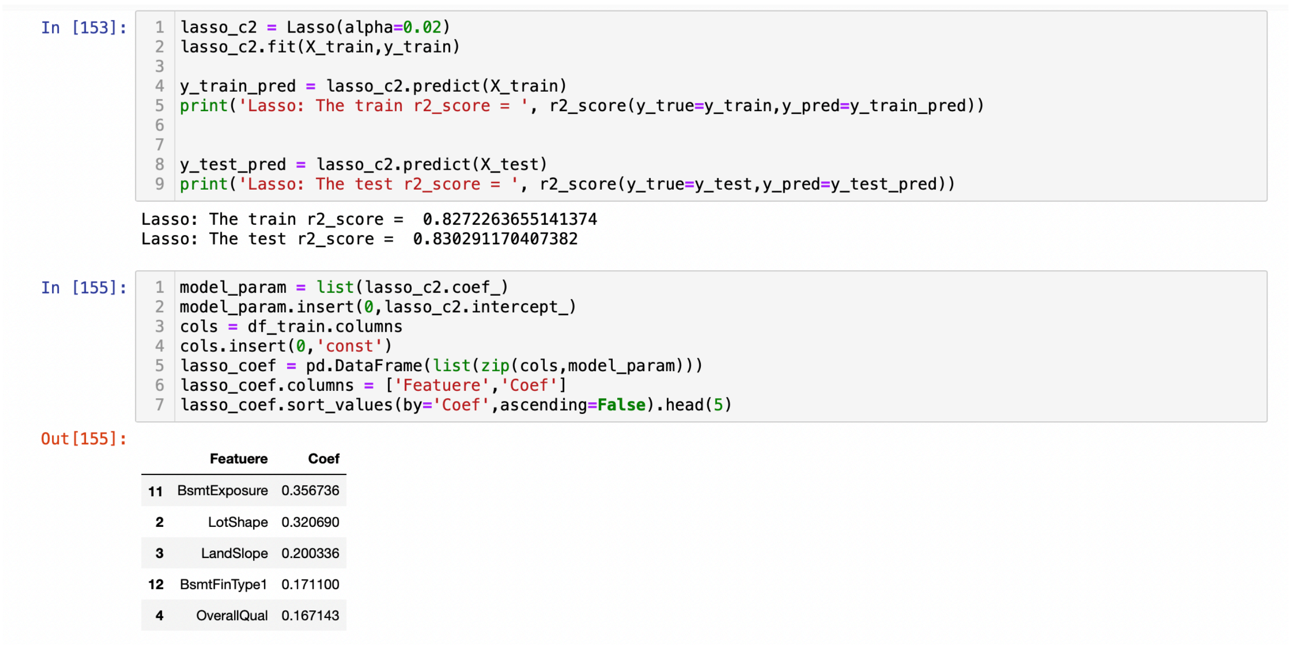
Assignment - Part II

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?

Ans:

- The alpha values for ridge is 1.0 and alpha value for lasso is 0.01
- After doubling alpha, the alpha for ridge is 2 and for lasso it is 0.02.



In [153]:
```
lasso_c2 = Lasso(alpha=0.02)
lasso_c2.fit(X_train,y_train)

y_train_pred = lasso_c2.predict(X_train)
print('Lasso: The train r2_score = ', r2_score(y_true=y_train,y_pred=y_train_pred))


y_test_pred = lasso_c2.predict(X_test)
print('Lasso: The test r2_score = ', r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
Lasso: The train r2_score =  0.8272263655141374
Lasso: The test r2_score =  0.830291170407382
```

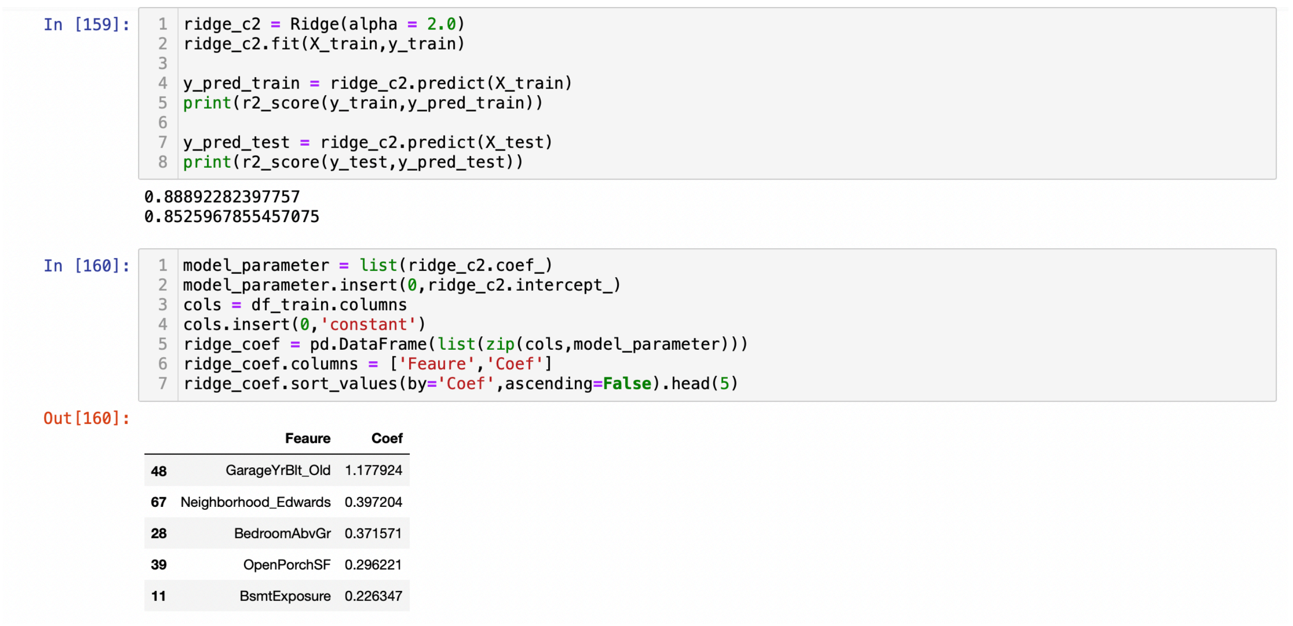
In [155]:
```
model_param = list(lasso_c2.coef_)
model_param.insert(0,lasso_c2.intercept_)
cols = df_train.columns
cols.insert(0,'const')
lasso_coef = pd.DataFrame(list(zip(cols,model_param)))
lasso_coef.columns = ['Featuere','Coef']
lasso_coef.sort_values(by='Coef',ascending=False).head(5)
```

Out[155]:

	Featuere	Coef
11	BsmtExposure	0.356736
2	LotShape	0.320690
3	LandSlope	0.200336
12	BsmtFinType1	0.171100
4	OverallQual	0.167143

When the alpha value is doubled in Lasso. The test r2 score is more than train and the top 5 significant features changed.
Before when alpha is 0.01, then the features were:

1. GrLivArea
2. OverallQual
3. Neighborhood_NridgHt
4. ExterQual
5. BsmtFinSF1



In [159]:
```
ridge_c2 = Ridge(alpha = 2.0)
ridge_c2.fit(X_train,y_train)

y_pred_train = ridge_c2.predict(X_train)
print(r2_score(y_train,y_pred_train))

y_pred_test = ridge_c2.predict(X_test)
print(r2_score(y_test,y_pred_test))
```

```
0.88892282397757
0.8525967855457075
```

In [160]:
```
model_parameter = list(ridge_c2.coef_)
model_parameter.insert(0,ridge_c2.intercept_)
cols = df_train.columns
cols.insert(0,'constant')
ridge_coef = pd.DataFrame(list(zip(cols,model_parameter)))
ridge_coef.columns = ['Feaure','Coef']
ridge_coef.sort_values(by='Coef',ascending=False).head(5)
```

Out[160]:

	Feaure	Coef
48	GarageYrBlt_Old	1.177924
67	Neighborhood_Edwards	0.397204
28	BedroomAbvGr	0.371571
39	OpenPorchSF	0.296221
11	BsmtExposure	0.226347

When the alpha value is doubled in Ridge the difference between r2 score of test and train is better than when alpha is 1.0 And the 5 significant features changed as compared to when the alpha was 1.0

1. RoofMatl_WdShngl
2. SaleType_New
3. Neighborhood_NridgHt
4. BldgType_2fmCon
5. RoofMatl_Membran

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?

Ans:
The optimal value of lambda for Ridge and Lasso is as below:

- Ridge 1.0
- Lasso: 0.01

R2_score in Ridge and Lasso is as below:

- Ridge:
  - rain: 0.89
  - Test: 0.84
- Lasso:
  - Train: 0.84
  - Test: 0.84

Even the test r2_score for both Ridge and Lasso is same. I would like to choose Lasso over ridge because, Lasso has automatic fine tuning of features, hence, few feature coefficients becomes zero. Which helps in feature reduction.

