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:

Optimal value of alpha

Ridge regression: 7.0

Lasso regression: 100

When the alpha is doubled, the r2 score decreases and the MSE/RMSE increases. Also the model prediction with respect to train and test data is comparable.

There is no change in the top 5 predictor variables. They are GrLivArea, OverallQual, Neighborhood_NoRidge, RoofMatl, GarageCars.

In [59]:
```
# Since Lasso is relatively the best model. Let us check the coeff for it
betas = pd.DataFrame(index=X_test.columns)
betas.rows = X_test.columns
betas['Coef'] = lasso.coef_
pd.set_option('display.max_rows', None)
betas.sort_values(by='Coef', ascending=False).head(200)
```

Out[59]:

	Coef
GrLivArea	180263.227637
OverallQual	91842.767008
Neighborhood_NoRidge	57016.847769
RoofMatl_WdShngl	56056.507574
GarageCars	44044.403703
Neighborhood_StoneBr	38082.387208
Neighborhood_NridgHt	32954.315038
2ndFlrSF	26006.898868

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:

For Ridge the optimal alpha is 7.0 and Lasso is 100. The Lasso regression is slightly more accurate than Ridge regression, hence will choose Lasso with alpha as 100.

Anything less than 100, causes a little overfitting and more than 120 causes a little underfitting. This is visible by noticing the r2 score of train and test data.

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:

If we remove the top 5 predictor variables, the below 5 are the top predictor variables.

TotalBsmtSF, 2ndFlrSF, LotArea, MasVnrArea, FullBath

```
alpha =120

lasso = Lasso(alpha=alpha)
X_train_2 = X_train.drop(["OverallQual","GrLivArea","Neighborhood_NoRidge","RoofMatl_WdShngl","GarageCars"], axis=1)
X_test_2 = X_test.drop(["OverallQual","GrLivArea","Neighborhood_NoRidge","RoofMatl_WdShngl","GarageCars"], axis=1)
lasso.fit(X_train_2, y_train)

# Lets calculate some metrics such as R2 score, RSS and RMSE for Lasso

y_pred_train = lasso.predict(X_train_2)
y_pred_test = lasso.predict(X_test_2)

betas = pd.DataFrame(index=X_test_2.columns)
betas.rows = X_test_2.columns
betas['Coef'] = lasso.coef_
pd.set_option('display.max_rows', None)
betas.sort_values(by='Coef', ascending=False).head(200)
```

t[87]:

	Coef
TotalBsmtSF	1.970156e+05
2ndFlrSF	1.395926e+05
LotArea	4.764078e+04
MasVnrArea	3.858734e+04
FullBath	3.795597e+04

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:

The model can be made robust and generalizable by choosing the optimal alpha/lambda value. This helps us achieve the right tradeoff between bias and variance.

The accuracy is on the training data is compromised by adding the lambda. But it makes the model work well on unseen test data.