**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?

**Ans** - In the case of ridge regression:- When we plot the curve between negative mean absolute error

and alpha we see that as the value of alpha increase from 0 the error term decrease and the train error

is showing increasing trend when value of alpha increases .when the value of alpha is 10 the test error is

minimum and the r2 score is good, so we decided to go with value of alpha equal to 10 for our ridge regression.

For lasso regression I have decided to keep very small value that is 0.001, when we increase the value of

alpha the model try to penalize more and try to make most of the coefficient value zero as well as decreasing the r2 score and increasing RMSE.

When we double the value of alpha for our ridge regression no we will take the value of alpha equal to

20 the model will apply more penalty on the curve and try to make the model more generalized that is

making model more simpler and not thinking to fit every data point of the data set.

For Ridge,

when alpha = 20 : (double)

R2 Train: 0.9114291839272939

R2 Test: 0.8893823503547578

RSS Train: 14.214928873795795

RSS Test: 7.972055941655041

RMSE Train: 0.013922555214295588

RMSE Test: 0.01820104096268274

when alpha = 10 : (initial)

R2 Train: 0.9175252123077811

R2 Test: 0.8901721539803837

RSS Train: 13.23656360989067

RSS Test: 7.915135922954551

RMSE Train: 0.0129643130361319

RMSE Test: 0.018071086582088017

Similarly when we double the value of alpha for lasso we try to penalize more our model and more

coefficient of the variable will reduced to zero, when we increase the value of alpha our r2 square also

decreases.

when alpha = 0.002 (double)

R2 Train: 0.8892343980875083

R2 Test: 0.8819212973586529

RSS Train: 17.77701982057784

RSS Test: 8.50976336953258

RMSE Train: 0.017411380823288778

RMSE Test: 0.019428683492083515

when alpha = 0.001 : (initial)

R2 Train: 0.903521071830823

R2 Test: 0.887216954710663

RSS Train: 15.484119516513388

RSS Test: 8.128112911459613

RMSE Train: 0.015165641054371584

RMSE Test: 0.018557335414291356

The most important variable after the changes has been implemented for ridge regression are as

follows:-

1. MSZoning\_FV

2. MSZoning\_RL

3. Neighborhood\_Crawfor

4. MSZoning\_RH

5. MSZoning\_RM

6. SaleCondition\_Partial

7. Neighborhood\_StoneBr

8. GrLivArea

9. SaleCondition\_Normal

10. Exterior1st\_BrkFace

The most important variable after the changes has been implemented for lasso regression are as

follows:-

1. GrLivArea

2. OverallQual

3. OverallCond

4. TotalBsmtSF

5. BsmtFinSF1

6. GarageArea

7. Fireplaces

8. LotArea

9. LotArea

10. LotFrontage

**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?

**Ans** - Regularizing coefficients is crucial for enhancing prediction accuracy while simultaneously reducing variance and ensuring model interpretability.

In Ridge regression, the introduction of a tuning parameter, lambda, facilitates penalty imposition proportional to the square of coefficient magnitudes, determined via cross-validation. This penalty aims to minimize the residual sum of squares, effectively constraining coefficients with larger values. By adjusting lambda, variance diminishes while bias remains stable. Unlike Lasso Regression, Ridge regression incorporates all variables into the final model.

On the other hand, Lasso regression also utilizes lambda as a tuning parameter, applying penalties based on the absolute magnitude of coefficients determined through cross-validation. As lambda increases, Lasso regression progressively shrinks coefficients towards zero, potentially eliminating certain variables entirely. This process of shrinkage facilitates variable selection, wherein smaller lambda values resemble simple linear regression, while larger values discard variables by setting their coefficients to zero.

**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?

**Ans**- The five most important variable in the Lasso Regression with the co-efficients were:  
OverallQual: 0.10168335896065891

GrLivArea: 0.09402014606824509

Neighborhood\_Crawfor: 0.07315705635029879

Neighborhood\_NridgHt: 0.061871455182839534

Neighborhood\_Somerst: 0.06022500564584801

After removing these features, the following are the top 5 significant features in Lasso:

Condition2\_PosN: -0.19081949826577516

Neighborhood\_Edwards: -0.11664339725849084

2ndFlrSF: 0.10133235763430341

1stFlrSF: 0.0906859760193153

MSZoning\_FV: 0.08417406026313934

**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**:When building machine learning models, achieving optimal performance requires careful consideration of simplicity, robustness, and generalizability. This concept is often explained through the **Bias-Variance trade-off**.

Imagine a model as a flexible curve trying to fit a set of data points.

**High bias**: Think of a very rigid, straight line. This model has low variance (consistent predictions) but high bias (inability to capture the nuances of the data). It performs poorly on both training and testing data, struggling to learn from the complexities.

**High variance**: Now, imagine a highly curvy line that closely follows every single data point. This model has high variance (predictions can vary significantly) and low bias (captures intricate details). While it performs exceptionally well on training data (overfitting), it fails miserably on unseen data (testing data) as it memorized specific noise rather than general patterns.

T**he sweet spot**: The ideal model lies somewhere in between. It should be flexible enough to learn the important patterns from the data without getting bogged down by noise, resulting in moderate bias and variance. This leads to robustness and generalizability, meaning the model performs consistently well on both training and testing data.

