Project 5 Report - Naive Bayes Classifier

Jonah Harmon & Caitlin Thaxton

Presentation:
https://docs.google.com/presentation/d/1KizsVGl6oImbd2Dbc72X8eWqk6fsN29MC5Mam2gcKWg/edit?usp=sharing

Github: https://github.com/JHamoni676/cs5830_project5

Introduction

Our project focuses on sentiment analysis of large datasets containing user reviews from Goodreads and IMDB. By using a multinomial Naive Bayes Classifier to categorize reviews as having a negative or positive sentiment, we provide insight into how users perceive books and movies. This analysis is important for building recommendation systems for platforms like Goodreads and IMDB. Understanding user sentiment will allow these platforms to deliver personalized recommendations to improve user engagement and satisfaction. Additionally, the sentiment data can be used by publishers, movie studios, and e-commerce platforms for market research. By analyzing reviews, these businesses can better understand the preferences of the consumer.

Dataset

We chose two different datasets to use in our analyses. They are an IMDB movie review dataset and a Goodreads book review dataset. The IMDB dataset contains numerous user reviews for roughly 1000 movies totaling almost 1,400,000 written reviews. Important attributes include the rating given by the reviewer (1-10), the written review itself, and the title of the movie. The Goodreads dataset contains almost 1,000,000 written reviews. The attributes we focused on were the rating given (1-5), and the written review. These are related to our domains because they are pulled from the platforms Goodreads and IMDB and pertain to book publishers and movie studios as well.

Analysis

We used a multinomial Naive Bayes Classifier to analyze our data. This allowed us to train a machine learning model that could take a review from either IMDB or Goodreads and tell us whether the review is positive or negative. Naive Bayes is a great tool for sentiment analysis, which is why we chose it for analyzing these datasets. We specifically used a multinomial Naive Bayes Classifier because our data (the reviews) consists of non-numerical information. This essentially allowed us to analyze the words in each review and match them with being more likely to come from a positive or a negative review. Then when given a review of unknown sentiment, we could use those probabilities to classify the sentiment as either positive or negative.

We also used some data visualization techniques to supplement our analysis. Count plots were used to visualize the distribution of the ratings and sentiments. We also used a confusion matrix to visualize the accuracy of our model.

Results

In order to make classification easier, we decided to sort the reviews into two categories: positive and negative. To provide context for where we split the data for both datasets, we plotted the distributions using count plots.

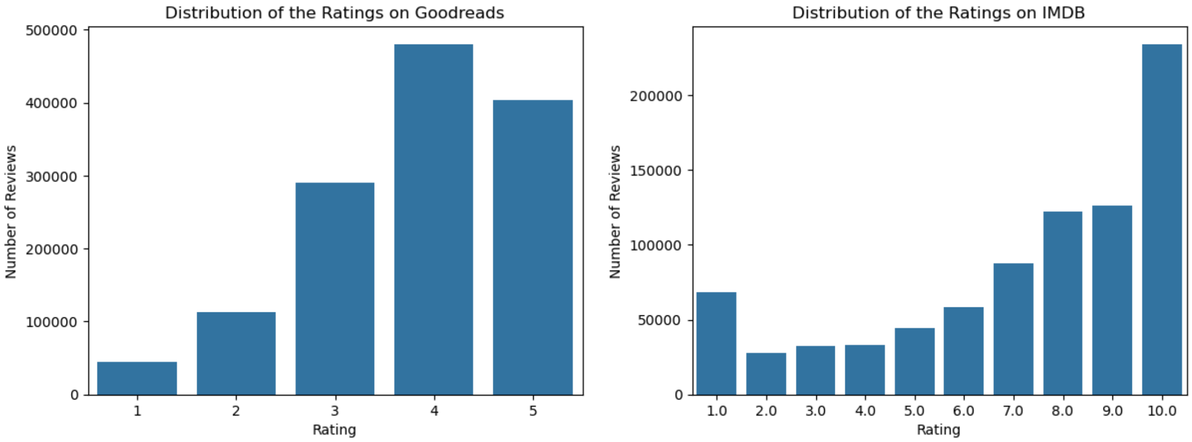
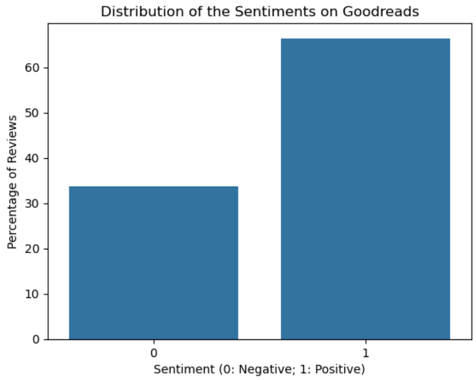


Figure 1 (left): Distribution of Goodreads ratings (1-5 out of 5 stars)
Figure 2 (right): Distribution of IMDB ratings (1-10 out of 10 stars)

Figure 1 shows the distribution of Goodreads ratings. The average review score is 3.7 with the most common rating being a 4 out of 5 stars. The data is left skewed with many more 5 star ratings than 3 stars, and much fewer 1 or 2 star reviews. Based on the suggested review meaning on Goodreads, we expected to split the data to make 1-2 stars negative and 3-5 stars positive. However, this data seems to indicate that if someone likes a book they are much more likely to rate it at 4 or 5 stars, and 3 stars indicates a neutral or conflicted review. As such, we included the 3 star reviews with the negative reviews for sentiment analysis.

Figure 2 shows the distribution of IMDB ratings. The average rating is 7.2, but the data is very left skewed. Surprisingly, 10 star reviews are vastly more popular than any other rating. Seven, 8 and 9 star reviews are also fairly common, with the frequency of rating decreasing as the rating value decreases. Interestingly, 1 star reviews appear to be the most common negative rating, which makes sense because people are more likely to review something if they love it or hate it, and if they hate it they are likely to enter the lowest rating possible. Since the average rating was about 7, we decided to split the reviews for sentiment analysis so that 1-6 were considered negative and 7-10 were considered positive.

With the negative/positive splits set at 1-3 & 4-5 for Goodreads ratings and 1-6 & 7-10 for IMDB ratings, both data sets consisted of approximately ⅓ negative reviews and ⅔ positive reviews, as shown in Figures 3 and 4.



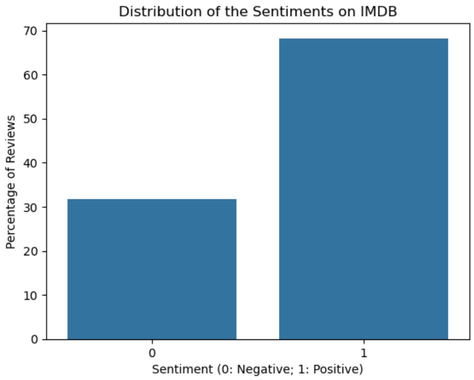


Figure 3 (left): Distribution of Goodreads sentiments (Negative vs. Positive)
Figure 4 (right): Distribution of IMDB sentiments (Negative vs. Positive)

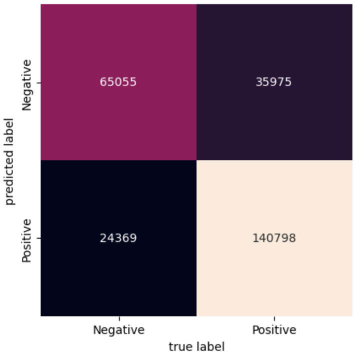
Once we decided how to classify positive vs. negative reviews, we created our Naive Bayes Classifier based on our training data. We then tested the classifier on the test data and got the precision, recall, and F1 scores listed in Tables 1 and 2. We got great F1 scores (greater than 80%) for reviews with positive sentiments in both datasets and okay F1 scores (just less than 70%) for reviews with negative sentiments.

