5/8/25

Lab 0

→ Initialize Values directly into DF

```
import pandas as pd
df = pd.DataFrame({
        "USN" : [23, 21, 24, 25],
        "Name" : ["Sam", "Ram", "Dam",
                  "rahul", "rakshith"],
        "Marks" : [45, 78, 24, 90, 60]
    })
df
```

→ import datasets from sklearn.datasets

```
from sklearn.datasets import
        load_diabetes
dia = load_diabetes()
df = pd.DataFrame(dia.data, columns=
        dia.feature_names)
df
```

→ Method 3: Import datasets from a specific .csv file

```
filepath = "/content/Sample_sales_data.csv"
pd.read_csv(filepath)
df.head()
```

Date / /
Page

→ Method 4: Downloading datasets from existing dataset repositories like Kaggle.

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

→ Using the code given the above slides, do the exercise of the "stock market data Analysis", considering the following.

1. HDFC Bank Ltd, ICICI Bank Ltd, Kotak Mahindra Band Ltd
   tickers = ["HDFCBank.NS", "ICICIBank.NS", "KotakBank.NS"]
2. Start date :- 2024-01-01, End_Date: 2024-12-30
3. Plot closing price and daily return for all three banks mentioned.

5/5/25

Lab-1

Date / /
Page

i) To load .csv into the dataframe
ii) To display info of all columns
iii) To displays statistical info of all numerical
iv) To displays the count of unique labels for "Ocean proximity" column
v) To display which att in dataset have missing values count greater than 2cr

```
import pandas as pd
df = pd.read_csv("housing.csv")
print(df.info)
print(df.describe())
df["Ocean proximity"].count
missing_values = df.isnull.sum()
print(missing_values[missingvalues>0])
```

Date
Page

```
import pandas as pd
import numpy as np

diabetes_df = pd.read_csv('Diabetes.csv')
adult_df = pd.read_csv("path/adult.csv")

def preprocess_data(df):
    missing_values = df.isnull().sum()
    print(missing_values[missing_values])
    numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
    categorical_cols = df.select_dtypes(include=['object']).columns

    imputer = SimpleImputer(strategy='most_frequent')
    df[categorical_cols] = imputer.fit_transform(df[categorical_cols])

    encoder = OneHotEncoder(drop='first', sparse_output=False)
    encoded_categorical = encoder.fit_transform(df[categorical_cols])
    encoded_categorical_df = pd.DataFrame(encoded_categorical, columns=
        encoder.get_feature_names_out(categorical_cols))
```

12/03/2025

Lab-2

Date
Page

~~Import~~

Use an appropriate dataset for building the decision tree (ID3) and apply this knowledge to classify a new sample. Eg:

```
import pandas as pd
from sklearn.model_selection import
        train_test_split.
from sklearn.preprocessing import
        LabelEncoder
from sklearn.tree import DecisionTreeClas
from sklearn import tree
import matplotlib.pyplot as plt

data = pd.read_csv('/seattle-weather.csv')

data_cleaned = data.drop('date', axis=
label_encoder = LabelEncoder()
data_cleaned['weather'] = label_encoder.
    fit_transform(data_cleaned['weathe

X = data_cleaned.drop('weather', axis=1
y = data_cleaned['weather']

X_train, X_test, y_train, y_test =
        train_test_split(X, y, test_size=0.2,
                random_state = 42)
id3_classifier = DecisionTreeClassifier(
    criterion='entropy', max_depth=5,
        random_state=42)
```

Date __/__/__
Page _____

```
id3_classifier.fit(X_train, y_train)

plt.figure(figsize=(15,8))
tree.plot_tree(id3_classifier, feature_names=
    X.columns, class_names=list(label_encoder.
    classes_), filled=True)
plt.title("ID3 Decision Tree Classifier
    (Improved)")
plt.show()
```

o/p

```
"Outlook":
    "Sunny": {
        "Humidity": {
            "High": "No"
            "Normal": "Yes"
    }}
Overcast: 'Yes'
"Rain": {
    "Wind": {
        "Weak": "Yes"
        "Strong": "No"
    }}}
```

