Name Supriya. S Sub. ML Lab

Std.: 6th Sem Div. 'E' Sec Roll No. 1BM22CS350

Telephone No. E-mail ID.

Blood Group. Birth Day.

Sr.No.	Title	Page No.	Sign./Remarks
1.	Different ways of importing datasets	1	
2.	Different data preprocessing techniques.	3	
3.	IDB implementation on weather dataset	5	
4.	Linear Regression	8	
5.	Logistic Regression	13	
6.	KNN Regression	15	
7.	Random Forest	17	
8.	Ada Boost	22	
9.	K Means Clustering	24	
10.	PCA	26	

-3-25

Lab - 0

1. To Do Exercise
Four different ways of importing datasets
1) Initializing values directly into DataFrame

```
df = pd.DataFrame({'USN': ['450', '451', '452',
                           '453', '454'],
                   'Name': ['Sita', 'Rita', 'Meena', 'Manya',
                            'Keerthi'],
                   'Marks': [94, 89, 87, 86, 84]
                   })
print(df)
```

2) Importing datasets from sklearn.datasets

```
from sklearn.datasets import load_diabetes
diabetes = load_diabetes()
df = pd.DataFrame(diabetes.data, columns=
                  diabetes.feature_names)
df['target'] = diabetes.target
print(df)
```

3) Importing datasets from a specific .csv file

```
file = 'sample_sales_data.csv'
df = pd.read_csv(file)
print(df.head())
```

4) Downloading datasets from existing dataset repositories like Kaggle, UCI, etc

```
file = 'Diabetes.csv'
df = pd.read_csv(file)
print(df.head())
```

2.

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

2 tickers = ["HDFCBANK.NS", "ICICIBANK.NS",
"KOTAKBANK.NS"]

data = yf.download(tickers, start="2024-01-
end="2024-12-30",
group_by='ticker')

hdfc_data = data['HDFCBANK.NS']
hdfc_data['Daily Return'] = hdfc_data['Close'].
pct_change()

icici_data = data['ICICIBANK.NS']
icici_data['Daily Return'] = icici_data['Close'].
pct_change()

kotak_data = data['KOTAKBANK.NS']
kotak_data['Daily Return'] = kotak_data
['Close'].pct_change()

plt.figure(figsize=(12,6))
plt.subplot(2,1,1)
hdfc_data['Close'].plot(title="Closing Price")
icici_data['Close'].plot(title="Closing Price")
kotak_data['Close'].plot(title="Closing
Price")

plt.subplot(2,1,2)
hdfc_data['Daily Return'].plot(title
="Daily Returns")
icici_data['Daily Return'].plot(title
="Daily Returns")
kotak_data['Daily Return'].plot(title="Daily
Returns")
plt.tight_layout()
plt.show()

Lab - 1

1. housing.csv : Demo of various data pre-processing technique

```
import pandas as pd
df = pd.read_csv("housing.csv")
print("Information of all columns:")
print(df.info())
print("\nStatistical information of all numerical columns:")
print(df.describe())
print(df['Ocean Proximity'].value_counts
missing_values = df.isnull().sum()
col = missing_values[missing_values>0]
print(col)
```

2. Diabetes dataset

1. Which columns in the dataset had missing values? How did you handle them?

None of the columns had missing values but if were present then numerical column's NaN can be replaced with median and categorical column's null can be replaced with mode.

2. Which categorical columns did you identify in the dataset? How did you encode them?

The diabetes dataset had gender, and class as categorical column and adult dataset had workclass

education', marital-status, occupation relationship, race, gender, native-country and income as categorical column.

Using Ordinal Encoder we can encode categorical columns.

3. What is the difference b/w Min-Max Scaling and Standardization? When would you use one over the other?

Min-Max Scaling :- transforms data to fit with a specific range (usually 0 to 1)

Standardization scales data by subtracting mean and dividing by S.D. with mean as 0 and S.D as 1

Use Min-Max scaling when you need to preserve the relative relationship b/w values within a feature and use standardization when your algorithm is sensitive to outliers or assumes a normal distribution.

