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 regression: **20**
Optimal value of alpha for ridge: **0.0001**

Doubling these values would mean;

- New value of alpha for ridge regression: 4**0**
- New value of alpha for ridge: **0.0002**

RIDGE REGRESSION:-

- ✓ ***Coeff values are increasing slightly as alpha increases.***
- ✓ ***r2_score for train data reduced slightly from 0.9141 to 0.9071***
- ✓ ***r2_score for test data reduced slightly from 0.8879 to 0.8860***

LASSO REGRESSION:-

- ✓ ***As alpha value increased more features removed from model.***
- ✓ ***r2_score dropped from 0.9466 to 0.9326 for train data.***
- ✓ ***r2_score dropped from 0.8773 to 0.8755 for train data.***

The top 5 features after the above change is implemented include:-

1. **MSSubClass**
2. **OverallCond**
3. **Neighborhood_Crawfor**
4. **BsmtFullBath**
5. **Neighborhood_NridgHt**

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 preferred choice between Ridge & Lasso would be ***Lasso*** purely due to feature selection option. Additionally, we can see that in Lasso, some of the model coefficients have become exactly 0 which is how it performs variable selection. Seemingly enough the model accuracy is not affected & it helps make the model generalized, simple, & easier to interpret (with fewer features).

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 5 features from the model:-

1. **MSSubClass**
2. **OverallCond**
3. **Neighborhood_Crawfor**
4. **BsmtFullBath**
5. **Neighborhood_NridgHt**

After dropping the above features, the new model's accuracy wasn't affected significantly as the drop was from **94.66%** to **94.18%** for trained data & **87.73%** to **86.98%** for test data. However, the top 5 features now include:-

1. **LotFrontage**
2. **Exterior1st_AsphShn**
3. **RoofMatl_Membran**
4. **RoofMatl_Metal**
5. **RoofMatl_Tar&Grv**

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:

When we say that a model is **robust**, it means it should be able to perform satisfactorily for any variations in the data, thereby, proving itself trustworthy to be used for predictive analysis. A model which is also **generalizable** will have the ability to adapt to any new, previously unseen data and is able to accurately predict the outcomes from whatever data that is thrown towards it during testing. To enable it to do this, we need to ensure that the model avoids **overfitting/underfitting** as much as possible. The model should not memorize the training data to blindly reproduce the same behaviour for the same set of data, but should rather identify underlying patterns/trends from unseen test data. To maintain optimum model accuracy, it needs to be tuned to a point where variance and bias both are within acceptable lower values, which could also help reduce complexity. Regularization techniques like Ridge and Lasso regression can be used for this purpose.