BOOKS	Negative Sentiment	Positive Sentiment
Precision	0.6439	0.8525
Recall	0.7275	0.7965
F1 Score	0.6832	0.8235

MOVIES	Negative Sentiment	Positive Sentiment
Precision	0.6183	0.9075
Recall	0.7957	0.8031
F1 Score	0.6959	0.8521

Table 1 (left): Precision, Recall, and F1 scores for Naive Bayes Classifier on Goodreads dataset
Table 2 (right): Precision, Recall, and F1 scores for Naive Bayes Classifier on IMDB dataset

We also plotted a confusion matrix for each model to visualize our precision and recall, as shown in Figures 5 and 6. The weakest link in both models is the precision for negative sentiment, which indicates that a lot of reviews that are predicted to be negative are actually positive (shown by the top rows in the confusion matrices, i.e. about 35-40% of the top row being listed in the top right box instead of the correct top left box). This shows that our models' predictions for positive reviews are quite good but could be better for negative reviews.



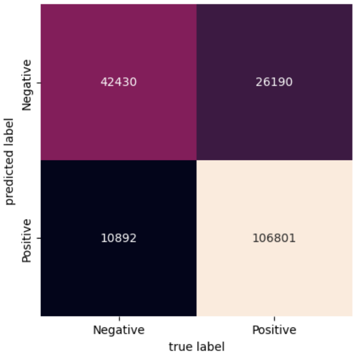


Figure 5 (left): Confusion matrix for Naive Bayes classifier on Goodreads reviews
Figure 6 (right): Confusion matrix for Naive Bayes classifier on IMDB reviews

Technical

The datasets were not incredibly difficult to deal with, but they did take some wrangling. The Goodreads dataset was a JSON file that kept giving us errors until we figured out the JSON was actually one JSON object per line. We also learned that a 0 in the Goodreads dataset meant there was no rating given, so those with a 0 were removed. The IMDB dataset was actually a folder with a .csv file per movie that contained all the reviews for that movie, so we had to figure out how to combine them all into one dataset and create a movie title column. The review ratings were not always integers, so they had to be converted to such. For both datasets, we added a sentiment column. We performed sentiment analysis on both datasets in two ways. For one analysis, we split the ratings into two categories (Negative and Positive), and for the second analysis we split the ratings into 3 categories (Negative, Neutral, and Positive). For the Goodreads dataset we split the ratings for two categories as such: 1-3 into Negative, and 4-5 into Positive, and for three categories as 1-2, 3, and 4-5 respectively. For IMDB: 1-6 into Negative and 7-10 into Positive, and for three categories as 1-3, 4-6, and 7-10 respectively. We only reported on the results with 2 categories because including a Neutral category had lower precision, recall, and F1 scores, probably due to more overlap among the categories. We feel these were good uses for a Naive Bayes Classifier to attempt to indicate sentiment for movies and books. We also used distribution plots of ratings and sentiment to help understand our dataset as well as the confusion matrix demonstrated in class to help understand our precision, recall, and F1 scores.

Our analysis process was simple, we took both datasets, added the sentiment column, used a 80/20 split for our training/testing sample sizes for both book and movie reviews on the multinomial Naive Bayes Classifier. As mentioned above, we tried it using Negative and Positive categories as well as Negative, Neutral, and Positive categories. Implementing the Multinomial classifier wasn’t too difficult. We found that our classifier performed better using two categories vs three. We might have been able to have better numbers with different ranges for Negative, Positive, and Neutral. An additional possibility for analysis would be to manually read reviews from a smaller training dataset and sort them into positive and negative categories ourselves. It is possible that people have different scales for how they like to rate books and movies, differing from person to person, which makes how we calculated positive vs. negative sentiments imperfect.