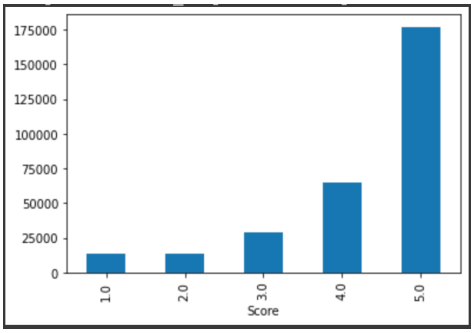
Name: Ruojia Li
BUID: U56240256
Kaggle: coletteli

CS 506 Midterm Report: Movies Ratings

Preliminary analysis

I started by analyzing the distribution of ratings to get a general idea of how the data was set. I noticed that there was an imbalance, and that there were a lot of 5s. I decided to take this into account later on in my other models by selecting a more randomized. I started off by looking at the missing and null values. I realized there were a lot of empty summaries, so I had to remove them before starting off with any analysis. I started to fill or delete them. Then I decided to plot the data to look for any correlations or trends.



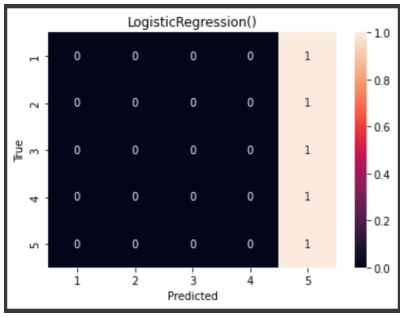
Feature Extraction

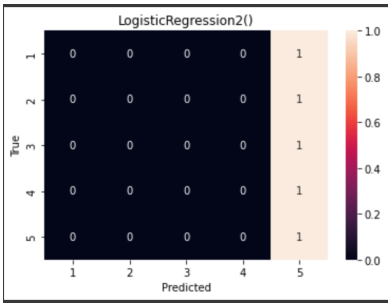
In the Process function, I used **Helpfulness** which was created through dividing HelpfulnessNumerator by HelpfulnessDominator that shows the overall helpfulness of the comment. I also used **UserAverage** and **ProductAverage** to get more of a scope on the overall dataset. I then created process2 function that used **UserAvg**, **MovieAvg**, **UserIDs**, and **MovieP**. The X_Train and X_tests within the train.csv had no overlap, so i calculated the average rates.

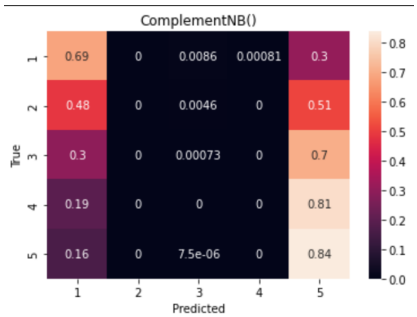
Flow of the progress

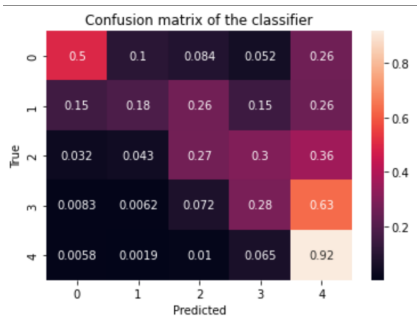
I ran several different models trying to figure out which one was the best. At first, I wanted to group all the reviews together, or get the average rating. Then I thought about running sentiment analysis to see if there were any patterns in the speech. However, i was worried about if I was able to use the various packages or not. So i decided to maybe tokenize the data.

Model Name	Performance	Runtime	Coefficients	Comments
Vectorizer		~1min	N/A	This was mainly to see if there were any trends within the data using TfidfVectorizer to see if there are any trigger words
RandomForestClassifier	The best RMSE was k = 13	~5 min		This was extremely helpful because it tested the k val up to 30. I was able to compare all the various values and figure out what was the best fit.
LinearSVM	RMSE = ?	3+ hours	N/A	SVM ran for an extremely long time and I never fully got the result. However, I don't believe it would have been particularly useful regardless of if it finished running or not.
Logistic Regression	RMSE = 1.7 Accuray = 0.59	~1min	Vectors of features	The logistic model regression took the shortest amount of time. Logistic regression is also a much better model for classification.
Naive Bayes	RMSE = 2.14 Accuracy = 0.56	~1 min	Y train and Test processed	This model was just for fun and because I was curious, the heat map was interesting to see since it was almost symmetric vertically.









Model Validation

I was using the RandomForestClassifier for most of the week since I got the best results with these. I was able to use the starter code to then fill in a model classifier found on scikit, and was able to find k=13, which had an accuracy of 0.68, and RMSE of 0.6. This was the model that gave me the highest accuracy overall, so I used this to submit on Kaggle and it gave me 1.14. However, I ended up using logistic regression on previous text vectorization that I built off on. This ran a lot faster than my previous models, and also gave me a higher accuracy overall. Accuracy level was 0.66 with a RMSE of 0.86. This ended up being my best overall prediction, and is what I used at the end. I then plotted a confusion matrix on the classifier with the scores to help visualize the data I ended up calculating. Visualization with the confusion matrix helped the most in figuring out which models were the best.

Conclusion/Challenges/Improvements

The biggest challenge for me was runtime and my laptop. VSCode would constantly crash, and my laptop just isn't strong enough to run the code for an extended amount of time either. I was able to overcome this through using googlecolab to help run so my laptop wouldn't be affected by this. It went a lot smoother and ran faster at times as well. Overall, I wish I had more servers to run so I could try more models. I feel like I learned a lot this past week since it was very open ended and hands on.