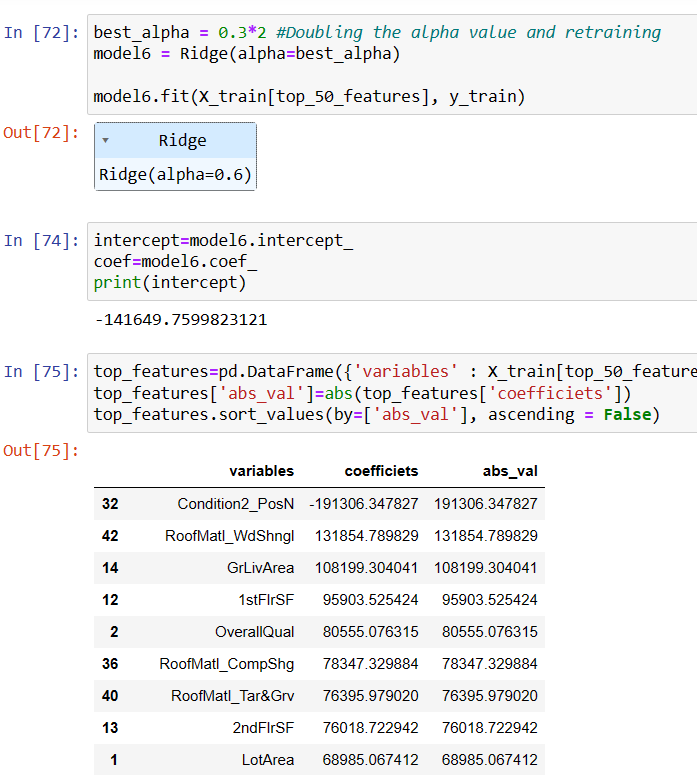
**Assignment - Part 2**

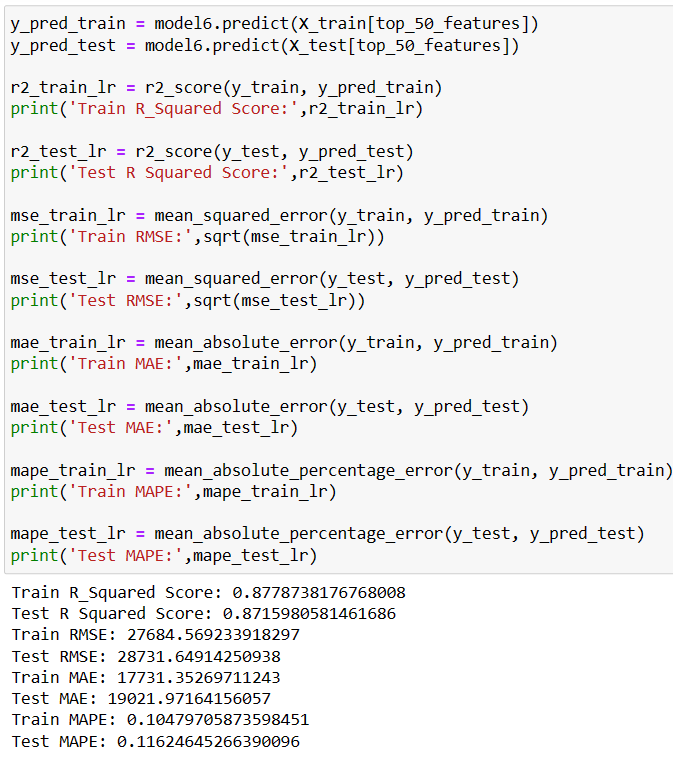
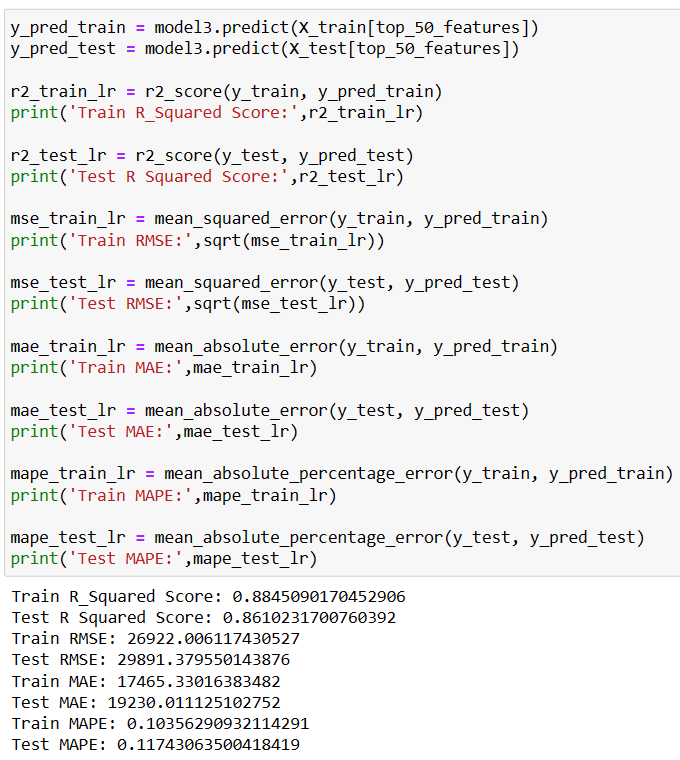
1. What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

