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?

- Optimal value of lambda for Ridge Regression = 4.0
- Optimal value of lambda for Lasso = **0.0001**

After doubling the values are below.

- • Optimal value of lambda for Ridge Regression = **8.0**
- • Optimal value of lambda for Lasso = 0.0002

Below are the values of alpha for Ridge Regression

```
# Printing the best hyperparameter alpha
print(ridge_model_cv.best_params_)
#print(ridge_model_cv.best_score_)
```

```
{'alpha': 4.0}
```

```
alpha = 8.0
ridge = Ridge(alpha=alpha)
ridge.fit(X_train, y_train)
ridge.coef_
```

Below are the results for R2-score (train), for R2-score (test), RSS (Train), RSS (Test), MSE (Train), MSE (Test), RMSE (Train), RMSE (Test)

```
0.9208978973133554
0.8594684264149942
0.9735302283203227
1.2062791757341298
0.0009535065899317558
0.002747788555203029
```

R2-score value is low in the test dataset when compared to train dataset

Below are the values of R2-score and metrics for Lasso Regression

```
{'alpha': 0.0001}
```

```
alpha =0.0002

lasso = Lasso(alpha=alpha)

lasso.fit(X_train, y_train)
```

```
Lasso(alpha=0.0002)
```

```
lasso.coef_
```

Below are the results for R2-score (train), for R2-score (test), RSS (Train), RSS (Test), MSE (Train), MSE (Test), RMSE (Train), RMSE (Test)

```
0.9227696953401989
0.8241552658206163
0.9504935213485601
1.5093963270452295
0.0009309437035735163
0.003438260426071138
```

Observations: R2-Score is better for Lasso than Ridge
Changes in Ridge Regression metrics after doubling alpha:

- R2 score of train set decreased from 0.9209 to 0.9036
- R2 score of test set remained same at 0.8593 to 0.8514

Changes in Lasso metrics after doubling alpha:

- R2 score of train set decreased from 0.9455 to 0.9227
- R2 score of test set decreased from 0.7635 to 0.8241

The most important predictor variables after the change are:

```
betas['Lasso'].sort_values(ascending=False)
```

```
GrLivArea                    0.371061
OverallQual_Very Excellent   0.142994
OverallQual_Excellent        0.092373
RoofMatl_WdShngl             0.055494
TotalBsmtSF                  0.050946
```

```
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

betas['Ridge'].sort_values(ascending=False)
```

```
OverallQual_Very Excellent    0.067865
2ndFlrSF                      0.063301
GrLivArea                     0.063038
Neighborhood_NoRidge          0.057155
OverallQual_Excellent         0.054396
```

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?

The optimum value of alpha for Ridge is 4 and for Lasso it is 0.0001
For Ridge, more than 24 coefficients were shrinked towards 0 and for Lasso, 30 coefficients, value is 0. Since there were more predictor variables, Lasso is better in this model.

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?

The most important variables for both Ridge and Lasso are before doubling with optimum values of alpha are:
Ridge: alpha- 4.0
Lasso: alpha – 0.0001

```
betas['Ridge'].sort_values(ascending=False)
```

```
OverallQual_Very Excellent    0.084712
2ndFlrSF                      0.079771
GrLivArea                     0.079328
OverallQual_Excellent         0.064340
RoofMatl_WdShngl              0.061571
                                 ...
BsmtQual_TA                  -0.033876
BsmtQual_Gd                  -0.035915
Condition2_PosN              -0.054905
MasVnrArea_762.0             -0.054905
MasVnrArea_796.0             -0.080713
Name: Ridge, Length: 580, dtype: float64
```

```
betas['Lasso'].sort_values(ascending=False)
```

```
GrLivArea                      0.370711
OverallQual_Very Excellent     0.145779
TotalBsmtSF                    0.123813
MasVnrArea_1170.0              0.100671
OverallQual_Excellent          0.089399
                                   ...
MSSubClass                    -0.024974
KitchenAbvGr                  -0.029588
MasVnrArea_762.0              -0.126476
Condition2_PosN               -0.302412
MasVnrArea_796.0              -0.650175
Name: Lasso, Length: 580, dtype: float64
```

Hence the top 5 variables to be dropped –
OverallQual_Very Excellent
GrLivArea
OverallQual_Excellent
2ndFlrSF
TotalBsmtSF
After dropping above one and redoing the Lasso regression, below are the top 5 variables
We get.

```
## View the top 5 coefficients of Lasso in descending order
betas['Lasso'].sort_values(ascending=False)[:5]
```

```
1stFlrSF            0.279429
MasVnrArea_1170.0   0.200418
MasVnrArea_1378.0   0.185128
BsmtFinSF1          0.121175
MasVnrArea_424.0    0.084153
Name: Lasso, dtype: float64
```

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?

A generalisable model should not change even if there is change in predictors. A small change in data should not impact model. Since the model doesn't impact due to change in data it means it doesn't overfit.

By doing Bias- Variance trade off , optimizing the coefficients will improve accuracy without compromising much on underlying patterns in data.