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 regression is 100

The optimal value of alpha for lasso regression is 500.

The top 10 important predictor variables at optimal value of alpha are:

Top 10 important features

In [66]:
```
a = pd.DataFrame(X_train.columns)
b = pd.DataFrame(lasso.coef_)
```

In [67]:
```
coefficients = pd.concat([a,b], axis=1)
coefficients.columns = ['Features','Coefficient']
```

In [68]:
```
coefficients.sort_values(by='Coefficient', ascending=False).head(10)
```

Out[68]:

	Features	Coefficient
13	2ndFlrSF	21200.198441
12	1stFlrSF	17926.455190
3	OverallQual	12912.818963
5	YearBuilt	8488.949606
11	TotalBsmtSF	8296.599102
8	BsmtFinSF1	6175.124465
2	LotArea	6064.669699
23	GarageArea	5573.551877
4	OverallCond	5389.654500
237	SaleType_New	5287.097342

Upon doubling the value of alpha:

Test r2 score decreased by 1% because more features were eliminated. Initially, we had 119 features and after doubling the alpha, we had 80 features only.

The difference in r2 score between train and test dataset also shrinks slightly.

The MSE / RMSE values went up and the magnitude of coefficients decreased. Total Basement square feet and Basement finish SF1 climbed to number 4 and 5 and Year built dropped to number 6. The top 3 features remain same.

Top 10 important features

In [116]:
```
a = pd.DataFrame(X_train.columns)
b = pd.DataFrame(lasso.coef_)
```

In [117]:
```
coefficients = pd.concat([a,b], axis=1)
coefficients.columns = ['Features','Coefficient']
```

In [118]:
```
coefficients.sort_values(by='Coefficient', ascending=False).head(10)
```

Out[118]:

	Features	Coefficient
13	2ndFlrSF	18733.811791
12	1stFlrSF	16663.880089
3	OverallQual	15826.690386
11	TotalBsmtSF	7774.597126
8	BsmtFinSF1	6454.597870
5	YearBuilt	6393.599252
23	GarageArea	5862.670992
2	LotArea	5665.411545
237	SaleType_New	4915.685707
66	Neighborhood_NridgHt	4557.296610

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:

Between Lasso and Ridge, I'll choose the Lasso regression model because the test r2 score was little bit higher on Lasso model than Ridge model. Also, the RMSE value was lower with Lasso regression.

With optimal lambda value, we had 119 features and after doubling the lambda, we had 80 features only at the cost of 1% r2 score and slight decrease in different between train and test r2 score.

Doubling the lambda in our case, will generalize little better on unseen data than optimal value of lambda.

Another thing to keep in mind is that Lasso model is computationally heavy. It depends on current company's computational power if Lasso is the right model for them because there is not significant difference between r2_score and RMSE of Lasso with 119 features, Lasso with 80 features, and Ridge model.

Question 3

After building the model, you realized 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:

After removing top 5 features, the new model's top 5 features are Basement finish SF1, Basement unfinish SF, Total rooms above ground, garage area, and full bath. The r2 score decreased by 2% and RMSE increased by $1200.

In [156]:
```
coefficients = pd.concat([c,d], axis=1)
coefficients.columns = ['Features','Coefficient']

coefficients.sort_values(by='Coefficient', ascending=False).head(10)
```

Out[156]:

	Features	Coefficient
6	BsmtFinSF1	20435.829953
8	BsmtUnfSF	14814.499322
16	TotRmsAbvGrd	12940.818728
18	GarageArea	9214.684732
12	FullBath	6399.187228
2	LotArea	6302.138177
60	Neighborhood_NoRidge	6140.332621
7	BsmtFinSF2	5993.498352
67	Neighborhood_StoneBr	5930.371643
159	BsmtExposure_Gd	5685.078726

Question 4

How can you make sure that a model is robust and generalizable? What are the implications of the same for the accuracy of the model and why?

Answer 4:

By doing train test split, we measure the training model on test data set (unseen). If the difference between train and test accuracy score is very wide, let's say > 0.10, the model seems to be overfitting and will not generalize well. The model will learn train data set perfectly and won't able to perform well on unseen data.

Contrary to it, the closer the difference in train and test accuracy score, the more generalizable is the model.

R2 score denotes how much variation in a model is explained by current set of features. Generally speaking, the test r2 score should be lower than train r2 score. The closer the gap between them, better is the model.

In our current assignment, the r2 score on train and test data set is very close (0.93 on train and 0.91 on test) with a significant number of feature elimination. This makes our model more robust and generalizable.