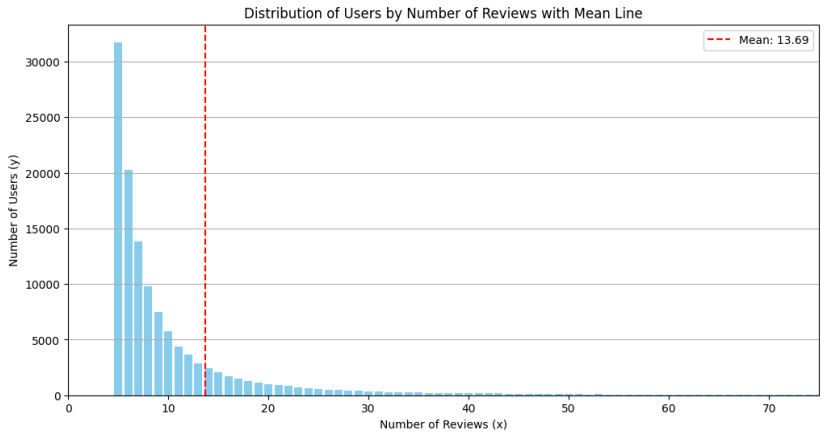
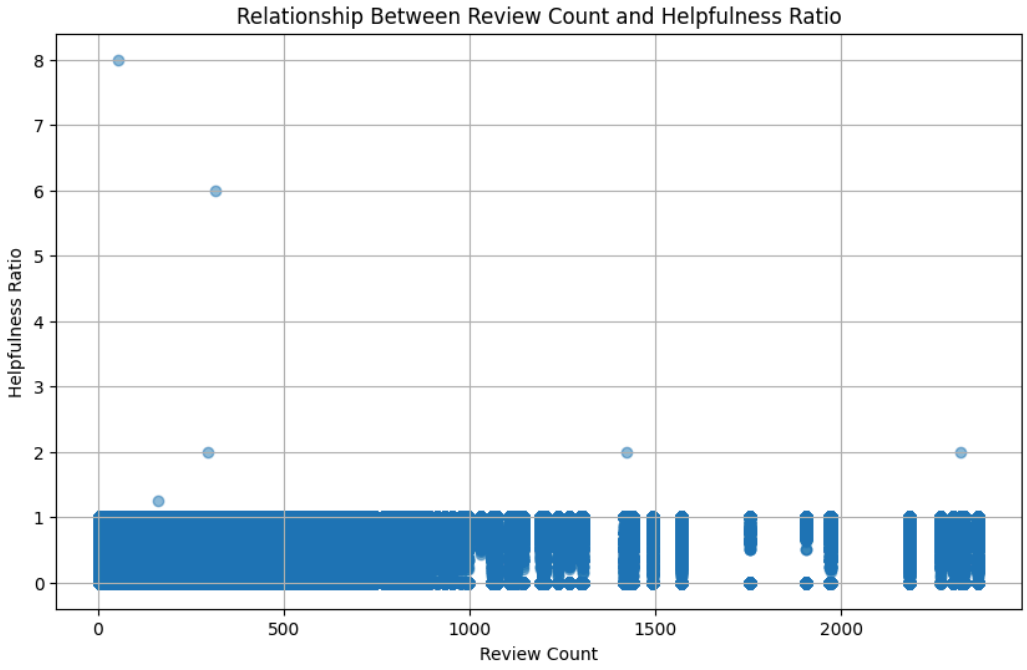
1) Exploring the dataset

So the first thing that I attempted to start this midterm was to explore what kinds of relationships I could find within the dataset. So for most of the midterm I was just looking at different variables and seeing if there was a relationship or something impactful that I could use for my model. I made graphs and visuals to aid me in understanding more about the dataset. For example I plotted this graph to see the number of reviews against the number of users that have that many reviews. I then used this to plot whether or not this had a relationship with whether the helpfulness of the review which turns out to be that it didn't. At first I thought that maybe there could be a relationship between the two since someone with more reviews may write more meaningful reviews. But I also thought maybe people with only few reviews leave reviews since they strongly dislike or like a product.





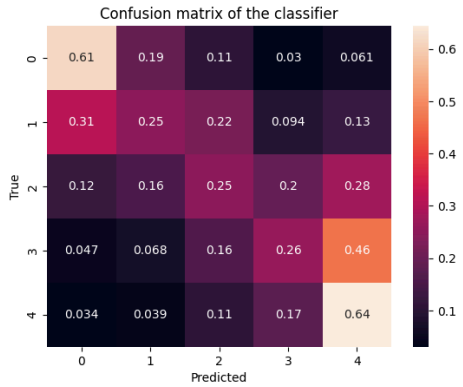
2) Adding features to dataframe

So the first one that I added was a sentiment analysis using VADER which I found from chat. It said that it was good for analyzing the sentiment of social media text but can also be effective with other short informal texts like product reviews. I used this on the text section only since I thought the summary section would not be as useful with this tool. The summaries were shorter and might not have any information pertaining to the score of the review but thinking back I think I should have tried it as well. I then used this to model the first model using knn and got around a 49-50% accuracy with it. Then I ran tests to find the best k value in range 1-100 which took way too long for an increase of just 1ish%. Something interesting I did was just do k = 500 and it just ended up classifying basically everything as a 5 star review. Something I also noticed was that most of the reviews in general were being classified as 5 stars. To fix this I added a negative word count and I also separated the VADER sentiment into negative, neutral and positive. I also added negative word count and neutral word count to try to differentiate between 5 stars and something like 4 stars and 3 stars.

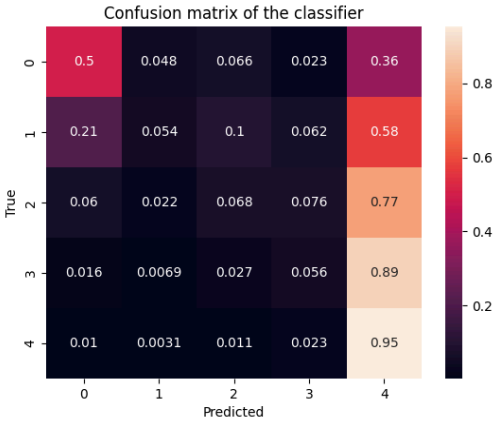
3) Modeling

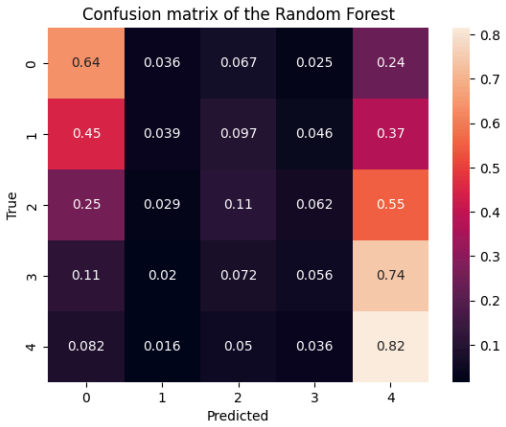
So for the modeling I tried 3 different methods, KNN, random foresting and logistic regression.

For logistic Regression



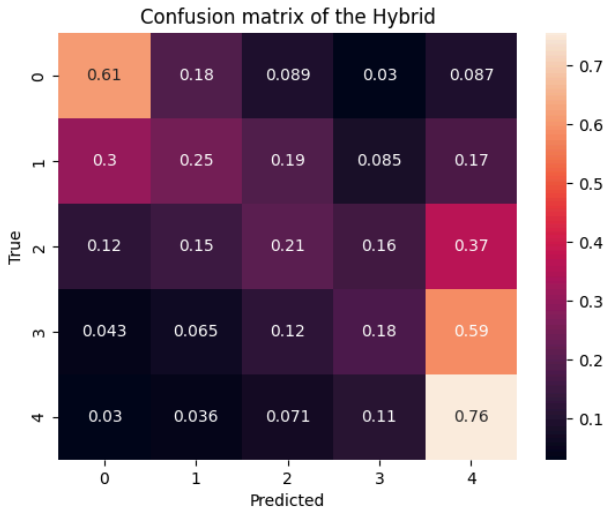
For KNN





For Random Forests

Something I noticed from the KNN matrix was that most of the reviews get misclassified as 5 stars at a pretty high probability. My assumption for why this happens is that there is an imbalance in the dataset since there are alot more 5 star reviews than any of the others so there might have been some sort of bias. So I tried random foresting and logistic regression since they are more tolerant to data imbalance. From the matrices we can see that there was some improvement in some of the classification but was lower value for the correct 5 stars. So I thought that maybe I could do something like a hybrid model where I used KNN to predict 5 stars and then use logistic regression to predict the rest of them since it was better at doing that than the other models. The confusion matrix looked like this and it seems that the 5 stars was not as good as I thought it would be.



In the end the final model with the highest accuracy for me was the KNN one. I am assuming that is the case since such large number of the dataset was 5 stars causing the other reviews to be overshadowed by the sheer amount of 5 star reviews.

4) Takeaways and conclusions

In the end I did find this to be a rather interesting challenge as we had to implement what we used in class as well as outside knowledge and our own research to find a solution for this problem. I think that if I could have better executed the hybrid model it would have outperformed the KNN model but I seemed to have failed the execution. Something I would have liked to try as well was maybe SMOTE to synthetically create data to fix the imbalance and see if that would yield better results. Some other takeaways from this was learning how to optimize code and learn to build upon what I already have to save time computing everything all over again.