In [18]:
```
import numpy as np
import pandas as pd

# Data Visualisation
import matplotlib.pyplot as plt
import seaborn as sns
```

In [19]:
```
housing = pd.DataFrame(pd.read_csv("Housing.csv"))
housing.head()
```

Out[19]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating
0	13300000	7420	4	2	3	yes	no	no	n
1	12250000	8960	4	4	4	yes	no	no	n
2	12250000	9960	3	2	2	yes	no	yes	n
3	12215000	7500	4	2	2	yes	no	yes	n
4	11410000	7420	4	1	2	yes	yes	yes	n

In [20]:
```
m = len(housing)
m
```

Out[20]:
```
545
```

In [21]:
```
housing.shape
```

Out[21]:
```
(545, 13)
```

In [118…
```
# You can see that your dataset has many columns with values as 'Yes' or 'No'.
# But in order to fit a regression line, we would need numerical values and not string
# List of variables to map

varlist =  ['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning',

# Defining the map function
def binary_map(x):
    return x.map({'yes': 1, "no": 0})

# Applying the function to the housing list
housing[varlist] = housing[varlist].apply(binary_map)

# Check the housing dataframe now
housing.head()
```

Out[118]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheatin
0	13300000	7420	4	2	3	NaN	NaN	NaN	Nal
1	12250000	8960	4	4	4	NaN	NaN	NaN	Nal
2	12250000	9960	3	2	2	NaN	NaN	NaN	Nal
3	12215000	7500	4	2	2	NaN	NaN	NaN	Nal
4	11410000	7420	4	1	2	NaN	NaN	NaN	Nal

In [119…

```
#Splitting the Data into Training and Testing Sets
from sklearn.model_selection import train_test_split

# We specify this so that the train and test data set always have the same rows, respe
np.random.seed(0)
df_train, df_test = train_test_split(housing, train_size = 0.7, test_size = 0.3)
df_train.shape
```

Out[119]:

```
(381, 13)
```

In [120…

```
df_test.shape
```

Out[120]:

```
(164, 13)
```

In [121…

```
num_vars = ['area', 'bedrooms', 'bathrooms', 'stories', 'parking', 'price']
df_Newtrain = df_train[num_vars]
df_Newtest = df_test[num_vars]
df_Newtrain.head()
```

Out[121]:

	area	bedrooms	bathrooms	stories	parking	price
454	4500	3	1	2	0	3143000
392	3990	3	1	2	0	3500000
231	4320	3	1	1	0	4690000
271	1905	5	1	2	0	4340000
250	3510	3	1	3	0	4515000

In [122…

```
df_Newtrain.shape
```

Out[122]:

```
(381, 6)
```

In [123…

```
# Here we can see that except for area, all the columns have small integer values.
#So it is extremely important to rescale the variables so that they have a comparable
#If we don't have comparable scales, then some of the coefficients as obtained by fitt
#This might become very annoying at the time of model evaluation.
##So it is advised to use standardization or normalization so that the units of the co

#As you know, there are two common ways of rescaling:
#1. Min-Max scaling
#2. Standardisation (mean-0, sigma-1)
```

```
#import warnings
#warnings.filterwarnings('ignore')

from sklearn.preprocessing import MinMaxScaler, StandardScaler
#scaler = StandardScaler()
scaler = MinMaxScaler()
df_Newtrain[num_vars] = scaler.fit_transform(df_Newtrain[num_vars])
df_Newtrain.head(5)
```

Out[123]:

	area	bedrooms	bathrooms	stories	parking	price
454	0.193548	0.5	0.0	0.333333	0.0	0.120606
392	0.156495	0.5	0.0	0.333333	0.0	0.151515
231	0.180471	0.5	0.0	0.000000	0.0	0.254545
271	0.005013	1.0	0.0	0.333333	0.0	0.224242
250	0.121622	0.5	0.0	0.666667	0.0	0.239394

In [124…
```
y_Normtrain = df_Newtrain.pop('price')
X_Normtrain = df_Newtrain.copy()
```

In [125…
```
X_Normtrain.head()
```

Out[125]:

	area	bedrooms	bathrooms	stories	parking
454	0.193548	0.5	0.0	0.333333	0.0
392	0.156495	0.5	0.0	0.333333	0.0
231	0.180471	0.5	0.0	0.000000	0.0
271	0.005013	1.0	0.0	0.333333	0.0
250	0.121622	0.5	0.0	0.666667	0.0

In [126…
```
y_Normtrain.head()
```

Out[126]:
```
454    0.120606
392    0.151515
231    0.254545
271    0.224242
250    0.239394
Name: price, dtype: float64
```

In [127…
```
Y = y_Normtrain.values  # get input values from first column
Y
```

Out[127]:
```
array([0.12060606, 0.15151515, 0.25454545, 0.22424242, 0.23939394,
       0.00148485, 0.11515152, 0.36363636, 0.32727273, 0.12121212,
       0.47272727, 0.09090909, 0.11212121, 0.09393939, 0.00606061,
       0.14848485, 0.21212121, 0.28484848, 0.48484848, 0.3030303 ,
       0.25454545, 0.27272727, 0.32424242, 0.        , 0.55151515,
       0.16666667, 0.61818182, 0.18787879, 0.24242424, 0.04545455,
       0.48484848, 0.42424242, 0.58121212, 0.3030303 , 0.4030303 ,
       0.30606061, 0.83636364, 0.07393939, 0.3       , 0.24242424,
       0.10909091, 0.78787879, 0.33787879, 0.18181818, 0.27272727,
       0.34848485, 0.1030303 , 0.36363636, 0.5969697 , 0.44242424,
       0.27272727, 0.60606061, 0.07818182, 0.16060606, 0.64848485,
       0.3030303 , 0.19393939, 0.09090909, 0.24181818, 0.26363636,
       0.0969697 , 0.45454545, 0.06060606, 0.24242424, 0.36363636,
       0.42121212, 0.13939394, 0.17515152, 0.13333333, 0.21212121,
       0.03030303, 0.04545455, 0.32121212, 0.03030303, 0.13636364,
       0.23575758, 0.22787879, 0.17515152, 0.15151515, 0.49090909,
       0.27878788, 0.28484848, 0.36969697, 0.10909091, 0.12727273,
       0.31515152, 0.38181818, 0.15151515, 0.45757576, 0.06666667,
       0.14848485, 0.29090909, 0.0569697 , 0.21818182, 0.06060606,
       0.12727273, 0.2030303 , 0.39393939, 0.16909091, 0.10909091,
       0.27575758, 0.33333333, 0.33333333, 0.53333333, 0.08484848,
       0.1969697 , 0.23939394, 0.21212121, 0.11515152, 0.90909091,
       0.26121212, 0.06363636, 0.33333333, 0.24242424, 0.2030303 ,
       0.36969697, 0.16666667, 0.3030303 , 0.3030303 , 0.07878788,
       0.18181818, 0.10484848, 0.34545455, 0.14545455, 0.23636364,
       0.18787879, 0.21212121, 0.11851515, 0.1030303 , 0.23636364,
       0.25151515, 0.35151515, 0.55757576, 0.40909091, 0.21212121,
       0.5030303 , 0.46060606, 0.27333333, 0.2       , 0.05454545,
       0.07878788, 0.18484848, 0.26666667, 0.21212121, 0.21212121,
       0.42727273, 0.21212121, 0.54813333, 0.18181818, 0.57575758,
       0.07878788, 0.15090909, 0.04848485, 0.20606061, 0.05454545,
       0.26666667, 0.57575758, 0.09393939, 0.18181818, 0.32666667,
       0.10909091, 0.14545455, 0.34545455, 0.27212121, 0.38484848,
       0.14545455, 0.14545455, 0.14242424, 0.26666667, 0.11818182,
       0.28484848, 0.26969697, 0.07878788, 0.27272727, 0.24848485,
       0.23575758, 0.44545455, 0.59090909, 0.26666667, 0.21212121,
       0.26060606, 0.3569697 , 0.37575758, 0.30909091, 0.07878788,
       0.10909091, 0.53030303, 0.22424242, 0.13939394, 0.42424242,
       0.12848485, 0.16363636, 0.19393939, 0.64848485, 0.21212121,
       0.19393939, 0.06363636, 0.23939394, 0.25454545, 0.14484848,
       0.32121212, 0.41818182, 0.36363636, 0.58787879, 0.37575758,
       0.33333333, 0.33030303, 0.63636364, 0.15151515, 0.18787879,
       0.37575758, 0.14242424, 0.34545455, 0.24848485, 0.23575758,
       0.17575758, 0.13939394, 0.2       , 0.1969697 , 0.1030303 ,
       0.24181818, 0.06666667, 0.06060606, 0.61515152, 0.19393939,
       0.21212121, 0.12121212, 0.22727273, 0.16969697, 0.25454545,
       0.11818182, 0.90606061, 0.14545455, 0.3630303 , 0.21818182,
       0.21515152, 0.42424242, 0.29090909, 0.16363636, 0.12121212,
       0.03030303, 0.22242424, 0.21212121, 0.06060606, 0.29393939,
       0.56666667, 0.42242424, 0.20909091, 0.23636364, 0.68666667,
       0.16969697, 0.13030303, 0.41212121, 0.22969697, 0.54484848,
       0.28424242, 0.47272727, 0.52727273, 0.22424242, 0.14545455,
       0.49090909, 0.27272727, 0.10848485, 0.00909091, 0.08484848,
       0.36363636, 0.26060606, 0.13939394, 0.46060606, 0.1030303 ,
       0.16363636, 0.35636364, 0.29090909, 0.5969697 , 0.12727273,
       0.51515152, 0.16363636, 0.15151515, 1.        , 0.15757576,
       0.27272727, 0.14848485, 0.16666667, 0.07272727, 0.39333333,
       0.15151515, 0.17575758, 0.21151515, 0.13939394, 0.21212121,
       0.30121212, 0.16666667, 0.23636364, 0.39393939, 0.27757576,
       0.39393939, 0.04242424, 0.31818182, 0.3030303 , 0.43636364,
```

```
0.21151515, 0.49393939, 0.18181818, 0.16363636, 0.15151515,
0.27272727, 0.21818182, 0.07272727, 0.25454545, 0.33333333,
0.33333333, 0.18787879, 0.3       , 0.17515152, 0.11575758,
0.35151515, 0.35757576, 0.4969697 , 0.12121212, 0.23333333,
0.19545455, 0.27272727, 0.41212121, 0.13878788, 0.36363636,
0.01212121, 0.06060606, 0.04545455, 0.15151515, 0.15757576,
0.40424242, 0.42121212, 0.13333333, 0.22969697, 0.36363636,
0.17272727, 0.21212121, 0.47575758, 0.13939394, 0.0969697 ,
0.1969697 , 0.23636364, 0.16363636, 0.57575758, 0.12727273,
0.51515152, 0.38181818, 0.15151515, 0.18787879, 0.22424242,
0.09545455, 0.48484848, 0.06060606, 0.3569697 , 0.57575758,
0.29393939, 0.11818182, 0.06666667, 0.33030303, 0.24242424,
0.22969697, 0.21212121, 0.13939394, 0.33333333, 0.30242424,
0.39090909, 0.12121212, 0.13333333, 0.36969697, 0.43030303,
0.3030303 , 0.40909091, 0.0969697 , 0.2030303 , 0.15151515,
0.10909091, 0.43636364, 0.22121212, 0.6969697 , 0.16969697,
0.28484848])
```

In [130…
```
X0 = df_Newtrain.values[:, 0]  # get input values from first column
```

In [138…
```
X0
```

Out[138]:
```
array([-0.73673364,  0.63289422, -0.95529128,  0.91459073,  1.37599019,
       -0.09563124, -0.55800206, -0.79501568,  0.84756639, -0.71244946,
       -1.02814382,  1.88595801,  0.40947975, -0.07620389,  3.08073976,
        2.12879983, -0.62016957,  0.9898717 ,  0.47747546, -1.02814382,
        1.20600092, -0.52303284,  0.35605455,  0.683891  ,  0.64260789,
       -0.74450458,  0.42890709, -0.55703069, -0.41618244, -0.65902426,
        0.91459073, -0.65902426,  0.16178109, -1.07185535,  0.99230011,
       -0.72216313, -0.66388109, -0.23162265, -0.80958618,  0.18363686,
        2.94231992, -1.17384891,  1.15743255, -0.52303284, -1.09808227,
        1.59454783,  0.37305348,  0.67174891,  0.72031728, -0.518176  ,
       -0.93829235,  0.77859931, -0.72216313, -0.785302  ,  0.42890709,
       -1.12528055, -0.54246018,  0.6037532 , -0.63959691,  0.95587384,
        1.11372102, -1.02814382,  0.42890709, -0.72216313, -0.54246018,
       -0.25105   , -0.785302  , -0.50409118, -0.56188753,  1.82524755,
       -0.54246018,  0.72031728,  1.3905607 , -0.50360549,  1.02678365,
       -0.73673364, -0.54246018, -1.21756044, -1.44340333, -1.20298993,
       -0.29961836,  0.18606527, -0.94072077,  0.91459073, -0.39675509,
       -0.63182597, -0.57645804, -0.71730629, -0.94072077,  0.88059288,
        0.6037532 ,  0.13749691, -1.44340333,  0.23463364, -0.72216313,
        0.42890709, -1.1981331 , -0.73673364,  3.92582929,  1.59454783,
       -1.01065921,  1.44884273,  0.42890709, -0.60074222, -0.34818673,
       -0.95529128, -0.52303284,  0.47747546, -1.27098564, -1.42883282,
       -1.03300066,  0.04036018,  0.18606527,  0.42890709,  0.42890709,
       -1.44340333,  0.42890709, -0.52011874, -1.44340333, -0.47932131,
       -0.4647508 , -0.11505858, -0.25105   , -1.44340333, -0.48417815,
        1.82767597,  0.34634088, -0.96986179, -0.16848378,  0.47747546,
        0.18606527, -0.71730629,  1.42455855, -1.02814382, -1.26127197,
       -0.80472935, -0.38704142, -0.25105   , -0.54246018, -0.11505858,
        1.52169528, -0.59102855, -0.94072077,  1.0700095 , -1.02814382,
       -0.85329771,  0.42890709, -0.63959691,  3.7801242 ,  1.40027437,
       -0.49632024,  1.52169528, -0.49632024,  0.77374248,  0.42890709,
       -1.15102178,  0.74460146, -0.80958618,  0.6037532 , -0.95529128,
        0.72031728, -0.29961836,  0.45319128, -0.40646876, -0.99463165,
       -0.88243873, -0.6784516 , -0.37732775, -0.52303284,  2.18708187,
       -1.04271433, -0.39675509, -0.34332989, -0.77558833, -0.49389182,
        0.66203524,  0.42890709, -1.32052537, -0.54731702, -0.80958618,
        0.38033873, -0.92615026,  0.33177037, -0.15391327, -0.77558833,
       -0.5745153 , -0.71730629, -0.94072077, -0.56188753, -0.69787895,
        0.45804811, -0.83387037, -0.37149954,  0.18606527,  0.42890709,
       -1.18161985, -0.15391327,  1.47312692, -0.29961836, -0.54246018,
        0.08892855,  2.61448347,  0.86602237, -0.785302  , -1.59347958,
       -0.52303284,  2.48820572, -1.02814382, -0.14662802, -0.98443229,
       -0.05677654, -1.00871648, -0.15391327, -1.02814382, -0.71973471,
       -0.89992334, -0.55703069, -0.22870855,  1.28856713,  1.78882128,
       -0.41618244,  0.6037532 , -0.98443229, -0.75130415,  1.30313764,
       -0.71730629,  0.13749691,  0.67174891,  1.08458001, -0.72701997,
        0.83688135, -1.42834714, -0.25105   , -0.20248164, -0.80472935,
       -0.82901353, -0.66630951,  0.72031728, -0.61531273, -0.29961836,
        1.88595801,  0.11806957,  0.42890709, -0.15391327,  0.40947975,
       -0.1441996 ,  2.61448347, -0.71730629, -0.28990469, -0.10534491,
       -0.05677654, -0.34818673,  0.58481154,  1.44884273,  1.25456928,
        0.72760253,  0.98841465,  0.69603309,  0.72031728, -1.05097095,
        1.68197088, -0.77558833, -1.08642586, -0.54246018, -1.24670146,
        3.92582929, -1.10099637, -1.13159444, -0.50360549,  1.96658149,
        0.63289422, -1.35598028,  3.05645558,  0.86602237,  0.55227074,
       -0.79501568,  0.64746473, -0.66388109, -0.29961836, -0.93829235,
        2.54648776,  0.45804811,  0.30748618, -0.77558833, -1.44340333,
       -0.07620389, -0.13448593, -0.54925975,  1.24776971, -0.23550812,
       -1.28312773, -0.66388109,  1.3905607 , -0.78044517, -0.54246018,
        0.09864222, -1.17384891,  0.18606527, -0.52303284,  0.35605455,
```

```
 0.63289422,  0.92430441, -0.05191971,  0.817454  , -0.49389182,
 0.08892855,  0.18606527,  0.02578967,  0.42890709, -0.37247091,
 0.42890709,  0.66203524, -0.78044517,  1.14869025,  0.91459073,
-0.29961836,  1.65282986,  0.43862077, -1.3049835 , -0.96986179,
-0.4647508 , -0.79501568, -1.02814382, -0.82415669, -0.37247091,
-0.54246018,  0.34536951, -0.518176  ,  1.44884273, -1.02814382,
-0.34818673,  1.11857786, -0.88729557, -0.61531273, -0.91157975,
-1.02814382, -0.85815455,  0.04036018, -0.64348238,  0.42890709,
-1.31955401,  1.64311619,  1.14529046,  0.16178109, -0.785302  ,
-0.04706287, -0.785302  , -0.48417815,  0.37062506,  0.72031728,
 0.64746473,  1.59454783, -0.29961836, -0.73673364, -0.29476153,
-0.36761407,  1.01172746,  0.6037532 ,  3.80149428, -0.73673364,
 2.3522143 ,  1.43912906, -0.06066201, -0.94072077,  1.12829153,
 0.42890709,  1.11857786,  0.7494583 , -0.16119853, -0.61531273,
 0.75431513,  1.01172746, -0.73673364,  0.38033873, -0.72216313,
 0.6037532 , -0.94072077,  0.016076  , -0.82415669,  0.42890709,
 1.25456928])
```

In [ ]: