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 variablesafter the change is implemented?

Ans: The Ridge regression model achieved its best performance with an optimal alpha value of 2, while Lasso performed optimally with an alpha of 0.001. These alpha settings resulted in an R-squared (R2) value of approximately 0.83, indicating a good fit.

Upon doubling the alpha values for both Ridge and Lasso, the prediction accuracy remains stable at around 0.82, with only minor adjustments in the coefficient values. This refined model is presented and illustrated in a Jupyter notebook. Below, we outline the modifications in the coefficients.

Ridge Regression Model

	Ridge Co-Efficient
Total_sqr_footage	0.169122
GarageArea	0.101585
TotRmsAbvGrd	0.067348
OverallCond	0.047652
LotArea	0.043941
CentralAir_Y	0.032034
LotFrontage	0.031772
Total_porch_sf	0.031639
Neighborhood_StoneBr	0.029093
Alley_Pave	0.024270
OpenPorchSF	0.023148
MSSubClass_70	0.022995
RoofMatl_WdShngl	0.022586
Neighborhood_Veenker	0.022410
SaleType_Con	0.022293
HouseStyle_2.5Unf	0.021873
PavedDrive_P	0.020160
KitchenQual_Ex	0.019378
LandContour_HLS	0.018595
SaleType_Oth	0.018123

	Ridge Doubled Alpha Co-Efficient
Total_sqr_footage	0.149028
GarageArea	0.091803
TotRmsAbvGrd	0.068283
OverallCond	0.043303
LotArea	0.038824
Total_porch_sf	0.033870
CentralAir_Y	0.031832
LotFrontage	0.027526
Neighborhood_StoneBr	0.026581
OpenPorchSF	0.022713
MSSubClass_70	0.022189
Alley_Pave	0.021672
Neighborhood_Veenker	0.020098
BsmtQual_Ex	0.019949
KitchenQual_Ex	0.019787
HouseStyle_2.5Unf	0.018952
MasVnrType_Stone	0.018388
PavedDrive_P	0.017973
RoofMatl_WdShngl	0.017856
PavedDrive_Y	0.016840

Lasso Regression Model

	Lasso Co-Efficient
Total_sqr_footage	0.202244
GarageArea	0.110863
TotRmsAbvGrd	0.063161
OverallCond	0.046686
LotArea	0.044597
CentralAir_Y	0.033294
Total_porch_sf	0.028923
Neighborhood_StoneBr	0.023370
Alley_Pave	0.020848
OpenPorchSF	0.020776
MSSubClass_70	0.018898
LandContour_HLS	0.017279
KitchenQual_Ex	0.016795
BsmtQual_Ex	0.016710
Condition1_Norm	0.015551
Neighborhood_Veenker	0.014707
MasVnrType_Stone	0.014389
PavedDrive_P	0.013578
LotFrontage	0.013377
PavedDrive_Y	0.012363

	Lasso Doubled Alpha Co-Efficient
Total_sqr_footage	0.204642
GarageArea	0.103822
TotRmsAbvGrd	0.064902
OverallCond	0.042168
CentralAir_Y	0.033113
Total_porch_sf	0.030659
LotArea	0.025909
BsmtQual_Ex	0.018128
Neighborhood_StoneBr	0.017152
Alley_Pave	0.016628
OpenPorchSF	0.016490
KitchenQual_Ex	0.016359
LandContour_HLS	0.014793
MSSubClass_70	0.014495
MasVnrType_Stone	0.013292
Condition1_Norm	0.012674
BsmtCond_TA	0.011677
SaleCondition_Partial	0.011236
LotConfig_CulDSac	0.008776
PavedDrive_Y	0.008685

Overall, since the alpha values are small, we do not see a huge change in the model after doubling the alpha.

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 to and why?

- The optimal lambda values for Ridge and Lasso are as follows:
  - Ridge: 2
  - Lasso: 0.0001
- The Mean Squared Error for Ridge and Lasso is calculated as follows:
  - Ridge: 0.0018396090787924262
  - Lasso: 0.0018634152629407766

Both models exhibit nearly identical Mean Squared Errors.

Given that Lasso facilitates feature reduction by driving some feature coefficients to zero, it offers a distinct advantage over Ridge. Therefore, Lasso should be selected as the final model for this analysis.

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?

Ans: The current Lasso model highlights the significance of the following five predictor variables:

1. Total_sqr_footage
2. GarageArea
3. TotRmsAbvGrd
4. OverallCond
5. LotArea

To further refine the model, we constructed a new Lasso model in a Jupyter notebook, excluding these top five attributes from the dataset. The R2 value for the updated model, without these critical predictors, decreases to 0.73. Simultaneously, the Mean Squared Error increases to 0.0028575670906482538.

The new set of top five predictors is as follows.

	Lasso Co-Efficient
LotFrontage	0.146535
Total_porch_sf	0.072445
HouseStyle_2.5Unf	0.062900
HouseStyle_2.5Fin	0.050487
Neighborhood_Veenker	0.042532

Question 4

How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

According to Occam's Razor, when we have two models that exhibit similar performance on finite training or test data, we should choose the one that generalizes better on test data. This principle is grounded in several compelling reasons:

1. Simpler models are typically more versatile and applicable to a broader range of scenarios.
2. Simpler models demand fewer training samples for effective training compared to their more complex counterparts, making them easier to work with.
3. Simpler models tend to be more robust in the face of changing training data. Complex models can exhibit significant variations in response to slight alterations in the dataset.
4. Simple models often possess low variance and high bias, whereas complex models have low bias and high variance. Striking the right balance between these two factors is essential.
5. Simpler models might make more errors during training, but they are less prone to overfitting. Overfit models excel on training data but falter on unseen test data.

To enhance model robustness and generalizability, it's vital to maintain simplicity without oversimplifying, as an excessively simplistic model may not yield useful results.

Regularization offers a means to achieve this balance. It involves introducing a regularization term in the cost function that accounts for the absolute values or squares of model parameters, helping to simplify the model without rendering it overly naive.

Additionally, the pursuit of model simplicity leads to the Bias-Variance Trade-off:

- Complex models exhibit instability and extreme sensitivity to even minor dataset changes.
- Simpler models, which capture essential data patterns, are less prone to wild fluctuations, even when the dataset grows or shrinks.

Bias quantifies a model's accuracy on test data. A highly complex model can provide accurate predictions if there is an ample training dataset. On the other hand, overly naive models, like those assigning the same output to all inputs, possess substantial bias as their expected error across various test inputs is quite high.
Variance relates to how much the model changes when the training data is altered. Hence, maintaining the right balance between bias and variance is crucial for optimizing model accuracy and minimizing total error, as depicted in the graph below.