INDEX

S No.	Practical	Date	Sign
1	Introduction to Data Analytics using Python • Six Steps of Data Analysis Process • Different Sources of Data for Data Analysis		
2	Introduction to Python Libraries: NumPy, Pandas, SciPy, Scikit-learn, Matplotlib, Seaborn.		
3	Introduction to Python Programming • Datatypes • Operators • Loops • Central Tendency Measures • MATRIX OPERATIONS: o Write a Python program to do the following operations: o Library: NumPy o a) Create multi-dimensional arrays and find its shape and dimension o b) Create a matrix full of zeros and ones o c) Reshape and flatten data in the array o d) Append data vertically and horizontally o e) Apply indexing and slicing on array o f) Use statistical functions on array - Min, Max, Mean, Median and Standard Deviation • LINEAR ALGEBRA ON MATRICES o Write a Python program to do the following operations: o Library: NumPy o a) Dot and matrix product of two arrays o b) Compute the Eigen values of a matrix o c) Solve a linear matrix equation such as $3 * x_0 + x_1 = 9, x_0 + 2 * x_1 = 8$ o d) Compute the multiplicative inverse of a matrix o e) Compute the rank of a matrix o f) Compute the determinant of an array		
4	UNDERSTANDING DATA Write a Python program to do the following operations: Data set: brain_size.csv		

	Library: Pandas a) Loading data from CSV file b) Compute the basic statistics of given data - shape, no. of columns, mean c) Splitting a data frame on values of categorical variables d) Visualize data using Scatter plot		
5	CORRELATION MATRIX Write a python program to load the dataset and understand the input data Dataset: Pima Indians Diabetes Dataset Library: Scipy a) Load data, describe the given data and identify missing, outlier data items b) Find correlation among all attributes c) Visualize correlation matrix		
6	Discretization (Binning) and normalization DATA PREPROCESSING – HANDLING MISSING VALUES Write a python program to impute missing values with various techniques on given dataset. a) Remove rows/ attributes b) Replace with mean or mode c) Write a python program to perform transformation of data using Discretization (Binning) and normalization (MinMaxScaler or MaxAbsScaler) on given dataset.		
7	Regression: Linear Regression, Logistic Regression		
8	Apriori Algorithm		
9	KNN Dataset: The data set consists of 50 samples from each of three species of Iris: Iris setosa, Iris virginica (Exploratory Data Analysis on Iris Dataset) and Iris versicolor. Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Libraries: import Numpy as np Write a python program to a) Calculate Euclidean Distance. b) Get Nearest Neighbors c) Make Predictions.		
10	K-means Clustering Dataset: Diabetes		
11	Decision Tree Classification		

Topic _____ Date _____

Practical-1

Introduction to data analysis using Python

- Six steps of Data Analysis Process

The collection, transformation & organization of data to draw conclusions make predictions for the future & make informed data driven decisions is called Data Analysis.

Six steps of Data Analysis are as follow.

ASK:

The first step in the data analysis process is to define the problem precisely or question that needs addressing. This steps involves identifying key objectives & understanding the scope of analysis wit business goals.

Key considerations:

→ what specific problem are we trying to solve?
→ what are the desired outcomes of this analysis?
→ who are the stakeholders & impact on them?

Prepare:

This step involves collecting & preparing the necessary data. This includes gathering data from various sources that includes internal databass, APIs or external datasets, ensuring that data is relevant reliable & sufficient for analysis. During

Topic _____ Date _____

this phase, its crucial to assess data for accuracy, completeness & any potential biases.

Key consideration:

→ Where data is coming from?
→ Is the data comprehensive?
→ Are there any inherit biases in the data?

3. Clean and Process Data:

In this stage focus is on cleaning, transforming & refining the data to ensure its quality. This involves handling missing values, removing duplicates addressing outliers & transforming variables into a format suitable for analysis

Key consideration:

→ What inconsistencies exists in datasets?
→ What transformations are necessary?
→ How will missing/incomplete data be treated?

4. Analyze:

The Next phase is to apply analytical techniques to extract insights. This includes performing exploratory data analysis (EDA) to identify trends, patterns, correlation & anamolies.

Key consideration:

→ What initial trends or patterns are emerging?
→ Which analytical methods are most appro.?
→ How do insights align [illegible] problem initially d?

Topic _____ Date _____

Share

The insights derived from the analysis are only as valuable as the ability to communicate them effectively. In this stage the focus shifts to presenting the findings in a clear & compelling manner. This involves data visualizations, such as charts, graphs & dashboards.

Key Consideration:

→ What is the best way to visualize the data?
→ How can results be comm. to maximize clarity?
→ What story is data telling?

Act or Report:

The final stp is to transform the insight into actionable recommendation. Basd on the findings, informed dicision can be made to address original problem or achieve business goals.

Key consideration:

→ What specific action should be taken basd on insights?
→ How will success be measurid?
→ What future steps or analyses are necessary to build on these insights?

BABA Design

Topic _____ Date _____

5. Share

The insights derived from the analysis are only as valuable as the ability to communicate them effectively. In this stage the focus shifts to presenting the findings in a clear & compelling manner. This involves data visualizations, such as charts, graphs & dashboards.

Key Consideration:

→ What is the best way to visualize the data?
→ How can results be comm. to maximize clarity?
→ What story is data telling?

6. Act or Report:

The final step is to transform the insight into actionable recommendation. Based on the findings, informed decision can be made to address original problem or achieve business goals.

Key consideration:

→ What specific action should be taken based on insights?
→ How will success be measured?
→ What future steps or analyses are necessary to build on these insights?

Topic _____ Date _____

• Different sources of data for Data Analysis

Data can be gathered from two places: internal & external sources. The information collected from internal sources is also called primary data while the information gathered from outside references is called secondary data.

1) Primary Data
The data which is raw, original & extracted directly from the official sources is known as primary data. This type of data is collected directly by performing techniques such as questionnaires, interviews & surveys.

Few Methods of collecting primary data:

a) Interview Method: The data collected during this process is through interviewing the target audience.

b) Survey Method: The survey method is the process of research where a list of relevant questions are asked & answers are noted down in the form of audio, text or video.

c) Observation Method: In this the researchers keenly observes the behavior & practices of target audien ce using some data collecting tools & stores the observed data in the form of text, audio or raw format.

Topic _____ Date _____

Different sources of data for Data Analysis

Data can be gathered from two places: internal & external sources. The information collected from internal sources is also called primary data. while the information gathered from outside references is called secondary data.

1) Primary Data

The data which is raw, original & extracted directly from the official sources is known as primary data. This type of data is collected directly by performing techniques such as questionnaires, interviews & surveys.

Few Methods of collecting primary data:

a) Interview Method: The data collected during this process is through interviewing the target audience.

b) Survey Method: The survey method is the process of research where a list of relevant questions are asked & answers are noted down in the form of audio, text or video.

c) Observation Method: In this the researchers keenly observes the behavior & practices of target audience using some data collecting tools & stores the observed data in the form of text, audio or raw format.

Topic _____ Date _____

d) Experimental Method: The experimental method is the process of collecting data through performing experiments, research & investigation. Most frequently used experiments are completely randomize design (CRD), Randomize Block design (RBD), Latin Square Design (LSD), Factorial Design.

2) Secondary Data:

Secondary data is the data which has already been collected & reused again for some valid purpose. It has 2 types of sources:

a) Internal source: These type of data can easily be found within the organisation such as market record, a sales record, customer data etc. The cost & time consumption is less in obtaining internal sources.

b) External source: This type of data can be gained through external third party resources. The cost & time consumption is more because this contain a huge amount of data.
Eg: Govt. publications, News publications, planning commission etc.

Topic _____ Date _____

Practical-2

Introduction to Python libraries: Numpy, Pandas, Scipy, Scikit-learn, Matplotlib, Seaborn.

