Experiment 10: Implement Lasso and Ridge Regression by using Melbourne Dataset

In [9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [10]:
```
dataset = pd.read_csv('melb_data.csv')
dataset.head()
```

Out[10]:

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date
0	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016
1	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016
2	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017
3	Abbotsford	40 Federation La	3	h	850000.0	PI	Biggin	4/03/2017
4	Abbotsford	55a Park St	4	h	1600000.0	VB	Nelson	4/06/2016

5 rows × 21 columns

In [11]:
```
dataset.nunique() # Finding unique values in a particular column
```

Out[11]:
```
Suburb            314
Address         13378
Rooms               9
Type                3
Price            2204
Method              5
SellerG           268
Date               58
Distance          202
Postcode          198
Bedroom2           12
Bathroom            9
Car                11
Landsize         1448
BuildingArea      602
YearBuilt         144
CouncilArea        33
Lattitude        6503
Longtitude       7063
Regionname          8
Propertycount     311
dtype: int64
```

In [12]:
```
dataset.shape
```

Out[12]:
```
(13580, 21)
```

In [13]:
```
col_to_use = ['Suburb', 'Rooms', 'Type', 'Method', 'SellerG', 'Regionname',
```

In [14]:
```
dataset = dataset[col_to_use]
```

In [15]:
```
dataset.shape
```

Out[15]:
```
(13580, 15)
```

In [16]:
```
dataset.head()
```

Out[16]:

	Suburb	Rooms	Type	Method	SellerG	Regionname	Propertycount	Dis
0	Abbotsford	2	h	S	Biggin	Northern Metropolitan	4019.0	
1	Abbotsford	2	h	S	Biggin	Northern Metropolitan	4019.0	
2	Abbotsford	3	h	SP	Biggin	Northern Metropolitan	4019.0	
3	Abbotsford	3	h	PI	Biggin	Northern Metropolitan	4019.0	
4	Abbotsford	4	h	VB	Nelson	Northern Metropolitan	4019.0	

In [17]:
```
dataset.shape
```

Out[17]: (13580, 15)

In [18]:
```
dataset.isna().sum()
```

Out[18]:
```
Suburb             0
Rooms              0
Type               0
Method             0
SellerG            0
Regionname         0
Propertycount      0
Distance           0
CouncilArea     1369
Bedroom2           0
Bathroom           0
Car               62
Landsize           0
BuildingArea    6450
Price              0
dtype: int64
```

In [19]:
```
cols_to_fill_zero = ['Car']
dataset[cols_to_fill_zero] = dataset[cols_to_fill_zero].fillna(0)
dataset.isna().sum()
```

Out[19]:
```
Suburb             0
Rooms              0
Type               0
Method             0
SellerG            0
Regionname         0
Propertycount      0
Distance           0
CouncilArea     1369
Bedroom2           0
Bathroom           0
Car                0
Landsize           0
BuildingArea    6450
Price              0
dtype: int64
```

In [20]:
```
dataset['Landsize'] = dataset['Landsize'].fillna(dataset.Landsize.mean())
dataset['BuildingArea'] = dataset['BuildingArea'].fillna(dataset.Landsize.me
```

In [21]:
```
dataset.isna().sum()
```

Out[21]:
```
Suburb             0
Rooms              0
Type               0
Method             0
SellerG            0
Regionname         0
Propertycount      0
Distance           0
CouncilArea     1369
Bedroom2           0
Bathroom           0
Car                0
Landsize           0
BuildingArea       0
Price              0
dtype: int64
```

In [22]:
```
dataset.dropna(inplace=True)
dataset.isna().sum()
```

Out[22]:
```
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
```

In [23]:
```
dataset = pd.get_dummies(dataset, drop_first = True)
dataset.head()
```

Out[23]:

	Rooms	Propertycount	Distance	Bedroom2	Bathroom	Car	Landsize	Bui
0	2	4019.0	2.5	2.0	1.0	1.0	202.0	5
1	2	4019.0	2.5	2.0	1.0	0.0	156.0	
2	3	4019.0	2.5	3.0	2.0	0.0	134.0	1
3	3	4019.0	2.5	3.0	2.0	1.0	94.0	5
4	4	4019.0	2.5	3.0	1.0	2.0	120.0	1

5 rows × 613 columns

In [24]:
```
# from sklearn.preprocessing import StandardScaler
# from pandas import DataFrame
```

In [25]:
```
#std = StandardScaler()
#data = std.fit_transform(x)

#dataset = DataFrame(data)
#dataset.head()
```

In [26]:
```
x = dataset.drop('Price', axis = 1)
y = dataset['Price']
#x
```

In [46]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, rar
```

In [48]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

In [50]:
```
model.fit(x_train, y_train)
```

Out[50]:
```
LinearRegression()
```

In [52]:
```
model.score(x_test, y_test)
```

Out[52]:
```
-258533644277.60342
```

In [54]:
```
model.score(x_train, y_train)
```

Out[54]:
```
0.7101777405358054
```

In [56]:
```
from sklearn.linear_model import Lasso
lasso_model = Lasso()
```

In [58]:
```
lasso_model.fit(x_train, y_train)
```

```
C:\Users\Rishi\anaconda3\Lib\site-packages\sklearn\linear_model\_coordinate_
descent.py:678: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations, check the scale of the features or
consider increasing regularisation. Duality gap: 5.045e+14, tolerance: 3.475
e+11
  model = cd_fast.enet_coordinate_descent(
```

Out[58]:
```
Lasso()
```

In [60]:
```
lasso_model.score(x_test, y_test)
```

Out[60]: 0.6525752784636165

In [62]:
```
lasso_model.score(x_train, y_train)
```

Out[62]: 0.7101608461738054

In [64]:
```
from sklearn.linear_model import Ridge
ridge_model = Ridge()
```

In [70]:
```
ridge_model.fit(x_train, y_train)
```

Out[70]:

Ridge

Ridge()

In [72]:
```
ridge_model.score(x_test, y_test)
```

Out[72]: 0.6622828936615058

In [74]:
```
ridge_model.score(x_train, y_train)
```

Out[74]: 0.7081160032985205