Aishwarya

Lab-3
Linear Regression

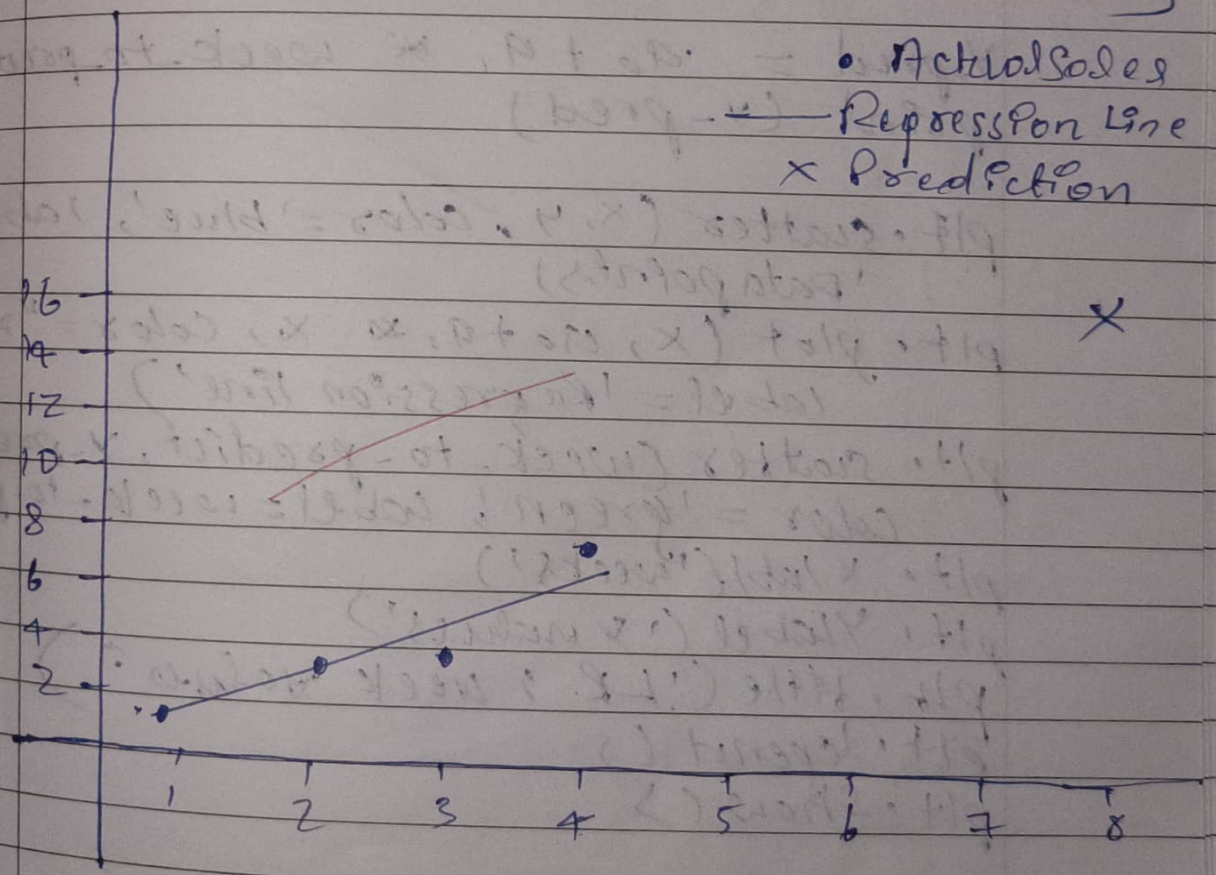
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

filepath = "/content/Book1.csv"
df = pd.read_csv(filepath)

