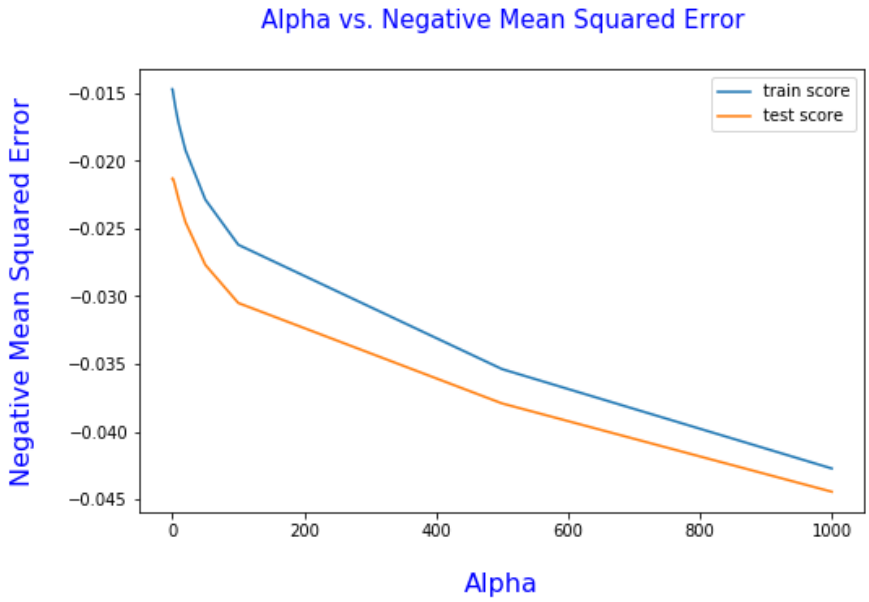
Question 1:

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 regression** is **0.3**

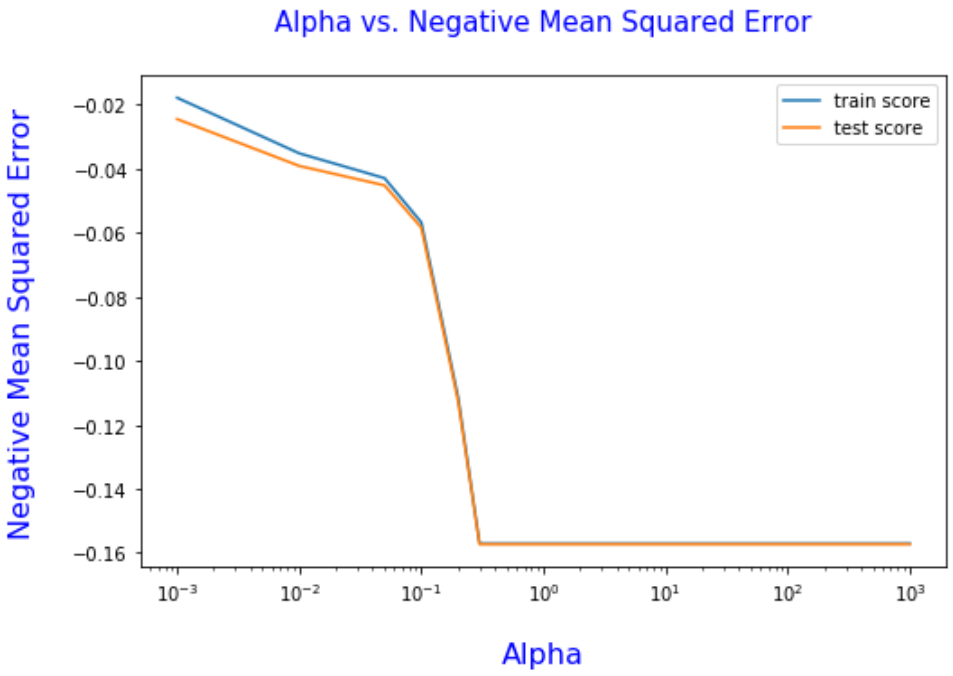


Model performance metrics for Ridge regression with Alpha **0.3 on both Train and Test dataset** is shown below:

```
Train Dataset:
-------------------------------------
Ridge Regression: Evaluation metrics
-------------------------------------
MSE: 0.015256828673062094
RMSE: 0.12351853574691571
R2 Square: 0.9029120221348146
```

```
Test Dataset:
-------------------------------------
Ridge Regression: Evaluation metrics
-------------------------------------
MSE: 0.021362169298106772
RMSE: 0.14615802851060483
R2 Square: 0.8703868072450047
```

