

L1 - Lasso And L2 - Ridge method to solve over fitting

June 21, 2023

```
[33]: import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
[34]: df=pd.read_csv(r"C:\Users\Rakesh\Downloads\melb_data.csv")
df.head(3)
```

```
[34]:
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	\
0	Abbotsford	85 Turner St	2	h	1480000	S	Biggin	
1	Abbotsford	25 Bloomburg St	2	h	1035000	S	Biggin	
2	Abbotsford	5 Charles St	3	h	1465000	SP	Biggin	

	Date	Distance	Postcode	...	Bathroom	Car	Landsize	BuildingArea	\
0	03-12-2016	2.5	3067	...	1	1.0	202	NaN	
1	04-02-2016	2.5	3067	...	1	0.0	156	79.0	
2	04-03-2017	2.5	3067	...	2	0.0	134	150.0	

	YearBuilt	CouncilArea	Lattitude	Longtitude	Regionname	\
0	NaN	Yarra	-37.7996	144.9984	Northern Metropolitan	
1	1900.0	Yarra	-37.8079	144.9934	Northern Metropolitan	
2	1900.0	Yarra	-37.8093	144.9944	Northern Metropolitan	

	Propertycount
0	4019
1	4019
2	4019

[3 rows x 21 columns]

```
[35]: df.columns
```

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

```
[36]: col_useful=['Suburb','Rooms','Type','Method','SellerG','Distance','Bedroom2','Bathroom','Car']
      ↵
      ↪,'Landsize','CouncilArea','BuildingArea','Regionname','Propertycount',
      'Price']
```

```
[37]: df1=df[col_useful]
      df1.head(3)
```

```
[37]:      Suburb  Rooms Type Method SellerG  Distance  Bedroom2  Bathroom  Car  \
0  Abbotsford    2   h     S   Biggin    2.5         2         1  1.0
1  Abbotsford    2   h     S   Biggin    2.5         2         1  0.0
2  Abbotsford    3   h    SP   Biggin    2.5         3         2  0.0

      Landsize CouncilArea  BuildingArea  Regionname  Propertycount  \
0         202         Yarra          NaN  Northern Metropolitan    4019
1         156         Yarra        79.0  Northern Metropolitan    4019
2         134         Yarra       150.0  Northern Metropolitan    4019

      Price
0  1480000
1  1035000
2  1465000
```

```
[38]: df1.dtypes
```

```
[38]: Suburb      object
      Rooms      int64
      Type      object
      Method     object
      SellerG    object
      Distance   float64
      Bedroom2   int64
      Bathroom   int64
      Car        float64
      Landsize   int64
      CouncilArea object
      BuildingArea float64
      Regionname object
      Propertycount int64
      Price      int64
      dtype: object
```

```
[39]: df1.isnull().sum()
```

```
[39]: Suburb      0
      Rooms      0
      Type      0
```

```

Method          0
SellerG         0
Distance        0
Bedroom2        0
Bathroom        0
Car             62
Landsize        0
CouncilArea     1369
BuildingArea    6450
Regionname      0
Propertycount   0
Price           0
dtype: int64

```

```
[40]: df1.isna().sum()
```

```

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

```

```
[41]: na=['CouncilArea', 'BuildingArea']
```

```
[42]: df1[na]=df1[na].fillna(0)
```

```
[43]: df1.isna().sum()
```

```

[43]: Suburb          0
Rooms              0
Type              0
Method            0
SellerG           0
Distance          0
Bedroom2          0
Bathroom          0

```

```

Car                62
Landsize           0
CouncilArea        0
BuildingArea       0
Regionname         0
Propertycount      0
Price              0
dtype: int64

```

```
[44]: df1['Car']=df1['Car'].fillna(df1.Car.mean())
```

```
[45]: df1.isna().sum()
```

```

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

```

```
[46]: df1.dtypes
```

```

[46]: Suburb                object
Rooms                      int64
Type                       object
Method                     object
SellerG                    object
Distance                   float64
Bedroom2                   int64
Bathroom                   int64
Car                        float64
Landsize                   int64
CouncilArea                object
BuildingArea               float64
Regionname                 object
Propertycount              int64
Price                      int64

```

dtype: object

