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 value of alpha for ridge is 2 and for Lasso is 0.0001.

299]:

```
# Building Ridge Model by doubling the value of alpha to 4
ridge_double = Ridge(alpha=4,random_state=100)
ridge_double.fit(X_train_scaled_signi,y_train)
ridge_double_coef = ridge_double.coef_
y_test_pred = ridge_double.predict(X_test_scaled_signi)
print('The R2 Score of the model on the test dataset for doubled alpha is',r2_score(y_test, y_test_pred))
print('The RMSE of the model on the test dataset for doubled alpha is', mean_squared_error(y_test, y_test_pred,squared=False))
ridge_double_coeff = pd.DataFrame(np.atleast_2d(ridge_double_coef),columns=X_train_scaled_signi.columns)
ridge_double_coeff = ridge_double_coeff.T
ridge_double_coeff.rename(columns={0: 'Ridge Doubled Alpha Co-Efficient'},inplace=True)
ridge_double_coeff.sort_values(by=['Ridge Doubled Alpha Co-Efficient'], ascending=False,inplace=True)
print('The most important predictor variables are as follows:')
ridge_double_coeff.head(20)
```

```
The R2 Score of the model on the test dataset for doubled alpha is 0.8222476942982999
The RMSE of the model on the test dataset for doubled alpha is 0.05452959050520473
The most important predictor variables are as follows:
```

299]:

	Ridge Doubled Alpha Co-Efficient
1stFlrSF	0.202030
GarageArea	0.103809
TotRmsAbvGrd	0.076042
OverallCond	0.065697
LotArea	0.059202
HouseStyle_2Story	0.054084
FullBath	0.054035
HouseStyle_2.5Unf	0.052495
LotFrontage	0.045946
BsmtFullBath	0.042373
OpenPorchSF	0.041333
CentralAir_Y	0.040436
HouseStyle_1.5Fin	0.038235
WoodDeckSF	0.035737
BsmtQual_Ex	0.034517
Neighborhood_StoneBr	0.033529
Neighborhood_Veenker	0.031806
KitchenQual_Ex	0.028903
PavedDrive_Y	0.027071
ScreenPorch	0.026551

In [300]:

```
# Building Lasso Model by doubling the value of alpha to 0.0002
lasso_double = Lasso(alpha=0.0002,random_state=100)
lasso_double.fit(X_train_scaled_signi,y_train)
lasso_double_coef = lasso_double.coef_
y_test_pred = lasso_double.predict(X_test_scaled_signi)
print('The R2 Score of the model on the test dataset for doubled alpha is',r2_score(y_test, y_test_pred))
print('The RMSE of the model on the test dataset for doubled alpha is', mean_squared_error(y_test, y_test_pred,squared=False
lasso_double_coeff = pd.DataFrame(np.atleast_2d(lasso_double_coef),columns=X_train_scaled_signi.columns)
lasso_double_coeff = lasso_double_coeff.T
lasso_double_coeff.rename(columns={0: 'Lasso Doubled Alpha Co-Efficient'},inplace=True)
lasso_double_coeff.sort_values(by=['Lasso Doubled Alpha Co-Efficient'], ascending=False,inplace=True)
print('The most important predictor variables are as follows:')
lasso_double_coeff.head(20)
```

```
The R2 Score of the model on the test dataset for doubled alpha is 0.8189168050762153
The RMSE of the model on the test dataset for doubled alpha is 0.05503813245384374
The most important predictor variables are as follows:
```

Out[300]:

	Lasso Doubled Alpha Co-Efficient
1stFlrSF	0.316258
GarageArea	0.107076
HouseStyle_2.5Unf	0.076554
HouseStyle_2Story	0.074855
OverallCond	0.072092
LotArea	0.056392
HouseStyle_1.5Fin	0.044178
CentralAir_Y	0.041397
TotRmsAbvGrd	0.041032
FullBath	0.039079
BsmtFullBath	0.038216
OpenPorchSF	0.033149
BsmtQual_Ex	0.032677
Neighborhood_StoneBr	0.028387
WoodDeckSF	0.027557
KitchenQual_Ex	0.024213
LotFrontage	0.022921
PavedDrive_Y	0.021986
BsmtCond_TA	0.021165
Condition1_Norm	0.018987

We don't see any significant change in R2 score and RMSE after doubling the alpha value, smaller of alpha could be the reason. We cannot see significant change in the coefficient value for variables but there is some changes like Total rooms above grade is more significant than Overall Condition and Lot Area in case of Ridge and the Central Air Conditioned gains higher importance than Lot Area in the Lasso Model.

