```
import pandas
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from matplotlib import pylab
from pylab import *

# Read data and excel source and store it in Dataframe.
df = pandas.read_excel("/content/RealEstate.xlsx")

#Display the Dataframe
display(df)
```

	No	TransactionDate	HouseAge	DistanceFromAirport	NumberOfConvenienceStores	Latitude	Longitude	HousePriceInThousands
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	37.9
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	42.2
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	47.3
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	54.8
4	5	2012.833	5.0	390.56840	5	24.97937	121.54245	43.1
...	...	...	...	...	...	...	...	...
409	410	2013.000	13.7	4082.01500	0	24.94155	121.50381	15.4
410	411	2012.667	5.6	90.45606	9	24.97433	121.54310	50.0
411	412	2013.250	18.8	390.96960	7	24.97923	121.53986	40.6
412	413	2013.000	8.1	104.81010	5	24.96674	121.54067	52.5
413	414	2013.500	6.5	90.45606	9	24.97433	121.54310	63.9

414 rows × 8 columns

```
[27] # Remove rows and columns with Null/NaN values.
     df=df.dropna(how='any')

     # Since we want to predict the HousePrice based on HouseAge, DistanceFromAirport and NumberOfConvenienceStores, Remove all other columns from the Dataset.
     df= df.drop(columns=['No','TransactionDate','Latitude','Longitude'])


     # Select the column to predict.
     y=df.pop('HousePriceInThousands')

     # Select the columns based on which prediction will be done.
     x =df.values

     #Display the y value
     display(y)
```

```
0      37.9
1      42.2
2      47.3
3      54.8
4      43.1
       ...
409    15.4
410    50.0
411    40.6
412    52.5
413    63.9
Name: HousePriceInThousands, Length: 414, dtype: float64
```

```
[28] #Let's divide the dataset into two parts, one for training and other for testing.
     x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.33, test_size=0.67, random_state=100)

     #Selecting the Algorithm for the model: Linear Regression
     HousePriceModel = linear_model.LinearRegression(normalize=True)

     #Fitting the Model
     HousePriceModel.fit(x_train, y_train)

     #Predicting the House Price
     HousePrice_predicted= HousePriceModel.predict(x_test)

     #Display the Predicted value
     display(HousePrice_predicted)
```

```
array([45.01032238, 36.88889727, 55.28439056, 39.19484954, 52.28076885,
       32.9896596 , 49.36387845, 52.28076885, 52.14144462, 33.43628311,
       34.54622145, 45.10427792, 41.27857979, 38.47830482, 50.17631825,
       38.275791  , 54.51656601, 43.05257175, 33.97904698, 28.44343264,
       12.81333582, 43.59277269, 38.2670058 , 52.58125208, 31.08806928,
       31.96087097, 25.54780597, 41.2316683 , 41.17832718, 40.53009298,
       56.20083367, 37.81927189, 43.61348375, 45.20722049, 29.01501327,
       48.80922479, 37.4581225 , 18.14243029, 48.77727958, 55.06283112,
       45.20178085, 44.30347778, 35.53220029, 35.22990878, 40.70253611,
       45.1729063 , 40.85413609, 31.72819064, 41.20398208, 49.13700839,
       41.32082845, 34.70196245, 40.5687417 , 40.4068103 , 52.14144462,
       48.32024021, 48.77424721, 35.33284335, 40.66569508, 32.88467142,
       39.52107708, 48.91543861, 50.03905348, 36.42348211, 40.36285656,
       47.953739  , 50.24493861, 45.09358431, 38.10072291, 51.37190255,
        7.56351689, 34.8018245 , 45.10427792, 48.91150399, 48.84287501,
       33.92097696, 49.47067004, 35.63571993, 41.04107042, 26.65805061,
       40.75513559, 56.18867925, 49.00936403, 45.21628753, 44.73160981,
       45.82518592, 43.42199219, 47.92905967, 48.89275128, 40.69652612,
       48.40447051, 20.09839338, 41.42199219, 55.35301894, 46.29370803,
       44.6231144 , 44.81910077, 47.23006063, 46.01060422,  7.05900658,
       42.11955552, 44.12821315, 48.93643861, 30.0791896 , 18.04095278,
       32.46428699, 45.09646222, 48.0138075 , 39.91056121, 36.53947224,
       44.05913196, 36.87752565, 45.11326496, 44.76067239, 36.00366308,
       54.45698245, 34.51746893, 32.68856312, 36.67235112, 34.13617902,
       33.32619415, 47.91942481, 19.17185682, 43.98484524, 39.58500432,
       41.25054676, 40.93071487, 43.66767002, 34.52657763, 33.72801649,
       48.35255228, 55.25007637, 45.92511327, 34.72721995, 34.6638544 ,
       55.45596152, 49.81444657, 40.59872136, 49.17063795, 45.37879144,
       19.11672742, 49.00500699, 45.1729063 , 37.25377187, 41.9212661 ,
       32.47312236, 36.13691474, 54.25358887, 33.11526821, 42.68715286,
       41.11874341, 44.82633388, 37.89329799, 35.8838216 , 52.87748728,
```

```
[29] #Choosing the Graph type
     pylab.scatter(HousePrice_predicted,y_test)

     #Applying labels
     pylab.xlabel('Predicted')
     pylab.ylabel('Actual')
```

```
Text(0, 0.5, 'Actual')
```