Problem Statement

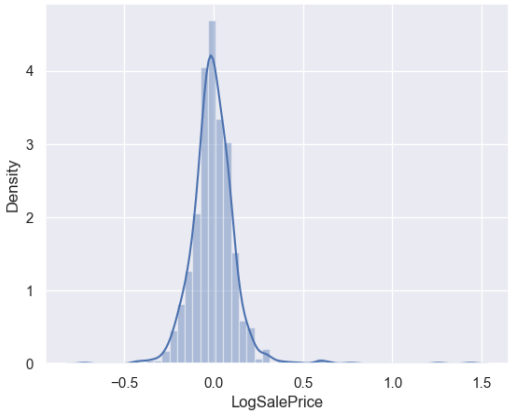
Question 1

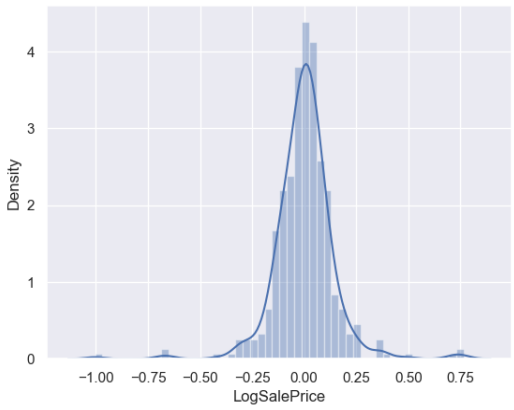
What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

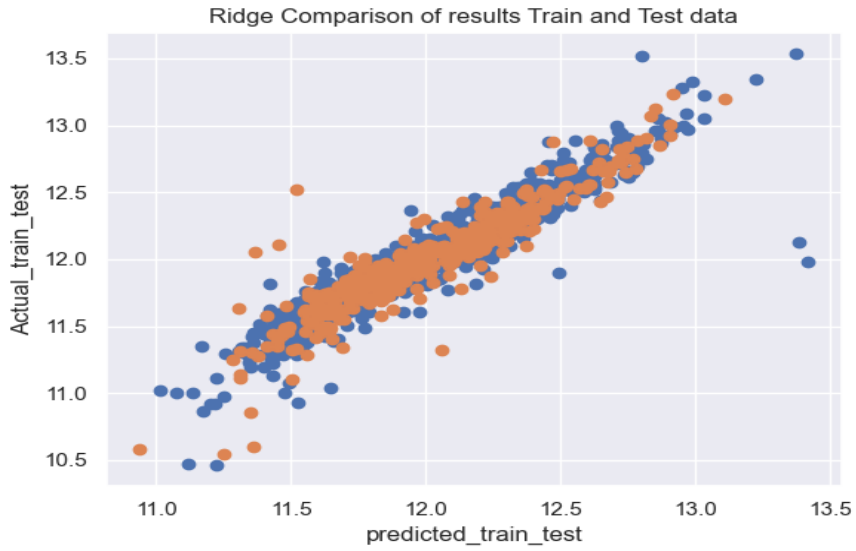
Ans: The optimal value of alpha for Ridge is 100 and Lasso is 0.01,

With Ridge R2_score=0.89(train) & 0.86(test)

