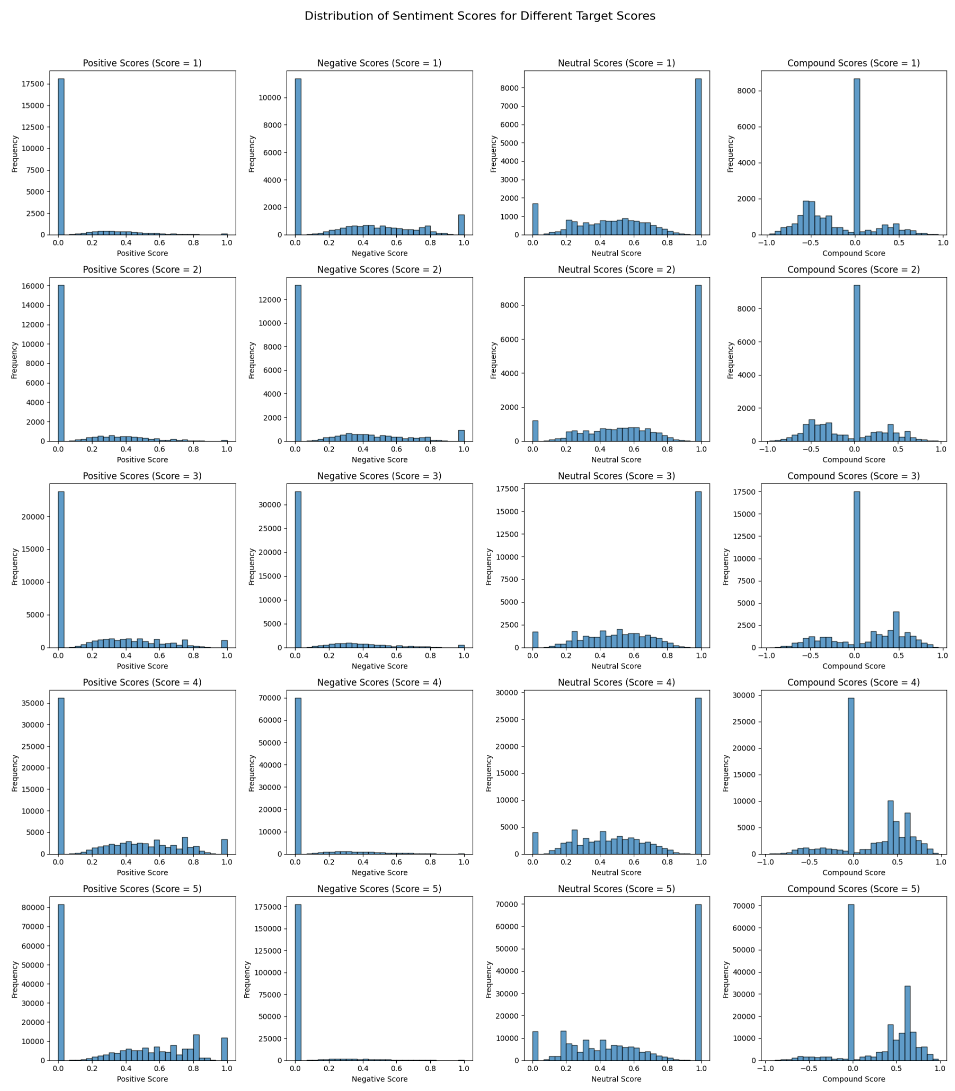
Strategy

For my strategy, I wanted to find features I thought would be useful and then add them to different machine-learning models. To find these features, I started analyzing the problem.

I first looked at the research paper linked in the Kaggle. In their abstract, they talked about how users need to be adapted to more unique movies. As in they recommend products that a user will enjoy now while acknowledging that their tastes may have changed over time, and may change again in the future. Therefore, I wanted to do something similar and model the user's experience, the number of reviews for each movie, and the difference between them. Therefore, if a user is watching a movie that is out of their 'experience level', then that will be noted as a feature and used to predict the score. I also added a time decay, where newer reviews are weighted more.

Another feature I wanted to look at is the sentiment of the review. My thought was that if a review is very bad or very good, it will be reflected in the score. As a result, I ran a sentiment analyzer from nltk, for every review summary. I felt like the text was too long and people might ramble for too long, and the summary is more concise. For the sentiment analysis, I got a positive score, a negative score, a neutral score, and a compound score. I also wanted to get the average sentiment score for each movie and user, in case a user gives a vastly different sentiment, or a movie has a vastly different sentiment. I then wanted to see the distribution of the sentiments for each score. From the graph below, we can see that the frequency of positive scores increases as scores increase, the frequency of negative scores decreases, neutral is ambiguous, and compound scores shift to the right. Since the frequencies were different, I hoped this feature would better my model in identifying scores.

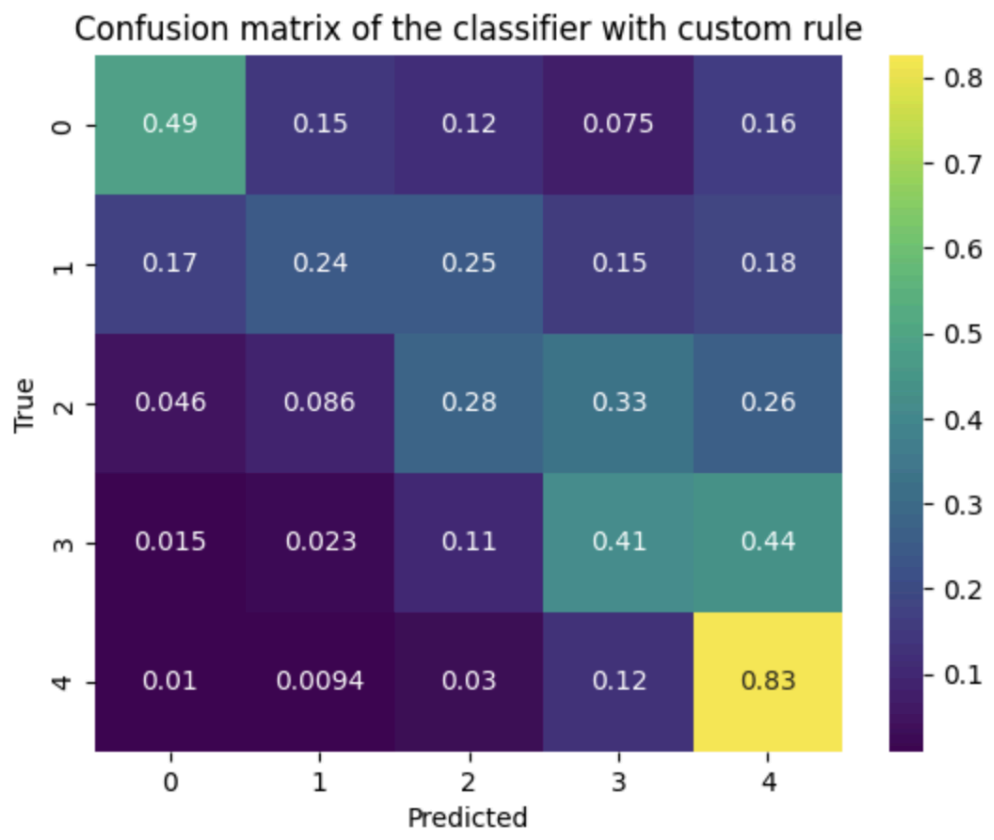


Also, If we wanted to find the score of the review, I thought it would mostly depend on what movie was being reviewed, and who was reviewing it. However, how would I know if those were actually significant or not? To find out, I conducted a Kruskal-Wallis test, which finds out if at least one group differs significantly. After conducting a Kruskal-Wallis test, I got a p-value of about 0, meaning it's extremely likely that at least one group differs significantly. Rather than using the mean though, I thought of getting the lower and upper confidence bounds for the movie and user. I did this because I wanted to create a stat that takes into account the mean, standard deviation, and number of reviews it's based on. Therefore, I took the 95% confidence bound by having the mean added or subtracted from the product of the z-score associated with the 95%, multiplied by the standard deviation over the square root of the number of reviews. This means that we are 95% confident that the true average of the user's score and the true average of the movie's scores are within those intervals respectively.

Methodology

For my methodology, I aggregated the testing data by merging it with the training data with the ID's. I then removed the testing data from the training data. Next, I split the training data into training and testing data that will be used to train the model. For my features, since I applied them to my training data, when I tried applying them to my testing data set, I had some NaN values. To remove them, I didn't want to fill it in with 0 or remove the rows. To handle this, I used the mean and sometimes median values of the column when NaN values existed. I thought this would have a better interpretation than a 0, but in the future, I can explore how the model performs with 0 instead.

For my model, I tried a variety. I first started with the KNN starter code model, however, after playing around with different models, I discovered that XGboost and Random forests performed much better. Both XGboost and random forests gave me about 60% accuracy, however, XGboost training finished much quicker, so I stuck with that model. This was my final confusion matrix. At first, I was concerned that I was predicting a lot of value 5. However, I realized that since our data set was skewed with mostly 5s, then we should be predicting a lot of 5s.



In the future, if I were to continue working on this project, I want to try using different models and play around with new features. In class, the leaderboard panelists spoke about hyperparametric tuning, and so I want to implement that in the future. I'd be interested in implementing a latent factor model to capture deeper patterns in user preferences and movie characteristics, especially as they evolve over time. Additionally, I'd consider logistic regression as a potential approach to classify reviews based on probabilistic thresholds, which could offer a more interpretable perspective on how various features contribute to scoring tendencies. This iterative process of enhancing both the model and feature set could bring the project closer to a highly accurate and adaptive recommendation system.