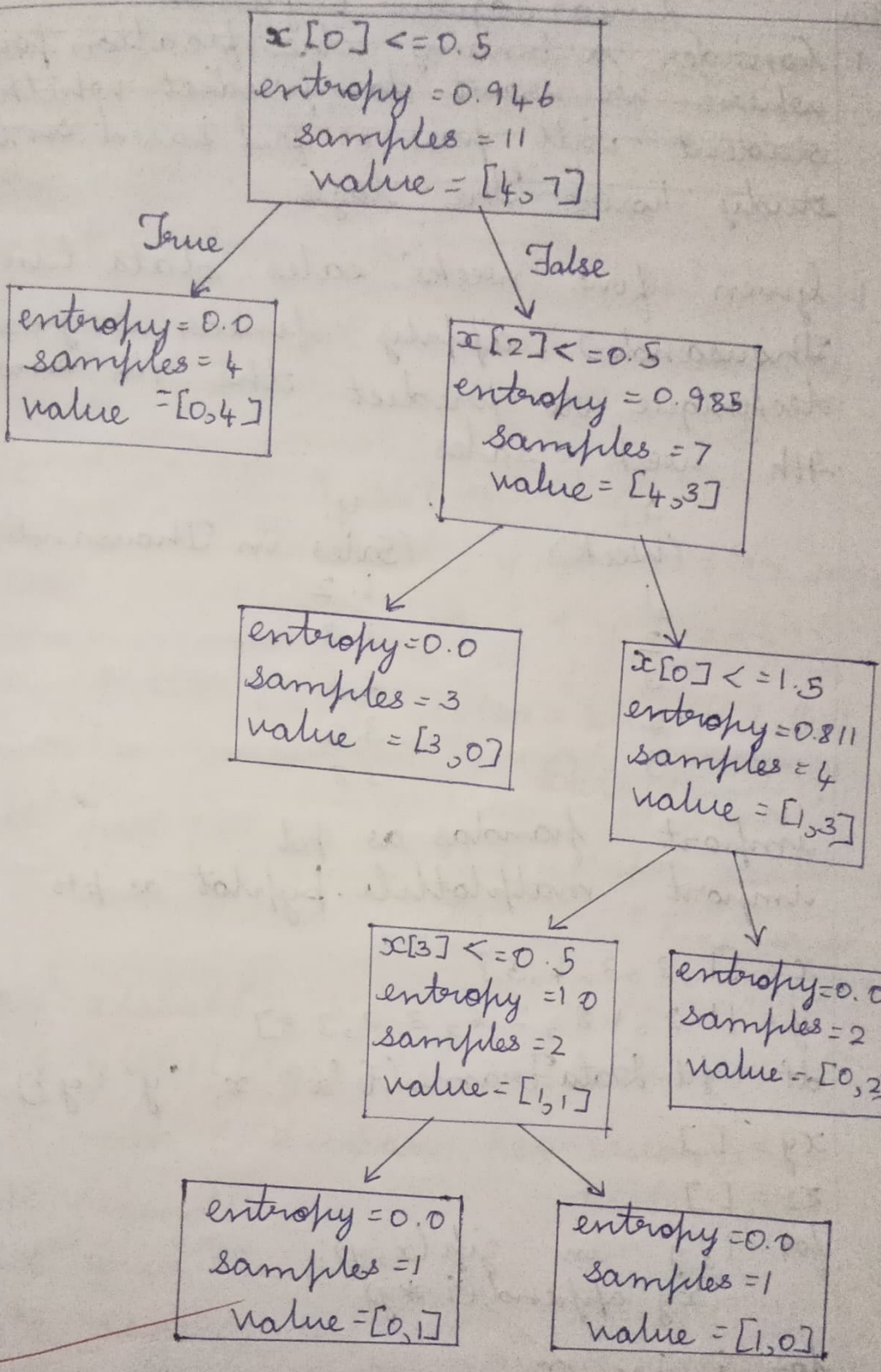
-3-25

Lab No. 2

Implement ID3 (Iterative Dichotomiser 3) algorithm on Tennis Weather dataset

```
import pandas as pd
data = pd.read_csv('tennis.csv')
print(data)

from sklearn.preprocessing import LabelEncoder
outlook = LabelEncoder()
temp = LabelEncoder()
humidity = LabelEncoder()
wind = LabelEncoder()
play = LabelEncoder()
data['outlook'] = outlook.fit_transform
                  (data['outlook'])
data['temp'] = temp.fit_transform
               (data['temp'])
data['humidity'] = humidity.fit_transfo
                   (data['humidity'])
data['windy'] = wind.fit_transform
                (data['windy'])
data['play'] = play.fit_transform
               (data['play'])
print(data)
features_cols = ['outlook', 'temp', 'humidity
                 'windy']
x=data[features_cols]
y=data.play
print(x)
print(y)
```

```
from sklearn.model_selection import
                    train_test_split

x_train, x_test, y_train, y_test =
        train_test_split(x, y, test_size = 0.2)

from sklearn.tree import
        DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion
                    = 'entropy')
classifier.fit(x_train, y_train)
classifier.predict(x_test)
classifier.score(x_test, y_test)
clf = DecisionTreeClassifier(criterion
                    = 'entropy')
clf.fit(x, y)
importances = clf.feature_importances_
feature_importance_df = pd.DataFrame
  ({'Feature': x.columns,
    'Information Gain': importances
  })
feature_importance_df = feature_
  importance_df.sort_values(by=
                'Information Gain',
                ascending = False)
print(feature_importance_df)
from sklearn import tree
tree.plot_tree(classifier)
```