Numpy:- Numpy (Numerical Python) is the foundation for scientific computing in python. It provides support for large, multi-dimensional arrays & matrices, along with a collection of mathematical functions to operate on these arrays.

Example:

```
import numpy as np
arr = np.array([1,2,3,4,5])
print("Mean:", np.mean(arr))
print("sum:", np.sum(arr))
```

Pandas:- Pandas is a fast, powerful & flexible library for data analysis & manipulation. It provides two primary data structures Series (one-dimensional data) & DataFrame (two-dimensional tabular data). With pandas, it's easy to perform filtering, grouping & aggregations operations.

Example:

```
import pandas as pd
df = pd.read_csv("data.csv")
df.head()
```

Topic _____ Date _____

Practical-2

Introduction to Python libraries: Numpy, Pandas, Scipy, Scikit-learn, Matplotlib, Seaborn.

1) Numpy:- Numpy (Numerical Python) is the foundation for scientific computing in python. It provides support for large, multi-dimensional arrays & matrices, along with a collection of mathematical functions to operate on these arrays.

Example:

```
import numpy as np
arr = np.array([1,2,3,4,5])
print("Mean:", np.mean(arr))
print("sum:", np.sum(arr))
```

2) Pandas:- Pandas is a fast, powerful & flexible library for data analysis & manipulation. It provides two primary data structures Series (one-dimensional data) & DataFrame (two-dimensional tabular data). With pandas, it's easy to perform filtering, grouping & aggregations operations.

Example:

```
import pandas as pd
df = pd.read_csv("data.csv")
df.head()
```

Topic _____ Date _____

3) **SciPy:** Build on top of Numpy, provides additional tools for scientific & engineering computations. It is widely used for tasks such as optimization, integration, interpolation & eigenvalue problem etc. It contains several packages like scipy.optimize, scipy.stats & scipy.linalg

Example:

```
from scipy import stats
data = [1,2,2,3,4,4,4,5,6]
mode_value = stats.mode(data)
print(mode_value)
```

4) **Scikit-learn:** Scikit-learn is a powerful library for machine learning & data mining, offering tools for classification, regression, clustering, and model selection. It also provides utilities for data preprocessing, splitting datasets & evaluating models

Example:-

```
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(...)
```

Topic _____ Date _____

5) Matplotlib :- Matplot lib is the most widely used library for creating static, interactive & animated plots in python. It can generate a wide variety of charts incl. bar, line, histogram, scatter plot etc.

Example:

```
import matplotlib.pyplot as plt
X = [1, 2, 3, 4]
Y = [10, 20, 25, 30]
plt.plot(X, Y)
plt.show()
```

6) Seaborn : Seaborn is a data visualization library built on top of matplotlib, designed to make statistical graphics more attractive & informative. It simplifies the process of creating complex plots & offers built-in themes & color palettes for making aesthetically pleasing visualization.

Example:

```
import seaborn as sns
import pandas as pd
df = pd.read_csv("data.csv")
sns.barplot(x="category", y="values", dat
plt.show()
```

Topic _____ Date _____

Practical-3

Introduction to Python Programming:

→ Datatypes
→ Operators
→ Loops
→ Central Tendancy Measures
→ Matrix Operations
→ Linear algebra on Matrices

Python is a versatile, high-level programming language known for its clear syntax & ease of learning. It supports multiple programming paradigms such as object-oriented, procedural & functional programming. Python's interpreted nature make it an ideal language for web development, data analysis, ML, automations & more.

Datatypes:- Python has several built in datatypes used to classify data as:

Numeric Types:

1) int : Represents whole number
2) float: Represents numbers with decimal pt.
3) Complex: Numbers with real & imaginary part

Text Type:

1) str: Represents a sequence of characters (eg: "Hello")

Topic _____ Date _____

Sequence Types:

1) List: Ordered, Mutable collection of items ([1,2,3])
2) tuple: Ordered, Immutable coll. of items ((1,2,3))
3) Range: Generate a sequence of no. (range(0,5))

Mapping Type:

1) dict: Unordered collection of key value pair
({"name": "Anshul", "age": 23}).

Set Types:

1) Set: Unordered, mutable collection of unique items
(eg: {1,2,3})

Boolean Type:

1) Bool: Represents 'True' or 'False' (True, False)

Operators: Operators are special symbols that perform computation or operations on operands & values

Arithematic operators:- Perform Mathematical operations

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
//	Floor Division
**	Exponentiation

Eg:- 5+3
= 8

Topic _____ Date _____

Comparison operators: Compare two values & return 'True' or 'False'

==	equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Eg:
(5 == 6) 'False'.

Logical operators: Combine boolean expressions
and : True if both operands are True
or : True if At least one operand is True
not : Reverse the boolean value

Eg:-
(5>3) and (3>2) # 'True'

Assignment operators: Assign values to variables

=	Simple assignment
+=	Add & assign
-=	Subtract & Assign
*=	Multiply & assign
/=	divide & assign

Eg
x = 5
x += 3
(x: 8)

Topic _____ Date _____

Bitwise operators: Perform bit level operations

&	AND
|	OR
^	XOR
~	NOT
<<	Left Shift
>>	Right Shift.

Loops: Loops are used in python to repeat a code multiple times

For Loop: Iterates over a sequence & executes the block of code for each item

Eg:-

```
for i in range(5):
    print(i)
```

while loop: Repeats a block of code as long as a condition is 'True'.

Eg:-

```
i = 0
while (i<5)
    print(i)
    i += 1
```

Central Tendency Measures: Mathematically central tendency means measuring the center or distribution of location of values of a data set. It gives the idea of the average value of the data in the data set & also an indication of how widely the

Topic

values are spread in the dataset:

Mean: The arithematic average of all values in a dataset. It's calculated by summing all the values & dividing by no. of values.

Eg:- data = [10, 20, 30, 40]
mean = sum(data)/len(data) #25.

Median: The middle value in a sorted dataset.

suppose: n is no. of values in a dataset.

if n is odd; Median will be at n/2

else median will be $\frac{\frac{n}{2} + \frac{n+1}{2}}{2}$

Eg:- data = [10, 20, 30, 40]
median = (data[1] + data[2])/2 #25.

Mode: The value that appears most frequently in the dataset. If no value repeats, the dataset is considered to have no mode.

Eg:-
data = [10, 20, 20, 30, 40]
mode_value = mode(data) #20.

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Shape of the Matrix (3, 3)
Dimension of the Matrix 2
```

```
Matrix of Zeros
 [[0. 0. 0.]
 [0. 0. 0.]]
Matrix of Ones
 [[1. 1.]
 [1. 1.]
 [1. 1.]]
```

```
Array after reshaping
 [[[1. 1.]
  [1. 1.]
  [1. 1.]]]
Array after flattening
 [0. 0. 0. 0. 0. 0.]
```

```
Horrizontal Append [0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1.]
Vertical Append
 [[1. 1. 1.]
 [1. 1. 1.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

Topic _____ Date _____

Matrix Operation:

```
import numpy as np
```

a) Create a multi-dimensional array & find its shape & dimension

```
arr = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(arr)
print("shape of the matrix", arr.shape)
print("Dimension of the matrix", arr.ndim)
```

b) Create a matrix full of zeros & ones.

```
arr0 = np.zeros((2,3))
arr1 = np.ones((3,2))
print("Matrix of zeros\n", arr0)
print("Matrix of ones\n", arr1)
```

c) Reshape & flatten data in the array.

```
arr1 = arr1.reshape((1,3,2))
arr0 = arr0.flatten()
print("Array after Reshaping\n", arr1)
print("Array after flattening\n", arr0)
```

d) Append data horizontally & Vertically.

```
print("Horizontal append", np.append(arr0, arr1))
print("Vertical append", np.append(arr1.reshape((2,3)),
                                   arr0.reshape((2,3)),
                                   axis=0))
```

```
Element at index (2,1) in arr: 8
Elements of 2nd Row:  [4 5 6]
```

```
Min element of arr is:  1
Max element of arr is:  9
Mean of arr is:  5.0
Median of arr is:  5.0
Standard Deviation of arr1 is: 2.58
```

