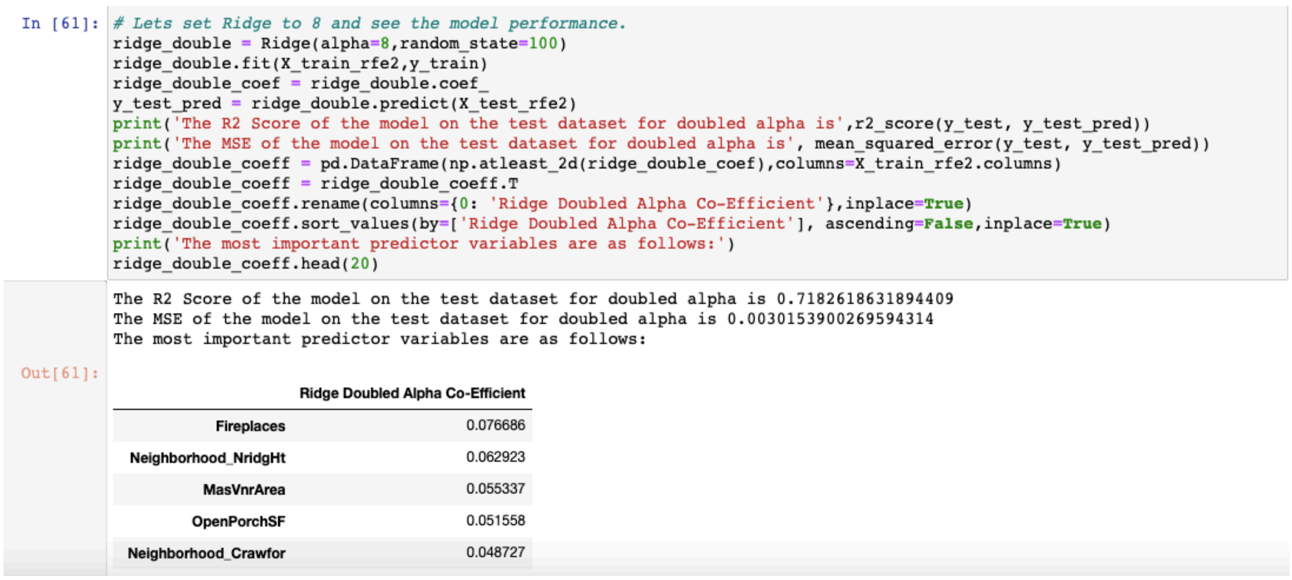
Q1. What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Ans: The Optimal Ridge is: 4. Optimal Lasso is: 0.0002

Doubling Ridge has no effect on model, However doubling lasso reduces the R2 score of model even further.