After doing hyperparameter tuning for both Ridge and Lasso I got the optimal alpha values of 0.3 and 100 respectively. If we double the values of both the hyperparameters we should see the Ridge Regression coefficients taking slightly lower absolute values due to the doubling of the hyperparameter, due to the fact that the weights would be penalized more due to higher value of regularization parameter. The Lasso Regression would also see similar trend but apart from this, it’s coefficient space would become even sparser as more of the coefficients would become zero. Also, I do feel that the top features would remain same in case of lasso as it is gives feature importance as well.

**RIDGE**

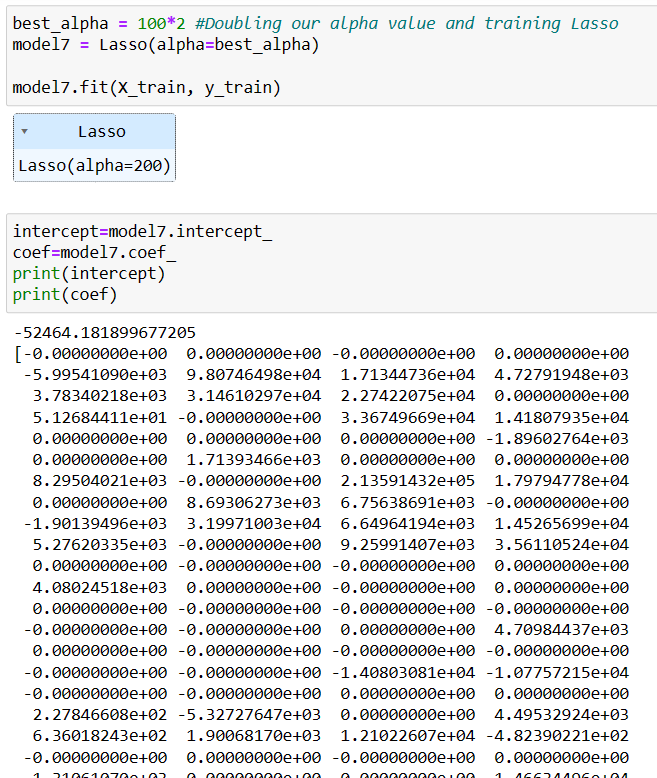
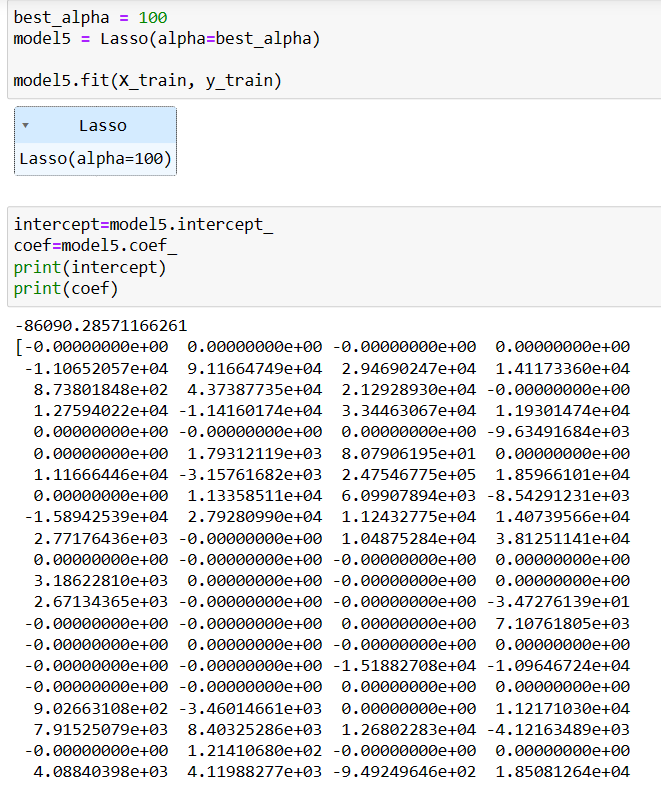


