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:

Optimal value of lambda for Ridge Regression = 10
Optimal value of lambda for Lasso = 0.001

```
## Let us build the ridge regression model with double value of alpha i.e. 20
ridge = Ridge(alpha=20)

# Fit the model on training data
ridge.fit(X_train, y_train)
```

```
Ridge(alpha=20)
```

```
## Make predictions
y_train_pred = ridge.predict(X_train)
y_pred = ridge.predict(X_test)
```

```
## Check metrics
ridge_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R-Squared (Train) = 0.9341484010457403
R-Squared (Test) = 0.9276743387402124
RSS (Train) = 9.37431113166613
RSS (Test) = 2.8211986583505144
MSE (Train) = 0.008025951311357988
MSE (Test) = 0.009661639240926419
RMSE (Train) = 0.08958767388071859
RMSE (Test) = 0.09829363784562264
```

```
## Now we will build the lasso model with double value of alpha i.e. 0.002
lasso = Lasso(alpha=0.002)

# Fit the model on training data
lasso.fit(X_train, y_train)
```

```
Lasso(alpha=0.002)
```

```
## Make predictions
y_train_pred = lasso.predict(X_train)
y_pred = lasso.predict(X_test)
```

```
## Check metrics
lasso_metrics = show_metrics(y_train, y_train_pred, y_test, y_pred)
```

```
R-Squared (Train) = 0.9052352842244675
R-Squared (Test) = 0.9116378026982589
RSS (Train) = 13.490240845948152
RSS (Test) = 3.4467339549258558
MSE (Train) = 0.01154986373796931
MSE (Test) = 0.011803883407280329
RMSE (Train) = 0.10747029235081344
RMSE (Test) = 0.10864567827244823
```

```
# Again creating a table which contain all the metrics

lr_table = {'Metric': ['R2 Score (Train)','R2 Score (Test)','RSS (Train)','RSS (Test)',
                       'MSE (Train)','MSE (Test)', 'RMSE (Train)', 'RMSE (Test)'],
            'Ridge Regression' : ridge_metrics,
            'Lasso Regression' : lasso_metrics
        }

final_metric = pd.DataFrame(lr_table, columns = ['Metric', 'Ridge Regression', 'Lasso Regression'] )
final_metric.set_index('Metric')
```

Metric	Ridge Regression	Lasso Regression
R2 Score (Train)	0.93	0.91
R2 Score (Test)	0.93	0.91
RSS (Train)	9.37	13.49
RSS (Test)	2.82	3.45
MSE (Train)	0.01	0.01
MSE (Test)	0.01	0.01
RMSE (Train)	0.09	0.11
RMSE (Test)	0.10	0.11

Changes in Ridge Regression metrics:

- R2 score of train set decreased from 0.94 to 0.93
- R2 score of test set remained same at 0.93

Changes in Lasso metrics:

- R2 score of train set decreased from 0.92 to 0.91
- R2 score of test set decreased from 0.93 to 0.91

The most important predictor variables after we double the alpha values are:

- GrLivArea
- OverallQual_8
- OverallQual_9
- Functional_Typ
- Neighborhood_Crawfor
- Exterior1st_BrkFace
- TotalBsmtSF
- CentralAir_Y

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:

- The model we will choose to apply will depend on the use case.
- If we have too many variables and one of our primary goal is feature selection, then we will use “Lasso”.
- If we don't want to get too large coefficients and reduction of coefficient magnitude is one of our prime goals, then we will use “Ridge Regression”.

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:

After dropping our top 5 lasso predictors, we get the following new top 5 predictors:

- 2ndFlrSF
- Functional_Typ
- 1stFlrSF
- MSSubClass_70
- Neighborhood_Somerst

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:

- A model is robust when any variation in the data does not affect its performance much.
- A generalizable model is able to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model.
- To make sure a model is robust and generalizable, we have to take care it doesn't overfit. This is because an overfitting model has very high variance and a smallest change in data affects the model prediction heavily. Such a model will identify all the patterns of a training data, but fail to pick up the patterns in unseen test data.
- In other words, the model should not be too complex in order to be robust and generalizable.
- If we look at it from the perspective of Accuracy, a too complex model will have a very high accuracy. So, to make our model more robust and generalizable, we will have to decrease variance which will lead to some bias. Addition of bias means that accuracy will decrease.
- In general, we have to find strike some balance between model accuracy and complexity. This can be achieved by Regularization techniques like Ridge Regression and Lasso.