	Feature	Information Gain
0	outlook	0.362629
3	windy	0.274205
2	humidity	0.211237
	temp	0.151929

Tue 12/3/25

Lab-3

19-3-25

Linear Logistic Regression

+. Consider a binary classification problem where we want to predict whether a student will pass or fail based on their study hours. The logis

1. Given five weeks' sales data (in Thousands), apply linear regression technique to predict the 7th and 9th week sales.

x_i (Week)	y_i (Sales in Thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

```
import pandas as pd
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [1.2, 1.8, 2.6, 3.2, 3.8]
df = pd.DataFrame({'x': x, 'y': y})
xy = []
x2 = []
for i, j in zip(x, y):
    xy.append(i * j)
for i in x:
    x2.append(i**2)
xmean = sum(x) / len(x)
ymean = sum(y) / len(y)
xy_mean = sum(xy) / len(xy)
x2_mean = sum(x2) / len(x2)
```

```
a1 = (xy_mean - x_mean * y_mean) /
     (x2_mean - x_mean ** 2)
a0 = y_mean - a1 * x_mean
print(f"a0 = {a0}, a1 = {a1}")
x_input = int(input("Enter the
          value of x: "))
ypred = a0 * a1 * x_input
print(f"Predicted y for x = {x_input}
          is : {y_pred}")
plt.scatter(x, y, color='blue', label
          = 'Data Points')
plt.plot(x, [a0 + a1 * xi for xi in x],
     color='red', label='Regression
          line')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Regression')
plt.legend()
plt.grid(True)
plt.show()
```

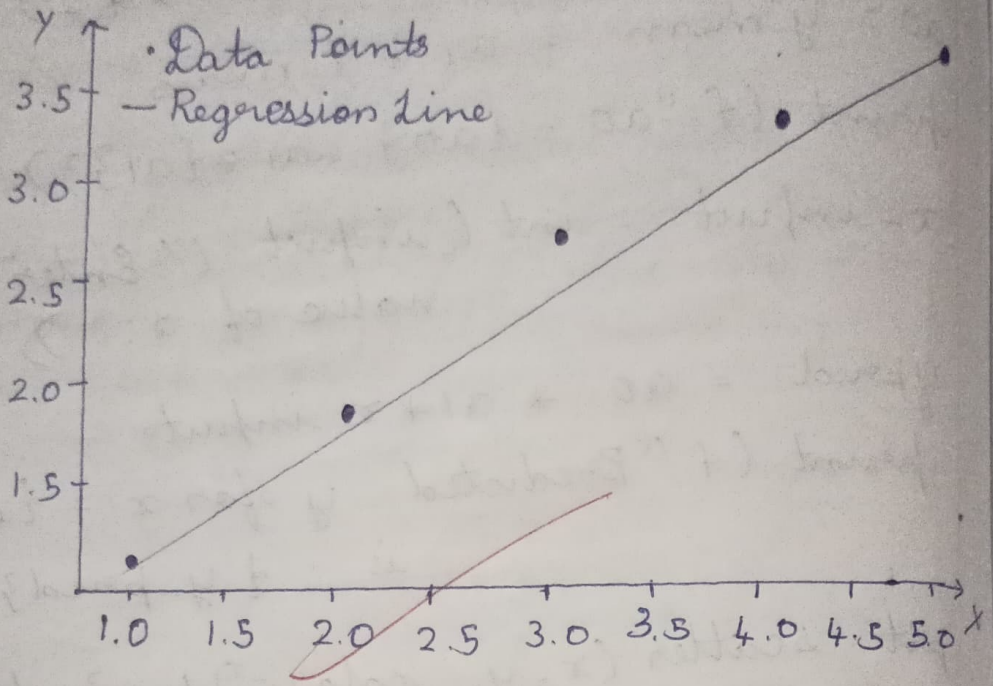
O/P:-

a0 = 0.54, a1 = 0.65

Enter the value of x: 7

Predicted y for x = 7 is : 5.16

Linear Regression



2. Find linear regression of the data of week & product sales. Use matrix approach for finding Linear Regression

X_i (week)	Y_i (sales in thousand)
1	1
2	3
3	4
4	8

```
import numpy as np
import matplotlib.pyplot as plt
def matrix (x, y, n):
    ax = np.ones ((n, 2))
    ax [:, 1] = x
    ay = np.array (y). reshape (-1, 1)
```

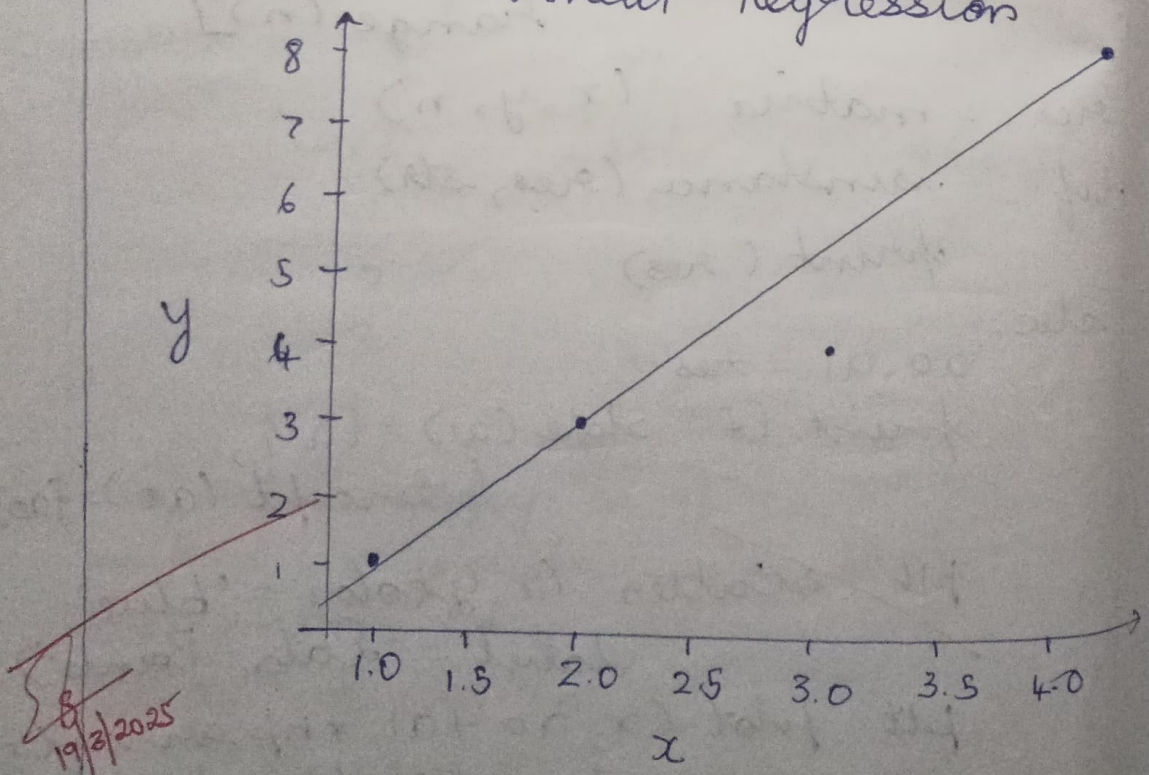
```
xt = ax.T
A = np.dot(xt, ax)
det_A = np.linalg.det(A)
if det_A != 0:
    A_inv = np.linalg.inv(A)
    a = np.dot(A_inv, np.dot(xt, ay))
    return a.flatten()
else:
    return "Inverse doesn't exist"

n = int(input())
x = [int(input(f"Enter ")) for i in range(n)]
y = [int(input()) for i in range(n)]
res = matrix(x, y, n)
if isinstance(res, str):
    print(res)
else:
    a0, a1 = res
    print(f"Slope (a1) = {a1}, Intercept (a0) = {a0}")
    plt.scatter(x, y, color='blue', label='Data Points')
    plt.plot(x, a0 + a1 * np.array(x), color='red', label='Regression Line')
```

```
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Regression')
plt.legend()
plt.grid(True)
plt.show()
```

O/P:-

```
4
1
2
3
4
1
3
4
8
```

Slope (a1) = 2.2 , Intercept (a0) = 1.