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?

Ans:
The 5 most important features for Lasso are:

1. GrLivArea
2. OverallQual
3. Neighborhood_NridgHt
4. ExterQual
5. BsmtFinSF1

In [162]:
```
Removing the 5 most important predictor variables from the incoming dataset
_test_c3 = X_test.drop(['GrLivArea', 'OverallQual', 'Neighborhood_NridgHt', 'ExterQual', 'BsmtFinSF1'],axis=1)
_train_c3 = X_train.drop(['GrLivArea', 'OverallQual', 'Neighborhood_NridgHt', 'ExterQual', 'BsmtFinSF1'],axis=1)

 Building Lasso Model with the new dataset
asso_c3 = Lasso(alpha=0.01)
asso_c3.fit(X_train_c3,y_train)

_train_pred = lasso_c3.predict(X_train_c3)
rint('Lasso: The train r2_score = ', r2_score(y_true=y_train,y_pred=y_train_pred))


_test_pred = lasso_c3.predict(X_test_c3)
rint('Lasso: The test r2_score = ', r2_score(y_true=y_test,y_pred=y_test_pred))
```

```
Lasso: The train r2_score =  0.7990207080760099
Lasso: The test r2_score =  0.7960763689724174
```

In [163]:
```
model_param = list(lasso_c3.coef_)
model_param.insert(0,lasso_c3.intercept_)
cols = df_train.columns
cols.insert(0,'const')
lasso_coef = pd.DataFrame(list(zip(cols,model_param)))
lasso_coef.columns = ['Featuere','Coef']
lasso_coef.sort_values(by='Coef',ascending=False).head(5)
```

Out[163]:

	Featuere	Coef
8	ExterCond	0.454309
6	MasVnrArea	0.410821
4	OverallQual	0.333556
5	OverallCond	0.308958
62	Neighborhood_BrDale	0.247122

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?

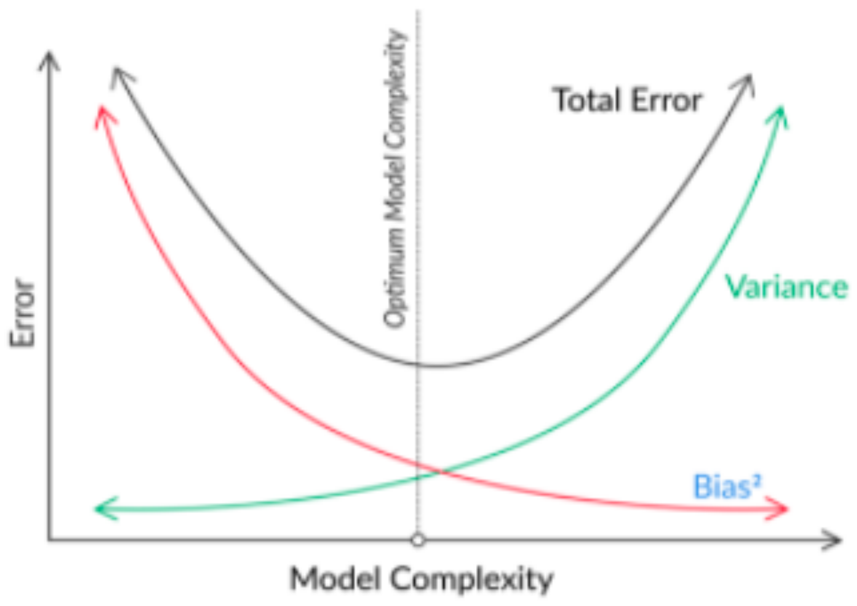
Ans:

To make sure, model is robust and generalisable following are to be taken care of:

- Model should not be complex, it should be made simple to make it more generic and more widely acceptable. That makes simpler models are more robust than complex ones
- Simpler models are easier to train than the complex ones

However, models should be made quite simpler and lose their value because:
- Simpler models have low variance and high bias, whereas complex models have low bias and high variance
- Simpler models make more errors in training data, whereas complex models may overfit

The above concerns can be taken care by Regularisation
Regularisation is used to strike the balance between bias and variance
We need lowest total error possible, so that model identifies all the patterns and also performs well for unseen data


Optimum Model Complexity
Total Error
Variance
Bias²
Error
Model Complexity