Sahil Rangwala
MSDS422

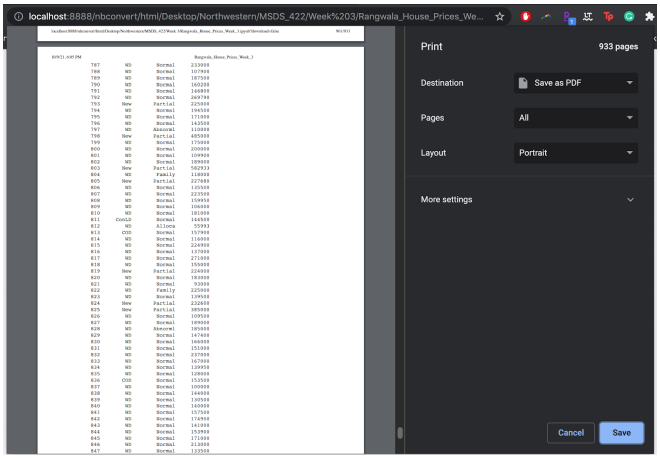
Assignment 3 Write-Up

Once we clean the data, it showcases which features are relevant towards the model. Last week, the linear regression model offered a decent source for prediction for SalePrice. This was corroborated by both the cross validation methods via the error scores as well as the hidden test cases offered by Kaggle.

In this assignment, we explored 3 new models such as the Ridge, Lasso, and the ElasticNet. In both Lasso and ElasticNet, I needed to tune the hyperparameters because the defaulted parameters would cause a skewness that would mask the entire dataset accidentally. We found that via the errors provided, it seemed like Ridge would perform the worst in comparison to Lasso and ElasticNet. Further graphical modeling is necessary to understand the reasoning on why the ElasticNet fared better than the Lasso modelling technique. I believe it is due to the nature of ElasticNet that it would cause a more specific and accurate result.

ElasticNet was my predicted best model as it is a hybrid between Lasso and Ridge where lacking elements would be made up by the other. The overall breakdown of the three modeling techniques is that Lasso modeling will help eliminate many features, and overall reduce overfitting the linear model. Furthermore, Ridge will reduce the impact of unimportant features that do not offer strength in predicting the y values. Additionally, ElasticNet combines the technique of feature elimination from Lasso and the coefficient reduction from the Ridge model to provide a best of both world approach in predicting the models.

In the PDF of the code, some dataframe information was relevant to see while coding as it was more interactive within the Jupyter Notebook to manipulate the HTML file to see the overall effects of the dataset. However, if I tried printing each line, per the request of the assignment, it would cause the PDF to be abnormally large and would bring down the overall readability of the file. Due to this, I chose not to showcase all of the dataframe data in the print lines. If I kept the data frame prints, it would cause the PDF to be about 993 pages long.



It is great that Kaggle offers hidden test cases that rigorously test the submission csv files. Per my predictions, I would assume ElasticNet performs the best and it could be a toss up between Ridge and Lasso based on how I cleaned the dataset to remove unnecessary features or

constraints. The findings of all three models are below, where it confirms my prediction of ElasticNet performing the best. Surprisingly, the gradient boosting model technique I used last week instead of gradient descent offered a comparable result to ElasticNet. I hope a future assignment would help explore the key differences of these two techniques as I did not expect the strength of Gradient boosting to be this high. The Kaggle scores of all three modeling techniques are below. Not surprisingly, my Ridge score did better than the generic linear model however I expected a larger improvement.

submission_elastic.csv just now by Sahil Rangwala elastic	0.20991
submission_lasso.csv a few seconds ago by Sahil Rangwala lasso	0.21389
submission_ridge.csv a minute ago by Sahil Rangwala add submission details	0.38284

The hidden test cases were great as it allowed us to play with different implementations and see if it would fare in comparison to our previous results. This was an instance of a horrible ridge solution which I did not realize until the hidden test case told us it was not a feasible solution.

house_pricing_submission.csv an hour ago by Sahil Rangwala possible ridge solution	11.64006
submission.csv 6 days ago by Sahil Rangwala Gradient boosting model	0.20537
submission.csv 6 days ago by Sahil Rangwala Linear regression model on test data	0.38331

A peculiar item I saw was copying cells for modeling I received this error, could be the reason why the graphs look so similar/exactly the same but the error values from the cross-validation is used to help distinguish strength differences of the models.

```
[E 16:00:11.634 NotebookApp] Notebook JSON is invalid: Non-unique cell id '073e2
c94' detected. Corrected to '6ba7dd78'.
```

Additionally, the pdf is slightly larger than what is expected - I had a difficult time trying to make the feature plots as subplots because I was iterating and caused some errors due to trying to

configure a subplot. Instead, there is a lot of dead white space I could not remove. Ideally for next week, all this copied code I will pull out and make it into a method thus it will save on space in the PDF. It would require a lot of refactoring that time does not permit for and I am worried my results would change accidentally from hasty refactoring. Sadly, it is the last step in the assignment process to see how long a PDF would be, thus it is difficult to gauge the size of the PDF if we would print it within the Jupyter Notebook. I downloaded the Ruler extension to help with code readability, however, I hope there is an extension that exists that could estimate the size of the PDF file before trying to print it through Chrome.