PRACTICAL MANUAL

Subject Name: : Data Science and Big Data Analytics

Subject Code:310251

Class: TE

Department of Computer Engineering

Sir Visvesvaraya Institute of Technology, Nashik

Vision of Institute

To provide quality technical education in rural area to create competent human resources.

Mission of Institute

Committed to produce competent engineers to cater the needs of society by imparting skill based education through effective teaching learning process.

Vision of Computer Engineering Department

Develop Department of Computer Engineering into center of excellence through imparting technical education of international standards and research in the field of Computer Engineering.

Mission of Computer Engineering Department

To provide quality engineering education to the students through state of art education in Computer Engineering.

Program Outcomes (POs):

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

Program Specific Outcomes (PSO):

PSO1: Professional Skills-The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying.

PSO2: Problem-Solving Skills- The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO3: Successful Career and Entrepreneurship- The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

Program Education Objectives:

PEO 1: To prepare globally competent graduates having strong fundamentals, domain knowledge, updated with modern technology to provide the effective solutions for engineering problems.

PEO 2: To prepare the graduates to work as a committed professional with strong professional ethics and values, sense of responsibilities, understanding of legal, safety, health, societal, cultural and environmental issues.

PEO 3: To prepare committed and motivated graduates with research attitude, lifelong learning, investigative approach, and multidisciplinary thinking.

PEO 4: To prepare the graduates with strong managerial and communication skills to work effectively as individual as well as in teams.

Scheme, Course Outcomes, Syllabus, and Evaluation guidelines

TE (Computer Engineering) Sem-II,

Data Science and Big Data Analytics (310251)

Course Code	Course Name	Lectures Assigned			
		Theory	Practical	Tutorial	Total
310251	**Data Science and Big Data Analytics**	03	04	--	07

Evaluation GuidelinesInternal Assessment (IA)

Course Code	Course Name	Evaluation Scheme								
310251	**Data Science and Big Data Analytics**	Theory					Practical			Total Credit
		Internal	External			Total	Internal/term work	External/practical	Total	
			Insem	Endsem	Total					
		00	30	70	100	100	50	25	75	3+2

Two tests must be conducted which should cover at least 65% of syllabus. The average marks of two tests will be considered for term work marks.

Insem Exam(30 Marks)

Insem Exam of 30 marks will be conducted by university which covers first 2 units (32%) of syllabus. The marks will be considered as final marks.

End Semester Examination

Question Paper will carry 8 questions.Total 4 questions need to be solved.

Attempt Q1. Or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8 .

310256:Data Science and Big Data Analytics Laboratory

Teaching Scheme Practical: 04 Hours/Week	**Credit:02**	**Examination Scheme and Marks Term work: 50 Marks** **Practical: 25 Marks**

Companion Course: Data Science and Big Data Analytics (310251)

Course Objectives:

- To understand principles of Data Science for the analysis of real time problems
- To develop in depth understanding and implementation of the key technologies in Data Science and Big Data Analytics
- To analyze and demonstrate knowledge of statistical data analysis techniques for decision- making
- To gain practical, hands-on experience with statistics programming languages and Big Data tools

Course Outcomes:

On completion of the course, learners will be able to

CO1: Apply principles of Data Science for the analysis of real time problems

CO2: Implement data representation using statistical methods

CO3: Implement and evaluate data analytics algorithms

CO4: Perform text preprocessing

CO5: Implement data visualization techniques

CO6: Use cutting edge tools and technologies to analyze Big Data

Subject: Data Science and Big Data Analytics (310251)

CO-PO /PSO mapping

The CO-PO Mapping Matrix

CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	3	2	1	-	-	-	-	1	-	-	1
CO2	1	2	1	2	-	1	-	-	1	-	-	1
CO3	2	1	2	1	-	1	-	-	1	-	-	1
CO4	1	2	2	2	2	-	-	-	1	-	-	1
CO5	1	2	2	1	2	-	-	-	1	-	-	1
CO6	1	2	1	2	2	-	-	-	1	-	-	1

Suggested List of Laboratory Experiments/Assignments

Assignments from all Groups (A,B,C) are compulsory.

Sr. No.	Group A : Data Science
1.	**Data Wrangling, I** Perform the following operations using Python on any open source dataset (e.g., data.csv) 1. Import all the required Python Libraries. 2. Locate an open source data from the web (e.g. https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site). 3. Load the Dataset into pandas data frame. 4. Data Preprocessing: check for missing values in the data using pandas insult(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame. 5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions. 6. Turn categorical variables into quantitative variables in Python. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set.
2.	**Data Wrangling II** Create an "Academic performance" dataset of students and perform the following operations using Python. 1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them. 2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them. 3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution. Reason and document your approach properly.

3.	**Descriptive Statistics - Measures of Central Tendency and variability** Perform the following operations on any open source dataset (e.g., data.csv) 1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable. 2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset. Provide the codes with outputs and explain everything that you do in this step.
4.	**Data Analytics I** Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset. The objective is to predict the value of prices of the house using the given features.
5.	**Data Analytics II** 1. Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset. 2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.
6.	**Data Analytics III** 1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset. 2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.
7.	**Text Analytics** 1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization. 2. Create representation of document by calculating Term Frequency and Inverse Document Frequency.
8.	**Data Visualization I** 1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data. 2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

9.	**Data Visualization II** 1. Use the inbuilt dataset 'titanic' as used in the above problem. Plot a box plot for distribution of age with respect to each gender along with the information about whether they survived or not. (Column names : 'sex' and 'age') 2. Write observations on the inference from the above statistics.
10.	**Data Visualization III** Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset and give the inference as: 1. List down the features and their types (e.g., numeric, nominal) available in the dataset. 2. Create a histogram for each feature in the dataset to illustrate the feature distributions. 3. Create a box plot for each feature in the dataset. 4. Compare distributions and identify outliers.
	Group B- Big Data Analytics – JAVA/SCALA (Any three)
1.	Write a code in JAVA for a simple Word Count application that counts the number of occurrences of each word in a given input set using the Hadoop Map-Reduce framework on local-standalone set-up.
2.	Design a distributed application using Map-Reduce which processes a log file of a system.
3.	Locate dataset (e.g., sample_weather.txt) for working on weather data which reads the text input files and finds average for temperature, dew point and wind speed.
4.	Write a simple program in SCALA using Apache Spark framework
	Group C- Mini Projects/ Case Study – PYTHON/R (Any TWO Mini Project)
1.	Write a case study on Global Innovation Network and Analysis (GINA). Components of analytic plan are 1. Discovery business problem framed, 2. Data, 3. Model planning analytic technique and 4. Results and Key findings.
2.	Use the following dataset and classify tweets into positive and negative tweets. https://www.kaggle.com/ruchi798/data-science-tweets
3.	Develop a movie recommendation model using the scikit-learn library in python. Refer dataset https://github.com/rashida048/Some-NLP-Projects/blob/master/movie_dataset.csv
4.	Use the following covid_vaccine_statewise.csv dataset and perform following analytics on the given dataset https://www.kaggle.com/sudalairajkumar/covid19-in-india?select=covid_vaccine_statewise.csv a. Describe the dataset b. Number of persons state wise vaccinated for first dose in India c. Number of persons state wise vaccinated for second dose in India d. Number of Males vaccinated d. Number of females vaccinated

5. Write a case study to process data driven for Digital Marketing **OR** Health care systems with Hadoop Ecosystem components as shown. (Mandatory)

    - HDFS: Hadoop Distributed File System
    - YARN: Yet Another Resource Negotiator
    - MapReduce: Programming based Data Processing
    - Spark: In-Memory data processing
    - PIG, HIVE: Query based processing of data services
    - HBase: NoSQL Database (Provides real-time reads and writes)
    - Mahout, Spark MLLib: (Provides analytical tools) Machine Learning algorithm libraries
    - Solar, Lucene: Searching and Indexing

Learning Resources

Reference Books :

1. Chirag Shah, "A Hands-On Introduction To Data Science", Cambridge University Press,(2020), ISBN : ISBN 978-1-108-47244-9.
2. Wes McKinney, "Python for Data Analysis", O' Reilly media, ISBN : 978-1-449-31979-3.
3. "Scikit-learn Cookbook", Trent hauk,Packt Publishing, ISBN: 9781787286382
4. R Kent Dybvig, "The Scheme Programming Language", MIT Press, ISBN 978-0-262-51298-5.
5. Jenny Kim, Benjamin Bengfort, "Data Analytics with Hadoop", OReilly Media, Inc.
6. Jake VanderPlas,"Python Data Science Handbook" https://tanthiamhuat.files.wordpress.com/2018/04/pythondatasciencehandbook.pdf
7. Gareth James, "An Introduction to Statistical Learning" https://www.ime.unicamp.br/~dias/Intoduction%20to%20Statistical%20Learning.pdf
8. Cay S Horstmann, "Scala for the Impatient", Pearson, ISBN: 978-81-317-9605-4,
9. Alvin Alexander, "Scala Cookbook", O'Reilly, SPD,ISBN: 978-93-5110-263-2

Web Links:

- https://www.simplilearn.com/data-science-vs-big-data-vs-data-analytics-article
- https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
- https://www.edureka.co/blog/hadoop-ecosystem
- https://www.edureka.co/blog/mapreduce-tutorial/#mapreduce_word_count_example
- https://github.com/vasanth-mahendran/weather-data-hadoop
- https://spark.apache.org/docs/latest/quick-start.html#more-on-dataset-operations
- https://www.scala-lang.org/

MOOCs Courses link:

- https://nptel.ac.in/courses/106/106/106106212/
- https://onlinecourses.nptel.ac.in/noc21_cs33/preview
- https://nptel.ac.in/courses/106/104/106104189/
- https://onlinecourses.nptel.ac.in/noc20_cs92/preview

The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	3	1	1	-	-	-	1	-	1	3
CO2	2	2	3	1	2	-	-	-	1	-	-	3
CO3	2	2	3	2	2	2	-	-	2	-	1	3
CO4	2	2	2	2	2	-	-	-	-	-	-	3
CO5	2	2	3	3	3	1	-	-	2	-	2	3
CO6	2	2	1	1	3	2	1	-	2	-	2	1

ASSIGNMENT NO.01

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	**4**	**4**	**4**	**4**	**20**	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment: Data Wrangling, I

Perform the following operations using Python on any open source dataset (e.g., data.csv) Import all the required Python Libraries.

1. Locate open source data from the web (e.g. https://www.kaggle.com).
2. Provide a clear description of the data and its source (i.e., URL of the web site).
3. Load the Dataset into the pandas data frame.
4. Data Preprocessing: check for missing values in the data using pandas insult(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.
5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
6. Turn categorical variables into quantitative variables in Python.

Objective of the Assignment: Students should be able to perform the data wrangling operation using Python on any open source dataset

Prerequisite:

1. Basic of Python Programming
2. Concept of Data Preprocessing, Data Formatting , Data Normalization and Data Cleaning.

Theory:

I. Introduction to Dataset:

A dataset is a collection of records, similar to a relational database table. Records are similar to table rows, but the columns can contain not only strings or numbers, but also nested data structures such as lists, maps, and other records.

Instance: A single row of data is called an instance. It is an observation from the domain. Feature: A single column of data is called a feature. It is a component of an observation and is also called an attribute of a data instance. Some features may be inputs to a model (the predictors) and others may be outputs or the features to be predicted.

Data Type: Features have a data type. They may be real or integer-valued or may have a categorical or ordinal value. You can have strings, dates, times, and more complex types, but typically they are reduced to real or categorical values when working with traditional machine learning methods.

Datasets: A collection of instances is a dataset and when working with machine learning methods we typically need a few datasets for different purposes.

Training Dataset: A dataset that we feed into our machine learning algorithm to train our model.

Testing Dataset: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.

II. Pandas Data Types

Pandas dtype	Python type	NumPy type	Usage
object	str or mixed	string_, unicode_, mixed types	Text or mixed numeric and non-numeric values
int64	int	int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64	Integer numbers
float64	float	float_, float16, float32, float64	Floating point numbers
bool	bool	bool_	True/False values
datetime64	NA	datetime64[ns]	Date and time values

timedelta[ns]	NA	NA	Differences between two datetimes
category	NA	NA	Finite list of text values

a. Pandas

Pandas is an open-source Python package that provides high-performance, easy-to-use data structures and data analysis tools for the labeled data in Python programming language.

b. What can you do with Pandas?

1. Indexing, manipulating, renaming, sorting, merging data frame
2. Update, Add, Delete columns from a data frame
3. Impute missing files, handle missing data or NANs
4. Plot data with histogram or box plot
5.

III. Panda Dataframe functions for Data Preprocessing :

Sr. No	Data Frame Function	Description
1	dataset.head(n=5)	Return the first n rows.
2	dataset.tail(n=5)	Return the last n rows.
3	dataset.index	The index (row labels) of the Dataset.
4	dataset.columns	The column labels of the Dataset.
5	dataset.shape	Return a tuple representing the dimensionality of the Dataset.
6	dataset.dtypes	Return the dtypes in the Dataset.

		This returns a Series with the data type of each column. The result's index is the original Dataset's columns. Columns with mixed types are stored with the object dtype.
7	**dataset.columns.values**	Return the columns values in the Dataset in array format
8	**dataset.describe(include='all')**	Generate descriptive statistics. to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. Analyzes both numeric and object series, as well as Dataset column sets of mixed data types.
9	**dataset['Column name]**	Read the Data Column wise.
10	**dataset.sort_index(axis=1, ascending=False)**	Sort object by labels (along an axis).
11	**dataset.sort_values(by="Column name")**	Sort values by column name.
12	**dataset.iloc[5]**	Purely integer-location based indexing for selection by position.
13	**dataset[0:3]**	Selecting via [], which slices the rows.
14	**dataset.loc[:, ["Col_name1", "col_name2"]]**	Selection by label

15	**dataset.iloc[:n, :]**	a subset of the first n rows of the original data
16	**dataset.iloc[:, :n]**	a subset of the first n columns of the original data
17	**dataset.iloc[:m, :n]**	a subset of the first m rows and the first n columns

IV. Checking of Missing Values in Dataset:

- **isnull()** is the function that is used to check missing values or null values in pandas python.
- isna() function is also used to get the count of missing values of column and row wise count of missing values

a. is there any missing values across each column Function: DataFrame.isnull().any()

b. count of missing values across each column using isna() and isnull() In order to get the count of missing values of the entire dataframe isnull() function is used. sum() which does the column wise sum first and doing another sum() will get the count of missing values of the entire dataframe. Function: dataframe.isnull().sum().sum()

c. count row wise missing value using isnull() Function: dataframe.isnull().sum(axis = 1)

d. count Column wise missing value using isnull()
Function: dataframe.isnull().sum()

e. count of missing values of a specific column.
Function:dataframe.col_name.isnull().sum()

f. groupby count of missing values of a column. In order to get the count of missing values of the particular column by group in pandas we will be using isnull() and sum() function with apply() and groupby() which performs the group wise count of missing values as shown below.

Function: df1.groupby(['Gender'])['Score'].apply(lambda x: x.isnull().sum())

V. Conclusion-

ASSIGNMENT NO.02

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment: Data Wrangling, II

Perform the following operations using Python on any open source dataset (e.g., data.csv) Import all the required Python Libraries.

1. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
2. Turn categorical variables into quantitative variables in Python.

Objective of the Assignment: Students should be able to perform the data wrangling operation using Python on any open source dataset

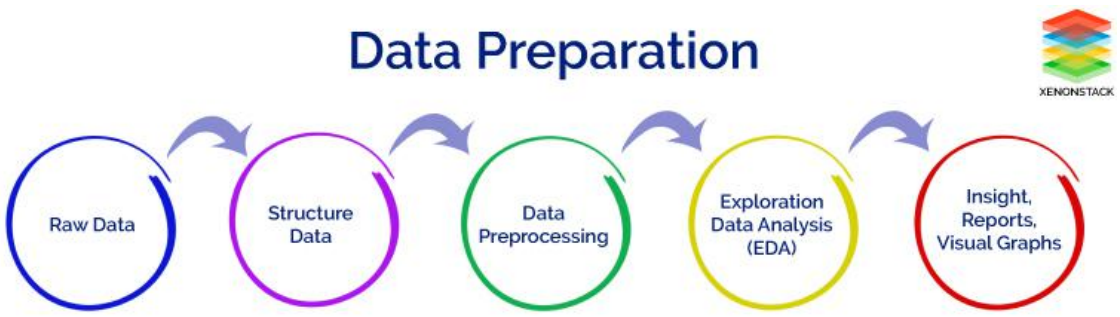
Prerequisite:

1. Basic of Python Programming
2. Concept of Data Preprocessing, Data Formatting , Data Normalization and Data Cleaning.

Theory:

I. Data Preparation Architecture

Data Preparation process s an important part of Data Science. It includes two concepts such as **Data Cleaning** and **Feature Engineering**. These two are compulsory for achieving better accuracy and performance in the Machine Learning and Deep Learning projects.



II. What is Data Preprocessing?

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set. This technique is performed before the execution of the **Iterative Analysis**. The set of steps is known as Data Preprocessing. It includes -

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

- **What Is Data Wrangling?**

Data Wrangling is a technique that is executed at the time of making an interactive model. In other words, it is used to convert the raw data into the format that is convenient for the consumption of data. This technique is also known as **Data Munging**. This method also follows certain steps such as after extracting the data from different data sources, sorting of data using the certain algorithms are performed, decompose the data into a different structured format and finally store the data into another database.

III. Panda functions for Data Formatting and Normalization

The Transforming data stage is about converting the data set into a format that can be analyzed or modelled effectively, and there are several techniques for this process.

a. Data Formatting: Ensuring all data formats are correct (e.g. object, text, floating number, integer, etc.) is another part of this initial 'cleaning' process. If you are working with dates in Pandas, they also need to be stored in the exact format to use special date-time functions.

Functions used for data formatting

Sr. No	Data Frame Function	Description	Output

1.	**df.dtypes**	To check the data type	`df.dtypes`
			`sepal length (cm)  float64`
			`sepal width (cm)  float64`
			`petal length (cm)  float64`
			`petal width (cm)  float64`
			`dtype: object`

2.	**df['petal length (cm)']= df['petal length (cm)'].astype("int**	To change the data type (data type of 'petal length (cm)'changed to int)	

	")		df.dtypes sepal length (cm) float64 sepal width (cm) float64 petal length (cm) int64 petal width (cm) float64 dtype: object

B. Data normalization:

Mapping all the nominal data values onto a uniform scale (e.g. from 0 to 1) is involved in data normalization. Making the ranges consistent
across variables helps with statistical analysis and ensures better comparisons later on.It is also known as Min-Max scaling

Algorithm:

Step 1 : Import pandas and sklearn library for preprocessing

from sklearn import preprocessing

Step 2: Load the iris dataset in dataframe object df

Step 3: Print iris dataset.

df.head()

Step 5: Create a minimum and maximum processor object

min_max_scaler = preprocessing.MinMaxScaler()

Step 6: Separate the feature from the class label

x=df.iloc[:,:4]

Step 6: Create an object to transform the data to fit minmax processor

x_scaled = min_max_scaler.fit_transform(x)

Step 7: Run the normalizer on the dataframe

df_normalized = pd.DataFrame(x_scaled)

Step 8: View the dataframe

df_normalized

IV. Panda Functions for handling categorical variables

- **Categorical variables** have values that **describe a 'quality' or 'characteristic'** of a data unit, like **'what type' or 'which category'.**
- Categorical variables fall into **mutually exclusive (in one category or in another)** and **exhaustive (include all possible options)** categories.

Therefore, categorical variables are qualitative variables and t**end to be represented by a non-numeric value.**

- Categorical features refer **to string type data** and can be easily understood by human beings. But in case of a **machine, it cannot interpret the categorical data directly**. Therefore, the categorical data must be **translated into numerical data that can be understood by machine.**

There are many ways to convert categorical data into numerical data. Here the three most used methods are discussed.

a. **Label Encoding:** Label Encoding refers to **converting the labels into a numeric form** so as to convert them into the machine-readable form. **It is an important preprocessing step for the structured dataset** in supervised learning.

Example : Suppose we have a column Height in some dataset. After applying label encoding, the Height column is converted into:

Height
Tall
Medium
Short

Height
0
1
2

where 0 is the label for tall, 1 is the label for medium, and 2 is a label for short height. **Label Encoding on iris dataset:** For iris dataset the target column which is Species. It contains three species Iris-setosa, Iris-versicolor, Iris-virginica.

Sklearn Functions for Label Encoding:

- **`preprocessing.LabelEncoder :`** It Encode labels with value between 0 and n_classes-1.
- **`fit_transform(y):`**

  `Parameters:` `yarray-like of shape (n_samples,)` **`Target values.`**

  `Returns:` `yarray-like of shape (n_samples,)` **`Encoded labels.`**

This transformer should be used to encode target values, and not the input.

Algorithm:

Step 1 : Import pandas and sklearn library for preprocessing

```
from sklearn import preprocessing
```

Step 2: Load the iris dataset in dataframe object df

Step 3: Observe the unique values for the Species column.

```
df['Species'].unique()
```

output: `array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)`

Step 4: define label_encoder object knows how to understand word labels.

```
label_encoder = preprocessing.LabelEncoder()
```

Step 5: Encode labels in column 'species'.

```
df['Species']= label_encoder.fit_transform(df['Species'])
```

Step 6: Observe the unique values for the Species column.

```
df['Species'].unique()
```

Output: `array([0, 1, 2], dtype=int64)`

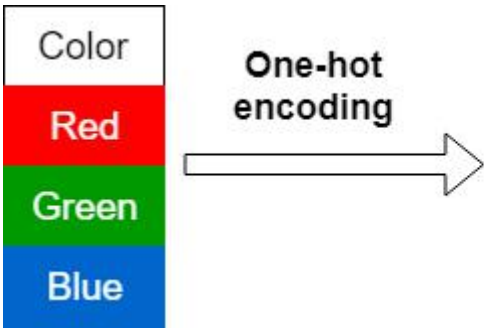
- Use LabelEncoder when there are only two possible values of a categorical feature. For example, features having value such as yes or no. Or, maybe, gender features when there are only two possible values including male or female.

Limitation: Label encoding converts the data in machine-readable form, but it assigns a **unique number(starting from 0) to each class of data**. This may lead to the generation of **priority issues in the data sets**. A label with a high value may be considered to have high priority than a label having a lower value.

b. **One-Hot Encoding:**

In one-hot encoding, we create a new set of dummy (binary) variables that is equal to the number of categories (k) in the variable. For example, let's say we have a categorical variable Color with three categories called "Red", "Green" and "Blue", we need to use three dummy variables to encode this variable using one-hot

encoding. A dummy (binary) variable just takes the value 0 or 1 to indicate the exclusion or inclusion of a category.



d1	d2	d3
1	0	0
0	1	0
0	0	1

In one-hot encoding,

"Red" color is encoded as **[1 0 0]** vector of size 3.

"Green" color is encoded as **[0 1 0]** vector of size 3.

"Blue" color is encoded as **[0 0 1]** vector of size 3.

One-hot encoding on iris dataset: For iris dataset the target column which is Species. It contains three species Iris-setosa, Iris-versicolor, Iris-virginica.

Sklearn Functions for One-hot Encoding:

`sklearn.preprocessing.OneHotEncoder()`: Encode categorical integer features using a one-hot aka one-of-K scheme

Algorithm:

Step 1 : Import pandas and sklearn library for preprocessing

```
from sklearn import preprocessing
```

Step 2: Load the iris dataset in dataframe object df

Step 3: Observe the unique values for the Species column.

```
df['Species'].unique()
```

output: `array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)`

Step 4: Apply label_encoder object for label encoding the Observe the unique values for the Species column.

```
df['Species'].unique()
```

Output: `array([0, 1, 2], dtype=int64)`

Step 5: Remove the target variable from dataset

```
features_df=df.drop(columns=['Species'])
```

Step 6: Apply one_hot encoder for Species column.

```
enc = preprocessing.OneHotEncoder()
enc_df=pd.DataFrame(enc.fit_transform(df[['Species']])).to
array()
```

Step 7: Join the encoded values with Features variable

```
df_encode = features_df.join(enc_df)
```

Step 8: Observe the merge dataframe

```
df_encode
```

Step 9: Rename the newly encoded columns.

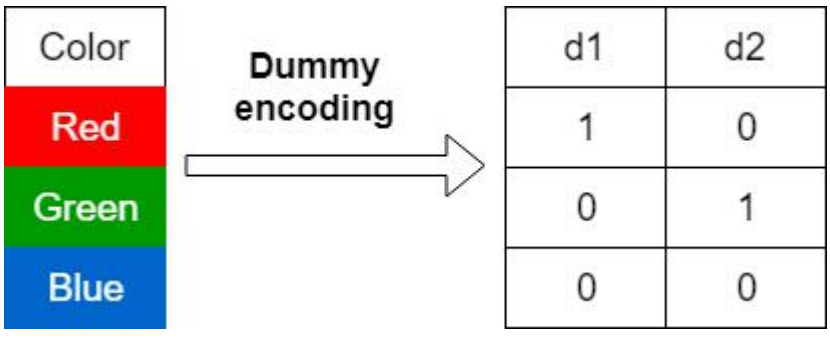
```
df_encode.rename(columns = {0:'Iris-Setosa',
1:'Iris-Versicolor',2:'Iris-virginica'}, inplace = True)
```

Step 10: Observe the merge dataframe

```
df_encode
```

c. Dummy Variable Encoding

Dummy encoding also uses dummy (binary) variables. Instead of creating a number of dummy variables that is equal to the number of categories (k) in the variable, dummy encoding uses k-1 dummy variables. To encode the same Color variable with three categories using the dummy encoding, we need to use only two dummy variables.



In dummy encoding,

"Red" color is encoded as **[1 0]** vector of size 2.

"Green" color is encoded as **[0 1]** vector of size 2. **"Blue"** color is encoded as **[0 0]** vector of size 2.

Dummy encoding removes a duplicate category present in the one-hot encoding.

Pandas Functions for One-hot Encoding with dummy variables:

- **`pandas.get_dummies(data, prefix=None, prefix_sep='_', dummy_na=False, columns=None, sparse=False, drop_first=False, dtype=None):`** Convert categorical variable into dummy/indicator variables.
- **Parameters**:

  data:array-like, Series, or DataFrame Data of which to get dummy indicators.

  prefixstr: list of str, or dict of str, default None String to append DataFrame column names. **prefix_sep**: str, default '_' If appending prefix, separator/delimiter to use. Or pass a list or dictionary as with prefix.

  dummy_nabool: default False Add a column to indicate NaNs, if False NaNs are ignored.

  columns: list:like, default None Column names in the DataFrame to be encoded. If columns is None then all the columns with object or category dtype will be converted.

  sparse: bool: default False Whether the dummy-encoded columns should be backed by a SparseArray (True) or a regular NumPy array (False).

  drop_first:bool, default False Whether to get k-1 dummies out of k categorical levels by removing the first level.

  dtype: dtype, default np.uint8 Data type for new columns. Only a single dtype is allowed.

- **Return :** DataFrame with Dummy-coded data.

Algorithm:

Step 1 : Import pandas and sklearn library for preprocessing

```
from sklearn import preprocessing
```

Step 2: Load the iris dataset in dataframe object df

Step 3: Observe the unique values for the Species column.

```
df['Species'].unique()
```

output: **array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)**

Step 4: Apply label_encoder object for label encoding the Observe the unique values for the Species column.

```
df['Species'].unique()
```

Output: array([0, 1, 2], dtype=int64)

Step 6: Apply one_hot encoder with dummy variables for Species column.

```
one_hot_df = pd.get_dummies(df,
prefix="Species", columns=['Species'],
drop_first=True)
```

Step 7: Observe the merge dataframe

```
one_hot_df
```

Conclusion:

ASSIGNMENT NO.03

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment: Descriptive Statistics - Measures of Central Tendency and variability

Perform the following operations on any open source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.
2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris- versicolor' of iris.csv dataset.

Objective of the Assignment: Students should be able to perform the Descriptive Statistics operation using Python on any open source dataset

Prerequisite:

1. Basic of Python Programming
1. Concept of measures of Central Tendency and variability

Theory:

I. Data Analysis

With data analysis, we use two main statistical methods- *Descriptive* and *Inferential*.

- Descriptive statistics uses tools like mean and standard deviation on a sample to summarize data.
- Inferential statistics, on the other hand, looks at data that can randomly vary, and then draw conclusions from it.

A. Python Descriptive Statistics – Central Tendency in Python

import the *Python statistics* module.(>>>import statistics as st)

a. mean()

This function returns the arithmetic average of the data it operates on. If called on an empty container of data, it raises a StatisticsError.

b. mode()

This function returns the most common value in a set of data. This gives us a great idea of where the center lies.

c. median()

For data of odd length, this returns the middle item; for that of even length, it returns the average of the two middle items.

d. harmonic_mean()

This function returns the harmonic mean of the data. For three values a, b, and c, the harmonic mean is-
3/(1/a + 1/b +1/c)
It is a measure of the center; one such example would be speed

e. median_low()

When the data is of an even length, this provides us the low median of the data. Otherwise, it returns the middle value.

f. median_high()

Like median_low, this returns the high median when the data is of an even length. Otherwise, it returns the middle value.

g. median_grouped()

This function uses interpolation to return the median of grouped continuous data. This is the 50^{th} percentile.

B. Python Descriptive Statistics – Dispersion in Python

Dispersion/spread gives us an idea of how the data strays from the typical value

.

a. variance()

This returns the variance of the sample. This is the second moment about the mean and a larger value denotes a rather spread-out set of data. You can use this when your data is a sample out of a population.

b. pvariance()

This returns the population variance of data. Use this to calculate variance from an entire population.

c. stdev()

This returns the standard deviation for the sample. This is equal to the square root of the sample variance.

d. pstdev()

This returns the population standard deviation. This is the square root of population variance.

e.Interquartile Range (IQR)

The Interquartile Range (IQR) is a measure of statistical dispersion, and is calculated as the difference between the upper quartile (75th percentile) and the lower quartile (25th percentile). The IQR is also a very important measure for identifying outliers and could be visualized using a boxplot.

II. pandas with Descriptive Statistics in Python

We can do the same things using pandas too-

```
>>> import pandas as pd
nums=[1,2,3,5,7,9,7,2,7,6]
>>> df=pd.DataFrame(nums)
>>> df.mean()

        0 4.9
dtype: float64

>>> df.mode()

        0 7

>>> df.std() #Standard deviation

        0 2.726414
dtype: float64
```

```
>>> df.skew()

        0 -0.115956 #The distribution is symmetric
dtype: float64
```

A value less than -1 is skewed to the left; that greater than 1 is skewed to the right. A value between -1 and 1 is symmetric.

Conclusion:

ASSIGNMENT NO.04

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment:

Data Analytics I

Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset.

Objective of the Assignment: The objective is to predict the value of prices of the house using the given features

Prerequisite:

1. Basic of Python Programming
1. Concept of Linear Regression Model

Theory:

What Is Regression?

Regression searches for relationships among variables.

For example, you can observe several employees of some company and try to understand how their salaries depend on the **features**, such as experience, level of education, role, city they work in, and so on.

This is a regression problem where data related to each employee represent one **observation**. The presumption is that the experience, education, role, and city are the independent features, while the salary depends on them.

Similarly, you can try to establish a mathematical dependence of the prices of houses on their areas, numbers of bedrooms, distances to the city center, and so on.

In other words, **you need to find a function that maps some features or variables to others sufficiently well**.

The dependent features are called the **dependent variables**, **outputs**, or **responses**.

The independent features are called the **independent variables**, **inputs**, or **predictors**.

