**Program - 5**

**Aim:**

Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.naive\_bayes import GaussianNB

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

dataset=pd.read\_csv('Social\_Network\_Ads.csv')

x=dataset.iloc[:,[2,3]].values

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

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.30)

sc=StandardScaler()

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

x\_test=sc.transform(x\_test)

classifier=GaussianNB()

classifier.fit(x\_train,y\_train)

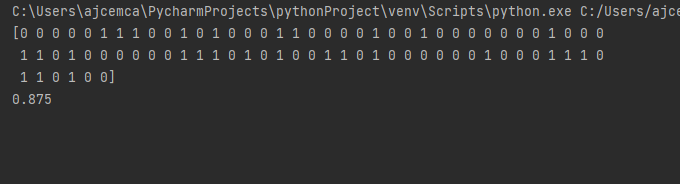
y\_pred=classifier.predict(x\_test)

print(y\_pred)

ac = accuracy\_score(y\_test,y\_pred)

print(ac)

**OUTPUT**

****

**Program - 6**

**Aim:**

Program to implement linear and multiple regression techniques using any standard dataset available in the public domain and evaluate its performance.

**Program(inbuilt):**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

x=np.array([5,15,25,35,45,55]).reshape((-1,1))

y=np.array([5,20,14,32,22,38])

print(x)

print(y)

model=LinearRegression()

model.fit(x,y)

r\_sq=model.score(x,y)

print('coefficent of determination: ',r\_sq)

print('intercept: ',model.intercept\_)

print('slope : ',model.coef\_)

y\_pred=model.predict(x)

print('Predicted response: ',y\_pred)

plt.scatter(x,y,color="g")

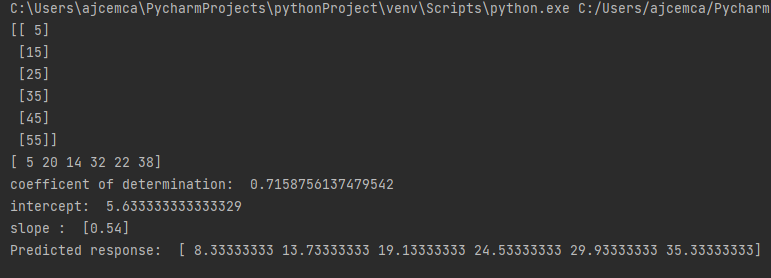
plt.plot(x,y\_pred)

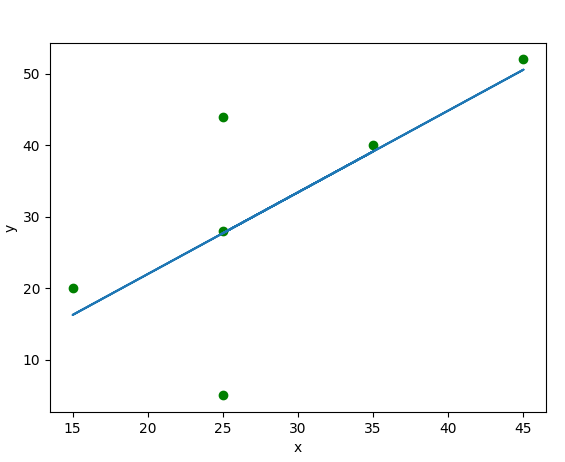
plt.xlabel('x')

plt.ylabel('y')

plt.show()

**OUTPUT**

****

****

**Program(Without inbuilt):**

import numpy as np

import matplotlib.pyplot as plt

def estimate\_coef(x,y):

n=np.size(x)

m\_x=np.mean(x)

m\_y=np.mean(y)

SS\_xy=np.sum(y\*x) - n \*m\_y\* m\_x

SS\_xx=np.sum(x\*x) - n \*m\_x\* m\_x

b\_1=SS\_xy / SS\_xx

b\_0=m\_y - b\_1\* m\_x

return (b\_0,b\_1)

def plot\_regr\_line(x,y,b):

plt.scatter(x,y,color="m",marker="o",s=30)

y\_pred=b[0]+b[1]\*x

plt.plot(x,y\_pred,color="g")

plt.xlabel('x')

plt.ylabel('y')

plt.show()

def main():

x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])

b = estimate\_coef(x, y)

print("Estimated coefficients:\nb\_0 = {} \

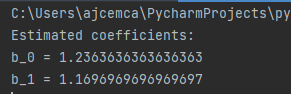
\nb\_1 = {}".format(b[0], b[1]))

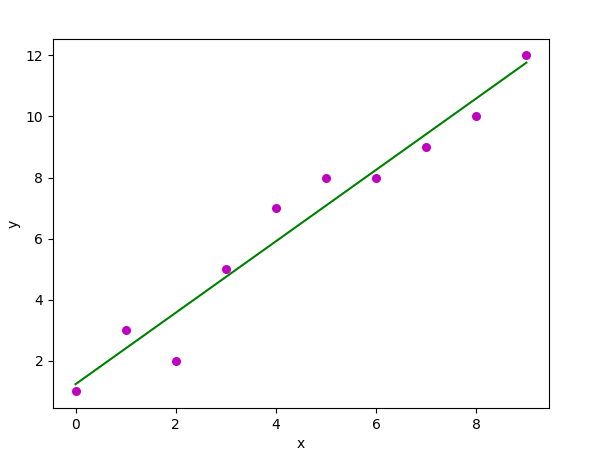
plot\_regr\_line(x, y, b)

if \_\_name\_\_=="\_\_main\_\_":

main()

**OUTPUT**

****

****