X = df[['weeks']].values
Y = df[['sales in months']].values

X_b = np.c_[np.ones((X.shape[0],1)),X]

theta_best = np.linalg.inv(X_b.T @ X_b) @
             X_b.T @ Y

week_to_predict = int(input("Enter the
  week number to predict Sales:"))

future_week = np.array([[1, week_to_
              predict]])
predicted_sales = future_week @ theta_b

print("Theta (Intercept and Slope):",
      theta_best.flatten())
print(f"Predicted Sales for week
  {week_to_predict}: {predicted_sales.flat
  [0]}")

plt.scatter(X, Y, color='blue', label=
  'Actual Sales')
plt.plot(X, X_b @ theta_best, color='re
  label='Regression Line')
```

```
plt.scatter([week_to_predict], predicted_sales,
    color='green', marker='x', label=
        'prediction')
plt.xlabel('Weeks')
plt.ylabel('Sales In Months')
plt.legend()
plt.title('Linear Regression using Matrix
    Method')
plt.show()
```

o/p

Enter the week number to predict Sales : 8

Theta (Intercept and slope) : [-1.5, 2.2]



2nd Approach

```
import numpy as np
import matplotlib.pyplot as plt
x = np.array([1, 2, 3, 4])
y = np.array([1, 3, 4, 8])
x_mean = np.mean(x)
y_mean = np.mean(y)

a1 = (np.mean(x*y) - x_mean * y_mean)
     (np.mean(x**2) - x_mean**2)

a0 = y_mean - a1 * x_mean

week_to_predict = 5

y_pred = a0 + a1 * week_to_predict
print(y_pred)

