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?

Ans: The optimal value of alpha for Ridge was **"Ridge(alpha=0.8)"** and for Lasso was **"(Lasso(alpha=50)".**

Ridge ()

In [88]:
```
alpha = 0.8*2
ridge = Ridge(alpha=alpha)

ridge.fit(X_train, y_train)
```

Out[88]:
```
Ridge(alpha=1.6)
```

In [89]:
```
y_train_pred = ridge.predict(X_train)
print("Train score: ",r2_score(y_train, y_train_pred))

y_test_pred = ridge.predict(X_test)
print("Test score: ", r2_score(y_test, y_test_pred))
```

```
Train score:  0.9378987769626825
Test score:  0.8709857151847396
```

In [90]:
```
coefs = list(zip(X_train.columns,ridge.coef_))
coefs[:5]
```

Out[90]:
```
[('MSSubClass', -2711.163887527553),
 ('LotFrontage', -1663.3146730774606),
 ('LotArea', 1.8732077165817642),
 ('OverallQual', 8646.671086828834),
 ('OverallCond', 5145.23158600108)]
```

In [91]:
```
# lets sort the the features
ridge_coef = pd.DataFrame(sorted(coefs, key= lambda x: x[1],reverse=True), columns=["Features","Coef"])
ridge_coef.head(10)
```

Out[91]:

	Features	Coef
0	GrLivArea	61729.748408
1	TotalBsmtSF	21115.792101
2	Neighborhood_NoRidge	18341.367300
3	Neighborhood_NridgHt	17500.371210
4	Condition2_PosA	16996.960194
5	Street_Pave	15572.697741
6	SaleType_CWD	15293.439214

The most important predictor variables after doubling the alpha value are mentioned in the last dataframe.

Lasso()

In [93]:
```
alpha =50*2

lasso = Lasso(alpha=alpha)

lasso.fit(X_train, y_train)
```

Out[93]:
```
Lasso(alpha=100)
```

In [94]:
```
y_train_pred_lasso = lasso.predict(X_train)
print("Train score: ", r2_score(y_train, y_train_pred_lasso))

y_test_pred_lasso = lasso.predict(X_test)
print("Test score: ", r2_score(y_test, y_test_pred_lasso))
```

```
Train score:  0.9206450519267309
Test score:  0.8770750819348347
```

In [95]:
```
coefs = list(zip(X_train.columns,lasso.coef_))
coefs[:5]
```

Out[95]:
```
[('MSSubClass', -3262.0138205489707),
 ('LotFrontage', -1745.8611851227329),
 ('LotArea', 1.7557197089667866),
 ('OverallQual', 10312.727875089611),
 ('OverallCond', 6159.039122928093)]
```

In [96]:
```
# lets sort the the features
lasso_coef = pd.DataFrame(sorted(coefs, key= lambda x: x[1],reverse=True), columns=["Features","Coef"])
lasso_coef.head(10)
```

Out[96]:

	Features	Coef
0	GrLivArea	66219.755807
1	Neighborhood_NridgHt	18467.925583
2	Neighborhood_Somerst	16920.157223
3	Neighborhood_NoRidge	16502.030446
4	Exterior1st_BrkFace	13745.240347
5	Functional_Typ	12225.573936

The most important predictor variables after doubling the alpha value are mentioned in the last dataframe.

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?

Ans: We have derived the optimal value of lambda for both Ridge and Lasso regression. But considering the we can choose one of it considering few cases. In case of lasso it is used in feature elimination during the process which is very helpful but it consumes a lot of time performing that task. While in case of Ridge it is not like that. It does not concentrate on feature elimination rather just concentrates on hyperparameter tuning which is pretty fast compared to Lasso. And coming to the final result Lasso is slightly better than Ridge. Though the change is very minute. We better go for Ridge for fast output.

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?

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?

Ans: When considering the Robustness of the model we verify its performance on the data. That is how strong the model is while predicting the values for the given data correctly. It can be measured using R2 score in case of the Regression problems and Accuracy in case of Classification problems. The model can be considered as generalised if the model is simple enough to reduce the overfitting of the model and good enough to predict the correct output for the given data.

As I mentioned earlier Accuracy of the model is calculated for the classification problems. It can be widely relayed on Robustness and generalisability of the model because, the model should be strong enough to predict the output correctly in case of train as well as test data. That means it should be robust. And the model should not be too complex since the complex models may overfit, i.e. learning or we can say byhearting the data points in the training data, that may show very good accuracy during the training but when it comes to test data it will fail completely.