

**CLASS 12th PRACTICAL FILE**  
**ARTIFICIAL INTELLIGENCE**  
**2025-26**



**Arnav Thakur**

**Roll No:** \_\_\_\_\_

**School/Institution:**  
**The Maurya School**

**XII-B**

# INDEX

S. No	Practical	Page no.
1.	RMSE and MSE	2
2.	Confusion Matrix	3
3.	Orange Data Mining	4-6
4.	Data Storytelling	7
5.	Python Programs	8-12

### Q.1

Calculate MSE and RMSE values for the data given below using MS Excel.

Actual Values	Predicted Values
10	8
12	10
15	14
18	20
20	18
25	22
30	29
35	37

**Solution:**

A	B	C	D	E	F	G
1	Actual Values	Predicted Values	error sq			
2	10	8	4			
3	12	10	4			
4	15	14	1			
5	18	20	4			
6	20	18	4			
7	25	22	9			
8	30	29	1			
9	35	37	4			
10				MSE	RMSE	
				3.875	1.968502	

**where:**

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

$y_i - \hat{y}_i$  = actual value - predicted value = error

## Q. 2

Given below is the confusion matrix, calculate Precision, Recall, F1 score and Accuracy for it. ( You can do the work in MS Word.)

Confusion Matrix		Reality	
		Yes	No
Prediction	Yes	40	30
	No	10	120

### Solution:

We can conclude:

TP = 40	FP = 30
FN = 10	TN = 120

Therefore,

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN) = 0.8 = 80\%$$

$$\text{Precision} = TP/(TP+FP) = 0.5714$$

$$\text{Recall} = TP/(TP+FN) = 0.8$$

$$F1 \text{ Score} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) = 2 * 0.5714 * 0.8 / (0.5714 + 0.8)$$

$$F1 \text{ Score} = 0.66664$$

### Q. 3

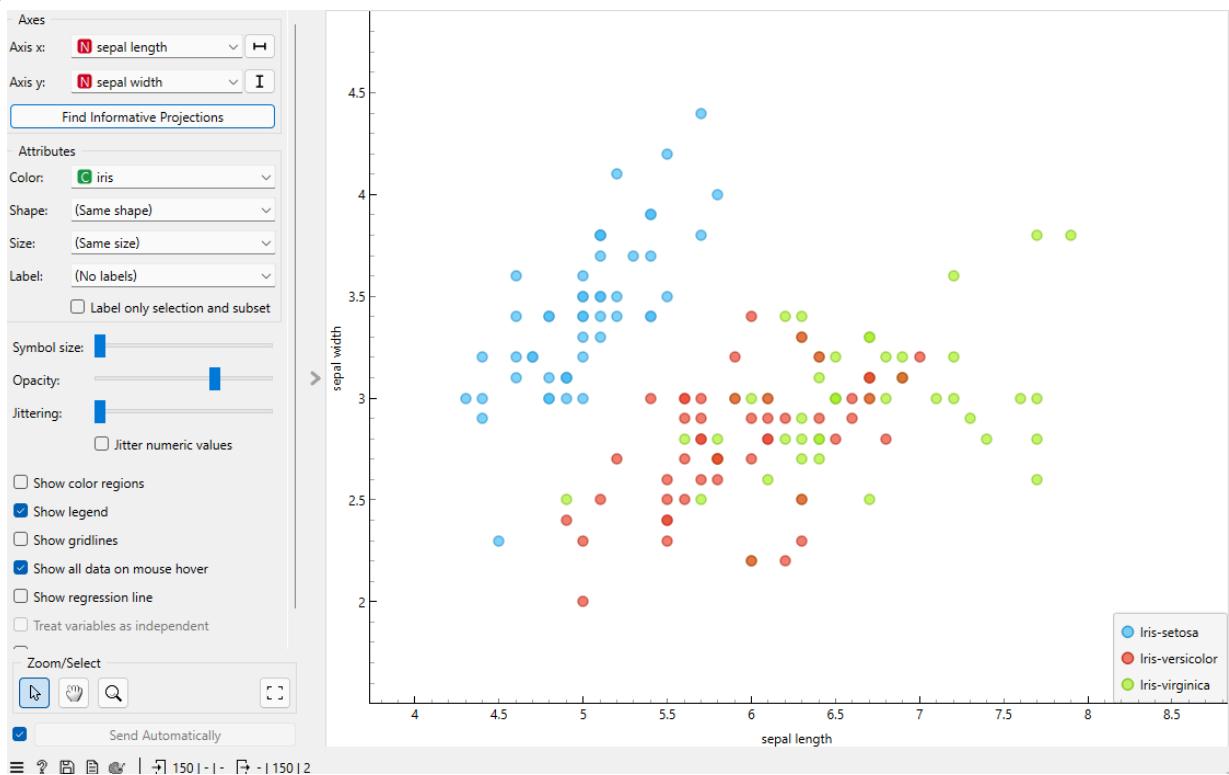
Use Orange Data Mining Tool to:

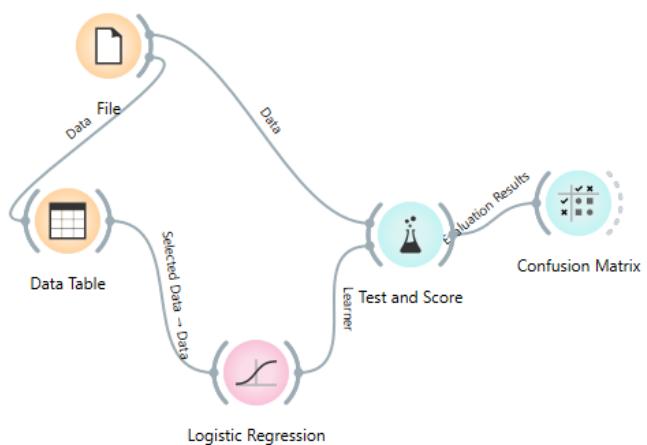
You can take datasets from Kaggle. (Take screenshot of each and every step)

- a. Using the Orange Data Mining Tool, visualize the relationship between "Petal Length" and "Petal Width" in the Iris dataset to show how different species are classified.
- b. Perform Classification using the Orange Data Mining Tool on the Titanic dataset. Predict whether a passenger survived based on features like age, sex, and class.
- c. Cluster images of birds and animals into distinct groups based on their visual characteristics.
  - Collect datasets of images containing various species of birds and animals.
  - Ensure that each dataset contains a sufficient number of images representing different species within the respective categories.
  - Import the collected image datasets into Orange Data Mining.
  - Apply clustering algorithms to group similar images together based on their numerical representations.
  - Analyze the clustering results and interpret the grouping of images into different clusters.
  - Identify any patterns or similarities observed within each cluster and between clusters.

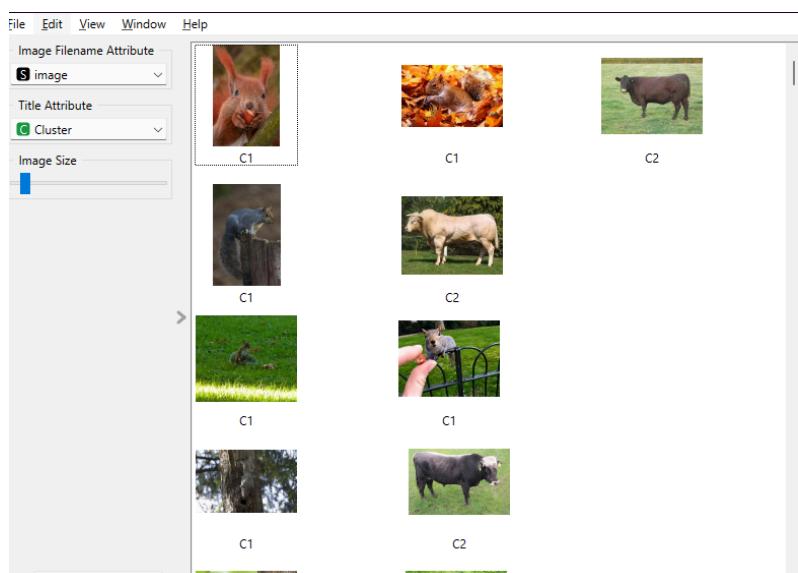
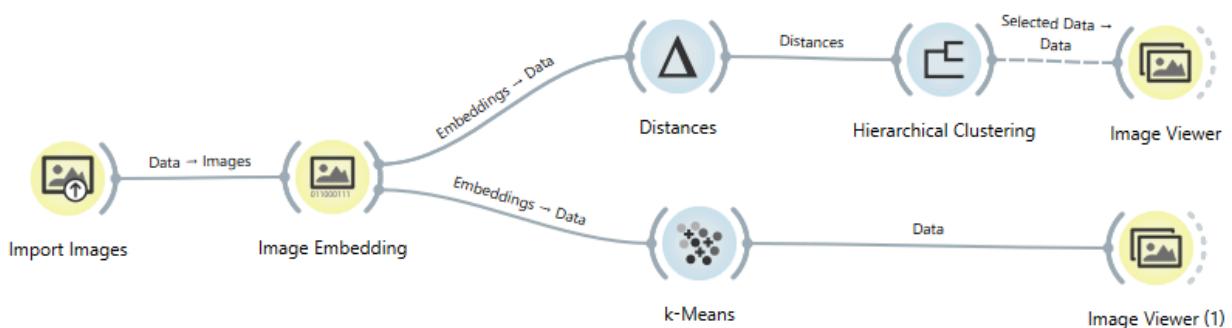
### Solution:

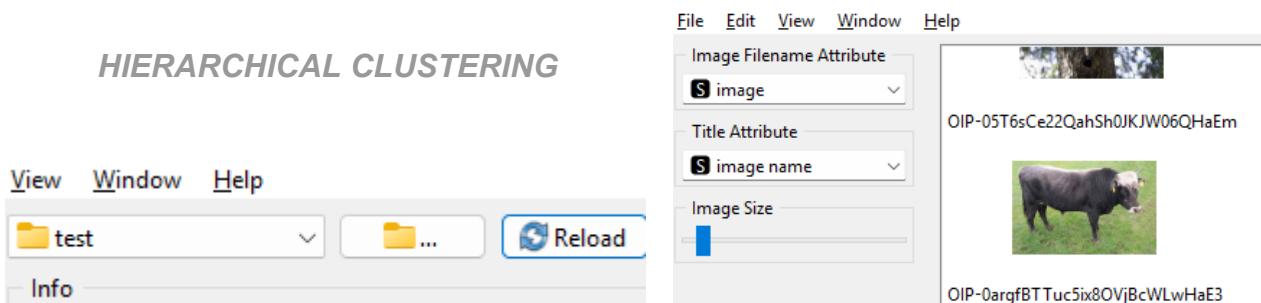
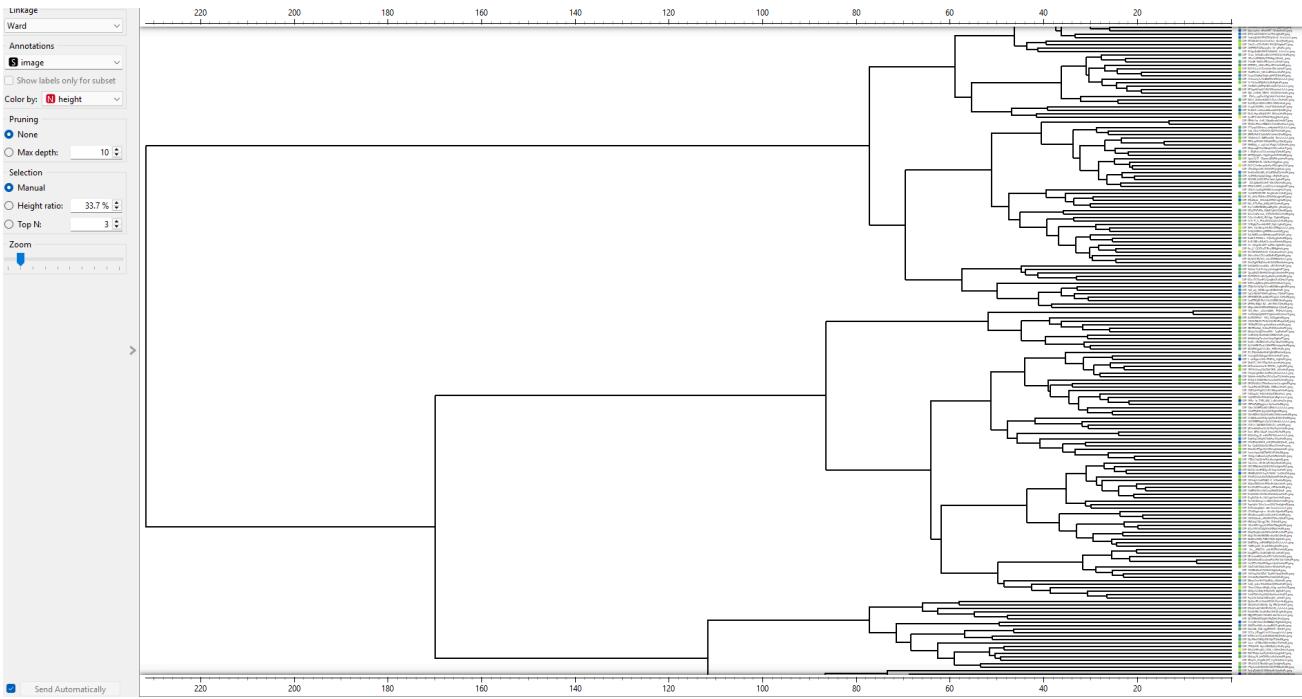
a)



**b)**

		Predicted		$\Sigma$
		no	yes	
Actual	no	4644	426	5070
	yes	1265	1155	2420
		$\Sigma$	1581	7490

**c)****K-means**



The images are grouped into different clusters both in Hierarchical clustering and K-means based on their visual features extracted through image embeddings.

**Cluster 1:** mainly contains images of smaller beings like birds and squirrels.

**Cluster 2:** contains bigger animals such as cows, sheep, dogs and cats.

This shows that the clustering algorithm successfully grouped images based on visual similarity.

**Q. 4** Create a Data story from the data given below:

Month	2022 Price (in Lacs)	2023 Price (in Lacs)	2024 Price (in Lacs)
January	240	252	264
February	244	256	268
March	248	260	272
April	252	264	276
May	256	268	280
June	260	272	284
July	264	276	288
August	268	280	292
September	272	284	296
October	276	288	300
November	280	292	304
December	284	296	308

**Solution:**



Prices show a steady rise across all months from 2022 to 2024. Each year displays a consistent monthly upward trend, reflecting stable growth. Year-over-year, prices for the same months climbed higher, reaching their peak in 2024. Without sharp fluctuations, the market proves predictable and healthy, underscoring continuous appreciation and robust long-term potential.

# Python Programs

## 1. Create a Pandas DataFrame using the following employee data:

Name	Age	Department	Salary
Alice	28	HR	50000
Bob	34	IT	65000
Charlie	25	Marketing	45000

- a) Display the DataFrame.
- b) Print the index of the DataFrame.
- c) Print the column names of the DataFrame.
- d) Display the data types of each column.
- e) Show the shape (number of rows and columns) of the DataFrame.

## *Output 1:*

```
a) DataFrame:  
      Name  Age Department  Salary  
0    Alice    28        HR   50000  
1     Bob    34        IT   65000  
2  Charlie    25  Marketing   45000  
b) Index of the DataFrame:  
RangeIndex(start=0, stop=3, step=1)  
c) Column names:  
['Name', 'Age', 'Department', 'Salary']  
d) Data types of each column:  
Name          object  
Age         int64  
Department    object  
Salary        int64  
dtype: object  
e) Shape (rows, columns):  
(3, 4)
```

## Code 1:

```
1 import pandas as pd
2
3 def main():
4     data = {
5         'Name': ['Alice', 'Bob', 'Charlie'],
6         'Age': [28, 34, 25],
7         'Department': ['HR', 'IT', 'Marketing'],
8         'Salary': [50000, 65000, 45000]
9     }
10
11     df = pd.DataFrame(data)
12
13     print("a) DataFrame:")
14     print(df)
15
16     print("b) Index of the DataFrame:")
17     print(df.index)
18
19     print("c) Column names:")
20     print(list(df.columns))
21
22     print("d) Data types of each column:")
23     print(df.dtypes)
24
25     print("e) Shape (rows, columns):")
26     print(df.shape)
27
28
29 main()
```

2. Write Python code to create a Pandas DataFrame using a dictionary containing data of your choice.

- a) Display the entire DataFrame.
- b) Display the first 5 records in the DataFrame.
- c) Display the last 10 records in the DataFrame.

## Code 2:

```
1 import pandas as pd
2
3 def main():
4     dict= {"Age" : [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],
5            | "weight" : [15,19,20,23,25,26,27,28,30,34,36,38,40,42,45],}
6
7     df = pd.DataFrame(dict)
8
9     print("a) DataFrame:")
10    print(df)
11
12    print("b) first five records of the DataFrame:")
13    print(df.head(5))
14
15    print("c) last ten records of the DataFrame:")
16    print(df.tail(10))
17
18 main()
```

## ***Output 2:***

a) DataFrame:

	Age	weight
0	1	15
1	2	19
2	3	20
3	4	23
4	5	25
5	6	26
6	7	27
7	8	28
8	9	30
9	10	34
10	11	36
11	12	38
12	13	40
13	14	42
14	15	45

b) first five records of the DataFrame:

	Age	weight
0	1	15
1	2	19
2	3	20
3	4	23
4	5	25

c) last ten records of the DataFrame:

	Age	weight
5	6	26
6	7	27
7	8	28
8	9	30
9	10	34
10	11	36
11	12	38
12	13	40
13	14	42
14	15	45

3. Write a Python program to predict the weight of a person using Linear Regression based on Height. You may use the Kaggle “Height and Weight Dataset”.

### Code 3:

```
1 import numpy as np
2 import pandas as pd
3 from sklearn.linear_model import LinearRegression
4 from sklearn.model_selection import train_test_split
5
6
7 def main():
8     filename = "height_weight.csv"
9
10    df = pd.read_csv(filename)
11    X = df[['Height(Inches)']].values
12    y = df['Weight(Pounds)'].values
13
14    # Split data
15    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
16
17    # Train model
18    model = LinearRegression()
19    model.fit(X_train, y_train)
20
21    # Take input from user
22    height = float(input("Enter height (inches): "))
23
24    new_data = np.array([[height]])
25    prediction = model.predict(new_data)
26    pr_weight = prediction[0]*0.454
27    print(f"Predicted weight:", pr_weight, "kg")
28
29
30
31 while True:
32     main()
```

### Output 3:

```
Enter height (inches): 67
Predicted weight: 56.312595214409036 kg
```