The **Optimal_value** of alpha for **Lasso regression** is **0.001**



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

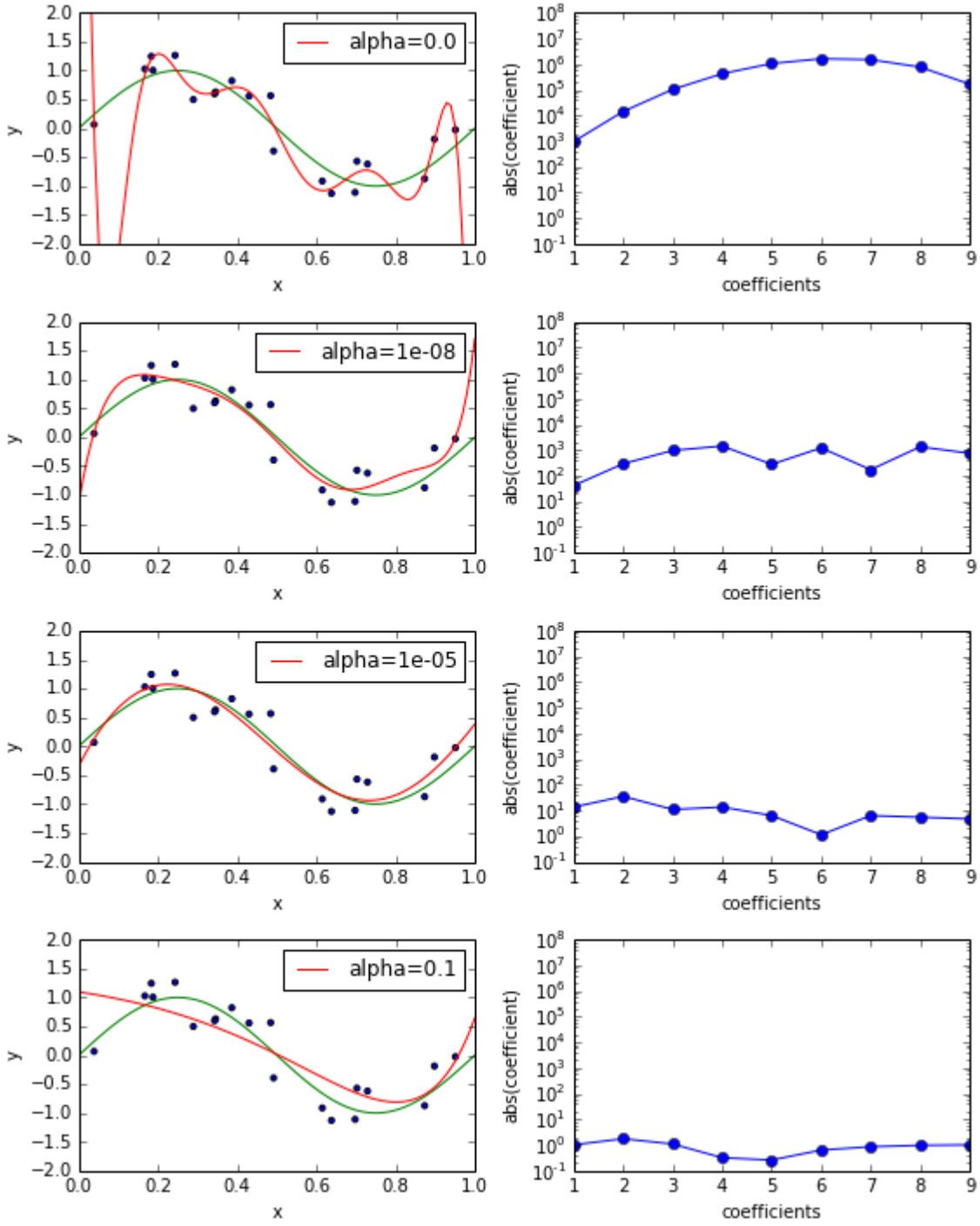
```
Train Dataset:
-------------------------------------
Lasso Regression: Evaluation metrics
-------------------------------------
MSE: 0.018552541023282743
RMSE: 0.1362077127892644
R2 Square: 0.8819395084778187


Test Dataset:
-------------------------------------
Lasso Regression: Evaluation metrics
-------------------------------------
MSE: 0.022156752401838172
RMSE: 0.14885144407038237
R2 Square: 0.8655657400796524
```

If we **double the value alpha for both Ridge and Lasso**, below will be the impact on the model.

