Question 1

What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer

1. The optimal value of alpha for ridge and lasso regression

   Ridge Alpha 2

   lasso Alpha 20

Ridge Regression

In [136]:

```
#Change the alpha value from 2 to 4
alpha = 4
ridge2 = Ridge(alpha=alpha)
ridge2.fit(X_train1, y_train)
```

Out[136]:

```
Ridge(alpha=4)
```

In [137]:

```
# Lets calculate some metrics such as R2 score, RSS and RMSE
y_pred_train = ridge2.predict(X_train1)
y_pred_test = ridge2.predict(X_test1)

metric2 = []
r2_train_lr = r2_score(y_train, y_pred_train)
print(r2_train_lr)
metric2.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print(r2_test_lr)
metric2.append(r2_test_lr)

rss1_lr = np.sum(np.square(y_train - y_pred_train))
print(rss1_lr)
metric2.append(rss1_lr)

rss2_lr = np.sum(np.square(y_test - y_pred_test))
print(rss2_lr)
metric2.append(rss2_lr)

mse_train_lr = mean_squared_error(y_train, y_pred_train)
print(mse_train_lr)
metric2.append(mse_train_lr**0.5)

mse_test_lr = mean_squared_error(y_test, y_pred_test)
print(mse_test_lr)
metric2.append(mse_test_lr**0.5)
```

```
0.8840201023484701
0.8763670118000035
586315302835.9323
```

```
307643551160.9661
656568088.282119
699189889.0021957
```

if you see slight decrease in r2 scroes for both train and test with metrics which we calcauted before

train is 0.8840201023484701 before 0.8878800

test is 0.8763670118000035 before 0.8776117

Lasso

In [143]:

```
#Changed alpha 20 to 40
alpha =40
lasso40 = Lasso(alpha=alpha)
lasso40.fit(X_train1, y_train)
```

Out[143]:

```
Lasso(alpha=40)
```

In [144]:

```
# Lets calculate some metrics such as R2 score, RSS and RMSE
y_pred_train = lasso40.predict(X_train1)
y_pred_test = lasso40.predict(X_test1)

metric3 = []
r2_train_lr = r2_score(y_train, y_pred_train)
print(r2_train_lr)
metric3.append(r2_train_lr)

r2_test_lr = r2_score(y_test, y_pred_test)
print(r2_test_lr)
metric3.append(r2_test_lr)

rss1_lr = np.sum(np.square(y_train - y_pred_train))
print(rss1_lr)
metric3.append(rss1_lr)

rss2_lr = np.sum(np.square(y_test - y_pred_test))
print(rss2_lr)
metric3.append(rss2_lr)

mse_train_lr = mean_squared_error(y_train, y_pred_train)
print(mse_train_lr)
metric3.append(mse_train_lr**0.5)

mse_test_lr = mean_squared_error(y_test, y_pred_test)
print(mse_test_lr)
metric3.append(mse_test_lr**0.5)

#R2score at alpha-20
#0.89067
```

```
#0.8757681
```

```
0.8885143427403547
0.8775918827343511
563595486990.7546
304595629651.96765
631125965.2752012
692262794.6635629
```

if you see slight decrease in r2 scroes for train and slight increase in test with metrics which we calcauted before also mse got increased alot

r2 in train is 0.8885143 , before 0.89067

r2 in train is 0.8775918, before 0.8757681

- LotArea---------------Lot size in square feet
- OverallQual---------Rates the overall material and finish of the house
- OverallCond--------Rates the overall condition of the house
- YearBuilt-------------Original construction date
- BsmtFinSF1--------Type 1 finished square feet
- TotalBsmtSF------- Total square feet of basement area
- GrLivArea-----------Above grade (ground) living area square feet
- TotRmsAbvGrd----Total rooms above grade (does not include bathrooms)
- Street_Pave--------Pave road access to property
- RoofMatl_Metal----Roof material_Metal

Predictors are same but the coefficient of these predictor has changed

Question 2

You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why? Question 2 You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Answer:

The r2_score of ridge is slightly higher than lasso for the test dataset so we will choose ridge regression to solve this problem

Question 3

After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

Ans :

- 11stFlrSF-----------First Floor square feet
- GrLivArea-----------Above grade (ground) living area square feet
- Street_Pave---------Pave road access to property
- RoofMatl_Metal------Roof material_Metal
- RoofStyle_Shed------Type of roof(Shed)

Steps we followed to arrive on above columns :

first we dropped the columns

Let's drop these columns

In [158…
```
X_train2 = X_train1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF'],axis=1)
X_test2 = X_test1.drop(['LotArea','OverallQual','YearBuilt','BsmtFinSF1','TotalBsmtSF'],axis=1)
```

Second we applied ridge with alpha 2:

Ridge

In [161…
```
#Ridge with alpha 2
alpha = 2
ridge21 = Ridge(alpha=alpha)
ridge21.fit(X_train2, y_train)
```

Out[161…
```
Ridge(alpha=2)
```

In [162…
```
# Lets calculate some metrics such as R2 score, RSS and RMSE
y_pred_train = ridge21.predict(X_train2)
y_pred_test = ridge21.predict(X_test2)
```

Calculated r2 score on train and test :

```
0.8152661464385286
0.8152390176285883
```

```
In [164]: #important predictor variables
betas = pd.DataFrame(index=X_train2.columns)
betas.rows = X_train1.columns
betas['ridge21'] = ridge21.coef_
pd.set_option('display.max_rows', None)
betas.head(68)
```

Out[164]:

	ridge21
OverallCond	4087.335143
1stFlrSF	154583.277465
2ndFlrSF	25450.373821
GrLivArea	156992.045399
BedroomAbvGr	-28522.779993
LandSlope_Sev	-14307.573645
Condition2_PosN	83.238580
RoofStyle_Shed	24386.459926
RoofMatl_Metal	23073.005761
RoofMatl_WdShake	-25855.110145

Question 4

How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?
Answer

Ans :

The model should be generalized so that the test accuracy is not lesser than the training score. The model should be accurate for datasets other than the ones which were used during training. Too much importance should not given to the outliers so that the accuracy predicted by the model is high. To ensure that this is not the case, the outliers analysis needs to be done and only those which are relevant to the dataset need to be retained. Those outliers which it does not make sense to keep must be removed from the dataset. If the model is not robust, It cannot be trusted for predictive analysis.