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 optimal lambda value in case of Ridge and Lasso is as below:
    - Ridge – 20.0
    - Lasso 0.001
- With the above alphas the R2 of the model was approximately 0.86
- R2 score and MSE on doubling the alpha are provided below:

Alpha	Double alpha
ridge Regression with 20 =================================== R2 score (train) : 0.8744811312557577 R2 score (test) : 0.8628703193206747 RMSE (train) : 0.14005438023005398 RMSE (test) : 0.15365041085680592	ridge Regression with 40 =================================== R2 score (train) : 0.8738655740830925 R2 score (test) : 0.8614279210058255 RMSE (train) : 0.14039738060966275 RMSE (test) : 0.15445638279098745
lasso Regression with 0.001 =================================== R2 score (train) : 0.8743761965772907 R2 score (test) : 0.8632198641239766 RMSE (train) : 0.14011291123491332 RMSE (test) : 0.15345445775817845	lasso Regression with 0.002 =================================== R2 score (train) : 0.8732840108822111 R2 score (test) : 0.8617533054856525 RMSE (train) : 0.14072067082457884 RMSE (test) : 0.15427493481220578

- Top 7 predictor variables remain the same on doubling the alphas in Ridge and Lasso. There are minor differences in coefficient values. (Ref Next Page for screenshot and Jupyter Notebook for Code)

	Ridge (alpha=20.0) - Original	Lasso (alpha=0.001 - Original)	Ridge (alpha=40.0) - DoubledAlpha	Lasso
2ndFlrSF	0.130826	0.132435	0.123791	
1stFlrSF	0.105843	0.111171	0.100834	
MSZoning_RL	0.038389	0.037804	0.037725	
Fireplaces_2	0.034854	0.033804	0.034940	
OverallQual_8	0.031418	0.030810	0.032303	
Fireplaces_1	0.030974	0.029886	0.031742	
KitchenQual	0.030145	0.030106	0.031397	
CentralAir_Y	0.028831	0.028787	0.028381	
SaleCondition_Normal	0.029641	0.028959	0.028333	
Neighborhood_Somerst	0.029012	0.028898	0.028016	
BsmtQual_5	0.024699	0.024236	0.025925	
BsmtFinType1	0.026642	0.026445	0.025866	
SaleCondition_Partial	0.025832	0.025334	0.024773	
TotalBsmtSF	0.019441	0.017821	0.023269	
Neighborhood_NridgHt	0.023640	0.023395	0.023207	
Condition1_Norm	0.024135	0.024045	0.023088	
BsmtExposure_4	0.023082	0.022608	0.022811	

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 optimal lambda value in case of Ridge and Lasso is as below:
    - Ridge – 20.0
    - Lasso 0.001
- The Mean Squared error in case of Ridge and Lasso are:
    - Ridge - 0.15390088041290248
    - Lasso - 0.151448836843912 55

The Mean Squared Error of Lasso is slightly lower than that of Ridge.

Also, since Lasso helps in feature reduction (as the coefficient value of one of the feature became 0), Lasso has a better edge over Ridge.

Therefore, the variables predicted by Lasso can be applied to choose significant variables for predicting the price of a house

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 Top 5 Predictor variables from our Model analysis is as below –

Features	Description
2ndFlrSF	Second Floor Square Feet
1stFlrSF	First Floor Square Feet
MSZoning_RL	Residential Low Density zoning classification
Fireplaces_2	Two fireplaces
OverallQual_8	Rates the overall material and finish of the house

We have created another model excluding the above predictor variables, where below set of variables were identified (Refer Jupyter Notebook)

```
In [325]:  model_coefficients[['Lasso (alpha=0.001)']].sort_values(by='Lasso (alpha=0.001)', a
Out[325]:
```

	Lasso (alpha=0.001)
TotalBsmtSF	0.097659
HouseStyle_2Story	0.065078
KitchenQual	0.056875

However the R2 score was reduced and MSE was increased , when removing the top 5 predictors

```
In [316]:  params = {'alpha': [0.000001, 0.00001,0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 500, 1000,

           lasso_final_model, y_test_predicted = build_model(X_train_rfe3, y_train, X_test_rfe3, p

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) :  0.796564262813937
R2 score (test) :  0.7774235211606022
```

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?

Per, Occam's Razor— given two models that show similar 'performance' in the finite training or test data, we should pick the one that makes fewer on the test data due to following reasons:-

- Simpler models are usually more 'generic' and are more widely applicable
- Simpler models require fewer training samples for effective training than the more complex ones and hence are easier to train.
- Simpler models are more robust.
    - Complex models tend to change wildly with changes in the training data set
    - Simple models have low variance, high bias and complex models have low bias, high variance
- Simpler models make more errors in the training set. Complex models lead to overfitting - they work very well for the training samples, fail miserably when applied to other test samples

Therefore to make the model more robust and generalizable, make the model simple but not simpler which will not be of any use.

Regularization can be used to make the model simpler. Regularization helps to strike the delicate balance between keeping the model simple and not making it too naive to be of any use. For regression, regularization involves adding a regularization term to the cost that adds up the absolute values or the squares of the parameters of the model.

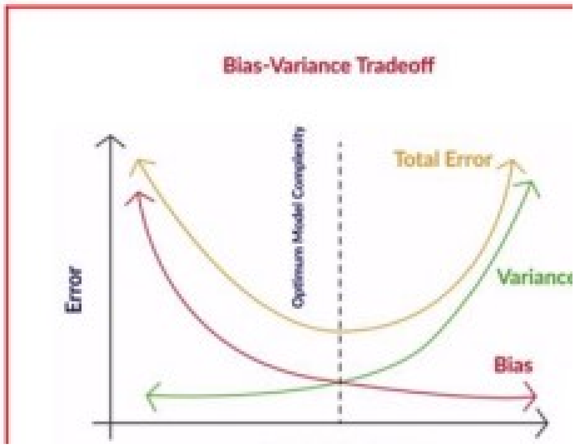
Also, Making a model simple leads to Bias-Variance Trade-off:

- A complex model will need to change for every little change in the dataset and hence is very unstable and extremely sensitive to any changes in the training data.
- A simpler model that abstracts out some pattern followed by the data points given is unlikely to change wildly even if more points are added or removed.

Bias quantifies how accurate is the model likely to be on test data. A complex model can do an accurate job prediction provided there is enough training data. Models that are too naïve, for e.g. one that gives same answer to all test inputs and makes no discrimination whatsoever has a very large bias as its expected error across all test inputs are very high.

Variance refers to the degree of changes in the model itself with respect to changes in the training data.

Thus accuracy of the model can be maintained by keeping the balance between Bias and Variance as it minimizes the total error as shown in the below graph


Bias-Variance Tradeoff
Error
Optimum Model Complexity
Total Error
Varianc
Bias