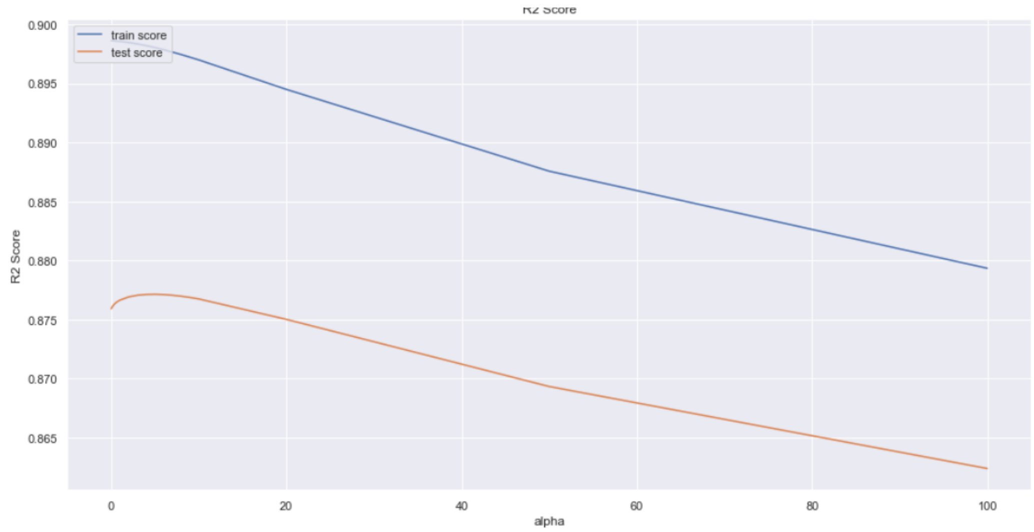
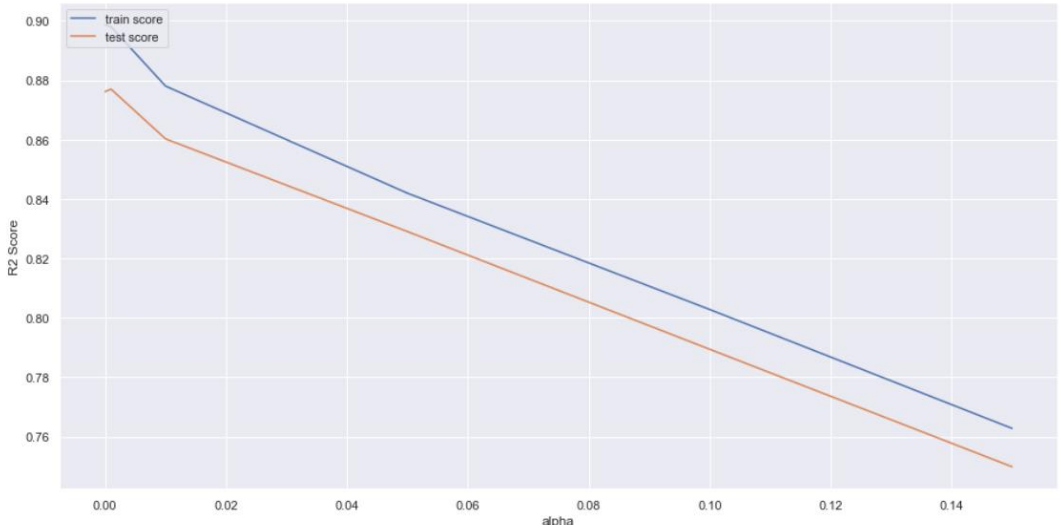
Question 1

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?

Optimal Alpha Value for Ridge: 5



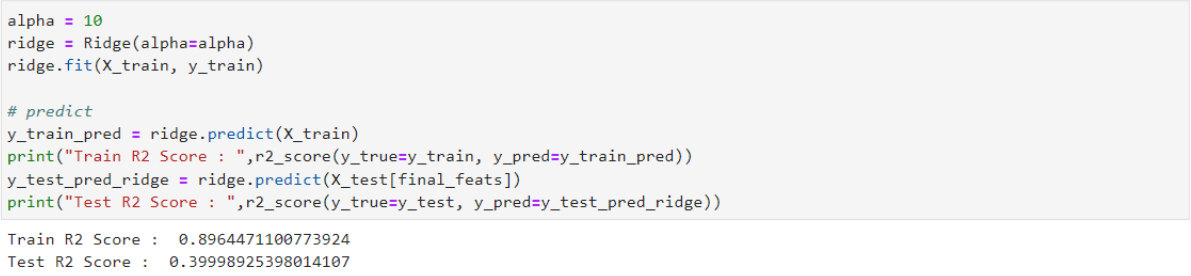
Optimal Alpha Value for Lasso: 0.001



If we double the value for alpha in Ridge and Lasso; it will have the following effect on the Model

Performance

Ridge: 10

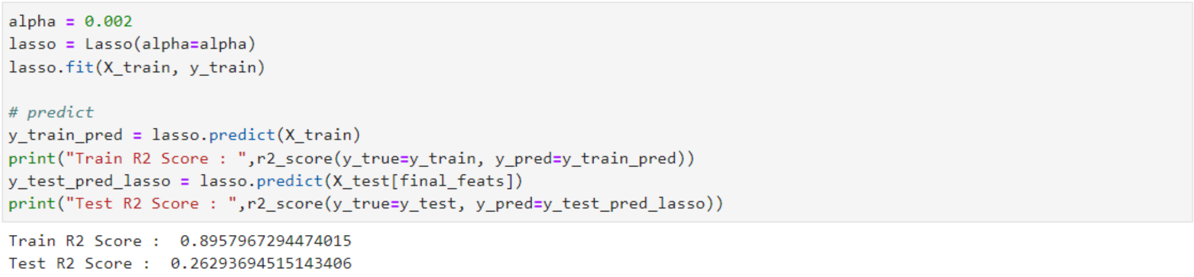


```
alpha = 10
ridge = Ridge(alpha=alpha)
ridge.fit(X_train, y_train)

# predict
y_train_pred = ridge.predict(X_train)
print("Train R2 Score : ",r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred_ridge = ridge.predict(X_test[final_feats])
print("Test R2 Score : ",r2_score(y_true=y_test, y_pred=y_test_pred_ridge))
```

```
Train R2 Score :  0.8964471100773924
Test R2 Score :  0.39998925398014107
```

Lasso: 0.002

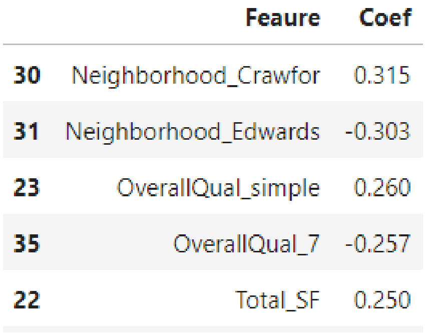


```
alpha = 0.002
lasso = Lasso(alpha=alpha)
lasso.fit(X_train, y_train)

# predict
y_train_pred = lasso.predict(X_train)
print("Train R2 Score : ",r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred_lasso = lasso.predict(X_test[final_feats])
print("Test R2 Score : ",r2_score(y_true=y_test, y_pred=y_test_pred_lasso))
```

```
Train R2 Score :  0.8957967294474015
Test R2 Score :  0.26293694515143406
```

Important Predictor Variables after doubling alpha:



	Feaure	Coef
30	Neighborhood_Crawfor	0.315
31	Neighborhood_Edwards	-0.303
23	OverallQual_simple	0.260
35	OverallQual_7	-0.257
22	Total_SF	0.250

