Stevens Institute Technology

Web Mining Full Project Report

Date: December 01, 2018

Professor: Rong Liu
Course: BIA-660 Web Mining
Group Member: Yue Yang, ZhuoHui Jiang, Fang Zhang, Shan Gao

Abstract

In the past one year, PUBG became one of most popular game in the word and at the same time it also received thousands of negative comments and thus we planned to start a project to help those developers and operators extract valuable information from those critics. In this project, we firstly used unsupervised learning and found some neutral players have been classified as supporters. However, the purpose of our project is to get suggestions from those critics and then we plan to assign label for players manually to classify those clear-minded supporters and critics and then train some classification models to discover more critics with firm stand. Next, we made a word interpretation base on those comments made by critics and made some suggestions to PUBG. Finally, we proposed some future improvements for the project.

1. Background

PUBG, one of the most popular shooting games on steam, was published by PUBG Corporation, a subsidiary of South Korean video game company Bluehole. the game was released officially in September 2018. The game is one of the best-selling of all time, with over fifty million sold across all platforms by June 2018. In addition, the Windows version holds a peak concurrent player count of over three million on Steam, which is an all-time high on the platform. PUBG received thousands of reviews from players, who found that there are still many problems existed in the game. Good operation is beneficial to the long-term development of a company. However, it's so difficult for operators to extract valuable information from so many players' reviews because of the limited labor cost, and thus we try to do research on discovering useful information from those critics.

2. Purpose

In this project, we are trying to detect some useful information from the reviews and have deep understanding of the game by analysis crawled from Steam reviews community. After this project, we hope our analysis will help game developers identify the problems that most critics concerned.

3.Data Preparation

In this project, the data source comes from steam and the precise website is https://steamcommunity.com/app/578080/reviews. We use the technique named Selenium to scrape this dynamic page. And we put all reviews until bottom of the page by executing Javascript. We use CSS selectors to get all the attributes including user comment date, user reviews, whether user recommend or not, how many hours spend on the game, and how many games user hold in the account.

4. Data Description

The data size is 2070. The data contains three independent variables including spent hours, hold products and user reviews and one dependent variable which is “recommended or not”. Spent hours refers to how many hours that players have spent on the game. Hold products refers to how many games user hold in their accounts. User reviews refers to what players comment on the website. Recommend or not refers to whether the player recommends this game or not. The sample data presented below:

comment_date	user_reviews	Recommended or not	spent_hours	hold_products
November22	"Great game, good to play with friends and.."	Recommended	1,592.0	5,234
November 22	'Product received for free', '', 'Good gamepl...'	Recommended	439.7	42

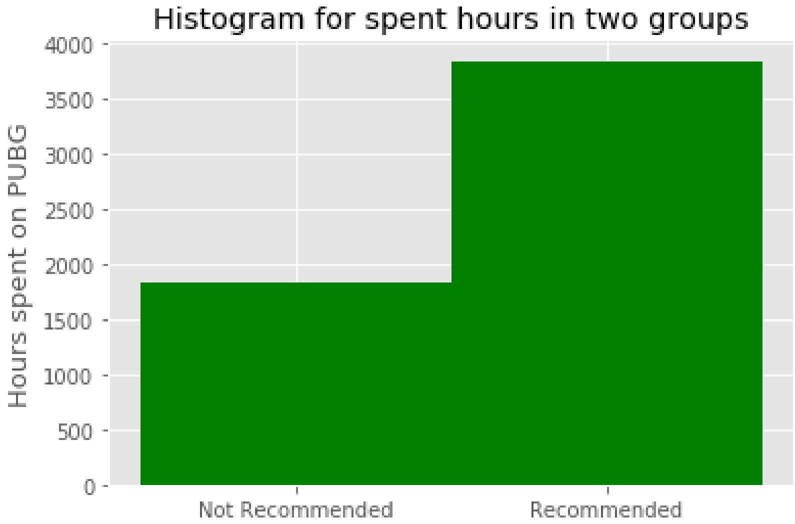
On the platform, each user can easily write down their feedbacks toward the game and they will assign a label of whether they want to recommend or not-recommend this game to other players. Steam will also list information of this users such as hours of playing this game, the Steam product in the user’s account and other players’ attitudes toward this comment. This review system will give game company or developer an overall insight of how this running on this platform by collecting all the data from users’ reviews. The user interface shows below:

5. Data cleaning

The data contains some symbols and stop words which cause problems for us to find meaningful information from the user reviews and thus we remove these noise and transfer all characters into lower case and split the reviews into words.