19/3/2025

4-28

Lab - 5

Logistic Regression

1. Consider a binary classification problem where we want to predict whether a student will pass or fail based on their study hours. The logistic regression model has been trained, and the learned parameters are $a0 = -5$ (intercept) and $a1 = 0.8$ (coefficient for study hours).

a. Write the logistic regression equation for this problem.

$$P(y=1|x) = \frac{1}{1+e^{-(a_0+a_1x)}} = \frac{1}{1+e^{-(-5+0.8x)}}$$

b. Calculate the probability that a student who studies for 7 hours will pass

$$P=(y=1|7) = \frac{1}{1+e^{-(-5+0.8(7))}} = \frac{1}{1+e^{-0.6}}$$

$$\simeq 0.6457 \text{ or } 64.57\%$$

c. Determine the predicted class (pass or fail) for this student based on a threshold of 0.5

Since $P=(y=1|7) = 64.57\%$.

above 0.5 or 50%.

predicted class is pass.

2. Consider $z = [2, 1, 0]$ for three classes. Apply SoftMax function to find the probability values of three classes.

$$P(y = i \mid z) = \frac{e^{z_i}}{\sum_j e^{z_j}} \qquad z = [2, 1, 0]$$

is the SoftMax function

$z_1 = 2 \quad z_2 = 1 \quad z_3 = 0$

$$P(y = 1 \mid z) = \frac{e^{z_1}}{e^{z_3} + e^{z_1} + e^{z_2}} = \frac{e^2}{e^{2+1+0}} = \frac{e^2}{e^2 \cdot e^1 \cdot e^0} = 7.38 / 11.10$$

$$\approx 0.665$$

$$P(y = 2 \mid z) = \frac{e^{z_2}}{e^{z_3} + e^{z_1} + e^{z_2}} = \frac{e^1}{e^{0+2+1}} = \frac{e^1}{e^0 \cdot e^2 \cdot e^1} = 2.78 / 11.107$$

$$\approx 0.245$$

$$P(y = 3 \mid z) = \frac{e^{z_3}}{e^{z_3} + e^{z_1} + e^{z_2}} = \frac{e^0}{e^0 + e^2 + e^1}$$

$$= \frac{1}{11.107} \approx 0.090$$

Lab-6

Build KNN Classification model for ~~dataset~~ a given dataset

1. Consider the following dataset, for k = 3 and test data (X, 35, 100) as (Person, Age, SalaryK) solve using Knn classifier model and predict the target.

Person	Age	SalaryK	Target	Euclidean Distance
A	18	50	N	$\sqrt{(35-18)^2+(100-50)^2}$ = 52.8
B	23	55	N	46.6
C	24	70	N	31.9
D	41	60	Y	40.5
E	43	70	Y	31.1
F	38	40	Y	60.1
X	35	100	?	

$$d = \sqrt{(Age_1 - Age_2)^2 + (SalaryK_1 - SalaryK_2)^2}$$

First 3 as K = 3

(41, 60)	31.1	Y
(24, 70)	31.9	N
(41, 60)	40.5	Y

Since majority is Y

(35, 100) Target is Y

2. For Iris dataset
How to choose the k value? Demonstrate using accuracy rate and error rate

1. Split the data into training and testing sets.
2. Train the model with various values of K (e.g. 1, 3, 5, 7, 9)

3. Evaluate accuracy and error rate for each k.

4. Plot accuracy = $\frac{\text{Number of correct predictions}}{\text{Total prediction}}$

error rate = 1 - accuracy.

4. Plot accuracy vs k or error rate vs k and select k with the highest accuracy or lowest error rate.

For diabetes dataset

What is the purpose of feature scaling? How to perform it?