- With increasing value of alpha from its optimal value model will start underfitting.
- As a result of that, the model coefficients will decrease more because for underfitted model the cost function increases.

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).

From below figure, we can see how the model moves from **overfitting to underfitting with increasing alpha** value and how **coefficients values are decreased with alpha increases.**



Below we can see how our model behaves after doubling the values of alpha

The value of alpha for **Ridge regression** after double is **0.6**

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

```
Train Dataset:
----------------------------------
Ridge Regression: Evaluation metrics
----------------------------------
MSE: 0.015315795613068091
RMSE: 0.12375700227893406
R2 Square: 0.9025367815727843


Test Dataset:
----------------------------------
Ridge Regression: Evaluation metrics
----------------------------------
MSE: 0.02115576451921567
RMSE: 0.14545021319756005
R2 Square: 0.8716391511441018
```

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

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

```
Train Dataset:
----------------------------------
Lasso Regression: Evaluation metrics
----------------------------------
MSE: 0.021367158330540134
RMSE: 0.1461750947683638
R2 Square: 0.8640284793457645


Test Dataset:
----------------------------------
Lasso Regression: Evaluation metrics
----------------------------------
MSE: 0.024209052259580997
RMSE: 0.15559258420497102
R2 Square: 0.8531135806879446
```

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

Important top 5 Predictors Ridge Regression (alpha=0.3)

Features	Coefficient
OverallQual_Poor	-0.483466
HeatingQC_Po	-0.261457
OverallQual_Excellent	0.244520
OverallCond_Fair	-0.222608
Neighborhood_Crawfor	0.217257

Important top 5 Predictors Ridge Regression (alpha=0.6)

Features	Coefficient
OverallQual_Poor	-0.422018
OverallQual_Excellent	0.241445
OverallCond_Fair	-0.222134
Neighborhood_Crawfor	0.214584
HeatingQC_Po	-0.211291

Important top 5 Predictors Lasso Regression (alpha=0.001)

Features	Coefficient
OverallQual_Excellent	0.252778
OverallCond_Fair	-0.216573
Neighborhood_Crawfor	0.175040
OverallQual_Very Good	0.155333
GrLivArea	0.154402

Important top 5 Predictors Lasso Regression (alpha=0.002)

Features	Coefficient
OverallQual_Excellent	0.221404
OverallCond_Fair	-0.164484
GrLivArea	0.158397
OverallQual_Very Good	0.146679
Neighborhood_Crawfor	0.141567

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 - 0.3
    - Lasso - 0.001
    -

- The Mean Squared error in case of Ridge and Lasso on both Train and Test set are:

    - Ridge (MSE on Train) -> 0.015256828673062094
    - Ridge (MSE on Test) -> 0.021362169298106772

    - Lasso (MSE on Train) -> 0.018552541023282743
    - Lasso (MSE on Test) -> 0.022156752401838172

- The R2 score in case of Ridge and Lasso on both Train and Test set are:

    - Ridge (R2 on Train) -> 0.9029120221348146
    - Ridge (R2 on Test) -> 0.8703868072450047

    - Lasso (R2 on Train) -> 0.8819395084778187
    - Lasso (R2 on Test) -> 0.8655657400796524

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**

Features	Coefficient
OverallQual_Excellent	0.252778
OverallCond_Fair	-0.216573
Neighborhood_Crawfor	0.175040
OverallQual_Very Good	0.155333
GrLivArea	0.154402

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

Features	Coefficient
BsmtQual_No Basement	-0.207192
Neighborhood_NridgHt	0.159540
KitchenQual_Fa	-0.140506
1stFlrSF	0.136783
2ndFlrSF	0.131410

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:

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.