plt.scatter(x, y, color='blue', label=
    'Data points')
plt.plot(x, a0 + a1 * x, color="red"
    label='Regression line')
plt.scatter(week_to_predict, y_pred,
    color='green', label=week_to_pred
plt.xlabel('weeks')
plt.ylabel('y values')
plt.title('LR : week values')
plt.legend()
plt.show()
```

o/p

Intercept (a0) : -1.5
Slope (a1) : 2.2
predicted value for week 5 : 9.5

2/4/25

Lab-4

Date / /
Page

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 logistic regression model has been trained, and the learned parameters are $\theta_0 = -5$ (intercept) and $\theta_1 = 0.8$ (coefficient for study hours).

a) Write a logistic regression equation for this problem.

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

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

Answer

a) $\frac{1}{1+e^{-z}}$ $\quad z = \theta_0 + \theta_1 x$

$\frac{1}{1+e^{-5+0.8x}}$

b) $\frac{1}{1+e^{-5+0.8(7)}} = 0.64$

c) Threshold of 0.5

0.64 is less than 0.5 so the student passed.

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

$$\frac{e^{zk}}{\sum_{j=1}^{k} e^{zk}}$$

$e^2 = 7.389$
$e^1 = 2.718$
$e^0 = 1$

Sum of exponentials

$S = e^2 + e^1 + e^0$
$= 7.389 + 2.718 + 1$
$= 11.107$

Softmax Probabilities

$$\frac{e^2}{S} = \frac{7.389}{11.107} = 0.665$$

$$\frac{e^1}{S} = \frac{2.718}{11.107} = 0.245$$

$$\frac{e^0}{S} = \frac{1}{11.107} = 0.090$$

Probabilities $= [0.665, 0.245, 0.090]$

Date __/__/__
Page

1. For dataset file "HR_comma_sep.csv"

i) Which variables did you identify as having a direct and clear impact on employee retention? why?

Satisfaction level - Employees with lower satisfaction levels are more likely to leave

Salary - Low salaries could be a major reason for employees quitting.

Avg Monthly hours - Excessive working hours could lead to burnout.

2. For 700 dataset

i) Did you perform any data preprocessing steps? If yes, what were they, and why were they necessary?

Yes,
Data cleaning : Checked for missing values and inconsistencies.
Encoding categorical data : the column animal-name is categorical and was removed as it does not contribute classification
class Balancing check : Ensured that dataset was balanced across class type

Date __/__/__
Page _____

ii) Were there any missing or inconsistent values in the dataset? How did you handle them?

- No missing values
- Replaced duplicate entities

iii) Interpretation of the confusion matrix

- Shows classification accuracy of predictions
- High diagonal indicates high performance & off diagonal miss classifications

02/4/25 Lab-5

Date / /
Page

Consider the following dataset, for k=3 and test data (x, 35, 100) as (Person, Age, Salary) solve using knn classifier model and predict the target.

Person	Age	Salary	Target	Dis
A	18	50	N	52.
B	23	55	N	46.5
C	24	70	N	31.9
D	41	60	Y	40.4
E	43	70	Y	31.0
F	38	40	Y	60.0
x	35	100	?=Y	

k=1 → N
k=2 → N
k=3 → Y
} Majority Y

For Iris dataset

8 How to choose the K value? Demonstrate using accuracy rate and error rate.

→ Train KNN for different K values (K=1 to 10)

Compute Accuracy Rate

$$\frac{\text{Correct Prediction}}{\text{Total Prediction}} \times 100$$

Compute error rate:

Error Rate = 1 - Accuracy rate

For Diabetes dataset

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

Min - Max Scaling

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Standardization

$$x' = \frac{x - \mu}{\sigma}$$

4-4-2025

15/04/25

Date
Page

Lab-6

Random forest Algorithm

Step1: Select random k data points from the training set

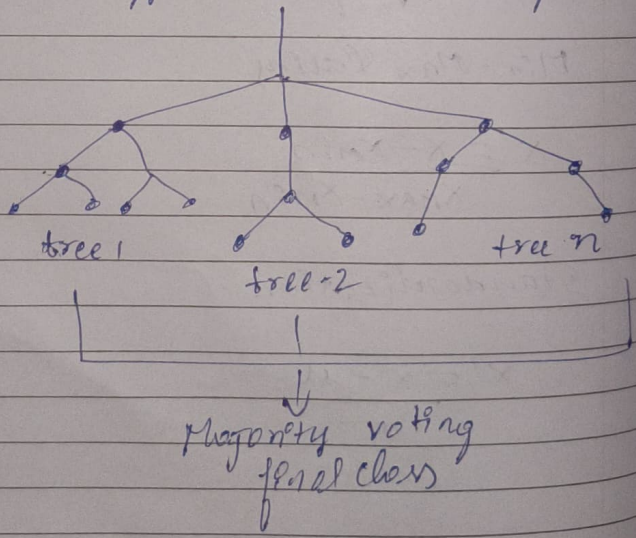
Step2: Build the decision tree associated with selected data points

Step3: Choose the number N for decision trees that you want to build

Step4: Repeat Step 1 & 2

Step5: For new data points, find the prediction of each decision tree and assign the new data points to the category that wins the majority votes

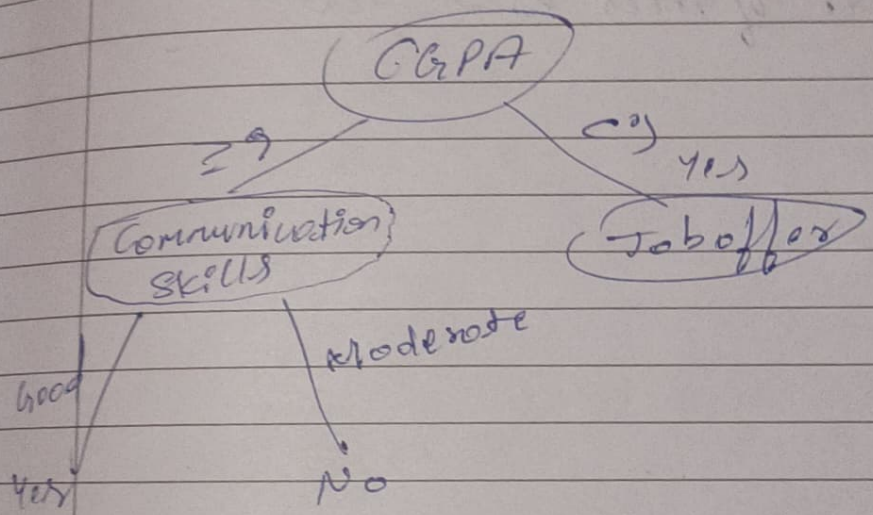
classes: Apple, banana, Grapes

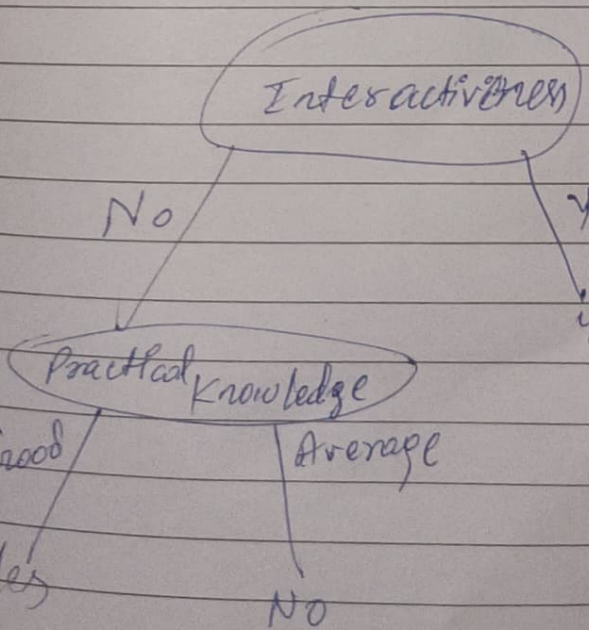


Random Forest

1) Sample SI

SI.No	CGPA	Interactiveness	Communication Prac	Job offer
1	≥9	Yes	G	G	Y
2	<9	No	M	G	Y
3	≥9	No	M	A	N
4	≥9	No	M	A	N
5	≥9	Yes	M	G	Y





Date / /
Page

1. For Iris.csv dataset

2. Best accuracy score: 100%.

3. Confusion matrix:

	Predicted Setosa	Versicolor	Verginica
Setosa	50	0	0
Versicolor	0	50	0
Verginical	0	0	50

4. Number of trees used 100

Date / /
Page

Lab - 8

Date / /
Page

Verginica
0
0
5.6

AdaBoost

CGPA	Interactiveness	Practical Knowledge	Communication	Job Prof.
>=9	Y	Good	Good	Y
<9	N	Good	Moderate	Y
>=9	N	Avg	Moderate	N
<9	N	Avg	Good	N
>=9	Y	Good	Moderate	Y
>=9	Y	Good	Moderate	Y

Step ①

CGPA	Predicted	Actual	Weight
>=9	Y	Y	1/6
<9	N	Y	1/6
>=9	Y	N	1/6
<9	N	N	1/6
>=9	Y	Y	1/6
>=9	Y	Y	1/6

Step ②

No of wrong prediction / Total no of (i)

$$\text{error} = \frac{2 \times 1}{6} = 1/3$$

Step 3

$$\alpha = \frac{1}{2} \log\left(\frac{1 - \text{error}}{\text{error}}\right)$$

$$\alpha = \frac{1}{2} \log\left(\frac{1 - 1/3}{1/3}\right) = 0.347$$

Step4:

$z = wt(correct) * No.\ of\ correct * e^{-a} + wt(incorrect) * No.\ of\ Incorrect * e^{a}$

$= 1/6 * 4 * e^{-0.347} + 1/6 * 2 * e^{0.347}$

$= 0.9428$

Step5:

Update the weight

$$\frac{w(dj)_{CGPA\ of\ correct} * e^{-a}}{z}$$

$$\frac{1/6 * e^{-0.347}}{0.9428} = 0.1249$$

$$\frac{w(dj)\ CGPA\ of\ Incorrect * e^{a}}{0.9428}$$

$$\frac{1/6 * e^{0.347}}{0.9428} = 0.2501$$

CGPA	Predicted	Actual	Weight
>=9	Y	Y	0.1249
<9	N	Y	0.2501
>=9	Y	N	0.2501
<9	N	N	0.7248
>=9	Y	Y	0.1249
>=9	Y	Y	0.1249

LAB 9

Date __/__/__
Page _____

K-means, 2nd iteration

Record Number	A	B
R1	1.0	1.0
R2	1.5	2.0
R3	3.0	4.0
R4	5.0	7.0
R5	3.5	5.0
R6	4.5	5.0
R7	3.5	4.5

Formula $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Record	Point (A,B)	d(c1)	d(c2)	Clusters
R1	(1.0, 1.0)	0.00	7.2	C1
R2	(1.5, 2.0)	1.12	6.20	C1
R3	(3.0, 4.0)	3.61	3.61	C1/C2 (C1)
R4	(5.0, 7.0)	7.21	0.00	C2
R5	(3.5, 5.0)	5.00	2.50	C2
R6	(4.5, 5.0)	5.32	2.24	C2
R7	(3.5, 4.5)	4.30	3.20	C2

C1 clusters :- R1, R2, R3

C2 clusters :- R4, R5, R6, R7

C1 mean

$\frac{1 + 1.5 + 3}{3} = 1.83$ $\frac{1 + 2 + 4}{3} = 2.3$

C2 Mean

$\frac{5 + 3.5 + 4.5 + 3.5}{4} = 4.125$, $\frac{7 + 5 + 5 + 4.5}{4}$

$= 5.375$

$C_1 = (1.83, 2.33)$

$C_2 = (4.125, 5.375)$

Record	Point (A,B)	d_1	d_{C_2}	Cluster
R1	(1, 1)	1.57	5.47	C1
R2	(1.5, 2)	0.47	4.32	C1
R3	(3, 4)	1.98	1.53	C2
R4	(5, 7)	4.91	1.76	C2
R5	(3.5, 5)	2.68	0.73	C2
R6	(4.5, 5.0)	3.14	0.52	C2
R7	(3.5, 4.5)	2.28	1.00	C2

Lab-10

PCA

Feature	Ex1	Ex2	Ex3	Ex4
X_1	4	8	13	7
X_2	11	4	5	14

$d1 = 30.3849$
$d2 = 6.6151$

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

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

(a) data Matrix $= \begin{bmatrix} 4 & 8 & 13 & 7 \\ 11 & 4 & 5 & 14 \end{bmatrix}$

(b) Mean centre the data

$$\text{Mean } X_1 = \frac{4+8+13+7}{4} = 8$$

$$\text{Mean } X_2 = \frac{11+4+5+14}{4} = 8.5$$

$$X \text{ centered} = \begin{bmatrix} 4-8 & 8-8 & 13-8 & 7-8 \\ 11-8.5 & 4-8.5 & 5-8.5 & 14-8.5 \end{bmatrix}$$

$$= \begin{bmatrix} -4 & 0 & 5 & -1 \\ 2.5 & -4.5 & -3.5 & 5.5 \end{bmatrix}$$

(c) $Z = X^T e^{-T} \times$ centered

$X_1 = (0.5574)(-4) + (-0.8303)(2.5) = 6$
$X_2 = (0.5574)(0) + (-0.8303)(-4.5) = 3.73635$
$X_3 = (0.5574)(5) + (-0.8303)(-3.5) = 5.69505$
$X_4 = (0.5574)(-1) + (-0.8303)(5.5) = -5.12405$