Sentiment Analysis of Steam Game Reviews Using Naïve-Bayes and Support Vector Machine Classifiers

Carter Squires, Hunter Kavanagh, Shrijeet Mishra, B. Ezra Weiland
Beacom College of Computer and Cyber Sciences
Dakota State University, Madison, South Dakota
carter.squires@trojans.dsu.edu, hunter.kavanagh@trojans.dsu.edu, shrijeet.mishra@trojans.dsu.edu, bergen.weiland@trojans.dsu.edu

Abstract*-- Steam is a video game distribution platform which users can leave reviews for the game they purchase. We performed sentiment analysis on Steam Review data using the well known Natural Language Processing models of Naive Bayes and Support Vector Machines (SVM). We analyzed the results from each of the models and concluded that the BOW model was the most effective at classifying steam reviews followed by the Term Frequency - Inverse Term frequency (TF-IDF) model and lastly the SVM model. The BOW model had an accuracy of 88.8%-89.2% while the TF-IDF model had an accuracy of 87.9%-88.2%. Lastly the SVM had the lowest accuracy of 86.0%-88.8%. Overall this shows that BOW models are the best to use with these types of datasets.

I. Introduction

A. Background

Steam is a video game distribution platform, launched in 2002 by its parent company Valve. Steam (and by extension, Valve) is hugely successful, having accounted for up to 70% of total PC game downloads in 2011 and pulling in $4.3 billion in 2017 [1]. Steam currently offers thousands of titles, ranging from simple arcade-like games (Pac-Man Championship Edition DX+) to simulations (Football Manager) to AAA behemoths (Monster Hunter World) [2]. Since 2019, Steam has had over 90 million active users, with over 14 million of those using the platform daily. [3] Needless to say, Steam is a massively important retailer in the games industry with a well-developed platform comprised of many loyal users.

B. Our Project

In the age of the internet, product reviews have a lot of power. They may be the deciding factor in whether someone chooses to buy a given product. Products like video games are increasingly available online only as opposed to being sold in storefronts, where consumers would not have easy access to reviews. Steam is one such platform, an online games retailer where consumers may review games that they have purchased and played.

Each Steam review is comprised of the text of the review, time posted, time updated (if applicable), number of hours that the reviewer spent playing the game at the time the review was written, the number of Steam users who found the review helpful, the number of Steam users who found the review funny, and a categorization signifying whether the reviewer would recommend the game or not. On the game's page, there is a signifier of whether the reviews are overwhelmingly positive, very positive, positive, mostly positive, mixed, mostly negative, negative, very negative, or overwhelmingly negative [4], in order of how often the reviewers marked that they recommended the game.

Sentiment analysis is an analytical technique that uses machine learning to try and figure out what a person who wrote something was feeling about the subject of the writing [5]. Sentiment analysis falls under the umbrella of Natural Language Processing, or NLP, which is defined as "a field of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages," [6].

In this project, our team used sentiment analysis to analyze the text data in a subset of Steam reviews. We employed three different methods of sentiment analysis to categorize this text data and predict whether the reviewer recommended the game.

II. Data

Our dataset is called Steam Reviews Dataset 2021 [7], and we got the dataset from the Kaggle website. The dataset contains around 21 million reviews spread across about 300 games and many languages. The dataset creators obtained the data using Steam's API. Out of all the games, we narrowed it down to just one game, GTA V (Grand Theft Auto V), which has over three hundred thousand English reviews. The dataset contained many columns, but we only used two columns, review and recommended columns.

We focused on just one game for our classification project for several reasons. First, the dataset we are working with is huge, and using all the data would have a negative impact on performance. Second, much of the dataset's information is specific to the game it was written for, so it makes more sense to focus on one game rather than trying to analyze a few reviews for hundreds of different games.

We chose Grand Theft Auto V (GTA V) as the game to focus on because it is the fourth most reviewed game in our dataset and has a recommended rate of 74%, which is lower than the recommended rates of the first, second, and third most reviewed games. This makes GTA V a good choice because it provides a less biased dataset that will be easier to work with. Fig. 1 shows a word cloud generated with matplotlib comprised of the words most frequently used in GTA V reviews on Steam.

The dataset contains twenty three columns most of them were not useful to us such as the “review_id”, “author.steamid”, or “author.last_played” columns. The two that were are the review column which contains the text we are classifying and the recommended column which contains a true or false value that indicates whether the person writing the review recommends the game or not.

Fig. 1. A word cloud derived from text data from reviews for the game Grand Theft Auto V on the Steam platform.

A. Preprocessing

The first step we took when preprocessing the data was to remove all the non-English reviews, as working with multiple languages is beyond the scope of our project. We chose to lowercase all the reviews because it will lower the number of types and increase performance. Next, we removed all punctuation which will also reduce the number of types and improve performance. Next, we removed all stopwords as they offer little value to the model. We got our list of stopwords from the nltk library. We chose to lemmatize over stemming because lemmatizing can break down words that stemming cannot, and we wanted to lower the number of types as much as possible. We used the WordNetLemmatizer from the nltk library to lemmatize the data.

