Practical 1

Aim : Introduction to googe collab .

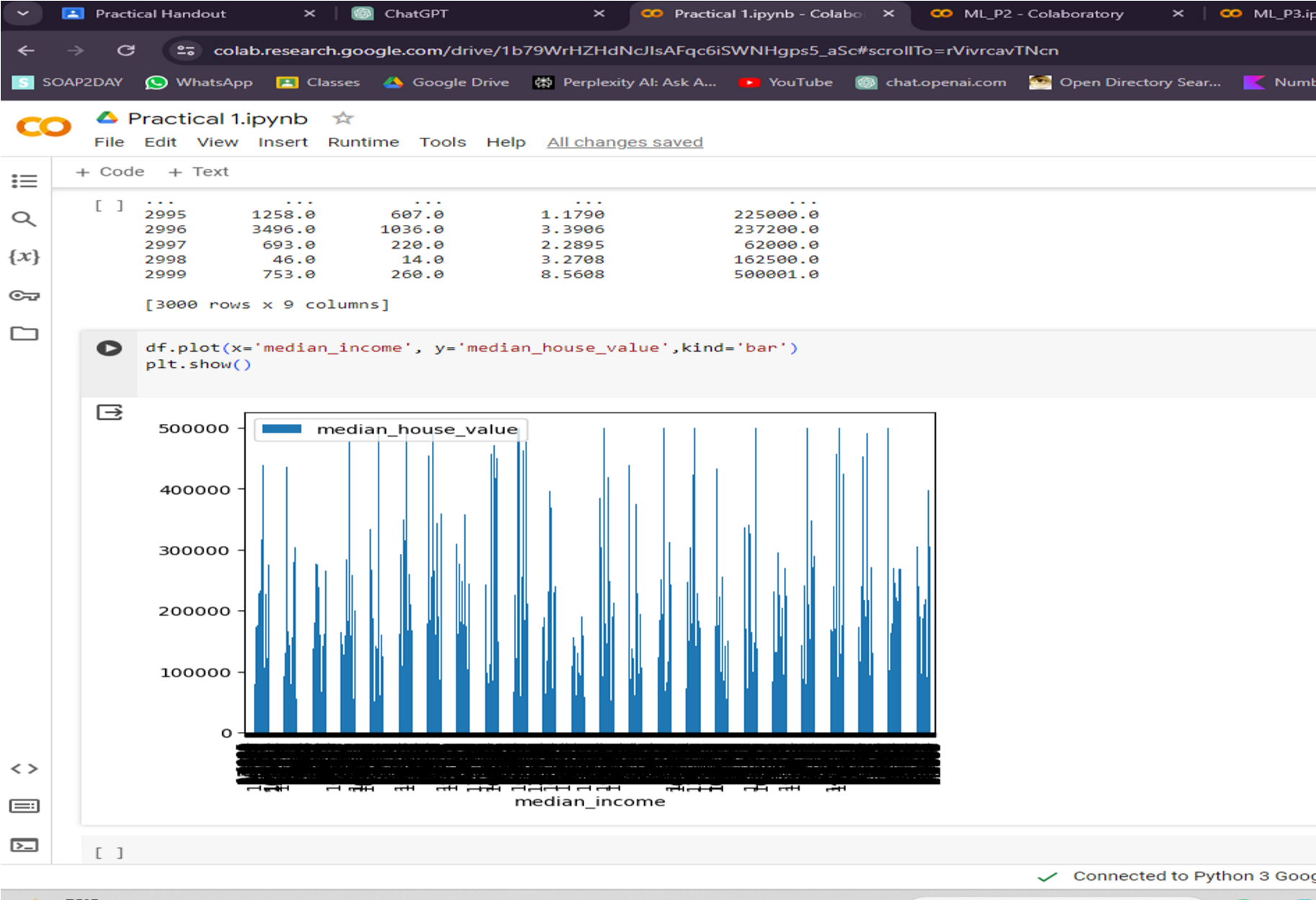
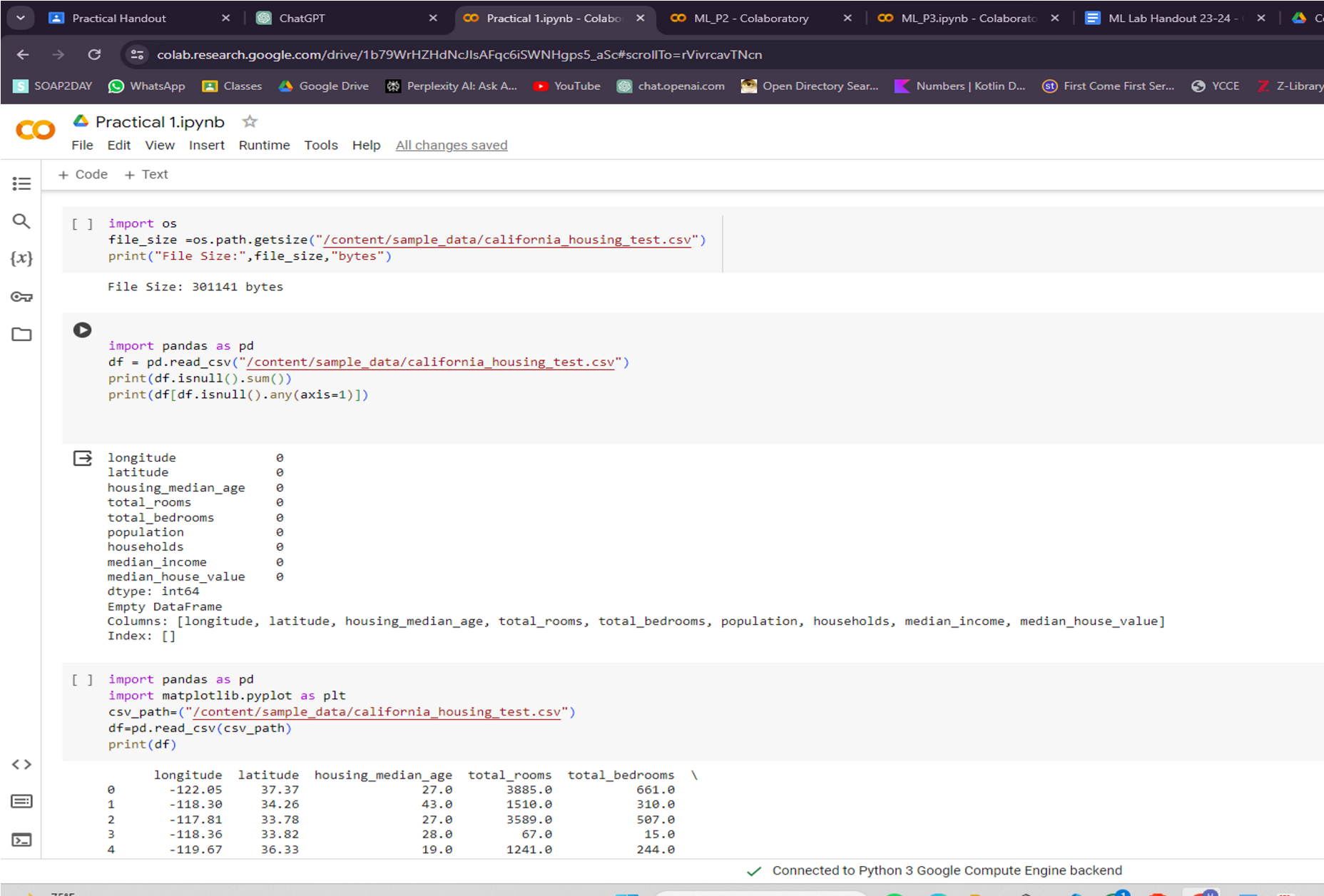
Code:

|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  import os csv\_path=("/content/sample\_data/california\_housing\_test.csv") v=pd.read\_csv(csv\_path)  print(v)  file\_size  =os.path.getsize("/content/sample\_data/california\_housing\_test.csv") print("File Size:",file\_size,"bytes") csv\_path=("/content/sample\_data/california\_housing\_test.csv") df=pd.read\_csv(csv\_path)  print(df) |

df.plot(x='median\_income', y='median\_house\_value',kind='bar') plt.show()

Output:

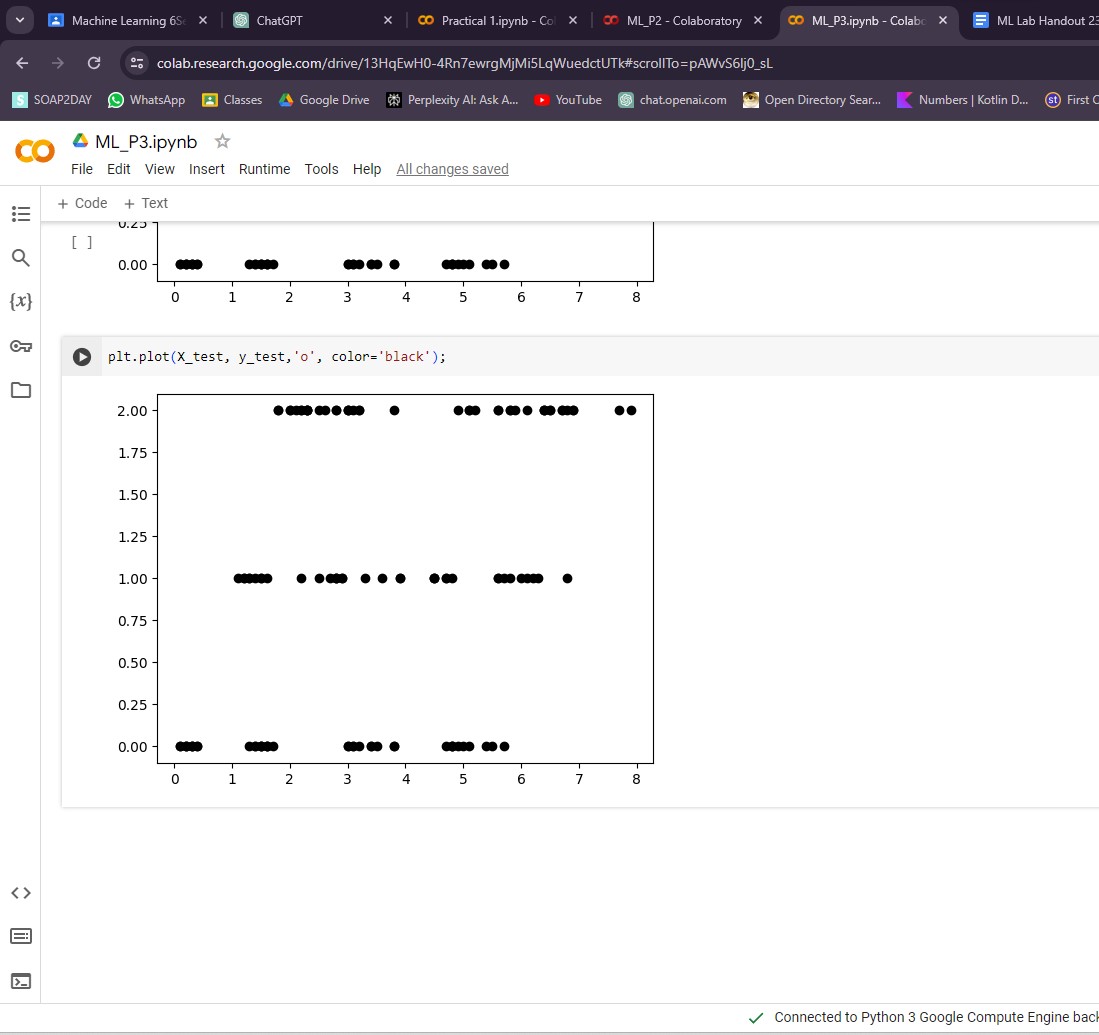
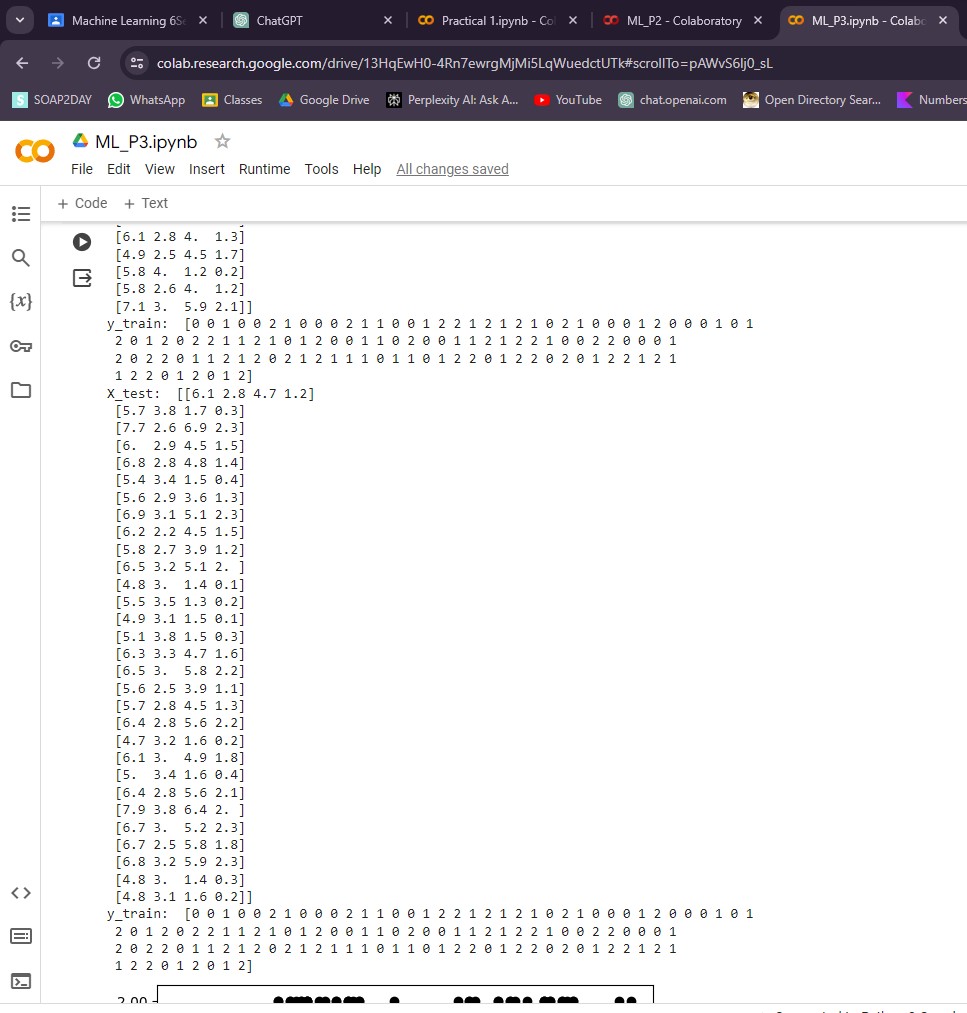
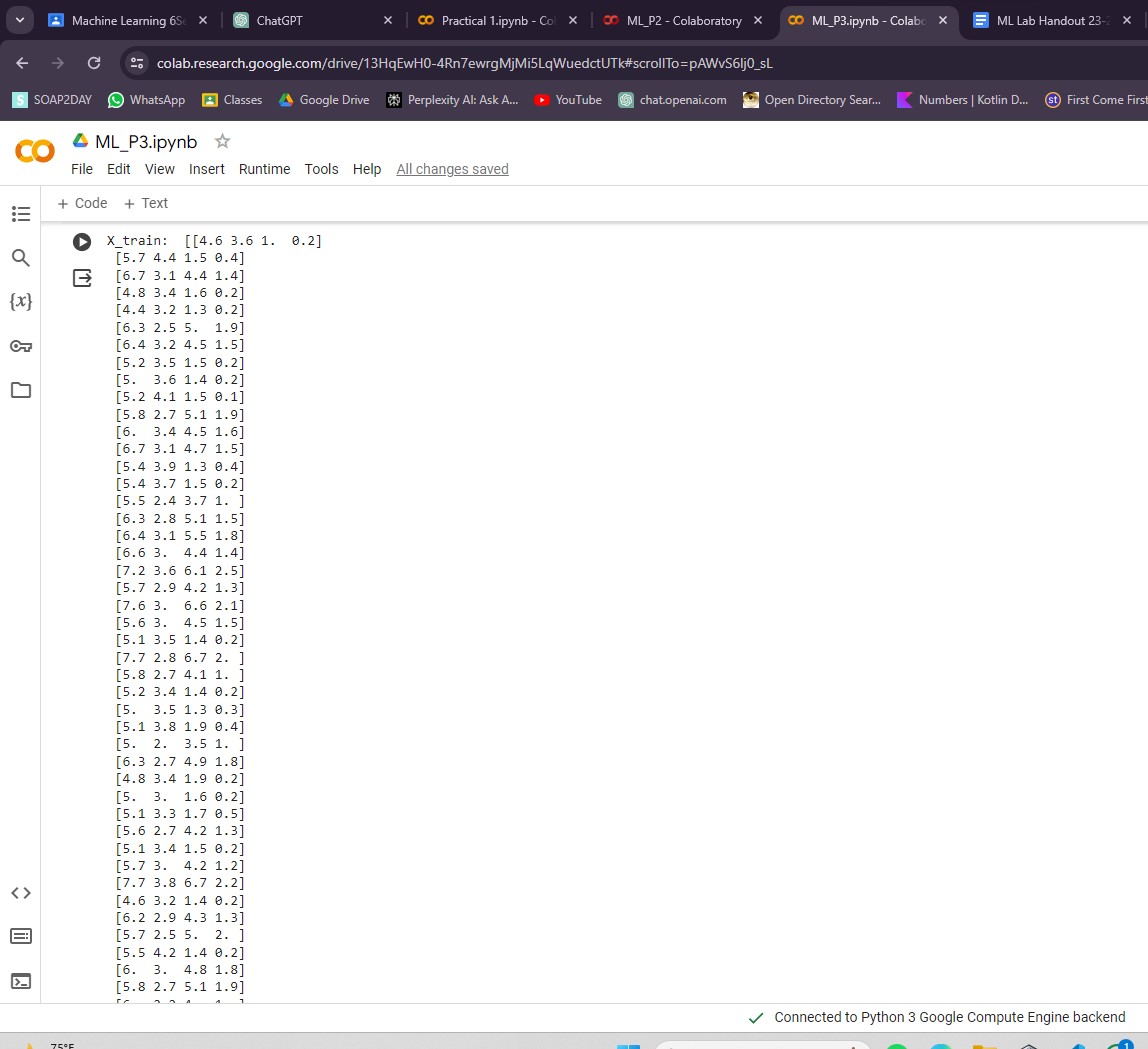
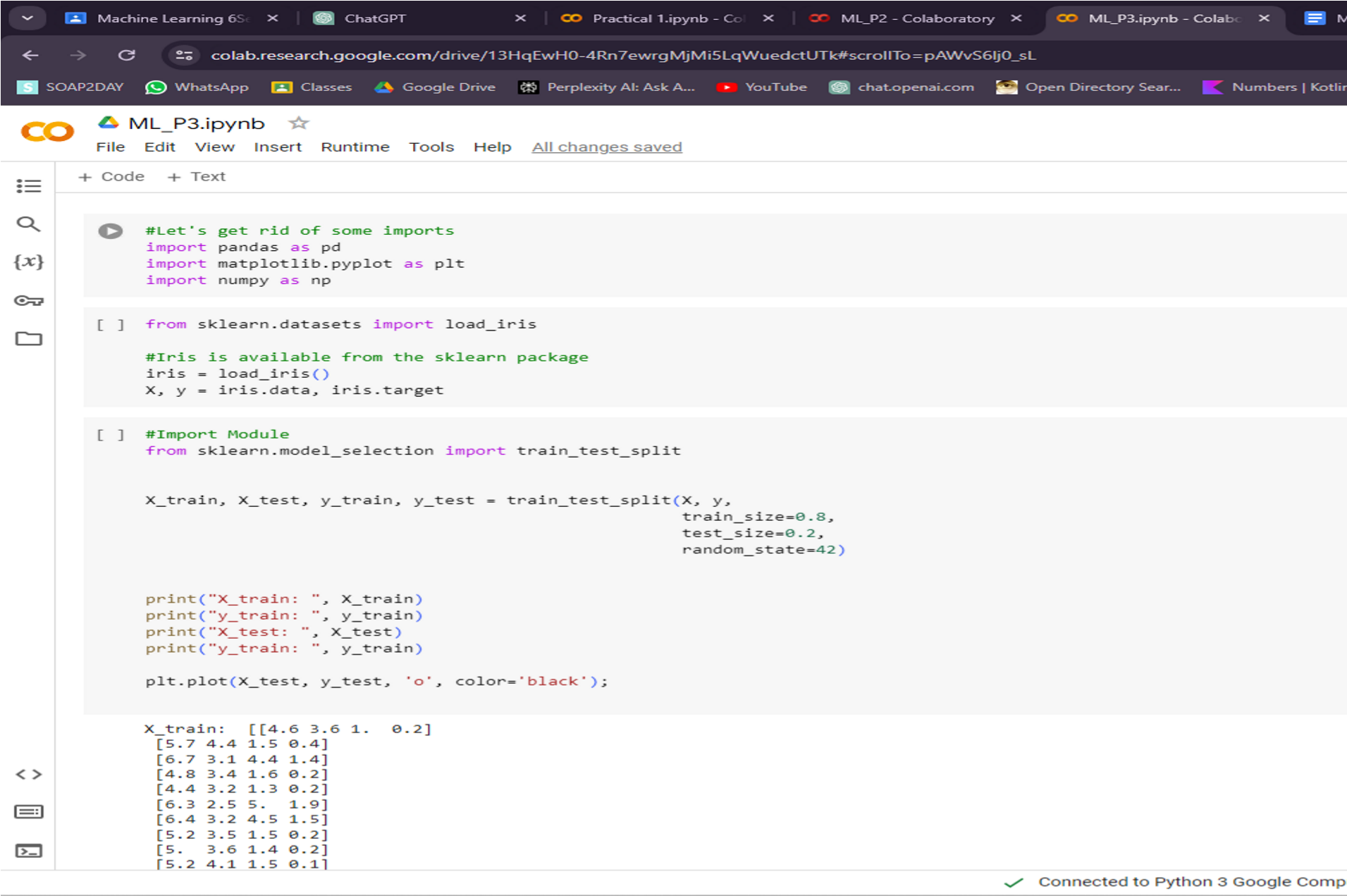




Practical 3

Aim : Classify the data into training , testing and validation.

|  |
| --- |
| #Let's get rid of some imports import pandas as pd import matplotlib.pyplot as plt import numpy as np from sklearn.datasets import load\_iris    #Iris is available from the sklearn package iris iris = load\_iris()  X, y = iris.data, iris.target  #Import Module  from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.8, test\_size=0.2, random\_state=42)    print("X\_train: ", X\_train) print("y\_train: ", y\_train) print("X\_test: ", X\_test) print("y\_train: ", y\_train)  plt.plot(X\_test, y\_test, 'o', color='black'); plt.plot(X\_test, y\_test,'o', color='black'); |



Practical 4

Aim : Program based on linear regression .

Code

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.model\_selection import train\_test\_split from pandas.core.common import random\_state  from sklearn.linear\_model import LinearRegression  df\_sal = pd.read\_csv('/content/Salary\_Data.csv') df\_sal.head()      df\_sal.describe()    plt.scatter(df\_sal['YearsExperience'], df\_sal['Salary'], color =  'lightcoral')  plt.title('Salary vs Experience') plt.xlabel('Years of Experience') plt.ylabel('Salary') plt.box(False)  plt.show()    X = df\_sal.iloc[:, :1] |

|  |
| --- |
| y = df\_sal.iloc[:, 1:]    X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2 random\_state = 0)  regressor = LinearRegression() regressor.fit(X\_train, y\_train)    X\_pred\_test = regressor.predict(X\_test)  y\_pred\_train = regressor.predict(X\_train)  plt.scatter(X\_train, y\_train, color = 'lightcoral')  plt.plot(X\_train, y\_pred\_train, color = 'firebrick')  plt.title('Salary vs Experience (Training Set)')  plt.xlabel('Years of Experience')  plt.ylabel('Salary')  plt.legend(['X\_train/Pred(y\_test)', 'X\_train/y\_train'],  title = 'Sal/Exp' loc='best', facecolor='white')  plt.box(False) plt.show()  print(f'Coefficient: {regressor.coef\_}') print(f'Intercept: {regressor.intercept\_}') |

,

,