RSS is Normally distributed for train and test

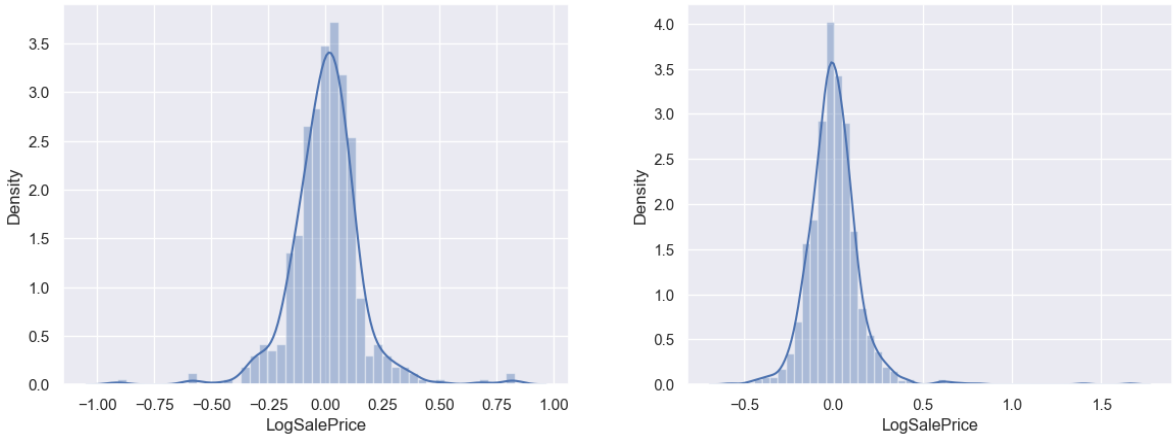




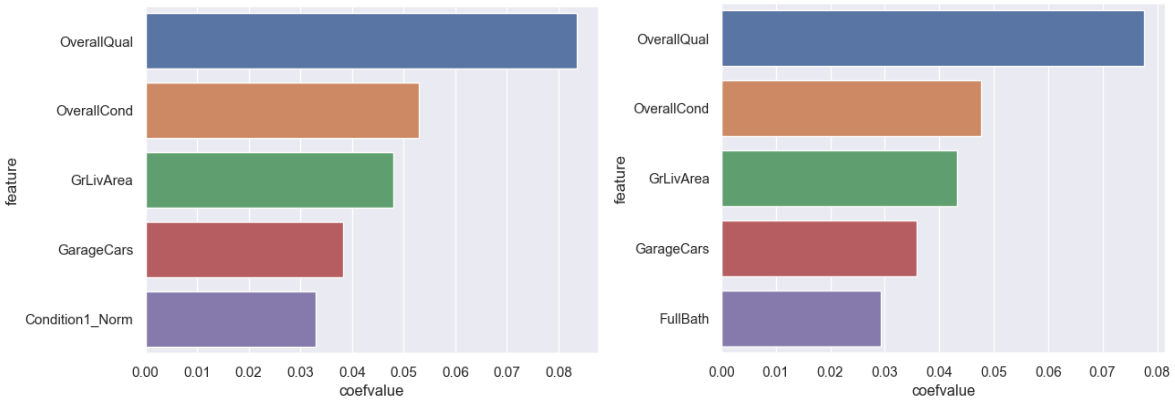


Lasso R2_Score= 0.86(train) and 0.84(test)