Topic _____ Date _____

e) Apply indexing & slicing on array

```
print("Element at index (2,1) in arr : ", arr[2,1])
print("Elements of 2nd Row : ", arr[1,0:3])
```

f) Use statistical functions on array; Min, max, Mean, Median, standard deviation.

```
print("Min element of arr is: ", np.min(arr))
print("Max element of arr is: ", np.max(arr))
print("Mean of arr is: ", np.mean(arr))
print("Median of arr is: ", np.median(arr))
print("Standard deviation of arr is : ", np.std(arr)
```

```
Dot Product of arr1 & arr2:
 [[ 42  36  30]
 [ 96  81  66]
 [150 126 102]]

Cross Product of arr1 & arr2:
 [[ -4   8  -4]
 [-10  20 -10]
 [-16  32 -16]]


Eigen Values of arr2 : [1.36846584e+01 1.31534156e+00 1.06234298e-
```

Solution of the linear equation $3*x_0 + x_1 = 9$ & $x_0 + 2*x_1 = 8$ is

```
Multiplicative Inverse of arr3:
 [[-3.         2.        ]
 [ 2.66666667 -1.66666667]]
```

Topic _____ Date _____

Linear algebra on Matrices:

```
import numpy as np
```

a) Compute Dot & matrix (cross) product of two arrays

```
arr1 = np.array([[1,2,3],[4,5,6],[7,8,9]])
arr2 = np.array([[3,2,1],[6,5,4],[9,8,7]])
arr3 = np.array([[5,6],[8,9]])

DotProd = np.dot(arr1, arr2)
CrossProd = np.cross(arr1, arr2)
print("Dot Product of arr1 & arr2:\n", DotProd)
print("Cross Product of arr1 & arr2:\n", CrossProd)
```

17]

b) Compute the Eigen values of the matrix.

```
print("Eigen values of arr2:", np.linalg.eigvals(arr2))
```

[2 3]

c) Solve a linear equation $3*x_0 + x_1 = 9$, $x_0 + 2x_1 = 8$

```
A = np.array([[3,1],[1,2]])
B = np.array([9,8])
print("Solution of the linear eqn 3*x\u2080 + x\u2081 = 9
      & x\u2080 + 2*x\u2081 = 8 is", np.linalg.solve
      (A,B)).
```

d) Compute the multiplicative inverse of a matrix.

```
print("Multiplicative inverse of arr3:\n", np.linalg.inv(arr3
))
```

```
Rank of Matrix arr1 is: 2

Determinant of Matrix A is: 5
```

Topic _____ Date _____

e) compute the Rank of a Matrix

```
print("Rank of matrix arr1 is :", np.linalg.matrix_rank
(arr1))
```

f) Compute the determinant of a Matrix:

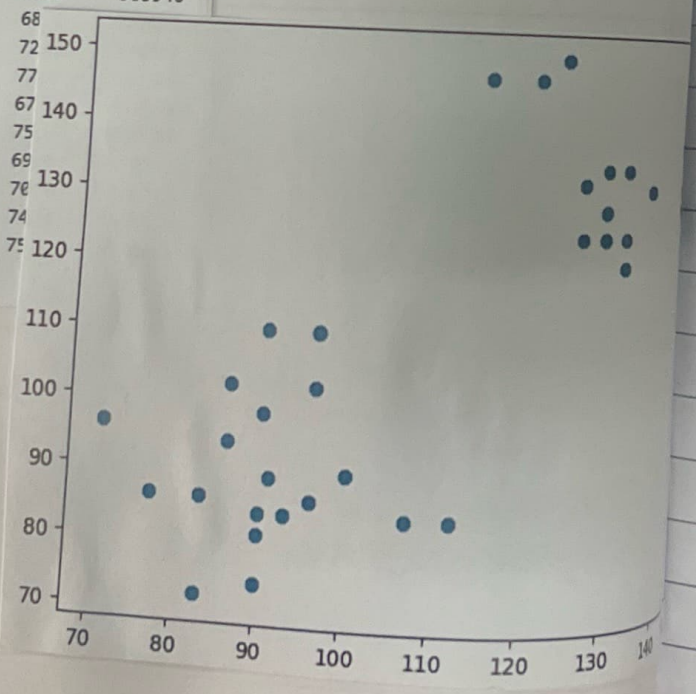
```
print("Determinant of Matrix A is:", np.linalg.det(A).
astype(np.int64)).
```

	id	Gender	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
0	1	Female	133	132	124	118	64.5	816932
1	2	Male	140	150	124	.	72.5	1001121
2	3	Male	139	123	150	143	73.3	1038437
3	4	Male	133	129	128	172	68.8	965353
4	5	Female	137	132	134	147	65.0	951545

```
Shape of the dataset is:  (40, 8)
Number of Columns in the dataset:  8
Number of Rows in the dataset:  40
Mean of FSIQ col in the dataset:  113
```

	id	Gender	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
0	1	Female	133	132	124	118	64.5	816932
4	5	Female	137	132	134	147	65.0	951545
5	6	Female	99	90	110	146	69.0	928799
6	7	Female	138	136	131	138	64.5	991305
7	8	Female	92	90	98	175	66.0	854258
10	11	Female	132	129	124	118	64.5	833868
13	14	Female	140	120	147	155	70.5	856472
14	15	Female	96	100	90	146	66.0	878897
15	16	Female	83	71	96	135	68.0	865363
16	17	Female	132	132	120	127	68.5	852244
18	19	Female	101	112	84	136	66.3	808020
22	23	Female	135	129	134	122	62.0	790619
24	25	Female	91	86	102	114	63.0	831772
26	27	Female	85	90	84	140	68.0	798612
28	29	Female	77	83	72	106	63.0	793549
29	30	Female	130	126	124	159	66.5	866662
30	31	Female	133	126	132	127	62.5	857782
34	35	Female	83	90	81	143	66.5	834344
35	36	Female	133	129	128	153	66.5	948066
37	38	Female	88	86	94	139	64.5	893983

	id	Gender	FSIQ	VIQ	PIQ	Weight	Height	MRI_Count
1	2	Male	140	150	124	.	72.5	1001121
2	3	Male	139	123	150	143	73.3	1038437
3	4	Male	133	129	128	172	68.8	965353
8	9	Male	89	93	84	134	66.3	904858
9	10	Male	133	114	147	172	68.8	955466
11	12	Male	141	150	128	151	70.0	1079549
12	13	Male	135	129	124	155	69.0	924059
17	18	Male	100	96	102	178	73.5	945088
19	20	Male	80	77	86	180	70.0	889083
20	21	Male	83	83	86	.	.	892420
21	22	Male	97	107	84	186	76.5	905940
23	24	Male	139	145	128	132	68	
25	26	Male	141	145	131	171	72	
27	28	Male	103	96	110	187	77	
31	32	Male	144	145	137	191	67	
32	33	Male	103	96	110	192	75	
33	34	Male	90	96	86	181	69	
36	37	Male	140	150	124	144	70	
38	39	Male	81	90	74	148	74	
39	40	Male	89	91	89	179	75	



Topic _____ Date _____

Practical-4

Write a python program to do the following operations
Data set: Brainsize.csv

```
import pandas as pd
import matplotlib.pyplot as plt
```

Loading data from csv file.

```
df = pd.read_csv("brainsize.csv", sep=";")
df.head()
```

b) Compute the basic statistics: shape, no. of columns, mean

```
print("Shape of the dataset is:", df.shape)
print("Nomber of columns in the dataset:", len(df.axes[1]))
print("Number of Rows in the dataset:", len(df.axes[0]))
print("Mean of FSIQ col in the dataset:", df["FSIQ"].mean())
```

c) Splitting a dataframe on values of a categorical variable

```
SD = df["Gender"].unique()
for gender in SD:
    print(df[df["Gender"] == gender])
```

d) Visualize data using scatter plot.

```
plt.scatter(df["VIQ"], df["PIQ"])
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
Dataset Description:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
Missing values per column:
 Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickne
Insulin
BMI
DiabetesPe
Age
Outcome
dtype: int

Number of outliers detected: 80
Outlier data points:
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
4              0      137             40             35      168  43.1
7             10      115              0              0        0  35.3
8              2      197             70             45      543  30.5
9              8      125             96              0        0   0.0
13             1      189             60             23      846  30.1
..           ...      ...            ...            ...      ...   ...
695            7      142             90             24      480  30.4
697            0       99              0              0        0  25.0
703            2      129              0              0        0  38.5
706           10      115              0              0        0   0.0
753            0      181             88             44      510  43.3

     DiabetesPedigreeFunction  Age  Outcome
4                       2.288   33        1
7                       0.134   29        0
8                       0.158   53        1
9                       0.232   54        1
13                      0.398   59        1
..                        ...  ...      ...
695                     0.128   43        1
697                     0.253   22        0
706                     0.304   41        0
753                     0.261   30        1
                        0.222   26        1

[80 rows x 9 columns]
```

Topic _____ Date _____

Practical-5

Write a python program to load the dataset & understand the input data.
Use Pima Indians Diabetes dataset

Load data, describe the given data & identify missing, outlier data items

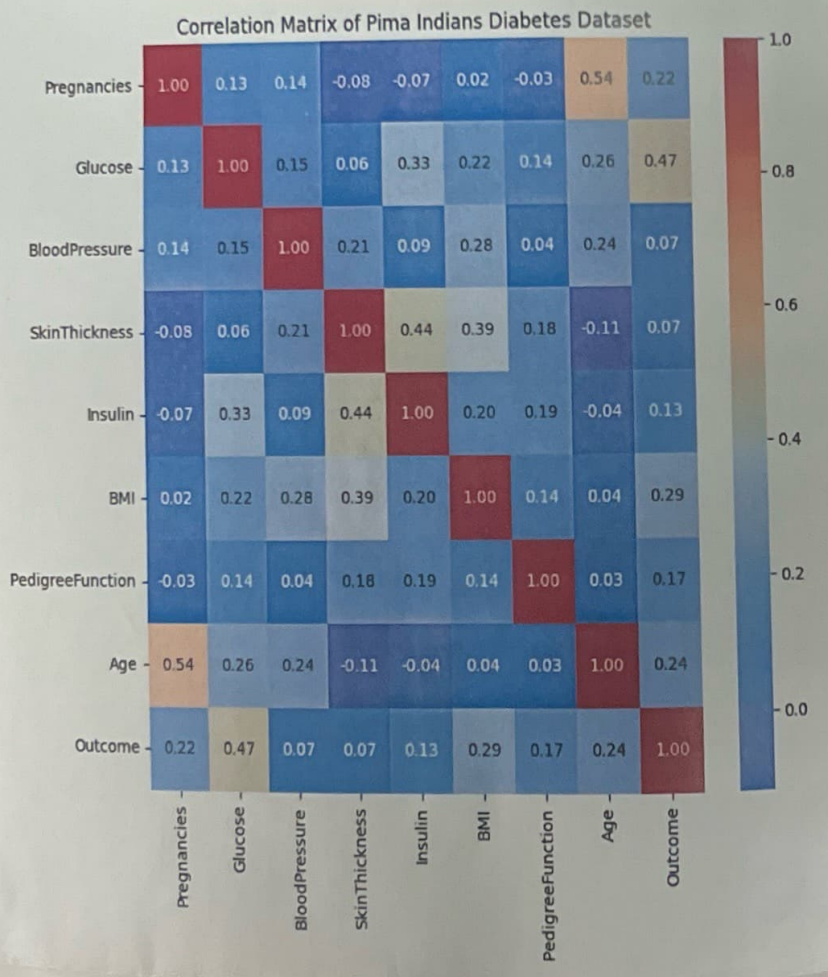
```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns.

df = pd.read_csv("diabetes.csv")
df.head()

print("Dataset description:\n",df.describe())
print("Missing values per column:\n",
                    df.isnull().sum())
z_scores = np.abs(stats.zscore(df))
outliers = (z_scores>3).any(axis=1)
print("\n Number of outliers detected:", outliers.
                                    count())

print("Outlier data points:\n", df[outliers])
```

Correlation Matrix:

	Pregnancies	Glucose	BloodPressure	SkinThickness
Pregnancies	1.000000	0.129459	0.141282	-0.081672
Glucose	0.129459	1.000000	0.152590	0.057328
BloodPressure	0.141282	0.152590	1.000000	0.207371
SkinThickness	-0.081672	0.057328	0.207371	1.000000
Insulin	-0.073535	0.331357	0.088933	0.436783
BMI	0.017683	0.221071	0.281805	0.392573
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928
Age	0.544341	0.263514	0.239528	-0.113970
Outcome	0.221898	0.466581	0.065068	0.074752

	Insulin	BMI	DiabetesPedigreeFunction	\
Pregnancies	-0.073535	0.017683	-0.033523	
Glucose	0.331357	0.221071	0.137337	
BloodPressure	0.088933	0.281805	0.041265	
SkinThickness	0.436783	0.392573	0.183928	
Insulin	1.000000	0.197859	0.185071	
BMI	0.197859	1.000000	0.140647	
DiabetesPedigreeFunction	0.185071	0.140647	1.000000	
Age	-0.042163	0.036242	0.033561	
Outcome	0.130548	0.292695	0.173844	

	Age	Outcome
Pregnancies	0.544341	0.221898
Glucose	0.263514	0.466581
BloodPressure	0.239528	0.065068
SkinThickness	-0.113970	0.074752
Insulin	-0.042163	0.130548
BMI	0.036242	0.292695
DiabetesPedigreeFunction	0.033561	0.173844
Age	1.000000	0.238356
Outcome	0.238356	1.000000



Date

Find the correlation among all attributes.

```
correlation_matrix = df.corr()
print("\n Correlation Matrix:\n",
                    correlation_matrix)
```

Visualize Correlation Matrix.

```
plt.figure(figsize=(8,8))
sns.heatmap(correlation_matrix, annot=True,
            Cmap="coolwarm", fmt=".2f")
plt.title("Correlation matrix of Pima Indians
           Diabetes Dataset")
plt.show()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

Practical-6

Data Preprocessing - Handling Missing Values

Write a python program to perform the following on given dataset

Remove rows/attributes:

```
import pandas as pd
df = pd.read_csv("titanic.csv")
df1 = df.copy()

df.isnull().sum()
df1.dropna(inplace=True)
```

Replace rows with missing value with mean or mode

```
Age_mean = df["Age"].mean().round()
Embarked_mode = df["Embarked"].mode()[0]
Cabin_mode = df["cabin"].mode()[0][0:3]

df["Age"].fillna(value=Age_mean, inplace=True)
df["Embarked"].fillna(value=Embarked_mode, inplace=True)
df["Cabin"].fillna(value=Cabin_mode, inplace=True)

df.isnull().sum()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_Range
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96	S	Adult
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	Adult
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96	S	Adult
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	Adult
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96	S	Adult

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_Range	Age_minmax
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96	S	Adult	0.271174
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	Adult	0.472229
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96	S	Adult	0.321438
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	Adult	0.434531
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96	S	Adult	0.434531
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Topic _____ Date _____

