## PRACTICAL JOURNAL

in

# DATA SCIENCE

Submitted by

## GAURAV VISHWAS TAMBE KFMSCIT038

for the award of the Degree of

**MASTERS OF SCIENCE (INFORMATION TECHNOLOGY) PART – I**

**DEPARTMENT OF INFORMATION TECHNOLOGY KISHINCHAND CHELLARAM COLLEGE**

**(Affiliated to University of HSNCU) MUMBAI,400020 MAHARASHTRA**

**2023-24**

SUBJECT CODE - MS-FIT-209

# DATA SCIENCE

**KISHINCHAND CHELLARAM COLLEGE**

CHURCHGATE, MUMBAI – 400 020.

**DEPARTMENT OF INFORMATION TECHNOLOGY M.SC.I.T PART- I**

**CERTIFICATE**

This is to certify that the Practical conducted by

Mr. **GAURAV VISHWAS TAMBE** for M.Sc. (IT) Part- I Semester- II, Seat No: **KFMSCIT038** at Kishinchand Chellaram College in partial fulfillment for the MASTERS OF SCIENCE (INFORMATION TECHNOLOGY). Degree Examination for Semester II has been periodically examined and signed, and the course of term work has been satisfactorily carried out for the year 2023 - 2024. This Practical journal had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

**Signature Signature Signature Lecturer-In-Charge External Examiner Course Coordination**

**Guided By Examined By Certified By**

**College Stamp**

# INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **TITLE** | **Date** | **Page No.** | **Signature** |
|  | **PRACTICALS** |  |  |  |
| **1** | 1. Design a simple machine learning model (Linear Regression) g instance and test the same using Python. 2. For a given set of training data examples stored in a .CSV file implement Logistic Regression algorithm. | 09/03/2024 | 1-4 |  |
| **2** | 1. Introduction to R Language. 2. Write a program to implement k-Nearest Neighbor algorithm to classify the following data set. | 16/03/2024 | 5-10 |  |
| **3** | Naïve Bayes Algorithm | 16/03/2024 | 11-13 |  |
| **4** | K-Means Clustering | 17/03/2024 | 14-15 |  |
| **5** | Data Visualization Using Python | 17/03/2024 | 16-18 |  |
| **6** | Pie Diagram, Bar Chart Diagram using Python | 23/03/2024 | 19-20 |  |
| **7** | ETL process in Python for Web Scrapping | 23/03/2024 | 21 |  |
| **8** | Data visualization using python libraries | 24/03/2024 | 22-24 |  |
| **9** | Decision Tree Classifier | 24/03/2024 | 25-26 |  |

### PRACTICAL 1-A

**Aim:** Design a simple machine learning model (Linear Regression) g instance and test the same using Python.

**Code:**

import random

from sklearn.linear\_model import LinearRegression print("Gaurav Tambe 038")

rows = 200

limits = 2000 featureSet = [] targetSet = []

for i in range(0,rows):

x = random.randint(0,limits)

y = random.randint(0,limits)

z = random.randint(0,limits) g = ((10\*x)+(2\*y)+(3\*z))

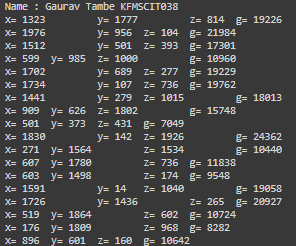
print(i+1,". ",x," ",y," ",z," = ",g) print("==========================================")

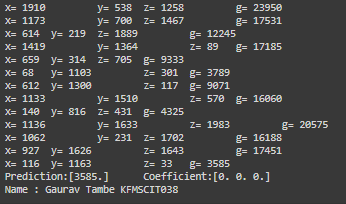
featureSet.append([x,y,z]) targetSet.append(g)

model = LinearRegression( model.fit(featureSet,targetSet) test\_data = [[1,2,3]]

prediction = model.predict(test\_data)

print("Prediction: ",prediction," Cofficient: ",model.coef\_) print("Gaurav Tambe 038")





### PRACTICAL 1-B

**Aim:** For a given set of training data examples stored in a .CSV file implement Logistic Regression algorithm.

**Code:**

print("Gaurav Tambe KFMSCIT038") import pandas as pd

from sklearn.model\_selection import train\_test\_split from sklearn.preprocessing import StandardScaler from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix,accuracy\_score

dataset = pd.read\_csv(r"C:\Users\hp\Desktop\KC MScIT Part 1\Sem 1\AAI\Practical\diabetes.csv") print(dataset.head())

x = dataset.iloc[:,[0,1,2,3,4,5,6,7]].values

y = dataset.iloc[:,[-1]].values.ravel()

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=100) sc = StandardScaler()

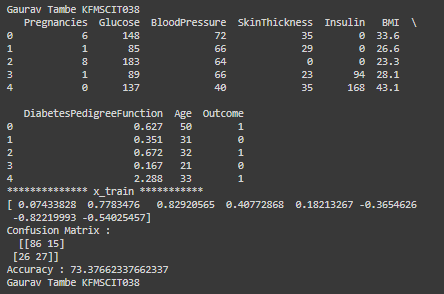
x\_train = sc.fit\_transform(x\_train) x\_test = sc.transform(x\_test)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\* x\_train \*\*\*\*\*\*\*\*\*\*\*") print(x\_train[1])

model = LogisticRegression() model.fit(x\_train, y\_train) y\_pred = model.predict(x\_test)

cm = confusion\_matrix(y\_test, y\_pred) print("Confusion Matrix :\n ", cm)

print("Accuracy :", accuracy\_score(y\_test, y\_pred) \* 100) print("Gaurav Tambe 038")



### PRACTICAL 2-A

**Aim**: Introduction to R programming

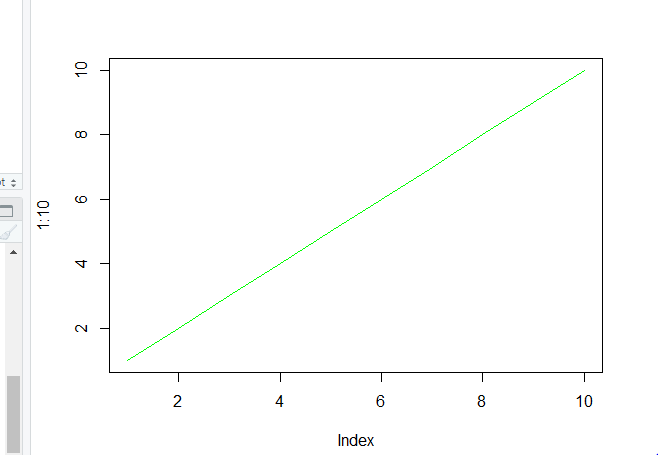
**Code & Output:**

name <- "Gaurav Tambe" rollNo <- "roll 038"

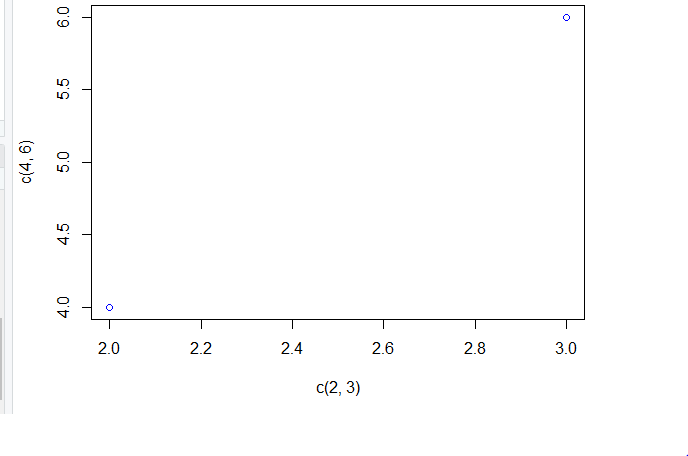
cat("Name: ",name,"\nRoll No: ",rollNo)



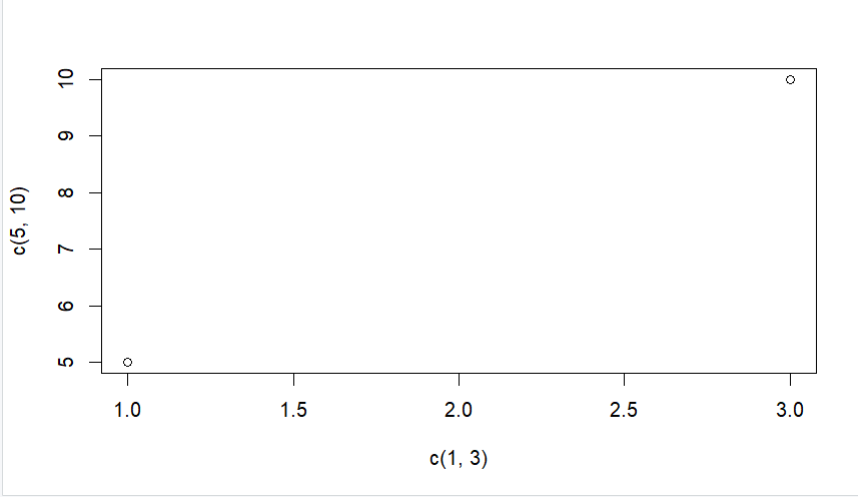
plot(1:10,type="l",col="green")



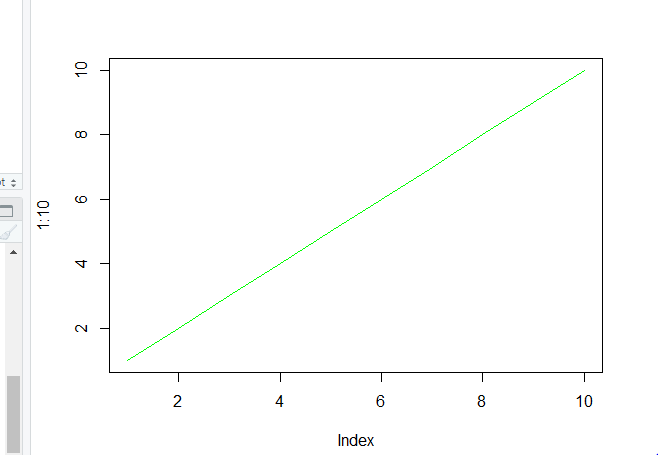
plot(1,4, col="red") plot(c(2,3),c(4,6) ,col="blue")



plot(c(1,3),c(5,10))

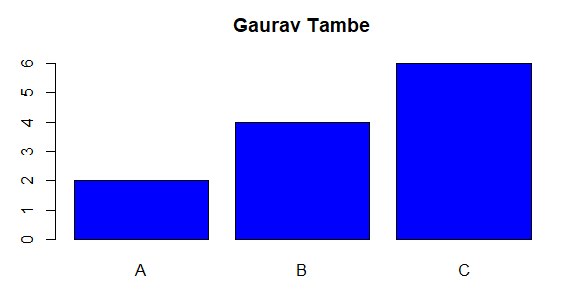


plot(1:10,type="l",col="green")



#Bar Charts

x<-c("A","B","C") y<-c(2,4,6)

barplot(y,main="Gaurav Tambe 038 ",names.arg = x,col="blue")

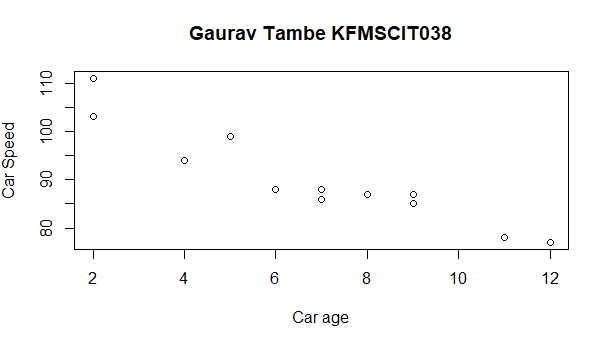
#Scatter plots

x<-c(5,7,8,7,2,2,9,4,11,12,9,6)

y<-c(99,86,87,88,111,103,87,94,78,77,85,88)

plot(x,y)

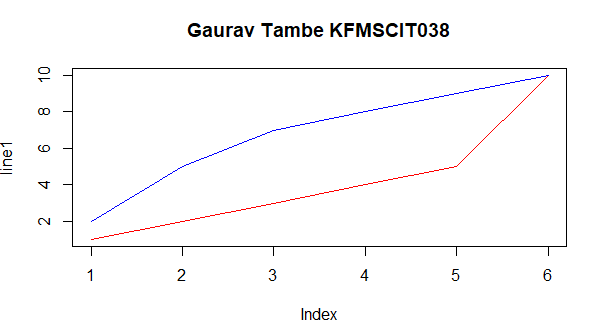
plot(x,y,main="Gaurav Tambe 038 ",xlab="Car age",ylab="Car Speed")



line1<-c(1,2,3,4,5,10)

line2<-c(2,5,7,8,9,10)

plot(line1,main="Gaurav Tambe 038 ",type="l",col="red") lines(line2,type="l",col="blue")



### PRACTICAL 2-B

**Aim:** Write a program to implement k-Nearest Neighbor algorithm to classify the following data set.

**Code & Output:**

import matplotlib.pyplot as plt

from sklearn.neighbors import KNeighborsClassifier #Data Set

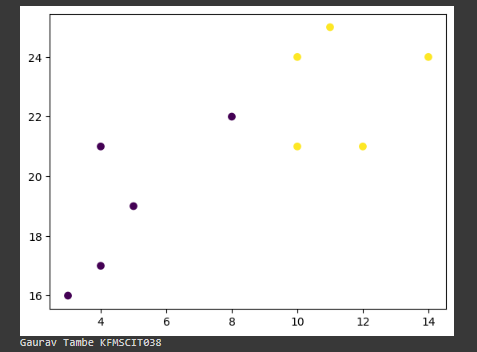
x = [4, 5, 10, 4, 3, 11, 14 , 8, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

classes = [0, 0, 1, 0, 0, 1, 1, 0, 1, 1]

#To plot data set plt.scatter(x, y, c=classes) plt.show()

print("Gaurav Tambe 038")



#To merge x and y into single list data = list(zip(x, y))

print(data) #model creation

model = KNeighborsClassifier(n\_neighbors=1)

# Model is trained model.fit(data, classes)

print("Gaurav Tambe 038")

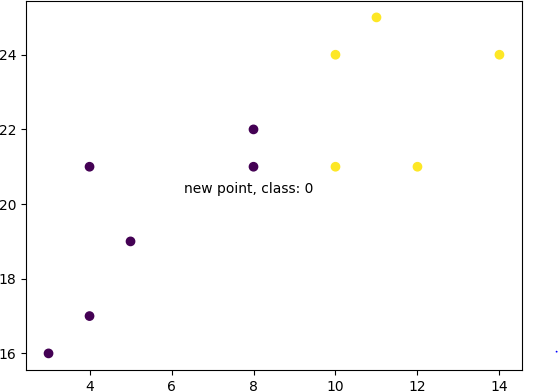
new\_x = 8

new\_y = 21

new\_point = [(new\_x, new\_y)] prediction = model.predict(new\_point)

plt.scatter(x + [new\_x], y + [new\_y], c=classes + [prediction[0]]) plt.text(x=new\_x-1.7, y=new\_y-0.7, s=f"new point, class: {prediction[0]}") plt.show()

print("Gaurav Tambe 038")

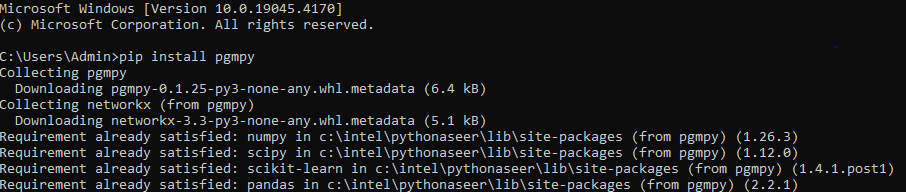


### PRACTICAL 3

**Aim:** Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set.

**Code & Output:**

!pip install pgmpy



import pandas as pd

from pgmpy.estimators import MaximumLikelihoodEstimator from pgmpy.models import BayesianModel

from pgmpy.inference import VariableElimination

data = pd.read\_csv("ds4.csv") heart\_disease = pd.DataFrame(data) print(heart\_disease)

model = BayesianModel([ ('age', 'Lifestyle'),

('Gender', 'Lifestyle'),

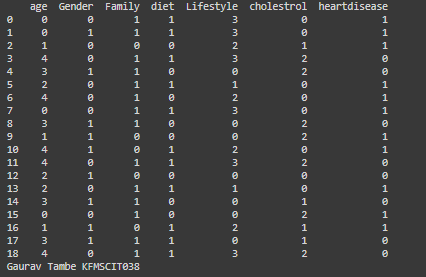
('Family', 'heartdisease'),

('diet', 'cholestrol'),

('Lifestyle', 'diet'), ('cholestrol', 'heartdisease'), ('diet', 'cholestrol')

])

print("Gaurav Tambe 038")



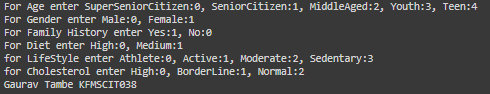
model.fit(heart\_disease, estimator=MaximumLikelihoodEstimator)

HeartDisease\_infer = VariableElimination(model)

print('For Age enter SuperSeniorCitizen:0, SeniorCitizen:1, MiddleAged:2, Youth:3, Teen:4') print('For Gender enter Male:0, Female:1')

print('For Family History enter Yes:1, No:0') print('For Diet enter High:0, Medium:1')

print('for LifeStyle enter Athlete:0, Active:1, Moderate:2, Sedentary:3') print('for Cholesterol enter High:0, BorderLine:1, Normal:2') print("Gaurav Tambe 038")



q = HeartDisease\_infer.query(variables=['heartdisease'], evidence={

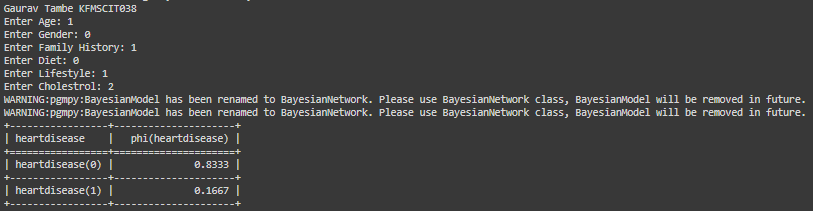
'age': int(input('Enter Age: ')),

'Gender': int(input('Enter Gender: ')), 'Family': int(input('Enter Family History: ')), 'diet': int(input('Enter Diet: ')),

'Lifestyle': int(input('Enter Lifestyle: ')), 'cholestrol': int(input('Enter Cholestrol: '))

})

print(q)



### PRACTICAL 4

**Aim:** K-Means Clustering

**Code:**

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]

y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

plt.scatter(x, y) plt.show()

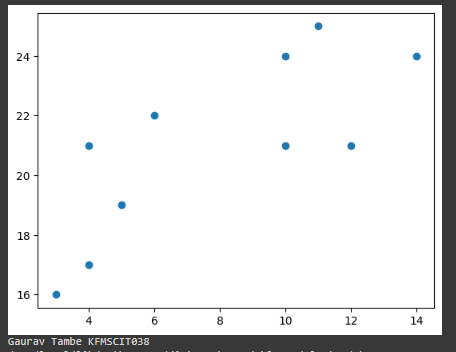
print("Gaurav Tambe 038") data = list(zip(x, y))

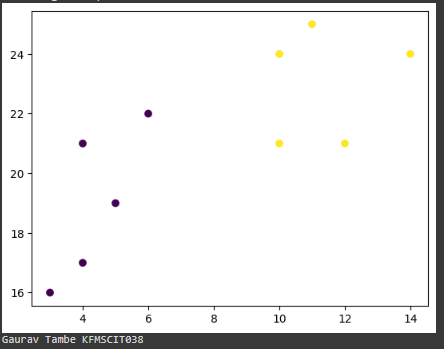
kmeans = KMeans(n\_clusters=2) kmeans.fit(data)

plt.scatter(x, y, c=kmeans.labels\_) plt.show()

print("Gaurav Tambe 038")

**Output :**





### PRACTICAL 5

**Aim**: Data Visualization Using Python.

**Code 1:**

import pandas as pd

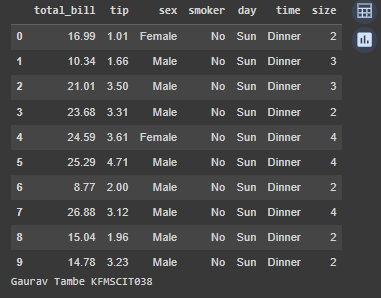
# reading the database

data = pd.read\_csv("tips.csv")

# printing the top 10 rows display(data.head(10))

print("Gaurav Tambe 038")

**Output 1:**



**Code 2:**

import pandas as pd

import matplotlib.pyplot as plt

# reading the database

data = pd.read\_csv("tips.csv")

# Scatter plot with day against tip plt.scatter(data['day'], data['tip'])

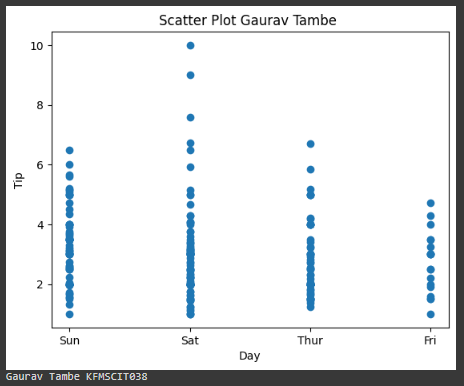
# Adding Title to the Plot plt.title("Scatter Plot Gaurav Tambe")

# Setting the X and Y labels plt.xlabel('Day') plt.ylabel('Tip')

plt.show()

print("Gaurav Tambe 038")

**Output 2:**



import pandas as pd

import matplotlib.pyplot as plt

# reading the database

data = pd.read\_csv("tips.csv")

# Scatter plot with day against tip

plt.scatter(data['day'], data['tip'], c=data['size'], s=data['total\_bill'])

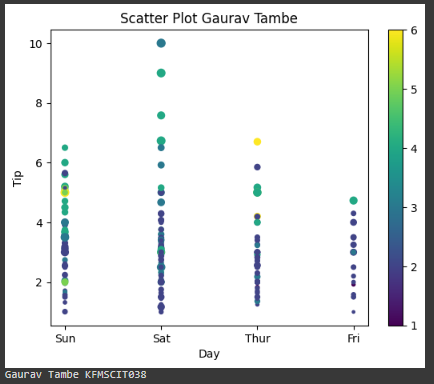
# Adding Title to the Plot plt.title("Scatter Plot Gaurav Tambe 038")

# Setting the X and Y labels plt.xlabel('Day') plt.ylabel('Tip') plt.colorbar()

plt.show()

print("Gaurav Tambe 038")

**Output 3:**



**Aim:** Pie Diagram, Bar Chart Diagram using Python.

**Code 1:**

import pandas as pd

import matplotlib.pyplot as plt

# reading the database

data = pd.read\_csv("tips.csv")

# Bar chart with day against tip plt.bar(data['day'], data['tip'])

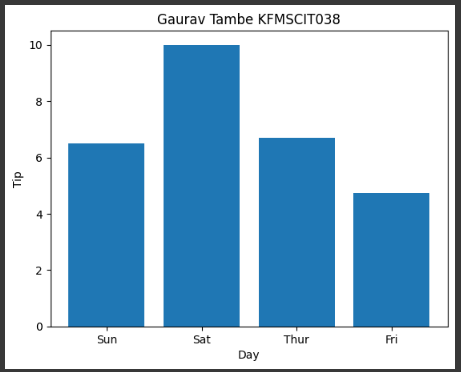
plt.title("Gaurav Tambe 038") # Setting the X and Y labels

plt.xlabel('Day')

plt.ylabel('Tip')

# Adding the legends plt.show()

**Output:**



**Code 2:**

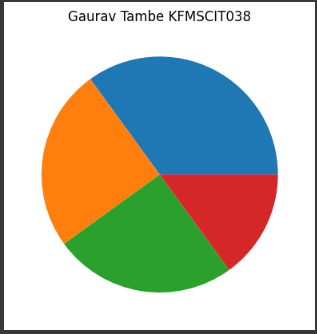
import numpy as np

import matplotlib.pyplot as plt y = np.array([35,25,25,15])

plt.title("Gaurav Tambe 038") plt.pie(y)

plt.show()

**Output 2:**



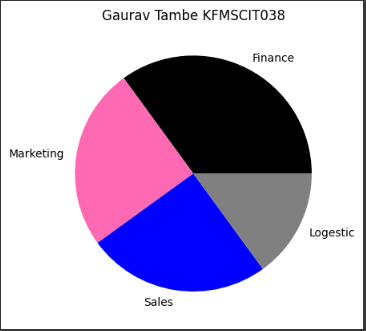
**Code 3:**

import numpy as np

import matplotlib.pyplot as plt y = np.array([35,25,25,15])

mylabels = ["Finance", "Marketing","Sales","Logestic"] mycolors = ["black","hotpink","b","gray"] plt.title("Gaurav Tambe 038") plt.pie(y, labels = mylabels, colors = mycolors) plt.show()

**Output 3:**



### PRACTICAL 7

**Aim:** ETL process in Python for Web Scrapping.

**Code:**

import requests

from bs4 import BeautifulSoup print("Gaurav Tambe 016") def check\_word\_in\_webpage(url, word):

response = requests.get(url)

if response.status\_code == 200:

soup = BeautifulSoup(response.content, 'html.parser') text\_content = soup.get\_text()

if word.lower() in text\_content.lower():

print(f"The word '{word}' is present in the webpage.") else:

print(f"The word '{word}' is not present in the webpage.")

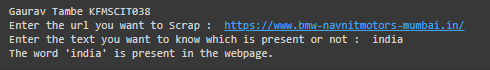
else:

print("Failed to retrieve webpage.")

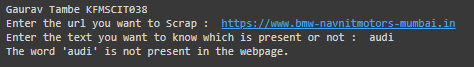
url = input("Enter the url you want to Scrap : ")

word\_to\_check = input("Enter the text you want to know which is present or not : ") check\_word\_in\_webpage(url, word\_to\_check)

**Output : Successful attempt**



**Failed attempt**



### PRACTICAL 8

**Aim:** Data visualization using python.

**Code 1:**

#STEP -1 : Import Libraries import pandas as pd

import matplotlib. pyplot as plt #Uploading Dataset

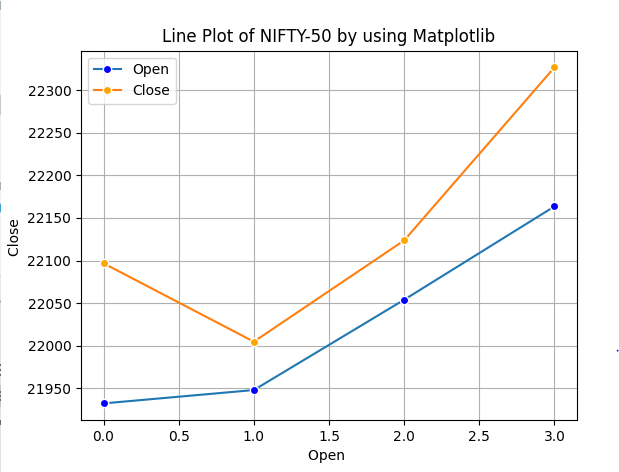
data = pd.read\_csv("Nifty.csv") print(data)

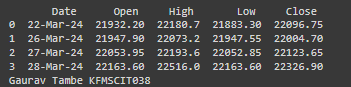
print(“Gaurav Tambe 038") #Plotting of x-axis & y-axis plot

plt. plot(data['Open '],marker = "o",markeredgecolor="white", markerfacecolor="blue") plt. plot(data['Close '],marker = "o",markeredgecolor="white", markerfacecolor="orange") plt.grid()

plt. title("Line Plot of NIFTY-50 by using Matplotlib") plt. xlabel('Open ')

plt. ylabel('Close ') plt.legend(['Open', 'Close']) plt. show()

**Output :**



**Code 2:**

#DATA VISUALIZATION USING SEABORN

import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt # Load dataset

data = pd.read\_csv("Nifty.csv")

print("Gaurav Tambe 038") # Plotting

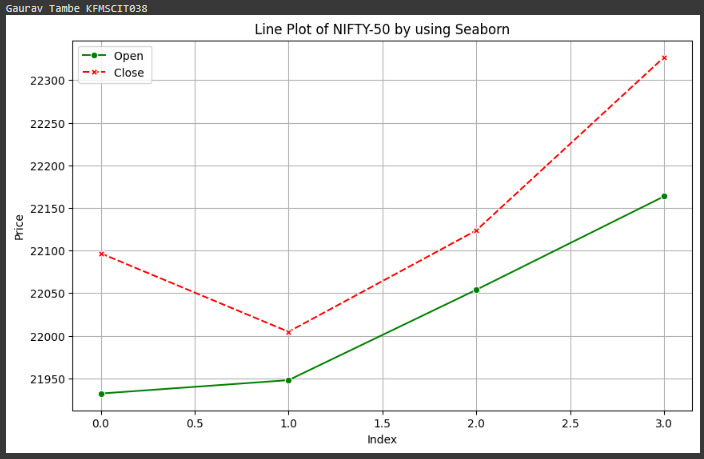
plt.figure(figsize=(10, 6))

sns.lineplot(data=data[['Open ', 'Close ']], markers=True, marker='o', markeredgecolor="white", palette=['green', 'red'])

plt.title("Line Plot of NIFTY-50 by using Seaborn") plt.xlabel('Index')

plt.ylabel('Price') plt.grid(True) plt.show()

**Output 2:**



**Code 3:**

#DATA VISUALIZATION USING PLOTLY

import pandas as pd import plotly.express as px # Load dataset

data = pd.read\_csv("Nifty.csv")

print("Gaurav Tambe 038") # Plotting

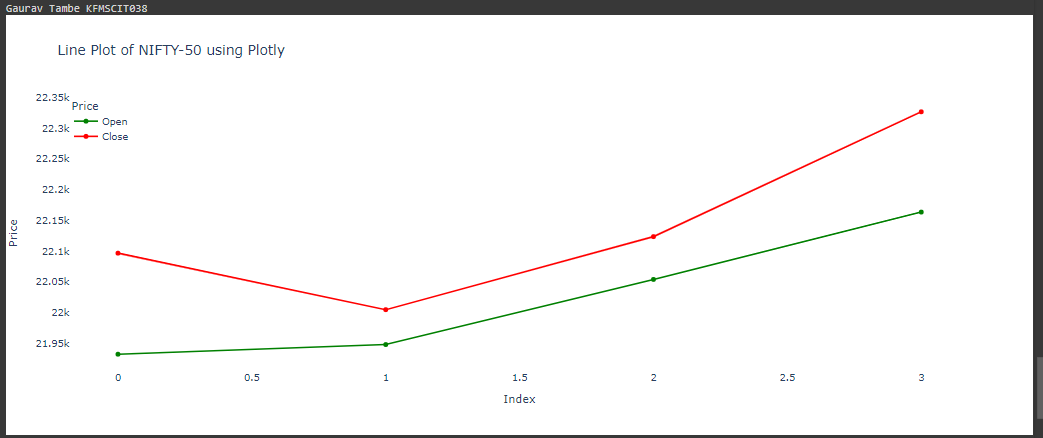
fig = px.line(data, x=data.index, y=['Open ', 'Close '], markers=True,

title="Line Plot of NIFTY-50 using Plotly", labels={'index': 'Index', 'value': 'Price'}, color\_discrete\_sequence=['green', 'red'])

fig.update\_layout(legend\_title='Price', legend=dict(x=0, y=1, traceorder='normal'), plot\_bgcolor='rgba(0,0,0,0)')

fig.show()

**Output 3:**



### PRACTICAL 9

**Aim:** Decision Tree Classifier.

**Code:**

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.tree import DecisionTreeClassifier, plot\_tree from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score print("Gaurav Tambe 038")

data = pd.read\_csv('iris.csv') print(data.head())

X = data.drop('variety', axis=1) y = data['variety']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42) model = DecisionTreeClassifier()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred) print(f"Accuracy: {accuracy:.2f}")

plt.figure(figsize=(12, 8))

plot\_tree(model, filled=True, feature\_names=X.columns.tolist(), class\_names=y.unique().tolist()) plt.title("Decision Tree Visualization")

plt.show()

**Output:**

