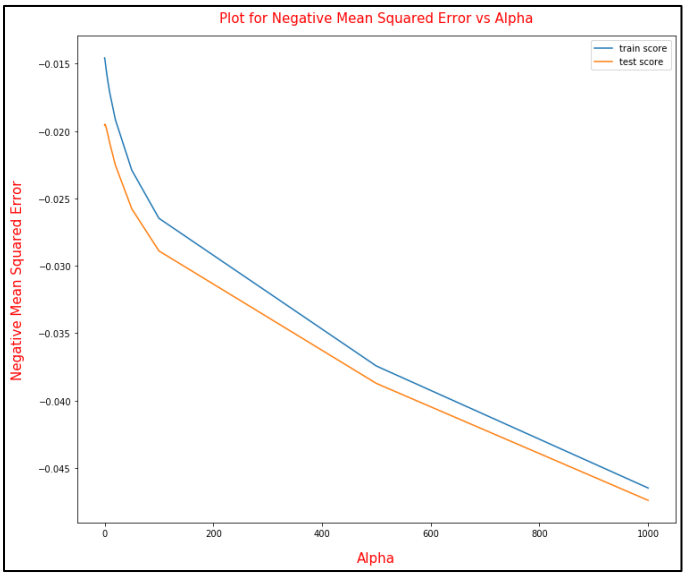
Question 1:

What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose to 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** is **0.6**

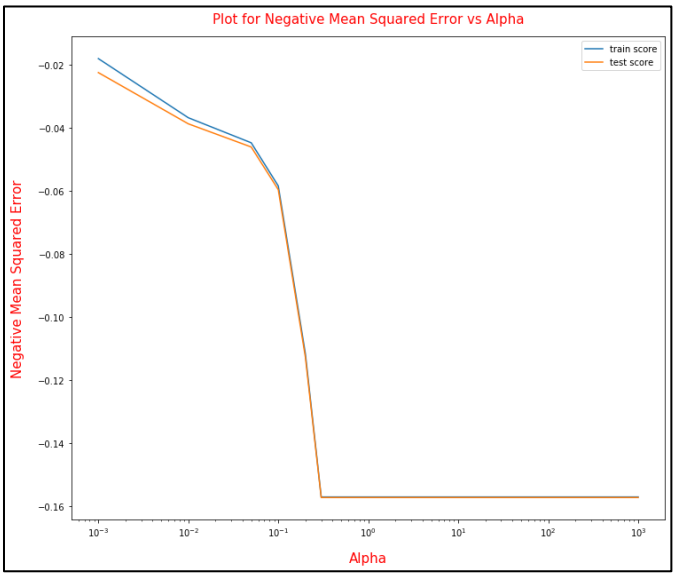


Model performance metrics for Ridge regression with Alpha **0.6 on both Train and Test dataset** is shown below:

```
For Train Dataset Evaluation : Ridge Regression
-----------------------------------
MSE:0.015092532018860733
RMSE:0.12285166673212347
R2 Score:0.9039575362628318
R2 Score Percentage:90.39575362628318
-----------------------------------
```

```
For Test Dataset Evaluation : Ridge Regression
-----------------------------------
MSE:0.02064973247081077
RMSE:0.14370014777588355
R2 Score:0.8747094586823874
R2 Score Percentage:87.47094586823874
-----------------------------------
```

The **Optimal value** of alpha for **Lasso** is **0.001**



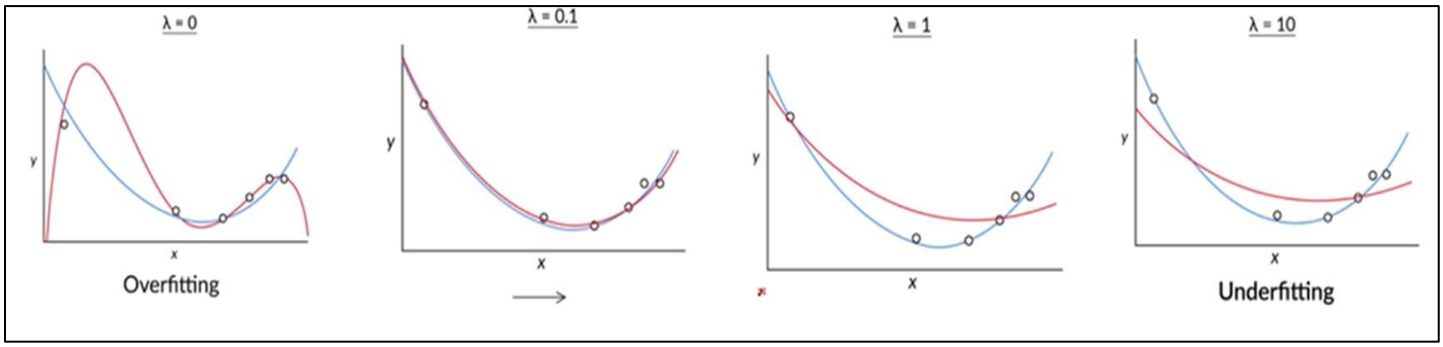
Model performance metrics for Lasso regression with Alpha **0.001 on both Train and Test dataset** is shown below:

```
For Train Dataset Evaluation : Lasso Regression
-----------------------------------
MSE:0.018510898357847964
RMSE:0.13605476234901873
R2 Score:0.8822045047143641
R2 Score Percentage:88.22045047143641
-----------------------------------
```

```
For Test Dataset Evaluation : Lasso Regression
-----------------------------------
MSE:0.021874904081283005
RMSE:0.14790167031268783
R2 Score:0.8672758314186919
R2 Score Percentage:86.72758314186919
-----------------------------------
```

If we **double the value alpha for both Ridge and Lasso**, find below what happens to evaluation metrics and predictors.

Basically, if we keep on **increasing** the value of alpha from its optimal value, the model will become **underfit** and if keep on **decreasing** the value for alpha from its optimal value the model will **overfit** (alpha =0) as shown below:



Now let's how our model behave after doubling the values of alpha

The value of alpha for **Ridge** after double is **1.2**

Model performance metrics for Ridge regression with Alpha **1.2 on both Train and Test dataset** is shown below:

```
For Train Dataset Evaluation : Ridge Regression
-----------------------------------
MSE:0.015240514339808409
RMSE:0.12345247806264728
R2 Score:0.9030158396226923
R2 Score Percentage:90.30158396226923
-----------------------------------
```

```
For Test Dataset Evaluation : Ridge Regression
-----------------------------------
MSE:0.020409439860617225
RMSE:0.1428616108708607
R2 Score:0.876167414190931
R2 Score Percentage:87.6167414190931
-----------------------------------
```

The value of alpha for **Lasso** after double is **0.002**

Model performance metrics for Lasso regression with Alpha **0.002 on both Train and Test dataset** is shown below:

```
For Train Dataset Evaluation : Lasso Regression
-----------------------------------
MSE:0.02127440417062082
RMSE:0.14585747896704104
R2 Score:0.8646187274253705
R2 Score Percentage:86.46187274253705
-----------------------------------
```

```
For Test Dataset Evaluation : Lasso Regression
-----------------------------------
MSE:0.024275272761846867
RMSE:0.15580523984079248
R2 Score:0.8527117932755923
R2 Score Percentage:85.27117932755924
-----------------------------------
```

Below are the **most important predictor variables after and before the change is implemented** are shown below:

Important top 5 Predictors Ridge Regression (alpha=0.6)

Features	Coefficients
OverallQual_Poor	-0.430963
OverallQual_Excellent	0.233964
OverallCond_Fair	-0.225721
Neighborhood_Crawfor	0.216982
HeatingQC_Po	-0.194974

Important top 5 Predictors Lasso Regression (alpha=0.001)

Features	Coefficients
OverallQual_Excellent	0.246243
OverallCond_Fair	-0.214216
Neighborhood_Crawfor	0.167193
BsmtQual_No Basement	-0.165331
OverallQual_Very Good	0.148884

Important top 5 Predictors Ridge Regression (alpha=1.2)

Features	Coefficients
OverallQual_Poor	-0.344325
OverallQual_Excellent	0.229300
OverallCond_Fair	-0.222252
Neighborhood_Crawfor	0.210999
Neighborhood_ClearCr	0.179105

Important top 5 Predictors Lasso Regression (alpha=0.002)

Features	Coefficients
OverallQual_Excellent	0.215806
OverallCond_Fair	-0.154575
GrLivArea	0.145179
OverallQual_Very Good	0.140872
Neighborhood_Crawfor	0.136954

Note:

All python coding and models for above explanation and there in Jupyter notebook

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 optimal lambda value in case of Ridge and Lasso is as below:**
  - **Ridge - 0.6**
  - **Lasso - 0.001**
- **The Mean Squared error in case of Ridge and Lasso on both Train and Test set are:**
  - Ridge (MSE on Train) - 0.015092532018860733
  - Ridge (MSE on Test) - 0.02064973247081077
  - Lasso (MSE on Train) - 0.018510898357847964
  - Lasso (MSE on Test) - 0.021874904081283005
- **The R2 score in case of Ridge and Lasso on both Train and Test set are:**
  - Ridge (R2 on Train) - 0.9039575362628318
  - Ridge (R2 on Test) - 0.8747094586823874
  - Lasso (R2 on Train) - 0.8822045047143641
  - Lasso (R2 on Test) - 0.8672758314186919

As we **got almost good score for both the models (Ridge and Lasso)**, there are very minute difference between performance of Ridge and Lasso as shown above, but specifically Ridge has very slight improvement on the performance over Lasso.

But in this **case, we will go with Lasso regression as it helps in feature reduction/selection (as the coefficient value of some of the feature became exactly 0),** here, Lasso has a better edge over Ridge. Also, the performance of Lasso is good (very minute difference than Ridge on both MSE and in R2score). Therefore, the variables predicted by Lasso can be applied to choose significant variables for predicting the price of a house.

Note:

All python coding and models for above explanation and there in Jupyter notebook

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:

Important top 5 Predictors for Lasso Regression (alpha=0.001) earlier	**Important top 5** Predictors for Lasso Regression (alpha=0.001) after **dropping the earlier 5 important predictors**

Features	Coefficients
OverallQual_Excellent	0.246243
OverallCond_Fair	-0.214216
Neighborhood_Crawfor	0.167193
BsmtQual_No Basement	-0.165331
OverallQual_Very Good	0.148884

Features	Coefficients
BsmtFinType1_No Basement	-0.202419
Neighborhood_NridgHt	0.157007
GrLivArea	0.155257
KitchenQual_Fa	-0.134786
MSSubClass_2-STORY PUD - 1946 & NEWER	-0.128252

Note:

All python coding and models for above explanation and there in Jupyter notebook

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?

Answer:

Per, **Occam's Razor**—is perhaps the **most important thumb rule in machine learning and is incredibly 'simple' at the same time**. Advantage of Simple model are as per below reasons: -

- A **simple model is usually more generic than a complex model**. This becomes important because generic models are bound to perform better on unseen data sets.
- A **simple model requires fewer training data points**. This becomes extremely important because in many cases, one must work with limited data points.
- A **simple model is more robust and does not change significantly if the training data points undergo small changes.**
- A **simple model may make a greater number of errors in the training phase, but it is bound to outperform complex models when it processes new data.**

Therefore, **to make the model more robust and generalizable, make the model simple but not simpler which will not be of any use.**

We can make sure the model is robust and general if the model performs well on unseen data and this can be achieved by **regularization** by **introducing some bias** (**decrease some accuracy on training data**) in the model and **in return significant reduction of the variance will occur in test as well as increase in accuracy on test/unseen data.**

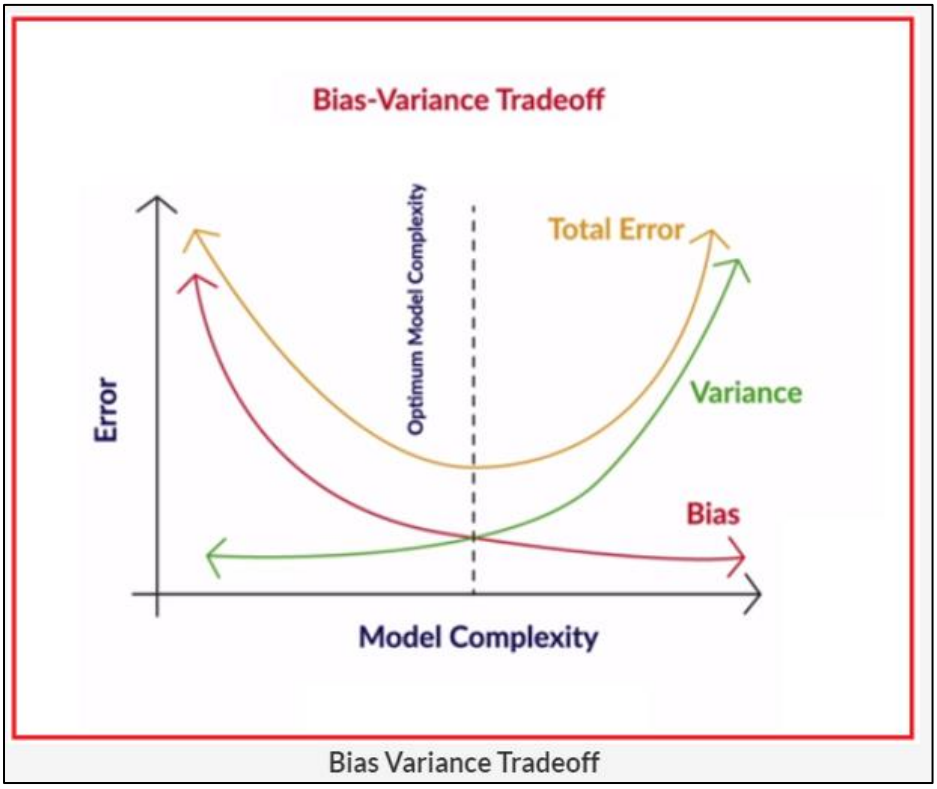
Regularization helps with managing model complexity by essentially shrinking the model coefficient estimates towards 0. This discourages the model from becoming too complex, thus avoiding the risk of overfitting. Also, Making a model simple leads **to Bias-Variance Trade-off**:

- A complex model will need to change for every little change in the dataset and hence is very unstable and extremely sensitive to any changes in the training data.
- A simple model that abstracts out some pattern followed by the data points given is unlikely to change wildly even if more points are added or removed.

Bias is the difference between this estimator's expected value and the true value of the parameter being estimated. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

Variance is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).

Thus, **accuracy of the model can be maintained by keeping the balance between Bias and Variance as it minimizes the total error as shown in the below graph.**



Bias Variance Tradeoff