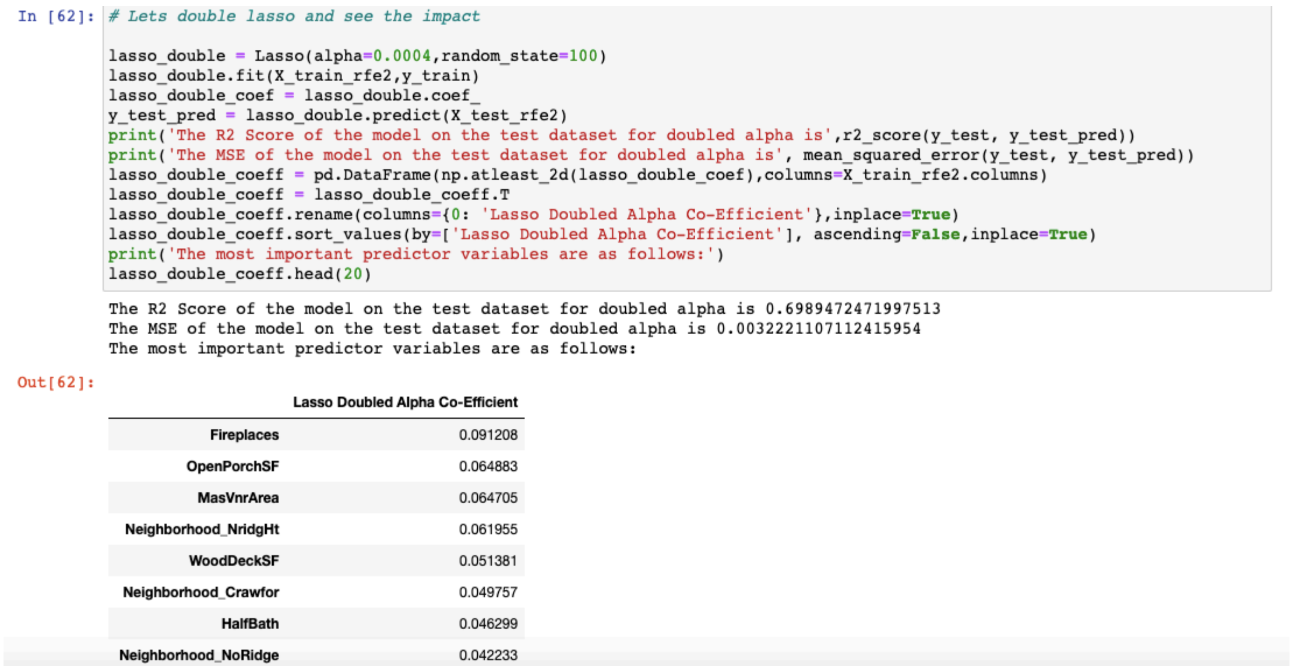
```
In [61]: # Lets set Ridge to 8 and see the model performance.
         ridge_double = Ridge(alpha=8,random_state=100)
         ridge_double.fit(X_train_rfe2,y_train)
         ridge_double_coef = ridge_double.coef_
         y_test_pred = ridge_double.predict(X_test_rfe2)
         print('The R2 Score of the model on the test dataset for doubled alpha is',r2_score(y_test, y_test_pred))
         print('The MSE of the model on the test dataset for doubled alpha is', mean_squared_error(y_test, y_test_pred))
         ridge_double_coeff = pd.DataFrame(np.atleast_2d(ridge_double_coef),columns=X_train_rfe2.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 0.7182618631894409
The MSE of the model on the test dataset for doubled alpha is 0.0030153900269594314
The most important predictor variables are as follows:
```

Out[61]:

	Ridge Doubled Alpha Co-Efficient
Fireplaces	0.076686
Neighborhood_NridgHt	0.062923
MasVnrArea	0.055337
OpenPorchSF	0.051558
Neighborhood_Crawfor	0.048727

PIC: Ridge Double



```
In [62]: # Lets double lasso and see the impact

         lasso_double = Lasso(alpha=0.0004,random_state=100)
         lasso_double.fit(X_train_rfe2,y_train)
         lasso_double_coef = lasso_double.coef_
         y_test_pred = lasso_double.predict(X_test_rfe2)
         print('The R2 Score of the model on the test dataset for doubled alpha is',r2_score(y_test, y_test_pred))
         print('The MSE of the model on the test dataset for doubled alpha is', mean_squared_error(y_test, y_test_pred))
         lasso_double_coeff = pd.DataFrame(np.atleast_2d(lasso_double_coef),columns=X_train_rfe2.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 0.6989472471997513
The MSE of the model on the test dataset for doubled alpha is 0.0032221107112415954
The most important predictor variables are as follows:
```

Out[62]:

	Lasso Doubled Alpha Co-Efficient
Fireplaces	0.091208
OpenPorchSF	0.064883
MasVnrArea	0.064705
Neighborhood_NridgHt	0.061955
WoodDeckSF	0.051381
Neighborhood_Crawfor	0.049757
HalfBath	0.046299
Neighborhood_NoRidge	0.042233