As expected, we can clearly see that when we double our Ridge regularization Hyperparameter. The overall coefficients have reduced their values, and a few of the parameters have also flipped spots in scale of importance.

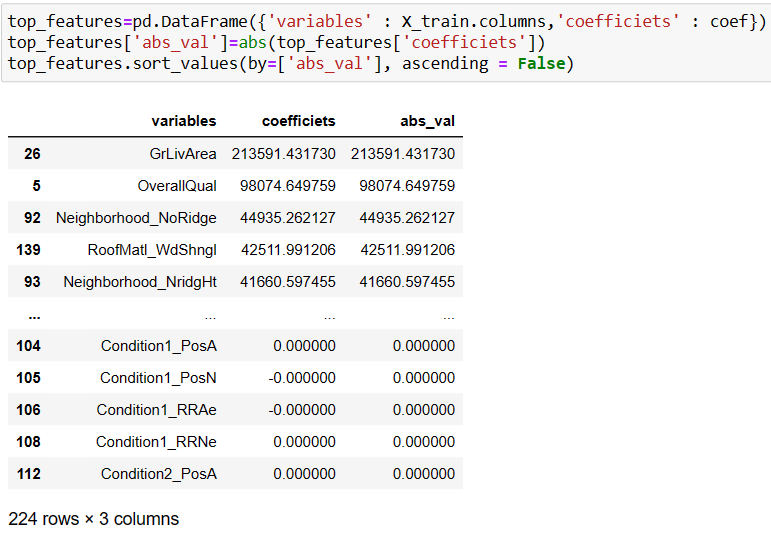
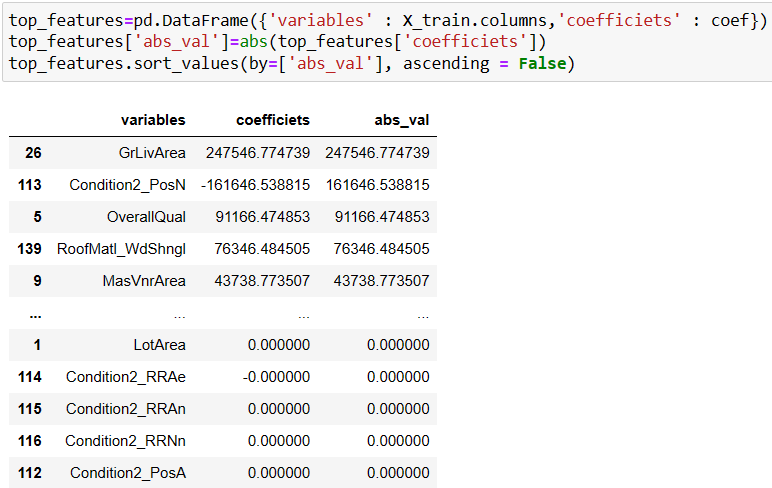


Also looking at the metrics the train accuracy has decreased very slightly but the model is performing slightly better on the test set now. But not a lot of change.

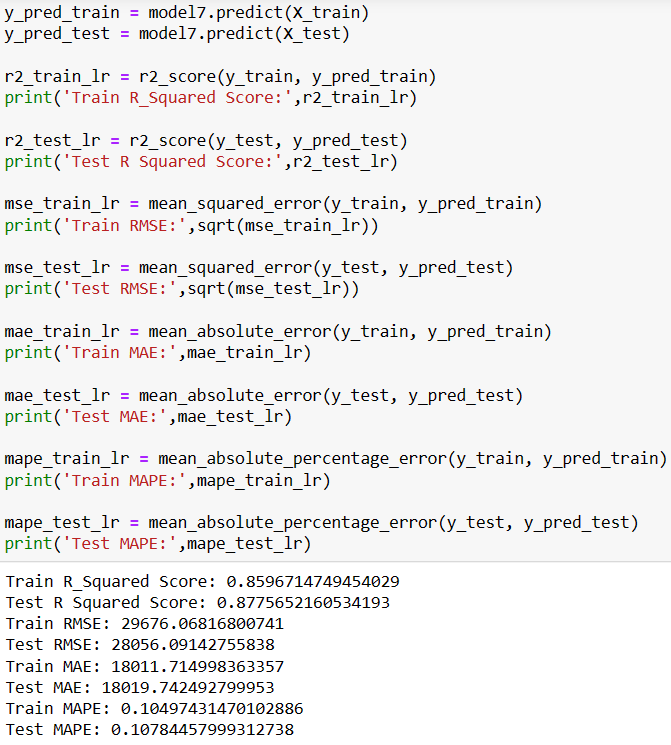
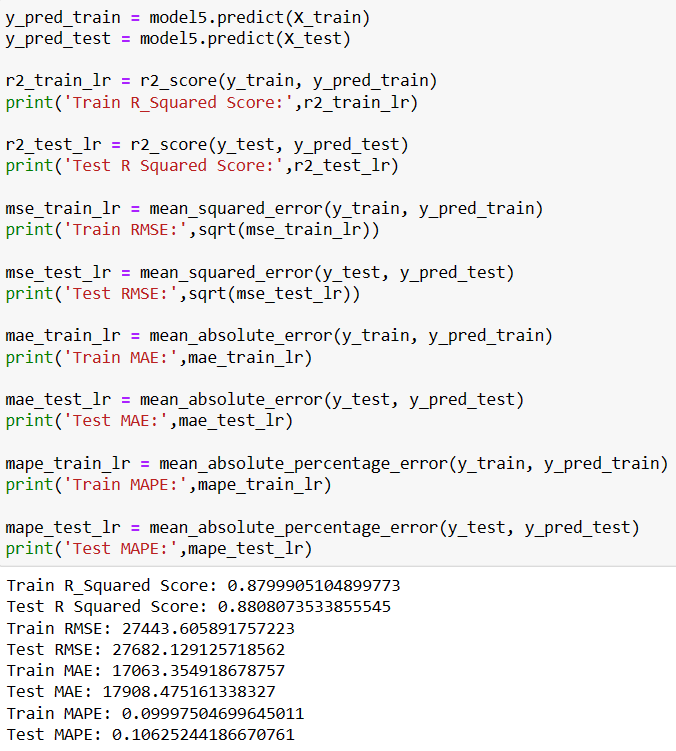
**LASSO**

****

As we were expecting a few more of the coefficients have become zero making the weights sparser.



Top features have flipped here as well which I was not expecting. I thought that it would not happen in lasso as it gives feature importance and they would decrease in the order of the coefficients assigned. Good Experiment to know this.



The metrics have also remained quite similar with very slight decrease after we doubled alpha.

1. 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?

I got alpha value of 0.3 in ridge regression using GridSearchCV but, I still chose my default Ridge Regression with alpha value of 1 as it seemed to be generalizing well to the test set. At alpha value 0.3 it was overfitting to the train set.

