Part II - Subjective Questions

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?

1.1. The optimal alpha value for Ridge is "0.6"

1.2. The optimal alpha value for Lasso is "0.001"

1.3.1. Let's double the value of alpha and check Ridge

```
#Fitting Ridge model for the best lambda parameter 0.12
alpha = 0.12
ridgeRegression_best = Ridge(alpha=alpha)

ridgeRegression_best.fit(X_train_rfe_con, y_train_rfe)
print(ridgeRegression_best.coef_)
```

```
[ 0.39570586  0.295576    0.31028958  0.68865281  1.36243861  1.58823657
 -0.24345003  0.09378364  0.15464616  0.22400766  0.26859141 -0.41520705
  0.27930523 -0.54960158  0.37156095  0.44122776  0.54322431  0.45550968
 -1.08229793 -0.36885768 -0.80089046  0.40989235 -0.42763153 -0.25268677
 -0.24931348 -0.49811432 -0.31212666]
```

```
y_pred_ridge_m4 = ridgeRegression_best.predict(X_test_rfe_con)
```

```
r2_score(y_test_rfe, y_pred_ridge_m4)
```

```
0.8559975167021554
```

For Ridge Coeff's are increased but R2 score decreased

1.3.2. Let's double the value of alpha and check Lasso

```
#Fitting Lasso model for the best lambda parameter 0.002
alpha = 0.002
lassoRegression_best = Lasso(alpha=alpha)

lassoRegression_best.fit(X_train_rfe_con, y_train_rfe)
print(lassoRegression_best.coef_)
```

```
[ 0.41890471  0.30416796  0.29819051  0.69737826  1.30181066  1.09740536
 -0.25689498  0.0496838   0.11473931  0.15416031  0.15535571 -0.12155423
  0.2244663  -0.33700191  0.24798162  0.40925744  0.38124986  0.30253451
 -0.91579794 -0.31226143 -0.72463474  0.36397842 -0.39864209 -0.18559192
 -0.19308745 -0.21424909 -0.20939126]
```

```
y_pred_lasso_m5 = lassoRegression_best.predict(X_test_rfe_con)
```

```
r2_score(y_test_rfe, y_pred_lasso_m5)
```

```
0.8619816621939008
```

For Lasso Coeff's are increased and R2 score increased

1.4. Most important predictor variables after the change

```
house_prc_pred = pd.DataFrame(index=X_train_rfe_con.columns)
```

```
house_prc_pred.rows = X_train_rfe_con.columns
```

```
house_prc_pred["Ridge"] = ridgeRegression_best.coef_
house_prc_pred["Lasso"] = lassoRegression_best.coef_
```

```
pd.set_option('display.max_rows', None)
```

```
np.abs(house_prc_pred["Ridge"]).sort_values(ascending = False)
```

```
Overall_Q10            1.588237
Overall_Q9             1.362439
BsmtQual_Fa            1.082298
BsmtQual_TA            0.800890
Overall_Q8             0.688653
```

```
np.abs(house_prc_pred["Lasso"]).sort_values(ascending = False).head()
```

```
Overall_Q9      1.301811
Overall_Q10     1.097405
BsmtQual_Fa     0.915798
BsmtQual_TA     0.724635
Overall_Q8      0.697378
Name: Lasso, dtype: float64
```

Top 5 Features Overall_Q10, Overall_Q9, BsmtQual_Fa, BsmtQual_TA, Overall_Q8

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?

For this House Pricing Data Set, I would choose Lasso over ridge, as Lasso performance is better and also has the ability to drop off the features when it's not needed. I would apply Lasso for further prediction of hosuse prices.

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?

```
X_train_rfe_con_15 = X_train_rfe_con
X_test_rfe_con_15 = X_test_rfe_con

X_train_rfe_con = X_train_rfe_con.drop(['Overall_Q10', 'Overall_Q9', 'BsmtQual_Fa',
                                        'BsmtQual_TA', 'Overall_Q8'], axis=1)
X_test_rfe_con = X_test_rfe_con.drop(['Overall_Q10', 'Overall_Q9', 'BsmtQual_Fa',
                                      'BsmtQual_TA', 'Overall_Q8'], axis=1)
```

```
# Lamda values to penalize the cost function.

alpha_params = {'alpha': [0.00001, 0.0001, 0.001, 0.01, 0.05, 0.1,
 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0,
 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20.0, 30.0, 40.0, 50.0, 100.0, 500.0, 1000.0 ]}

#Instantiate Lasso Regression
lassoRegression = Lasso()

# Use 5 fold cross validation
folds = 5
X_train_rfe_lm_8 = GridSearchCV(estimator = lassoRegression,
                        param_grid = alpha_params,
                        scoring= 'neg_mean_absolute_error',
                        cv = folds,
                        return_train_score=True,
                        verbose = 1)
X_train_rfe_lm_8.fit(X_train_rfe_con, y_train_rfe)
```

```
Fitting 5 folds for each of 31 candidates, totalling 155 fits

GridSearchCV(cv=5, estimator=Lasso(),
             param_grid={'alpha': [1e-05, 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2,
                                   0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0,
                                   3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0,
                                   20.0, 30.0, 40.0, 50.0, 100.0, 500.0, ...]},
             return_train_score=True, scoring='neg_mean_absolute_error',
             verbose=1)
```

```
print(X_train_rfe_lm_8.best_params_)
```

```
{'alpha': 0.001}
```

```
#Fitting Lasso model for the best lambda parameter 0.002
alpha = 0.001
lassoRegression_best = Lasso(alpha=alpha)

lassoRegression_best.fit(X_train_rfe_con, y_train_rfe)
print(lassoRegression_best.coef_)
```

```
[ 0.55787178  0.39768604  0.07084261 -0.49118688 -0.10605121 -0.0389994
  0.          0.08034535 -0.15942239  0.21433446 -0.76592105  0.43555315
  1.23731853  0.91533011  0.80199295  0.28912288  0.56146024 -0.69300707
 -0.15831035 -0.25567405 -0.38359864 -0.37235253]
```

```
house_prc_pred_1 = pd.DataFrame(index=X_train_rfe_con.columns)
```

```
house_prc_pred_1.rows = X_train_rfe_con.columns
```

```
house_prc_pred_1["Lasso"] = lassoRegression_best.coef_
```

```
np.abs(house_prc_pred_1["Lasso"]).sort_values(ascending = False).head()
```

```
Neighborhood_NridgHt     1.237319
Neighborhood_StoneBr     0.915330
Exterior2nd_CmentBd      0.801993
Neighborhood_MeadowV     0.765921
MSSub_90                 0.693007
Name: Lasso, dtype: float64
```

The Five most important predictors now are Neighborhood_NridgHt, Neighborhood_StoneBr, Exterior2nd_CmentBd, Neighborhood_MeadowV, MSSub_90

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?

1. Keep minimum number of features as possible to make the model more generalisable. More number of features will overfit the model and changes in the input would result in poor performance which in turn will need to change the features to fit the new data.
2. Model will be more robust when Data is cleaned, imputated, outliers removed properly during the model training.
3. Feature selection is very important process to increase the model accuracy, so need to make sure p-value is less than 0.05 and VIF is less than 5.
4. Testing the model with different set of data will improve the robustness as well, using techniques like Cross Validation would be of great advantage.