In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df=pd.read_csv(r"C:\Users\Hp\Downloads\archive (3)\Melbourne_housing_FULL.csv")
df.head(20)
```

Out[2]:

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Bathroom	Car	Landsize	BuildingArea	Yea
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016	2.5	3067.0	...	1.0	1.0	126.0	NaN	
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	...	1.0	1.0	202.0	NaN	
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	3067.0	...	1.0	0.0	156.0	79.0	1
3	Abbotsford	18/659 Victoria St	3	u	NaN	VB	Rounds	4/02/2016	2.5	3067.0	...	2.0	1.0	0.0	NaN	
4	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	3067.0	...	2.0	0.0	134.0	150.0	1
5	Abbotsford	40 Federation La	3	h	850000.0	PI	Biggin	4/03/2017	2.5	3067.0	...	2.0	1.0	94.0	NaN	
6	Abbotsford	55a Park St	4	h	1600000.0	VB	Nelson	4/06/2016	2.5	3067.0	...	1.0	2.0	120.0	142.0	2
7	Abbotsford	16 Maugie St	4	h	NaN	SN	Nelson	6/08/2016	2.5	3067.0	...	2.0	2.0	400.0	220.0	2
8	Abbotsford	53 Turner St	2	h	NaN	S	Biggin	6/08/2016	2.5	3067.0	...	1.0	2.0	201.0	NaN	1
9	Abbotsford	99 Turner St	2	h	NaN	S	Collins	6/08/2016	2.5	3067.0	...	2.0	1.0	202.0	NaN	1
10	Abbotsford	129 Charles St	2	h	941000.0	S	Jellis	7/05/2016	2.5	3067.0	...	1.0	0.0	181.0	NaN	
11	Abbotsford	124 Yarra St	3	h	1876000.0	S	Nelson	7/05/2016	2.5	3067.0	...	2.0	0.0	245.0	210.0	1
12	Abbotsford	121/56 Nicholson St	2	u	NaN	PI	Biggin	7/11/2016	2.5	3067.0	...	2.0	1.0	4292.0	82.0	2
13	Abbotsford	17 Raphael St	4	h	NaN	W	Biggin	7/11/2016	2.5	3067.0	...	2.0	0.0	230.0	147.0	1
14	Abbotsford	98 Charles St	2	h	1636000.0	S	Nelson	8/10/2016	2.5	3067.0	...	1.0	2.0	256.0	107.0	1
15	Abbotsford	217 Langridge St	3	h	1000000.0	S	Jellis	8/10/2016	2.5	3067.0	...	NaN	NaN	NaN	NaN	
16	Abbotsford	18a Mollison St	2	t	745000.0	S	Jellis	8/10/2016	2.5	3067.0	...	NaN	NaN	NaN	NaN	
17	Abbotsford	6/241 Nicholson St	1	u	300000.0	S	Biggin	8/10/2016	2.5	3067.0	...	1.0	1.0	0.0	NaN	
18	Abbotsford	10 Valiant St	2	h	1097000.0	S	Biggin	8/10/2016	2.5	3067.0	...	1.0	2.0	220.0	75.0	1
19	Abbotsford	403/609 Victoria St	2	u	542000.0	S	Dingle	8/10/2016	2.5	3067.0	...	NaN	NaN	NaN	NaN	

20 rows × 21 columns

In [3]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34857 entries, 0 to 34856
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Suburb         34857 non-null  object 
 1   Address        34857 non-null  object 
 2   Rooms          34857 non-null  int64  
 3   Type           34857 non-null  object 
 4   Price          27247 non-null  float64
 5   Method         34857 non-null  object 
 6   SellerG        34857 non-null  object 
 7   Date           34857 non-null  object 
 8   Distance       34856 non-null  float64
 9   Postcode       34856 non-null  float64
 10  Bedroom2       26640 non-null  float64
 11  Bathroom       26631 non-null  float64
 12  Car            26129 non-null  float64
 13  Landsize       23047 non-null  float64
 14  BuildingArea   13742 non-null  float64
 15  YearBuilt      15551 non-null  float64
 16  CouncilArea    34854 non-null  object 
 17  Lattitude      26881 non-null  float64
 18  Longtitude     26881 non-null  float64
 19  Regionname     34854 non-null  object 
 20  Propertycount  34854 non-null  float64
dtypes: float64(12), int64(1), object(8)
memory usage: 5.6+ MB
```

In [4]:

```
1 df.nunique()
```

Out[4]:

```
Suburb            351
Address         34009
Rooms              12
Type                3
Price            2871
Method              9
SellerG           388
Date               78
Distance          215
Postcode          211
Bedroom2           15
Bathroom           11
Car                15
Landsize         1684
BuildingArea      740
YearBuilt         160
CouncilArea        33
Lattitude       13402
Longtitude      14524
Regionname          8
Propertycount     342
dtype: int64
```

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
Suburb              0
Address             0
Rooms               0
Type                0
Price            7610
Method              0
SellerG             0
Date                0
Distance            1
Postcode            1
Bedroom2         8217
Bathroom         8226
Car              8728
Landsize        11810
BuildingArea    21115
YearBuilt       19306
CouncilArea         3
Lattitude        7976
Longtitude       7976
Regionname          3
Propertycount       3
dtype: int64
```

In [6]:

```
Columns_use=['Suburb','Rooms','Type','Price','Method','Distance','Regionname','Propertycount','Bedroom2','Bathroom','Car','Landsize',
    'BuildingArea','CouncilArea']
df=df[Columns_use]
```

In [7]:

```
df.head(20)
```

Out[7]:

	Suburb	Rooms	Type	Price	Method	Distance	Regionname	Propertycount	Bedroom2	Bathroom	Car	Landsize	BuildingArea	CouncilA
0	Abbotsford	2	h	NaN	SS	2.5	Northern Metropolitan	4019.0	2.0	1.0	1.0	126.0	NaN	Yarra Cou
1	Abbotsford	2	h	1480000.0	S	2.5	Northern Metropolitan	4019.0	2.0	1.0	1.0	202.0	NaN	Yarra Cou
2	Abbotsford	2	h	1035000.0	S	2.5	Northern Metropolitan	4019.0	2.0	1.0	0.0	156.0	79.0	Yarra Cou
3	Abbotsford	3	u	NaN	VB	2.5	Northern Metropolitan	4019.0	3.0	2.0	1.0	0.0	NaN	Yarra Cou
4	Abbotsford	3	h	1465000.0	SP	2.5	Northern Metropolitan	4019.0	3.0	2.0	0.0	134.0	150.0	Yarra Cou
5	Abbotsford	3	h	850000.0	PI	2.5	Northern Metropolitan	4019.0	3.0	2.0	1.0	94.0	NaN	Yarra Cou
6	Abbotsford	4	h	1600000.0	VB	2.5	Northern Metropolitan	4019.0	3.0	1.0	2.0	120.0	142.0	Yarra Cou
7	Abbotsford	4	h	NaN	SN	2.5	Northern Metropolitan	4019.0	3.0	2.0	2.0	400.0	220.0	Yarra Cou
8	Abbotsford	2	h	NaN	S	2.5	Northern Metropolitan	4019.0	4.0	1.0	2.0	201.0	NaN	Yarra Cou
9	Abbotsford	2	h	NaN	S	2.5	Northern Metropolitan	4019.0	3.0	2.0	1.0	202.0	NaN	Yarra Cou
10	Abbotsford	2	h	941000.0	S	2.5	Northern Metropolitan	4019.0	2.0	1.0	0.0	181.0	NaN	Yarra Cou
11	Abbotsford	3	h	1876000.0	S	2.5	Northern Metropolitan	4019.0	4.0	2.0	0.0	245.0	210.0	Yarra Cou
12	Abbotsford	2	u	NaN	PI	2.5	Northern Metropolitan	4019.0	2.0	2.0	1.0	4292.0	82.0	Yarra Cou
13	Abbotsford	4	h	NaN	W	2.5	Northern Metropolitan	4019.0	6.0	2.0	0.0	230.0	147.0	Yarra Cou
14	Abbotsford	2	h	1636000.0	S	2.5	Northern Metropolitan	4019.0	2.0	1.0	2.0	256.0	107.0	Yarra Cou
15	Abbotsford	3	h	1000000.0	S	2.5	Northern Metropolitan	4019.0	NaN	NaN	NaN	NaN	NaN	Yarra Cou
16	Abbotsford	2	t	745000.0	S	2.5	Northern Metropolitan	4019.0	NaN	NaN	NaN	NaN	NaN	Yarra Cou
17	Abbotsford	1	u	300000.0	S	2.5	Northern Metropolitan	4019.0	1.0	1.0	1.0	0.0	NaN	Yarra Cou
18	Abbotsford	2	h	1097000.0	S	2.5	Northern Metropolitan	4019.0	3.0	1.0	2.0	220.0	75.0	Yarra Cou
19	Abbotsford	2	u	542000.0	S	2.5	Northern Metropolitan	4019.0	NaN	NaN	NaN	NaN	NaN	Yarra Cou

In [8]:

```
#Columns that we fill to zero
Columns_fill_zero=['Propertycount', 'Distance', 'Bedroom2', 'Bathroom', 'Car']
df[Columns_fill_zero]=df[Columns_fill_zero].fillna(0)
```

In [9]:

```
#Columns that we fill with mean
df['Landsize'] = df['Landsize'].fillna(df.Landsize.mean())
df['BuildingArea'] = df['BuildingArea'].fillna(df.BuildingArea.mean())
```

In [10]:

```
df.dropna(inplace=True)
```

In [11]:

```
df.shape
```

Out[11]:

```
(27244, 14)
```

In [12]:

```
df.head()
```

Out[12]:

	Suburb	Rooms	Type	Price	Method	Distance	Regionname	Propertycount	Bedroom2	Bathroom	Car	Landsize	BuildingArea	CouncilAr
1	Abbotsford	2	h	1480000.0	S	2.5	Northern Metropolitan	4019.0	2.0	1.0	1.0	202.0	160.2564	Yarra C Coun
2	Abbotsford	2	h	1035000.0	S	2.5	Northern Metropolitan	4019.0	2.0	1.0	0.0	156.0	79.0000	Yarra C Coun
4	Abbotsford	3	h	1465000.0	SP	2.5	Northern Metropolitan	4019.0	3.0	2.0	0.0	134.0	150.0000	Yarra C Coun
5	Abbotsford	3	h	850000.0	PI	2.5	Northern Metropolitan	4019.0	3.0	2.0	1.0	94.0	160.2564	Yarra C Coun
6	Abbotsford	4	h	1600000.0	VB	2.5	Northern Metropolitan	4019.0	3.0	1.0	2.0	120.0	142.0000	Yarra C Coun

In [13]:

```
#One Hot Encoding
df = pd.get_dummies(df, drop_first=True)
```

In [14]:

```
df.head()
```

Out[14]:

mbik uncil	CouncilArea_Port Phillip City Council	CouncilArea_Stonnington City Council	CouncilArea_Whitehorse City Council	CouncilArea_Whittlesea City Council	CouncilArea_Wyndham City Council	CouncilArea_Yarra City Council	CouncilA Ran
0	0	0	0	0	0	1	
0	0	0	0	0	0	1	
0	0	0	0	0	0	1	
0	0	0	0	0	0	1	
0	0	0	0	0	0	1	

In [15]:

```
#Define X and Y variable
X=df.drop('Price',axis=1)
y=df['Price']
```

In [16]:

```
#Test and train Dataset
from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=0.3, random_state=2)
```

In [17]:

```
#Fit a regression model
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(train_X, train_y)
```

In [18]:

```
reg.score(test_X, test_y)
```

Out[18]:

```
-85257752396.6321
```

In [19]:

```
reg.score(train_X, train_y)
```

Out[19]:

```
0.6620209634941601
```

In [ ]:

```
#We can clearly see that training data have overfitted
```

In [20]:

```
#Fit Lasso Regression
from sklearn import linear_model
lasso_reg = linear_model.Lasso(alpha=50, max_iter=100, tol=0.1)
lasso_reg.fit(train_X, train_y)
```

Out[20]:

```
Lasso(alpha=50, max_iter=100, tol=0.1)
```

In [21]:

```
lasso_reg.score(test_X, test_y)
```

Out[21]:

```
0.6445487835443848
```

In [22]:

```
lasso_reg.score(train_X, train_y)
```

Out[22]:

```
0.6582724990758242
```

In [23]:

```
#Fit Ridge Resgression
from sklearn.linear_model import Ridge
ridge_reg= Ridge(alpha=50, max_iter=100, tol=0.1)
ridge_reg.fit(train_X, train_y)
```

Out[23]:

```
Ridge(alpha=50, max_iter=100, tol=0.1)
```

In [24]:

```
ridge_reg.score(test_X, test_y)
```

Out[24]:

```
0.6512606721036396
```

In [25]:

```
ridge_reg.score(train_X, train_y)
```

Out[25]:

```
0.6470932192456824
```

In [ ]:

```
#We see that Lasso and Ridge Regularizations prove to be beneficial when our Simple Linear Regression Model overfits
```