KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

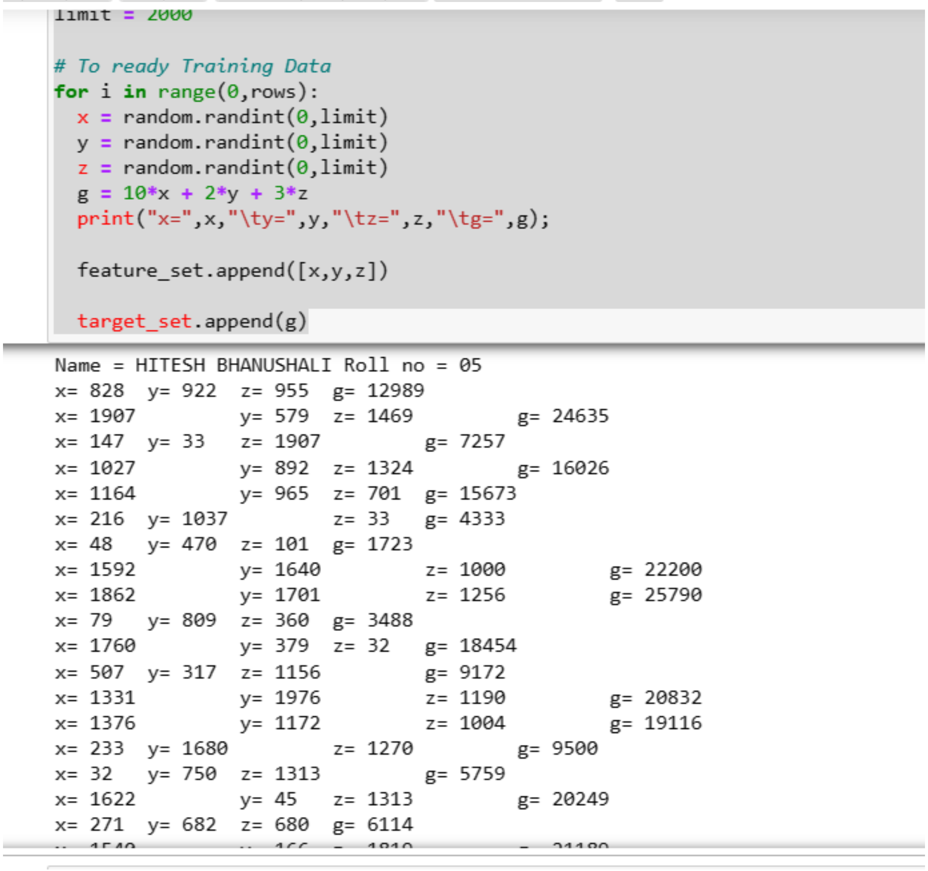
M.Sc (I.T.) Part-1 Semester I

PRACTICAL NO 1-A:

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

Code:

```
#Libraries Required
import random
from sklearn.linear_model import LinearRegression
print("Name = HITESH BHANUSHALI Roll no = 05")
#Declarartion Code
feature_set = []
target_set = []
rows = 200
limit = 2000
# To ready Training Data
for i in range(0,rows):
  x = random.randint(0,limit)
  y = random.randint(0,limit)
  z = random.randint(0,limit)
  g = 10*x + 2*y + 3*z
  print("x=",x,"\ty=",y,"\tz=",z,"\tg=",g);
  feature_set.append([x,y,z])
  target_set.append(g)
```

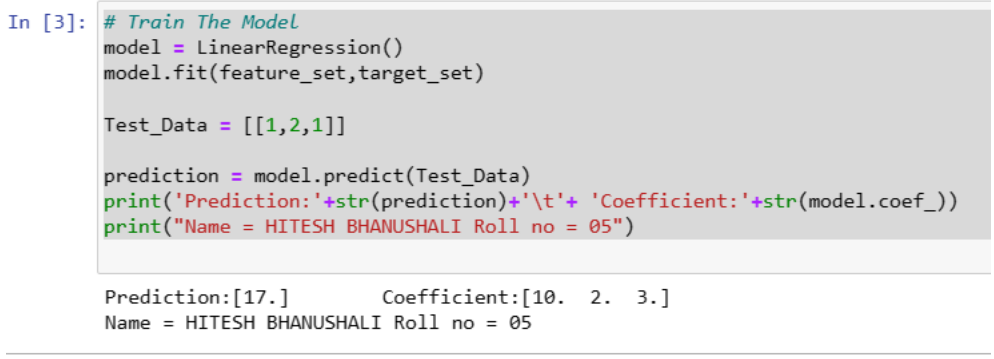


```
limit = 2000

# To ready Training Data
for i in range(0,rows):
  x = random.randint(0,limit)
  y = random.randint(0,limit)
  z = random.randint(0,limit)
  g = 10*x + 2*y + 3*z
  print("x=",x,"\ty=",y,"\tz=",z,"\tg=",g);

  feature_set.append([x,y,z])

  target_set.append(g)
```

```
Name = HITESH BHANUSHALI Roll no = 05
x= 828  y= 922  z= 955  g= 12989
x= 1907         y= 579  z= 1469         g= 24635
x= 147  y= 33   z= 1907         g= 7257
x= 1027         y= 892  z= 1324         g= 16026
x= 1164         y= 965  z= 701  g= 15673
x= 216  y= 1037         z= 33   g= 4333
x= 48   y= 470  z= 101  g= 1723
x= 1592         y= 1640         z= 1000         g= 22200
x= 1862         y= 1701         z= 1256         g= 25790
x= 79   y= 809  z= 360  g= 3488
x= 1760         y= 379  z= 32   g= 18454
x= 507  y= 317  z= 1156         g= 9172
x= 1331         y= 1976         z= 1190         g= 20832
x= 1376         y= 1172         z= 1004         g= 19116
x= 233  y= 1680         z= 1270         g= 9500
x= 32   y= 750  z= 1313         g= 5759
x= 1622         y= 45   z= 1313         g= 20249
x= 271  y= 682  z= 680  g= 6114
```

```
# Train The Model
```

```
model = LinearRegression()
model.fit(feature_set,target_set)

Test_Data = [[1,2,1]]

prediction = model.predict(Test_Data)
print('Prediction:'+str(prediction)+'\t'+ 'Coefficient:'+str(model.coef_))
print("Name = HITESH BHANUSHALI Roll no = 05")
```



```
In [3]: # Train The Model
        model = LinearRegression()
        model.fit(feature_set,target_set)

        Test_Data = [[1,2,1]]

        prediction = model.predict(Test_Data)
        print('Prediction:'+str(prediction)+'\t'+ 'Coefficient:'+str(model.coef_))
        print("Name = HITESH BHANUSHALI Roll no = 05")
```

```
Prediction:[17.]        Coefficient:[10.  2.  3.]
Name = HITESH BHANUSHALI Roll no = 05
```

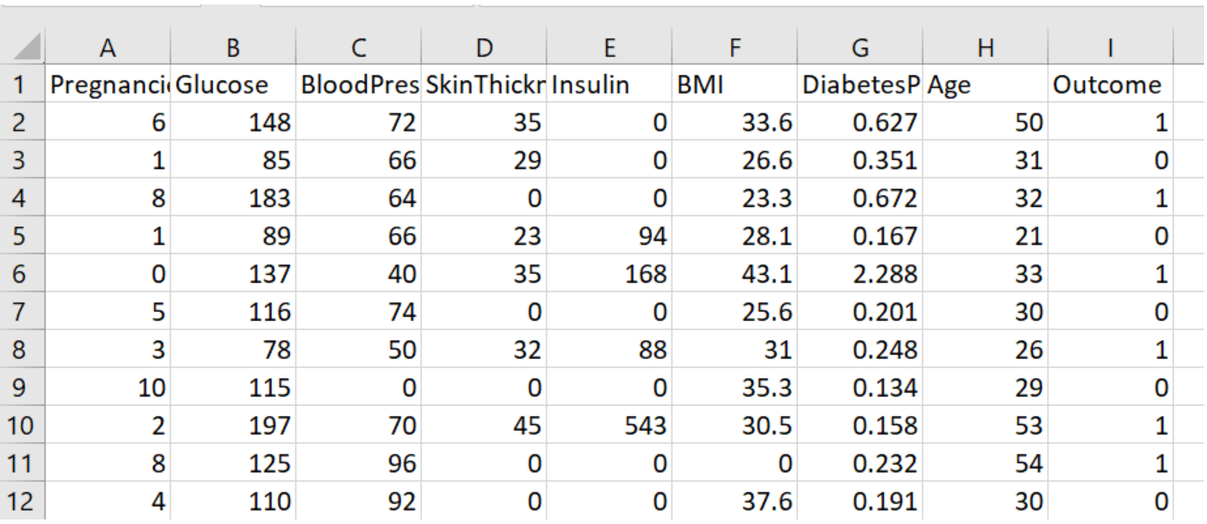
KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

M.Sc (I.T.) Part-1 Semester I

PRACTICAL NO 1-B

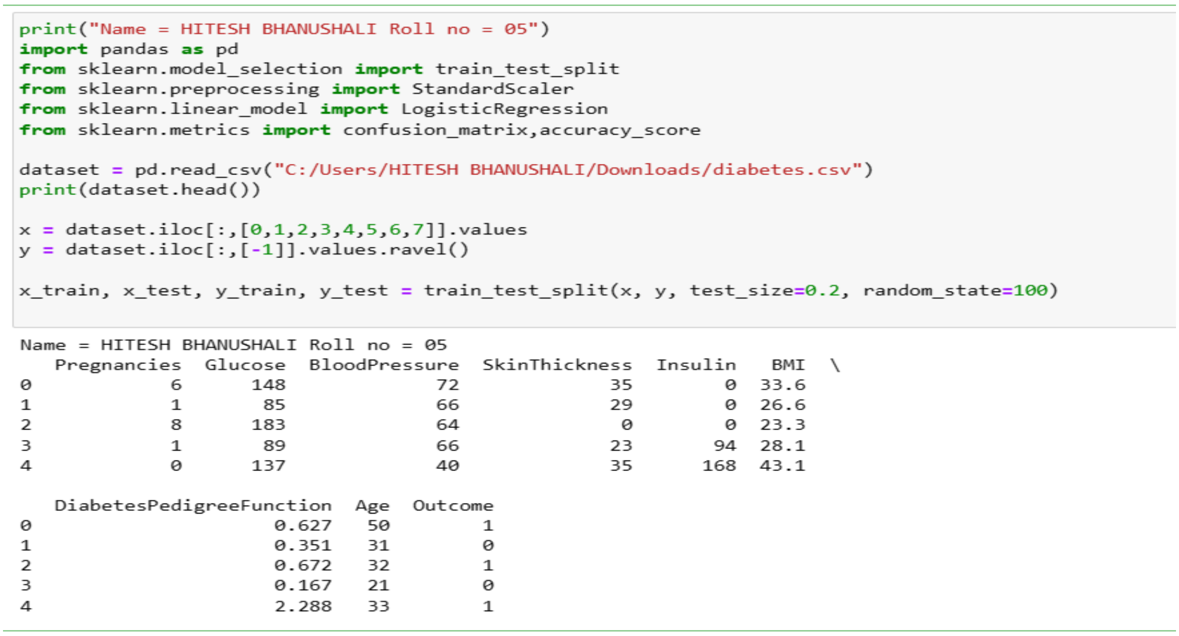
Aim: For a given set of training data examples stored in a .CSV file implement Logistic Regression algorithm.

Code:



	A	B	C	D	E	F	G	H	I
1	Pregnanci	Glucose	BloodPres	SkinThickr	Insulin	BMI	DiabetesP	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1
9	10	115	0	0	0	35.3	0.134	29	0
10	2	197	70	45	543	30.5	0.158	53	1
11	8	125	96	0	0	0	0.232	54	1
12	4	110	92	0	0	37.6	0.191	30	0

```
print("Name = HITESH BHANUSHALI Roll no = 05")
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("C:/Users/HITESH BHANUSHALI/Downloads/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)
```



```
print("Name = HITESH BHANUSHALI Roll no = 05")
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("C:/Users/HITESH BHANUSHALI/Downloads/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)
```

```
Name = HITESH BHANUSHALI Roll no = 05
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

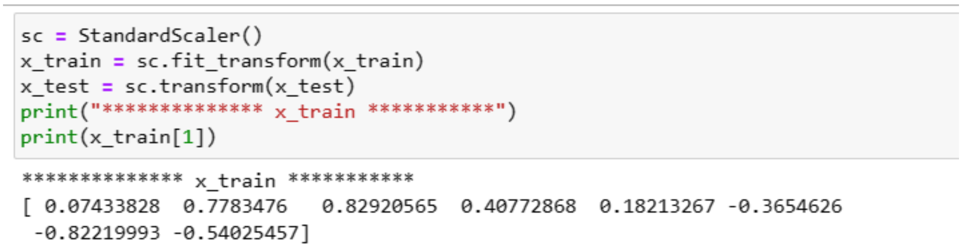
sc = StandardScaler()

x_train = sc.fit_transform(x_train)

x_test = sc.transform(x_test)

print("************** x_train ***********")

print(x_train[1])



```
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
print("************** x_train ***********")
print(x_train[1])
```

```
************** x_train ***********
[ 0.07433828  0.7783476   0.82920565  0.40772868  0.18213267 -0.3654626
 -0.82219993 -0.54025457]
```

model = LogisticRegression()

model.fit(x_train, y_train)

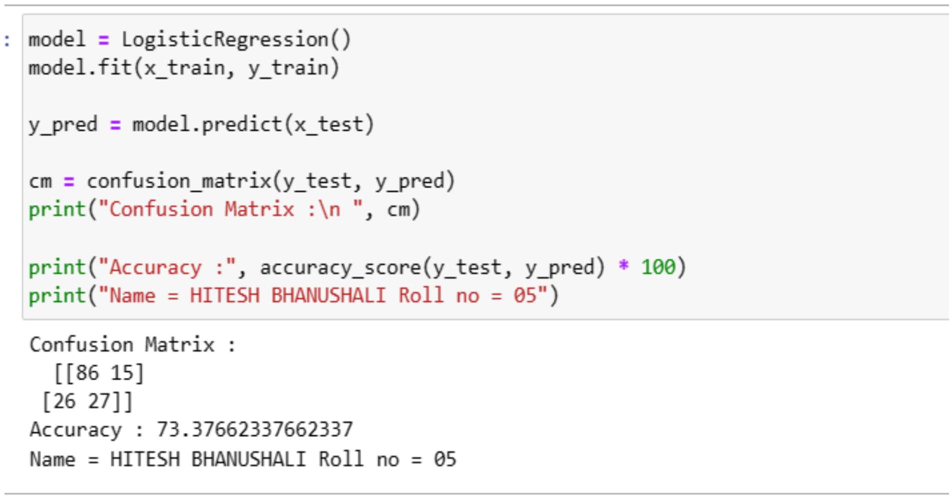
y_pred = model.predict(x_test)

cm = confusion_matrix(y_test, y_pred)

print("Confusion Matrix :\n ", cm)

print("Accuracy :", accuracy_score(y_test, y_pred) * 100)

print("Name = HITESH BHANUSHALI Roll no = 05")



```
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("Name = HITESH BHANUSHALI Roll no = 05")
```

```
Confusion Matrix :
   [[86 15]
 [26 27]]
Accuracy : 73.37662337662337
Name = HITESH BHANUSHALI Roll no = 05
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

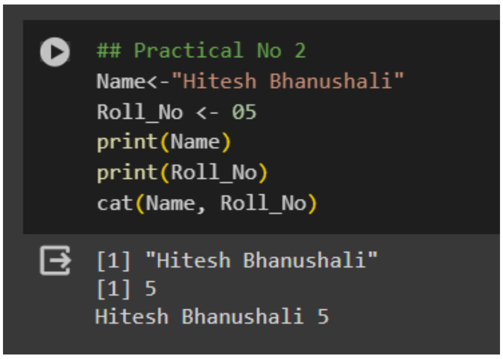
M.Sc (I.T.) Part-1 Semester I

PRACTICAL NO 2-A

Aim: Introduction to R programming

Code:

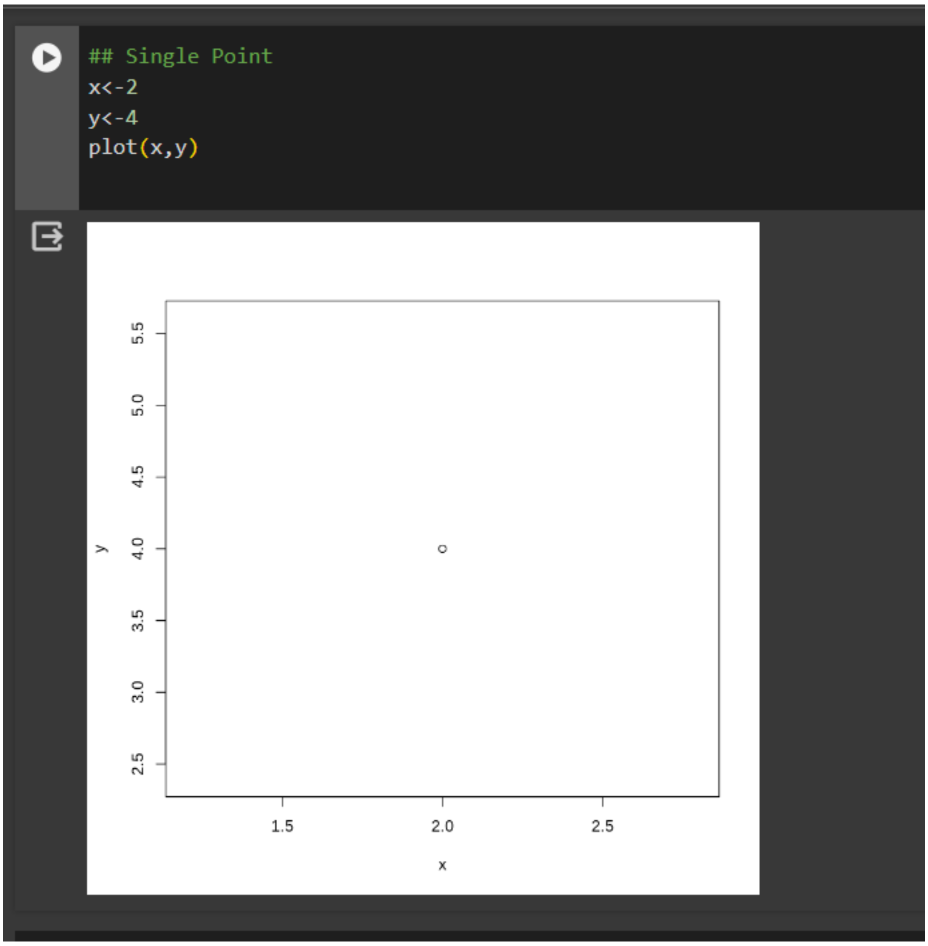
Creating Variables in R:



```
## Practical No 2
Name<-"Hitesh Bhanushali"
Roll_No <- 05
print(Name)
print(Roll_No)
cat(Name, Roll_No)
```

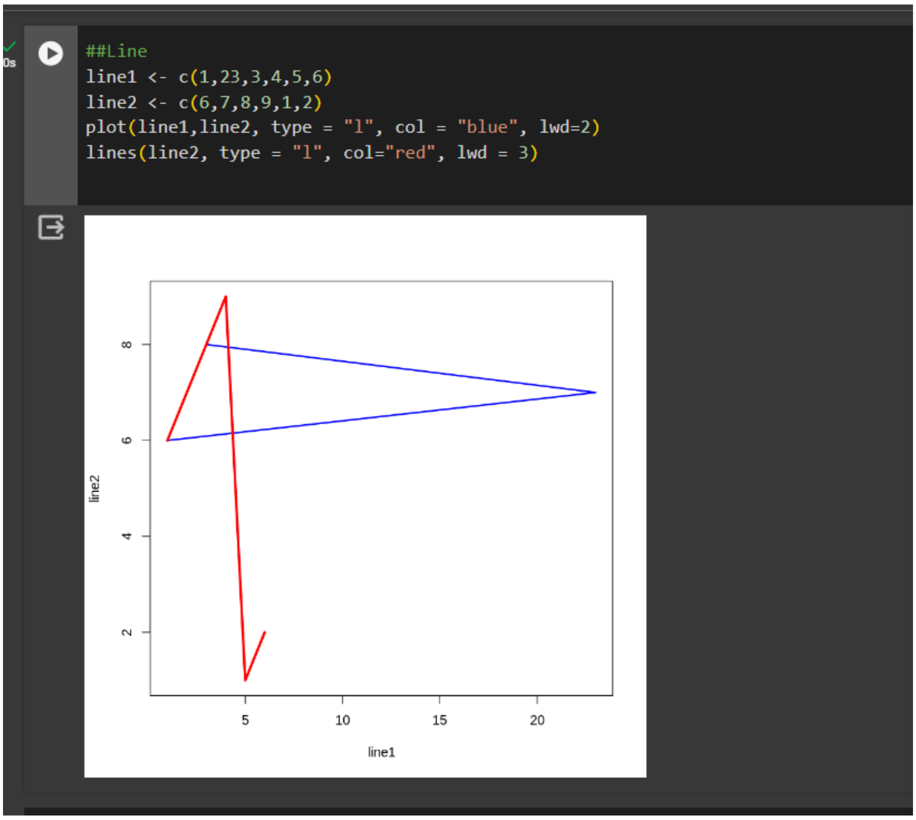
```
[1] "Hitesh Bhanushali"
[1] 5
Hitesh Bhanushali 5
```

Single Point:



```
## Single Point
x<-2
y<-4
plot(x,y)
```

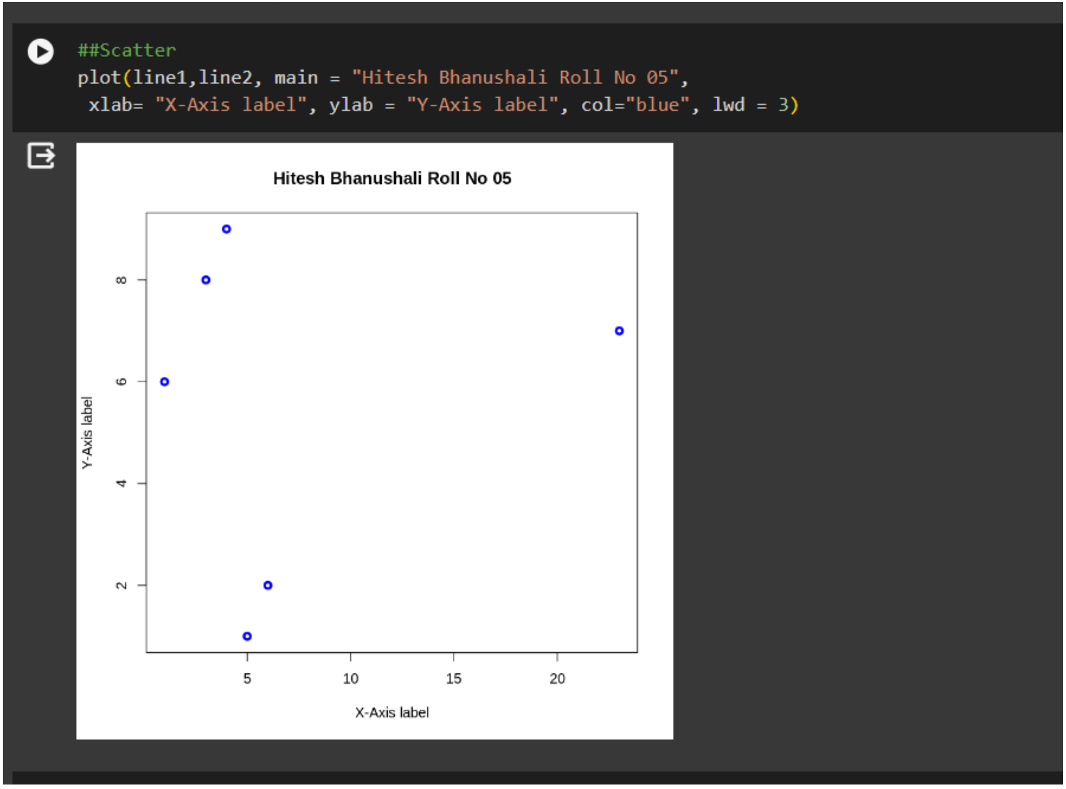
R Plotting, R Line, Color, Width & Multiline:



```
##Line
line1 <- c(1,23,3,4,5,6)
line2 <- c(6,7,8,9,1,2)
plot(line1,line2, type = "l", col = "blue", lwd=2)
lines(line2, type = "l", col="red", lwd = 3)
```

Scatter Plot:

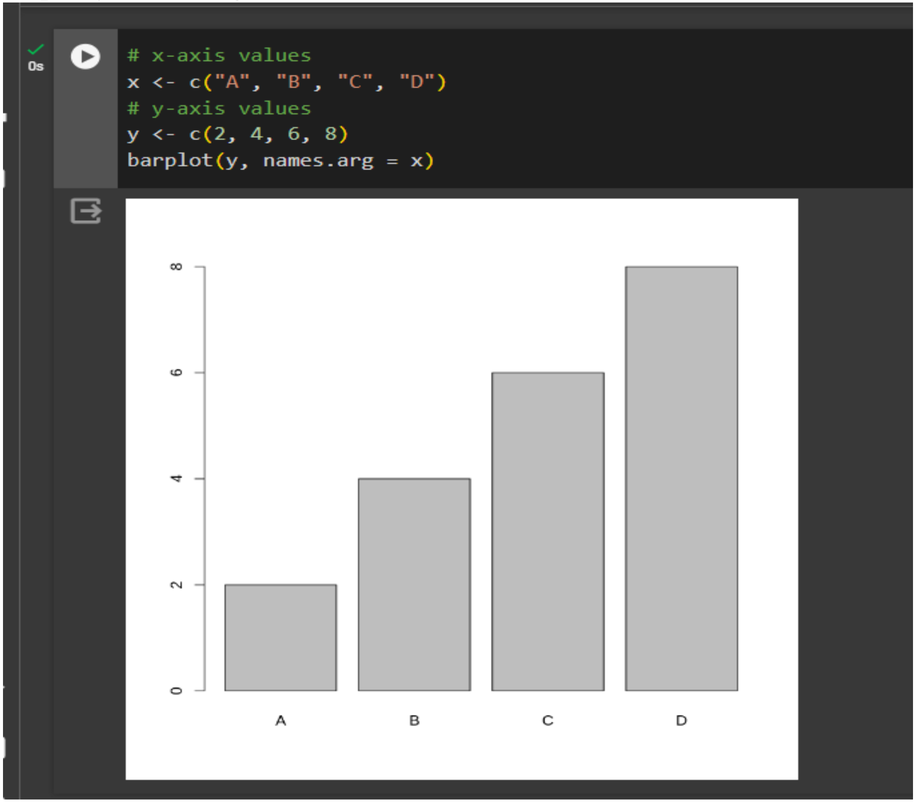
plot(line1,line2, main = "Hitesh Bhanushali Roll No 05",
 xlab= "X-Axis label", ylab = "Y-Axis label", col="blue", lwd = 3)



```
##Scatter
plot(line1,line2, main = "Hitesh Bhanushali Roll No 05",
 xlab= "X-Axis label", ylab = "Y-Axis label", col="blue", lwd = 3)
```

Bar Charts:

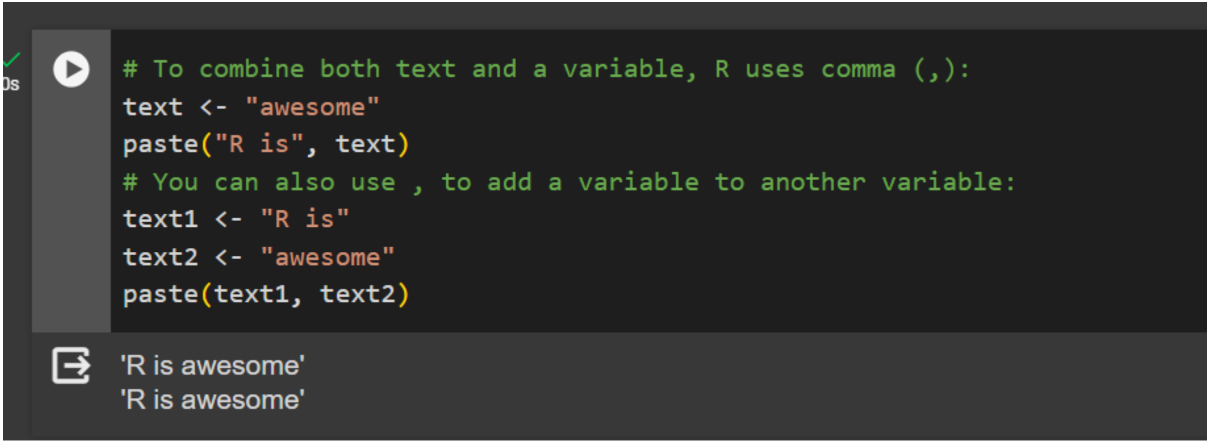
```
# x-axis values
x <- c("A", "B", "C", "D")
# y-axis values
y <- c(2, 4, 6, 8)
barplot(y, names.arg = x)
```



```
# x-axis values
x <- c("A", "B", "C", "D")
# y-axis values
y <- c(2, 4, 6, 8)
barplot(y, names.arg = x)
```

Concatenate Elements:

```
# To combine both text and a variable, R uses comma (,):
text <- "awesome"
paste("R is", text)
# You can also use , to add a variable to another variable:
text1 <- "R is"
text2 <- "awesome"
cat(text1, text2)
```



```
# To combine both text and a variable, R uses comma (,):
text <- "awesome"
paste("R is", text)
# You can also use , to add a variable to another variable:
text1 <- "R is"
text2 <- "awesome"
paste(text1, text2)
```

```
'R is awesome'
'R is awesome'
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

M.Sc (I.T.) Part-1 Semester I

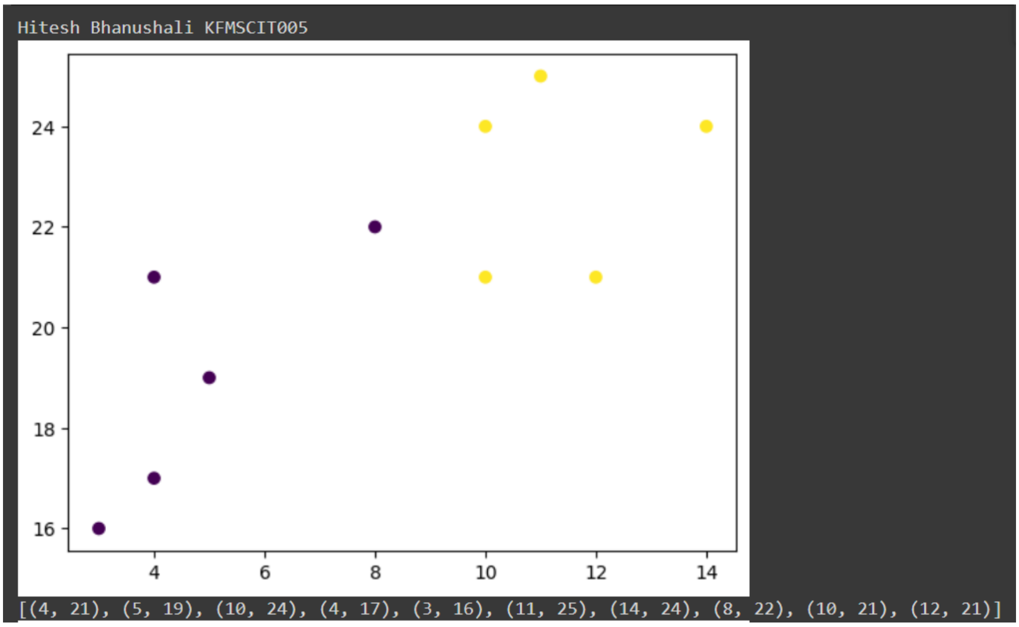
Practical No.2-B

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

Code:

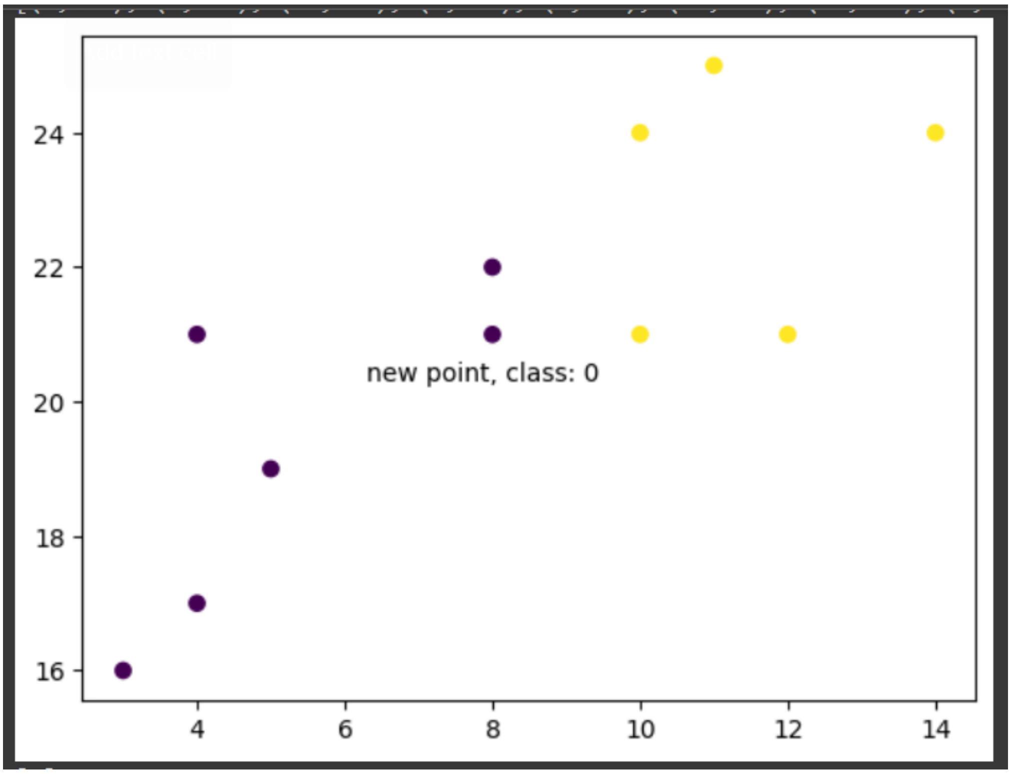
```
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
print("Hitesh Bhanushali KFMSCIT005")
#Data Set
x = [4, 5, 10, 4, 3, 11, 14 , 8, 10, 12]
y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]
classes = [0, 0, 1, 0, 0, 1, 1, 0, 1, 1]
#To plot data set
plt.scatter(x, y, c=classes)
plt.show()

#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)

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()

model = KNeighborsClassifier(n_neighbors=5)
model.fit(data, classes)
prediction = model.predict(new_point)
print(prediction)
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("Hitesh Bhanushali KFMSCIT005")
```

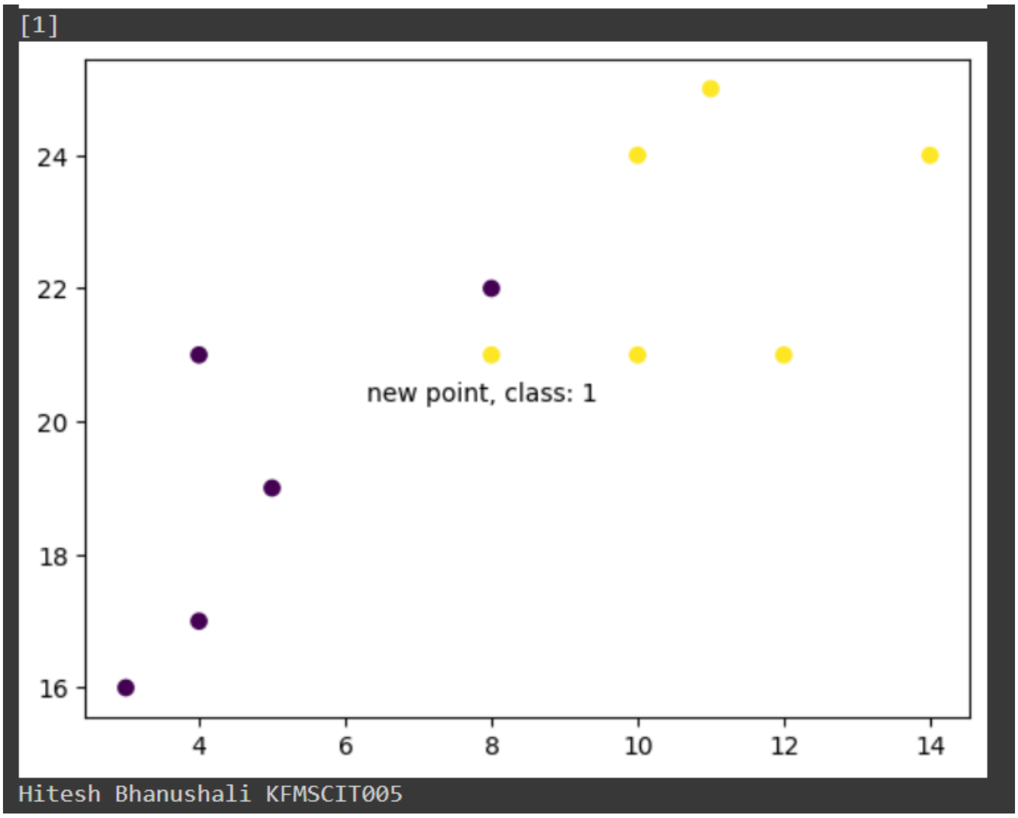
Output:



```
Hitesh Bhanushali KFMSCIT005
```

```
[(4, 21), (5, 19), (10, 24), (4, 17), (3, 16), (11, 25), (14, 24), (8, 22), (10, 21), (12, 21)]
```





```
[1]
```

```
Hitesh Bhanushali KFMSCIT005
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

M.Sc (I.T.) Part-1 Semester I

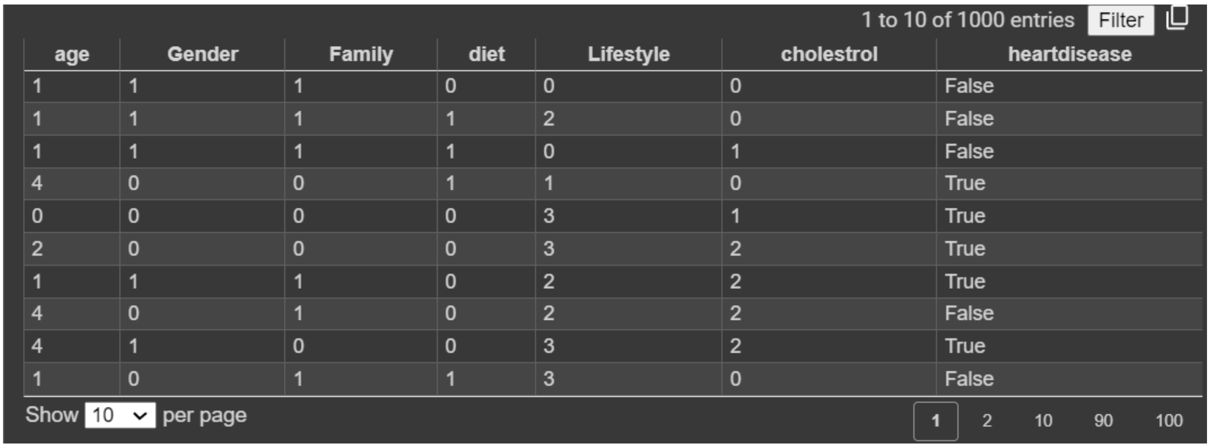
Practical No.3: Naive Bayes Algorithm

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:

```
!pip install pgmpy
import pandas as pd
from pgmpy.estimators import MaximumLikelihoodEstimator
from pgmpy.models import BayesianModel
from pgmpy.inference import VariableElimination
print("Hitesh Bhanushali KFMSCIT005")
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')
])
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')

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)
print("Hitesh Bhanushali KFMSCIT005")
```

Output:

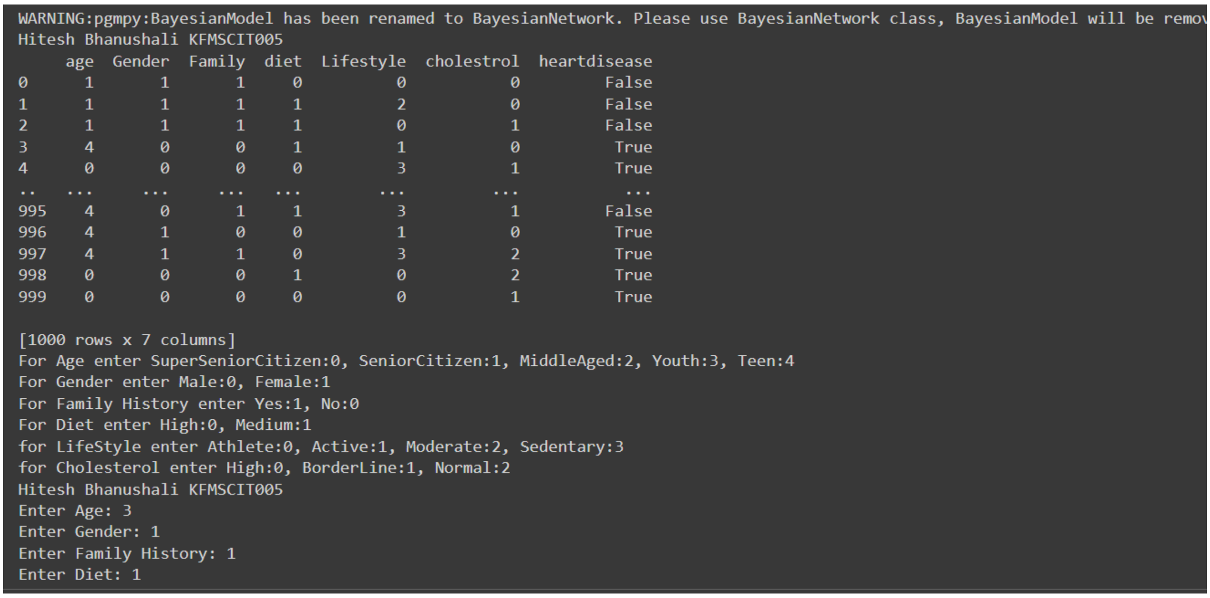


1 to 10 of 1000 entries Filter

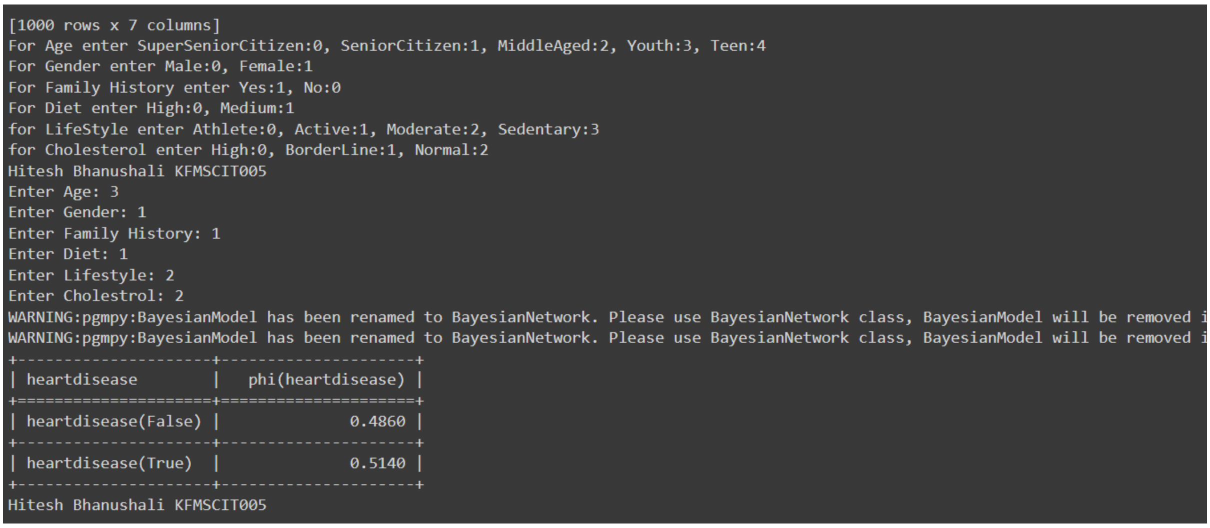
age	Gender	Family	diet	Lifestyle	cholestrol	heartdisease
1	1	1	0	0	0	False
1	1	1	1	2	0	False
1	1	1	1	0	1	False
4	0	0	1	1	0	True
0	0	0	0	3	1	True
2	0	0	0	3	2	True
1	1	1	0	2	2	True
4	0	1	0	2	2	False
4	1	0	0	3	2	True
1	0	1	1	3	0	False

Show 10 per page

1 2 10 90 100



```
WARNING:pgmpy:BayesianModel has been renamed to BayesianNetwork. Please use BayesianNetwork class, BayesianModel will be remov
Hitesh Bhanushali KFMSCIT005
     age  Gender  Family  diet  Lifestyle  cholestrol  heartdisease
0      1       1       1     0          0           0         False
1      1       1       1     1          2           0         False
2      1       1       1     1          0           1         False
3      4       0       0     1          1           0          True
4      0       0       0     0          3           1          True
..   ...     ...     ...   ...        ...         ...           ...
995    4       0       1     1          3           1         False
996    4       1       0     0          1           0          True
997    4       1       1     0          3           2          True
998    0       0       0     1          0           2          True
999    0       0       0     0          0           1          True

[1000 rows x 7 columns]
For Age enter SuperSeniorCitizen:0, SeniorCitizen:1, MiddleAged:2, Youth:3, Teen:4
For Gender enter Male:0, Female:1
For Family History enter Yes:1, No:0
For Diet enter High:0, Medium:1
for LifeStyle enter Athlete:0, Active:1, Moderate:2, Sedentary:3
for Cholesterol enter High:0, BorderLine:1, Normal:2
Hitesh Bhanushali KFMSCIT005
Enter Age: 3
Enter Gender: 1
Enter Family History: 1
Enter Diet: 1
```



```
[1000 rows x 7 columns]
For Age enter SuperSeniorCitizen:0, SeniorCitizen:1, MiddleAged:2, Youth:3, Teen:4
For Gender enter Male:0, Female:1
For Family History enter Yes:1, No:0
For Diet enter High:0, Medium:1
for LifeStyle enter Athlete:0, Active:1, Moderate:2, Sedentary:3
for Cholesterol enter High:0, BorderLine:1, Normal:2
Hitesh Bhanushali KFMSCIT005
Enter Age: 3
Enter Gender: 1
Enter Family History: 1
Enter Diet: 1
Enter Lifestyle: 2
Enter Cholestrol: 2
WARNING:pgmpy:BayesianModel has been renamed to BayesianNetwork. Please use BayesianNetwork class, BayesianModel will be removed i
WARNING:pgmpy:BayesianModel has been renamed to BayesianNetwork. Please use BayesianNetwork class, BayesianModel will be removed i
+---------------------+---------------------+
| heartdisease        |   phi(heartdisease) |
+=====================+=====================+
| heartdisease(False) |              0.4860 |
+---------------------+---------------------+
| heartdisease(True)  |              0.5140 |
+---------------------+---------------------+
Hitesh Bhanushali KFMSCIT005
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

M.Sc (I.T.) Part-1 Semester I

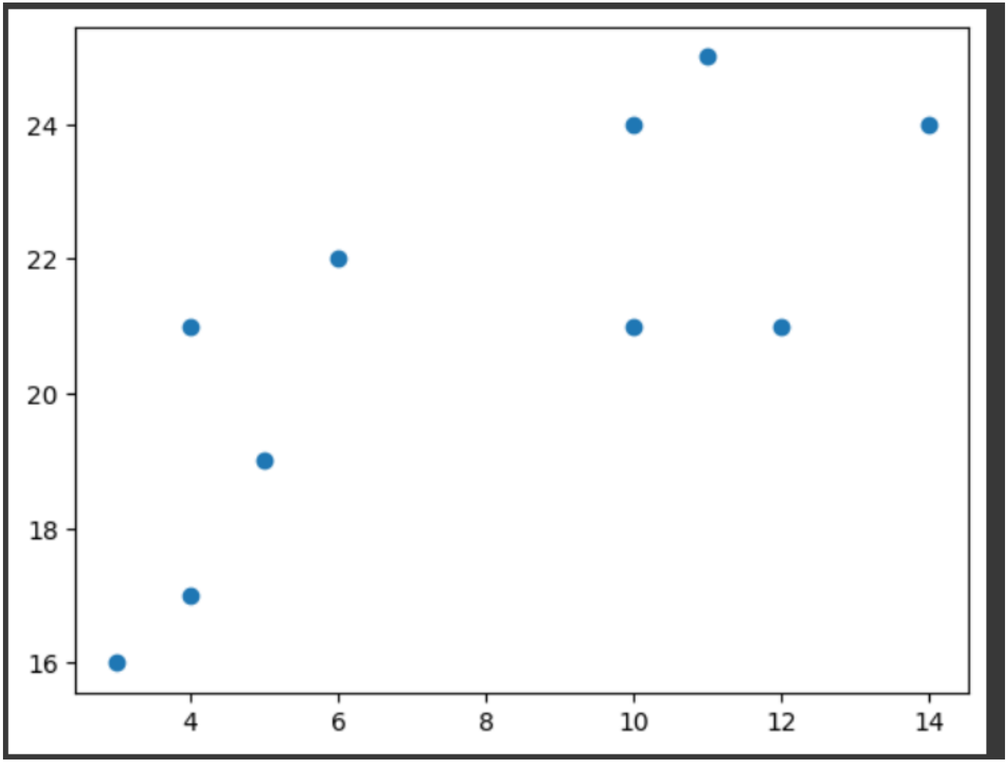
Practical No.4: K-Means Clustering

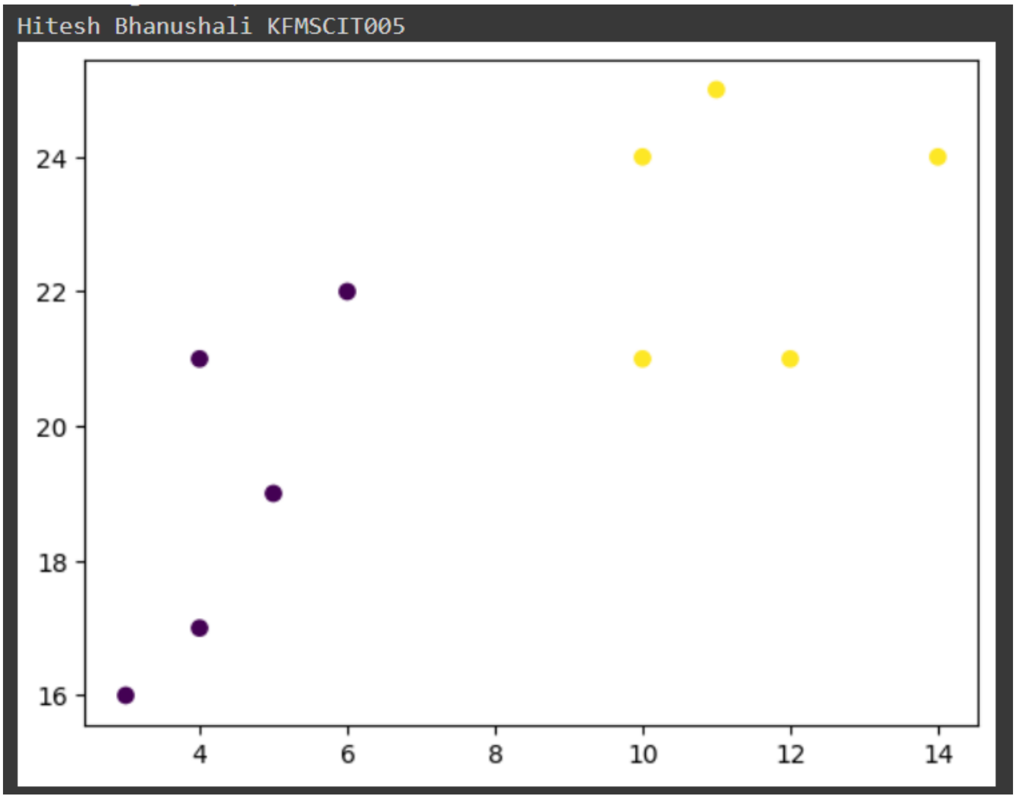
Aim: Implement the classification model using clustering for the following techniques with K means clustering with Prediction

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()
data = list(zip(x, y))

kmeans = KMeans(n_clusters=2)
kmeans.fit(data)
print("Hitesh Bhanushali KFMSCIT005")
plt.scatter(x, y, c=kmeans.labels_)
plt.show()
```

Output:





```
Hitesh Bhanushali KFMSCIT005
```

KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

M.Sc (I.T.) Part-1 Semester I

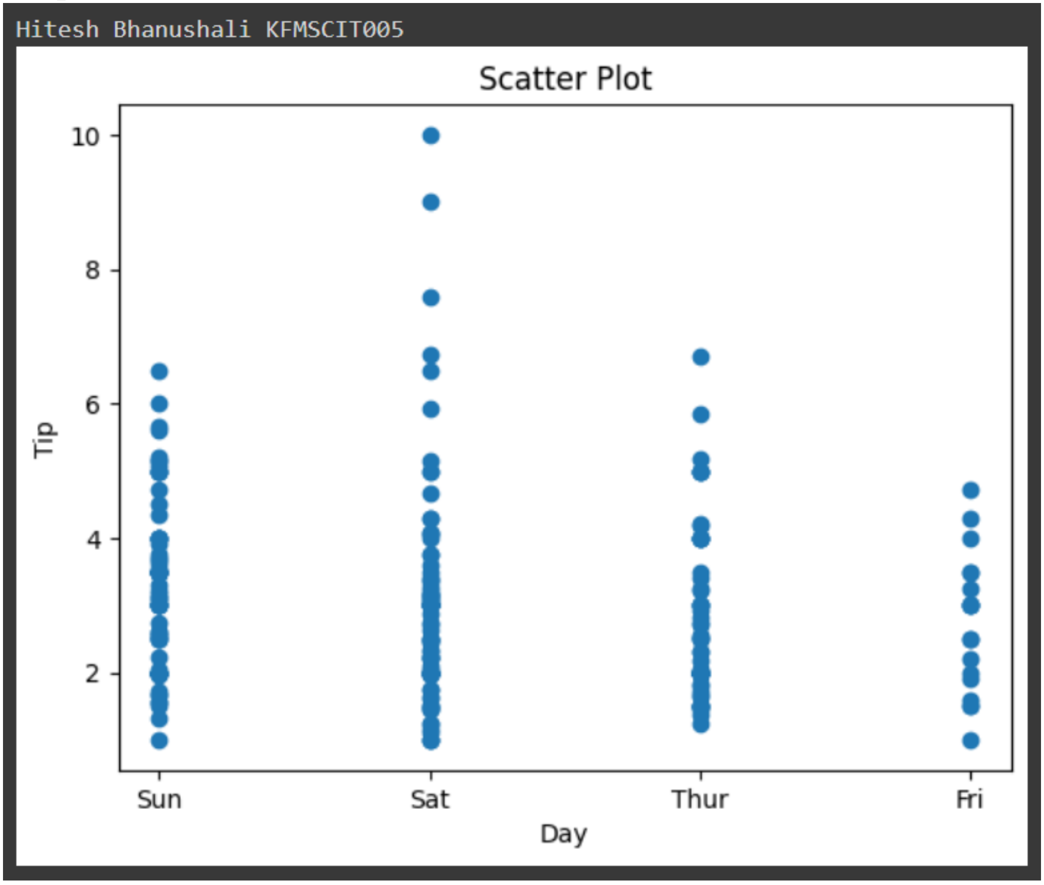
Practical No.5: Data Visualization Using Python

Aim: Python provides various libraries that come with different features for visualizing data. All these libraries come with different features and can support various types of graphs. Matplotlib, Seaborn, Bokeh, Plotly

Code

```
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")

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

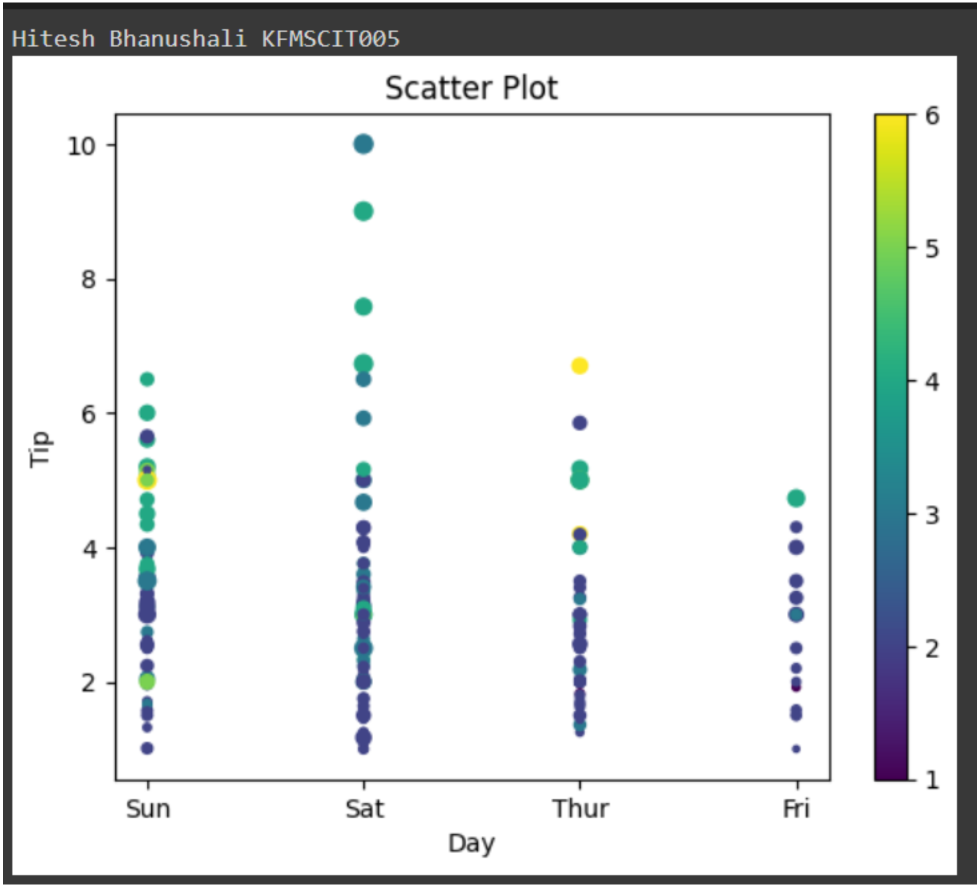
Output:



Code:

```
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")

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

Output:



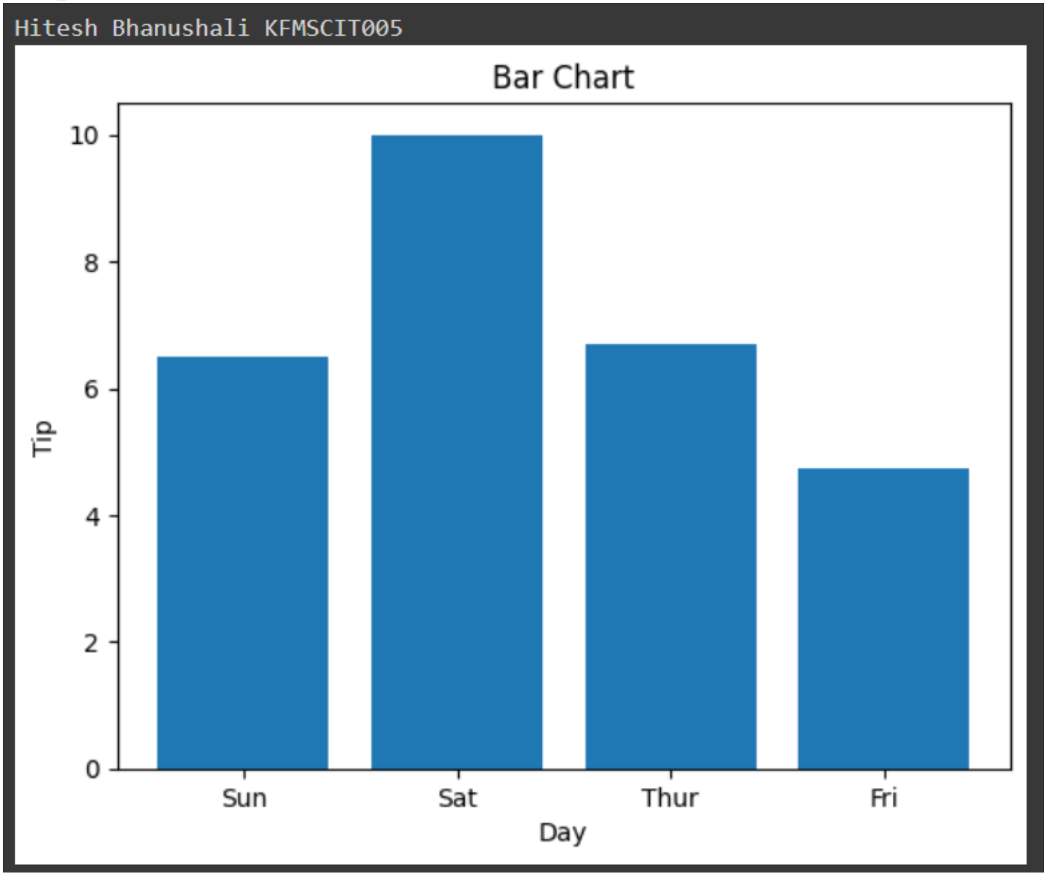
Practical No.6 Pie Diagram, Bar Chart Diagram using Python

Aim: Bar Chart. A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. It can be created using the bar() method.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
print("Hitesh Bhanushali KFMSCIT005")
# reading the database
data = pd.read_csv("tips.csv")

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

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

# Adding the legends
plt.show()
```

Output:

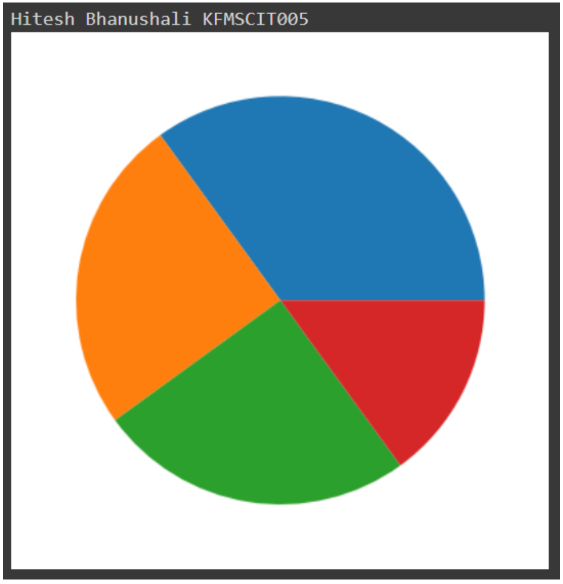


```
Hitesh Bhanushali KFMSCIT005
```

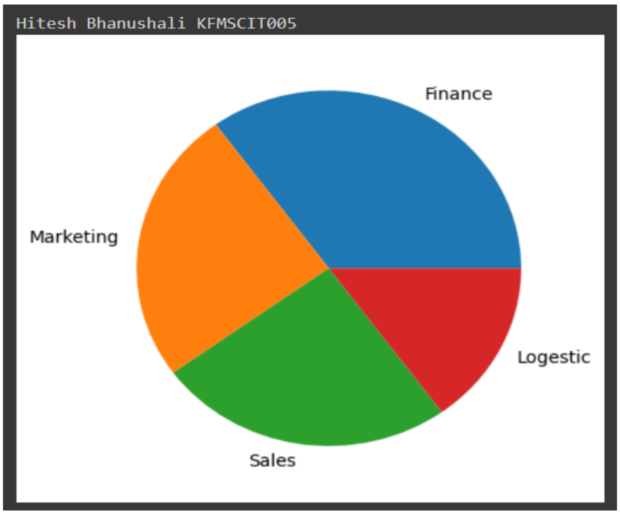
Creating Pie Charts. With Pyplot, you can use the pie() function to draw pie charts:

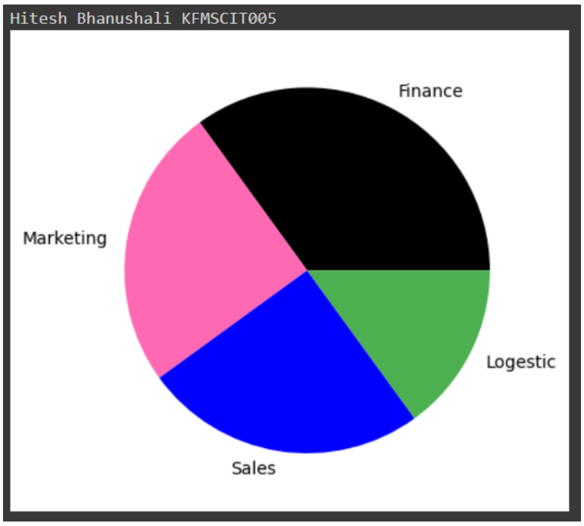
Code:

```
import pandas as pd
import matplotlib.pyplot as plt
print("Hitesh Bhanushali KFMSCIT005")
y = np.array([35,25,25,15])
plt.pie(y)
plt.show()
```



```
import pandas as pd
import matplotlib.pyplot as plt
print("Hitesh Bhanushali KFMSCIT005")
y = np.array([35,25,25,15])
mylabels = ["Finance", "Marketing","Sales","Logestic"]
mycolors = [“black”,”hotpink”,”b”,”#459CGF”]
plt.pie(y, labels = mylabels, colors = mycolors)
plt.show()
```





KISHINCHAND CHELLARAM COLLEGE, MUMBAI -20

M.Sc (I.T.) Part-1 Semester I

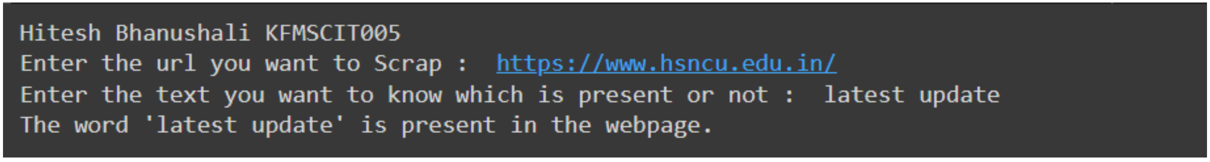
Practical No.7: ETL process in Python for Web Scrapping

Aim: Perform ETL Process in Python for Web Scrapping on any website and search for a specific data.

Code:

```
import requests
from bs4 import BeautifulSoup
print("Hitesh Bhanushali KFMSCIT005")
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:



```
Hitesh Bhanushali KFMSCIT005
Enter the url you want to Scrap :  https://www.hsncu.edu.in/
Enter the text you want to know which is present or not :  latest update
The word 'latest update' is present in the webpage.
```

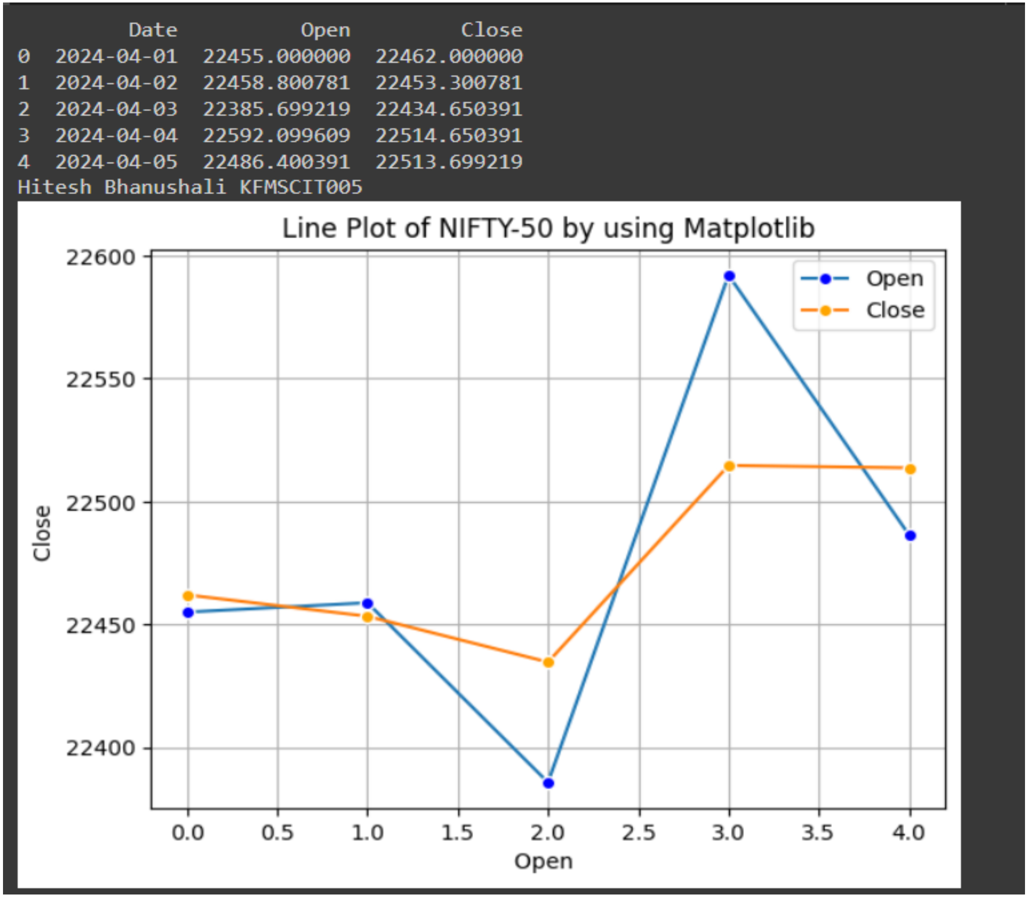
Practical No-8: Data visualization using python libraries?

MATPLOTLIB

CODE:

```
import Libraries
import pandas as pd
import matplotlib. pyplot as plt
#Uploading Dataset
data = pd.read_csv("Nifty.csv")
print(data)
print("Hitesh Bhanushali KFMSCIT005")

#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:



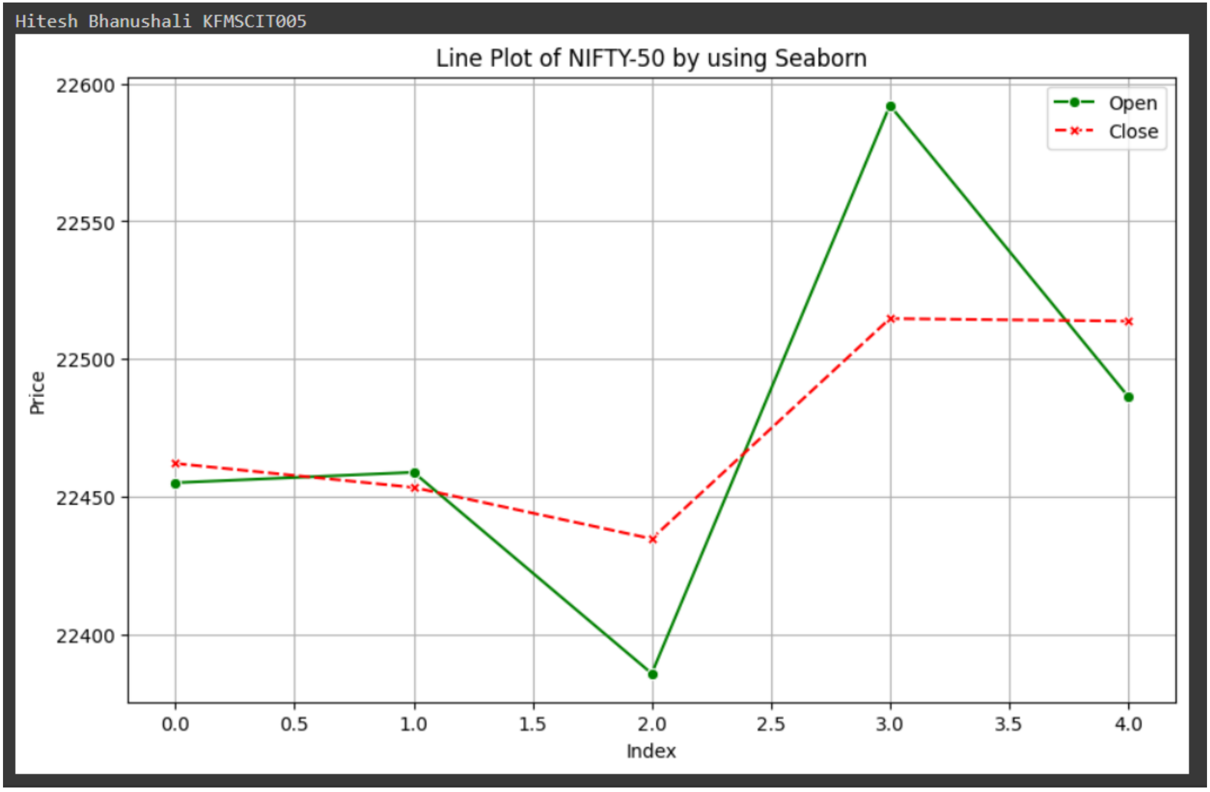
```
         Date          Open         Close
0  2024-04-01  22455.000000  22462.000000
1  2024-04-02  22458.800781  22453.300781
2  2024-04-03  22385.699219  22434.650391
3  2024-04-04  22592.099609  22514.650391
4  2024-04-05  22486.400391  22513.699219
Hitesh Bhanushali KFMSCIT005
```

SEABORN

CODE :

```
#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("Hitesh Bhanushali KFMSCIT005")
# 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:



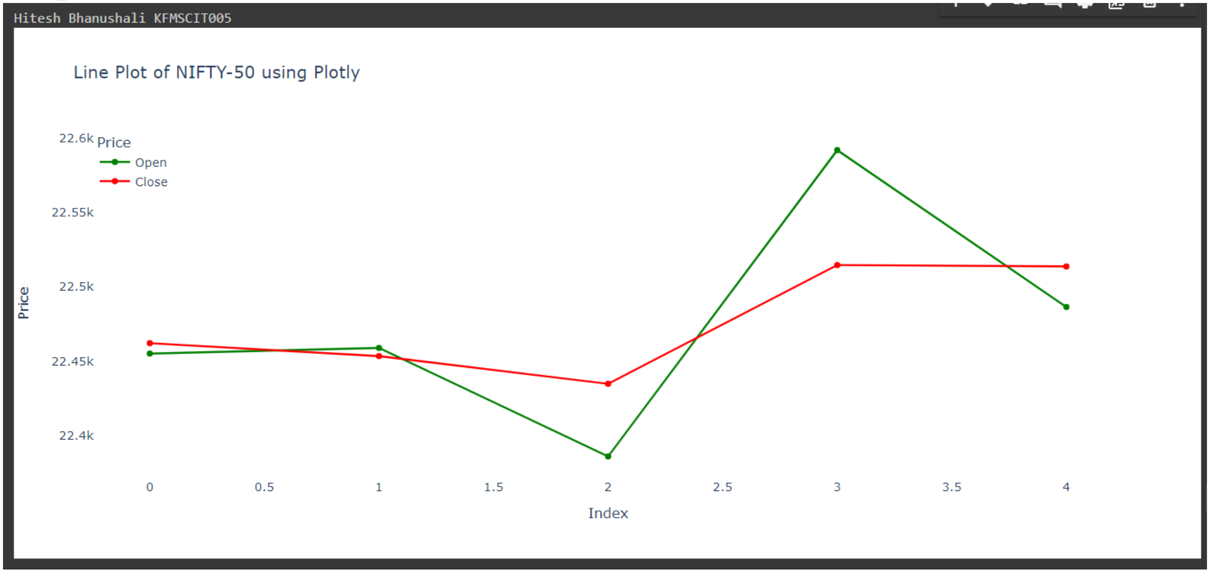
```
Hitesh Bhanushali KFMSCIT005
```

PLOTLY

CODE:

```
#DATA VISUALIZATION USING PLOTLY
import pandas as pd
import plotly.express as px

# Load dataset
data = pd.read_csv("Nifty.csv")
print("Hitesh Bhanushali KFMSCIT005")
# 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:



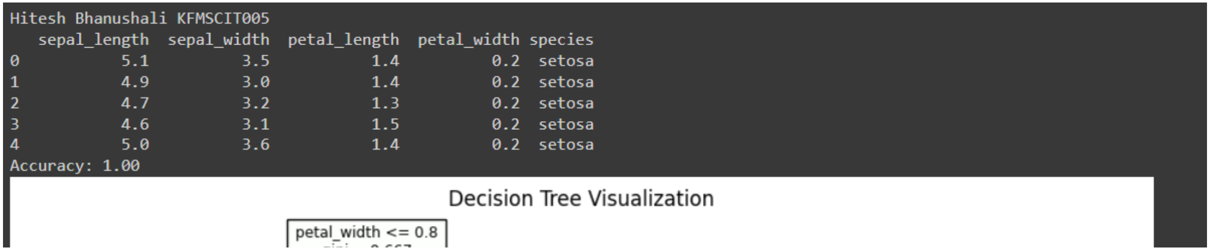
```
Hitesh Bhanushali KFMSCIT005
```

Practical No-9: Decision Tree Classifier

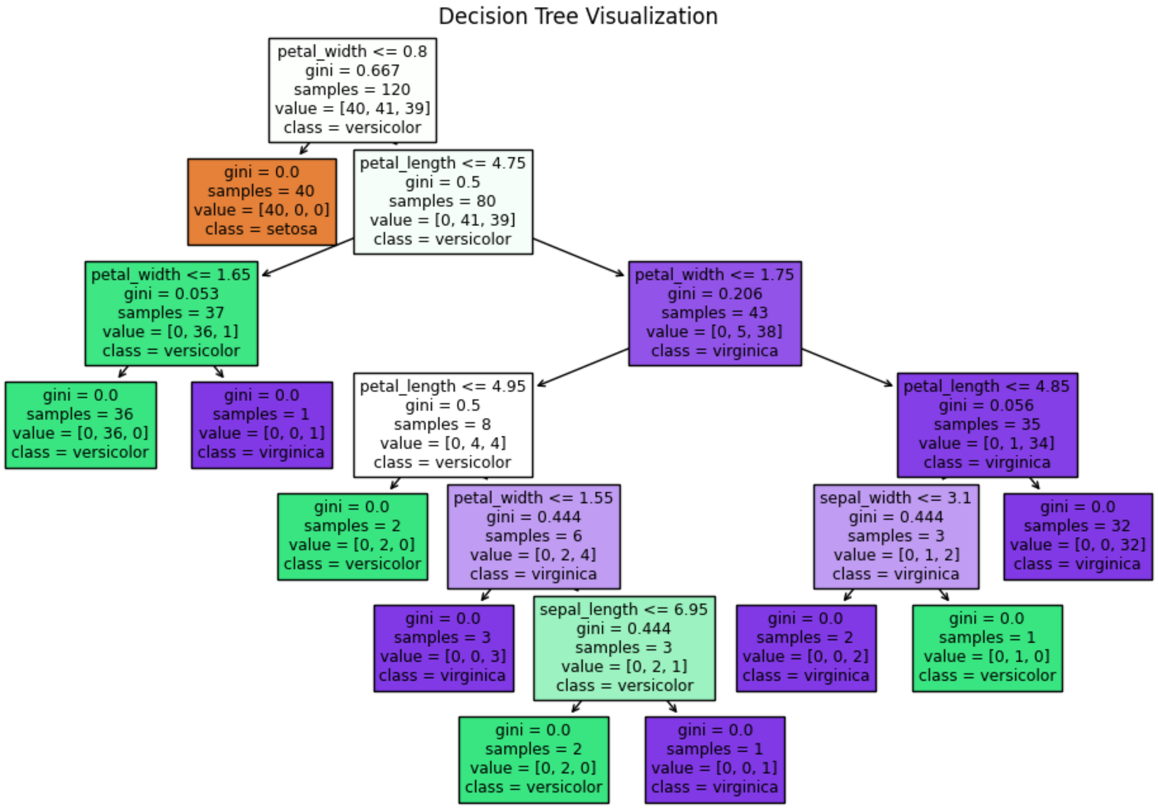
Code:

```
import numpy as np
import pandas as pd # Import Pandas for data loading
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("Hitesh Bhanushali KFMSCIT005")
data = pd.read_csv('iris.csv')
print(data.head())
# Assuming the target variable is in a column named 'target'
X = data.drop('species', axis=1)
y = data['species']
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}")
# Visualize and interpret the generated decision tree
plt.figure(figsize=(12, 8))
plot_tree(model, filled=True, feature_names=X.columns,
class_names=y.unique().astype(str))
plt.title("Decision Tree Visualization")
plt.show()
```

Output:



```
Hitesh Bhanushali KFMSCIT005
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
Accuracy: 1.00
```


Decision Tree Visualization
petal_width <= 0.8
gini = 0.667
samples = 120
value = [40, 41, 39]
class = versicolor
gini = 0.0
samples = 40
value = [40, 0, 0]
class = setosa
petal_length <= 4.75
gini = 0.5
samples = 80
value = [0, 41, 39]
class = versicolor
petal_width <= 1.65
gini = 0.053
samples = 37
value = [0, 36, 1]
class = versicolor
petal_width <= 1.75
gini = 0.206
samples = 43
value = [0, 5, 38]
class = virginica
gini = 0.0
samples = 36
value = [0, 36, 0]
class = versicolor
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
petal_length <= 4.95
gini = 0.5
samples = 8
value = [0, 4, 4]
class = versicolor
petal_length <= 4.85
gini = 0.056
samples = 35
value = [0, 1, 34]
class = virginica
gini = 0.0
samples = 2
value = [0, 2, 0]
class = versicolor
petal_width <= 1.55
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
sepal_width <= 3.1
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
gini = 0.0
samples = 32
value = [0, 0, 32]
class = virginica
gini = 0.0
samples = 3
value = [0, 0, 3]
class = virginica
sepal_length <= 6.95
gini = 0.444
samples = 3
value = [0, 2, 1]
class = versicolor
gini = 0.0
samples = 2
value = [0, 0, 2]
class = virginica
gini = 0.0
samples = 1
value = [0, 1, 0]
class = versicolor
gini = 0.0
samples = 2
value = [0, 2, 0]
class = versicolor
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica