Q1) 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 - Optimal Values

Ridge Regression – Alpha =500

Lasso Regression- Alpha=0.001

If the values for alpha is doubled for Ridge Regression-

1) Higher the value of Alpha, smaller will be will be the value of Co-efficients.
2) RSS will increase rapidly with increase in value of Alpha.
3) Higher Values of Alpha, can lead to Under fitting.

If the values for alpha is doubled for Lasso Regression-

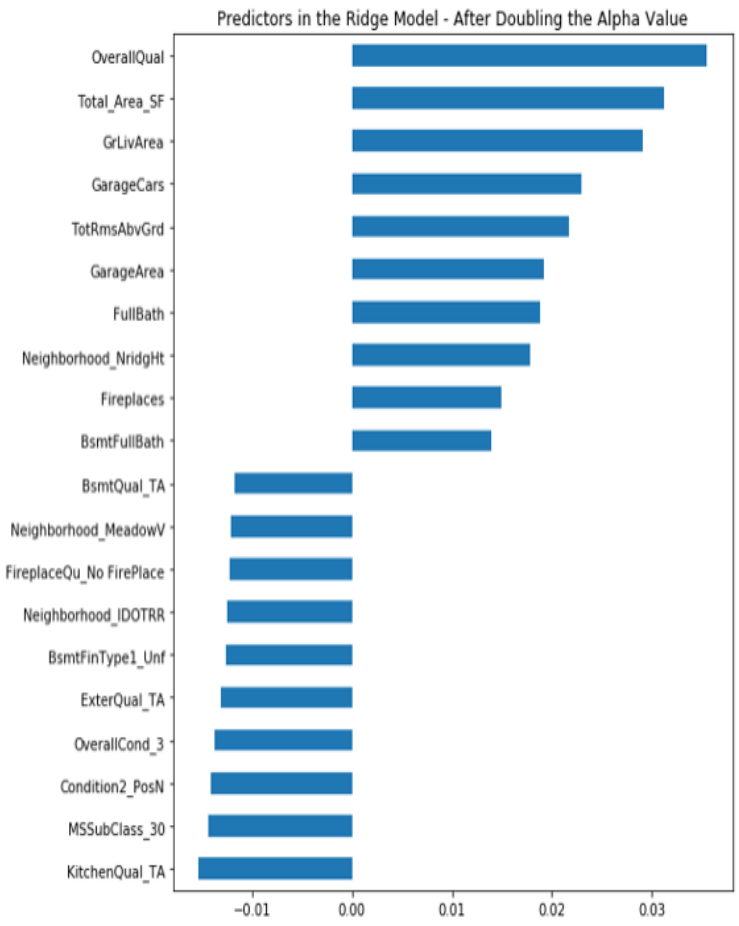
4) As Alpha Increases, Co-efficients will tend towards 0.
5) As Alpha Increases, RSS Increases Steadily.
6) As Alpha Increases, Variance Decreases Steadily.
7) As Alpha Increases, Bias Increases.

Post Doubling the values of Alpha Following are the Top 4 Predictor Variables-

Ridge Regression-

1) OverallQual
2) Total_Area_SF
3) GrLivArea
4) GarageCars

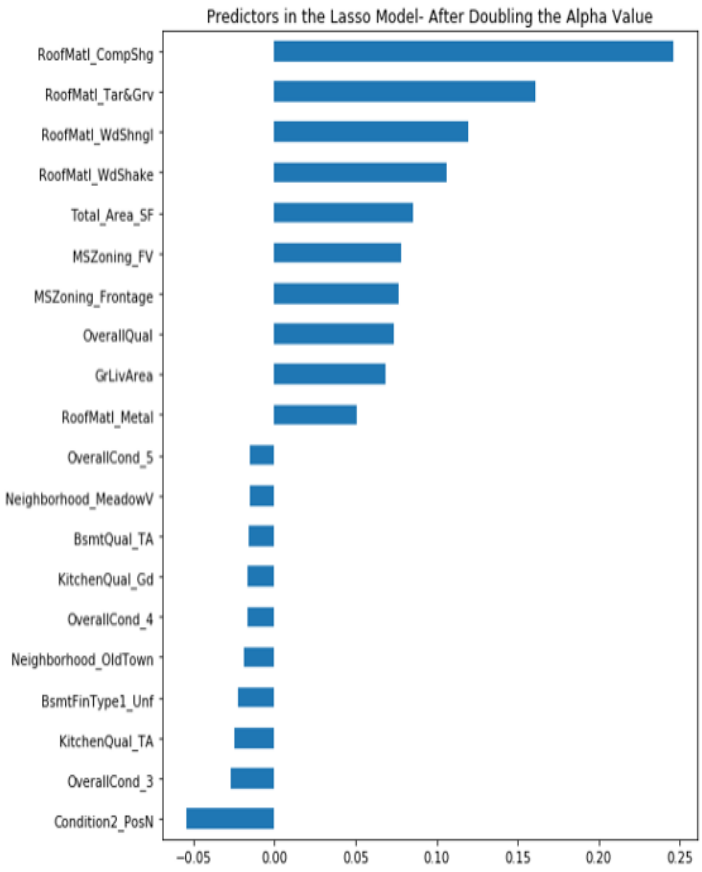
Other Predictor Variables for Ridge Regression



Lasso Regression-

a. RoofMatl_CompShg
b. RoofMatl_Tar&Grv
c. RoofMatl_WdShngl
d. RoofMatl_WdShake

Other Predictor Variables for Lasso Regression



Q2) 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-

Ridge Regression

```
#Ridge Scores
ridge_sc = Ridge(alpha=500)

#Train DaTa
ridge_sc.fit(X_train, y_train)
pred_train_rr= ridge_sc.predict(X_train)
print('Root Mean Squared Error [Train]- ',end='')
print(np.sqrt(mean_squared_error(y_train,pred_train_rr)))
print('R2 Score [Train]- ',end='')
print(r2_score(y_train, pred_train_rr)*100)
```

```
Root Mean Squared Error [Train]- 0.11338881148450287
R2 Score [Train]- 91.97771817120322
```

Lasso Regression –

```
#Lasso Scores

lasso_sc=Lasso(alpha=0.001)
lasso_sc.fit(X_train, y_train)
pred_train_lasso= lasso_sc.predict(X_train)

#Train Data

print('Root Mean Squared Error [Train]- ',end='')
print(np.sqrt(mean_squared_error(y_train,pred_train_lasso)))
print('R2 Score [Train]- ',end='')
print(r2_score(y_train, pred_train_lasso)*100)
```

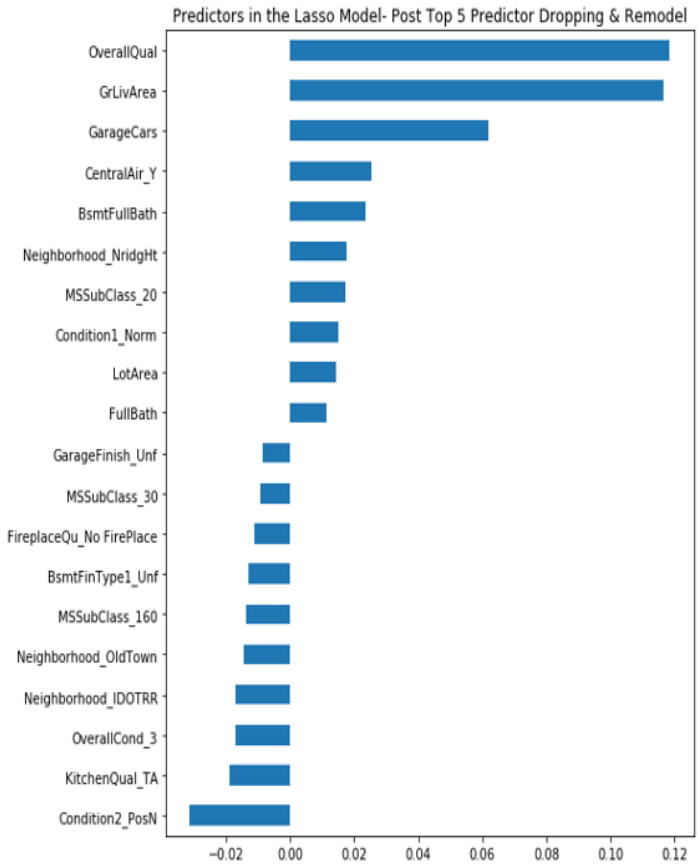
```
Root Mean Squared Error [Train]- 0.08334663415546764
R2 Score [Train]- 95.66555355891262
```

Comparing the above two statistics, R2 Score is better for Lasso Regression than Ridge Regression & also the RMSE is better for Lasso. Hence, we will proceed with Lasso Regression as Apt Model.

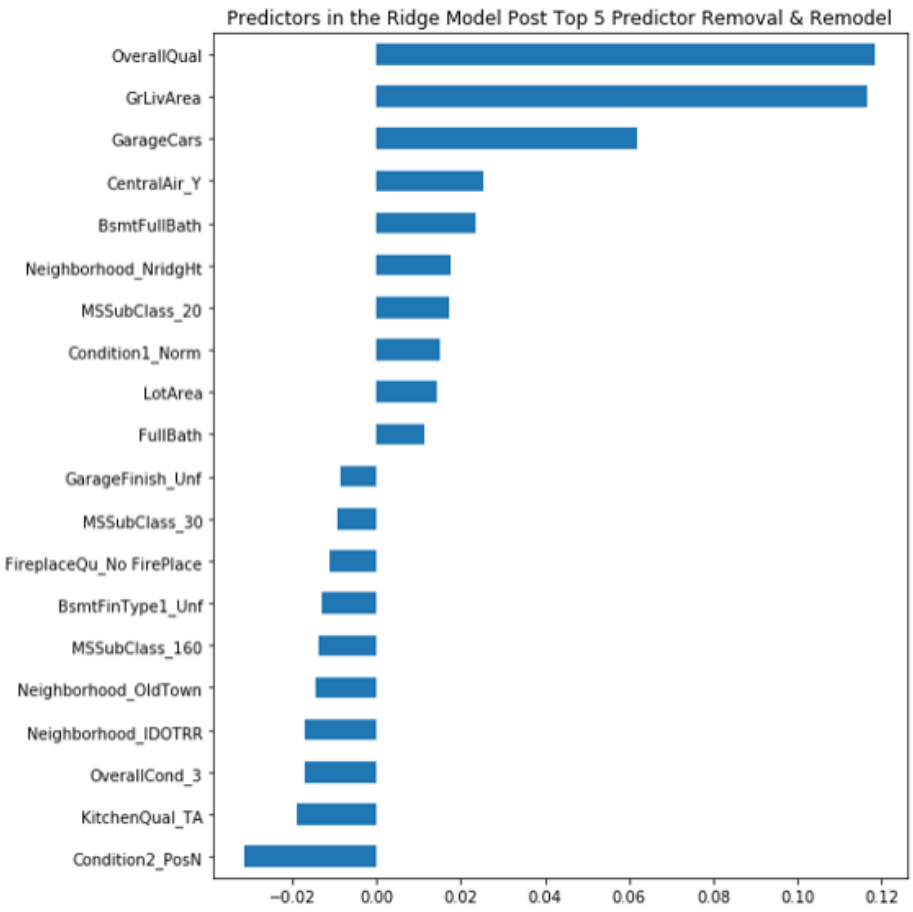
Q3) After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

Ans- Following are the Top 5 Variables-

Lasso Regression-



Ridge Regression -



Q4) How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

Ans-

For a model to be Robust and Generalizable, the model must be able to adapt & accommodate for multiple data sets. i.e. the model should not overfit. In case if the model overfitting, the model tends to absorb all the data points i.e. memorize the data points. In such a case, the model will perform excellent on Training data but poor on test data. The model should be low on Complexity i.e should not overfit. A Complex Model is sensitive to Training Data.

In other words, the trade-off between bias & variance should be optimum.

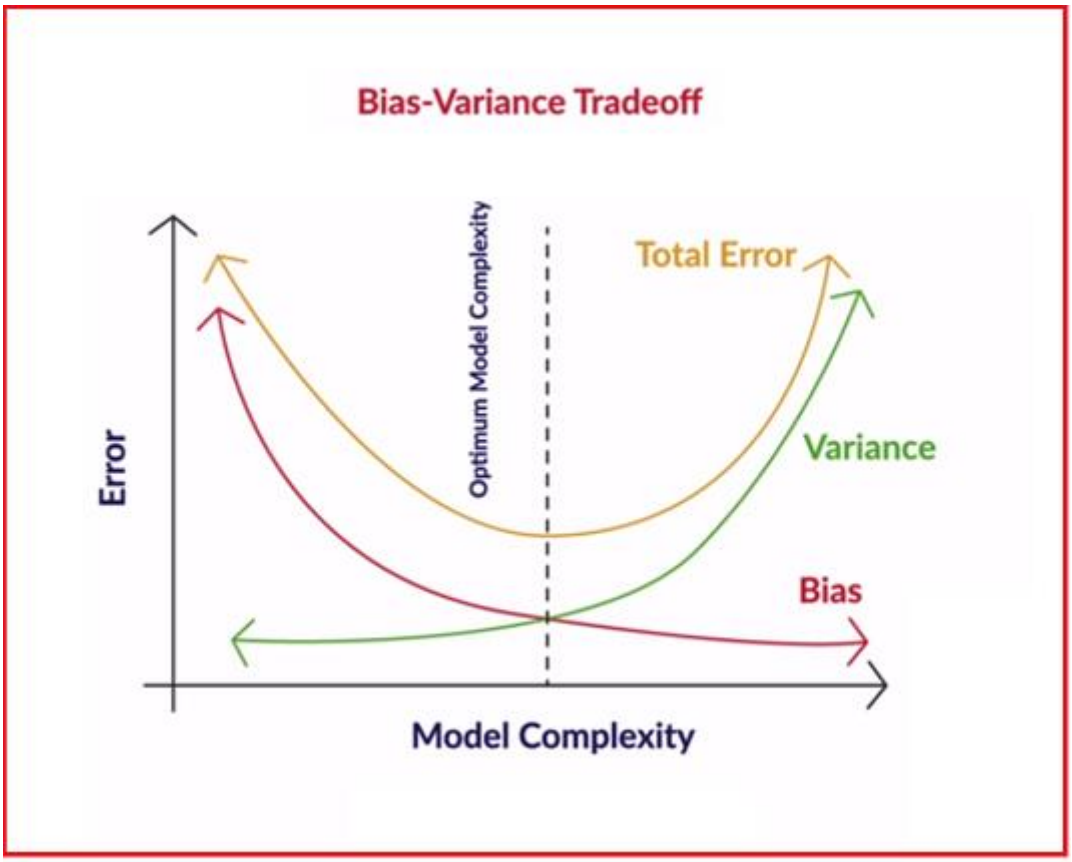
Variance – Tells how scattered are the predicted values from the actual Value

Bias- Difference between the predicted values from Actual Values.

For Robust and Generalized model, we need to have Low Variance, and Optimum Bias, with minimal Test & Train Errors. And Ideal scenario will be where both Variance & Bias are low.

Generalization can be achieved through Regularization Techniques.

Robustness & Accuracy are inversely proportional i.e. as Robustness Increases, Accuracy Decreases.



Bias Variance Tradeoff