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?

Ans:
Ridge Alpha: 6.0
Lasso Alpha: 0.001

After changing the Ridge and Lasso alpha value, Top5 predictor variables are

Ridge:

```
 ('SaleCondition_AdjLand', 0.06),
 ('SaleCondition_Alloca', 0.06),
 ('SaleCondition_Family', 0.061),
 ('SaleCondition_Normal', 0.07),
 ('SaleCondition_Partial', 0.107)
```

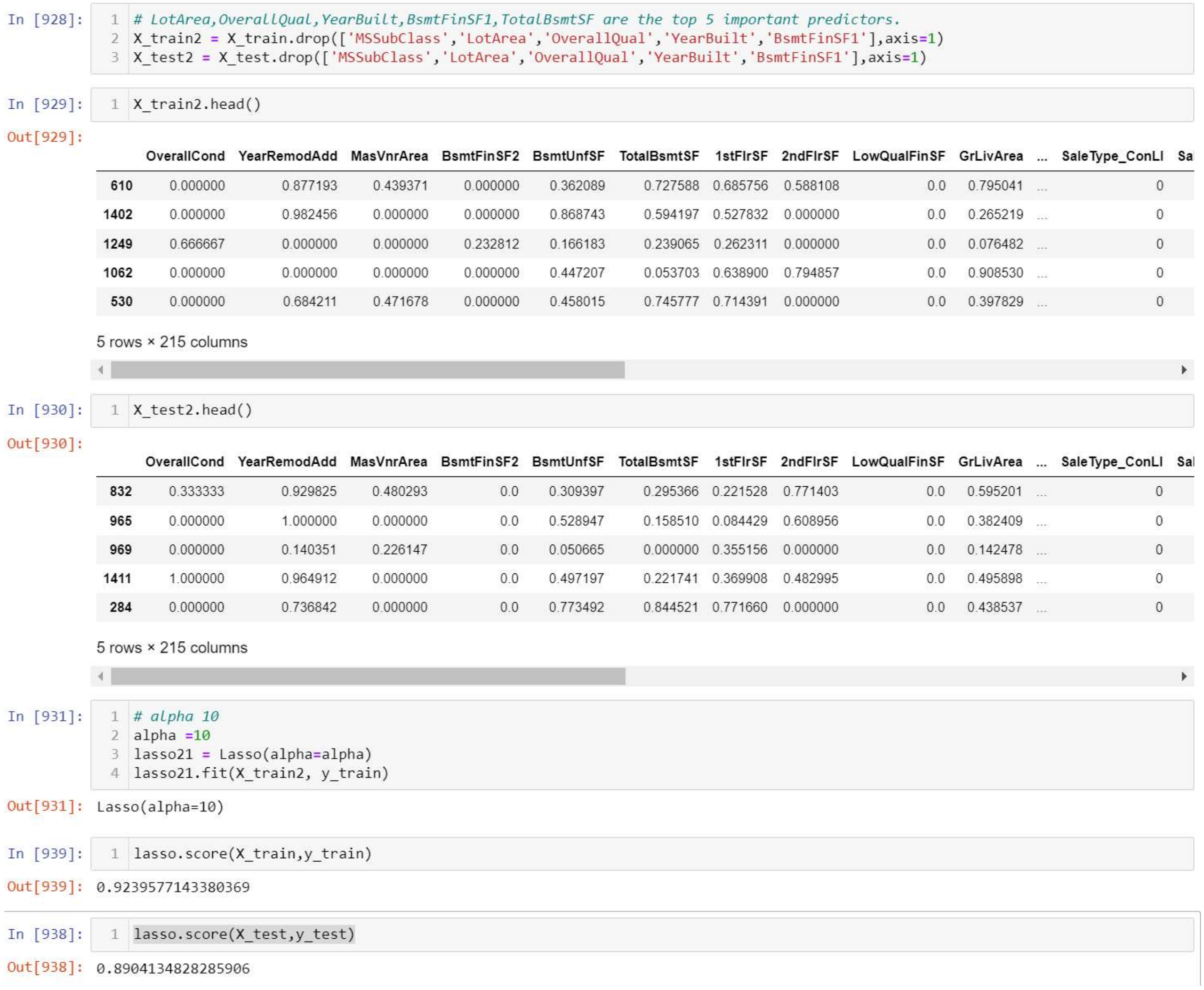
Lasso

```
 ('SaleCondition_AdjLand', 0.031),
 ('SaleCondition_Alloca', 0.035),
 ('SaleCondition_Family', 0.058),
 ('SaleCondition_Normal', 0.085),
 ('SaleCondition_Partial', 0.108)
```

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?

Ans: Lasso R-squared test value is greater than Ridge value, hence we can choose Lasso for module evulation

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 [928]:

```
# LotArea,OverallQual,YearBuilt,BsmtFinSF1,TotalBsmtSF are the top 5 important predictors.
X_train2 = X_train.drop(['MSSubClass','LotArea','OverallQual','YearBuilt','BsmtFinSF1'],axis=1)
X_test2 = X_test.drop(['MSSubClass','LotArea','OverallQual','YearBuilt','BsmtFinSF1'],axis=1)
```

In [929]:

```
X_train2.head()
```

Out[929]:

	OverallCond	YearRemodAdd	MasVnrArea	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	LowQualFinSF	GrLivArea	...	SaleType_ConLI	Sa
610	0.000000	0.877193	0.439371	0.000000	0.362089	0.727588	0.685756	0.588108	0.0	0.795041	...	0	
1402	0.000000	0.982456	0.000000	0.000000	0.868743	0.594197	0.527832	0.000000	0.0	0.265219	...	0	
1249	0.666667	0.000000	0.000000	0.232812	0.166183	0.239065	0.262311	0.000000	0.0	0.076482	...	0	
1062	0.000000	0.000000	0.000000	0.000000	0.447207	0.053703	0.638900	0.794857	0.0	0.908530	...	0	
530	0.000000	0.684211	0.471678	0.000000	0.458015	0.745777	0.714391	0.000000	0.0	0.397829	...	0	

5 rows × 215 columns

In [930]:

```
X_test2.head()
```

Out[930]:

	OverallCond	YearRemodAdd	MasVnrArea	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	LowQualFinSF	GrLivArea	...	SaleType_ConLI	Sal
832	0.333333	0.929825	0.480293	0.0	0.309397	0.295366	0.221528	0.771403	0.0	0.595201	...	0	
965	0.000000	1.000000	0.000000	0.0	0.528947	0.158510	0.084429	0.608956	0.0	0.382409	...	0	
969	0.000000	0.140351	0.226147	0.0	0.050665	0.000000	0.355156	0.000000	0.0	0.142478	...	0	
1411	1.000000	0.964912	0.000000	0.0	0.497197	0.221741	0.369908	0.482995	0.0	0.495898	...	0	
284	0.000000	0.736842	0.000000	0.0	0.773492	0.844521	0.771660	0.000000	0.0	0.438537	...	0	

5 rows × 215 columns

In [931]:

```
# alpha 10
alpha =10
lasso21 = Lasso(alpha=alpha)
lasso21.fit(X_train2, y_train)
```

Out[931]:

```
Lasso(alpha=10)
```

In [939]:

```
lasso.score(X_train,y_train)
```

Out[939]:

```
0.9239577143380369
```

In [938]:

```
lasso.score(X_test,y_test)
```

Out[938]:

```
0.8904134828285906
```

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?

Ans:

- Model should be generalized so that test accuracy is not lesser than the training score
- Extreme Outliers should be taken care as part of the data cleanup
- Label Encoding should be done