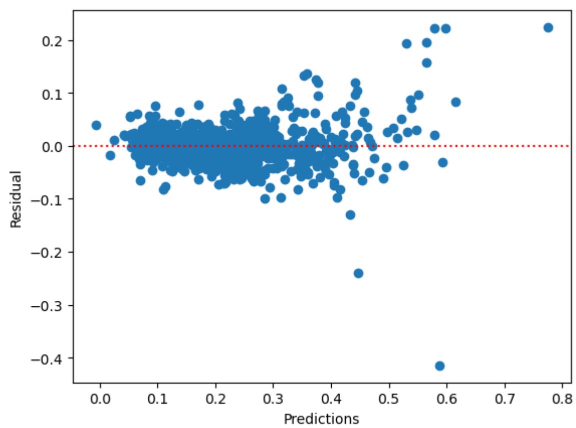
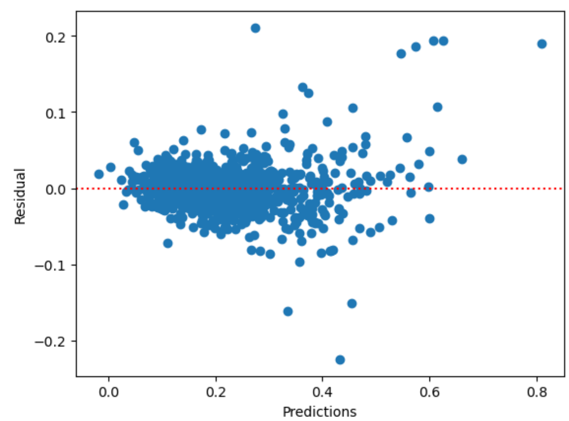
Question 1

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 the alpha in Lasso regression is 0.0001 and in Ridge regression is 0.05. The values of square of test has fallen from 0.732059793757553 to 0.5516543452666106 in ridge and from 0.837053204502147 to 0.8381385401041733 in lasso. The most important variable is GrLivArea.





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:

I chose the Lasso regression with lambda of 0.0001, because it is giving me the best r square values for train and test as follows,

train R-squared: 0.8926808740751461 &

test R square value: 0.837053204502147.

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:
OverallQual, GrLivArea, RoofMatl_WdShngl, TotRmsAbvGrd, Neighborhood_NoRidge are the top 5 most important columns.

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 effectiveness in making accurate predictions on new, unseen data. Here are some key practices to enhance the robustness and generalizability of a model:

Cross-Validation:

Use techniques like k-fold cross-validation to assess how well the model performs on different subsets of the data. This helps ensure that the model's performance is consistent across various data partitions and reduces the risk of overfitting to a specific dataset.
Train-Test Split:

Split your dataset into separate training and testing sets. Train the model on the training set and evaluate its performance on the testing set. This simulates the model's ability to generalize to new, unseen data.
Validation Set:

In addition to a training set and a test set, consider using a validation set during the model development process. This set can be used to fine-tune hyperparameters without contaminating the test set.
Feature Engineering:

Carefully engineer and select features. Remove irrelevant or redundant features, and ensure that the selected features are representative of the underlying patterns in the data. This helps prevent the model from fitting noise in the data.
Regularization:

Apply regularization techniques like Ridge or Lasso regression to prevent overfitting. Regularization helps control the complexity of the model and reduces the risk of fitting the training data too closely.
Data Augmentation:

In the case of image or text data, consider data augmentation techniques. This involves creating new training examples by applying random transformations to the existing data, making the model more robust to variations.
Hyperparameter Tuning:

Systematically tune hyperparameters using techniques like grid search or randomized search. Optimal hyperparameters can vary across datasets, and tuning helps find the best configuration for your specific problem.
Ensemble Methods:

Consider using ensemble methods like random forests or gradient boosting. These methods combine multiple models to improve overall predictive performance and can enhance robustness.
Implications for Accuracy:

High Training Accuracy vs. Generalization:

A model with very high training accuracy may not necessarily generalize well to new data. It might be overfitting to the training set, capturing noise or specific patterns that are not representative of the broader data distribution.
Balancing Bias and Variance:

Achieving high accuracy on the training set while maintaining good performance on the test set requires finding the right balance between bias and variance. Overly complex models can have low bias but high variance, leading to poor generalization.
Robustness in the Face of Perturbations:

A robust model should perform well even when faced with small changes or perturbations in the input data. If the model is overly sensitive to slight variations, its generalizability may be compromised.
In summary, robust and generalizable models strike a balance between fitting the training data well and being able to make accurate predictions on new, unseen data. Rigorous evaluation, thoughtful feature engineering, and careful model selection contribute to building models that can perform well in real-world scenarios.