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?

Solution: The optimal value of alpha for **Ridge** is **9.00** and for **Lasso** is **0.001.** When the alpha got doubled, for Lasso a greater number of Features coefficient turned out to be zero, **127 to 149,** R2 Score for Lasso for Training **reduced to 0.93 from 0.94** and the R2 for **Test** remained the same **0.91**, and for Ridge, Number of Features for which the coefficient turned out to be zero remained the same i.e. **8** and R2 for Training **reduced to 0.93 from 0.94** and the R2 for Test also **reduced to 0.9 from 0.91**. But even after the change Most important Predictor Turned out to be **Neighborhood** Features which means "Physical locations within Ames city limits" for the Category **Crawford.** So, for Lasso, when alpha increases more number of coefficient turns out to be zero, but for Ridge, increasing alpha leads to shrinkage of the coefficients and it helps to reduce the model complexity and multi-collinearity. One intuition can be made about the reduction is R2 Scores as alpha values is increasing, the model complexity reduces,

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?

Solution: Key Difference between **Lasso and Ridge** is Ridge considers all the features in the model and it does shrinkage of the coefficient and thus reduces Model Complexity, But, what Lasso does additionally is it performs feature selection as well, it makes the coefficients of some of the features to zero. Now in the current case prior to Model Building the Number of Features were more than **200.** And it was also found in case of **Lasso** the number of Features for which the coefficient turned out to be zero around **127**, but on ridge the number of features for which the coefficient turns out to be zero is very less.

Question 3: After building the model, you realized 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?

Solution: The five most important contributor are **'Neighborhood_Crawfor', 'GrLivArea', 'OverallQual_9.0', 'Neighborhood_StoneBr', 'KitchenQual_Ex',** so when these Features were removed, again a new Model was built. Using Cross Validation technique, the idea alpha value was identified, and it turned to be the same i.e. 0.001 for Lasso Regularization, so with the found alpha (learning parameter), model was build and the top five variables turned out to be

a. 2ndFlrSF: Second floor square feet
b. 1stFlrSF: First Floor square feet
c. Functional: Home functionality (for the category **Maj2**: Major Deduction 2)
d. MSSubClass: Identifies the type of dwelling involved in the sale (for the category **60**: 2-STORY 1946 & NEWER)
e. MSZoning: Identifies the general zoning classification of the sale (for the category C: Commercial)

<u>Question 4:</u> 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?

Solution: When we try to build the model, which is robust and generalizable, we tried different approach. Here we are considering the metrics **R2** and **MSE** for getting the ideal model.
So, we tried with simple **Linear Regression** model with all the features and we found that the

a. Training Dataset: R2 => 0.95 and MSE => 0.04
b. Test Dataset: R2 => 0.90 and MSE => 0.09

So, what we did first, we tried RFE, to select top 100 Features (To reduce computation time) and then apply Regularization technique, to ensure model is not so complex

	Ridge_RFE	Lasso_RFE
Alpha	4.00	0.001
Train_R2_Score	0.72	0.730
Train_MSE	0.20	0.200
Test_R2_Score	0.65	0.670
Test_MSE	0.26	0.250

It can be seen above the value of **R2 value** for Train and test Score is not so far, they are having **Δ** less than **15**, so they are quite near. But the problem here is **MSE** of **Train** and **Test**, is not so efficient.

To improve the performance, we considered all the Features and then using the Cross-Validation Technique the ideal Alpha value was identified, and then the regularization technique was applied.

	Ridge_RFE	Lasso_RFE	Ridge_All	Lasso_All
Alpha	4.00	0.001	9.00	0.001
Train_R2_Score	0.72	0.730	0.94	0.940
Train_MSE	0.20	0.200	0.06	0.050
Test_R2_Score	0.65	0.670	0.91	0.910
Test_MSE	0.26	0.250	0.08	0.080

It can be seen above **R2** improved a lot for Train and Test and we can also see that the MSE of Train and Train is quite less. And we can also see the MSE of Test is quite near to the Train Dataset. And it's quite small. But MSE might not be a good estimate of the test error, one reason can be the Dataset is not properly splitted into test and train i.e. different characteristic of the data points is not present in both the splits. So, what we tried next is, we ran the train test split without any random state, this step was repeated for 7 times and result was

Train R2 0.94	Train R2 0.94	Train R2 0.94	Train R2 0.94	Train R2 0.94	Train R2 0.94	Train R2 0.93
Train MSE 0.05	Train MSE 0.05	Train MSE 0.06	Train MSE 0.05	Train MSE 0.05	Train MSE 0.05	Train MSE 0.06
Test R2 0.91	Test R2 0.92	Test R2 0.93	Test R2 0.92	Test R2 0.91	Test R2 0.89	Test R2 0.92
Test MSE 0.09	Test MSE 0.07	Test MSE 0.07	Test MSE 0.08	Test MSE 0.08	Test MSE 0.1	Test MSE 0.07

So, as we can see, when the data splits differ for each run, but still Test MSE didn't differ so much from Train MSE. Hence it can be concluded model is robust and generalizable.