

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

```
from google.colab import files
uploaded = files.upload()
```

Choose Files

Melbourne...g_FULL.csv

• Melbourne_housing_FULL.csv

(text/csv) - 5018236 bytes, last modified: 9/20/2019 - 100% done

Saving Melbourne_housing_FULL.csv to Melbourne_housing_FULL.csv

```
df=pd.read_csv("Melbourne_housing_FULL.csv")
df
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016
3	Abbotsford	18/659 Victoria St	3	u	NaN	VB	Rounds	4/02/2016
4	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017
...
34852	Yarraville	13 Burns St	4	h	1480000.0	PI	Jas	24/02/2016
34853	Yarraville	29A Murray St	2	h	888000.0	SP	Sweeney	24/02/2016
34854	Yarraville	147A Severn St	2	t	705000.0	S	Jas	24/02/2016
34855	Yarraville	12/37 Stephen St	3	h	1140000.0	SP	hockingstuart	24/02/2016
34856	Yarraville	3 Tarrengower St	2	h	1020000.0	PI	RW	24/02/2016

34857 rows × 21 columns

◀

▶

```
df.head()
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016	
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	
3	Abbotsford	18/659 Victoria St	3	u	NaN	VB	Rounds	4/02/2016	
4	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	

5 rows × 21 columns

◀

▶

```
df.nunique()
```

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
Latitude   13402
Longitude  14524
Regionname   8
Propertycount 342
dtype: int64
```

```
df.shape

(34857, 21)
```

```
df.columns

Index(['Suburb', 'Address', 'Rooms', 'Type', 'Price', 'Method', 'SellerG',
      'Date', 'Distance', 'Postcode', 'Bedroom2', 'Bathroom', 'Car',
      'Landsize', 'BuildingArea', 'YearBuilt', 'CouncilArea', 'Latitude',
      'Longitude', 'Regionname', 'Propertycount'],
      dtype='object')
```

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

```
df1=df[cols]
```

```
df1
```

	Suburb	Rooms	Type	Method	SellerG	Regionname	Propertycount	Distance
0	Abbotsford	2	h	SS	Jellis	Northern Metropolitan	4019.0	
1	Abbotsford	2	h	S	Biggin	Northern Metropolitan	4019.0	
2	Abbotsford	2	h	S	Biggin	Northern Metropolitan	4019.0	
3	Abbotsford	3	u	VB	Rounds	Northern Metropolitan	4019.0	
4	Abbotsford	3	h	SP	Biggin	Northern Metropolitan	4019.0	
...	
34852	Yarraville	4	h	PI	Jas	Western Metropolitan	6543.0	
34853	Yarraville	2	h	SP	Sweeney	Western Metropolitan	6543.0	
34854	Yarraville	2	t	S	Jas	Western Metropolitan	6543.0	
34855	Yarraville	3	h	SP	hockingstuart	Western Metropolitan	6543.0	
34856	Yarraville	2	h	PI	RW	Western Metropolitan	6543.0	

34857 rows × 15 columns



```
#checking nan values
df1.isna().sum()
```

```
Suburb      0
Rooms       0
Type        0
Method      0
SellerG     0
Regionname   3
Propertycount 3
Distance     1
CouncilArea   3
```

```
Bedroom2      8217
Bathroom      8226
Car            8728
Landsize      11810
BuildingArea   21115
Price          7610
dtype: int64
```

```
filling_zero=['Propertycount','Distance','Bedroom2','Bathroom','Car']
```

```
#filling this columns with zero
df1[filling_zero]=df1[filling_zero].fillna(0)
```

```
df1.isna().sum()
```

```
Suburb      0
Rooms       0
Type        0
Method      0
SellerG     0
Regionname   3
Propertycount 0
Distance    0
CouncilArea  3
Bedroom2    0
Bathroom    0
Car         0
Landsize    11810
BuildingArea 21115
Price       7610
dtype: int64
```

```
df1['Landsize'] = df1['Landsize'].fillna(df1['Landsize'].mean())
df1['BuildingArea'] = df1['BuildingArea'].fillna(df1['BuildingArea'].mean())
```

```
df1.isna().sum()
```

```
Suburb      0
Rooms       0
Type        0
Method      0
SellerG     0
Regionname   3
Propertycount 0
Distance    0
CouncilArea  3
Bedroom2    0
Bathroom    0
Car         0
Landsize    0
BuildingArea 0
Price       7610
dtype: int64
```