Perform transformation of data using Discretization (Binning) & Normalization (MinMaxScaler).

```
df["Age_Range"]= pd.cut(df['Age'], bins=[0,9,18,60,100],
                 labels=["child", "Teenager", "Adult", "Aged"])
df.head().

df['Age_minmax'] = (df['Age']-df['Age'].min())/
                   (df['Age'].max()- df['Age'].min())
df.
```

Date

Practical - 7

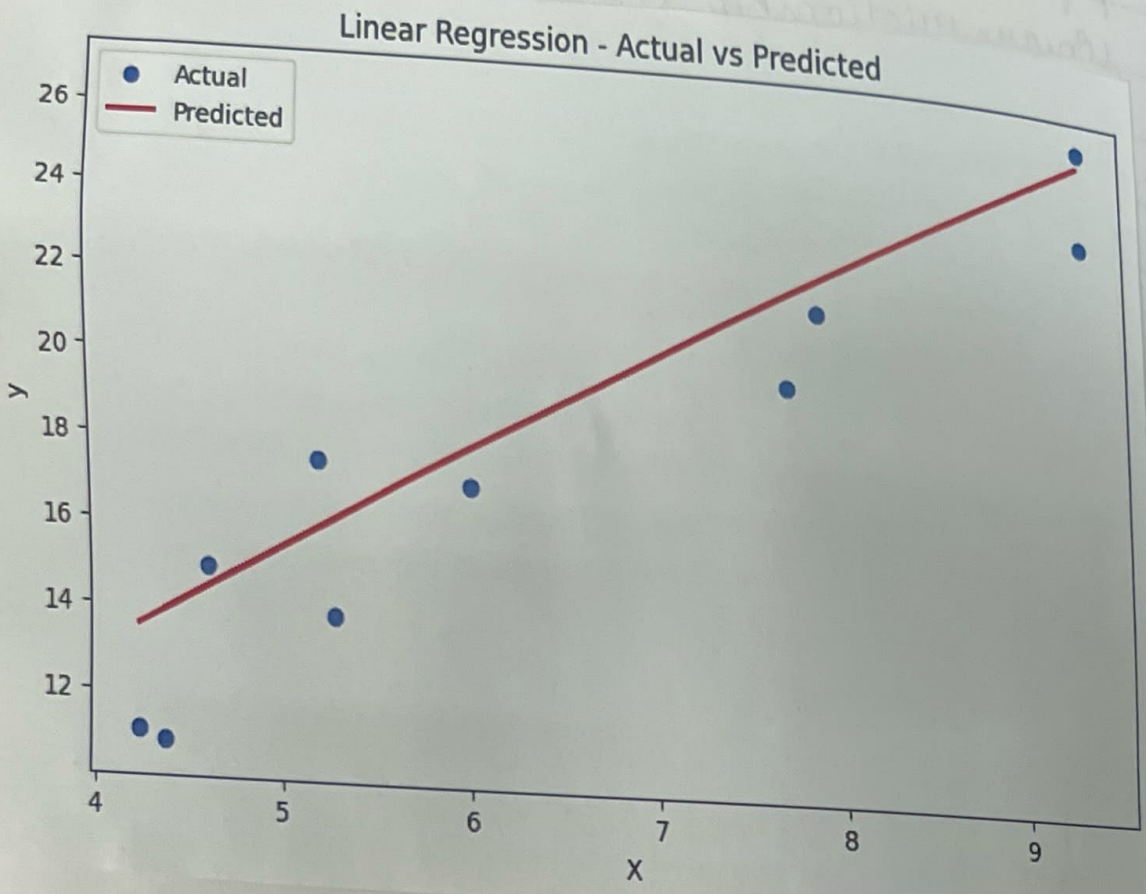
gression : Linear Regression, Logistic Regression

Linear Regression is one of the easiest & most popular machine learning algorithms. It shows a linear relationship b/w a dependent (y) & one or more independent variable (x).

Mathematically, it can be represented as:

$$y = a_0 + a_1 x + \epsilon$$

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression,
                                 LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, r2_score


np.random.seed(0)
X_lin = np.random.rand(50,1)*10
y_lin = 2.5 * np.squeeze() + 3 + np.random.randn(50) * 2

X_train_lin, X_test_lin, y_train_lin, y_test_lin = train_test_split
                                    (X_lin, y_lin, test_size=0.2,
                                     random_state = 0)
```


Linear Regression - Actual vs Predicted
Actual
Predicted
26
24
22
20
18
16
14
12
y
4
5
6
7
8
9
X

Topic _____ Date _____

```
linear_model = LinearRegression()
linear_model.fit(X_train_lin, y_train_lin)

y_pred_lin = linear_model.predict(X_test_lin)

plt.figure(figsize=(8,5))
plt.scatter(X_test_lin, y_test_lin, color="blue", label="Actual")
plt.plot(X_test_lin, y_pred_lin, color="red", linewidth=2,
         label="Predicted")
plt.xlabel("X")
plt.ylabel("y")
plt.title("Linear Regression - Actual vs Predicted")
plt.legend()
plt.show()
```

Topic

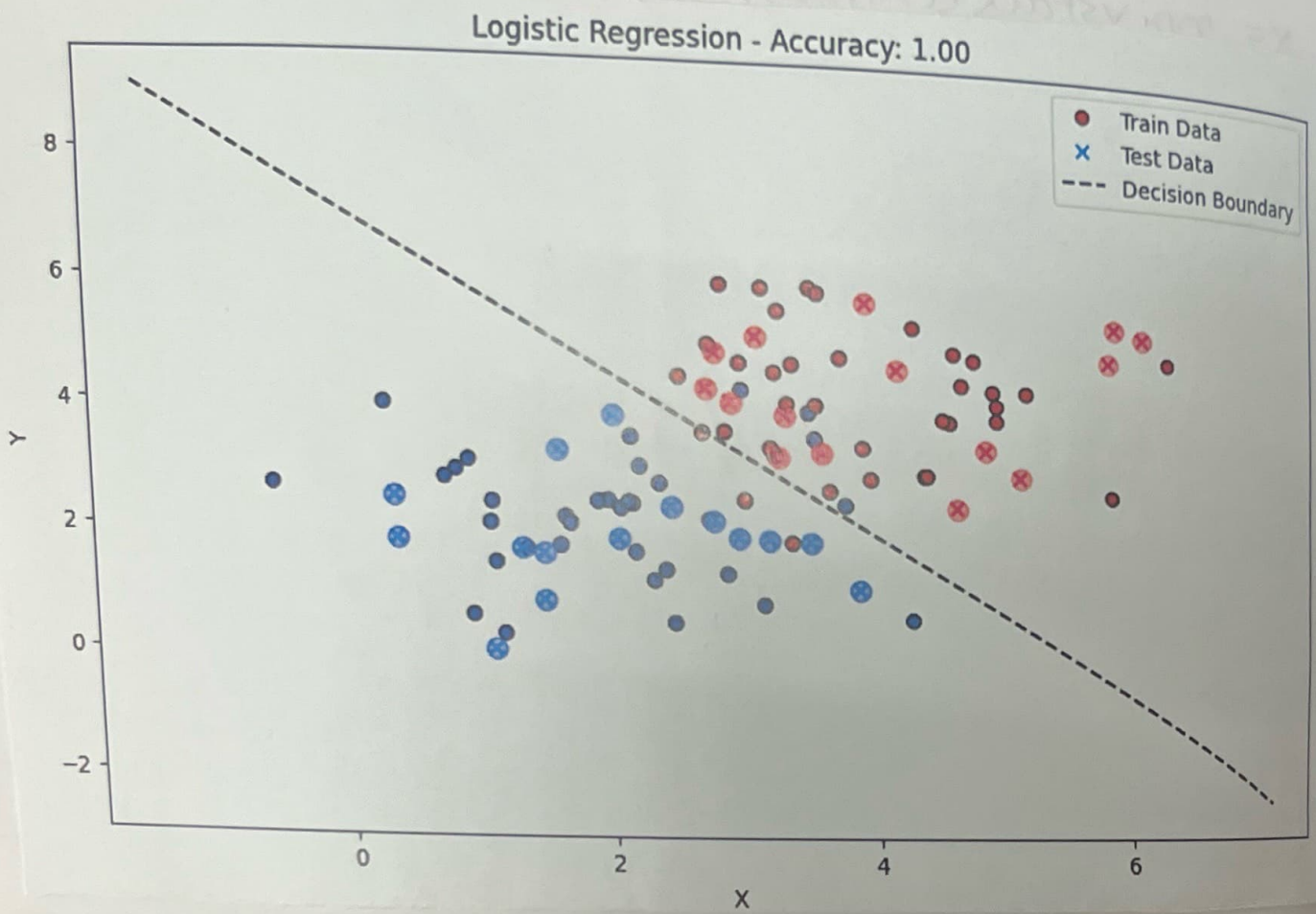
Date

Logistic Regression

Logistic Regression is one of the supervised learning technique, used for predicting the categorical dependent variable using a given set of independent variable.

Logistic Regression is similar to linear Regression except that how they are used. Linear Regression is used for solving Regression problem, whereas Logistic Regression is used for solving classification problems