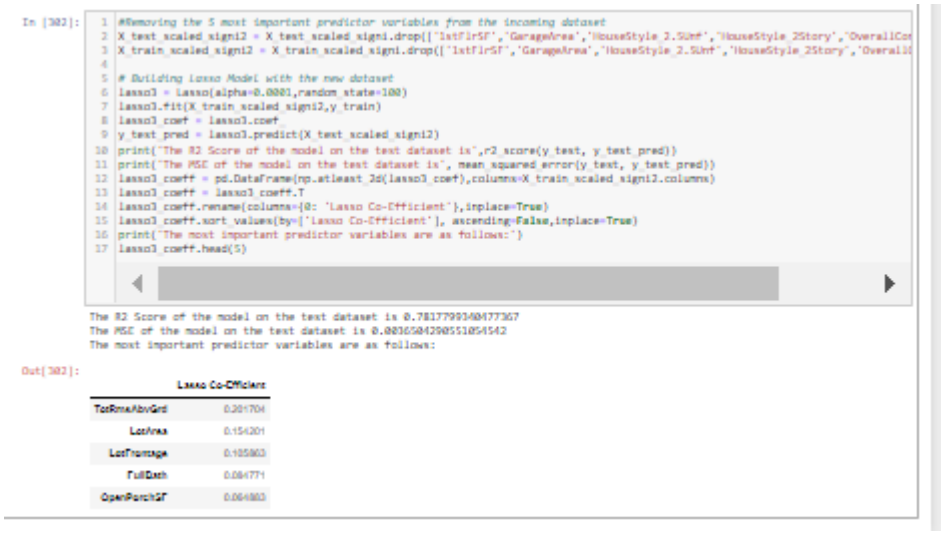
Question 2

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?

Lasso is better because it helps us to eliminate the less significant features while Ridge keep does not completely remove the columns and overall model will be more robust. It works like feature selection technique.

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?



```
In [302]:
#Removing the 5 most important predictor variables from the incoming dataset
X_test_scaled_signi2 = X_test_scaled_signi.drop(['1stFlrSF','GarageArea','HouseStyle_2.5Unf','HouseStyle_2Story','OverallCon
X_train_scaled_signi2 = X_train_scaled_signi.drop(['1stFlrSF','GarageArea','HouseStyle_2.5Unf','HouseStyle_2Story','Overall

# Building Lasso Model with the new dataset
lasso3 = Lasso(alpha=0.0001,random_state=100)
lasso3.fit(X_train_scaled_signi2,y_train)
lasso3_coef = lasso3.coef_
y_test_pred = lasso3.predict(X_test_scaled_signi2)
print('The R2 Score of the model on the test dataset is',r2_score(y_test, y_test_pred))
print('The MSE of the model on the test dataset is', mean_squared_error(y_test, y_test_pred))
lasso3_coeff = pd.DataFrame(np.atleast_2d(lasso3_coef),columns=X_train_scaled_signi2.columns)
lasso3_coeff = lasso3_coeff.T
lasso3_coeff.rename(columns={0: 'Lasso Co-Efficient'},inplace=True)
lasso3_coeff.sort_values(by=['Lasso Co-Efficient'], ascending=False,inplace=True)
print('The most important predictor variables are as follows:')
lasso3_coeff.head(5)

The R2 Score of the model on the text dataset is 0.7817799340477367
The MSE of the model on the text dataset is 0.0036504290551054542
The most important predictor variables are as follows:

Out[302]:
```

	Lasso Co-Efficient
TotRmsAbvGrd	0.201704
LotArea	0.154201
LotFrontage	0.105803
FullBath	0.084771
OpenPorchSF	0.064803

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 model is robust when any variation in the data does not affect the model performance.
A model is said to generalized if it able to adapt properly on unseen data or data which is not seen previously by model, drawn from same distribution as the one which used for creating the model.
To make sure a model is robust and generalisable, we must take care it doesn't overfit. This is because an overfitting model have high variance i.e., a small change in the data will affect the prediction heavily. Such model will be able to identify or memorize the pattern in training data but fail to identify the pattern in test data and perform poorly. In other words, model is too complex to be robust and generalisable.

If we look at it from the perceptive of accuracy a too complex model will have a very high accuracy on training data. So, to make model to decrease variance and more robust and generalisation, we need to add some bias in the model. Adding bias to model will decrease accuracy of model. So, we need to find the perfect balance between complexity and accuracy of model. This can be achieved by using regularization technique like Ridge and Lasso.