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?

In **Ridge Regression**:- While we plot the curve between negative mean absolute error (MAE) and alpha we can clearly see that 'as the value of alpha increases from 0' , ' the error term decrease and the train error is showing increasing trend when value of alpha increases'

When the value **of alpha is 2** the test error is **minimum** so we decided to go with value of alpha equal to **2 for ridge regression model.**

In **Lasso regression** it is a very small value that is **0.01**, when we increase the value of alpha the model try to penalize more and try to make most of the coefficient value zero. Initially it is 0.4 in negative mean absolute error(MAE) and alpha.

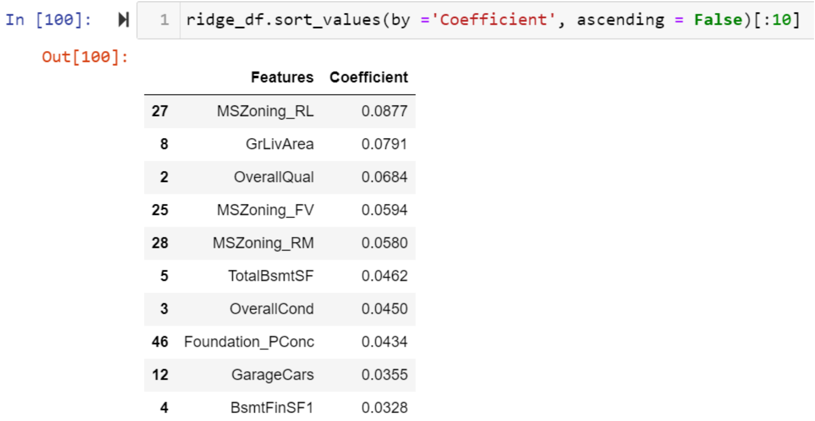
When we double the value of alpha in ridge regression and the value of alpha equal to 10 the model applies more penalty on the curve and try to make the model more generalized that is making model more simpler and it will not think to fit every data of the data set .

Also from the graph we can see that when alpha is 10 we get more error for both test and train data.

Similarly when we increase the value of alpha for lasso we try to penalize more our model and more coefficient of the variable will reduced to zero, when we increase the value of our r2 square also decreases.

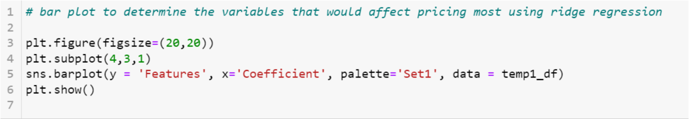
The most important variable after the changes has been implemented for ridge regression are as follows:- 1. MSZoning_RL 2. GrLivArea 3. OverallQual 4. MSZoning_FV 5. MSZoning_RM

6.TotalBsmtSqft 7. OverallCond 8 Foundation_PConc 9. GarageCaes 10. BsmntFinSF1



```
In [100]:  ridge_df.sort_values(by ='Coefficient', ascending = False)[:10]
Out[100]:
```

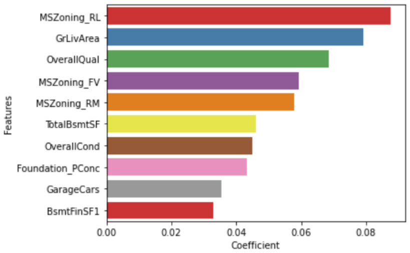
	Features	Coefficient
27	MSZoning_RL	0.0877
8	GrLivArea	0.0791
2	OverallQual	0.0684
25	MSZoning_FV	0.0594
28	MSZoning_RM	0.0580
5	TotalBsmtSF	0.0462
3	OverallCond	0.0450
46	Foundation_PConc	0.0434
12	GarageCars	0.0355
4	BsmtFinSF1	0.0328



```
# bar plot to determine the variables that would affect pricing most using ridge regression

plt.figure(figsize=(20,20))
plt.subplot(4,3,1)
sns.barplot(y = 'Features', x='Coefficient', palette='Set1', data = temp1_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?

- It will mainly depend on the use case
- If we have too many variables and one of our primary goal is feature selection, then we will use **Lasso**.
- If we don't want to get too large coefficients and reduction of coefficient magnitude is one of our prime goals, then we will use **Ridge Regression**.
- The optimal lambda value in case of Ridge and Lasso is as below: Ridge - 10 Lasso - 0.0004

· The Mean Squared error in case of Ridge and Lasso are: Ridge - 0.013743

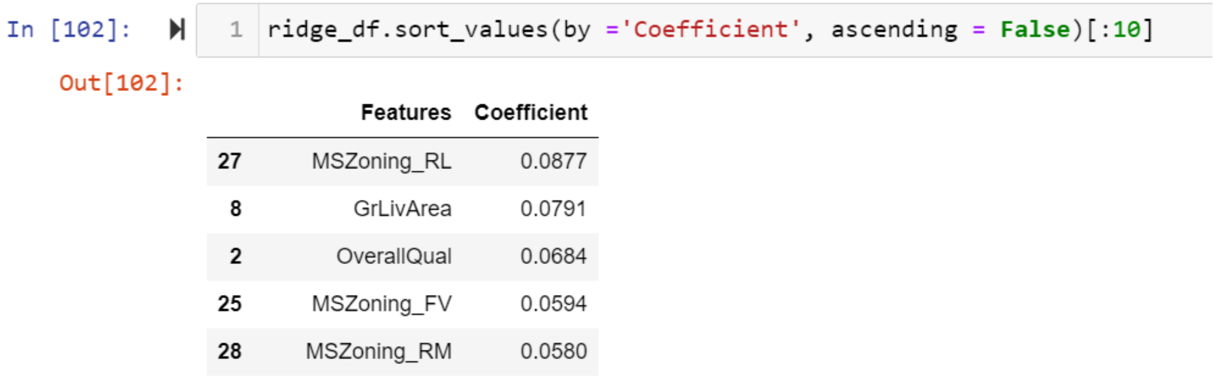
Lasso - 0.013556

· The Mean Squared Error of Lasso is slightly lower than that of Ridge.

· Also, since Lasso helps in feature reduction (as the coefficient value of one of the feature became 0), Lasso has a better edge over Ridge.

Therefore, the variables predicted by Lasso can be applied to choose significant variables for predicting the price of a house.

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 [102]:

```
ridge_df.sort_values(by ='Coefficient', ascending = False)[:10]
```

Out[102]:

	Features	Coefficient
27	MSZoning_RL	0.0877
8	GrLivArea	0.0791
2	OverallQual	0.0684
25	MSZoning_FV	0.0594
28	MSZoning_RM	0.0580

1MSZoning_RL 2.GrLivArea 3.OverallQual 4. MSZoning-FV 5. MSZoning_RM

- A model can be called as a **robust model** when any variance in data doesn’t bring down the required performance
- A **generalizable** model is that kind of model which is able to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model.
- To satisfy the above two conditions we have to **ensure that it doesn't overfit**.
- In other words, the model should not be too complex in order to be robust and generalizable.

- In general, we have to find a healthy balance between model accuracy and complexity. This can be achieved by Regularization techniques like Ridge Regression and Lasso.