Q1. 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: Optimal value of alpha for Ridge and Lasso are 5.4 and 0.0008 respectively.

If we choose to double the value of alpha for both ridge and lasso, the model tends to underfit comparatively. The r squared value for train and test data decreases from 91 & 88 to 90 & 87 respectively. If we look into the RSS and RMSE values, that increases from 8.61 to 8.87 and 0.12 to 0.13 for train data.

GrLivArea is the most important predictor variables after the change is implemented. This feature having coefficient value of 0.587332.

Q2. 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: Lasso is the better model proven in this assignment. Because, the r square values for both train and test data is higher in lasso (0.90 & 0.89 respectively) compared to ridge(0.90 & 0.88 respectively) with minimal penalty of (0.0008). Also, the RMSE for lasso model (0.12 & 0.13) is lesser than that of ridge(0.12 & 0.14).

Q3. 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. After excluding the five most important predictor variables, created new lasso model with penalty 0.003 having r square value of 0.87 & 0.84 on train and test data respectively. With this new model, now the most important predictor variables are 1stFlrSF, 2ndFlrSF, BsmtFinSF1, GarageCars, YearRemodAdd.

Q4. 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 can be ensured by following the below processes -

Feature Engineering: Thoughtful feature engineering can help extract relevant information from the data and reduce noise, making the model more robust to variations in input.

Regularization Techniques: Regularization methods like ridge and lasso help prevent overfitting, which can improve the model's ability to generalize to unseen data.

Cross-Validation: Cross-validation techniques, such as k-fold cross-validation, help assess the model's performance on multiple subsets of the data, providing a more reliable estimate of its generalization ability.

Hyperparameter Tuning: Proper tuning of hyperparameters helps optimize the model's performance while preventing overfitting, contributing to its robustness.

Model Evaluation: Robust models are evaluated using appropriate metrics on separate validation or test datasets to assess their performance accurately.

Implications for Accuracy:

Real-world Performance: While achieving high accuracy on the training data is essential, the ultimate goal of a machine learning model is to perform well on unseen, real-world data. A model that is robust and generalizable is more likely to maintain its performance when deployed in production environments where it encounters diverse inputs and conditions.

Bias-Variance Tradeoff: Model accuracy is affected by the bias-variance tradeoff. A model with low bias (closely fitting the training data) but high variance (sensitive to small fluctuations in the training data) might have high accuracy on the training set but could perform poorly on unseen data due to overfitting. Techniques like regularization and cross-validation aim to balance bias and variance, improving the model's generalization without sacrificing too much accuracy.

Data Quality and Distribution Shift: Ensuring robustness and generalizability often involves training the model on high-quality, diverse datasets that accurately represent the distribution of real-world data. This can prevent issues related to dataset biases or distribution shifts, which might otherwise lead to decreased accuracy when the model encounters data from different sources or environments.