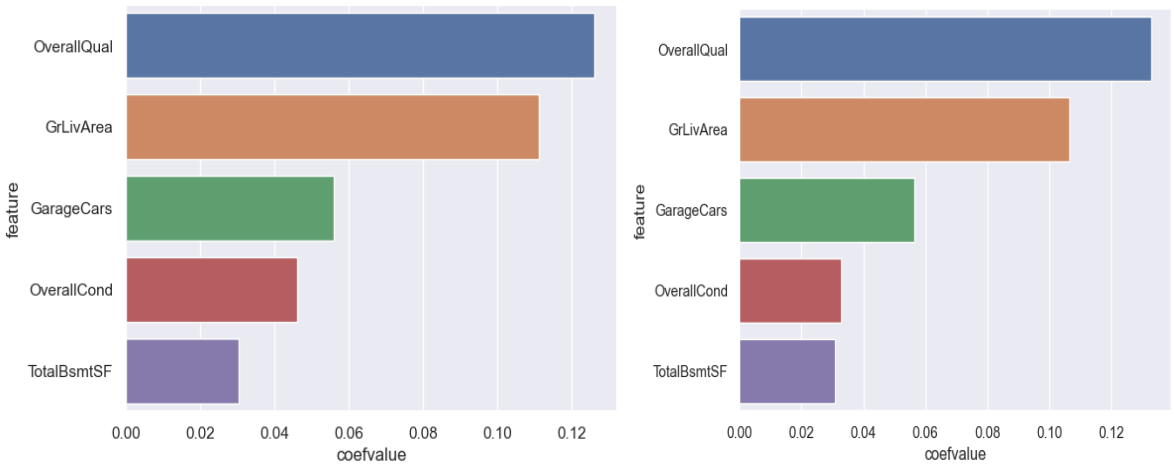
RSS for Lasso



If we double the value of alpha for ridge and lasso the variance increases where r2 value slowly goes on decreasing and in lasso the coefficients of important predictor variables increased but as the value of alpha increase all the predictor value tends to zero



Above images are of Ridge model important predictors



Above images are of Losso model important predictors

import predictor variable

Lasso : - OverallQual, GrLivArea, , GarageCars, OverallCond, TotalBsmtSF.

Ridge : - OverallQual, OverallCond, GrLivArea, GarageCars, FullBath.

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?

Ans: I choose Lasso because as Lasso gives me significant coefficients value and other remaining the tends zero and its RMSE value is okay with test data as compared to Ridge.

Lasso RMSE: Train- 0.011263837206319304, Test - 0.011723196243577996

Ridge RMSE: Train- 0.008574982914362565, Test-0.010485780656134683

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?

Ans: The other five important variables excluding the first important predictors are

1. MasVnrArea
2. HalfBath
3. 3SsnPorch
4. BsmtFinSF1
5. LowQualFinSF

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?

Ans: Ensuring that a model is robust and generalizable is crucial to its performance and reliability. A robust and generalizable model performs well on both the training data and unseen or new data.

Cross-Validation: Use techniques like k-fold cross-validation to evaluate your model's performance on different subsets of the data. Cross-validation helps ensure that the model generalizes well to different parts of the dataset.

Feature Engineering: Carefully select and engineer features to capture relevant information and remove noise. Feature selection techniques can help identify the most important variables and reduce the risk of overfitting.

Regularization: Apply regularization techniques such as Ridge, Lasso, or Elastic Net to prevent overfitting. Regularization adds a penalty for large coefficients, promoting a simpler model with better generalization.

Data Splitting: Split your data into training, validation, and test sets. Train your model on the training data, use the validation set to fine-tune hyperparameters, and evaluate the model's generalization on the test set.

Hyperparameter Tuning: Optimize model hyperparameters using techniques like grid search or random search. Proper hyperparameter tuning can significantly impact a model's ability to generalize.

Use Sufficient Data: Ensure that you have enough data to train and validate your model effectively. More data can help the model learn a wider range of patterns and improve generalization.

Regularly Update Models: Models can become less robust and less generalizable over time as the data distribution changes. Regularly update and retrain your models to keep them relevant.

Implications for Model Accuracy:

A robust and generalizable model may have a slightly lower accuracy on the training data compared to an overfit model because it avoids fitting to the noise in the data.

However, a robust and generalizable model is expected to have better accuracy on new, unseen data, which is the most important aspect of model performance. It can make reliable predictions on real-world data and handle variations and noise effectively.

The model should be as simple as possible, though its accuracy will decrease but it will be more robust and generalisable. It can be also understood using the Bias-Variance trade-off. The simpler the model the more the bias but less variance and more generalizable. Its implication in terms of accuracy is that a robust and generalisable model will perform equally well on both training and test data i.e. the accuracy does not change much for training and test data.

Bias: Bias is error in model, when the model is weak to learn from the data. High bias means model is unable to learn details in the data. Model performs poor on training and testing data.

Variance: Variance is error in model, when model tries to over learn from the data. High variance means model performs exceptionally well on training data as it has very well trained on this of data but performs very poor on testing data as it was unseen data for the model. It is important to have balance in Bias and Variance to avoid overfitting and under-fitting of data.