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?

The optimum value of alpha for Lasso regression is 0.001 and for the Ridge regression is 10.

```
# Lasso Regression

params = {'alpha': [0.000001, 0.00001,0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500, 1000, 10000]}

lasso_final_model, y_test_predicted = build_model(X_train_rfe, y_train, X_test_rfe, params, model='lasso')
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Optimum alpha for lasso is 0.001000
lasso  Regression with  0.001
==================================
R2 score (train) :  0.8685177003193485
R2 score (test) :  0.8237444726491693
RMSE (train) :  0.1465615972551752
RMSE (test) :  0.16287459405406637
```

```
params = {'alpha': [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.
                    9.0, 10.0, 20, 50, 100, 500, 1000 ]}

ridge_final_model, y_test_predicted = build_model(X_train_rfe, y_tra
```

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits
Optimum alpha for ridge is 10.000000
ridge  Regression with  10.0
==================================
R2 score (train) :  0.8688341032624504
R2 score (test) :  0.8322452280120162
RMSE (train) :  0.14638514592863358
RMSE (test) :  0.15889836011733077
```

If we double the alpha values, many of the important variables are changed, some of which are mentioned below. Please refer to the python notebook for the detailed results(sample screenshot provided below).

MSZoning
Street
Alley
LandContour
Utilities

	Ridge (alpha=10.0)	Lasso (alpha=0.001)	Ridge (alpha = 20.0)	Lasso (alpha = 0.002)
MSZoning	-0.044514	-0.045421	-0.045489	-0.047526
Street	0.036834	0.000000	0.022090	0.000000
Alley	0.027105	0.018290	0.024210	0.006787
LandContour	0.029949	0.026451	0.030098	0.024500
Utilities	-0.014594	-0.000000	-0.007487	-0.000000
LandSlope	0.031089	0.025710	0.030157	0.022071
Condition1	0.010477	0.009857	0.010379	0.009366
Condition2	0.022449	0.011186	0.019357	0.000000
BldgType	-0.015655	-0.010092	-0.012361	-0.001886
HouseStyle	-0.005613	-0.005567	-0.005589	-0.005429
OverallQual	0.100099	0.102993	0.100911	0.105637
OverallCond	0.046810	0.045895	0.046689	0.044829
RoofMatl	0.008748	0.009213	0.008360	0.006723
ExterQual	0.015692	0.014312	0.015971	0.013481
ExterCond	0.026855	0.021734	0.026196	0.016971
Foundation	0.017476	0.017354	0.017533	0.016899
BsmtQual	0.025878	0.024057	0.024862	0.020241
BsmtCond	-0.011657	-0.007832	-0.011955	-0.004497
BsmtExposure	-0.011208	-0.010972	-0.011554	-0.011029
BsmtFinType1	-0.007929	-0.008163	-0.008599	-0.008976
BsmtFinType2	0.006338	0.005654	0.006469	0.005178
Heating	0.011925	0.007070	0.011119	0.000000
HeatingQC	0.024834	0.023978	0.025568	0.024513
CentralAir	0.056225	0.046101	0.050616	0.029508
Electrical	-0.001034	-0.000000	-0.000051	0.000000

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?

Will choose to apply the Lasso regression as final model for having slightly better R-square value on test data.

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?

After removing the top 5 variables, the new top 5 variables are :

CentralAir
TotRmsAbvGrd
MiscFeature
ExterCond
ExterQual

```
[74]: model_coeff = pd.DataFrame(index=X_test_new.columns)
      model_coeff.rows = X_test_new.columns
      model_coeff['Lasso'] = lasso_model.coef_
      model_coeff.sort_values(by='Lasso', ascending=False).head(5)
```

Out[74]:

	Lasso
CentralAir	0.142944
TotRmsAbvGrd	0.117009
MiscFeature	0.070973
ExterCond	0.060636
ExterQual	0.057549

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?

The model should be simple as possible, and accuracy will be relatively decreased, however, it will be generalized. The simpler the model the more the bias but less variance and more generalizable. Its implication in terms of accuracy is that a robust and generalizable model will perform equally well on both training and test data i.e. the accuracy does not change much for training and test data. In addition to that, we can also ensure the basic checks like overfitting, underfitting, and multicollinearity is not happening in the model.