```
np.random.seed(0)
X1 = np.random.normal(2,1,(50,2))
X2 = np.random.normal(4,1,(50,2))

X = np.vstack((X1,X2))
y = np.hstack((np.zeros(50), np.ones(50)))

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                   test_size=0.3, random_state=0)

model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred).
```


Logistic Regression - Accuracy: 1.00
Train Data
Test Data
Decision Boundary
8
6
4
2
0
−2
Y
0
2
4
6
X

Topic _____ Date _____

```
plt.figure(figsize=(10,6))
plt.scatter(X_train[:,0], X_train[:,1], c=y_train, cmap='bwr',
            edgecolor='k', label='Train Data')
plt.scatter(X_test[:,0], X_test[:,1], c=y_test, cmap='cool',
            edgecolor='k', marker='x', label='Test Data')

#plot decision boundary
x_values = np.linspace(min(X[:,0])-1, max(X[:0])+1, 100)
y_values = -(model.coef_[0][0] * x_values + model.intercept_[0])
           /model.coef_[0][1]

plt.plot(x_values, y_values, color='black', linstyle='--',
         label='Decision Boundary')

for i in range(len(y_pred)):
    color='red' if y_pred[i]==1 else 'blue'
    plt.scatter(X_test[i,0], X_test[i,1], color=color, marker='o',
                s=80, alpha=0.4)

plt.xlabel('X')
plt.ylabel('Y')
plt.title(f'Logistic Regression - Accuracy: {accuracy:.2f}')
plt.legend()
plt.show()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

Topic _____ Date _____

Practical-8

Apriori Algorithm:

The algorithm was proposed in 1994 by Rakesh Agarwal & Ramakrishnan Srikant. It finds the most fruquent itemsets or elements in a transaction database & identifies association rules between the items.

To construct association rules between elements or items, the algorithm considers 3 important factors which are support, confidence & lift.

```
import numpy as np
import pandas as pd
from mlxtend.fruquent_patterns import apriori,
                                     association_rules

data = pd.read_csv('online-retail.csv')
data.head()

#Data Cleaning
data['description'] = data['description'].str.strip()
data.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
data['InvoiceNo'] = data['InvoiceNo'].astype('str')
data = data[~data['InvoiceNo'].str.contain('C')]
```

BABA Design

Description	10 COLOUR SPACEBOY PEN	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE WOODLAND	20 DOLLY PEGS RETROSPOT	3 PIECE SPACEBOY COOKIE CUTTER SET	3 STRIPEY MICE FELTCRAFT	36 FOIL HEART CAKE CASES	...	WRAP FLOWER SHOP	WRAP GINGHAM ROSE	WRAP PAISLEY PARK
InvoiceNo														
536990	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
537246	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.0	0.0	...	0.0	0.0	0.0
537818	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
537915	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
538311	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
539353	0.0	0.0	0.0	0.0	24.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
540519	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
540546	24.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
541430	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
542147	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
544495	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
545191	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
545937	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
547444	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
1170	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET 12 COLOUR PENCILS SPACEBOY)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf	1.0
1171	(SET 12 COLOUR PENCILS SPACEBOY)	(SET 12 COLOUR PENCILS DOLLY GIRL)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf	1.0
1172	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET OF 4 KNICK KNACK TINS LONDON)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf	1.0
1173	(SET OF 4 KNICK KNACK TINS LONDON)	(SET 12 COLOUR PENCILS DOLLY GIRL)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf	1.0
1174	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET OF 4 KNICK KNACK TINS POPPIES)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	inf	1.0

Topic

Date

Transaction done in Portugal

```
basket_Por = (data[data['Country'] == 'Portugal']
              .groupby(['InvoiceNo', 'Description'])
              ['Quantity'].sum().unstack().reset_index()
              .fillna(0).set_index('InvoiceNo'))

basket_Por

def encode(x):
    if (x <= 0):
        return 0
    if (x >= 1):
        return 1
basket_encoded = basket_Por.applymap(encode)
basket_Por = basket_encoded

frq_items = apriori(basket_Por, min_support=0.05,
                    use_colnames=True)
rules = association_rules(frq_items, metric="lift",
                          min_threshold=1)
rules = rules.sort_values(['confidence', 'lift'],
                          ascending=[False, False])

rules.head()
```

Topic _____ Date _____

Practical-9

KNN - Use Iris dataset & write a python program to

a) Calculate Euclidean distance
b) Get Nearest Neighbours
c) Make Predictions

K-Nearest Neighbours is a supervised machine learning algorithm & finds introse applications in pattern recog & data mining.

KNN algorithm makes predictions by calculating similarities b/w the input sample & each training instance. This algorithm doesn't make assumptions, the algo is free to learn any functional form from the training data.

```
import numpy as np
from collections import Counter
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

iris = load_iris()
X = iris.data
y = iris.target
class_names = iris.target_names
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                   random_state=42)
```

Page _____ Date _____

#No. of Neighbours
K=3

#a) Calculate Euclidean distance.
```
def euclidean_distance(row1, row2):
    return np.sqrt(np.sum((row1-row2)**2))
```

#b) Get Nearest Neighbours
```
def get_nearest_neighbours(trainX, trainy, test_row, K):
    distances = []
    for i in range(len(trainX)):
        dist = euclidean_distance(test_row, trainX[i])
        distances.append((trainy[i], dist))
    distances.sort(key=lambda x: x[1])
    neighbours = [distances[i][0] for i in range(K)]
    return neighbours
```