```
df1.dropna(inplace=True)
df1.isna().sum()
```

```
Suburb      0
Rooms       0
Type        0
Method      0
SellerG     0
Regionname   0
Propertycount 0
Distance    0
CouncilArea  0
Bedroom2    0
Bathroom    0
Car         0
Landsize    0
BuildingArea 0
Price       0
dtype: int64
```

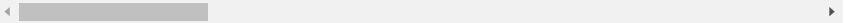
```
df1
```

	Suburb	Rooms	Type	Method	SellerG	Regionname	Propertycount	Dist:
1	Abbotsford	2	h	S	Biggin	Northern Metropolitan	4019.0	
2	Abbotsford	2	h	S	Biggin	Northern Metropolitan	4019.0	
4	Abbotsford	3	h	SP	Biggin	Northern Metropolitan	4019.0	
5	Abbotsford	3	h	PI	Biggin	Northern Metropolitan	4019.0	
6	Abbotsford	4	h	VB	Nelson	Northern Metropolitan	4019.0	
...
4852	Yarraville	4	h	PI	Jas	Western Metropolitan	6543.0	
4853	Yarraville	2	h	SP	Sweeney	Western Metropolitan	6543.0	
...

df1 =pd.get_dummies(df1,drop_first=True)
df1

	Rooms	Propertycount	Distance	Bedroom2	Bathroom	Car	Landsize	Build
1	2	4019.0	2.5	2.0	1.0	1.0	202.000000	
2	2	4019.0	2.5	2.0	1.0	0.0	156.000000	
4	3	4019.0	2.5	3.0	2.0	0.0	134.000000	
5	3	4019.0	2.5	3.0	2.0	1.0	94.000000	
6	4	4019.0	2.5	3.0	1.0	2.0	120.000000	
...
34852	4	6543.0	6.3	4.0	1.0	3.0	593.000000	
34853	2	6543.0	6.3	2.0	2.0	1.0	98.000000	
34854	2	6543.0	6.3	2.0	1.0	2.0	220.000000	
34855	3	6543.0	6.3	0.0	0.0	0.0	593.598993	
34856	2	6543.0	6.3	2.0	1.0	0.0	250.000000	

27244 rows × 745 columns

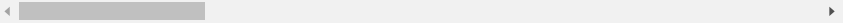


x=df1.drop('Price',axis='columns')
y=df1["Price"]

x

	Rooms	Propertycount	Distance	Bedroom2	Bathroom	Car	Landsize	Build
1	2	4019.0	2.5	2.0	1.0	1.0	202.000000	
2	2	4019.0	2.5	2.0	1.0	0.0	156.000000	
4	3	4019.0	2.5	3.0	2.0	0.0	134.000000	
5	3	4019.0	2.5	3.0	2.0	1.0	94.000000	
6	4	4019.0	2.5	3.0	1.0	2.0	120.000000	
...
34852	4	6543.0	6.3	4.0	1.0	3.0	593.000000	
34853	2	6543.0	6.3	2.0	2.0	1.0	98.000000	
34854	2	6543.0	6.3	2.0	1.0	2.0	220.000000	
34855	3	6543.0	6.3	0.0	0.0	0.0	593.598993	
34856	2	6543.0	6.3	2.0	1.0	0.0	250.000000	

27244 rows × 744 columns



```
y
1      1480000.0
2      1035000.0
4      1465000.0
5       850000.0
6      1600000.0
...
34852   1480000.0
34853   888000.0
34854   705000.0
34855   1140000.0
34856   1020000.0
Name: Price, Length: 27244, dtype: float64
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.3,random_state=0)
```

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(xtrain,ytrain)
```

```
LinearRegression()
```

```
lr.score(xtest,ytest)
```

```
0.46005593338256245
```

```
lr.score(xtrain,ytrain)
```

```
0.6889836348850227
```

```
from sklearn import linear_model
L1 = linear_model.Lasso(alpha=50,max_iter=100,tol=0.1)
L1.fit(xtrain,ytrain)
```

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

```
L1.score(xtest,ytest)
```

```
0.45553667151077504
```

```
L1.score(xtrain,ytrain)
```

```
0.6848817128447398
```

```
from sklearn.linear_model import Ridge
L2 = Ridge(alpha=50,max_iter=100,tol=0.1)
L2.fit(xtrain,ytrain)
```

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

```
L2.score(xtest,ytest)
```

```
0.427300726237817
```

```
L2.score(xtrain,ytrain)
```

```
0.6710057617210479
```