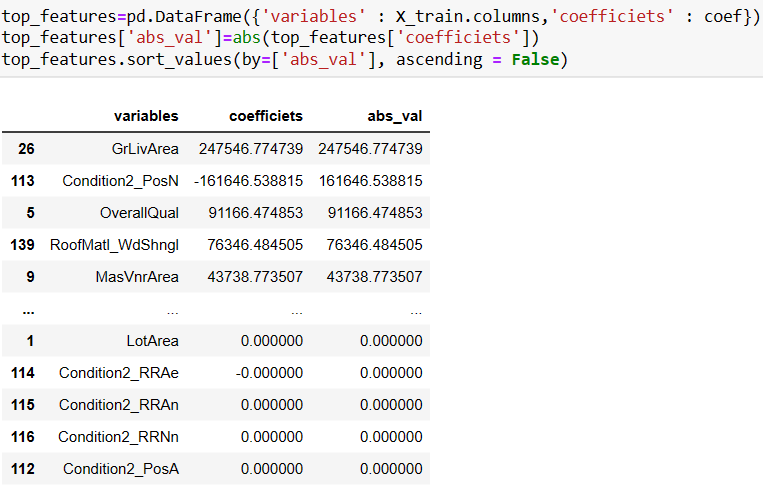
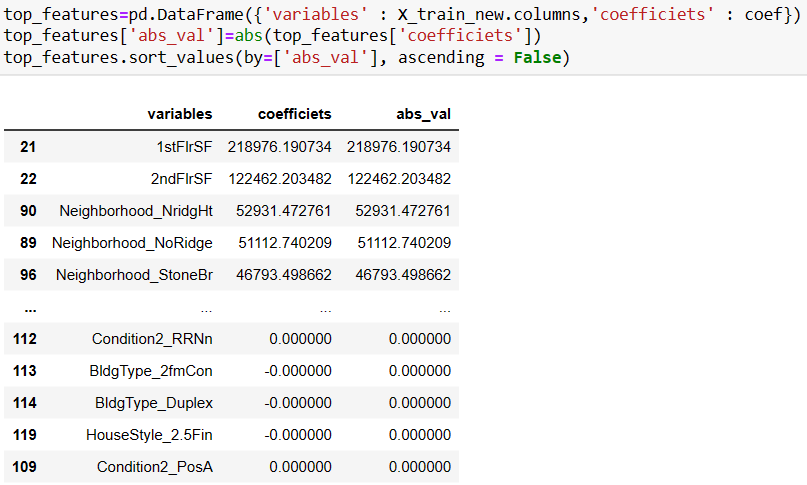
For Lasso I did find that it was giving better results on test set, than (default alpha value) at alpha value of 100. So, I chose it as my final model.

I would use the Test set metrics to pick my model any day if I see that both Train and Test metrics are very similar (meaning my model is generalizing well).

1. 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?

This question is quite experimental so let’s try it out by dropping our 5 most important predictors in Lasso Model.

The updated 5 most important predictors after having dropped top 5 predictors as shown in the right image, left image showing the previous 5 important predictors.

1. 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?

To make our model robust and generalisable one needs to have a good balance between overfitting and underfitting (bias variance trade-off).

One would get robust model if one archives good training accuracy but, in some cases, it might not reflect on the test data. This is clear case of overfitting. One needs to also try and bring the test accuracy on par with the train accuracy (as close as possible) to get a generalizable model.

In other words, in high variance situation our model is not only learning patterns that are useful but it is so complex or trained for such long periods that it trying to learn noise in the data as well. Then later on would not predict well in case of test data.

While in case of high bias situation we might have selected not optimized (not properly trained), very simple model that is not capable of modelling complex data. In this case both train and test accuracy will be very low.

The implications of the same on model metric in case of Overfitting would be unreasonably high metric scores on train data, and very low scores on test data. While in case of Underfitting it would give bad metric scores for both train and test data. To identify these one should always hold out some train data that the model has never seen for testing the model on unseen data points.