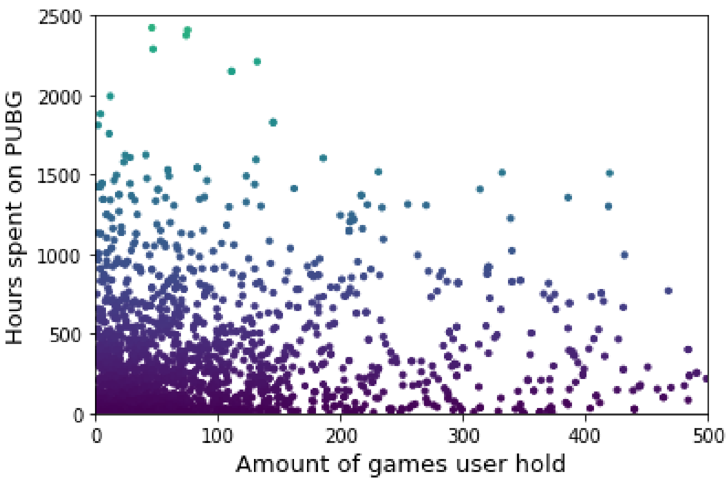
6. Exploratory Data Analysis

Next, we did some simple exploratory data analysis to find the correlation among these variables. Firstly, we made a bar chart to present the relation between spent hours spent on PUBG and recommend or not.



As the bar chart presents above, we find that the people who have spent more than 1900 hours on PUBG tend to recommend this game. In other word, this game is very popular with old players, and thus the reviews made by this part of players are more meaningful for operator.

And then, we made a scatter plot to explore the relation between spent hours and hold products.



As the plot show above, the more game products user hold, the less hours the user spent on PUBG game.

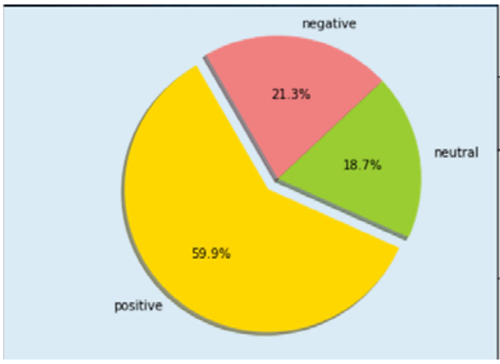
7. Modeling

In this part, we use both of supervised learning algorithms and unsupervised learning algorithms to make models.

7.1 Unsupervised learning

7.11 VADER

VADER analyzes a piece of text to see if any of the words in the text is present in the lexicon. Sentiment metrics are derived from the ratings of such words positive, neutral and negative, represent the proportion of the text that falls into those categories.
The final metric, the compound score, is the sum of all the lexicon ratings which have been standardized to range between -1 and 1 based on some heuristics.
In our project, we want to see user sentiment for each of the reviews by applying VADER analysis, and thus group those negative sentiment for company inspect.



Sentiment Altitude	Accuracy	Description
Positive Sentiment	77.45%	% contains <Recommended>
Neutral Sentiment	% of <Recommended>	75.33%
	% of <Not Recommended>	24.66%
Negative Sentiment	26.38%	% contains <Not Recommended>

From the result shown above, reviews have been classified into three categories: positive sentiment, neutral sentiment and negative sentiment. In positive sentiment, 77.45% of reviews have been correctly classified; In neutral sentiment, 75.33 are recommended reviews and 24.66% are not recommended reviews; In negative sentiment, 26.38% negative reviews have been correctly classified.
We visualize the model result. The model has a bad performance on negative sentiment. And then we try to find the reasons. After we dig into the reviews, we find out the reviews sometimes full of sarcasms, the computer is not capable of recolonizing the emotion behind the sentences. For example: “This game used to be good, but now it is just game for cheaters.” Then we got a conclusion that the labels in the review system didn’t truly reflect the attitudes of players and thus the game developer can also possibly mislead by the data of labels.

7.12 LDA

Due to the massive information of reviews, it’s difficult for operators to analyze the reviews one by one. So, we use LDA (Latent Dirichlet allocation) to generate topics for the document. This model reduces the dimensionalities of word and thus it worked more efficiently compared with bag of words model and got rid of overfitting. We visualized three top topics as below:

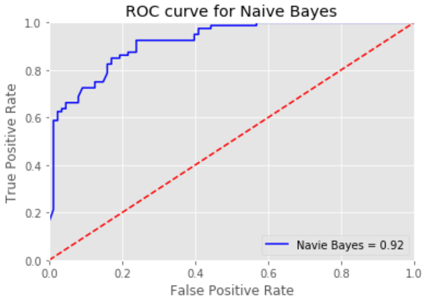
Through these pictures, we found more negative words than positive words. These words state that the server of the game is bad and mountains of cheaters in this game and this game didn’t any improvement overt time. This result may confuse us why there are more negative words and in the EDA stage we found that more people recommended this game, but actually, in the VADER model, we have found that 75.33% people who have neutral attitude have been classified as supporters. And by reading some comments, we found that these people’s comments contain a lot of negative words and that’s the reason why we found so many negative words here. Through the LDA and VADER model, we got a conclusion that we cannot simply classified people by their comments’ labels, and thus, we planned to select those clear-minded comments to train classification model to find more players with a firm stand.

