In [1]:

```python
import numpy as np
import pandas as pd

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

In [3]:

```python
housing = pd.DataFrame(pd.read_csv("Housing.csv"))
housing.head()
```

Out[3]:

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

In [4]:

```python
m = len(housing)
m
```

Out[4]:

```
545
```

In [5]:

```python
housing.shape
```

Out[5]:

```
(545, 13)
```

In [110…

```python
housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   price             545 non-null    int64 
 1   area              545 non-null    int64 
 2   bedrooms          545 non-null    int64 
 3   bathrooms         545 non-null    int64 
 4   stories           545 non-null    int64 
 5   mainroad          545 non-null    object
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64 
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
```

In [6]:

```python
housing.describe()
```

Out[6]:

	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

In [19]:

```python
# 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[19]:

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

In [20]:

```python
#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, respec
np.random.seed(0)
df_train, df_test = train_test_split(housing, train_size = 0.7, test_size = 0.3, random
df_train.shape
```

Out[20]:
```
(381, 13)
```

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

Out[21]:
```
(164, 13)
```

In [22]:
```
num_vars = ['area', 'bedrooms', 'bathrooms', 'stories', 'parking','price']
df_Newtrain = df_train[num_vars]
df_Newtest = df_test[num_vars]
df_Newtrain.head()
```

Out[22]:

	area	bedrooms	bathrooms	stories	parking	price
359	3600	3	1	1	1	3710000
19	6420	3	2	2	1	8855000
159	3150	3	2	1	0	5460000
35	7000	3	2	4	2	8080940
28	7950	5	2	2	2	8400000

In [23]:
```
df_Newtrain.shape
```

Out[23]:
```
(381, 6)
```

In [28]:
```
# 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 s
#If we don't have comparable scales, then some of the coefficients as obtained by fitti
#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 coe

#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

# define standard scaler
#scaler = StandardScaler()
scaler = MinMaxScaler()
df_Newtrain[num_vars] = scaler.fit_transform(df_Newtrain[num_vars])
df_Newtrain.head(20)
```

Out[28]:

	area	bedrooms	bathrooms	stories	parking	price
359	0.155227	0.4	0.0	0.000000	0.333333	0.169697
19	0.403379	0.4	0.5	0.333333	0.333333	0.615152
159	0.115628	0.4	0.5	0.000000	0.000000	0.321212

	area	bedrooms	bathrooms	stories	parking	price
35	0.454417	0.4	0.5	1.000000	0.666667	0.548133
28	0.538015	0.8	0.5	0.333333	0.666667	0.575758
267	0.271383	0.4	0.0	0.333333	0.333333	0.229697
263	0.187610	0.4	0.0	0.333333	0.000000	0.230303
433	0.144667	0.6	0.0	0.333333	0.333333	0.133333
217	0.442274	0.4	0.0	0.333333	0.666667	0.266667
154	0.159627	0.4	0.5	0.333333	0.666667	0.327273
534	0.102429	0.6	0.0	0.333333	0.000000	0.030303
96	0.630412	0.4	0.0	0.000000	0.333333	0.393939
33	0.362900	0.4	1.0	0.333333	0.333333	0.557576
477	0.274903	0.2	0.0	0.000000	0.000000	0.103030
129	0.846885	0.4	0.0	0.666667	0.666667	0.356970
401	0.674410	0.4	0.0	0.333333	1.000000	0.151515
240	0.176346	0.4	0.0	0.333333	0.333333	0.245455
21	0.468057	0.4	0.5	0.000000	0.666667	0.600000
155	0.375220	0.4	0.5	0.000000	0.666667	0.327273
532	0.102429	0.2	0.0	0.000000	0.000000	0.033333

In [30]:

```
y_Newtrain = df_Newtrain.pop('price')
X_Newtrain = df_Newtrain
```

In [119…

```
X_Normtrain.head()
```

Out[119…

	area	bedrooms	bathrooms	stories	parking
359	0.155227	0.4	0.0	0.000000	0.333333
19	0.403379	0.4	0.5	0.333333	0.333333
159	0.115628	0.4	0.5	0.000000	0.000000
35	0.454417	0.4	0.5	1.000000	0.666667
28	0.538015	0.8	0.5	0.333333	0.666667

In [120…

```
y_Normtrain.head()
```

Out[120…

```
359    0.169697
19     0.615152
159    0.321212
35     0.548133
28     0.575758
Name: price, dtype: float64
```

In [137…

```
Y = y_Normtrain.values  # get input values from first column
Y
```

Out[137…

```
array([-5.75831482e-01,  2.25423943e+00,  3.86777672e-01,  1.82845815e+00,
        2.00396105e+00, -1.94638257e-01, -1.90787821e-01, -8.06857680e-01,
        4.02383765e-02,  4.25282038e-01, -1.46143190e+00,  8.48830066e-01,
        1.88844795e+00, -9.99379511e-01,  6.13953433e-01, -6.91344581e-01,
       -9.45269052e-02,  2.15797852e+00,  4.25282038e-01, -1.44217972e+00,
        1.55751475e-01, -8.06857680e-01, -9.99379511e-01,  6.56308235e-01,
       -2.11642634e-03, -5.75831482e-01, -3.67703559e-02, -1.36517099e+00,
        1.04135190e+00, -1.13779088e-01,  2.09861934e-02,  9.64343165e-01,
       -5.79681919e-01, -1.65395374e+00,  6.17803869e-01, -5.95083666e-01,
       -7.68353313e-01,  5.40795137e-01, -1.33031271e-01, -8.95417722e-01,
        1.23387373e+00, -9.99379511e-01,  3.35161387e+00, -7.87605496e-01,
       -5.37327116e-01, -5.37327116e-01, -1.52283454e-01,  6.17803869e-01,
        1.50340429e+00, -4.21814018e-01, -7.68353313e-01,  7.87427426e-02,
       -1.03788388e+00, -1.26891007e+00,  1.02980059e+00,  1.09546236e-01,
       -1.13414479e+00, -1.56133891e-01,  2.71264574e-01,  2.59713264e-01,
        9.45090982e-01, -1.24965789e+00,  1.54190866e+00, -1.15339698e+00,
       -9.45269052e-02,  2.90516757e-01, -8.83866412e-01, -3.06300919e-01,
       -4.21814018e-01,  2.00396105e+00, -6.33588032e-01,  7.33316968e-01,
        4.06029855e-01, -4.98822750e-01,  6.56308235e-01, -5.37327116e-01,
       -1.11489261e+00, -7.29848947e-01, -8.06857680e-01, -1.05713606e+00,
        7.71821334e-01, -4.60318384e-01, -6.14335849e-01,  5.75449066e-01,
        1.48415211e+00, -4.60318384e-01, -8.26109863e-01, -1.32666662e+00,
       -1.23040571e+00, -5.41177553e-01, -2.67796553e-01, -4.02561835e-01,
       -7.29848947e-01, -3.64057468e-01, -4.02561835e-01,  1.31088246e+00,
       -1.07638824e+00, -6.91344581e-01,  2.77404838e+00, -1.13779088e-01,
       -6.91344581e-01,  2.81255274e+00,  1.34938683e+00, -7.52747221e-02,
        4.44534221e-01, -5.18074933e-01, -5.75831482e-01,  3.86777672e-01,
       -8.06857680e-01, -2.67796553e-01, -1.59619719e+00,  5.79299503e-01,
        4.63786404e-01,  2.03861498e+00,  8.29577883e-01, -4.60318384e-01,
        1.73443049e+00, -3.25553102e-01, -7.83755060e-01, -6.14335849e-01,
       -7.52747221e-02, -1.94638257e-01, -9.60875144e-01, -3.06300919e-01,
       -7.68353313e-01,  1.19536936e+00, -5.37327116e-01, -5.79681919e-01,
        7.87427426e-02,  1.94255841e-01,  6.13953433e-01, -8.37661173e-01,
       -9.64725581e-01, -1.23040571e+00, -5.41177553e-01, -4.12187926e-01,
       -8.45362046e-01, -3.06300919e-01, -3.06300919e-01,  1.94620450e+00,
       -3.10151356e-01, -2.10040004e-01, -4.79570567e-01,  6.56308235e-01,
       -3.83309652e-01, -1.46143190e+00,  1.09910845e+00, -9.41622961e-01,
        7.87427426e-02,  6.94812602e-01, -8.83866412e-01,  9.64343165e-01,
        7.87427426e-02,  3.48273306e-01,  1.42639556e+00, -3.06300919e-01,
       -2.87048736e-01, -3.10151356e-01,  5.40795137e-01, -9.99379511e-01,
        3.86777672e-01, -4.21814018e-01,  2.67414137e-01, -6.14335849e-01,
       -8.83866412e-01, -1.05713606e+00, -7.68353313e-01, -3.83309652e-01,
       -1.52283454e-01,  7.29466531e-01,  7.87427426e-02,  7.29466531e-01,
       -3.06300919e-01, -1.13779088e-01, -1.17629525e-01,  1.61891739e+00,
        1.00284753e+00, -9.01000855e-01, -6.91344581e-01, -8.83866412e-01,
       -6.91344581e-01, -1.26891007e+00, -3.06300919e-01,  1.51901038e-01,
       -6.14335849e-01, -1.15339698e+00, -1.13779088e-01,  2.52012390e-01,
        9.45090982e-01, -3.64057468e-01, -3.48655722e-01, -2.63946116e-01,
       -9.80127327e-01,  8.10325700e-01,  2.70859095e+00, -8.45362046e-01,
        4.63786404e-01,  1.36499292e-01, -2.67796553e-01,  1.02209971e+00,
       -1.38827361e+00,  1.02209971e+00,  4.21431602e-01, -7.68353313e-01,
       -1.36517099e+00,  1.04135190e+00,  1.73401028e-03, -9.03118595e-01,
        4.02383765e-02, -9.60875144e-01,  4.63786404e-01, -1.03788388e+00,
        6.10102996e-01, -1.26891007e+00, -1.29201269e+00, -1.30741444e+00,
       -1.36517099e+00,  7.87427426e-02, -2.67796553e-01,  2.10022197e+00,
        2.71264574e-01,  4.63786404e-01, -9.22370778e-01, -6.14335849e-01,
        2.46601345e+00, -6.52840215e-01, -1.03788388e+00,  1.04135190e+00,
       -6.91344581e-01, -1.65395374e+00,  1.84994359e+00, -4.98822750e-01,
       -6.91344581e-01,  4.63786404e-01, -4.02561835e-01,  2.13508024e-01,
       -5.60225390e-02,  1.06060408e+00, -9.03118595e-01, -6.52840215e-01,
        1.36863901e+00, -1.13779088e-01,  1.54190866e+00,  9.06586616e-01,
       -6.14335849e-01, -5.37327116e-01,  4.63786404e-01, -6.91344581e-01,
        7.87427426e-02, -1.90787821e-01,  5.40795137e-01, -2.48544370e-01,
        1.73401028e-03,  2.09861934e-02,  1.17611718e+00,  7.33316968e-01,
       -4.98822750e-01,  2.50451781e+00,  2.38900471e+00, -3.06300919e-01,
        9.25838799e-01, -3.06300919e-01, -1.23040571e+00, -7.29848947e-01,
       -1.13779088e-01,  1.19536936e+00, -6.14335849e-01, -2.87048736e-01,
       -3.25553102e-01,  1.73401028e-03,  7.52569151e-01, -3.06300919e-01,
        1.94255841e-01,  6.94812602e-01,  6.56308235e-01, -5.56579299e-01,
        3.44422869e-01, -1.71535638e-01, -8.26109863e-01, -7.29848947e-01,
       -2.48544370e-01, -8.06857680e-01,  2.77404838e+00, -1.33031271e-01,
       -7.10596764e-01, -7.68353313e-01, -3.25553102e-01, -8.83866412e-01,
        1.17247109e-01, -3.44805285e-01, -1.30741444e+00, -9.60875144e-01,
        2.13508024e-01, -1.33031271e-01, -1.52283454e-01, -7.10596764e-01,
       -7.72203750e-01,  7.91073517e-01, -3.64057468e-01, -3.06300919e-01,
        1.04135190e+00,  6.56308235e-01, -7.52747221e-02,  6.94812602e-01,
        8.48830066e-01, -4.98822750e-01,  2.71264574e-01, -3.67703559e-02,
        8.48830066e-01,  5.94905596e-02,  1.76330876e+00, -4.21814018e-01,
       -1.56133891e-01,  1.80758878e+00, -3.06300919e-01,  2.52012390e-01,
        2.71264574e-01, -5.75831482e-01, -1.15339698e+00, -8.45362046e-01,
        4.02383765e-02, -6.91344581e-01,  7.87427426e-02,  7.48923060e-02,
       -1.05713606e+00,  1.11836063e+00,  5.79299503e-01, -5.75831482e-01,
       -9.18520342e-01,  1.94255841e-01, -1.19190134e+00,  4.69926668e+00,
       -4.60318384e-01, -1.65395374e+00, -9.60875144e-01, -9.87828201e-01,
       -5.06523623e-01, -1.04750997e+00, -8.06857680e-01,  1.15686500e+00,
       -1.64452017e+00,  2.31199598e+00,  2.00396105e+00,  7.71821334e-01,
        2.46601345e+00,  1.07985626e+00, -2.29292187e-01,  7.87427426e-02,
       -8.06857680e-01,  1.55751475e-01, -2.29292187e-01,  2.71264574e-01,
       -3.06300919e-01, -5.95083666e-01, -7.91251587e-02, -2.67796553e-01,
        2.52012390e-01,  1.25312591e+00, -6.91344581e-01, -9.99379511e-01,
        4.12170119e+00, -4.60318384e-01, -3.83309652e-01, -8.06857680e-01,
        1.46489993e+00,  5.60047320e-01,  3.65964880e+00,  4.92664679e-01,
       -7.49101130e-01, -6.72092398e-01,  9.64343165e-01,  8.48830066e-01,
       -1.19190134e+00, -3.83309652e-01, -3.67703559e-02, -2.67796553e-01,
       -1.34591881e+00,  1.42639556e+00, -5.37327116e-01,  1.04135190e+00,
       -1.26891007e+00])
```

In [135…

```
X0 = df_Normtrain.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 [ ]:
```