Midterm Report

Duc Minh Nguyen – U17531382

Kaggle User Name: Minh Duc

1. Preliminary Analysis / Exploration:
   Exploration was done on the whole dataset, below are the steps I followed to understand the dataset:

   - Understanding the numeric data columns from the basic plots provided in the starter code.
   - I tried using the data with less reviews with score of 5 but the models ended up performing worse than with the whole dataset.

   - There are several nan values in the text / summary dataset, which has been filled with empty string values.

   - I created a word by review TFIDF matrix, each entry in this matrix represents the number of times a particular word (defined by the column) occurs in a review.

   - From this matrix, I use TruncatedSVD to create a 2-dimensional representation of each review and plot it for visualization with: purple color being score of 5; orange is 4, red is 3; blue is 2; green is 1. The visualization was not that helpful as all the points just got stuck together.

   - Multiple ML models were tested for the text, and summary analysis, and the performance from those models were quite different from each other. The Ridge performs the best from my examination. Some models that I tried were RandomForrestClassifer, Ridge, LGBMRegressor.

2. Feature Extraction:

- The first idea I thought of was sentiment analysis, getting the sentiment score of a review I think would be an important feature as a review being negative will most likely mean a really low score and vice versa. I used nltk pretrained model, SentimentIntensityAnalyzer to get the positive, negative, and compound sentiment score of a review. Then add all three of those columns to the dataset.
- Helpfulness of the review, which is the HelpfulnessNumerator divided by HelpfulnessDenominator
- Average Helpfulness for each user which may help if this user usually give review that are helpful or not. (I ended up not using this feature)
- TFIDF matrix: I combine the Summary column with the Text column as SummaryReview column and convert that column to a matrix of TF-IDF features. I experimented with the ngram range variable and find out that ngram=(1,4) works really well.

- Length of the review, a long review may correlate with bad scores but I ended up not using this feature in my model.
- Product count: the number of occurrences a product appear in the dataset may implies users have something good or bad to say about it. (I ended up not using this feature as well)
- Day, Month, Year extracted from the Time column.

The more experiment I did, I used less features than I originally planned. I ended up not using any of the time data after not seeing them affecting my model scores. I tried stemming the text for Text column and Summary column then apply TFIDF vectorizer as described above but that also did not change much of the score so I stick with the regular Summary and Text column (Maybe because I believe TFIDFVectorizer automatically do the stemming). The main feature I ended up using are: TFIDF matrix, Helpfulness, HelpfulnessNumerator, HelpfulnessDenominator, compound sentiment scores, negative sentiment scores and positive sentiment scores.

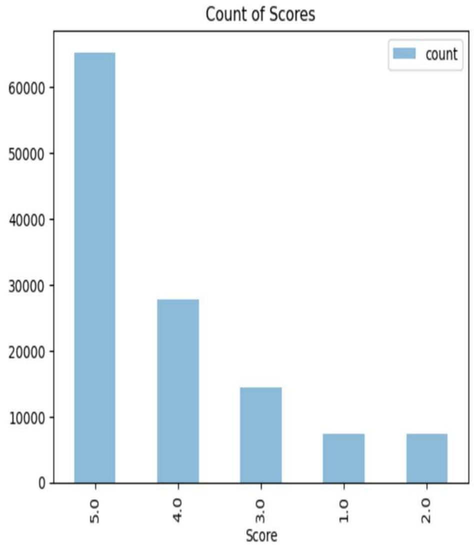

Figure 1: Scores' Distribution

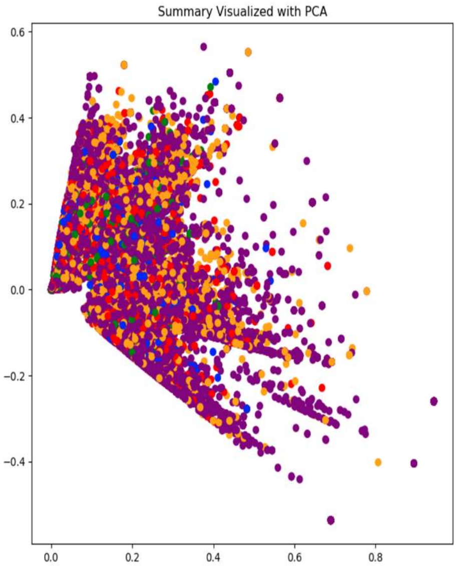

Figure 2: 2D Summary Representation

3. Workflows, decisions, techniques:

- Workflows: I separated data exploration for **exploration.ipynb** and **starter.ipynb** notebooks and I tried testing my models in **Examine.ipynb** notebook. The main feature that I used as a pillar is the TFIDF matrix. Then I will add each feature onto that separately to see if a model is improving or regressing.
- Decisions: I started of trying RandomForrestClassifier with the TIFIDF matrix as the feature but the result it showed was not satisfactory even after adding all of the other features I described above. I then think of the problem as a regression problem instead

of classification because the goal is also to try and lower the RMSE. I then look for regression model instead. One of the model that came up was Light Gradient Boosting Machine which gave a decent result using only the TFIDF matrix of RMSE score of 0.86. But I failed to improve the model after that no matter how many features I added on and the training time was taking too long so I looked for another solution (I ended up scrapped the LGB model from the final submission). Then I tried Ridge Regression model, which is a linear regression model with L2 regularization. It's a variant of linear regression that helps mitigate the problem of multicollinearity (high correlation between independent variables) and overfitting by adding a penalty term to the linear regression cost function. The model performed extremely well with the first experiment getting RMSE of 0.81 and I chose Ridge as the foundation to finetuning and testing features on. I also tried Bagging Regressor which is a type of bagging technique that combines multiple regression models to improve the overall predictive performance and reduce the variance of the predictions. The final model I ended up using was Ridge Regression.

- Techniques: From RMSE of 0.81 with the TFIDF matrix, I tried adding on Helpfulness, HelpfulnessNumerator, HelpfulnessDenominator features, then the score went down to 0.78. After adding on compound sentiment scores, negative sentiment scores and positive sentiment scores the score went down to 0.76. At this point I have used all of the important features I have, adding other columns from the dataset do not improve the score. I start thinking about modifying the ngram range of TFIDF vectorizer to run from 1 to 4 so from unigram to quadgram. The score start improving further, getting down to 0.74. I want to get down to 0.73 so I try tuning the alpha variable of Ridge I was seeing improvement. I changed from alpha=2.5 to alpha = 0.009 (changing the multiplier of the regularization term) and got the result that I wanted which was 0.73197 for RMSE.

4. Challenges

- I spent first 2 days trying to solve this problem as a classification problem and my result did not improve at all, being stuck at RMSE of 1.2
- Figuring out which models to use was a very difficult task as some features may work well with this model but not work well with other models. I tried all the features I described in section 2 and it did not improve LGBM.
- Computation time takes a long time when experimenting with models, especially when I was not sure if the features I picked worked well with the model. For example, I tried computing TFIDF matrix for review then use TruncatedSVD to PCA the matrix to lower dimensionality which took a long time and fitting that into RandomForrest did not change the RMSE at all.