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?

Ans:

Optimal Value for Ridge: 3.0
Optimal Value for Lasso: 0.001

We can see that if we increase the alpha it applies a penalty more on the coefficients.

- In Ridge regression, increasing alpha leads to smaller coefficient values but does not eliminate any coefficients.
- In Lasso regression, increasing alpha can lead to some coefficients being exactly zero, effectively eliminating the corresponding features from the model.

Top 10 Feature:

	Lasso (alpha=0.002)
GrLivArea	0.13512166926295166
OverallQual	0.09967806458927685
TotalBsmtSF	0.08313929218280948
YearBuilt	0.05534538537096426
OverallCond	0.046194262255863296
LotArea	0.03543908379229681
GarageArea	0.035186032736891505
YearRemodAdd	0.03098013814533002
Functional_Typ	0.028122778267624986
MSZoning_RL	0.02703167562965122

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: Both lasso and ridge has same value for R-square and RMSE. However Lasso seems more simple due to less number of predictor variable. SO we will be using Lasso model as our final model.

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?

	Lasso
MSZoning_RL	0.22802546790487363
GarageArea	0.16940686604267913
MSZoning_RM	0.13388720643245214
MSZoning_FV	0.09185784151344659
YearRemodAdd	0.08384179160371191

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 Model Robustness and Generalizability

- Train-Test Split: Divide the dataset into training and testing sets to evaluate the model's performance on unseen data.
- Cross-Validation: Employ techniques like k-fold cross-validation to assess model performance across multiple data subsets, reducing the risk of overfitting.
- Feature Selection and Engineering: Choose relevant features and perform engineering to improve the model's ability to generalize.
- Regularization: Utilize techniques such as Ridge and Lasso regression to prevent overfitting and encourage simpler, more generalizable models.
- Hyperparameter Tuning: Optimize model hyperparameters to strike a balance between bias and variance, leading to a more robust model.
- Outlier Detection and Handling: Identify and address outliers in the data to ensure they do not disproportionately influence model performance.
- Model Evaluation Metrics: Select appropriate evaluation metrics for the problem at hand to assess model performance accurately.
- Ensemble Methods: Consider employing ensemble methods like Random Forests or Gradient Boosting to improve generalization and robustness.

Implications for Model Accuracy:

- Prioritizing robustness and generalizability may require sacrificing a small amount of accuracy on the training data.
- Techniques like regularization constrain model flexibility to enhance generalization, potentially reducing accuracy on the training set.

- However, models that generalize well to unseen data are more valuable in real-world applications.
- Focus on building models that perform consistently well across various datasets and scenarios rather than merely memorizing training data.