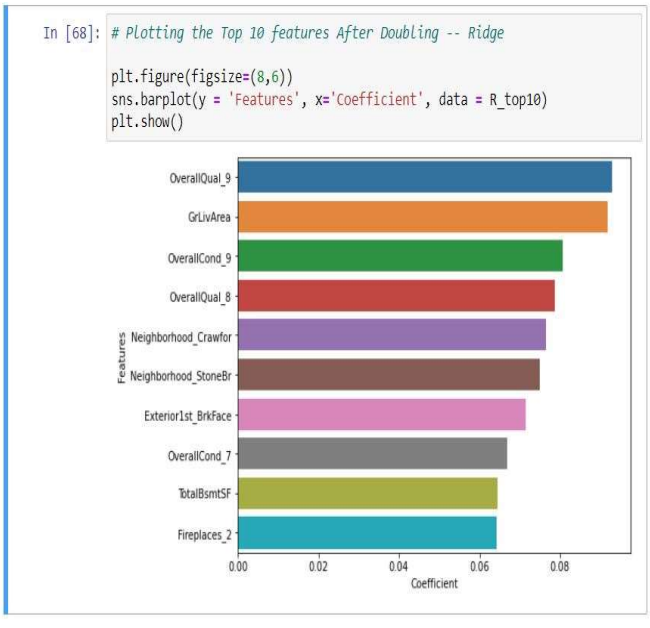
Subjective Answers

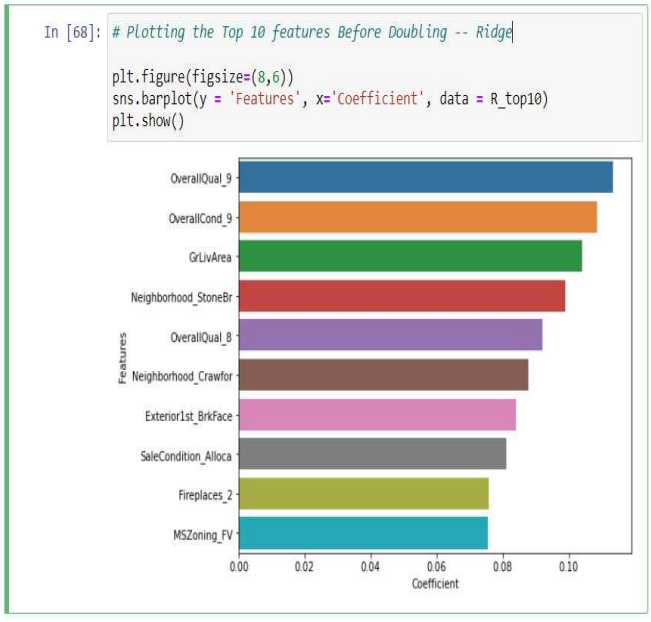
Question 1

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?

Answer 1

- The Optimal value of alpha for Ridge is *7*, while for Lasso it is ***0.0005***,
- If we were to **double** the value of alpha for both Ridge and Lasso, The Model **performance observed was more or less the Same** with very insignificant change in any of the metrics,
- After change in **Ridge** the most important predictor variable remained same as before which is ***Overall Quality,*** although the hierarchy for top 10 changed to a certain degree and some variables were replaced by others.

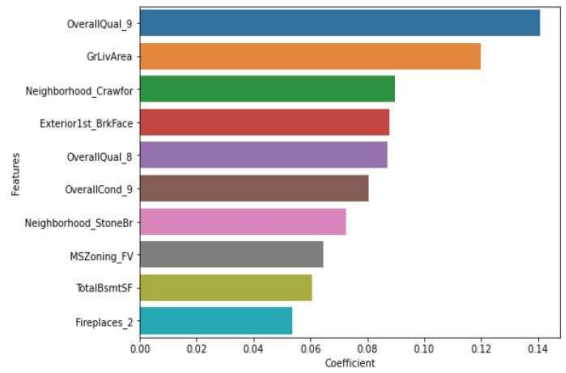




- After change in **Lasso**, ***Overall Quality*** became the most important variable in place of Overall Condition, with a considerable change in hierarchy of top 10, with OverallCond_7 being replaced by TotalBsmtSF.

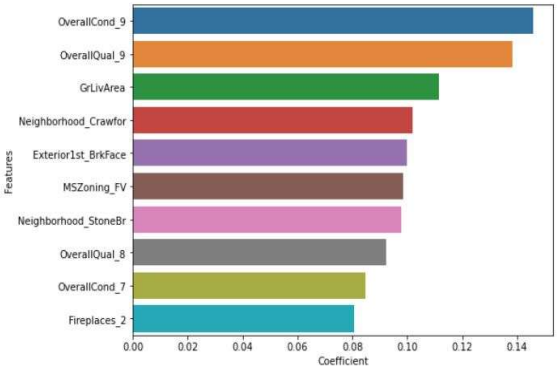
In [87]:
```
# Plotting the Top 10 features After Doubling -- Lasso

plt.figure(figsize=(8,6))
sns.barplot(y = 'Features', x='Coefficient', data = L_top10)
plt.show()
```



In [87]:
```
# Plotting the Top 10 features Before Doubling -- Lasso

plt.figure(figsize=(8,6))
sns.barplot(y = 'Features', x='Coefficient', data = L_top10)
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?

Answer 2

- The results we obtained in terms of R2 score, MSE, etc. for the training and testing dataset were quite similar for both Ridge and Lasso, in training Ridge was performing better, while in testing Lasso was performing better, although the results are fairly indistinguishable and the difference is negligible.

The Train R2 scores are -:

```
1. Ridge Regression    --->  0.9246
2. Lasso Regression    --->  0.9196
```

The Test R2 scores are -:

```
1. Ridge Regression    --->  0.9009
2. Lasso Regression    --->  0.9037
```

The Mean squared errors are -:

```
1. Ridge Regression Train   --->  0.0117
2. Ridge Regression Test    --->  0.0168
3. Lasso Regression Train   --->  0.0125
4. Lasso Regression Test    --->  0.0163
```

- The preference has to be given to Lasso because it provides 0 as coefficient to some features which in turn terminate those features from our model, thus reducing the number of features in our final model, while Ridge gives coefficient close to 0 but never 0 to any feature, therefore Lasso is preferred due to its tendency to help 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?

Answer 3

- If we are to exclude top 5 features as in only the label, `Overall_quality_9`, `Overall_Condition_9`, `GrLivArea`, `Nighborhood_Crawford` and `Exterior_Brickface` then the next top 5 features as per Lasso would be,

1. ***`MSZoning_FV`,***
2. ***`Neighborhood_StoneBr`,***
3. ***`OverallQual_8`,***
4. ***`OverallCond_7`, and***
5. ***`Fireplaces_2`***

	Features	Coefficient
29	MSZoning_FV	0.09865
61	Neighborhood_StoneBr	0.09767
81	OverallQual_8	0.09225
89	OverallCond_7	0.08483
185	Fireplaces_2	0.08081

Although if the complete feature is not available in incoming data,

i.e., `Overall quality`, `Overall condition`, `Neighborhood`, `GrLivArea`, and `Exterior` columns are not available anymore,

The top 5 predictor variables as per Lasso would then be,

1. ***`SaleCondition_Alloca`,***
2. ***`MSZoning_FV`,***
3. ***`2ndFlrSF`,***
4. ***`MSSubClass_70 `,***
5. ***`Fireplaces_2`,***

	Features	Coefficient
150	SaleCondition_Alloca	0.18923
28	MSZoning_FV	0.12677
7	2ndFlrSF	0.12283
19	MSSubClass_70	0.11861
114	Fireplaces_2	0.10958

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?

Answer 4

A good way to ensure that the model is generalised or robust is to make sure that the model is neither overfitting or underfitting,

As per **Occam's razor** models should be as simple as possible but not simpler, as complex models have a very high variance wherein a subtle change in input data can cause huge changes in output of data thus making model very unstable, while a very simple model has very high bias wherein even a change of great magnitude results in little or no change in output, both are very bad from practical purposes and there exists a trade-off between the said bias and variance, using which we need to determine a Optimal value such that, our model is most stable and robust and it can adapt changes in the input data without compromising on accuracy,

So, a simple model should be targeted in such a way that it learns the pattern in the data in best possible manner without learning much of the noise present in the data,

A general trait of robust model is that it performs quite similar on both test and train data and accuracy error is quite often very similar.

Whereas an overfit model performs exceptionally well on training data such that it provides near perfect training accuracy but on the test, data is fails quite horrendously,

At the same time underfit model performs poorly on both train and test datasets in terms of accuracy.

So, by making sure our model learns the pattern behind the data and not the data itself (i.e., avoiding both overfitting and underfitting) we can make our model is both generalised and robust.