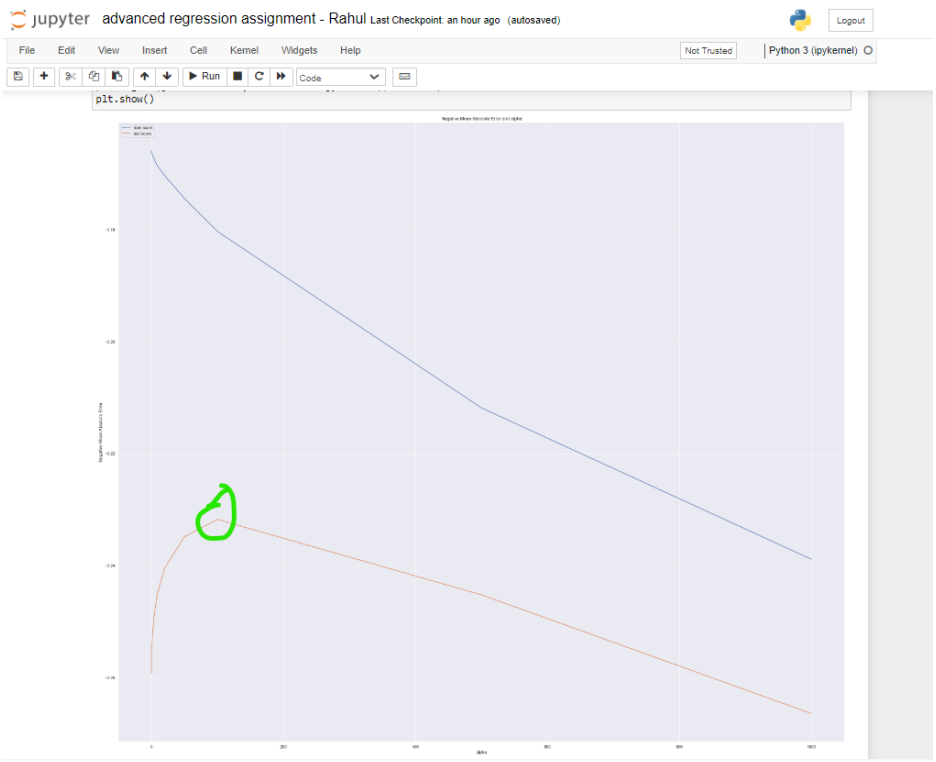
What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Ans :

Ridge Regression :



Here based on plot it is clear that when alpha is 100 then it gives us optimum values.

If we double the value of alpha there is small decrease in the accuracy.

Alpha : 100

```
# ridge regression
lm = Ridge(alpha=100)
lm.fit(X_train, y_train)

# predict
y_train_pred = lm.predict(X_train)
print(metrics.r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred = lm.predict(X_test)
print(metrics.r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
0.927486169464771
0.9087381310522238
```

Alpha : 200

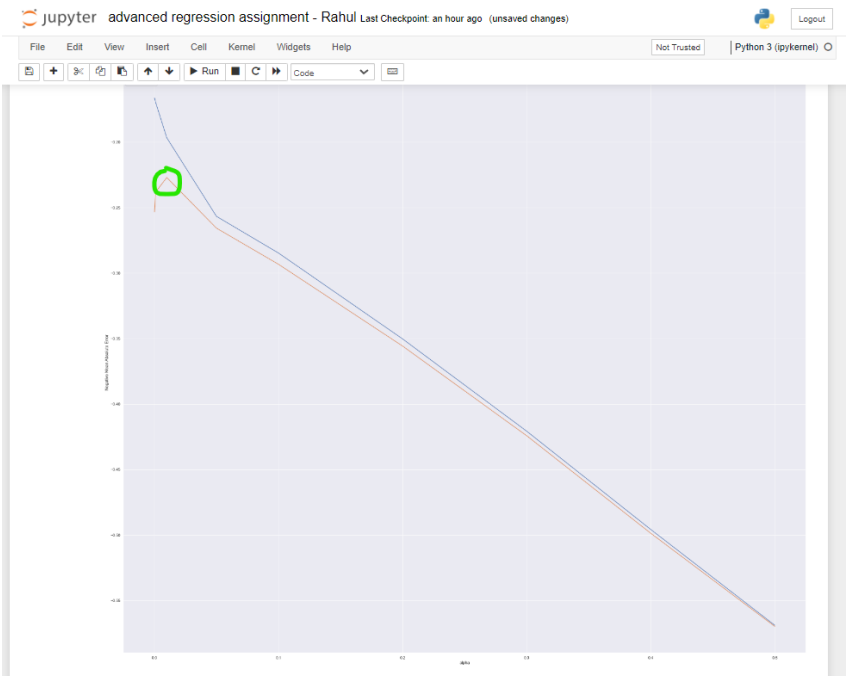
```
# ridge regression
lm = Ridge(alpha=200)
lm.fit(X_train, y_train)

# predict
y_train_pred = lm.predict(X_train)
print(metrics.r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred = lm.predict(X_test)
print(metrics.r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
0.9204142002262414
0.9012409065675521
```

Lasso Regression:

For Lasso Regression optimum alpha we get is 0.02 as per the plot created based on alpha and negative mean absolute error.



In [412]:
```
# lasso regression
lm = Lasso(alpha=0.02)
lm.fit(X_train, y_train)

# prediction on the test set(Using R2)
y_train_pred = lm.predict(X_train)
print(metrics.r2_score(y_true=y_train, y_pred=y_train_pred))
y_test_pred = lm.predict(X_test)
print(metrics.r2_score(y_true=y_test, y_pred=y_test_pred))
```

```
0.8964945120028462
0.8918894915144959
```

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 :

Ridge gives better accuracy but even lasso is not far behind the Ridge. There is very small difference when r2 score is measured. 0.9012 in case of Ridge while 0.8918 in case of Lasso.

With optimal value of alpha for lasso, it converts unnecessary coefficients to 0 giving us only 67 variables to predict the target variable, While Ridge regression gives 238 non zero coefficients. This will largely simplify and generalize the model.

So We can go with the Lasso.

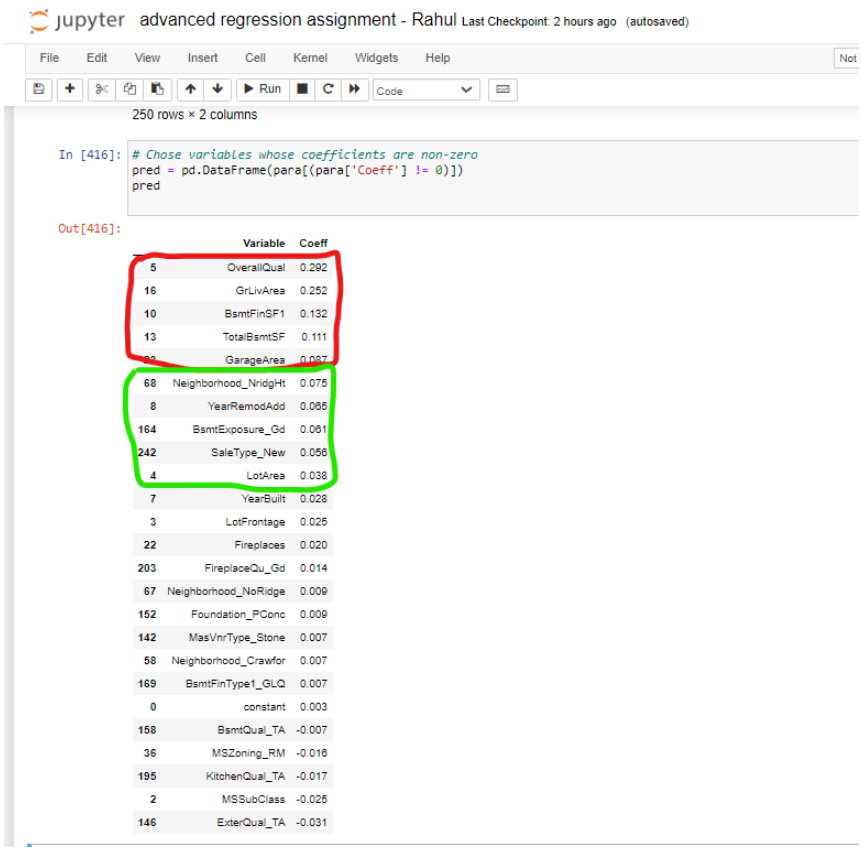
Question 3

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?

Ans :

Below screenshot shows the five most important predictor variables as:

5	OverallQual	0.292
16	GrLivArea	0.252
10	BsmtFinSF1	0.132
13	TotalBsmtSF	0.111
23	GarageArea	0.087



If those are removed, then next 5 most important variables are highlighted in green :

8	Neighborhood_NridgHt	0.075
8	YearRemodAdd	0.065
164	BsmtExposure_Gd	0.061
242	SaleType_New	0.056
4	LotArea	0.038

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 : Three ways that I can think of which make sure that model is robust and generalisable as :

1. Cross-Validation: Use k-fold cross-validation. By training and testing the model on different subsets of the data multiple times, you get a more reliable estimate of its performance on unseen data.

2. Train/Test Split: Always have a separate test set that the model has never seen during training or hyperparameter tuning. This gives an unbiased evaluation of its real-world performance.

3. Regularization: Techniques like Lasso and Ridge Regression can help prevent overfitting by adding penalty terms to the loss function, discouraging overly complex models.

Ensuring robustness often involves balancing trade-off between Bias and Variance. While reducing variance might lead to a slight decrease in training accuracy, it can significantly improve test accuracy.