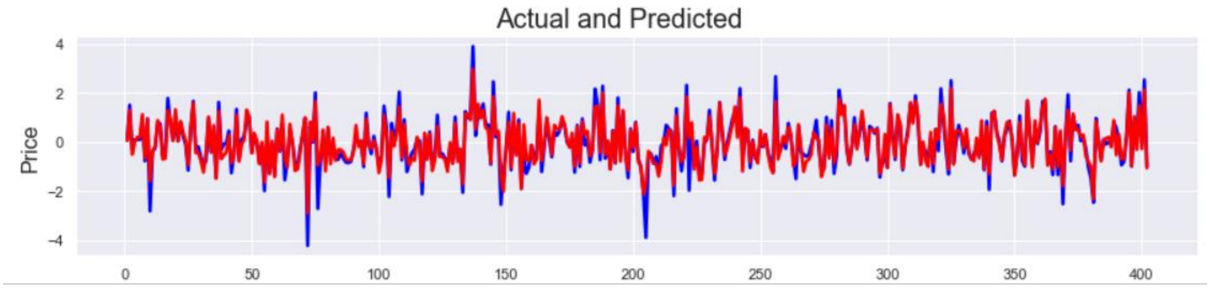
Question 2

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

Answer

Lasso (alpha 0.001) was chosen as the final model rather than the Ridge Model because both models' performances were comparable but Lasso had the feature selection advantage which made the model less complex.

Final Model Performance

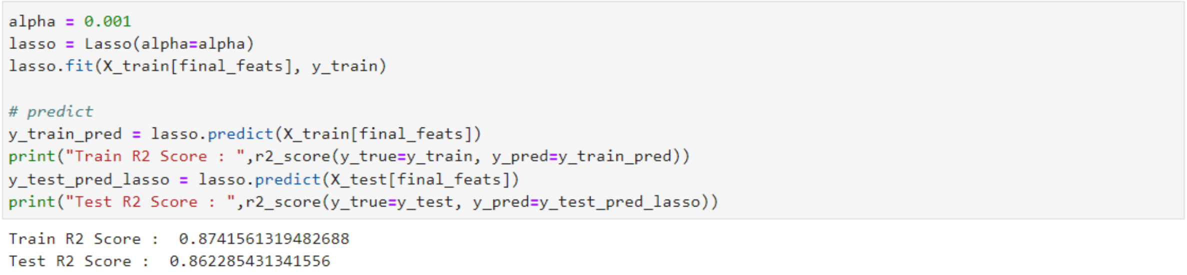


Question 3

After building the model, you realized 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

Removing the Top 5 Predictor Variables from the model and running Lasso again:

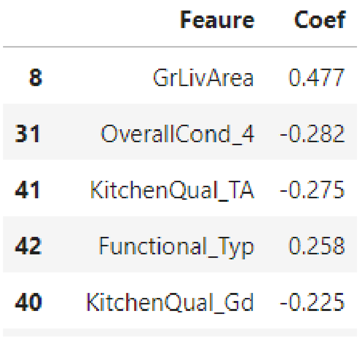


```
alpha = 0.001
lasso = Lasso(alpha=alpha)
lasso.fit(X_train[final_feats], y_train)

# predict
y_train_pred = lasso.predict(X_train[final_feats])
print("Train R2 Score : ",r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred_lasso = lasso.predict(X_test[final_feats])
print("Test R2 Score : ",r2_score(y_true=y_test, y_pred=y_test_pred_lasso))
```

```
Train R2 Score :  0.8741561319482688
Test R2 Score :  0.862285431341556
```

After removing the top 5 features, below are the new 5 top 5 features.



	Feaure	Coef
8	GrLivArea	0.477
31	OverallCond_4	-0.282
41	KitchenQual_TA	-0.275
42	Functional_Typ	0.258
40	KitchenQual_Gd	-0.225

Question 4

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?

A model is considered to be robust if the model is stable, i.e. does not change drastically upon changing the training set. The model is considered generalizable if it does not overfits the training data, and works well with new data.

Its implication in terms of accuracy is that a robust and generalizable model will perform equally well on both training and test data i.e. the accuracy does not change much for training and test data.

The final Model which was chosen had a very close Train and Test R2 Score which makes it robust with fewer chances of overfitting also when alpha values were doubled the Top 5 predictors did not change which again proves the model’s robustness and generalizability.