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 most optimal value of alpha for lasso and ridge is :

Final Conclusion :

- The optimal lambda value in case of Ridge and Lasso is as below:
  - Ridge - 0.001
  - Lasso - 0.0001
- The Mean Squared error in case of Ridge and Lasso are:
  - Ridge - 6.730943175119492e-05
  - Lasso - 6.822861594720902e-05

Doubling Ridge :

Alpha	Test R2	Train R2	MSE
20	0.8822619692141302	0.9038601608452475	0.0007408214320541976
40	0.8813480444197265	0.9033203580383578	0.0007465719535336287
Difference	0.0009139247944040	0.0005398028068900	-0.000005750521479

There is an increase in MSE on increasing the alpha of ridge.

Doubling Lasso :

Alpha	Test R2	Train R2	MSE
0.0002	0.8826145980443642	0.9037507971447643	0.0007386026502956293
0.0004	0.8821870728042912	0.9031936539598070	0.0007412926890067923
Difference	0.0004275252400730	0.0005571431849570	-0.000002690038711

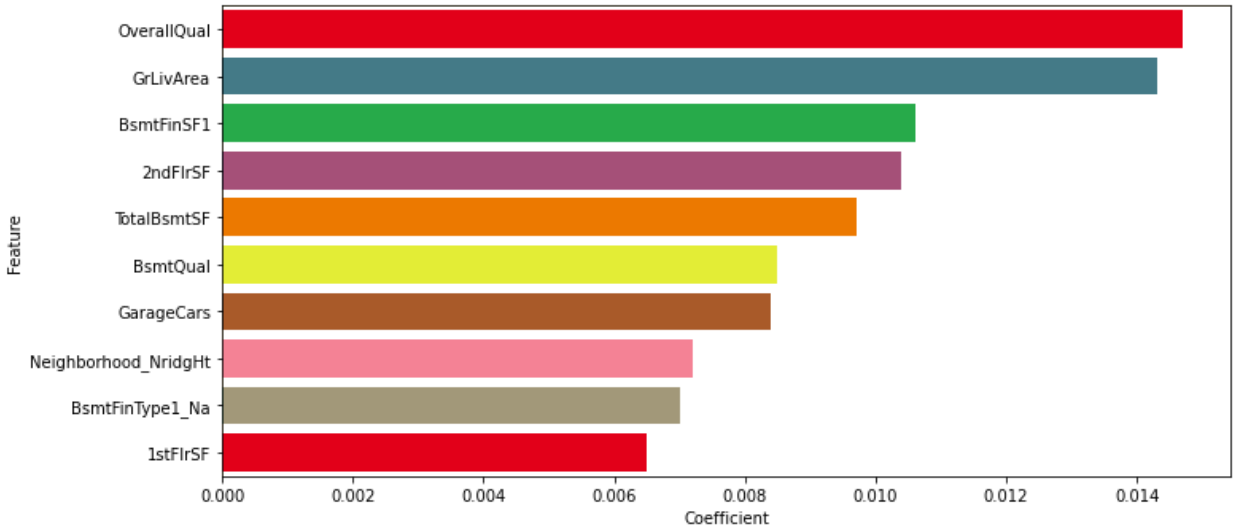
There is an increase in MSE on increasing the alpha for lasso also.

Most important predictor variables for Ridge are:

```
ridge_coeffs_df = ridge_coeffs.sort_values(by=['Coefficient'], ascending=False)

ridge_coeffs_df = ridge_coeffs_df.head(10)
# bar plot

plt.figure(figsize=(25,25))
plt.subplot(4,2,1)
sns.barplot(y = 'Feature', x='Coefficient', palette='Set1', data = ridge_coeffs_df)
plt.show()
```

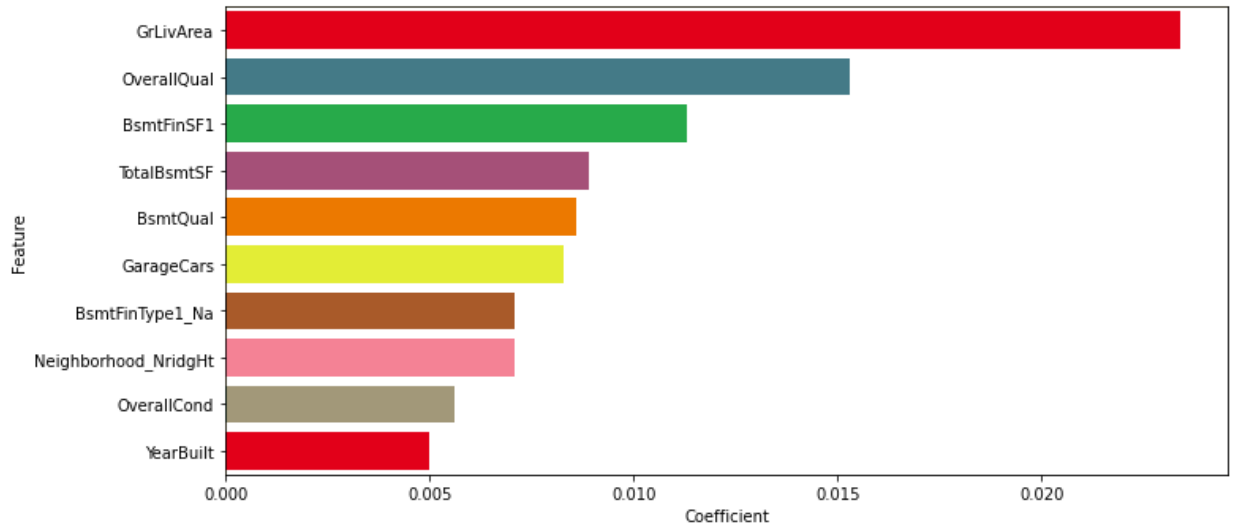


Most important predictor variables for Lasso are:

```
lasso_coeffs_df = lasso_coeffs.sort_values(by=['Coefficient'], ascending=False)

lasso_coeffs_df = lasso_coeffs_df.head(10)
# bar plot

plt.figure(figsize=(25,25))
plt.subplot(4,2,1)
sns.barplot(y = 'Feature', x='Coefficient', palette='Set1', data = lasso_coeffs_df)
plt.show()
```



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?

1. MSE in case of Ridge and Lasso is 0.000740 and 0.000738 resp.

2. MSE of Lasso is lower than that of Ridge and Lasso helps in feature reduction, Lasso has a better upper hand over Ridge, So i will choose Lasso.

3. We end up with fewer features included in the model than we started with, which has a huge advantage.

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?

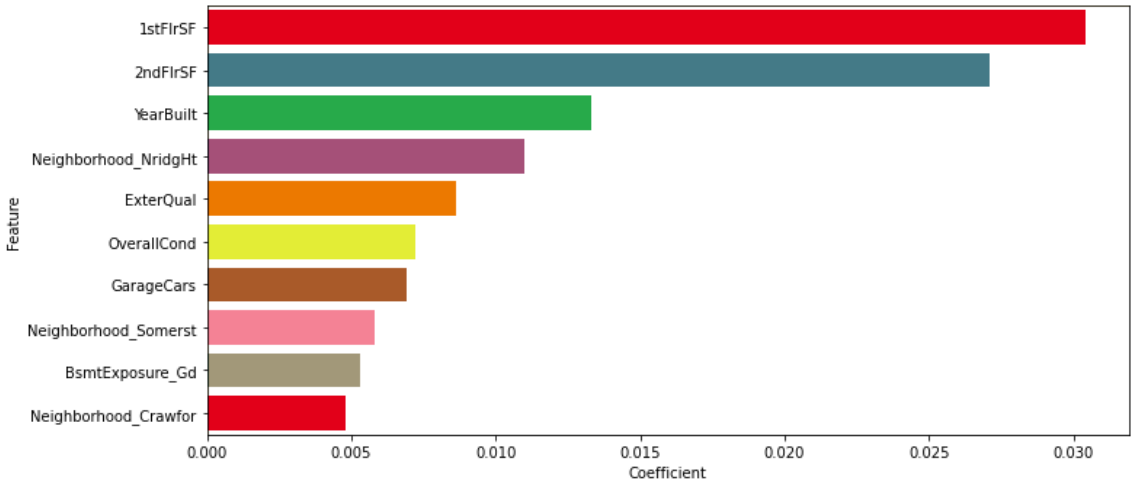
After removing five most important predictor variables in lasso these are the next top 5 most important predictor variables.

	Feature	Coefficient
7	1stFlrSF	0.0304
8	2ndFlrSF	0.0271
3	YearBuilt	0.0133
16	Neighborhood_NridgHt	0.0110
5	ExterQual	0.0086
2	OverallCond	0.0072
11	GarageCars	0.0069
17	Neighborhood_Somerst	0.0058
13	BsmtExposure_Gd	0.0053
15	Neighborhood_Crawfor	0.0048

In [1995]:
```
lasso_coeffs_df = lasso_coeffs.sort_values(by=['Coefficient'], ascending=False)

lasso_coeffs_df = lasso_coeffs_df.head(10)
# bar plot

plt.figure(figsize=(25,25))
plt.subplot(4,2,1)
sns.barplot(y = 'Feature', x='Coefficient', palette='Set1', data = lasso_coeffs_df)
plt.show()
```



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?

To make model more robust we can do following things:

1. **Transform your data**. If your data has a tail, try a log transformation.
2. **Remove the outliers**. If there are very few of them and you're fairly certain they're anomalies and not worth predicting.
3. **Use Regularization.**