases in an ANN are learned during the process of:
A. Forward propagation
B. Backpropagation
C. Gradient descent
D. Feature engineering

Answers for Self Assessment

1.	D	2.	A	3.	A	4.	B	5.	C
6.	B	7.	A	8.	B	9.	C	10.	C
11.	C	12.	D	13.	A	14.	A	15.	B

Review Questions

1. Explain the concept of a perceptron and how it functions within an artificial neural network.
2. Discuss the importance of activation functions in artificial neural networks. Provide examples of commonly used activation functions and their characteristics.
3. Describe the backpropagation algorithm and its role in training artificial neural networks. Explain how gradient descent is utilized in backpropagation.
4. Compare and contrast feedforward neural networks and recurrent neural networks. Discuss the advantages and applications of each type.
5. Explain the architecture and working principles of convolutional neural networks (CNNs). Discuss their significance in image processing tasks such as image classification and object detection.
6. Describe the concept of regularization in neural networks. Discuss common regularization techniques used to prevent overfitting and improve model generalization.
7. Discuss the importance of hyperparameter tuning in neural networks. Explain different methods and strategies for finding optimal hyperparameter configurations.
8. Explain the concept of model evaluation in artificial neural networks. Discuss commonly used evaluation metrics and their significance in assessing model performance.
9. Discuss the challenges and limitations of artificial neural networks. Highlight specific areas where neural networks may face difficulties or exhibit limitations.
10. Describe the applications of artificial neural networks in real-world scenarios, such as natural language processing, time series analysis, or recommendation systems. Provide examples and discuss their effectiveness in these applications.

Further Readings

- Madan Gopal, Applied Machine Learning, McGraw Hill Education, India, 2018.
- S. N. Sivanandam, S.N. Deepa, Principles Of Soft Computing, Wiley Publications, Second Edition, 2011.
- Rajasekaran, S., Pai, G. A. Vijayalakshmi, Neural Networks, Fuzzy Logic and Genetic Algorithm Synthesis And Applications, Prentice Hall of India, 2013.
- N. P. Padhy, S. P. Simon, Soft Computing With Matlab Programming, Oxford University Press, 2015.

Web Links

- https://www.javatpoint.com/artificial-neural-network
- https://www.analyticsvidhya.com/blog/2021/05/beginners-guide-to-artificial-neural-network/
- https://www.techopedia.com/definition/5967/artificial-neural-network-ann

- https://towardsdatascience.com/an-introduction-to-deep-learning-af63448c122c

LOVELY PROFESSIONAL UNIVERSITY
Jalandhar-Delhi G.T. Road (NH-1)
Phagwara, Punjab (India)-144411
For Enquiry: +91-1824-521360
Fax.: +91-1824-506111
Email: odl@lpu.co.in