```
[47]: df1=pd.get_dummies(df1,drop_first=True)
df1()
```

```
[47]:
```

	Rooms	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	\
0	2	2.5	2	1	1.0	202	0.0	
1	2	2.5	2	1	0.0	156	79.0	
2	3	2.5	3	2	0.0	134	150.0	
3	3	2.5	3	2	1.0	94	0.0	
4	4	2.5	3	1	2.0	120	142.0	
...	
13575	4	16.7	4	2	2.0	652	0.0	
13576	3	6.8	3	2	2.0	333	133.0	
13577	3	6.8	3	2	4.0	436	0.0	
13578	4	6.8	4	1	5.0	866	157.0	
13579	4	6.3	4	1	1.0	362	112.0	

	Propertycount	Price	Suburb_Aberfeldie	...	CouncilArea_Wyndham	\
0	4019	1480000		0	...	0
1	4019	1035000		0	...	0
2	4019	1465000		0	...	0
3	4019	850000		0	...	0
4	4019	1600000		0	...	0
...	
13575	7392	1245000		0	...	0
13576	6380	1031000		0	...	0
13577	6380	1170000		0	...	0
13578	6380	2500000		0	...	0
13579	6543	1285000		0	...	0

	CouncilArea_Yarra	CouncilArea_Yarra	Ranges	\
0	1		0	
1	1		0	
2	1		0	
3	1		0	
4	1		0	
...	
13575	0		0	
13576	0		0	
13577	0		0	
13578	0		0	
13579	0		0	

	Regionname_Eastern Victoria	Regionname_Northern Metropolitan	\
0	0	1	
1	0	1	

2	0	1
3	0	1
4	0	1
...
13575	0	0
13576	0	0
13577	0	0
13578	0	0
13579	0	0

	Regionname_Northern Victoria	Regionname_South-Eastern Metropolitan \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...
13575	0	1
13576	0	0
13577	0	0
13578	0	0
13579	0	0

	Regionname_Southern Metropolitan	Regionname_Western Metropolitan \
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...
13575	0	0
13576	0	1
13577	0	1
13578	0	1
13579	0	1

	Regionname_Western Victoria
0	0
1	0
2	0
3	0
4	0
...	...
13575	0
13576	0
13577	0
13578	0

13579

0

[13580 rows x 635 columns]

```
[48]: df1.head(3)
```

```
[48]:   Rooms  Distance  Bedroom2  Bathroom  Car  Landsize  BuildingArea  \
0      2        2.5         2         1  1.0        202          0.0
1      2        2.5         2         1  0.0        156          79.0
2      3        2.5         3         2  0.0        134         150.0

      Propertycount  Price  Suburb_Aberfeldie  ...  CouncilArea_Wyndham  \
0             4019  1480000                0  ...                0
1             4019  1035000                0  ...                0
2             4019  1465000                0  ...                0

      CouncilArea_Yarra  CouncilArea_Yarra Ranges  Regionname_Eastern Victoria  \
0                  1                0                0
1                  1                0                0
2                  1                0                0

      Regionname_Northern Metropolitan  Regionname_Northern Victoria  \
0                  1                0
1                  1                0
2                  1                0

      Regionname_South-Eastern Metropolitan  Regionname_Southern Metropolitan  \
0                  0                0
1                  0                0
2                  0                0

      Regionname_Western Metropolitan  Regionname_Western Victoria
0                  0                0
1                  0                0
2                  0                0
```

[3 rows x 635 columns]

```
[ ]:
```

```
[49]: x=df1.drop('Price',axis=1)
      y=df1['Price']
```

```
[50]: from sklearn.model_selection import train_test_split
      X_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```
[51]: x_train.shape
```

```
[51]: (2716, 14)
```

```
[52]: x_test.shape
```

```
[52]: (2716, 634)
```

- 1 By comparing model score with train and test we get to know that weather the model is fit or over fit we can reduce it by L1 and L2 as shown below

```
[56]: from sklearn.linear_model import LinearRegression  
li=LinearRegression()
```

```
[59]: model=li.fit(X_train,y_train)
```

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

```
[60]: -35030922194.65845
```

```
[62]: model.score(X_train,y_train)
```

```
[62]: 0.7039666660524344
```

```
[63]: from sklearn.linear_model import Lasso  
lasso=Lasso(alpha=50,max_iter=100,tol=0.1)
```

```
[65]: model1=lasso.fit(X_train,y_train)
```

```
[66]: model1.score(x_test,y_test)
```

```
[66]: 0.6829756830823868
```

```
[67]: model1.score(X_train,y_train)
```

```
[67]: 0.6980680697791062
```

```
[68]: from sklearn.linear_model import Ridge  
ridge=Ridge(alpha=50,max_iter=100,tol=0.1)
```

```
[69]: model2=ridge.fit(X_train,y_train)
```

```
[70]: model2.score(x_test,y_test)
```

```
[70]: 0.6745301647170383
```

```
[71]: model2.score(X_train,y_train)
```


[71]: 0.6685215170207097

[]: