Advanced Regression Subjective Questions

Assignment-based Subjective Questions

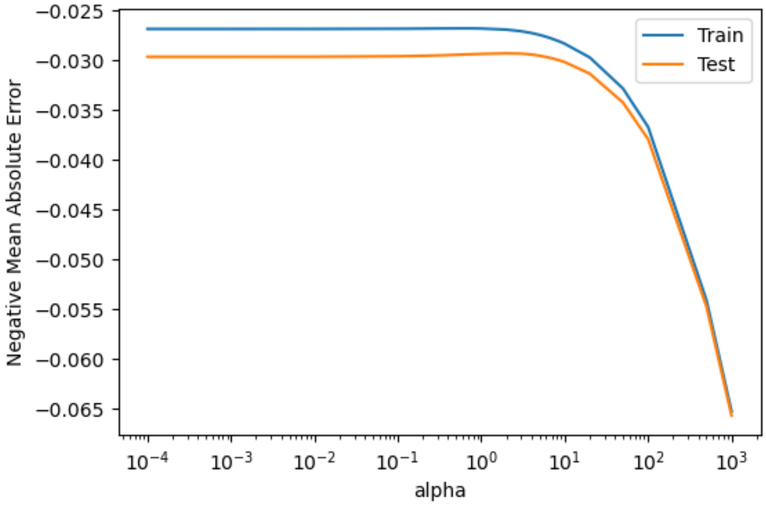
Question 1.

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?

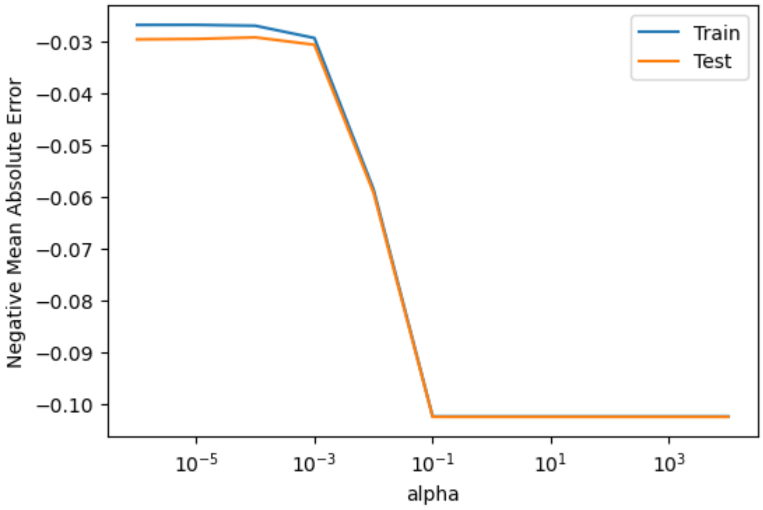
Answer –

- The Optimal Value for alpha for **Ridge Regression is 2.0**
- The Optimal Value for alpha for **Lasso Regression is 0.0001**

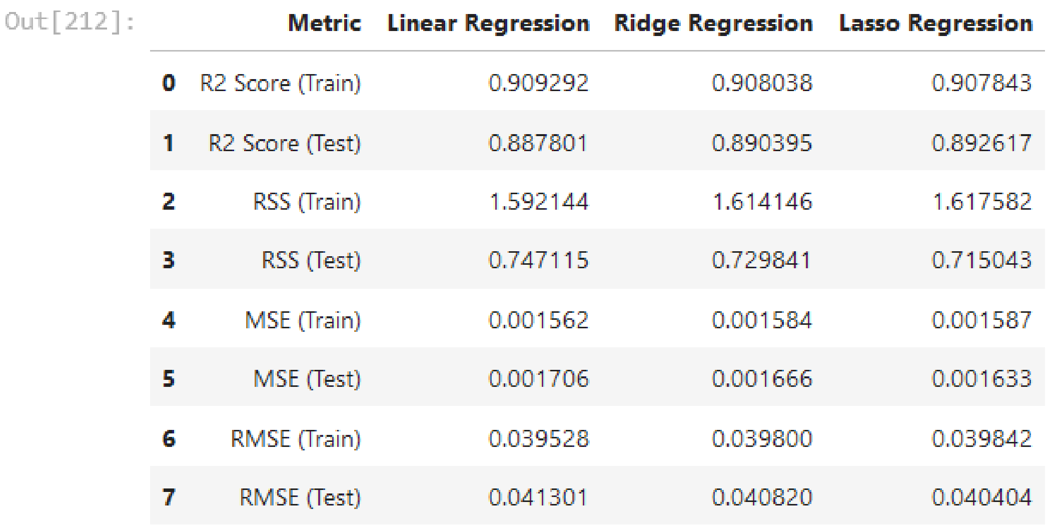
Ridge Regression -



Lasso Regression –



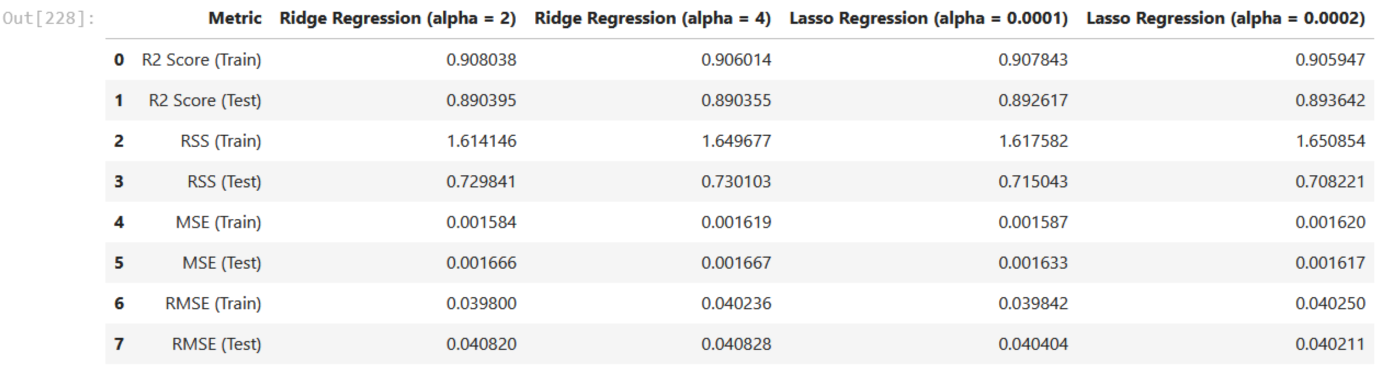
Below are the Metrics with above values of alpha:



Out[212]:

	Metric	Linear Regression	Ridge Regression	Lasso Regression
0	R2 Score (Train)	0.909292	0.908038	0.907843
1	R2 Score (Test)	0.887801	0.890395	0.892617
2	RSS (Train)	1.592144	1.614146	1.617582
3	RSS (Test)	0.747115	0.729841	0.715043
4	MSE (Train)	0.001562	0.001584	0.001587
5	MSE (Test)	0.001706	0.001666	0.001633
6	RMSE (Train)	0.039528	0.039800	0.039842
7	RMSE (Test)	0.041301	0.040820	0.040404

Below are the changes in Metrics after doubling the values of alpha for both Ridge and Lasso Regression -



Out[228]:

	Metric	Ridge Regression (alpha = 2)	Ridge Regression (alpha = 4)	Lasso Regression (alpha = 0.0001)	Lasso Regression (alpha = 0.0002)
0	R2 Score (Train)	0.908038	0.906014	0.907843	0.905947
1	R2 Score (Test)	0.890395	0.890355	0.892617	0.893642
2	RSS (Train)	1.614146	1.649677	1.617582	1.650854
3	RSS (Test)	0.729841	0.730103	0.715043	0.708221
4	MSE (Train)	0.001584	0.001619	0.001587	0.001620
5	MSE (Test)	0.001666	0.001667	0.001633	0.001617
6	RMSE (Train)	0.039800	0.040236	0.039842	0.040250
7	RMSE (Test)	0.040820	0.040828	0.040404	0.040211

Changes in Metrics after doubling alpha-

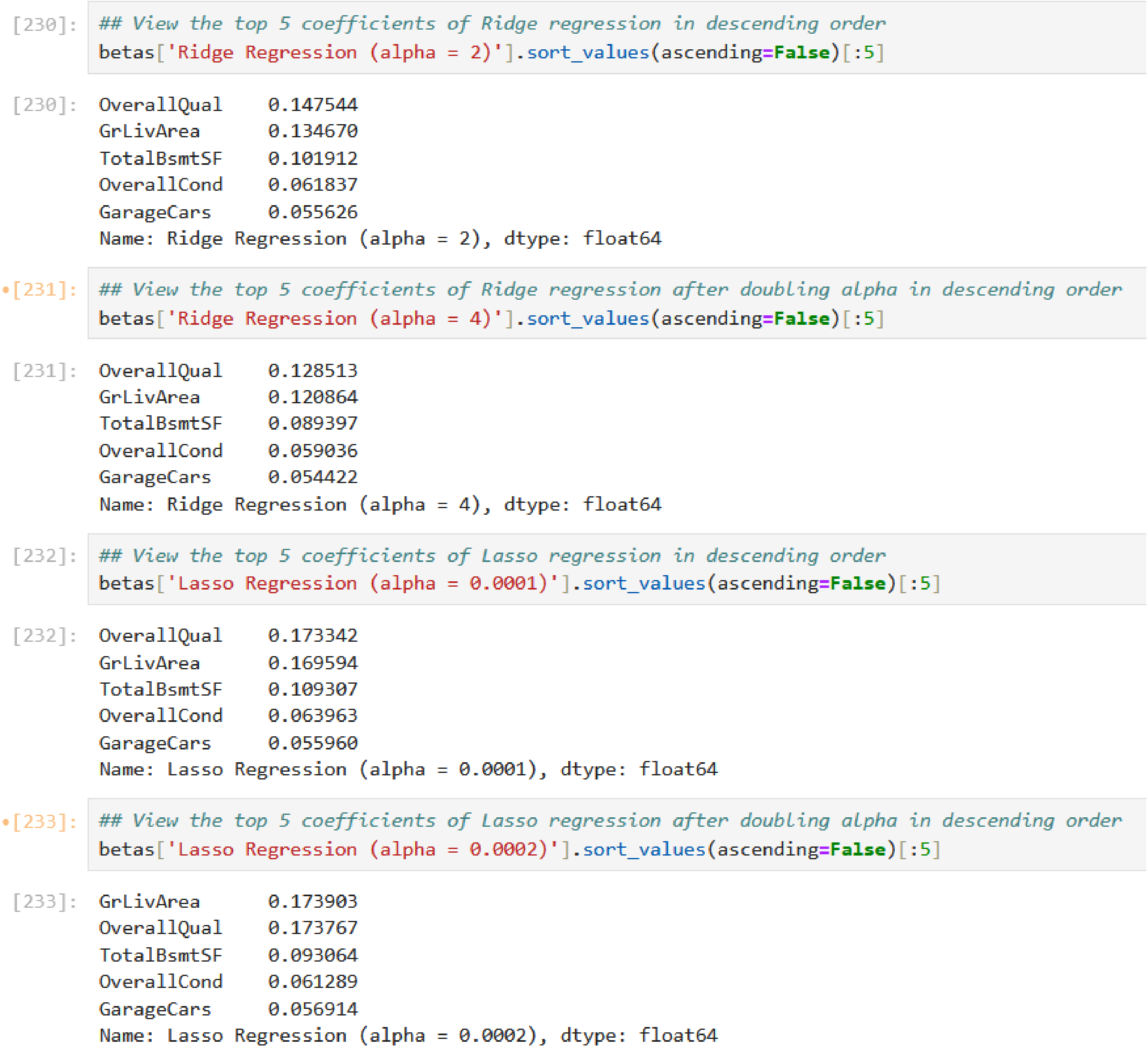
Changes in Ridge Regression metrics:

- R2 score of train decreased very slightly from 0.9080 to 0.9060 by 0.002
- R2 score of test decreased very slightly from 0.89039 to 0.89035 by 0.00004

Changes in Lasso Regression metrics:

- R2 score of train decreased very slightly from 0.9078 to 0.9059 by 0.0016
- R2 score of test increased very slightly from 0.8926 to 0.8936 by 0.001

Top 5 Significant Features for Ridge and Lasso Regression before and after doubling -



```
[230]: ## View the top 5 coefficients of Ridge regression in descending order
       betas['Ridge Regression (alpha = 2)'].sort_values(ascending=False)[:5]

[230]: OverallQual    0.147544
       GrLivArea      0.134670
       TotalBsmtSF    0.101912
       OverallCond    0.061837
       GarageCars     0.055626
       Name: Ridge Regression (alpha = 2), dtype: float64

•[231]: ## View the top 5 coefficients of Ridge regression after doubling alpha in descending order
       betas['Ridge Regression (alpha = 4)'].sort_values(ascending=False)[:5]

[231]: OverallQual    0.128513
       GrLivArea      0.120864
       TotalBsmtSF    0.089397
       OverallCond    0.059036
       GarageCars     0.054422
       Name: Ridge Regression (alpha = 4), dtype: float64

[232]: ## View the top 5 coefficients of Lasso regression in descending order
       betas['Lasso Regression (alpha = 0.0001)'].sort_values(ascending=False)[:5]

[232]: OverallQual    0.173342
       GrLivArea      0.169594
       TotalBsmtSF    0.109307
       OverallCond    0.063963
       GarageCars     0.055960
       Name: Lasso Regression (alpha = 0.0001), dtype: float64

•[233]: ## View the top 5 coefficients of Lasso regression after doubling alpha in descending order
       betas['Lasso Regression (alpha = 0.0002)'].sort_values(ascending=False)[:5]

[233]: GrLivArea      0.173903
       OverallQual    0.173767
       TotalBsmtSF    0.093064
       OverallCond    0.061289
       GarageCars     0.056914
       Name: Lasso Regression (alpha = 0.0002), dtype: float64
```

Observation –

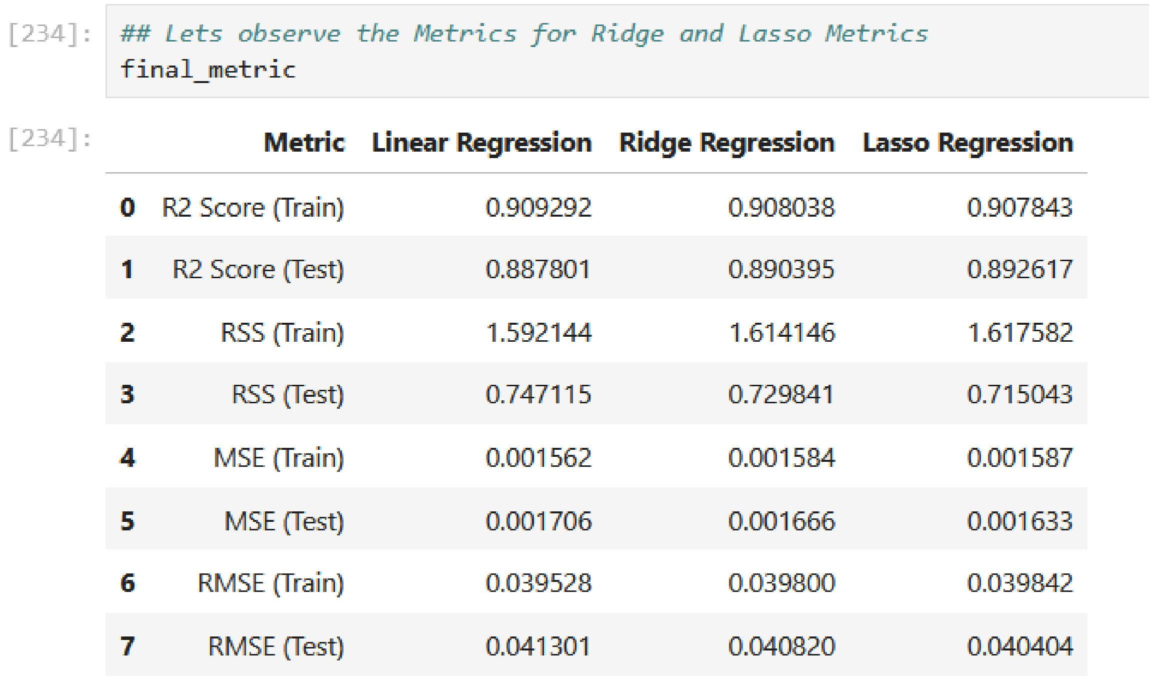
The top 5 Predictors are same but the coefficients of these predictors have changed.

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 –

- The Regression Model we choose to apply will depend on the use case.
- If there are too many feature variables and one of our primary goal is feature selection, then we can use Lasso.
- If we don't want to get too large coefficients and reduction of coefficient magnitude is one of our prime goals, then we will use Ridge Regression.



```
[234]: ## Lets observe the Metrics for Ridge and Lasso Metrics
       final_metric
```

[234]:

	Metric	Linear Regression	Ridge Regression	Lasso Regression
0	R2 Score (Train)	0.909292	0.908038	0.907843
1	R2 Score (Test)	0.887801	0.890395	0.892617
2	RSS (Train)	1.592144	1.614146	1.617582
3	RSS (Test)	0.747115	0.729841	0.715043
4	MSE (Train)	0.001562	0.001584	0.001587
5	MSE (Test)	0.001706	0.001666	0.001633
6	RMSE (Train)	0.039528	0.039800	0.039842
7	RMSE (Test)	0.041301	0.040820	0.040404

Observation

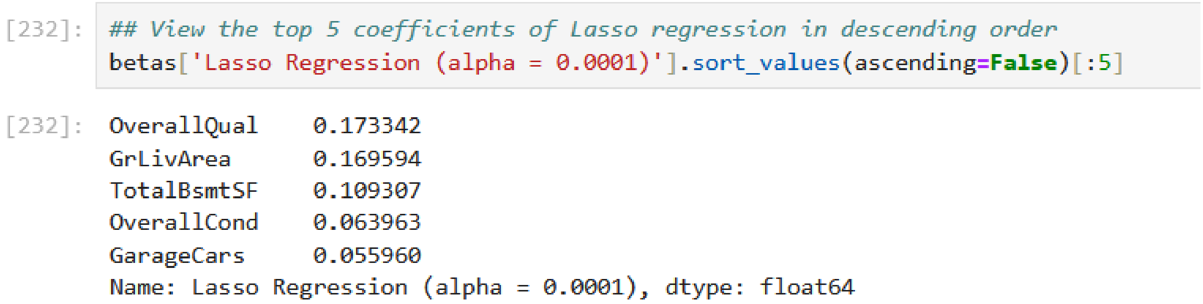
- In the final model we observe that Lasso has performed feature selection and only included 60 features from total of 71 features and the R2 Score of lasso is slightly higher than ridge for the test dataset so we will choose lasso regression to solve Surprise Housing Use Case.

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?

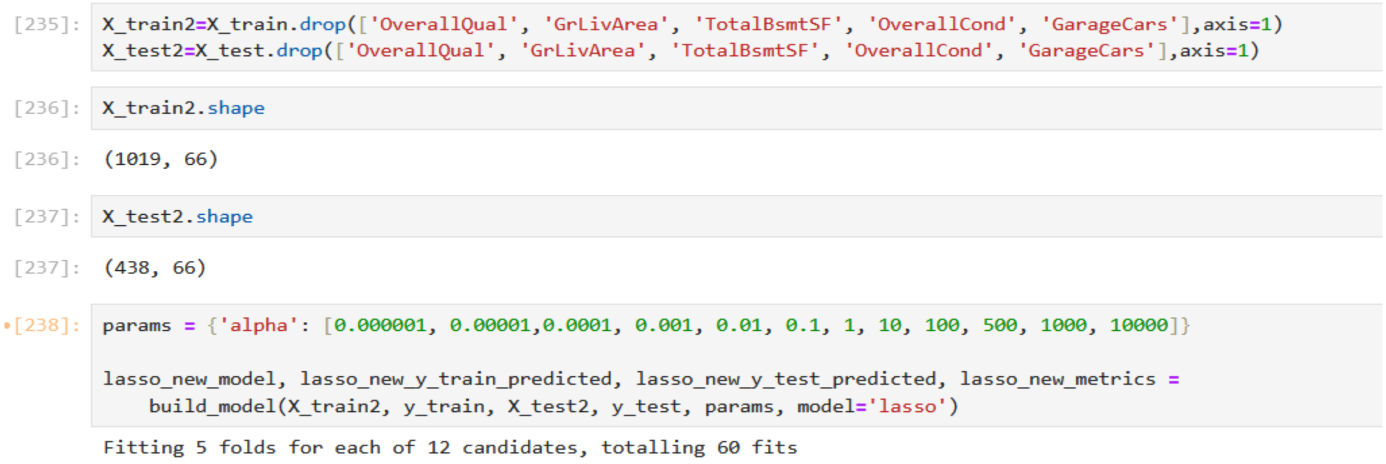
Answer –

Top 5 Features of Lasso Regression before dropping the most important 5 predictors-

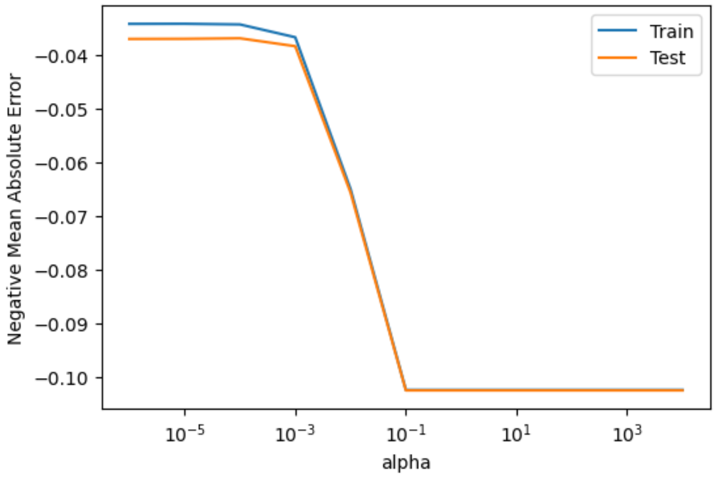


```
[232]: ## View the top 5 coefficients of Lasso regression in descending order
       betas['Lasso Regression (alpha = 0.0001)'].sort_values(ascending=False)[:5]

[232]: OverallQual    0.173342
       GrLivArea      0.169594
       TotalBsmtSF    0.109307
       OverallCond    0.063963
       GarageCars     0.055960
       Name: Lasso Regression (alpha = 0.0001), dtype: float64
```

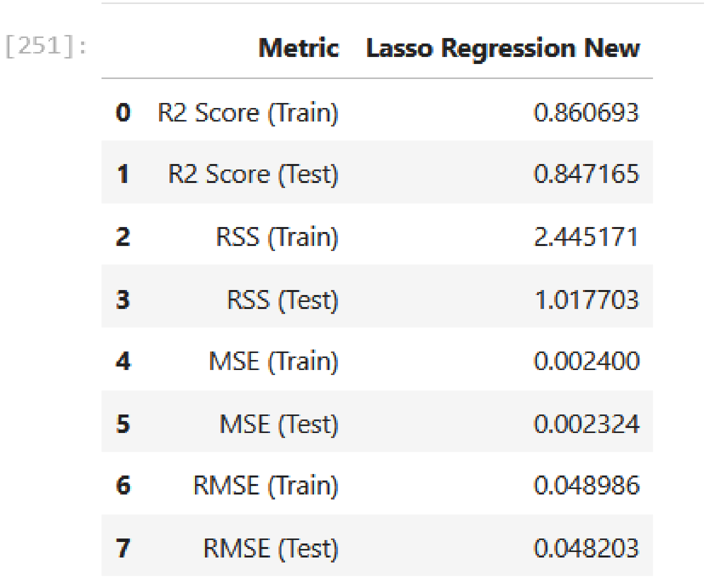
Dropping the 5 most important predictors for Lasso and rebuilding new model –



```
[235]: X_train2=X_train.drop(['OverallQual', 'GrLivArea', 'TotalBsmtSF', 'OverallCond', 'GarageCars'],axis=1)
       X_test2=X_test.drop(['OverallQual', 'GrLivArea', 'TotalBsmtSF', 'OverallCond', 'GarageCars'],axis=1)

[236]: X_train2.shape

[236]: (1019, 66)

[237]: X_test2.shape

[237]: (438, 66)

•[238]: params = {'alpha': [0.000001, 0.00001,0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500, 1000, 10000]}

        lasso_new_model, lasso_new_y_train_predicted, lasso_new_y_test_predicted, lasso_new_metrics =
            build_model(X_train2, y_train, X_test2, y_test, params, model='lasso')
        Fitting 5 folds for each of 12 candidates, totalling 60 fits
```



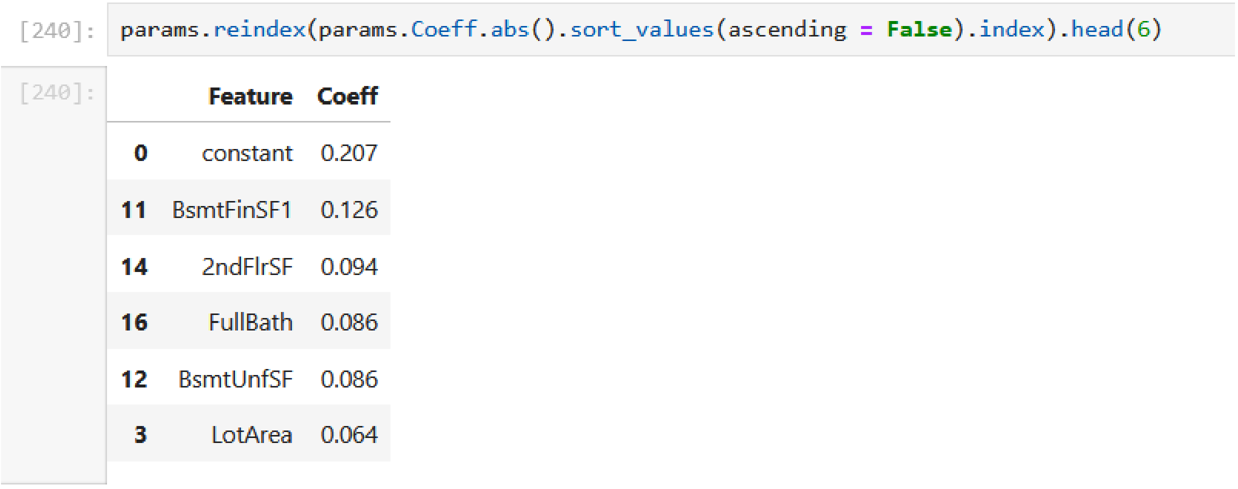
New Lasso Regression Model Metrics –



[251]:

	Metric	Lasso Regression New
0	R2 Score (Train)	0.860693
1	R2 Score (Test)	0.847165
2	RSS (Train)	2.445171
3	RSS (Test)	1.017703
4	MSE (Train)	0.002400
5	MSE (Test)	0.002324
6	RMSE (Train)	0.048986
7	RMSE (Test)	0.048203

Top 5 Important Predictors of New Lasso Regression Model –



[240]:

```
params.reindex(params.Coeff.abs().sort_values(ascending = False).index).head(6)
```

[240]:

	Feature	Coeff
0	constant	0.207
11	BsmtFinSF1	0.126
14	2ndFlrSF	0.094
16	FullBath	0.086
12	BsmtUnfSF	0.086
3	LotArea	0.064

Top 5 Predictor Variables using Lasso after Dropping 5 Important Predictor are -

- BsmtFinSF1
- 2ndFlrSF
- FullBath
- BsmtUnfSF
- LotArea

Question 4.

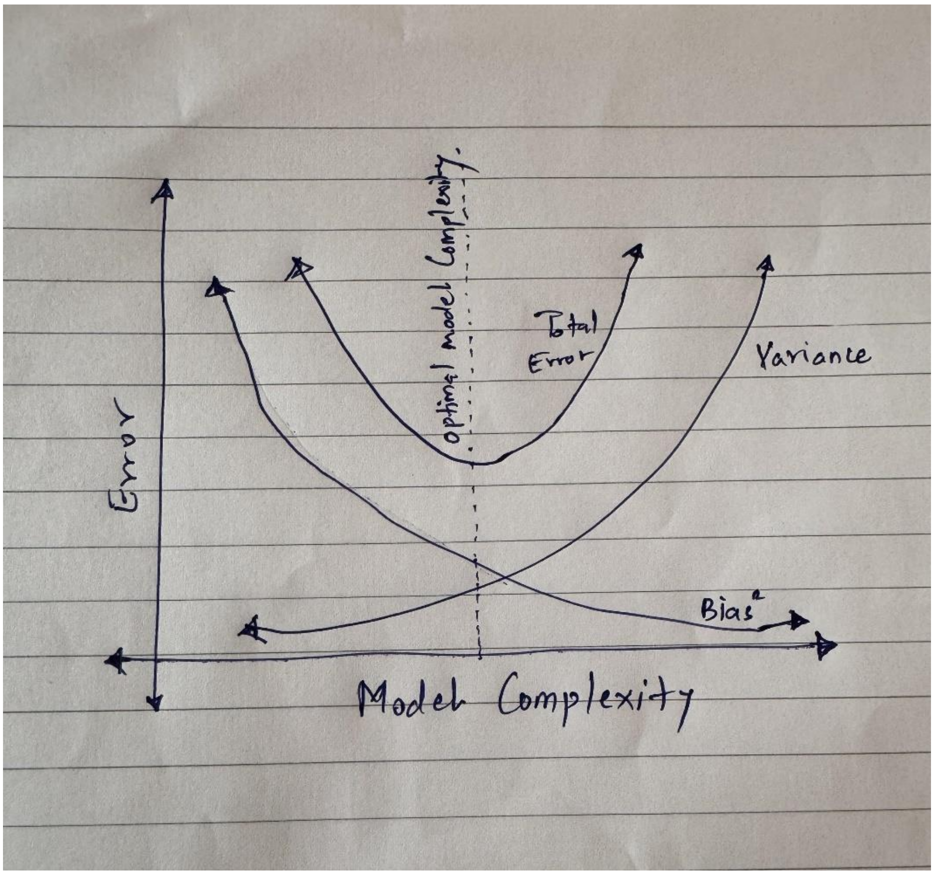
How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?

Answer –

- A generalizable model is able to adapt and learn properly to new and previously unseen data, from the same distribution as the one used to build the model.
- A model is robust when any variation in the data does not impact its performance to great extent.
- To ensure a model is both robust and generalizable, we have to make sure it doesnot overfit. This is because an overfitting model has very high variance and a minor change in data impacts the model prediction heavily. Such a model will identify and memorise all the patterns of a training data, but fail to pick up the patterns in unseen test data. The model should be generalized so that the test accuracy is similar to the training score.
- Outliers should not be given too much significance so that the accuracy predicted by the model is high. To ensure that this is not the case, the outlier analysis needs to be done and only those which are relevant to the dataset need to be retained. Those outliers which it does not make sense to keep must be removed from the dataset. We have done outlier analysis for our use case and treated them so that we do not loose data and at the same time the impact of outliers is reduced.
- In other words, the model should not be too complex in order to be robust and generalizable.
- From Accuracy perspective, a too complex model will have a very high accuracy but if the model is not robust, it cannot be trusted for predictive analysis. 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.
- In general, we have to find strike some balance between model accuracy and complexity. This can be achieved by Regularization techniques like Ridge Regression and Lasso.

Bias- Variance trade-off -

- Model with high bias pay very little attention to training data on the other hand model with high variance pays lot of attention to training data. Accuracy of simple, robust and generalizable model will not have much difference in Training and Testing Data. We need to find the good balance without overfitting or underfitting the dataset to make the model more robust and generalisable.


optimal model Complexity.
Total Error
Variance
Error
Bias²
Model Complexity