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:

1. Optimal Value of **alpha** for RIDGE regression is 5
2. Optimal value of **alpha** for LASSO regression is 0.0006.
3. With the above values of alpha, **R2 Score** of the **RIDGE** regression model is **0.84**, whereas for **LASSO**, it is **0.86**. Hence, there is no significant difference in their R2 scores on test set.

After doubling the alpha values,

1. R2 score of the **RIDGE** regression model is **0.84**, whereas for **LASSO**, it is **0.85. (**Refer jupyter notebook file)
2. Hence, R2 score remaied the same for both LASSO and RIDGE regression models, after doubling their corresponding alpha values.

After the change is implemented, there are few dissimilarities in the **ridge** coefficients. The top-10 influencers to SalePrice as derived per **Ridge regression models** is nearly identical before and after doubling the alpha value. Below shows the changes in the coefficients. Refer notebook where these coefficients were learned as part of the training process.

	Ridge Coef
TotRmsAbvGrd	0.248
GarageArea	0.219
TotalBsmtSF	0.190
FullBath	0.171
Neighborhood_NoRidge	0.157
Neighborhood_StoneBr	0.137
Neighborhood_NridgHt	0.122
OverallCond	0.122
TotalPorchAreaSF	0.108
Neighborhood_Crawfor	0.104
Heating_GasW	0.103
BsmtExposure_Gd	0.101
FireplaceQu_Ex	0.097
CentralAir_Y	0.095
WoodDeckSF	0.087
KitchenQual_Ex	0.086
BsmtQual_Ex	0.086
Fireplaces	0.083
HouseStyle_2.5Unf	0.082
Alley_Pave	0.081

	Ridge Coef - DOUBLE ALPHA
TotRmsAbvGrd	0.218
GarageArea	0.184
FullBath	0.162
Neighborhood_NoRidge	0.138
TotalBsmtSF	0.135
Neighborhood_StoneBr	0.110
Neighborhood_NridgHt	0.103
OverallCond	0.095
Neighborhood_Crawfor	0.095
CentralAir_Y	0.094
BsmtExposure_Gd	0.091
FireplaceQu_Ex	0.089
BsmtQual_Ex	0.088
TotalPorchAreaSF	0.086
KitchenQual_Ex	0.086
Fireplaces	0.082
WoodDeckSF	0.080
Heating_GasW	0.074
BsmtFinType1_GLQ	0.071
Neighborhood_Somerst	0.066

Similarly, there are noticeable changes in the **lasso** coefficients. The top-10 influencers as derived per **Lasso regression** is approximately same before and after doubling the alpha value. Below show the change in the coefficients. Refer notebook where the coefficients were learned as part of the training process

	Lasso Coef
TotalBsmtSF	0.740
TotRmsAbvGrd	0.306
GarageArea	0.297
OverallCond	0.168
FullBath	0.149
Neighborhood_NoRidge	0.142
Neighborhood_StoneBr	0.137
Neighborhood_Crawfor	0.128
Neighborhood_NridgHt	0.112
CentralAir_Y	0.110
Fireplaces	0.099
TotalPorchAreaSF	0.095
WoodDeckSF	0.086
BsmtQual_Ex	0.081
Neighborhood_Somerst	0.075
Heating_GasW	0.073
KitchenQual_Ex	0.071
FireplaceQu_Ex	0.070
BsmtExposure_Gd	0.068
HouseStyle_2Story	0.062

	Lasso Coef - DOUBLE ALPHA
TotRmsAbvGrd	0.339
TotalBsmtSF	0.315
GarageArea	0.285
FullBath	0.179
Neighborhood_NoRidge	0.149
OverallCond	0.128
Neighborhood_Crawfor	0.119
Neighborhood_StoneBr	0.116
CentralAir_Y	0.110
Neighborhood_NridgHt	0.105
Fireplaces	0.091
BsmtQual_Ex	0.087
KitchenQual_Ex	0.077
TotalPorchAreaSF	0.076
WoodDeckSF	0.076
BsmtExposure_Gd	0.075
Neighborhood_Somerst	0.074
FireplaceQu_Ex	0.063
LotConfig_CulDSac	0.055
GarageCond_TA	0.054

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 best value for lambda for Ridge and Lasso regression models are 5 and 0.0006 respectively. Their mean square errors are also very small in magnitude.

- Mean Square Error for Ridge on test set = 0.023
- Mean Suare Error for Lasso on test set = 0.0198

Since the mean square errors are lower as we can see above and R2 score for both Lasso and Ridge regression models are approximately same, we can use any one of them. But since lasso penalize the coefficients to Zero values, helps in feature elimination; I may prefer lasso over ridge regression.

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:

The top five improtant predictor variables in lasso model are:

1. TotalBsmtSF
2. TotRmsAbvGrd
3. GarageArea
4. OverallCond
5. FullBath

Created a new lasso model after eliminating them from the incoming dataset (Refer jupyter notebook).

New set of top-5 influencers to SalePrice are:

- **Neighborhood_NoRidge**
- **LotArea**
- **Neighborhood_StoneBr**
- **TotalPorchAreaSF**
- **Neighborhood_NridgHt**

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:

Model overfitting is the phenomena when it learns not only the hidden insights of the data but also the noise. Such a model is senitive to the training set, meaning if we perform a small change in the training set, we can observe a drastic change in the model itself. Thus, we need a model which is simpler enough which should learn the core principles from the training examples, instead of just memorizing them which results into the overfitting problem.

Due to the multicolinearity between the predictors, the learning algorithm sees many predictor values that can lead it to explain too much of the variability of the target and hence the model tries to pass thorugh most of the training datapoints, result into overfiiting. Such a model becomes complex, flexible and often fails to produce correct inferences on unseen datapoints.

When it comes to using a model for inference on unseen datapoints, we talk about OCCUM's RAZOR.

- When most of the unseen datapoints are of similar kind as that of the training datapioints which the model has already seen at the time of its training process, then the expected value of the target by the estimated model could be very similar to the true value of the target given the unseen datapoints. So, a complex model that memorizes the training examples very well by making assumptions about the unseen datapoints which fortunately have similar variability, then it can produce inferences very near to the expected target. So error due to bias would be very low.

- But, the same complex model which has learned the training examples including the noise, would produce incorrect inferences on the unseen datapoints which are NOT of similar kind as that of the training datapoints. If such unseen points are large in number, then model's error due to variance would be very high.

- Hence when we are not sure about the randomness of the unseen instances (whether it changes continuously or periodically or remain consistent), we are in dilemma, meaning you are facing a trade-off between BIAS and the VARIANCE. When you are in dilemma, always choose a simpler model as per OCCUM's RAZOR, which will be more **generic** as it will likely to explain the trend of the unseen instances as it does for the training examples, provided that the unseen variability is similar to that of the training exmapes.

- On the same note, a simpler model is more **robust**, **generalizable**. Meaning, it can generalize the unseen data points, even if there are some which are of dissimilar nature as that of the training datapoints, resulting into low error due to variation.

- Simpler models makes less assumptions about the unseen or test instances and it will have a tendency to learn the core principles, so it can perform better while being used for inferences.