III. Methods

We used two different models for the purpose of this comparison. The Bag of Words (BOW) model and the Term Frequency - Inverse Term frequency (TF-IDF). We used two different classifiers to further explore the best algorithms.

A. Models

1) Bag of Words Model

In the BOW model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity [8]. For example, we list the frequency of the term in an array like this.

"I","like","the","game"
BoW1 = {"I":1,"like":1,"the":1,"game":1
(1) [1, 1, 1, 1]

2) Term Frequency-Inverse Document Frequency Model

The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general [9].

B. Classifiers

1) Multinomial Naive Bayes

MultinomialNB implements the naive Bayes algorithm for multinomial distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although TF-IDF vectors are also known to work well in practice).

2) Support Vector Machine

The support vector machines support both dense and convertible to that by and sparse sample vectors as input. However, to use an SVM to make predictions for sparse data, it must have been fit on such data. We fit the previously mentioned models into the SVM. The SVM model turned out to be more expensive compared to Naive Bayes. SVM’s split the data by mapping it into different planes. Since our model contained a large number of features, which are each unique word in the document, we could not implement it on our full-scale data. We instead used 10,000 instances which was about 3.1% of the original data.

C. Splitting the Data

We decided to go with a 80-20 split for the training and testing data. Since we had a large number of reviews, 20% of data for testing was deemed to be appropriate. We ran each model multiple times to get the ranges of the results to better represent the performances of the models.

IV. Results

The BOW model performed the best out of all the models we tested. Finishing with an average of 88.8%-89.2% accuracy score for each test. Out of the correct results, the BOW model had 70.7% true positive results and 18.5% true negative results. For the incorrect results, the BOW model had 6.6% false positive results and 4.3% false negative results (as seen in the chart in Fig. 2). Considering the amount of nonsensical data in our data, the BOW model performed very well at predicting the classification of dubious data. Most misclassification by the BOW model tended to be annotation mistakes from the reviewer side such as leaving the review “good game” and marking the review as not recommended.

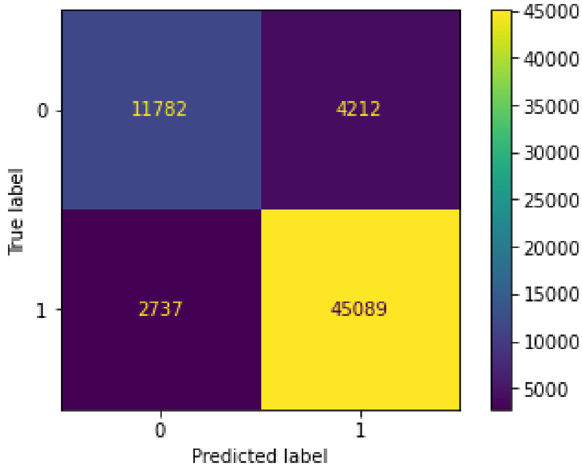

Fig. 2. A chart showing true positives, false positives, false negatives, and true negatives, showing the performance of the Bag of Words model with the Naïve-Bayes classifier.

The TF-IDF model performed slightly worse than the BOW model having an average accuracy of about 87.9%-88.2% with each test. It produced 72.8% true positive results and 15.5% true negative results. Interestingly, it also produced 9.7% false positive results and only 2.1% false negative results (as seen in the chart in Fig. 3). This model tended to assign positive classifications more frequently than the BOW model which lend to more true positive and false positive results. Consequently, this also lend to less true negative and false negative results due to the number of positive classification.

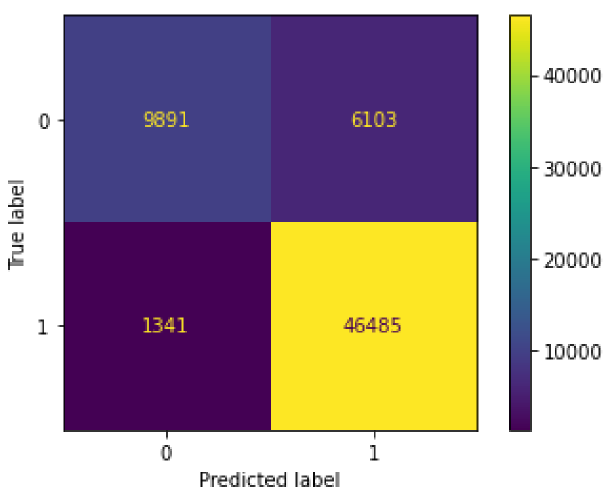

Fig. 3. A chart showing true positives, false positives, false negatives, and true negatives, showing the performance of the Term Frequency-Inverse Document Frequency model with the Naïve-Bayes classifier.