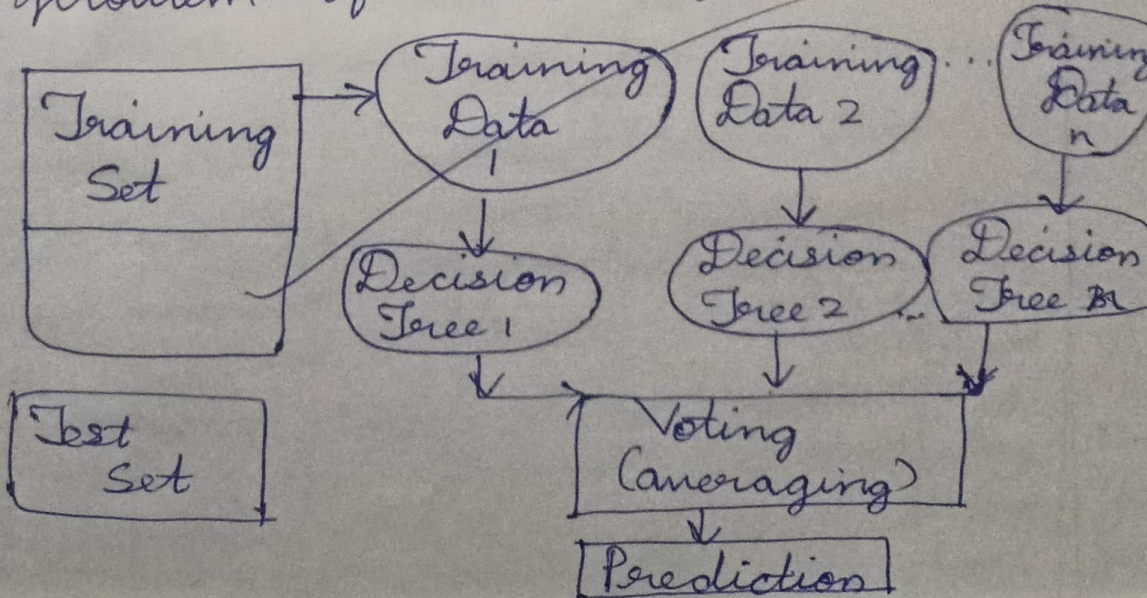
- KNN uses distance, so features with larger values dominate.
Scaling ensures fairness.
Using standardization (z-score normalization).

2/4/2025

16-4-25

Random Forest

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.
- Can be used for both Classification and Regression problems in ML.
- Based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Random Forest algorithm

1. Select random K data points from the training set.
2. Build the decision trees associated with the selected data points (Subsets)
3. Choose the number N for decision trees that you want to build.
4. Repeat 1 & 2
5. For new data points, find the predictions of each decision tree and assign the new data points to the category that wins the majority votes.

16.0

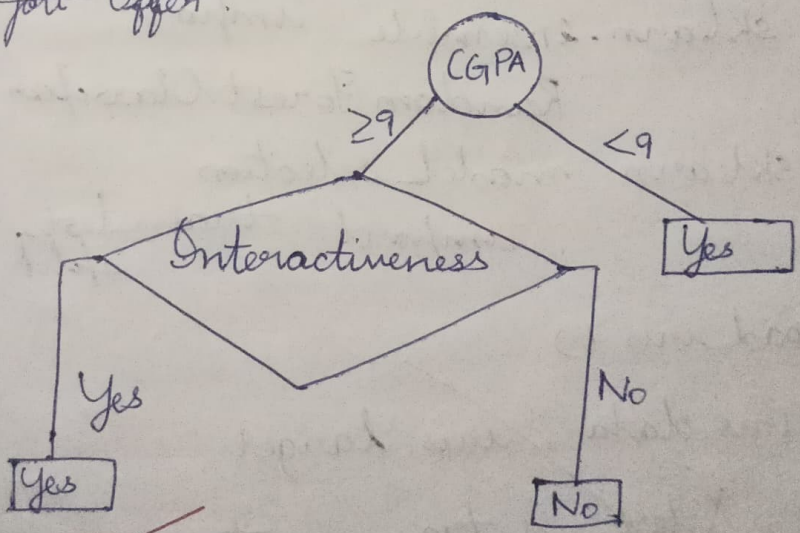
Implement Random Forest ensemble method.

Draw the Decision tree considering CGPA as root node

Sample - S_1

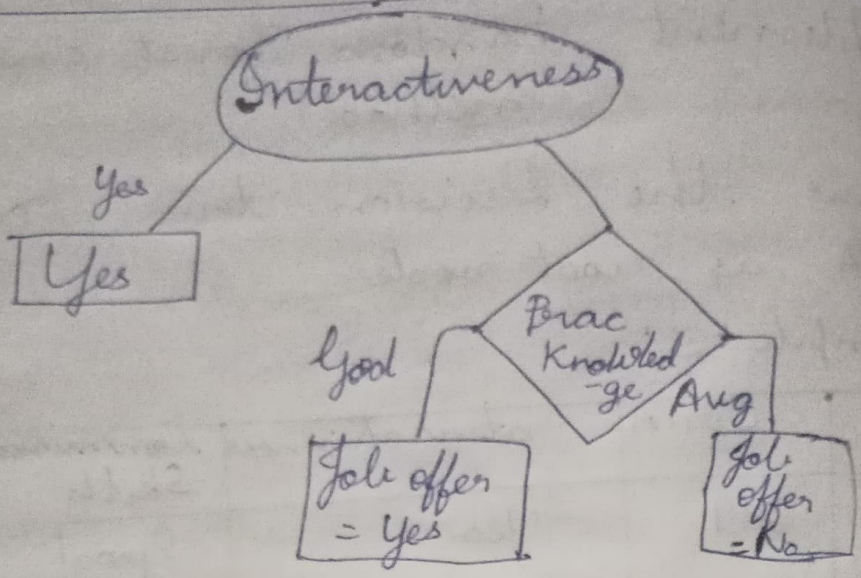
SNo.	CGPA	Interactiveness	Communication Skills	Prac Know	Job offe
1.	≥9	Yes	Good	Good	Yes
2.	<9	No	Moderate	Good	Y
3.	≥9	No	Moderate	Avg	No
4.	≥9	No	Moderate	Avg	No
5.	≥9	Yes	Moderate	Good	Yes

~~Class Entropy for the target class 'Job Offer'~~



Sample S2

S.No.	CGPA	Interactiveness	Comm Skills	Prac Know	Job Offer
2.	<9	No	Moderate	Good	Yes
3.	≥9	No	Moderate	Avg	No
3.	≥9	No	Moderate	Avg	No
5.	≥9	Yes	"	Good	Yes
5.	≥9	Yes	"	Good	Yes



Write Python Code to implement the following using "iris.csv" dataset
Build a RF classifier to classify IRIS flower dataset.

```
import pandas as pd
from sklearn.ensemble import
                    RandomForestClassifier
from sklearn.model_selection
                    import train_test_split
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test =
  train_test_split(X, y, test_size=0.3,
                                random_state=4
rf = RandomForestClassifier(n_estimator=
rf.fit(X_train, y_train)
```

print ("Accuracy with 10 trees :")

O/P:-

Accuracy with 10 trees : 1

Best accuracy : 1

Best accuracy Score : 1

Confusion Matrix

TP	FP
FN	TN

Lab-8

Implement Boosting ensemble method

Using AdaBoost Algorithm show the Decision stump calculation steps for the attribute CGPA.

CGPA	Interactiveness	Prac Know	Comm Skill	Job Prof
>=9	Yes	Good	Good	Yes
<9	No	Good	Moderate	Yes
>=9	No	Avg	"	No
<9	"	"	Good	"
>=9	Yes	Good	Moderate	Yes
>=9	"	"	"	"

α_c

Initial weight $= 1/6$

$$\varepsilon_i = \sum_{\forall j=1} H_i(d_j)\, Wt(d_j)$$

$$\varepsilon_{CGPA} = 2 \times \frac{1}{6} = 0.33$$

$$\alpha_{CGPA} = \frac{1}{2} \frac{\ln(1 - \varepsilon_{CGPA})}{\varepsilon_{CGPA}}$$

$$= 0.347$$

$$Z_{LGPA} = \frac{1}{6} \times 4 \times e^{-0.347} + \frac{1}{6} \times 2 \times e^{+0.347}$$

$$= 0.9428$$

$$Wt(dj)_{i+1} = \frac{1/6 \times e^{-0.347}}{0.9428} = 0.1249$$

for incorrect instance

$$= \frac{1/6 \times e^{+0.347}}{0.9428} = 0.2601$$

CGPA.	Initial Weight	Updated Weight
>=9	1/6	0.1249
<9	1/6	0.2501
>=9	1/6	0.1249
<9	1/6	0.2501
>=9	1/6	0.1249
>=9	1/6	0.1249

Best accuracy score = 1

Confusion matrix

TN	FP
FN	TP

Lab -9

K Means Algorithm

Compute two clusters using K-means algorithm for clustering where initial cluster centers are (1.0, 1.0) and (5.0, 7.0). Execute for 2 iterations

Iteration 1

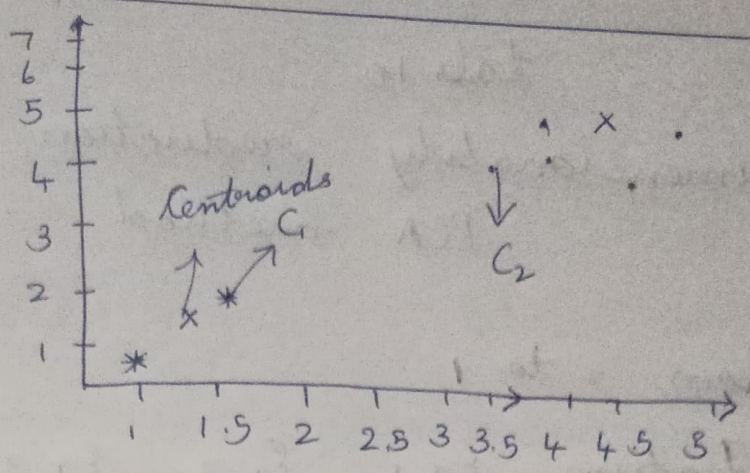
Record No.	A	B	Cluster (1.0, 1.0)	Cluster (5.0, 7.0)
R1	1.0	1.0	C_1	
R2	1.5	2.0	C_1	
R3	3.0	4.0	C_1	
R4	5.0	7.0		C_2
R5	3.5	5.0		C_2
R6	4.5	5.0		C_2
R7	3.5	4.5		C_2

New centroids (1.83, 2.33)

(4.125, 5.375)

Iteration 2

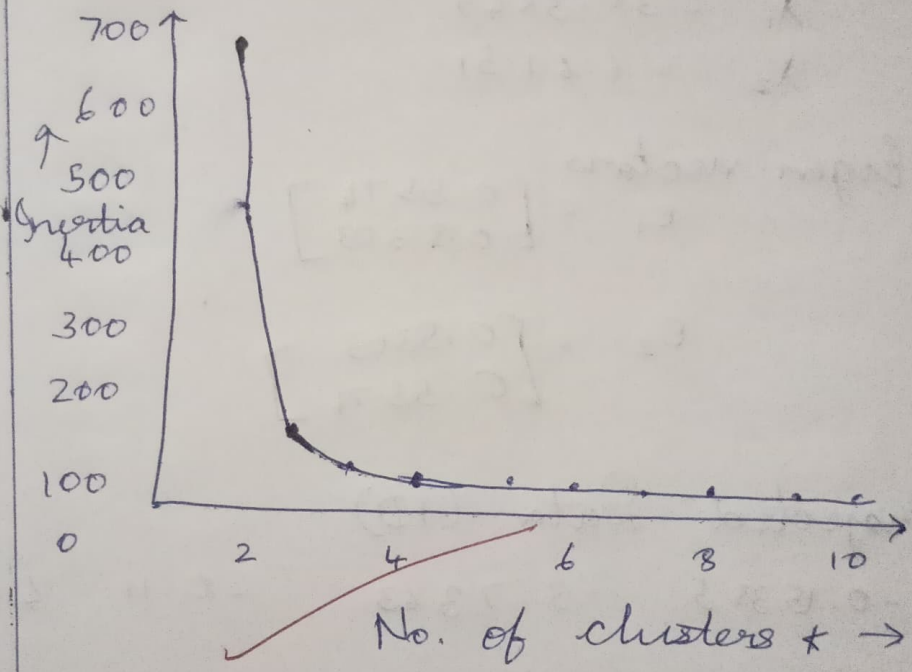
Record No	A	B	
R_1	1.0	1.0	C_1
R_2	1.5	2.0	C_1
R_3	3.0	4.0	C_2
R_4	5.0	7.0	C_2
R_5	3.5	5.0	C_2
R_6	4.5	5.0	C_2
R_7	3.5	4.5	C_2



New centroids

[1.25, 1.5] [3.9, 5.1]

Elbow Method for Optimal k



Lab 10

Dimensionality reduction using PCA method

From 2 to 1

Feature	Ex 1	Ex 2	Ex 3	Ex 4
X_1	4	8	13	7
X_2	11	4	5	14

Eigen Values

$\lambda_1 = 30.3849$

$\lambda_2 = 6.6151$

Eigen vectors

$e_1 = \begin{bmatrix} 0.5574 \\ 0.8303 \end{bmatrix}$

$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$

Projected Data (1D)

[-0.15385 -3.7363 -0.11 4.0]

heart.csv

Accuracy before PCA : 0.85

Accuracy after PCA : 0.83

7/5/2025