Problem Statement - Part 2

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 : We plot the curve between mean absolute error (negative) and alpha. Now, to get the optimal value for both as follows :

Ridge Regression : When the value of alpha is 2 the test error is minimum. Hence, alpha equal to 2 for our ridge regression.

Lasso Regression : The value here we take is 0.01. Since, the value of alpha increases, the model tries to make most of the coefficient value 0.

In both the cases when the value of alpha increases, the model does not perform well. For Ridge, we get more error in testing and training data. Moreover, for Lasso the R2 value decreases when we increase alpha.

Variables after the changes are implemented in Lasso :
>GrLivArea

>OverallQual

>OverallCond

>TotalBsmtSF

>BsmtFinSF1

>GarageArea

>Fireplaces

>LotArea

>LotFrontage

Variables after the changes are implemented in Ridge :
>MSZoning_FV

>MSZoning_RL
>Neighborhood_Crawfor
>MSZoning_RM
>MSZoning_RH
>GrLivArea
>SaleCondition_Partial
>Exterior1st_BrkFace
>Neighborhood_StoneBr
>SaleCondition_Normal

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 : It is better to use Lasso, since it brings and assigns a zero value to insignificant features, enabling us to choose the predictive variables. It is always advisable to use simple yet robust model.

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 5 most important variables are :
>GrLivArea
>OverallQual
>OverallCond
>TotalBsmtSF
>GarageArea

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 : Ensuring that a model is robust and generalizable is crucial for its real-world applicability and performance. A robust and generalizable model can make accurate predictions on new, unseen data and is less likely to be affected by noise, variations, and changes in the input data. Here are some strategies to achieve model robustness and generalizability:

1. Cross-Validation: Use techniques like k-fold cross-validation to assess the model's performance on multiple subsets of the data. This helps you gauge how well the model generalizes to different data partitions and prevents overfitting to a single data split.

2. Regularization: Apply regularization techniques like Ridge, Lasso, or Elastic Net to prevent overfitting. Regularization constrains the model's complexity and discourages it from fitting noise in the training data.

3. Feature Engineering: Select relevant features and remove irrelevant or redundant ones. Feature engineering helps to reduce noise and focus the model on the most informative aspects of the data.

4. Data Augmentation: Increase the size and diversity of your training data through techniques like data augmentation. This can help the model learn more robust and generalizable patterns by exposing it to variations in the data.

5. Ensemble Methods: Use ensemble techniques like Random Forests, Gradient Boosting, or Stacking. Ensembles combine predictions from multiple models, which can improve generalization by reducing individual model biases.

Implications for Model Accuracy:
- Training Accuracy vs. Test Accuracy: A model that is overfitting the training data might achieve high accuracy on the training set but lower accuracy on new, unseen test data. Ensuring a balance between training and test accuracy is important for generalization.

- Bias-Variance Trade-off: Model accuracy can be influenced by the bias-variance trade-off. High-bias models (underfitting) might have lower accuracy on both training and test data, while high-variance models (overfitting) might have high training accuracy but significantly lower test accuracy.

- Overfitting and Underfitting: Overfitting can lead to poor generalization, as the model memorizes noise in the training data. Underfitting, on the other hand, fails to capture relevant patterns and relationships, leading to suboptimal accuracy.

- Generalization Gap: The difference between training accuracy and test accuracy is often referred to as the generalization gap. A robust and generalizable model aims to minimize this gap, indicating that it performs well on unseen data.

In conclusion, achieving model robustness and generalizability involves a combination of techniques and strategies that aim to reduce overfitting, capture meaningful patterns, and adapt well to new data. The implications for model accuracy are that a well-generalized model should exhibit balanced accuracy across training and test data and be less susceptible to fluctuations caused by noise and variations in the input data.