The SVM classifier performed the worst out of the models tested however this due to performance issues and the testing/training data given to this. It scored an 86.0%-88.8% accuracy on average with a large amount of variance due to its smaller sample size. Typically it produced 71% true positive and 16% true negative results while also producing 9% false positive and 4% false negative results (Fig. 4). The main reason behind the SVM classifier's struggles was most likely due to the limitations with the dataset. Otherwise it seemed to struggle more than any other model to classify positive reviews.

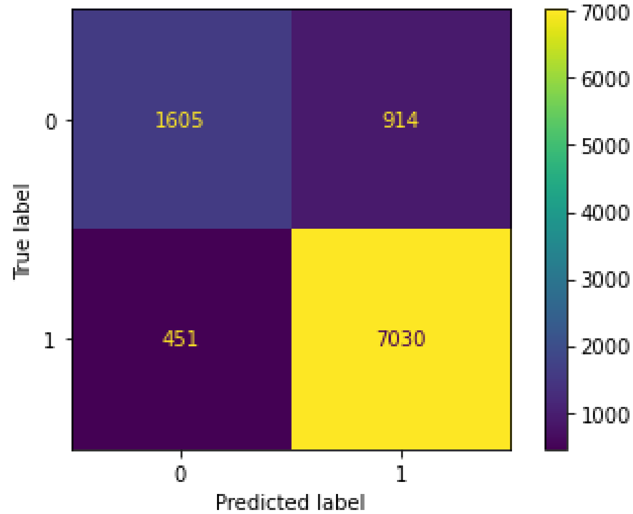

Fig. 4. A chart showing true positives, false positives, false negatives, and true negatives, showing the performance of the Support Vector Machine classifier.

All three models had common issues with certain kinds of reviews. Reviews with jokes in them tended to throw the models off. One example was, "best loading screen simulator 2018", the word best caused all three models to label the review as positive, however, the review was joking about a negative aspect of the game. Another common review type the models struggled with was review with both positive and negative elements. For instance, the review "single player good, multiplayer trash" was hard for the models to determine due to the mixed nature of the review. Mostly the models were able to identify which aspects the user preferred and tended to classify these reviews wrong. While these were some common mistakes, most misclassifications tended to be an annotation error. Many reviews and their recommended status would be misaligned.

V. Conclusions

Sentiment analysis using Natural Language Processing can be applied to game reviews from the platform Steam to predict based on the review text whether the reviewer would recommend the game. The best method that we found for this used the Bag of Words model in conjunction with the Naïve-Bayes classifier, which produced an accuracy score between 88.8%-89.2%. The Naïve-Bayes classifier with the Term Frequency-Inverse Document Frequency model had a similar score between 87.9%-88.2%. Finally, the Support Vector Machine Classifier performed the worst with a score between 86.0%-88.8%. In future research, we would like to look into using the votes_funny and/or votes_helpful features from the original dataset to potentially offset the decrease in accuracy due to sarcastic or joke reviews.

VI. References

[1] J. Selway, “How valve's early success impacts the company today,” Game Rant, 12-Apr-2022. [Online]. Available: https://gamerant.com/valve-video-game-success-half-life-portal-team-fortress-left-4-dead-steam-values/. [Accessed: 09-Dec-2022].

[2] J. L. Wilson and J. Minor, “Steam Review,” PCMAG, 20-Sep-2022. [Online]. Available: https://www.pcmag.com/reviews/steam-for-pc. [Accessed: 09-Dec-2022].

[3] R. Yuen, “The Rise of Steam: A Case Study on the Most Dominant Force in Gaming,” LinkedIn, 12-Nov-2021. [Online]. Available: https://www.linkedin.com/pulse/rise-steam-case-study-most-dominant-force-gaming-ryan-yuen. [Accessed: 09-Dec-2022].

[4] L. Doucet, “Fixing Steam's User Rating Charts,” Game Developer, 06-Oct-2014. [Online]. Available: https://www.gamedeveloper.com/business/fixing-steam-s-user-rating-charts. [Accessed: 09-Dec-2022].

[5] M. Korolov, “What is sentiment analysis? using NLP and ML to extract meaning,” CIO, 09-Sep-2021. [Online]. Available: https://www.cio.com/article/189218/what-is-sentiment-analysis-using-nlp-and-ml-to-extract-meaning.html. [Accessed: 09-Dec-2022].

[6] S. Rabby, “What is NLP & What do NLP Scientists Do?,” Medium, 22-Aug-2019. [Online]. Available: https://towardsdatascience.com/whatnlpscientistsdo-905aa987c5c0. [Accessed: 09-Dec-2022].

[7] Marko M., “Steam Reviews Dataset 2021” Kaggle, 2021. https://www.kaggle.com/datasets/najzeko/steam-reviews-2021?resource=download

[8] Rajaraman, A.; Ullman, J.D. (2011). "Data Mining" (PDF). *Mining of Massive Datasets*. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

[9] McTear, Michael (et al) (2016). *The Conversational Interface*. Springer International Publishing.