Regression problems usually have one continuous and unbounded dependent variable. The inputs, however, can be continuous, discrete, or even categorical data such as gender, nationality, brand, and so on.

When Do You Need Regression

Typically, you need regression to answer whether and how some phenomenon influences the other or **how several variables are related**. For example, you can use it to determine if and to what extent the experience or gender impact salaries.

Regression is also useful when you want **to forecast a response** using a new set of predictors. For example, you could try to predict electricity consumption of a household for the next hour given the outdoor temperature, time of day, and number of residents in that household.

Regression is used in many different fields: economy, computer science, social sciences, and so on. Its importance rises every day with the availability of large amounts of data and increased awareness of the practical value of data.

Linear Regression

Linear regression is probably one of the most important and widely used regression techniques. It's among the simplest regression methods. One of its main advantages is the ease of interpreting results.

Machine Learning (ML): ML is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn themselves and improve from the experience without being explicitly programmed. ML focuses on the development of computer programs that can access data and use it to learn themselves.

Data Set: A collection of related sets of information that is composed of separate elements but can be manipulated as a unit by a computer.

Data Visualisation: It is a representation of data or information in a graph, chart, or other visual formats which is helpful to conduct analyses such as predictive analysis which can serve as helpful Visualisation to present.

Data Cleaning: It is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

Supervised Learning: The model is trained using 'labeled data'. Datasets are said to contain labels that contain both input and output parameters. To simplify – 'Data is already tagged with the correct answer'.

Simple Linear Regression: It is a Regression Model that estimates the relationship between the independent variable and the dependent variable using a straight line [y = mx + c], where both the variables should be quantitative.

Models: Those are output by algorithms and are comprised of model data and a prediction algorithm.

Training Model: In supervised learning, an ML Algorithm builds a model by examining many examples and attempting to find a model that minimizes loss and improves prediction accuracy.

The steps involved are:

1. Importing the dataset.
2. Visualising the Data
3. Data Cleaning
4. Build the Model and Train it
5. Make Predictions on Unseen Data

Simple vs. Multiple Regression

- One dependent variable Y predicted from one independent variable X
- One regression coefficient
- $\mathbf{r^2}$: proportion of variation in dependent variable Y predictable from X

- One dependent variable Y predicted from a set of independent variables (X1, X2 ….Xk)
- One regression coefficient for each independent variable
- $\mathbf{R^2}$: proportion of variation in dependent variable Y predictable by set of independent variables (X's)

Conclusion:

ASSIGNMENT NO.05

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment:

Data Analytics II

1. Implement logistic regression using Python/R to perform classification on Social_
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision,

Objective of the Assignment: Students will learn how confusion matrices can be scaled up to include more than two classification classes and finally got hands-on experience with confusion matrices by implementing them in Python.

Prerequisite:

1. Basic of Python Programming
1. Concept of logistic regression

Theory:

In machine learning, Classification is used to split data into categories. But after cleaning and preprocessing the data and training our model, how do we know if our classification model performs well? That is where a confusion matrix comes into the picture.

A confusion matrix is used to measure the performance of a classifier in depth. In this simple guide to Confusion Matrix, we will get to understand and learn confusion matrices better.

What Are Confusion Matrices, and Why Do We Need Them?

Classification Models have multiple categorical outputs. Most error measures will calculate the total error in our model, but we cannot find individual instances of errors in our model. The model might misclassify some categories more than others, but we cannot see this using a standard accuracy measure.

Furthermore, suppose there is a significant class imbalance in the given data. In that case, i.e., a class has more instances of data than the other classes, a model might predict the majority class

for all cases and have a high accuracy score; when it is not predicting the minority classes. This is where confusion matrices are useful.

A confusion matrix presents a table layout of the different outcomes of the prediction and results of a classification problem and helps visualize its outcomes.

It plots a table of all the predicted and actual values of a classifier.

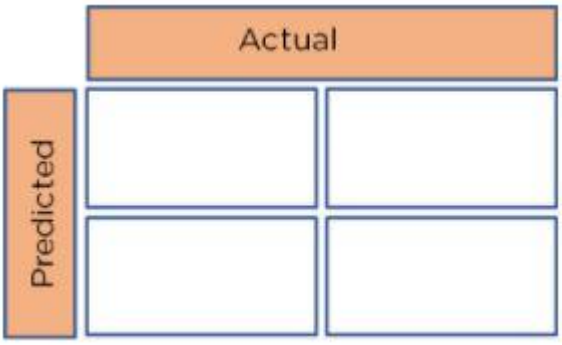


Figure 1: Basic layout of a Confusion Matrix

How to Create a 2x2 Confusion Matrix?

We can obtain four different combinations from the predicted and actual values of a classifier:

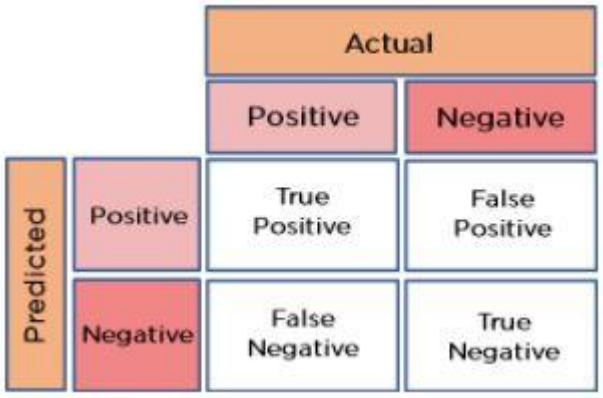


Figure 2: Confusion Matrix

- True Positive: The number of times our actual positive values are equal to the predicted positive. You predicted a positive value, and it is correct.

- False Positive: The number of times our model wrongly predicts negative values as positives. You predicted a negative value, and it is actually positive.
- True Negative: The number of times our actual negative values are equal to predicted negative values. You predicted a negative value, and it is actually negative.
- False Negative: The number of times our model wrongly predicts negative values as positives. You predicted a negative value, and it is actually positive.

Confusion Matrix Metrics

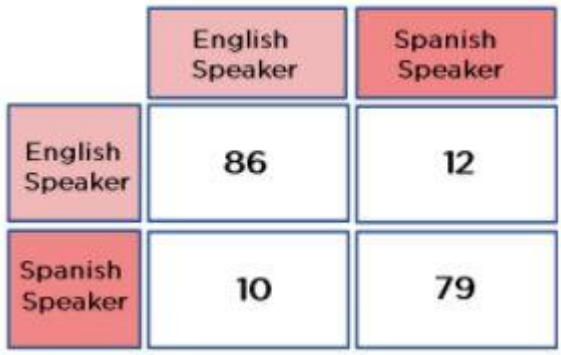


Figure 3: Confusion Matrix for a classifier

Consider a confusion matrix made for a classifier that classifies people based on whether they speak English or Spanish.

From the above diagram, we can see that:

True Positives (TP) = 86

True Negatives (TN) = 79

False Positives (FP) = 12

False Negatives (FN) = 10

- Accuracy: The accuracy is used to find the portion of correctly classified values. It tells us how often our classifier is right. It is the sum of all true values divided by total values.

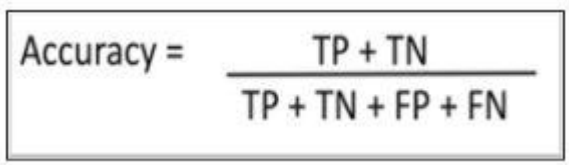


Figure 4: Accuracy

In this case:

Accuracy = (86 +79) / (86 + 79 + 12 + 10) = 0.8823 = 88.23%

- Precision: Precision is used to calculate the model's ability to classify positive values correctly. It is the true positives divided by the total number of predicted positive values.

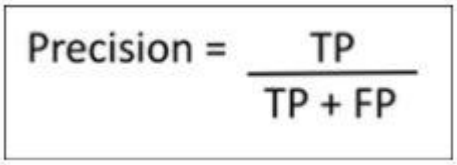


Figure 5: Precision

In this case,

Precision = 86 / (86 + 12) = 0.8775 = 87.75%

- Recall: It is used to calculate the model's ability to predict positive values. "How often does the model predict the correct positive values?". It is the true positives divided by the total number of actual positive values.

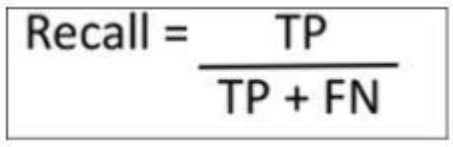


Figure 6: Recall

In this case,

Recall = 86 / (86 + 10) = 0.8983 = 89.83%

- F1-Score: It is the harmonic mean of Recall and Precision. It is useful when you need to take both Precision and Recall into account.

$$\text{F1-Score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 7: F1-Score

In this case,

F1-Score = (2* 0.8775 * 0.8983) / (0.8775 + 0.8983) = 0.8877 = 88.77%Scaling a Confusion Matrix

To scale a confusion matrix, increase the number of rows and columns. All the True Positives will be along the diagonal. The other values will be False Positives or False Negatives.

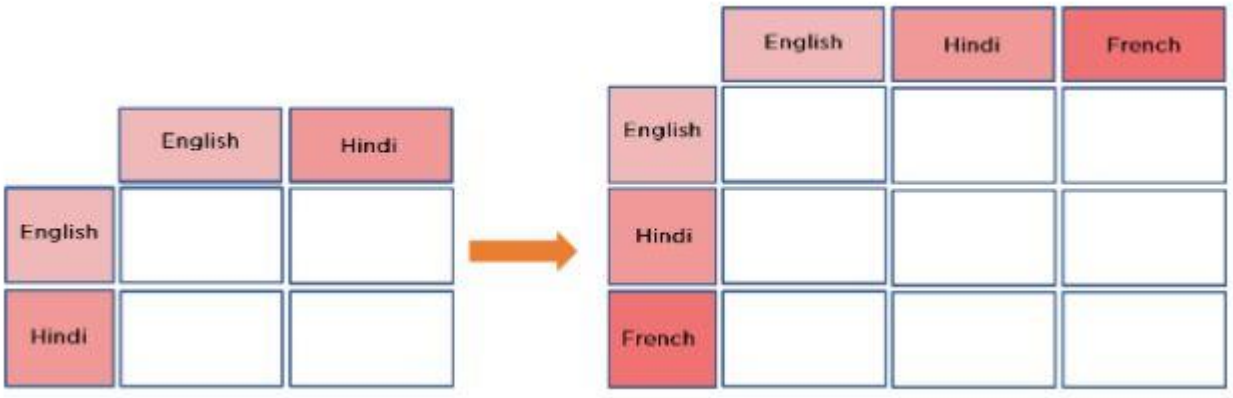


Figure 12: Scaling down our dataset

Conclusion:

ASSIGNMENT NO.06

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment: **Data Analytics III**

1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision,

---

Objective of the Assignment: Students should be able to Implement

Simple Naïve Bayes classification algorithm

Prerequisite:

1. Basic of Python Programming

1. Concept of Naïve Bayes classification algorithm

Theory:

Naive Bayes Classifier in Python

In machine learning, Naïve Bayes classification is a straightforward and powerful algorithm for the classification task.
Naïve Bayes classification is based on applying Bayes' theorem with strong independence assumption between the features. Naïve Bayes classification produces good results when we use it for textual data analysis such as Natural Language Processing.

Naïve Bayes models are also known as simple Bayes or independent Bayes. All these names refer to the application of Bayes' theorem in the classifier's decision rule. Naïve Bayes classifier applies the Bayes' theorem in practice. This classifier brings the power of Bayes' theorem to machine learning.

Naive Bayes algorithm intuition

Naïve Bayes Classifier uses the Bayes' theorem to predict membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as the **Maximum A Posteriori (MAP)**.

The **MAP for a hypothesis with 2 events A and B is**

MAP (A)

= max (P (A | B))

= max (P (B | A) * P (A))/P (B)

= max (P (B | A) * P (A))

Here, P (B) is evidence probability. It is used to normalize the result. It remains the same, So, removing it would not affect the result.

Naïve Bayes Classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.

In real world datasets, we test a hypothesis given multiple evidence on features. So, the calculations become quite complicated. To simplify the work, the feature independence approach is used to uncouple multiple evidence and treat each as an independent one.

Types of Naive Bayes algorithm

There are 3 types of Naïve Bayes algorithm. The 3 types are listed below:-

1. Gaussian Naïve Bayes
2. Multinomial Naïve Bayes
3. Bernoulli Naïve Bayes

Gaussian Naïve Bayes algorithm

When we have continuous attribute values, we made an assumption that the values associated with each class are distributed according to Gaussian or Normal distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let μi be the mean of the values and let σi be the variance of the values associated with the ith class. Suppose we have some observation value xi . Then, the probability distribution of xi given a class can be computed by the following equation –

$$p(x_i|y_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_i-\mu_j)^2}{2\sigma_j^2}}$$

Multinomial Naïve Bayes algorithm

With a Multinomial Naïve Bayes model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p1, . . . ,pn) where pi is the probability that event i occurs. Multinomial Naïve Bayes algorithm is preferred to use on data that is multinomially distributed. It is one of the standard algorithms which is used in text categorization classification.

Bernoulli Naïve Bayes algorithm

In the multivariate Bernoulli event model, features are independent boolean variables (binary variables) describing inputs. Just like the multinomial model, this model is also popular for document classification tasks where binary term occurrence features are used rather than term frequencies.

Applications of Naive Bayes algorithm

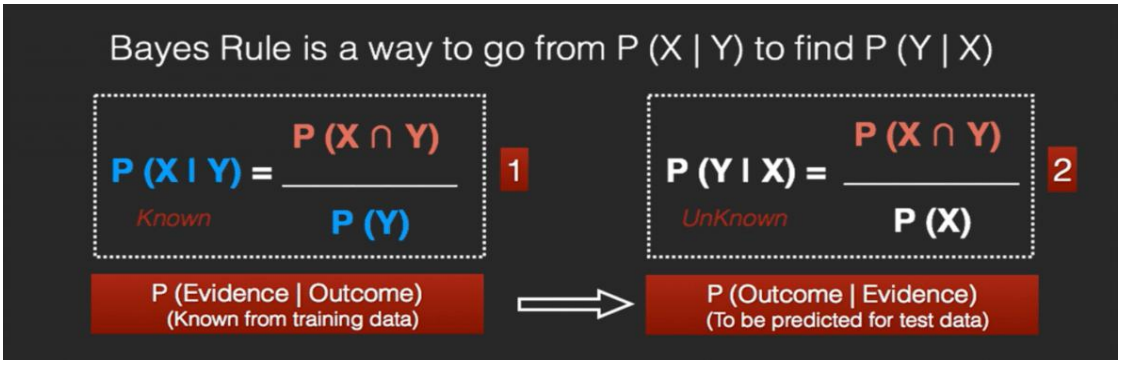
Naïve Bayes is one of the most straightforward and fast classification algorithm. It is very well suited for large volume of data. It is successfully used in various applications such as :

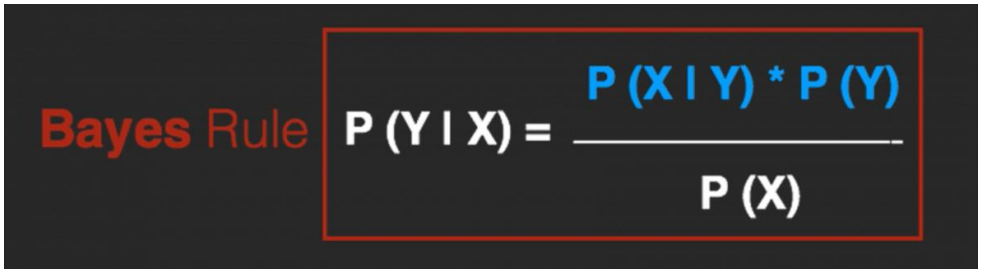
1. Spam filtering
2. Text classification
3. Sentiment analysis
4. Recommender systems

It uses Bayes theorem of probability for prediction of unknown class.

