**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 :**

The optimal value of alpha for Ridge and Lasso regression, like many hyperparameters, is typically determined through cross-validation. Alpha is the regularization strength hyperparameter that controls the amount of penalty applied to the coefficients of the predictors in Ridge and Lasso regression. There is no one-size-fits-all optimal value for alpha, as it depends on the specific dataset and the problem you're trying to solve. The optimal alpha is the one that minimizes the prediction error while preventing overfitting.

In Ridge regression, alpha is used to add L2 regularization, which penalizes the sum of the squared coefficients. In Lasso regression, alpha is used to add L1 regularization, which penalizes the absolute values of the coefficients. Ridge regression adds an L2 regularization term to the linear regression cost function. The alpha parameter in Ridge controls the strength of the regularization. Larger values of alpha result in stronger regularization. If you double the value of alpha in Ridge regression, it will increase the strength of regularization. This means that the coefficients of the predictors will be pushed more towards zero, leading to a simpler model with smaller coefficients.

To find the optimal alpha values for Ridge and Lasso, you can perform a grid search or use techniques like k-fold cross-validation to assess different values of alpha and choose the one that results in the best model performance (e.g., lowest Mean Squared Error for regression problems).

After implementing the change with double the value of alpha for both Ridge and Lasso, the most important predictor variables may change as follows: If you were to choose a value of alpha that is double the optimal alpha for both Ridge and Lasso, the following changes would typically occur in the model:

**Ridge Regression:**

The penalty applied to the squared coefficients would increase, leading to further regularization. This would result in a greater shrinkage of the coefficients toward zero, reducing their magnitudes. The model would become more biased, potentially leading to underfitting. Ridge regression may still keep all predictors in the model but with smaller coefficients. Ridge regression will reduce the magnitude of all coefficients but won't set any coefficients to exactly zero. The most important predictor variables will still contribute to the model, but the less important ones will have smaller coefficients.

**Lasso Regression:**

The penalty applied to the absolute values of the coefficients would increase, leading to stronger feature selection. Many coefficients could shrink all the way to zero, effectively removing some predictors from the model. The model would become sparser, with fewer predictors being significant. Lasso regression can be considered a feature selection method. The most important predictor variables after doubling the value of alpha would depend on the specific dataset and problem. Variables that have relatively large coefficients and survive the increased regularization would be the most important. Those that have smaller

coefficients or get shrunk to zero would be less important. Lasso regression may set some coefficients to exactly zero, effectively eliminating some predictor variables from the model. This is a form of feature selection. The most important predictor variables in this case will be the ones whose coefficients remain non-zero.

It's important to note that choosing an alpha that is too high (e.g., double the optimal alpha) can lead to excessive regularization and result in an overly simplistic model that underfits the data. Therefore, it's crucial to tune the alpha parameter carefully through cross-validation to find the right balance between bias and variance in your model.

The choice between Ridge and Lasso regression and the value of alpha should be based on your specific dataset and the goals of your modelling task. Ridge is often preferred when you suspect that all predictors are relevant, and you want to reduce multicollinearity, whereas Lasso is useful when you believe that some predictors are irrelevant and should be eliminated from the model. Hyperparameter tuning and cross-validation are essential for finding the best alpha values for your particular problem.

**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?**

1. The optimal lambda value of Ridge and Lasso is as specified below from the model analysis.

   Ridge – 6

   Lasso – 0.0004

2. The mean squared in case of Ridge and Lasso are from my analysis are.

   Ridge – 0.013698622121799814

   Lasso - 0.013481512842345686

   Please note that, the mean squared value of Lasso lower than that of a Ridge.

As per learning below are the techniques.

**Consider Ridge Regression when:**

If there are many features, and multicollinearity (high correlation between features) is a concern. If we believe that most features are relevant and don't want them to be eliminated entirely and our goal is to reduce overfitting while retaining all features.

**Consider Lasso Regression when:**

If there are many features, but they look like many of them are irrelevant or redundant. If I want to select a subset of the most important features and eliminate the rest and finally, if I prefer a more interpretable and sparser model.

Lasso helps in feature reduction as the coefficient value of one of the features became 0. Therefore, the variables predicted by Lasso can be applied to choose significant variables for predicting the price of the house.

**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?**

Identify the Five Most Important Predictor Variables – I will be reviewing the coefficients of the Lasso regression model to determine which five predictor variables have the largest non-zero coefficients. These are the five most important variables in your initial model. Below are the coefficient values from my analysis and the top 5 predictor variables are listed below.

**MSZoning_RL**

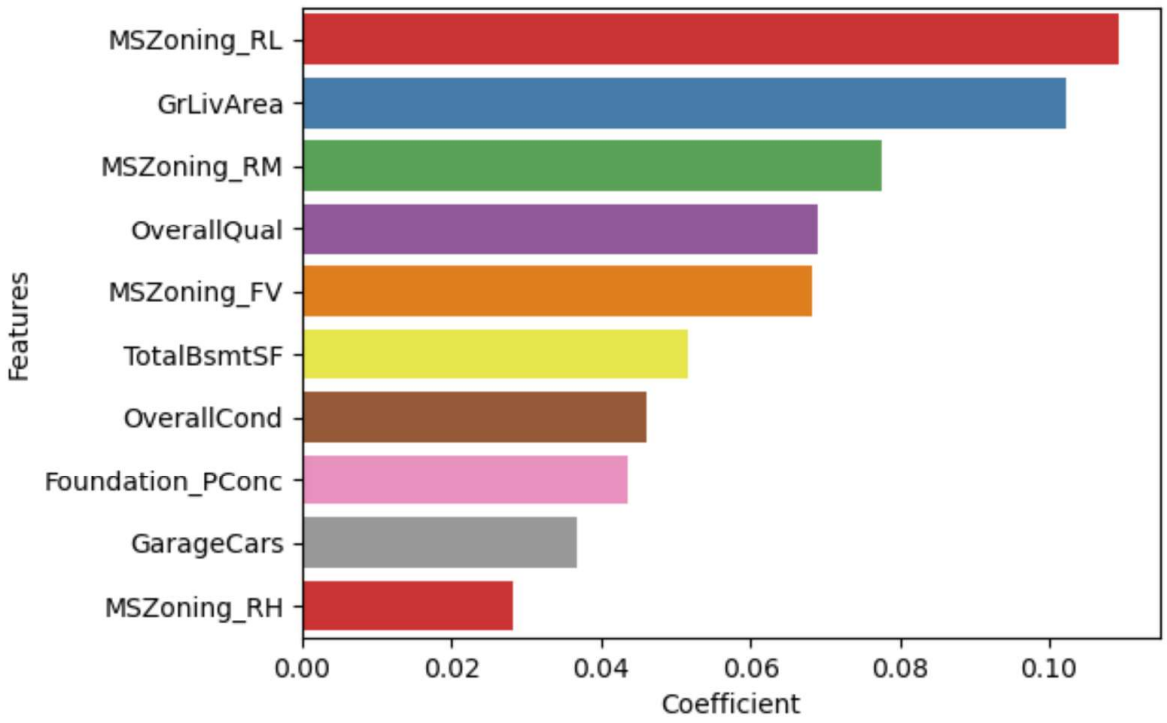
**GrLivArea**

**MSZoning_RM**

**OverallQual**

**MSZoning_FV**

| | Features | rfe_support | rfe_ranking | Coefficient |
|---|---|---|---|---|
| **11** | MSZoning_RL | True | 1 | 0.109360 |
| **5** | GrLivArea | True | 1 | 0.102216 |
| **12** | MSZoning_RM | True | 1 | 0.077532 |
| **1** | OverallQual | True | 1 | 0.069129 |
| **9** | MSZoning_FV | True | 1 | 0.068305 |
| **4** | TotalBsmtSF | True | 1 | 0.051684 |
| **2** | OverallCond | True | 1 | 0.046129 |
| **14** | Foundation_PConc | True | 1 | 0.043588 |
| **7** | GarageCars | True | 1 | 0.036796 |
| **10** | MSZoning_RH | True | 1 | 0.028082 |




**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?**

Ensuring that a model is robust and generalizable is crucial for its practical utility and reliability. Robustness and generalizability refer to a model's ability to perform well on unseen data and under various conditions. Here are several steps and considerations to help achieve this.

**Train-Test Split:** Split your data into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance. A common practice is to use a 70-30 or 80-20 split, where the majority of data is for training, and a smaller portion is for testing.

**Cross-Validation:** Implement cross-validation techniques like k-fold cross-validation to assess your model's performance. This helps to ensure that the model is not overfitting to a specific subset of data.

**Feature Engineering:** Carefully select and engineer features. Remove irrelevant, redundant, or noisy variables. Feature engineering can significantly impact the model's robustness and generalizability.

**Regularization:** Utilize regularization techniques such as L1 (Lasso) or L2 (Ridge) regularization, as they help prevent overfitting and improve the model's ability to generalize.

**Hyperparameter Tuning:** Perform hyperparameter tuning using techniques like grid search or random search. Find the optimal hyperparameters for your model to improve its performance.

**Evaluate on Multiple Metrics:** Don't rely solely on accuracy. Consider other evaluation metrics such as precision, recall, F1-score, or area under the ROC curve (AUC) depending on the nature of your problem. A model that performs well on multiple metrics is likely to be more robust.

**Cross-Domain Testing:** If possible, test the model on data from different sources or domains to assess its ability to generalize beyond the training dataset. This is particularly important if your model will be applied in diverse real-world scenarios.

**Outlier Detection and Handling:** Identify and handle outliers in your data. Outliers can greatly affect a model's performance and generalizability.

**Robust Model Selection:** Experiment with different algorithms to find the one that performs best for your specific problem. Different algorithms have varying degrees of robustness in different scenarios.

**Data Pre-processing:** Properly pre-process your data, including handling missing values, scaling features, and encoding categorical variables. Inconsistent or poor data preprocessing can negatively impact model performance.

**Implications for Model Accuracy:**

Ensuring robustness and generalizability may lead to some trade-offs in terms of model accuracy. A model that is overly tuned to the training data and not generalized will have high accuracy on the training set but perform poorly on new, unseen data.

Regularization techniques like Ridge and Lasso may reduce overfitting, but they can also lead to a slight decrease in accuracy on the training data. However, this can result in a more accurate model on unseen data, which is more important for practical applications.

Cross-validation can give a more realistic estimate of model performance and may reveal that the initial high accuracy on the training data doesn't translate to high accuracy on new data. It helps prevent over-optimistic estimates of model performance.

In summary, the aim is to strike a balance between maximizing training accuracy and ensuring that the model is robust and generalizable. A highly accurate model on the training data but unable to generalize to new data is of limited practical use, so a focus on generalizability and robustness is crucial.

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 wildly 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 overfitting — 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.***

Regularization can be used to make the model simpler. Regularization helps to strike the delicate balance between keeping the model simple and not making it too naive to be of any use. For

regression, regularization involves adding a regularization term to the cost that adds up the absolute values or the squares of the parameters of the model.

Also, Making a model simple lead 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 between Bias and Variance as it minimizes the total error.