import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score,mean\_squared\_error,mean\_absolute\_error

dataset = pd.read\_csv('train.csv')

dataset.head()

dataset.shape

dataset.isnull()

dataset.dropna(axis=0,inplace=True)

dataset.fillna(dataset.mean(),axis=0,inplace=True)

dataset.head(300)

dataset.at[214,'x']

x = dataset.drop('y',axis=1)

y = dataset.iloc[:,1]

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.25,random\_state=42)

regressor = LinearRegression()

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

x\_pred = regressor.predict(x\_train)

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

# graph for training set

plt.scatter(x\_train,y\_train,c='b')

plt.plot(x\_train,x\_pred,c='r',linewidth=5)

# graph for testing set

plt.scatter(x\_test,y\_test,c='b')

plt.plot(x\_train,x\_pred,c='r',linewidth=2)

# Provide your own input data for testing

x\_custom = np.array([[75]]) # Replace with your own input features

y\_custom\_pred = regressor.predict(x\_custom)

print(f"Predicted value for x = 9 is {y\_custom\_pred[0]}")

# Calculate evaluation metrics

mse\_train = mean\_squared\_error(y\_train, x\_pred)

mse\_test = mean\_squared\_error(y\_test, y\_pred)

mae\_test = mean\_absolute\_error(y\_test, y\_pred)

r2\_test = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error (Train): {mse\_train:.3f}")

print(f"Mean Squared Error (Test): {mse\_test:.3f}")

print(f"Mean Absolute Error (Test): {mae\_test:.3f}")

print(f"R² Score (Test): {r2\_test:.3f}")