Problem Statement - Part II

Advance Regression

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.

Optimal value of alpha for ridge: 1

Optimal value of alpha for lasso: 0.0001

Below is the table representing the effect of doubling alpha value on the model:

	Alpha		**Double alpha**	
	R^2 Train	**R^2 Test**	**R^2 Train**	**R^2 Test**
Ridge	0.887	0.885	0.884	0.886
Lasso	0.882	0.885	0.875	0.887

The most important predictor variables after the change are:

```
betas.sort_values(by = 'Lasso_abs', ascending = False, inplace = True)
betas[['Lasso', 'Lasso_abs']].head(5)
```

	Lasso	**Lasso_abs**
GrLivArea	0.338902	0.338902
Neighborhood_StoneBr	0.090696	0.090696
BsmtFinSF1	0.088095	0.088095
KitchenQual	0.082369	0.082369
Neighborhood_NridgHt	0.076361	0.076361

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 optimum lambda value in case of Ridge and Lasso is as follows:-

- Ridge – 1
- Lasso – 0.0001

The R^2 score is almost similar on test data for both Lasso and Ridge. But, Lasso helps in reducing the features in the model, helping to create a simpler final model. This is important for creating a robust and generalisable model.

Ridge Regression:

In ridge regression technique, we used lambda function along with the shrinkage penalty as sum of squared model coefficient.

LASSO Regression:

In LASSO regression uses the lambda function with the sum of the absolute values of all the coefficient present in the model. The LASSO regression shrinks the coefficient estimates towards zero (0). in this method the penalty term pushes some of the coefficient estimates to be exactly to 0. given that the hyper-parameter tuning is large.

while performing LASSO regression, the model performs the feature selection and because of this the interpretation of the model generated by LASSO makes it easier as compared to model generated by Ridge.

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.

After deleting variables: 'GrLivArea', 'Neighborhood_StoneBr', 'BsmtFinSF1', 'KitchenQual', and 'Neighborhood_NridgHt' from the dataset. We've created a new model with the remaining variables and found that the following are the five most important variables:

```
betas.sort_values(by = 'Lasso_abs', ascending = False, inplace = True)
betas[['Lasso', 'Lasso_abs']].head(5)
```

	Lasso	Lasso_abs
TotalBsmtSF	0.418256	0.418256
GarageType_NA	0.189478	0.189478
Neighborhood_MeadowV	-0.156596	0.156596
Neighborhood_Edwards	-0.155831	0.155831
Neighborhood_OldTown	-0.149144	0.149144

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.

As Per, Occam's Razor— given two models that show similar 'performance' in the finite training or test data, we should pick the one that makes fewer on the test data due to following reasons:-

- Simpler models are usually more 'generic' and are more widely applicable
- Simpler models require fewer training samples for effective training than the more complex ones and hence are easier to train.
- Simpler models are more robust.
  - Complex models tend to change widely with changes in the training data set
  - Simple models have low variance, high bias and complex models have low bias, high variance
  - Simpler models make more errors in the training set. Complex models lead to over-fitting —they work very well for the training samples, fail miserably when applied to other test samples

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

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 simpler 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 quantifies how accurate is the model likely to be on test data. A complex model can do an accurate job prediction provided there is enough training data. Models that are too naïve, for e.g., one that gives same answer to all test inputs and makes no discrimination whatsoever has a very large bias as its expected error across all test inputs are very high. Variance refers to the degree of changes in the model itself with respect to changes in the training data. Thus accuracy of the model can be maintained by keeping the balance etween Bias and Variance as it minimizes the total error as shown in the below graph.