**House price plot and trained model**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.metrics import r2\_score

from sklearn.linear\_model import LinearRegression

# Importing required libraries

data\_set = df = pd.read\_csv('housing.csv') # Reading the data set

data\_frame = pd.DataFrame(data\_set) # Making a data frame

data\_frame = data\_frame.drop(["longitude", "latitude","households"],axis=1) # Dropping the columns not required

data\_frame = data\_frame.dropna() # Dropping the values that contains nan values

data\_frame = pd.concat([data\_frame, pd.get\_dummies(data\_frame['ocean\_proximity'])],axis = 1) # Concatenating the columns

data\_frame = data\_frame.drop('ocean\_proximity',axis = 1)

model = LinearRegression() # Making variable to store the linear regression

X = data\_frame[['housing\_median\_age', 'total\_rooms', 'total\_bedrooms', 'population',

       'median\_income', '<1H OCEAN', 'INLAND', 'ISLAND',

       'NEAR BAY', 'NEAR OCEAN', '<1H OCEAN', 'INLAND', 'ISLAND', 'NEAR BAY',

       'NEAR OCEAN']] # Defining the dependent variables in a variable X

Y = data\_frame[['median\_house\_value']] # Defining the independent variable in Y

model.fit(X,Y) # Getting the relation between dependent and independent variables

predict1 = model.predict(X) # Building the model

plt.plot(data\_frame['median\_house\_value'], label = "Actual") # Making a dot/line of blue color for actual data

plt.plot(predict1, label = "Predicted") # Making a dot/line of orange color for predicted data

plt.title("Actual vs predicted") # Giving the title of the graph

plt.legend() # Shows/makes the legend

plt.xlabel("Input data") # Giving label to X axis as Input data

plt.ylabel("House value") # Giving label to Y axis as House value

plt.show() # Showing the graph

print(r2\_score(data\_frame['median\_house\_value'],predict1)) # This will show the accuracy of the model