#c) Make Predictions
```
def predict_classification(trainX, trainy, test_row, K):
    neighbours = get_nearest_neighbours(trainX, trainy, test
                                        row, K)
    most_common = Counter(neighbours).most_common(1)
    return most_common[0][0]
```

```
Test Sample 1: Actual class = versicolor, Predicted class = versicolor
Test Sample 2: Actual class = setosa, Predicted class = setosa
Test Sample 3: Actual class = virginica, Predicted class = virginica
Test Sample 4: Actual class = versicolor, Predicted class = versicolor
Test Sample 5: Actual class = versicolor, Predicted class = versicolor
Test Sample 6: Actual class = setosa, Predicted class = setosa
Test Sample 7: Actual class = versicolor, Predicted class = versicolor
Test Sample 8: Actual class = virginica, Predicted class = virginica
Test Sample 9: Actual class = versicolor, Predicted class = versicolor
Test Sample 10: Actual class = versicolor, Predicted class = versicolor
Test Sample 11: Actual class = virginica, Predicted class = virginica
Test Sample 12: Actual class = setosa, Predicted class = setosa
Test Sample 13: Actual class = setosa, Predicted class = setosa
Test Sample 14: Actual class = setosa, Predicted class = setosa
Test Sample 15: Actual class = setosa, Predicted class = setosa
Test Sample 16: Actual class = versicolor, Predicted class = versicolor
Test Sample 17: Actual class = virginica, Predicted class = virginica
Test Sample 18: Actual class = versicolor, Predicted class = versicolor
Test Sample 19: Actual class = versicolor, Predicted class = versicolor
Test Sample 20: Actual class = virginica, Predicted class = virginica
Test Sample 21: Actual class = setosa, Predicted class = setosa
Test Sample 22: Actual class = virginica, Predicted class = virginica
Test Sample 23: Actual class = setosa, Predicted class = setosa
Test Sample 24: Actual class = virginica, Predicted class = virginica
Test Sample 25: Actual class = virginica, Predicted class = virginica
Test Sample 26: Actual class = virginica, Predicted class = virginica
Test Sample 27: Actual class = virginica, Predicted class = virginica
Test Sample 28: Actual class = virginica, Predicted class = virginica
Test Sample 29: Actual class = setosa, Predicted class = setosa
Test Sample 30: Actual class = setosa, Predicted class = setosa

Accuracy of k-NN classifier: 100.00%
```

Topic _____ Date _____

```
# Evaluate the model
correct = 0
for i in range(len(X_test)):
    prediction = predict_classification(X_train, y_train,
                                        X_test[i], k)
    print(f"Test Sample {i+1}: Actual class = {class_name
          [y_test[i]]}, predicted class = {class_name[prediction]}")

    if prediction == y_test[i]:
        correct += 1

# Calculate Accuracy
accuracy = correct/len(X_test)
print(f"Accuracy of K-NN classifier: {accuracy*100:.2f}
      %").
```

Topic _____ Date _____

Practical-10

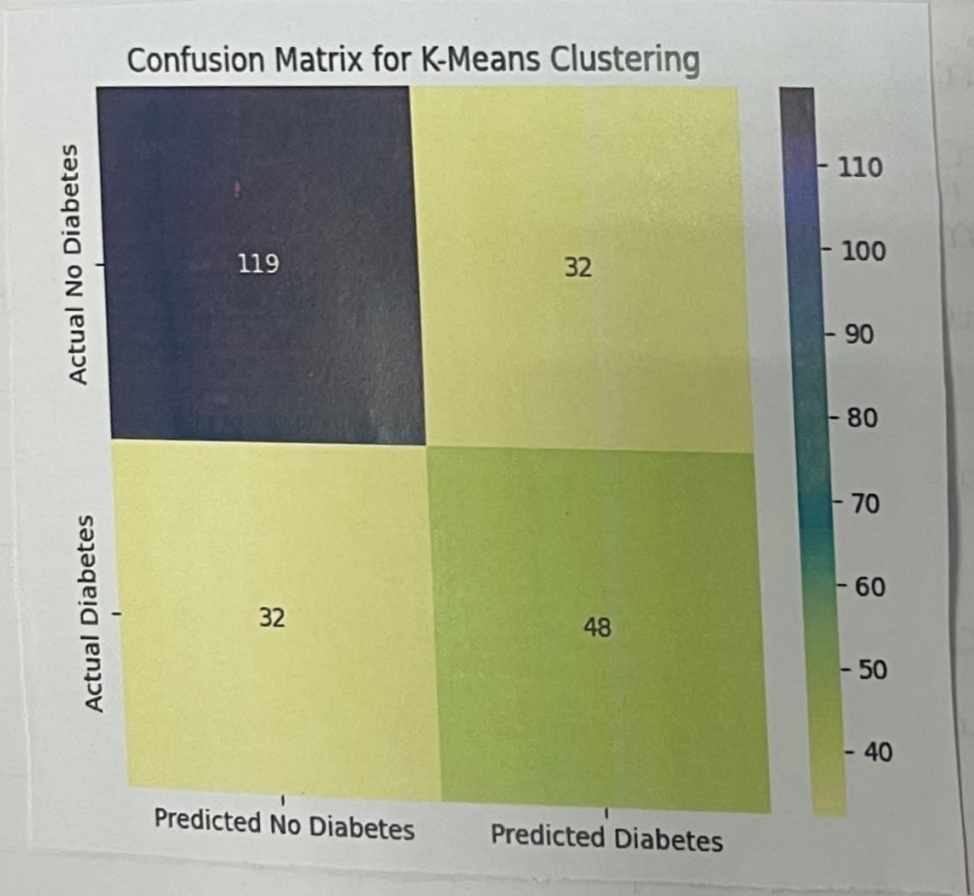
K-Means clustering

K-Means clustering is an unsupervised learning algorithm, which groups the unlabeled dataset into different clusters; Here K defines the no. of clusters that need to be created in the process.
It is a centroid based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distance b/w the data points & their corresponding clusters.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,
                            confusion_matrix

import seaborn as sns

data = pd.read_csv('diabetes.csv')
data.head()

X = data[['Glucose', 'BMI']].values
y = data['Outcome'].values
```

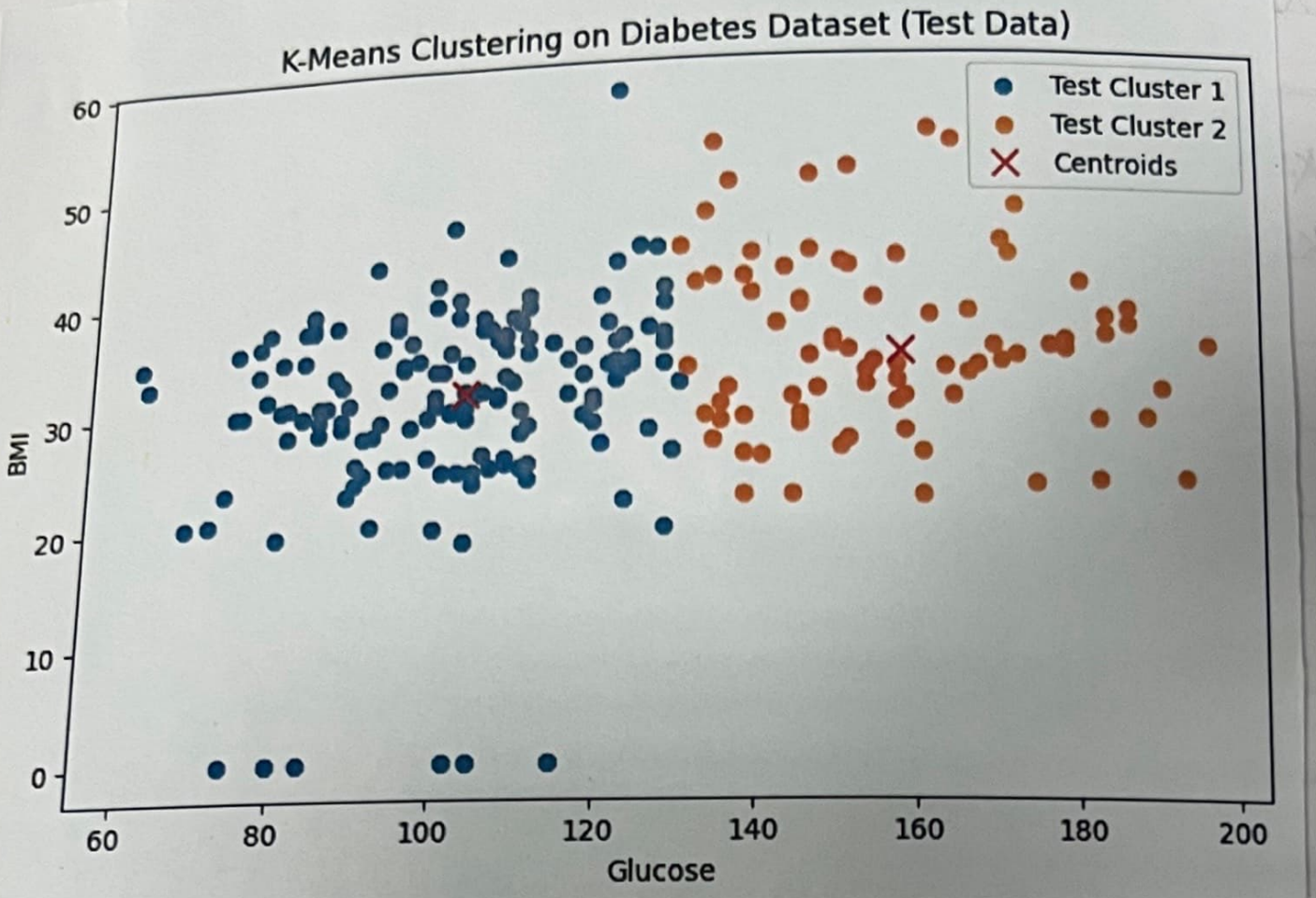
Classification report:

	precision	recall	f1-score	support
0	0.79	0.79	0.79	151
1	0.60	0.60	0.60	80
accuracy			0.72	231
macro avg	0.69	0.69	0.69	231
weighted avg	0.72	0.72	0.72	231



Topic _____ Date _____

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3,
                                   random_state=42)
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(X_train)

test_clusters = kmeans.predict(X_test)
centroids = kmeans.cluster_centers_

cluster_to_label = {0:1, 1:0} if confusion_matrix
        (Y_test, test_clusters)[0,0] < confusion_matrix
        (Y_test, test_clusters)[1,0] else {0:0, 1:1}
test_clusters = np.vectorize(cluster_to_label.get)(test_clusters)

print("Classification Report:\n\n", classification_report(
                                   Y_test, test_clusters))

plt.figure(figsize=(6,5))
sns.heatmap(confusion_matrix(Y_test, test_clusters),
            annot=True, fmt='d', cmap="YlGnBu",
            xticklabels=['Predicted No Diabetes',
                         'Predicted Diabetes']
            yticklabels=['Actual No Diabetes', 'Actual
                                               Diabetes']
plt.title("Confusion Matrix for K-means clustering")
plt.show()
```


K-Means Clustering on Diabetes Dataset (Test Data)
Test Cluster 1
Test Cluster 2
Centroids
BMI
Glucose
60
50
40
30
20
10
0
60
80
100
120
140
160
180
200

Topic _____ Date _____

```
plt.figure(figsize=(8,5))
for i in range(2):
    plt.scatter(X_test[test_cluster==i, 0],
                X_test[test_cluster==i, 1],
                label=f'Test Cluster {i+1}')
plt.scatter(centroids[:,0], centroids[:,1], color='red',
            marker='x', s=100, label='Centroids')
plt.xlabel("Glucose")
plt.ylabel("BMI")
plt.title("K-Means clustering on Diabetes dataset
                                    (Test Data)")
plt.legend()
plt.show()
```

	Variance	Skewness	Curtosis	Entropy	Class
0	3.62160	8.6661	-2.8073	-0.44699	0
1	4.54590	8.1674	-2.4586	-1.46210	0
2	3.86600	-2.6383	1.9242	0.10645	0
3	3.45660	9.5228	-4.0112	-3.59440	0
4	0.32924	-4.4552	4.5718	-0.98880	0

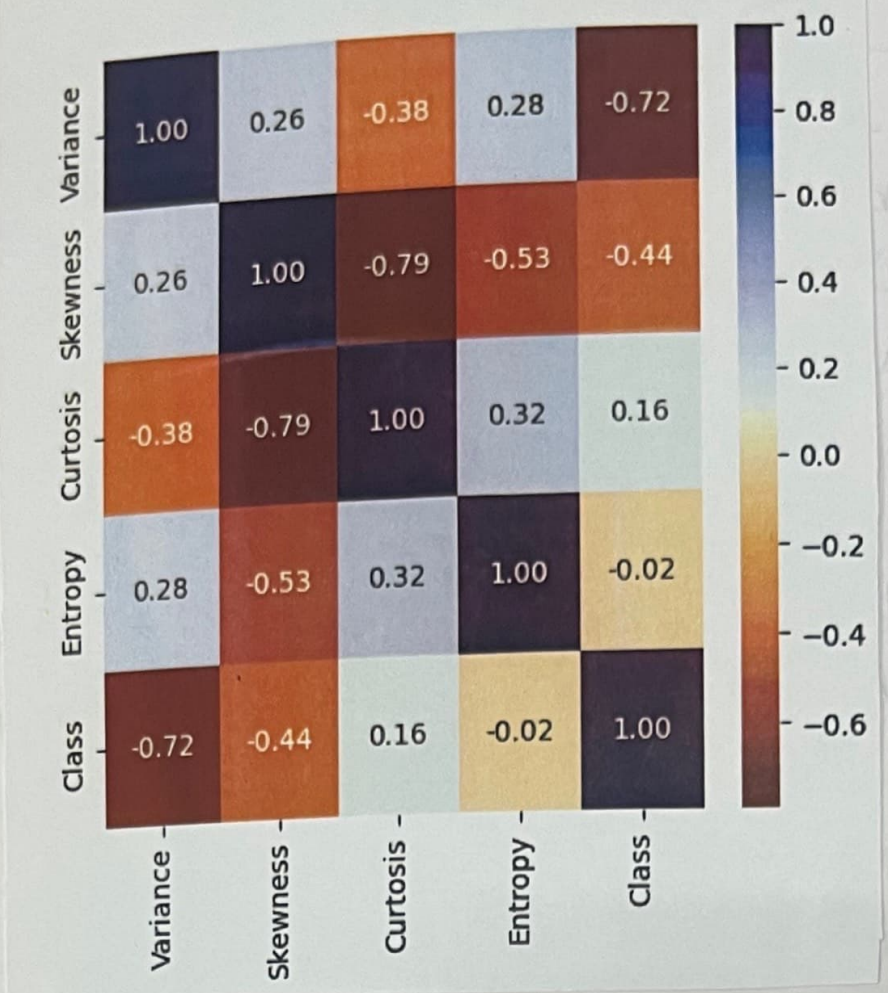
Topic _____ Date _____

Practical-11

Decision Tree Classification

Decision Tree is a supervised learning technique used for both classification & Regression problems. It is a tree structured classifier, where internal node represents the features of a dataset, branch represent the decision rule & each leaf node represents the outcome.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.tree import DecisionTreeClassifier,
                         plot_tree

from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,
                            accuracy_score.

df = pd.read_csv("bill_authentication.csv")
df.head()

# splitting
x = df.drop('class', axis=1)
y = df['class']
x.head()
y.head()
```



The Accuracy is 0.9927272727272727

	precision	recall	f1-score	support
0	1.00	0.99	0.99	157
1	0.98	1.00	0.99	118
accuracy			0.99	275
macro avg	0.99	0.99	0.99	275
weighted avg	0.99	0.99	0.99	275

Topic _____ Date _____

```
#Heatmap repn:
d = df.copy()
plt.figure(figsize=(5,5))
sns.heatmap(data = d.corr(), annot=True,
                    cmap='PuOr', fmt='.2f')
plt.show()

dt = DecisionTreeClassifier()
x1, x2, y1, y2 = train_test_split(x, y, test_size=0.2,
                    random_state = 0)
dt.fit(x1)
pred = dt.predict(x2)
plot_tree(dt).

print("The Accuracy is ", accuracy_report(y2, pred))
print(Classification_Report(y2, pred))
```