Problem Statement - Part II

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:

The optimal value of alpha for ridge and lasso regression:

For Ridge regression, alpha is 1.0

For Lasso regression, alpha is 0.0001

For Ridge regression alpha is 1.0 and now doubling it and making it 2.0.

For Ridge regression alpha is 1.0 and now doubling it and making it 2.0

In [125]:
```
# Model building using optimal alpha
ridge_modified = Ridge(alpha=2.0)
ridge_modified.fit(X_train, y_train)
```

Out[125]:
```
Ridge(alpha=2.0, copy_X=True, fit_intercept=True, max_iter=None,
      normalize=False, random_state=None, solver='auto', tol=0.001)
```

In [126]:
```
#creating coeffcients for the ridge regression
model_parameter = list(ridge.coef_)
model_parameter.insert(0,ridge.intercept_)
cols = house_train.columns
cols.insert(0,'const')
ridge_coef = pd.DataFrame(list(zip(cols,model_parameter,(abs(ele) for ele in model_parameter))))
ridge_coef.columns = ['Features','Coefficient','Mod']
#selecting the top 10 variables
ridge_coef.sort_values(by='Mod',ascending=False).head(10)
```

Out[126]:

	Features	Coefficient	Mod
0	LotFrontage	10.261566	10.261566
3	OverallCond	0.527282	0.527282
14	BsmtUnfSF	0.406426	0.406426
12	BsmtFinType2	0.355562	0.355562
2	OverallQual	0.338037	0.338037
11	BsmtFinSF1	0.322512	0.322512
9	BsmtExposure	0.316265	0.316265
33	GarageFinish	0.303286	0.303286
73	LotConfig_CulDSac	-0.272453	0.272453
6	ExterCond	0.264079	0.264079

In [127]:
```
y_train_pred = ridge_modified.predict(X_train)
y_test_pred = ridge_modified.predict(X_test)

print("Ridge Regression train r2:",r2_score(y_true=y_train,y_pred=y_train_pred))
print("Ridge Regression test r2:",r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
Ridge Regression train r2: 0.9238505663513401
Ridge Regression test r2: 0.7526833276310241
```

For Lasso regression alpha is 0.0001 and not doubling it and making it 0.0002.

For Lasso regression alpha is 0.0001 and now doubling it and making it 0.0002

In [128]:
```
#Model building using optimal alpha
lasso_modified = Lasso(alpha=0.0002)
lasso_modified.fit(X_train, y_train)
```

Out[128]:
```
Lasso(alpha=0.0002, copy_X=True, fit_intercept=True, max_iter=1000,
      normalize=False, positive=False, precompute=False, random_state=None,
      selection='cyclic', tol=0.0001, warm_start=False)
```

In [129]:
```
y_train_pred = lasso_modified.predict(X_train)
y_test_pred = lasso_modified.predict(X_test)

print("Lasso Regression train r2:",r2_score(y_true=y_train,y_pred=y_train_pred))
print("Lasso Regression test r2:",r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
Lasso Regression train r2: 0.924842196868562
Lasso Regression test r2: 0.7423837845239192
```

For all the models, the training score has decreased slightly and the testing score has increased slightly.

Most important predictor variables after the change is implemented are:

- LotFrontage
- BsmtFullBath
- OverallCond
- CentralAir
- OverallQual
- Exterior1st_CBlock
- MSZoning_RH
- MSZoning_RM
- Street_Pave
- GarageQual

In [131]:
```
#selecting the top 10 predictor variables
lasso_coef.sort_values(by='mod',ascending=False).head(10)
```

Out[131]:

	Feature	Coef	mod
0	LotFrontage	10.174821	10.174821
14	BsmtFullBath	0.822796	0.822796
3	OverallCond	0.568885	0.568885
9	CentralAir	0.492011	0.492011
2	OverallQual	0.484268	0.484268
73	Exterior1st_CBlock	-0.479688	0.479688
33	MSZoning_RH	0.379703	0.379703
35	MSZoning_RM	0.297522	0.297522
36	Street_Pave	0.246404	0.246404
20	GarageQual	0.240831	0.240831

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:

Even though Ridge has given good performance(R-Square value), I would choose the Lasso model for the following reasons:

- Lasso regression would help in feature elimination and the model will be more robust.
- Model is giving decent performance.
- Efficiently solved the high dimensionality problem by shrinking insignificant coefficients to zero.
- Simpler model and easy for maintenance.

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?

Answer:

The five most important predictor variables now are:

- LotArea
- FullBath
- 1stFlrSF
- ExterCond
- MSZoning_RH

```
#selecting the top 5 predictor variables
lasso_coef.sort_values(by='mod',ascending=False).head(5)
```

	Feature	Coef	mod
0	LotArea	10.205140	10.205140
10	FullBath	1.029837	1.029837
6	1stFlrSF	0.606857	0.606857
1	ExterCond	0.561906	0.561906
28	MSZoning_RH	0.512930	0.512930

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:

- A model demonstrates robustness when its performance remains largely unchanged in response to any variations in the data.
- A model with good generalizability can effectively adjust to new, previously unseen data that originates from the same distribution as the one used during model creation.
- To make sure a model is robust and generalizable, we have to take care it **doesn't overfit**. Overfitting occurs when a model becomes overly complex, leading to high variance and excessive sensitivity to minor fluctuations in the data. Consequently, an overfitted model may accurately capture patterns present in the training data but struggle to generalize to unseen test data.
- In other words, the model should not be too complex in order to be robust and generalizable.
- If we look at it from the perspective of Accuracy, a complex model will have a very high accuracy. So, to make our model more robust and generalizable, we will have to decrease variance which will lead to some bias. Addition of bias means that accuracy will decrease.
- Striking a balance between model accuracy and complexity is paramount. Regularization techniques such as Ridge Regression and Lasso offer effective means of achieving this balance. By imposing constraints on model parameters, regularization techniques help prevent overfitting, thereby promoting robustness and generalizability while tempering the potential decrease in accuracy associated with bias introduction.