The Bayes Rule

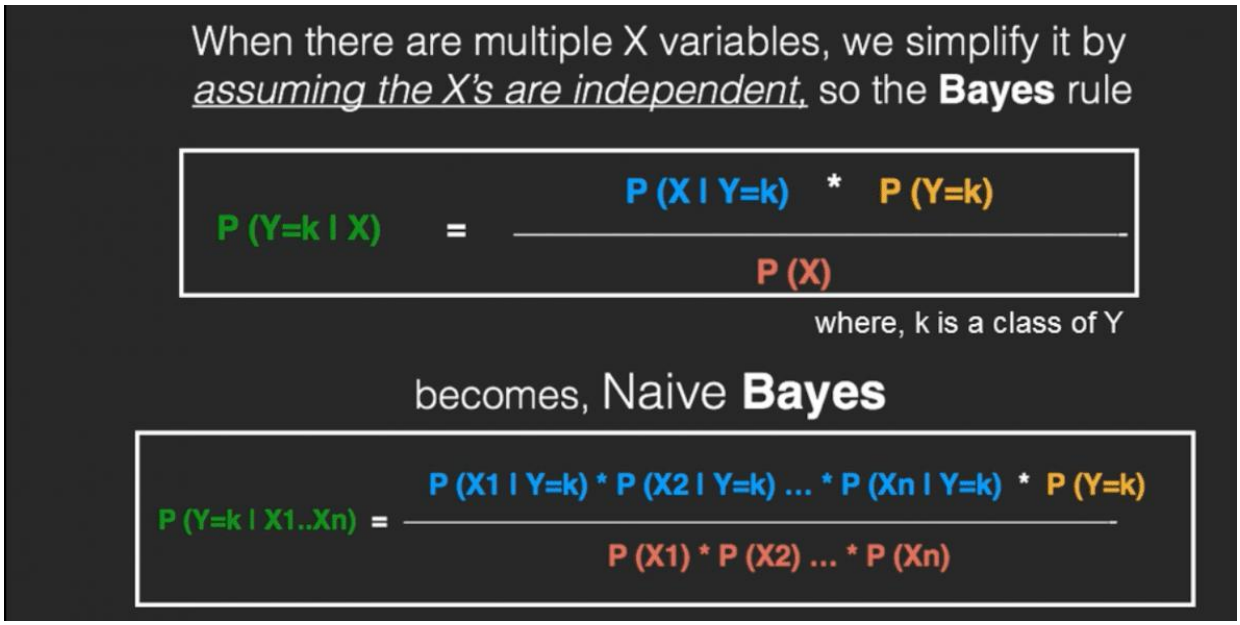
The Bayes Rule is a way of going from P(X|Y), known from the training dataset, to find P(Y|X). To do this, we replace A and B in the above formula, with the feature X and response Y. For observations in test or scoring data, the X would be known while Y is unknown. And for each row of the test dataset, you want to compute the probability of Y given the X has already happened. What happens if Y has more than 2 categories? we compute the probability of each class of Y and let the highest win.

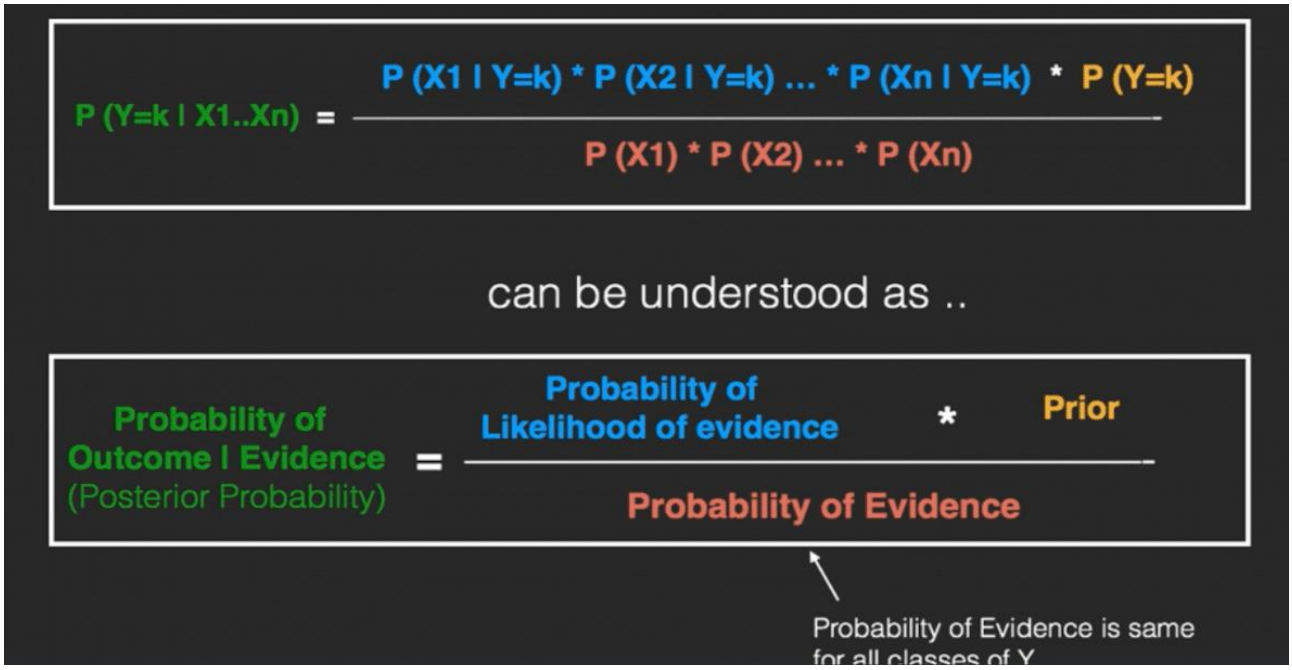




The Naive Bayes

The Bayes Rule provides the formula for the probability of Y given X. But, in real-world problems, you typically have multiple X variables. When the features are independent, we can extend the Bayes Rule to what is called Naive Bayes. It is called 'Naive' because of the naive assumption that the X's are independent of each other. Regardless of its name, it's a powerful formula.




P (Y=k | X1..Xn) = P (X1 | Y=k) * P (X2 | Y=k) ... * P (Xn | Y=k) * P (Y=k) / P (X1) * P (X2) ... * P (Xn)
can be understood as ..
Probability of Outcome | Evidence (Posterior Probability) = Probability of Likelihood of evidence * Prior / Probability of Evidence
Probability of Evidence is same for all classes of Y

Conclusion:

ASSIGNMENT NO.07

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment: **Text Analytics**

1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.
2. Create representation of document by calculating Term Frequency and Inverse Document Frequency.

Objective of the Assignment: Students should be able to Extract Sample document and apply following document preprocessing methods

Prerequisite:

1. Basic of Python Programming

2.Concept of preprocessing methods

Theory:

I. Tokenization:

In processing unstructured text, tokenization is the step by which the character string in a text segment is turned into units - tokens - for further analysis. Ideally, those tokens would be words, but numbers and other characters can also count as tokens. A big challenge in tokenization is determining delimiters that separate tokens. Delimiters could be white space, commas, periods, html tags, etc., and they might not always be delimiters. After the text is broken into tokens, a list of "types," or unique tokens, is created. In the previous sentence, the token "is" appears twice, but there is just a single "is" type.

For example, consider the sentence: “Never give up”.

The most common way of forming tokens is based on space. Assuming space as a delimiter, the tokenization of the sentence results in 3 tokens – Never-give-up. As each token is a word, it becomes an example of Word tokenization.

Similarly, tokens can be either characters or subwords. For example, let us consider “smarter”:

1. Character tokens: s-m-a-r-t-e-r
2. Subword tokens: smart-er

II. Parts Of Speech Tags

Parts of speech tags are the properties of the words, which define their main context, functions, and usage in a sentence. Some of the commonly used parts of speech tags are

Nouns: Which defines any object or entity

Verbs: That defines some action.

Adjectives and Adverbs: This acts as a modifier, quantifier, or intensifier in any sentence.

In a sentence, every word will be associated with a proper part of the speech tag. For example**,** consider the sentence below

In this sentence, every word is associated with a part of the speech tag which defines their functions. Here, David has an NNP tag which means it is a proper noun. Further, Has and purchased belong to the verb indicating that they are the actions. The Laptop and Apple store are the nouns. New is the adjective whose role is to modify the context of the laptop.

Parts of speech tags are defined by the relationship of words with the other words in the sentence.

We can apply machine learning models and rule-based models to obtain the parts of speech tags of a word. The most commonly used parts of speech tag notations are provided by the Penn Treebank corpus. In which, a total of 48 P.O.S tags are defined according to their usage.

Use cases of POS Tags

Parts of speech tags have a large number of applications and they are used in a variety of tasks such as

- Text Cleaning
- Feature Engineering tasks
- Word sense disambiguation

III. Why do we Need to Remove Stopwords?

- On removing stopwords, dataset size decreases and the time to train the model also decreases
- Removing stopwords can potentially help improve the performance as there are fewer and only meaningful tokens left. Thus, it could increase classification accuracy
- Even search engines like Google remove stopwords for fast and relevant retrieval of data from the database

When Should we Remove Stopwords?

We can remove stopwords while performing the following tasks:

- Text Classification
  - Spam Filtering
  - Language Classification
  - Genre Classification
- Caption Generation
- Auto-Tag Generation

Avoid Stopword Removal

- Machine Translation
- Language Modeling
- Text Summarization
- Question-Answering problems

IV. Introduction to Stemming

Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”. Stemming is an important part of the pipelining process in Natural language processing. The input to the stemmer is tokenized words.

Errors in Stemming:

There are mainly two errors in stemming –

- over-stemming
- under-stemming

Over-stemming occurs when two words are stemmed from the same root that are of different stems. Over-stemming can also be regarded as false-positives.

Under-stemming occurs when two words are stemmed from the same root that are not of different stems. Under-stemming can be interpreted as false-negatives.

Applications of stemming :

1. Stemming is used in information retrieval systems like search engines.
2. It is used to determine domain vocabularies in domain analysis.

Stemming algorithms are
Porter's Stemmer algorithm
Lovins Stemmer
Dawson Stemmer
Krovetz Stemmer
Xerox Stemmer
N-**Gram Stemmer**
Snowball Stemmer:
Lancaster Stemmer:

V. Lemmatization

In contrast to stemming, lemmatization looks beyond word reduction and considers a language's full vocabulary to apply a morphological analysis to words. The lemma of 'was' is 'be' and the lemma of 'mice' is 'mouse'.

Lemmatization is typically seen as much more informative than simple stemming, which is why Spacy has opted to only have Lemmatization available instead of Stemming

Lemmatization looks at surrounding text to determine a given word's part of speech, it does not categorize phrases.

lemmatization is that it is harder to create a lemmatizer in a new language than it is a stemming algorithm because we require a lot more knowledge about structure of a language in lemmatizers.

Stemming and Lemmatization both generate the foundation sort of the inflected words and therefore the only difference is that stem may not be an actual word whereas,

lemma is an actual language word.

Stemming follows an algorithm with steps to perform on the words which makes it faster. Whereas, in lemmatization, you used a corpus also to supply lemma which makes it slower than stemming. you furthermore might had to define a parts-of-speech to get the proper lemma.

The above points show that if speed is concentrated then stemming should be used since lemmatizers scan a corpus which consumes time and processing. It depends on the problem you're working on that decides if stemmers should be used or lemmatizers.

Conclusion:

ASSIGNMENT NO.08

Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:.....…

Actual Date of Completion:.....……

Title of the Assignment: **Data Visualization I**

1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.

---

2. 2.Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

Objective of the Assignment: Students should be able to view Data Visualization

Prerequisite:

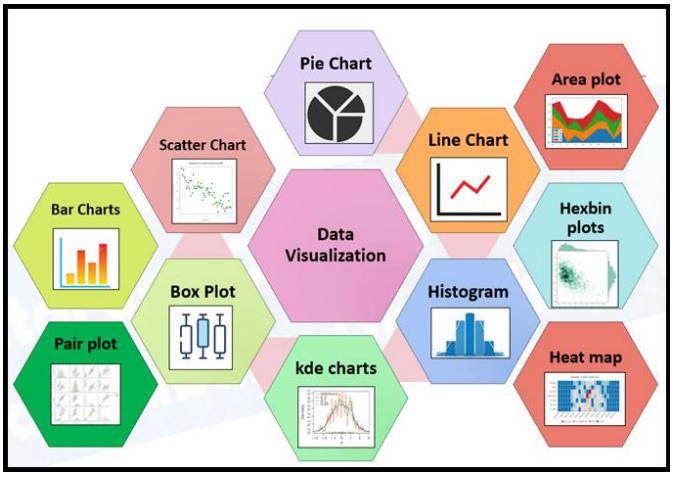
1. Basic of Python Programming

2.Concept of Data Visualization

Theory:

1. Data Visualization

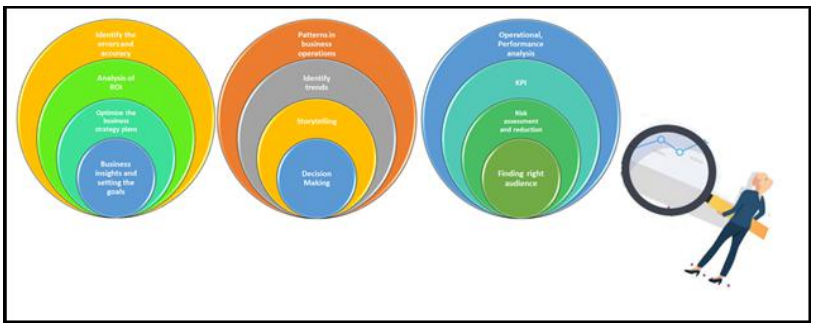
Data Visualization techniques involve the generation of graphical or pictorial representation of DATA, form which leads you to understand the insight of a given data set. This visualisation technique aims to identify the Patterns, Trends, Correlations, and Outliers of data sets.



2. Benefits of Data Visualization

- **Patterns in business operations:** Data visualization techniques help us to determine the patterns of business operations. By understanding the problem statement and identifying the solutions in terms of pattering and applied to eliminate one or more of the inherent problems.
- **Identify business trends and relate to data:** These techniques help us identify market trends by collecting the data on Day-To-Day business activities and preparing trend reports, which helps track the business how influences the market. So that we could understand the competitors and customers. Certainly, this helps to long-term perspective.
- **Storytelling and Decision making:** Knowledge of storytelling from available data is one of the niche skills for business communication, specifically for the Data Science domain which is playing a vital role. Using best visualization this role can be enhanced much better way and reaching the objectives of business problems.
- **Understand the current business insights and setting the goals:** Businesses can understand the insight

- of the business KPIs, finding tangible goals and business strategy plannings, therefore they could optimize the data for business strategy plans for ongoing activities.
- **Operational and Performance analysis:**
- **Increase the productivity of the manufacturing unit:** With the help of visualization techniques the clarity of KPIs depicting the trends of the productivity of the manufacturing unit, and guiding were to improve the productivity of the plant.



3. Data Visualization in Data Science

Data visualization techniques most important part of Data Science, There won't be any doubt about it. And even in the Data Analytics space as well the Data visualization doing a major role. We will discuss this in detail with help of Python

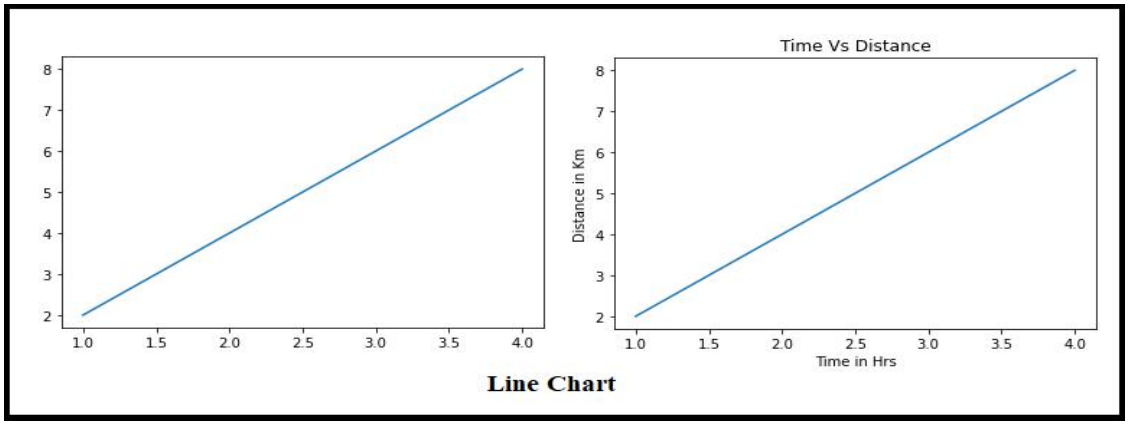
packages and how it helps during the Data Science process flow. This is a very interesting topic for every Data Scientist and Data Analyst.

I. Line Chart

Line Chart is a simple data visualization in Python, which is available under **Matplotlib**.

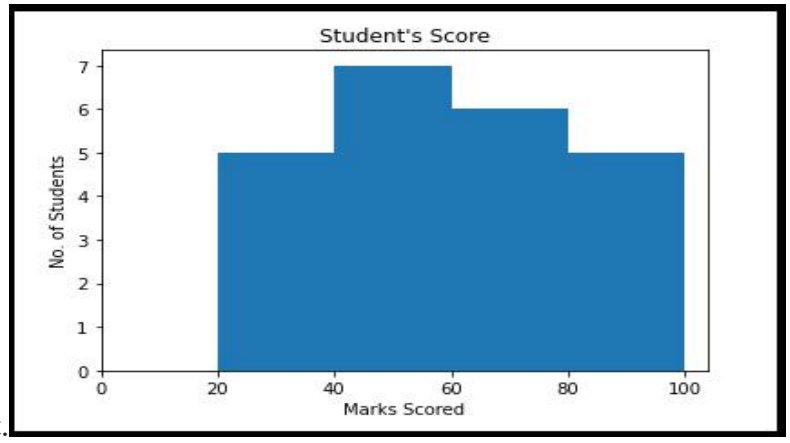
Line charts are used to represent the relation between two data X and Y on the respective axis.

Line Chart always a linear relationship between X and Y axis, we observe that in the above picture.



II.Histogram

The histogram is the graphical representation of a set of numerical data distribution across. It is a kind of bar plot with X-axis and Y-axis represents the bin ranges and frequency respectively. How to read or represent this chart.

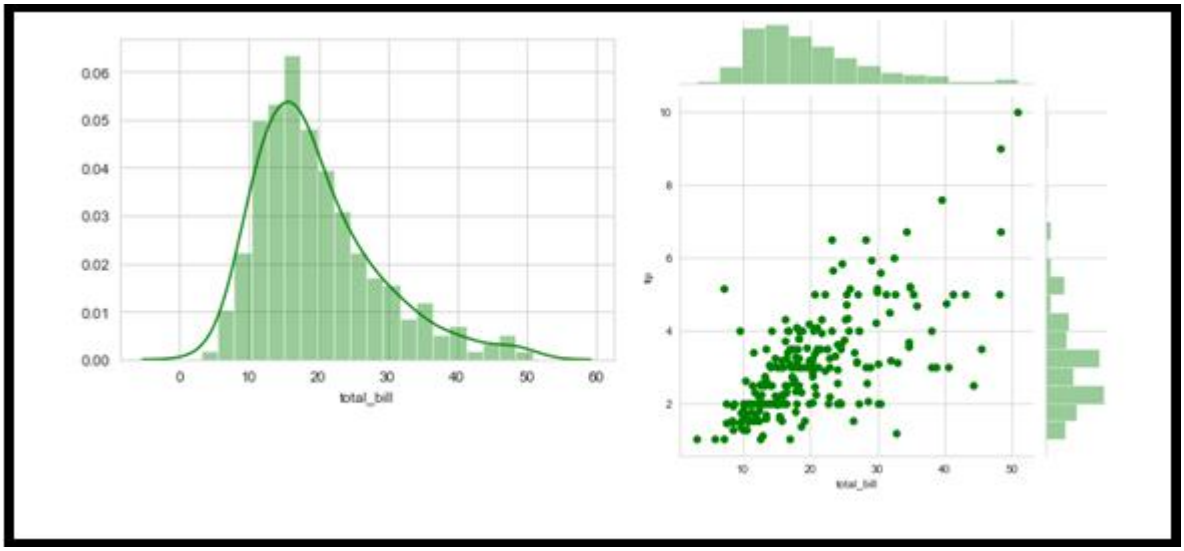


Characteristics Of Histogram

- The **Histogram** is used to get any unusual observations in the give en dataset.
- Measured on an interval scale of given numerical values with several data bins.
- The Y-axis represents the number of % of occurrences in the data
- The X-axis represents data distributions.

Displot – This is similar to the histogram in the graphical, but with additional features. And bringing **Kernel Density Estimation (KDE).**

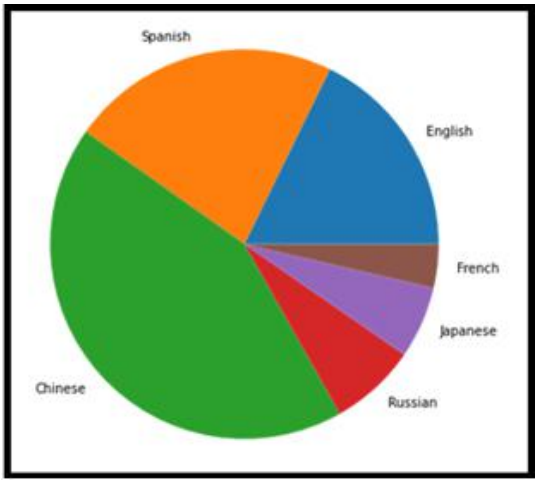
Jointplot – A combination of scattering and histogram.



III.Pie Chart

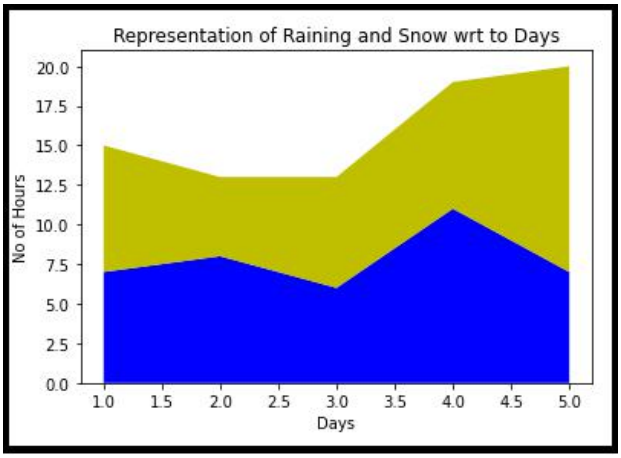
This is a very familiar chart and representation statistical plot in the form of circular from series of data. This is commonly used in business presentations to represent Order, Sales, Profit, Loss, etc., It consists of slices of data part in the collection of the same set and character-wise differentiation. Each of the slices of pie is called a wedge with values of different sizes.

This chart is widely used to represent the composition collection**.** Perfect for the categorical data type.



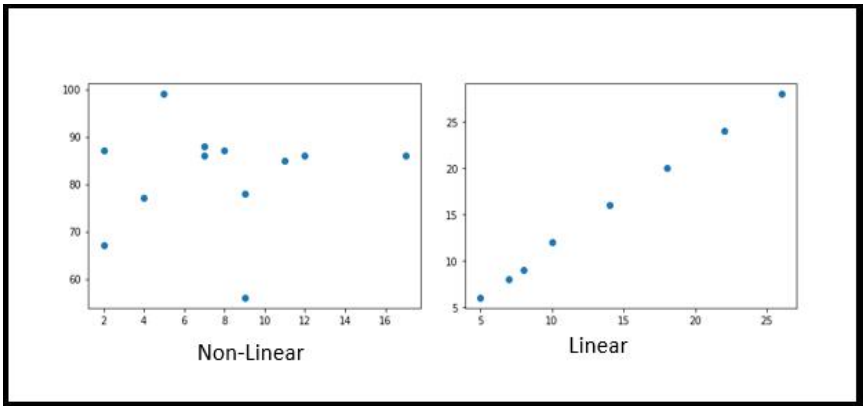
IV. Area plot

This is very similar to a line chart with fencing surrounded by a boundary line of different colours. Simple representation of the evolution of a numeric variable.



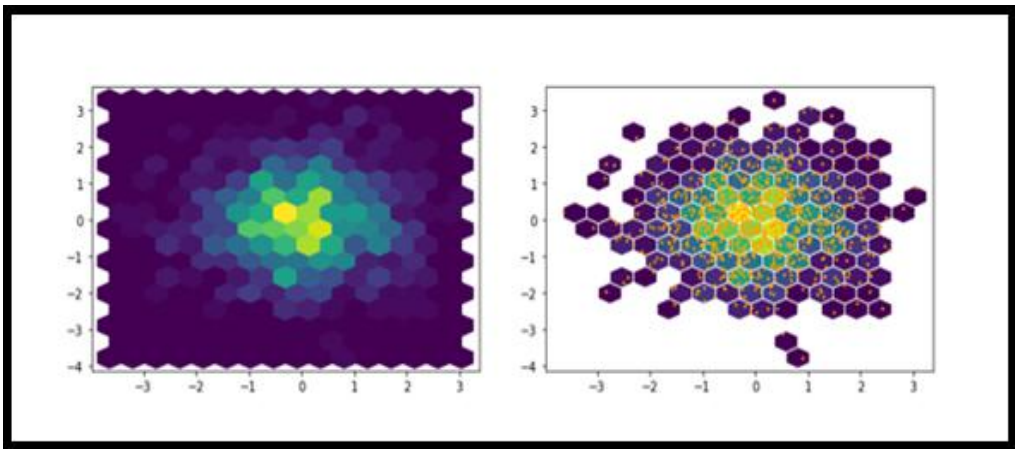
V. Scatter plots

Scatter plots are used to plot data points across both axes (Horizontal and Vertical) and represent how each axis correlated with each other. Mostly in Data Science/Machine Learning implementation and before the EDA process, generally we should analyse how dependent and independent aligned. It could positive or Negative or sometimes be scattered across the graph.



VI. Hexbins plots

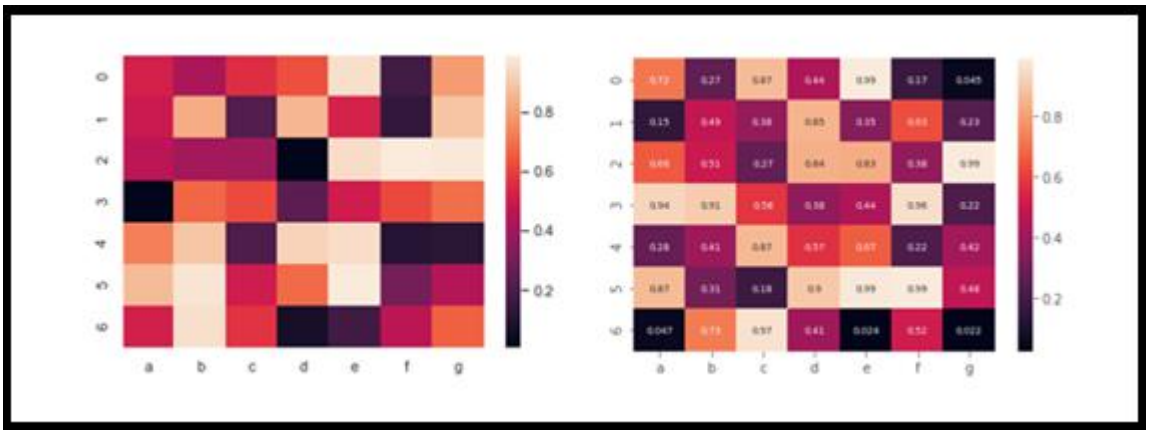
The objective of Hexbins is used to group the two sets of numeric values. Hexbins helps to improve the visualization of the scatter plots. Because for a larger dataset, a scatter plot makes a confused smattering of points. We can improve this with Hexbins. It provides two modes of representations 1.List of Coordinates 2.Geospatial Object.



VII. Heatmap

A **heatmap** is one of my favorite visualization techniques among the other charts. basically, a set of variable correlations is represented by various shades of the same color. Usually, the darker shades of the chart represent the higher correlations values than the lighter shade. this map would help Data Scientists to figure out how to target variable is correlated with other dependent variables in the given data set. Less correlated variables can be removed for further analysis, we could say this helps us

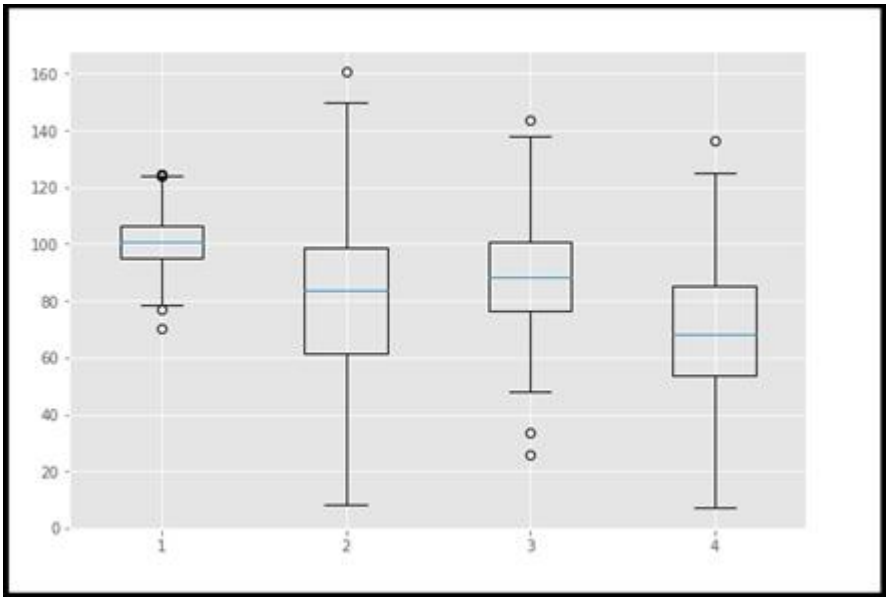
during the feature selection process. Later grouping them under X, Y as our target and followed by test and train split.



VIII. Boxplot

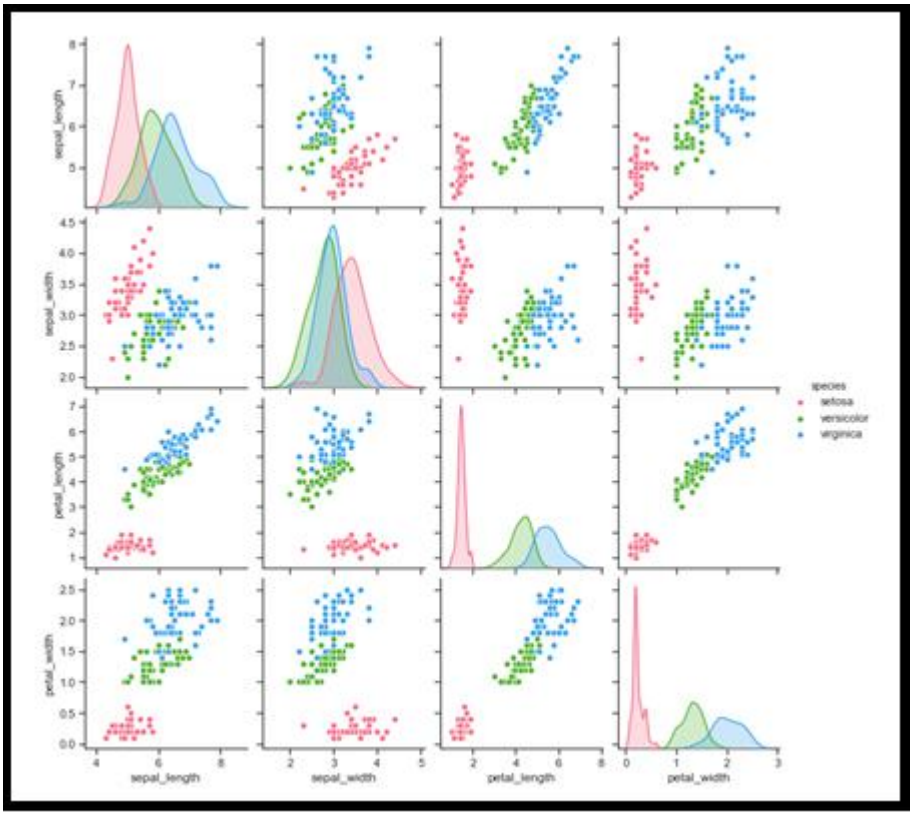
A Boxplot is a type of chart often used in the Data Science life cycle, especially during Explanatory Data Analysis (EDA). Which represents the distribution of data in the form of quartiles or percentiles. Q1 represents the first quartile (25th percentile), Q2 is the second quartile (50th percentile/median), Q3 represents the third quartile (Q3) and Q4 represents the fourth quartile or the largest value.

Using this plot we could identify the outliers very quickly and easily. This is a very effective plot all among the plots. So after the removal of outliers, the data set needs to undergo some sort of statistical test and fine-tune for further analysis.



IX. Pairplot

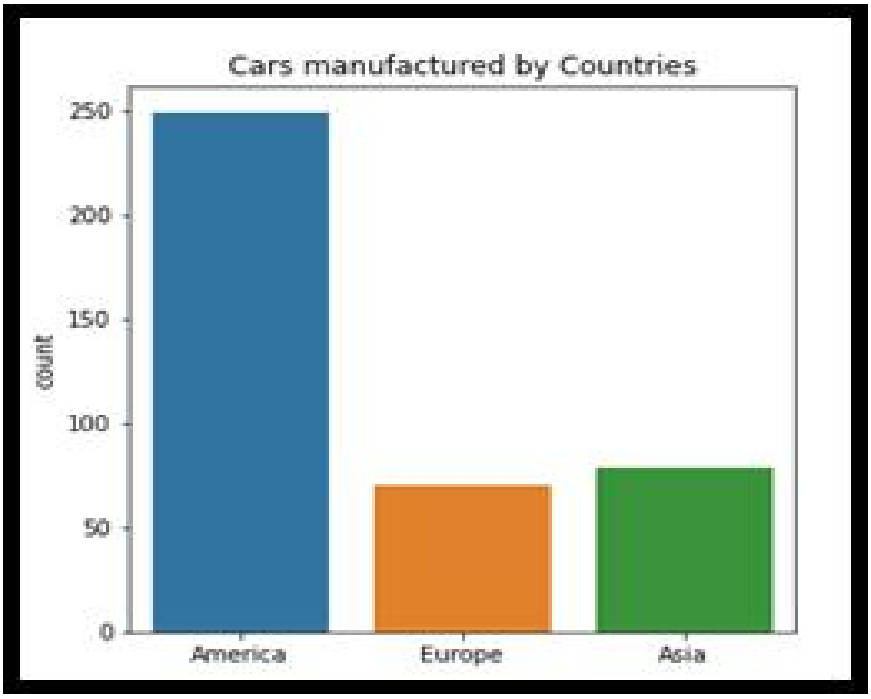
A pairplot is another important plot in the Data Science life cycle during the EDA process, to analyse how features are related to each other, in the form of grid-based miniature graphical representation along the X and Y axis, either positive correlated or negatively correlated. So obviously we could eliminate the negatively correlated, by considering positively corrected pairs and move for further analysis. This is very similar to Heat Map, but here we could see the relationship with our naked eyes. That is special over here. Hope you could fee this. Again this is best for doing the feature selection process.



Line Chart is always linear relation between X and Y axis, we observe that above picture

X. Bar Chart

A bar chart or bar graph is generally a very familiar chart to presents categorical data with rectangular bars. It can be plotted either way horizontally or vertically. this chart would represent the impact of the individual's category on the given dataset. First of first look. In the below chart "America" is much more impact than "Europe" and "Asia". This would derive some observation on the dataset and focus on the problem statement.


Cars manufactured by Countries
count
250
200
150
100
50
0
America
Europe
Asia

Conclusion:

ASSIGNMENT NO.09

Title:

Write a case study on Global Innovation Network and Analysis (GINA). Components of analytic plan are 1. Discovery business problem framed, 2. Data, 3. Model planning analytic technique and 4. Results and Key findings.

Case Study: Global Innovation Network and Analysis (GINA) :-

EMC's Global Innovation Network and Analytics (GINA) team is a group of senior technologists located in centers of excellence (COEs) around the world. This team's charter is to engage employees across global COEs to drive innovation, research, and university partnerships. In 2012, a newly hired director wanted to improve these activities and provide a mechanism to track and analyze the related information. In addition, this team wanted to create more robust mechanisms for capturing the results of its informal conversations with other thought leaders within EMC, in academia, or in other organizations, which could later be mined for insights. The GINA team thought its approach would provide a means to share ideas globally and increase knowledge sharing among GINA members who may be separated geographically. It planned to create a data repository containing both structured and unstructured data to accomplish three main goals.

- **Store formal and informal data.**
- **Track research from global technologists.**
- **Mine the data for patterns and insights to improve the team's operations and strategy.**

The GINA case study provides an example of how a team applied the Data Analytics Lifecycle to analyze innovation data at EMC. Innovation is typically a difficult concept to measure, and this team wanted to look for ways to use advanced analytical methods to identify key innovators within the company.

2.8.1 Phase 1: Discovery:-

In the GINA project's discovery phase, the team began identifying data sources. Although GINA was a group of technologists skilled in many different aspects of engineering, it had some data and ideas about what it wanted to explore but lacked a formal team that could perform these analytics. After consulting with various experts including Tom Davenport, a noted expert in analytics at Babson College, and Peter Gloor, an expert in collective

intelligence and creator of CoIN (Collaborative Innovation Networks) at MIT, the team decided to crowdsource the work by seeking volunteers within EMC. Here is a list of how the various roles on the working team were fulfilled.

- **Business User, Project Sponsor, Project Manager: Vice President from Office of the CTO**
- **Business Intelligence Analyst: Representatives from IT Data Engineer and Database Administrator (DBA): Representatives from IT**
- **Data Scientist: Distinguished Engineer, who also developed the social graphs shown in the GINA case study**

The project sponsor's approach was to leverage social media and blogging [26] to accelerate the collection of innovation and research data worldwide and to motivate teams of "volunteer" data scientists at worldwide locations. Given that he lacked a formal team, he needed to be resourceful about finding people who were both capable and willing to volunteer their time to work on interesting problems. Data scientists tend to be passionate about data, and the project sponsor was able to tap into this passion of highly talented people to accomplish challenging work in a creative way. The data for the project fell into two main categories. The first category represented five years of idea submissions from EMC's internal innovation contests, known as the Innovation Roadmap (formerly called the Innovation Showcase). The Innovation Roadmap is a formal, organic innovation process whereby employees from around the globe submit ideas that are then vetted and judged. The best ideas are selected for further incubation. As a result, the data is a mix of structured data, such as idea counts, submission dates, inventor names, and unstructured content, such as the textual descriptions of the ideas themselves. The second category of data encompassed minutes and notes representing innovation and research activity from around the world. This also represented a mix of structured and unstructured data. The structured data included attributes such as dates, names, and geographic locations. The unstructured documents contained the "who, what, when, and where" information that represents rich data about knowledge growth and transfer within the company. This type of information is often stored in business silos that have little to no visibility across disparate research teams.

The 10 main IHs that the GINA team developed were as follows:

IH1: Innovation activity in different geographic regions can be mapped to corporate strategic directions.

IH2: The length of time it takes to deliver ideas decreases when global knowledge transfer

occurs as part of the idea delivery process.

IH3: Innovators who participate in global knowledge transfer deliver ideas more quickly than those who do not.

IH4: An idea submission can be analyzed and evaluated for the likelihood of receiving funding.

IH5: Knowledge discovery and growth for a particular topic can be measured and compared across geographic regions.

IH6: Knowledge transfer activity can identify research-specific boundary spanners in disparate regions.

IH7: Strategic corporate themes can be mapped to geographic regions. IH8: Frequent knowledge expansion and transfer events reduce the time it takes to generate a corporate asset from an idea.

IH9: Lineage maps can reveal when knowledge expansion and transfer did not (or has not) resulted in a corporate asset.

IH10: Emerging research topics can be classified and mapped to specific ideators, innovators, boundary spanners, and assets.

The GINA (IHs) can be grouped into two categories:

Descriptive analytics of what is currently happening to spark further creativity, collaboration, and asset generation

Predictive analytics to advise executive management of where it should be investing in the future

2.8.2 Phase 2: Data Preparation:-

The team partnered with its IT department to set up a new analytics sandbox to store and experiment on the data. During the data exploration exercise, the data scientists and data engineers began to notice that certain data needed conditioning and normalization. In addition, the team realized that several missing datasets were critical to testing some of the analytic hypotheses. As the team explored the data, it quickly realized that if it did not have data of sufficient quality or could not get good quality data, it would not be able to perform the subsequent steps in the lifecycle process. As a result, it was important to determine what level of data quality and cleanliness was sufficient for the project being undertaken. In the case of the GINA, the team discovered that many of the names of the researchers and people interacting with the universities were misspelled or had leading and trailing spaces in the datastore. Seemingly small problems such as these in the data had to be addressed in

this phase to enable better analysis and data aggregation in subsequent phases.

2.8.3 Phase 3: Model Planning :-

In the GINA project, for much of the dataset, it seemed feasible to use social network analysis techniques to look at the networks of innovators within EMC. In other cases, it was difficult to come up with appropriate ways to test hypotheses due to the lack of data. In one case (IH9), the team made a decision to initiate a longitudinal study to begin tracking data points over time regarding people developing new intellectual property. This data collection would enable the team to test the following two ideas in the future:

- **IH8: Frequent knowledge expansion and transfer events reduce the amount of time it takes to generate a corporate asset from an idea.**
- **IH9: Lineage maps can reveal when knowledge expansion and transfer did not (or has not) result(ed) in a corporate asset.**

For the longitudinal study being proposed, the team needed to establish goal criteria for the study. Specifically, it needed to determine the end goal of a successful idea that had traversed the entire journey. The parameters related to the scope of the study included the following considerations:

- **Identify the right milestones to achieve this goal.**
- **Trace how people move ideas from each milestone toward the goal.**
- **Once this is done, trace ideas that die, and trace others that reach the goal.**
- **Compare the journeys of ideas that make it and those that do not. Compare the times and the outcomes using a few different methods (depending on how the data is collected and assembled). These could be as simple as t-tests or perhaps involve different types of classification algorithms.**

2.8.4 Phase 4: Model Building:-

In Phase 4, the GINA team employed several analytical methods. This included work by the data scientist using Natural Language Processing (NLP) techniques on the textual descriptions of the Innovation Roadmap ideas. In addition, he conducted social network analysis using R and RStudio, and then he developed social graphs and visualizations of the network of communications related to innovation using R's ggplot2 package. Examples of this work are shown in Figures 2.10 and 2.11.

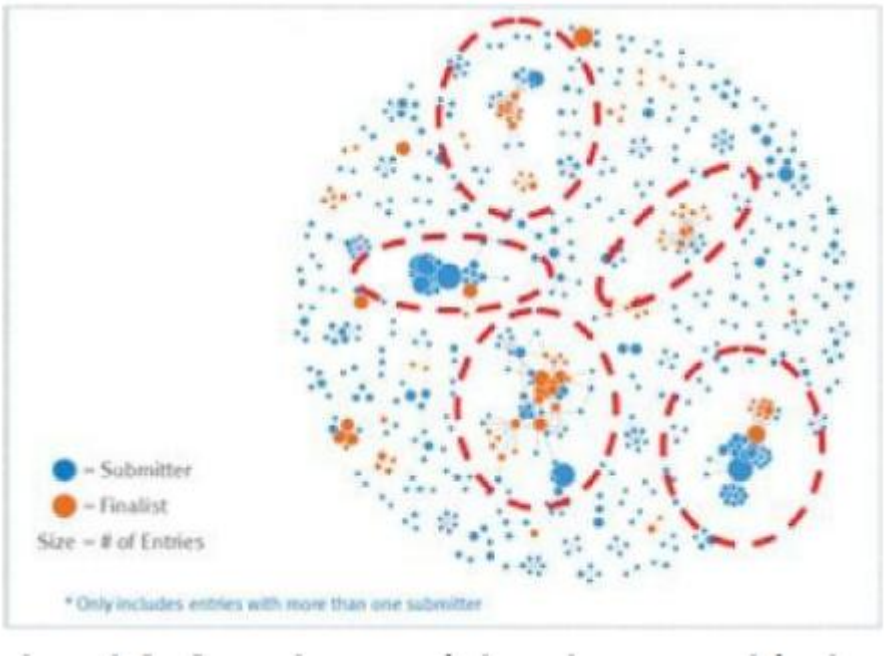


Figure 2.10 Social graph [27] visualization of idea submitters and finalists

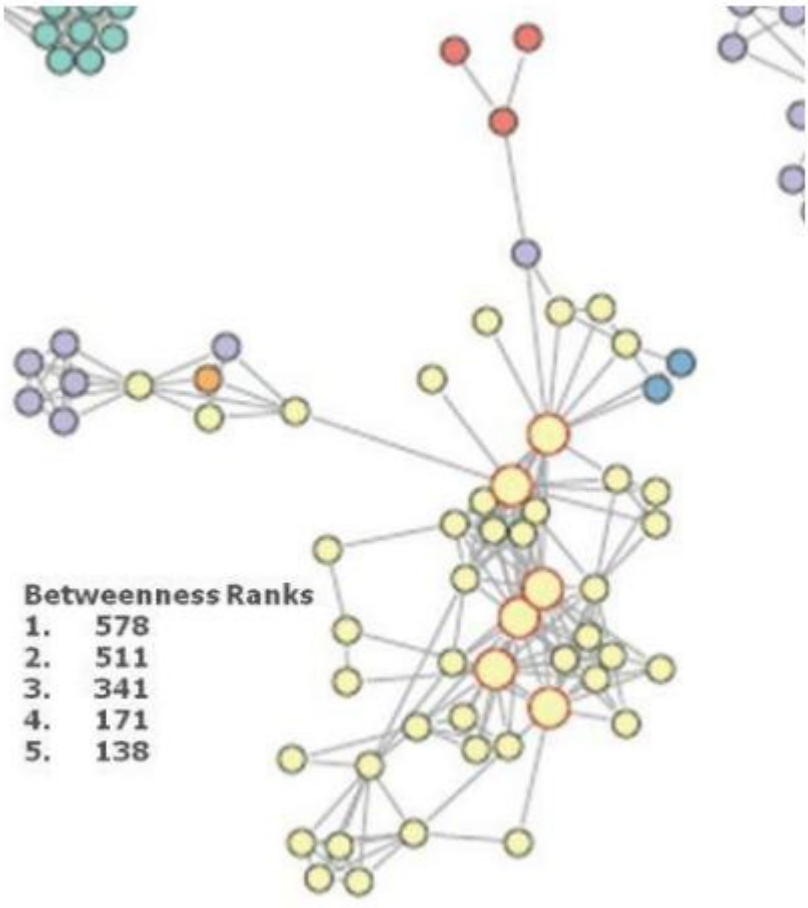


Figure 2.11 Social graph visualization of top innovation influencers

Figure 2.10 shows social graphs that portray the relationships between idea submitters within GINA. Each color represents an innovator from a different country. The large dots

with red circles around them represent hubs. A hub represents a person with high connectivity and a high "betweenness" score. The cluster in Figure 2.11 contains geographic variety, which is critical to prove the hypothesis about geographic boundary spanners. One person in this graph has an unusually high score when compared to the rest of the nodes in the graph. The data scientist identified this person and ran a query against his name within the analytic sandbox. These actions yielded the following information about this research scientist (from the social graph), which illustrated how influential he was within his business unit and across many other areas of the company worldwide:

- **In 2011, he attended the ACM SIGMOD conference, which is a top-tier conference on large-scale data management problems and databases.**
- **He visited employees in France who are part of the business unit for EMC's content management teams within Documentum (now part of the Information Intelligence Group, or IIG).**
- **He presented his thoughts on the SIGMOD conference at a virtual brownbag session attended by three employees in Russia, one employee in Cairo, one employee in Ireland, one employee in India, three employees in the United States, and one employee in Israel.**
- **In 2012, he attended the SDM 2012 conference in California.**
- **On the same trip he visited innovators and researchers at EMC federated companies, Pivotal and VMware.**
- **Later on that trip he stood before an internal council of technology leaders and introduced two of his researchers to dozens of corporate innovators and researchers.**

This finding suggests that at least part of the initial hypothesis is correct; the data can identify innovators who span different geographies and business units. The team used Tableau software for data visualization and exploration and used the Pivotal Greenplum database as the main data repository and analytics engine.

2.8.5 Phase 5: Communicate Results:-

In Phase 5, the team found several ways to cull results of the analysis and identify the most impactful and relevant findings. This project was considered successful in identifying boundary spanners and hidden innovators. As a result, the CTO office launched longitudinal studies to begin data collection efforts and track innovation results over longer

periods of time. The GINA project promoted knowledge sharing related to innovation and researchers spanning multiple areas within the company and outside of it. GINA also enabled EMC to cultivate additional intellectual property that led to additional research topics and provided opportunities to forge relationships with universities for joint academic research in the fields of Data Science and Big Data. In addition, the project was accomplished with a limited budget, leveraging a volunteer force of highly skilled and distinguished engineers and data scientists. One of the key findings from the project is that there was a disproportionately high density of innovators in Cork, Ireland. Each year, EMC hosts an innovation contest, open to employees to submit innovation ideas that would drive new value for the company. When looking at the data in 2011, 15% of the finalists and 15% of the winners were from Ireland. These are unusually high numbers, given the relative size of the Cork COE compared to other larger centers in other parts of the world. After further research, it was learned that the COE in Cork, Ireland had received focused training in innovation from an external consultant, which was proving effective. The Cork COE came up with more innovation ideas, and better ones, than it had in the past, and it was making larger contributions to innovation at EMC. It would have been difficult, if not impossible, to identify this cluster of innovators through traditional methods or even anecdotal, word-ofmouth feedback. Applying social network analysis enabled the team to find a pocket of people within EMC who were making disproportionately strong contributions. These findings were shared internally through presentations and conferences and promoted through social media and blogs.

2.8.6 Phase 6: Operationalize:-

Running analytics against a sandbox filled with notes, minutes, and presentations from innovation activities yielded great insights into EMC's innovation culture. Key findings from the project include these:

- **The CTO office and GINA need more data in the future, including a marketing initiative to convince people to inform the global community on their innovation/research activities.**
- **Some of the data is sensitive, and the team needs to consider security and privacy related to the data, such as who can run the models and see the results.**
- **In addition to running models, a parallel initiative needs to be created to improve basic Business Intelligence activities, such as dashboards, reporting, and queries on research activities worldwide.**

- **A mechanism is needed to continually reevaluate the model after deployment. Assessing the benefits is one of the main goals of this stage, as is defining a process to retrain the model as needed.**

In addition to the actions and findings listed, the team demonstrated how analytics can drive new insights in projects that are traditionally difficult to measure and quantify. This project informed investment decisions in university research projects by the CTO office and identified hidden, high-value innovators. In addition, the CTO office developed tools to help submitters improve ideas using topic modeling as part of new recommender systems to help idea submitters find similar ideas and refine their proposals for new intellectual property. Table 2.3 outlines an analytics plan for the GINA case study example. Although this project shows only three findings, there were many more. For instance, perhaps the biggest overarching result from this project is that it demonstrated, in a concrete way, that analytics can drive new insights in projects that deal with topics that may seem difficult to measure, such as innovation.

Table 2.3 Analytic Plan from the EMC GINA Project

Components of Analytic Plan	GINA Case Study
Discovery Business Problem Framed	Tracking global knowledge growth, ensuring effective knowledge transfer, and quickly converting it into corporate assets. Executing on these three elements should accelerate innovation.
Initial Hypotheses	An increase in geographic knowledge transfer improves the speed of idea delivery.
Data	Five years of innovation idea submissions and history; six months of textual notes from global innovation and research activities
Model Planning Analytic Technique	Social network analysis, social graphs, clustering, and regression analysis
Result and Key Findings	1. Identified hidden, high-value innovators and found ways to share their knowledge 2. Informed investment decisions in university research projects 3. Created tools to help submitters improve ideas with idea recommender systems

Innovation is an idea that every company wants to promote, but it can be difficult to measure innovation or identify ways to increase innovation. This project explored this

issue from the standpoint of evaluating informal social networks to identify boundary spanners and influential people within innovation subnetworks. In essence, this project took a seemingly nebulous problem and applied advanced analytical methods to tease out answers using an objective, fact-based approach. Another outcome from the project included the need to supplement analytics with a separate datastore for Business Intelligence reporting, accessible to search innovation/research initiatives. Aside from supporting decision making, this will provide a mechanism to be informed on discussions and research happening worldwide among team members in disparate locations. Finally, it highlighted the value that can be gleaned through data and subsequent analysis. Therefore, the need was identified to start formal marketing programs to convince people to submit (or inform) the global community on their innovation/research activities. The knowledge sharing was critical. Without it, GINA would not have been able to perform the analysis and identify the hidden innovators within the company.

Conclusion:

ASSIGNMENT NO.10

Title:

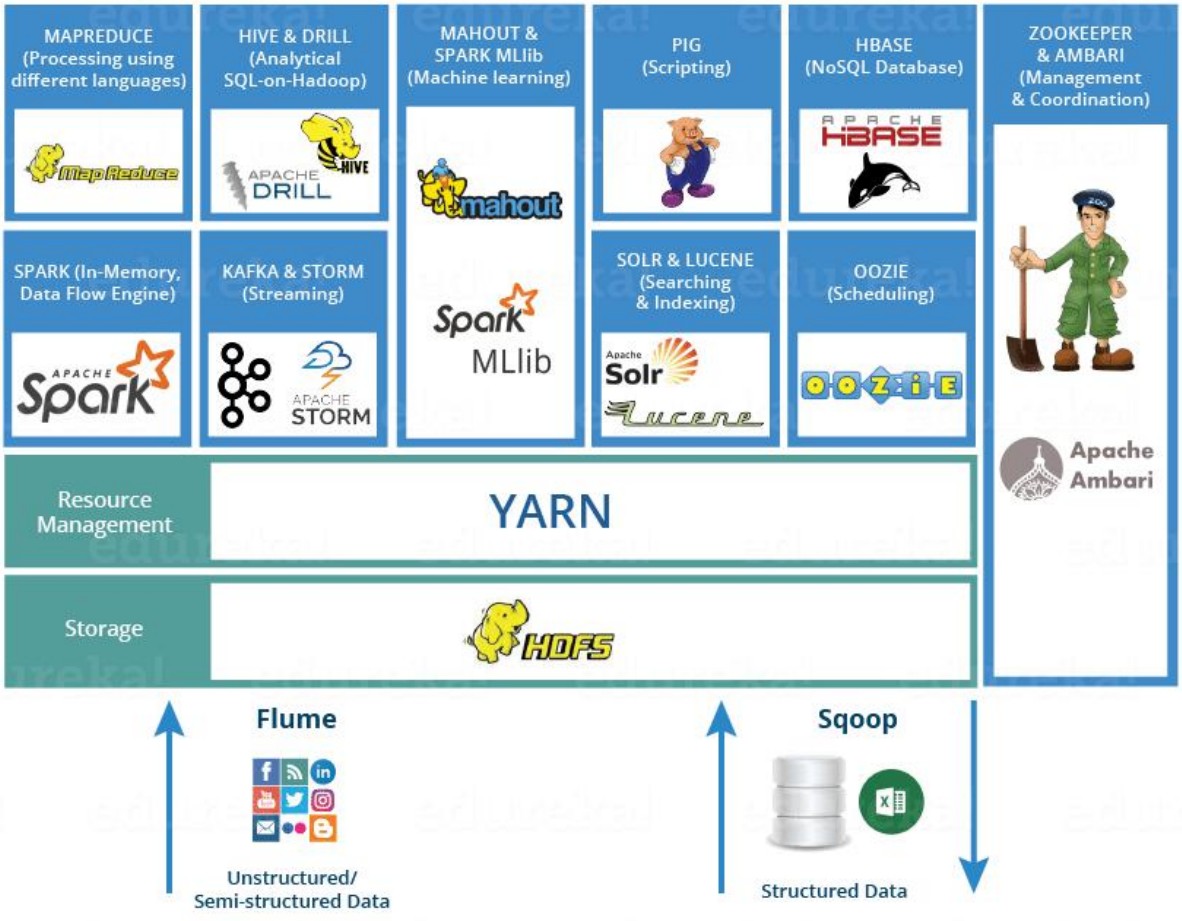
Write a case study on Hadoop Ecosystem components.

HADOOP ECOSYSTEM

Hadoop Ecosystem is neither a programming language nor a service, it is a platform or framework which solves big data problems. You can consider it as a suite which encompasses a number of services (ingesting, storing, analyzing and maintaining) inside it.

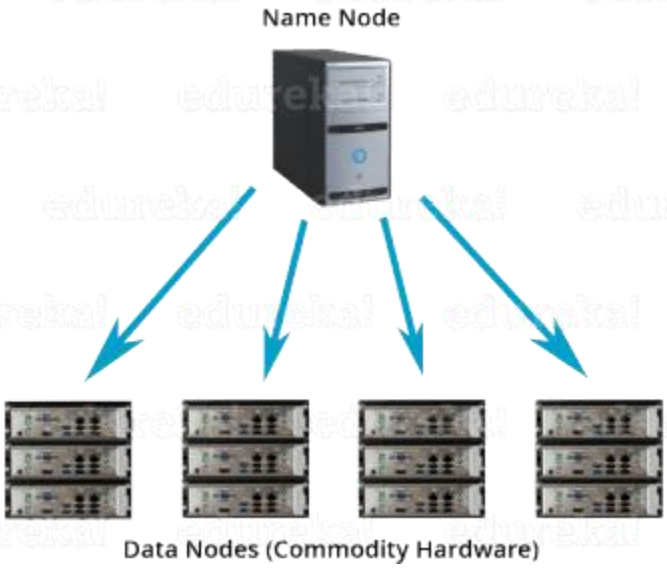
Below are the Hadoop components, that together form a Hadoop ecosystem.

- **HDFS** -> Hadoop Distributed File System
- **YARN** -> Yet Another Resource Negotiator
- **MapReduce** -> Data processing using programming
- **Spark** -> In-memory Data Processing
- **PIG, HIVE**-> Data Processing Services using Query (SQL-like)
- **HBase** -> NoSQL Database
- **Mahout, Spark MLlib** -> Machine Learning
- **Apache Drill** -> SQL on Hadoop
- **Zookeeper** -> Managing Cluster
- **Oozie** -> Job Scheduling
- **Flume, Sqoop** -> Data Ingesting Services
- **Solr& Lucene** -> Searching & Indexing
- **Ambari** -> Provision, Monitor and Maintain cluster



HDFS

- Hadoop Distributed File System is the core component or you can say, the backbone of Hadoop Ecosystem.
- HDFS is the one, which makes it possible to store different types of large data sets (i.e. structured, unstructured and semi structured data).
- HDFS creates a level of abstraction over the resources, from where we can see the whole HDFS as a single unit.
- It helps us in storing our data across various nodes and maintaining the log file about the stored data (metadata).
- HDFS has two core components, i.e. **NameNode and DataNode**.
    1. The **NameNode** is the main node and it doesn't store the actual data. It contains metadata, just like a log file or you can say as a table of content. Therefore, it requires less storage and high computational resources.
    2. On the other hand, all your data is stored on the **DataNodes** and hence it requires more storage resources. These DataNodes are commodity hardware (like your laptops and desktops) in the distributed environment. That's the reason, why Hadoop solutions are very cost effective.
    3. You always communicate to the NameNode while writing the data. Then, it internally sends a request to the client to store and replicate data on various DataNodes.



YARN

Consider YARN as the brain of your Hadoop Ecosystem. It performs all your processing activities by allocating resources and scheduling tasks.

- It has two major components, i.e. **Resource Manager and Node Manager**.
    1. **Resource Manager** is again a main node in the processing department.
    2. It receives the processing requests, and then passes the parts of requests to corresponding Node Managers accordingly, where the actual processing takes place.
    3. **Node Managers** are installed on every Data Node. It is responsible for execution of task on every single Data Node.

-
    1. **Schedulers:** Based on your application resource requirements, Schedulers perform scheduling algorithms and allocates the resources.
    2. **Applications Manager:** While Applications Manager accepts the job submission, negotiates to containers (i.e. the Data node environment where process executes) for executing the application specific Application Master and monitoring the progress. ApplicationMasters are the deamons which reside on DataNode and communicates to containers for execution of tasks on each DataNode.
    3. ResourceManager has two components: Schedulers and application manager

MAPREDUCE

It is the core component of processing in a Hadoop Ecosystem as it provides the logic of processing. In other words, MapReduce is a software framework which helps in writing applications that processes large data sets using distributed and parallel algorithms inside Hadoop environment.

- In a MapReduce program, **Map() and Reduce()** are two functions.
    1. The **Map function** performs actions like filtering, grouping and sorting.
    2. While **Reduce function** aggregates and summarizes the result produced by map function.
    3. The result generated by the Map function is a key value pair (K, V) which acts as the input for Reduce function.

Student	Department	Count	(Key, Value), Pair
Student 1	D1	1	(D1, 1)
Student 2	D1	1	(D1, 1)
Student 3	D1	1	(D1, 1)
Student 4	D2	1	(D2, 1)
Student 5	D2	1	(D2, 1)
Student 6	D3	1	(D3, 1)
Student 7	D3	1	(D3, 1)

Let us take the above example to have a better understanding of a MapReduce program.

We have a sample case of students and their respective departments. We want to calculate the number of students in each department. Initially, Map program will execute and calculate the students appearing in each department, producing the key value pair as mentioned above. This key value pair is the input to the Reduce function. The Reduce function will then aggregate each department and calculate the total number of students in each department and produce the given result.

Department	Total Student
D1	3
D2	2
D3	2

APACHE PIG

- PIG has two parts: **Pig Latin**, the language and **the pig runtime,** for the execution environment. You can better understand it as Java and JVM.
- It supports *pig latin* language, which has SQL like command structure.

10 line of pig latin = approx. 200 lines of Map-Reduce Java code

But don't be shocked when I say that at the back end of Pig job, a map-reduce job executes.

- The compiler internally converts pig latin to MapReduce. It produces a sequential set of MapReduce jobs, and that's an abstraction (which works like black box).
- PIG was initially developed by Yahoo.

- It gives you a platform for building data flow for ETL (Extract, Transform and Load), processing and analyzing huge data sets.

How Pig works?

In PIG, first the load command, loads the data. Then we perform various functions on it like grouping, filtering, joining, sorting, etc. At last, either you can dump the data on the screen or you can store the result back in HDFS.

APACHE HIVE

- Facebook created HIVE for people who are fluent with SQL. Thus, HIVE makes them feel at home while working in a Hadoop Ecosystem.
- Basically, HIVE is a data warehousing component which performs reading, writing and managing large data sets in a distributed environment using SQL-like interface.

HIVE + SQL = HQL

- The query language of Hive is called Hive Query Language(HQL), which is very similar like SQL.
- It has 2 basic components: **Hive Command Line and JDBC/ODBC driver**.
- The **Hive Command line** interface is used to execute HQL commands.
- While, Java Database Connectivity (**JDBC**) and Object Database Connectivity (**ODBC**) is used to establish connection from data storage.

- Secondly, Hive is highly scalable. As, it can serve both the purposes, i.e. large data set processing (i.e. Batch query processing) and real time processing (i.e. Interactive query processing).
- It supports all primitive data types of SQL.
- You can use predefined functions, or write tailored user defined functions (UDF) also to accomplish your specific needs.

APACHE MAHOUT

Now, let us talk about Mahout which is renowned for machine learning. Mahout provides an environment for creating machine learning applications which are scalable.

Machine learning algorithms allow us to build self-learning machines that evolve by itself without being explicitly programmed. Based on user behaviour, data patterns and past experiences it makes important future decisions. You can call it a descendant of Artificial Intelligence (AI).

What Mahout does?

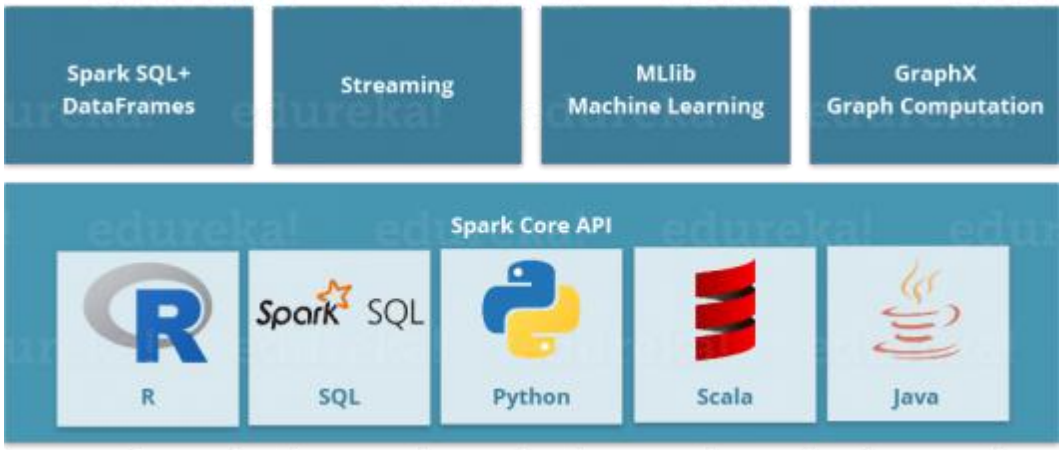
It performs **collaborative filtering, clustering and classification.** Some people also consider **frequent item set missing** as Mahout's function. Let us understand them individually:

1. **Collaborative filtering:** Mahout mines user behaviors, their patterns and their characteristics and based on that it predicts and make recommendations to the users. The typical use case is E-commerce website.
2. **Clustering:** It organizes a similar group of data together like articles can contain blogs, news, research papers etc.
3. **Classification**: It means classifying and categorizing data into various sub-departments like articles can be categorized into blogs, news, essay, research papers and other categories.
4. **Frequent item set missing**: Here Mahout checks, which objects are likely to be appearing together and make suggestions, if they are missing. For example, cell phone and cover are brought together in general. So, if you search for a cell phone, it will also recommend you the cover and cases.

Mahout provides a command line to invoke various algorithms. It has a predefined set of library which already contains different inbuilt algorithms for different use cases.

APACHE SPARK

- Apache Spark is a framework for real time data analytics in a distributed computing environment.
- The Spark is written in Scala and was originally developed at the University of California, Berkeley.
- It executes in-memory computations to increase speed of data processing over Map-Reduce.
- It is 100x faster than Hadoop for large scale data processing by exploiting in-memory computations and other optimizations. Therefore, it requires high processing power than Map-Reduce.



As you can see, Spark comes packed with high-level libraries, including support for R, SQL, Python, Scala, Java etc. These standard libraries increase the seamless integrations in complex workflow. Over this, it also allows various sets of services to integrate with it like MLlib, GraphX, SQL + Data Frames, Streaming services etc. to increase its capabilities.

. Apache Spark best fits for real time processing, whereas Hadoop was designed to store unstructured data and execute batch processing over it. When we combine, Apache Spark's ability, i.e. high processing speed, advance analytics and multiple integration support with Hadoop's low cost operation on commodity hardware, it gives the best results.

That is the reason why, Spark and Hadoop are used together by many companies for processing and analyzing their Big Data stored in HDFS.

APACHE HBASE

- HBase is an open source, non-relational distributed database. In other words, it is a NoSQL database.
- It supports all types of data and that is why, it's capable of handling anything and everything inside a Hadoop ecosystem.
- It is modelled after Google's BigTable, which is a distributed storage system designed to cope up with large data sets.
- The HBase was designed to run on top of HDFS and provides BigTable like capabilities.
- It gives us a fault tolerant way of storing sparse data, which is common in most Big Data use cases.
- The HBase is written in Java, whereas HBase applications can be written in REST, Avro and Thrift APIs.

For better understanding, let us take an example. You have billions of customer emails and you need to find out the number of customers who has used the word complaint in their emails. The request needs to be processed quickly (i.e. at real time). So, here we are handling a large data set while retrieving a small amount of data. For solving these kind of problems, HBase was designed.

APACHE DRILL

Apache Drill is used to drill into any kind of data. It's an open source application which works with distributed environment to analyze large data sets.

- It is a replica of Google Dremel.
- It supports different kinds NoSQL databases and file systems, which is a powerful feature of Drill. For example: Azure Blob Storage, Google Cloud Storage, HBase, MongoDB, MapR-DB HDFS, MapR-FS, Amazon S3, Swift, NAS and local files.

So, basically the main aim behind Apache Drill is to provide scalability so that we can process petabytes and exabytes of data efficiently (or you can say in minutes).

- The main power of Apache Drill lies in ***combining a variety of data stores just by using a single query.***

- Apache Drill basically follows the ANSI SQL.
- It has a powerful scalability factor in supporting millions of users and serve their query requests over large scale data.

APACHE ZOOKEEPER

- Apache Zookeeper is the coordinator of any Hadoop job which includes a combination of various services in a Hadoop Ecosystem.
- Apache Zookeeper coordinates with various services in a distributed environment.

Before Zookeeper, it was very difficult and time consuming to coordinate between different services in Hadoop Ecosystem. The services earlier had many problems with interactions like common configuration while synchronizing data. Even if the services are configured, changes in the configurations of the services make it complex and difficult to handle. The grouping and naming was also a time-consuming factor.

Due to the above problems, Zookeeper was introduced. It saves a lot of time by performing **synchronization, configuration maintenance, grouping and naming.**

Although it's a simple service, it can be used to build powerful solutions.

APACHE OOZIE

Consider Apache Oozie as a clock and alarm service inside Hadoop Ecosystem. For Apache jobs, Oozie has been just like a scheduler. It schedules Hadoop jobs and binds them together as one logical work.

There are two kinds of Oozie jobs:

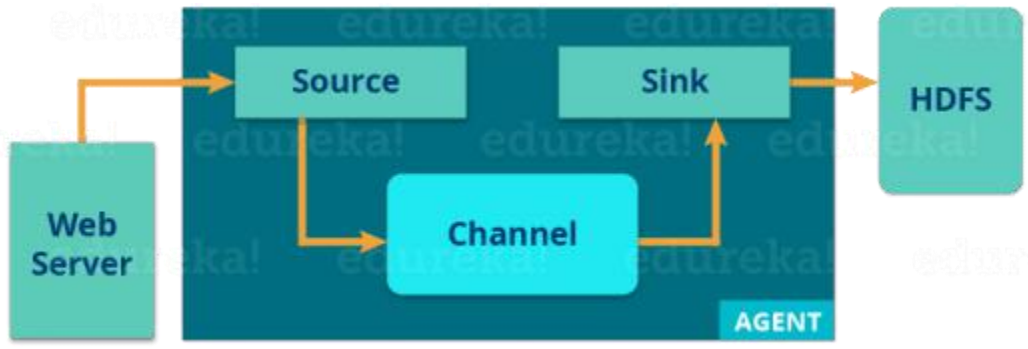
1. **Oozie workflow**: These are sequential set of actions to be executed**.** You can assume it as a relay race. Where each athlete waits for the last one to complete his part.
2. **Oozie Coordinator**: These are the Oozie jobs which are triggered when the data is made available to it. Think of this as the response-stimuli system in our body. In the same manner as we respond to an external stimulus, an Oozie coordinator responds to the availability of data and it rests otherwise.

APACHE FLUME

Ingesting data is an important part of our Hadoop Ecosystem.

- The Flume is a service which helps in ingesting unstructured and semi-structured data into HDFS.
- It gives us a solution which is reliable and distributed and helps us in **collecting, aggregating** and **moving large amount of data sets**.
- It helps us to ingest online streaming data from various sources like network traffic, social media, email messages, log files etc. in HDFS.

Now, let us understand the architecture of Flume from the below diagram:



There is a **Flume agent** which ingests the streaming data from various data sources to HDFS. From the diagram, you can easily understand that the web server indicates the data source. Twitter is among one of the famous sources for streaming data.

The flume agent has 3 components: **source, sink and channel**.

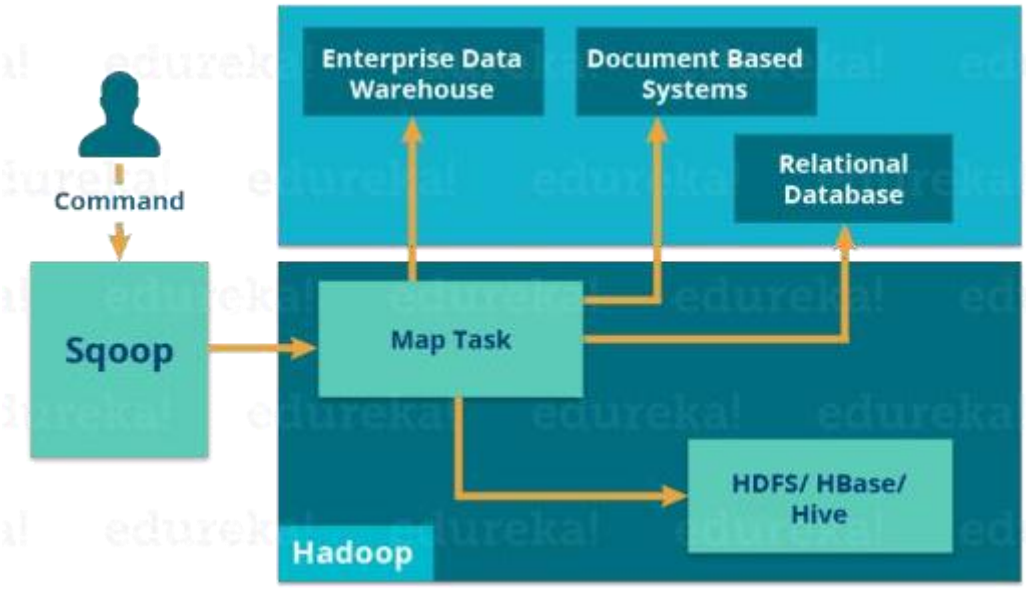
1. **Source**: it accepts the data from the incoming streamline and stores the data in the channel.
2. **Channel:** it acts as the local storage or the primary storage. A Channel is a temporary storage between the source of data and persistent data in the HDFS.
3. **Sink:** Then, our last component i.e. Sink, collects the data from the channel and commits or writes the data in the HDFS permanently.

APACHE SQOOP

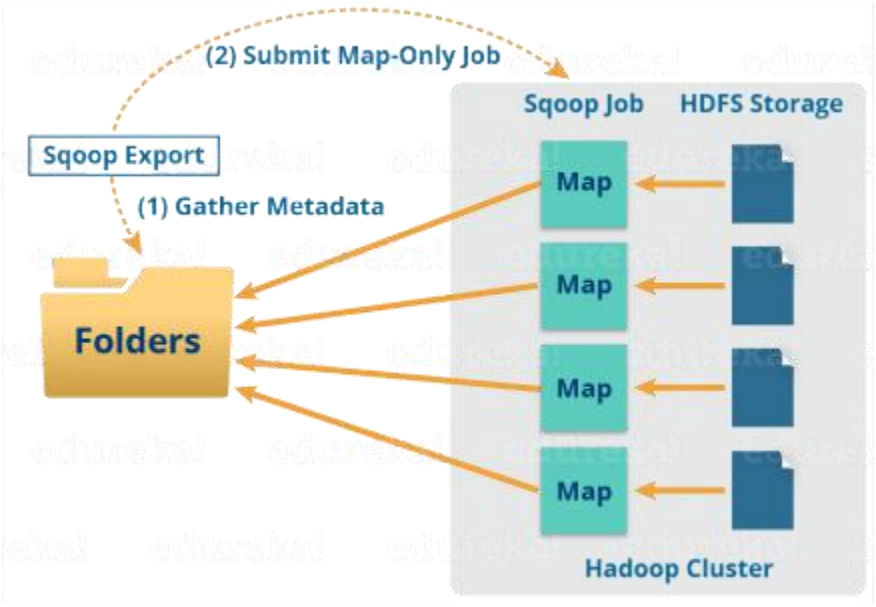
The major difference between Flume and Sqoop is that:

- Flume only ingests unstructured data or semi-structured data into HDFS.
- While Sqoop can import as well as export structured data from RDBMS or Enterprise data warehouses to HDFS or vice versa.

Let us understand how Sqoop works using the below diagram:



When we submit Sqoop command, our main task gets divided into sub tasks which is handled by individual Map Task internally. Map Task is the sub task, which imports part of data to the Hadoop Ecosystem. Collectively, all Map tasks imports the whole data.



Export also works in a similar manner.

When we submit our Job, it is mapped into Map Tasks which brings the chunk of data from HDFS. These chunks are exported to a structured data destination. Combining all these exported chunks of data, we receive the whole data at the destination, which in most of the cases is an RDBMS (MYSQL/Oracle/SQL Server).

APACHE SOLR & LUCENE

Apache Solr and Apache Lucene are the two services which are used for searching and indexing in Hadoop Ecosystem.

- Apache Lucene is based on Java, which also helps in spell checking.
- If Apache Lucene is the engine, Apache Solr is the car built around it. Solr is a complete application built around Lucene.

- It uses the Lucene Java search library as a core for search and full indexing.

APACHE AMBARI

Ambari is an Apache Software Foundation Project which aims at making Hadoop ecosystem more manageable.

It includes software for **provisioning, managing and monitoring** Apache Hadoop clusters.

The Ambari provides:

1. **Hadoop cluster provisioning**:
    - It gives us step by step process for installing Hadoop services across a number of hosts.
    - It also handles configuration of Hadoop services over a cluster.
2. **Hadoop cluster management:**
    - It provides a central management service for starting, stopping and re-configuring Hadoop services across the cluster.
3. **Hadoop cluster monitoring:**
    - For monitoring health and status, Ambari provides us a dashboard.
    - The **Amber Alert framework** is an alerting service which notifies the user, whenever the attention is needed. For example, if a node goes down or low disk space on a node, etc.

At last, I would like to draw your attention on three things importantly:

1. Hadoop Ecosystem owes its success to the whole developer community, many big companies like Facebook, Google, Yahoo, University of California (Berkeley) etc. have contributed their part to increase Hadoop's capabilities.
2. Inside a Hadoop Ecosystem, knowledge about one or two tools (Hadoop components) would not help in building a solution. You need to learn a set of Hadoop components, which works together to build a solution.
3. Based on the use cases, we can choose a set of services from Hadoop Ecosystem and create a tailored solution for an organization.

Conclusion: