# -\*- coding: utf-8 -\*-

"""

Created on Sun Jan 19 19:05:16 2020

@author: Dell 5530

"""

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

dataset = pd.read\_csv('housing price.csv')

X = dataset.iloc[:, :1]

Y = dataset.iloc[:, 1:2]

# Splitting the dataset into the Training set and Test set

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

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 1/3, random\_state = 0)

# Fitting Simple Linear Regression to the Training set

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, Y\_train)

# Predicting the Test set results

Y\_pred = regressor.predict(X\_test)

# Visualising the Training set results

plt.scatter(X\_train, Y\_train, color = 'pink')

plt.plot(X\_train, regressor.predict(X\_train), color = 'YELLOW')

plt.title(' (Training set)')

plt.xlabel('ID ')

plt.ylabel('SALE PRICE')

plt.show()

# Visualising the Test set results

plt.scatter(X\_test, Y\_test, color = 'BLACK')

plt.plot(X\_train, regressor.predict(X\_train), color = 'blue')

plt.title('(Test set)')

plt.xlabel('ID ')

plt.ylabel('SALE PRICE')

plt.show()

print('The sale price of ID 2950 is:')

print(regressor.predict([[3000]]))

print('The sale price of ID 3500 is:')

print(regressor.predict([[3900]]))

from sklearn import metrics

print('Mean Absolute Error:', metrics.mean\_absolute\_error(Y\_test, Y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(Y\_test, Y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(Y\_test, Y\_pred)))