PIC: Lasso Double

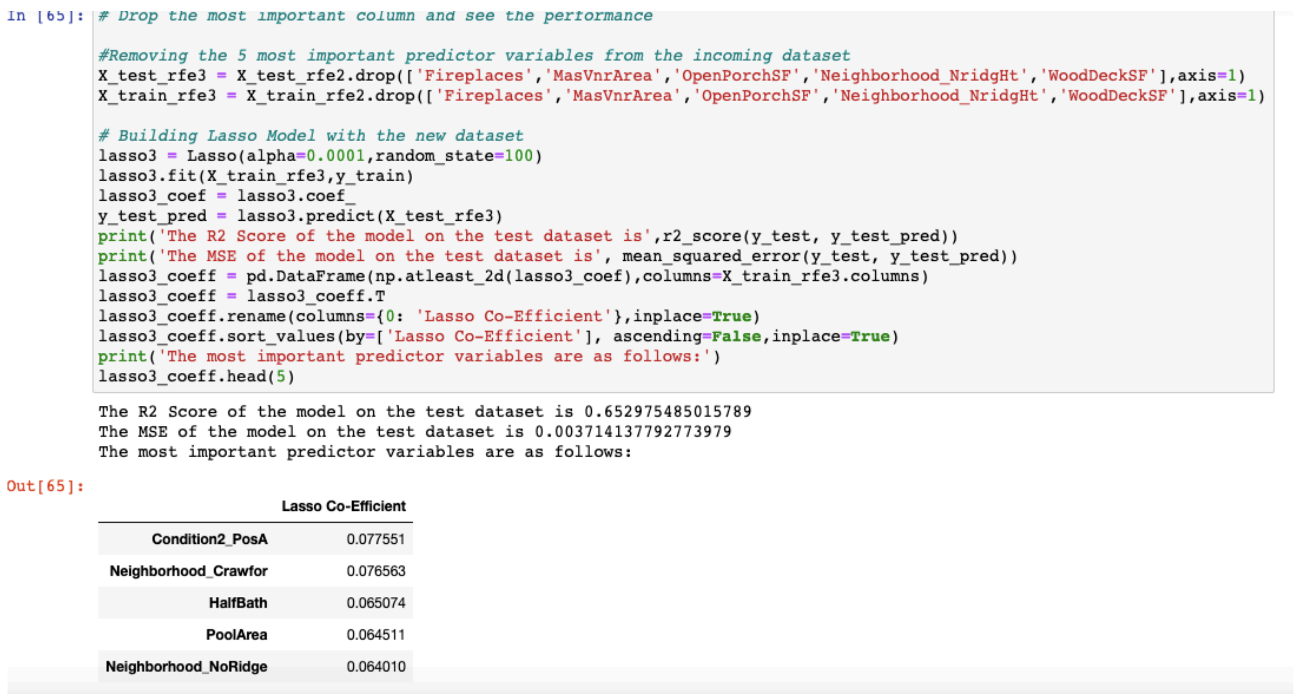
Q2.You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Lasso results in some of the coefficients going down to 0. By applying the principle of having simpler models, Lasso is a better choice here.

Q3. 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 dropping most important predictors, we see R2 score going down. Also, 5 new variables came as important.

1. Condition2_PosA
2. Neighbourhood
3. Halfbath
4. PoolArea
5. Neighbourhood with No Ridges



```
In [65]: # Drop the most important column and see the performance

         #Removing the 5 most important predictor variables from the incoming dataset
         X_test_rfe3 = X_test_rfe2.drop(['Fireplaces','MasVnrArea','OpenPorchSF','Neighborhood_NridgHt','WoodDeckSF'],axis=1)
         X_train_rfe3 = X_train_rfe2.drop(['Fireplaces','MasVnrArea','OpenPorchSF','Neighborhood_NridgHt','WoodDeckSF'],axis=1)

         # Building Lasso Model with the new dataset
         lasso3 = Lasso(alpha=0.0001,random_state=100)
         lasso3.fit(X_train_rfe3,y_train)
         lasso3_coef = lasso3.coef_
         y_test_pred = lasso3.predict(X_test_rfe3)
         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_rfe3.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 test dataset is 0.652975485015789
         The MSE of the model on the test dataset is 0.003714137792773979
         The most important predictor variables are as follows:

Out[65]:
```

	Lasso Co-Efficient
Condition2_PosA	0.077551
Neighborhood_Crawfor	0.076563
HalfBath	0.065074
PoolArea	0.064511
Neighborhood_NoRidge	0.064010

Q4. 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. Model should be resistant to outliers. After removing outlier, the model performance improved a lot for me, which I forgot to do in first iteration.
2. Do try and see if scaling and transformation is needed. Right tailed distribution can be transformed using log.
3. The accuracy may increase or decrease. The right amount of iteration should be performed to come up with right set of models.