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 to double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Optimal Value of Alpha for Ridge – 6
Optimal Value of Alpha for Lasso – 0.001

After doubling the alpha value in the assignment model, I observed the below:
In Ridge, the R2 didn't change much, but the RSS increased.
In Lasso, the R2 dropped from 0.91 to 0.89 on test data as well as an increase in the RSS value.

Top 5 features after the change:

Ridge:
GrLivArea
OverallQual_8
Neighborhood_Crawfor
OverallQual_9
Functional_Typ

Lasso:
GrLivArea
OverallQual_8
Neighborhood_Crawfor
Functional_Typ
OverallQual_9

Note: Please check for the results in the assignment ipython notebook.

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?

Ridge Regression:

- Ridge Regression adds the squared magnitude of coefficients as a penalty term to the loss function.
- It can shrink coefficients towards zero, but they never reach exactly zero.
- It is useful when you suspect that many features contribute to the outcome, and you want to control multicollinearity (high correlation between features).
- Ridge can provide a more stable solution when dealing with a dataset with a high number of correlated features.

Lasso Regression:
- Lasso Regression adds the absolute value of coefficients as a penalty term to the loss function.
- It can shrink coefficients all the way to zero, effectively performing feature selection.
- It is suitable when you believe that only a subset of features is relevant to the outcome.
- Lasso can help you identify the most important features in your model.

Choosing Between Them:
- If you have a strong reason to believe that only a few features are relevant, and you want a model that can eliminate less important features, Lasso Regression might be preferred.
- If you suspect multicollinearity among features and want to control their impact without necessarily removing them, Ridge Regression could be a better choice.

I will apply Lasso Regression as the R2 between Lasso and Ridge on test data is almost same and Lasso also helped in eliminating less important 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?

After removing the five most important predictor variables in lasso model, the below are the top predictor variables:
2ndFlrSF
Exterior1st_BrkFace
1stFlrSF
TotalBsmtSF
MSSubClass_70

Note: Please check for the results in the assignment ipython notebook.

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?

Steps to Make a Model Robust and Generalizable:

1. **Train-Test Split:** Split your dataset into training and testing sets. This allows you to train the model on one subset and evaluate its performance on an unseen subset.

2. **Cross-Validation:** Implement techniques like k-fold cross-validation to validate the model's performance across different subsets of the data. This helps assess how well the model generalizes to new data.

3. **Feature Engineering:** Select relevant features and preprocess them appropriately. Avoid overfitting by not using too many features, which can lead to poor generalization.

4. **Regularization:** Apply regularization techniques like Ridge or Lasso Regression to control overfitting and improve generalization.

5. **Hyperparameter Tuning:** Optimize hyperparameters through methods like grid search or random search to find the best settings for your model.

6. **Outlier Handling:** Identify and handle outliers in the data, which can significantly impact model performance.

7. **Ensemble Methods:** Consider using ensemble techniques like Random Forest or Gradient Boosting, which combine multiple models to improve robustness.

Implications for Model Accuracy:

1. **Overfitting:** A model that performs well on training data but poorly on test data is likely overfitting. Overfitting leads to high accuracy on known data but poor performance on new data. Ensuring robustness helps mitigate overfitting.

2. **Bias-Variance Trade-off:** A model with high variance can capture noise in the training data and may not generalize well. A balanced model, achieved through robustness, has lower variance and better generalization.

3. **Accuracy and Generalization:** A model's accuracy on training data doesn't necessarily guarantee its accuracy on new data. A model's generalization ability, assessed through test data or cross-validation, is a better indicator of real-world performance.

4. **Reduced Model Complexity:** A more robust model often has a simplified structure, reducing complexity and making it less prone to capturing noise in the data.

In essence, making a model robust and generalizable involves finding a balance between fitting the training data well and ensuring that the model's performance extends to unseen data. While a model's accuracy on training data might be high, its true test comes from how well it performs on new, real-world data. Prioritizing generalization is essential for a model's reliability in practical applications.