7.2 Supervised learning

The classification models are applied to detect positive and negative users. Since there are a great many reviews are not labeled in other sources. It is necessary to classify a large amount of reviews and get directions for later improvement, especially from the negative sentiments.

7.21 Multinomial Naïve Bayes

Multinomial Naïve Bayes is suitable for classification with word counts for text classification, such as TF-IDF weight.

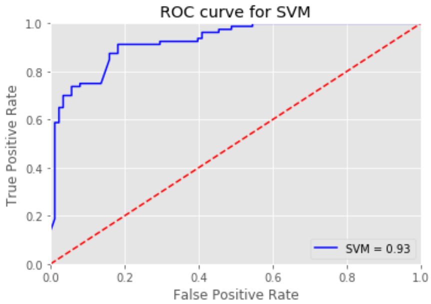


Outcome	Precision	Recall	F1-Score	Support
Not Recommended	0.76	0.95	0.84	88
Recommended	0.93	0.66	0.77	80
avg / total	0.84	0.82	0.81	168

As the result shows above, this model has a good performance. The precision rate is 0.84 and the recall rate is 0.82.

7.22 Support Vector Classification

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification.

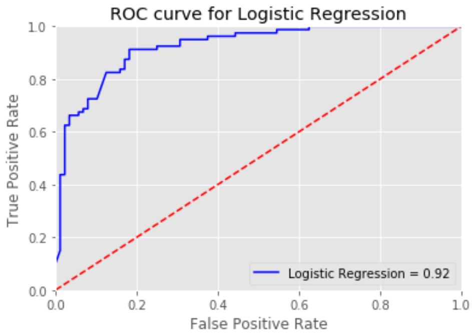


Outcome	Precision	Recall	F1-Score	Support
Not Recommended	0.77	0.97	0.85	88
Recommended	0.95	0.68	0.79	80
avg / total	0.85	0.83	0.82	168

As the picture show above, good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class. It has 0.85 of precision rate and 0.83 of recall rate.

7.23 Logistic Regression

Logistic Regression model the probabilities of a recommendation as a linear function of documents term matrix and classify reviews into two categories based on TF-IDF weights of bag of words.



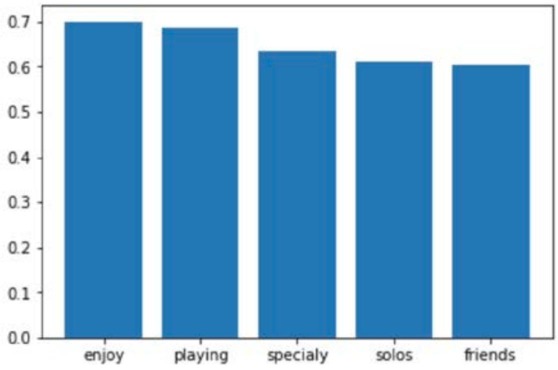
Outcome	Precision	Recall	F1-Score	Support
Not Recommended	0.73	0.98	0.83	88
Recommended	0.96	0.60	0.74	80
avg / total	0.84	0.80	0.79	168

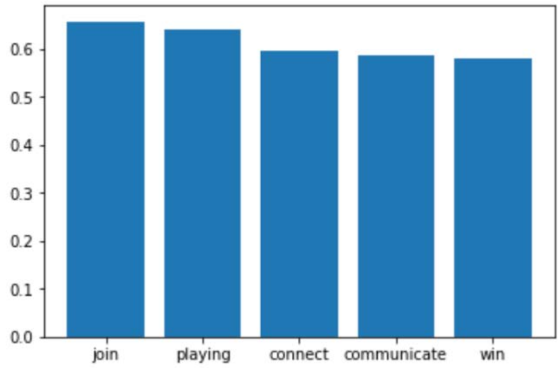
As the picture show above, it has 84% precision rate and 809% recall rate. In summary, MNB, SVM and Logistic Regression have good performance.

7.24 CNN

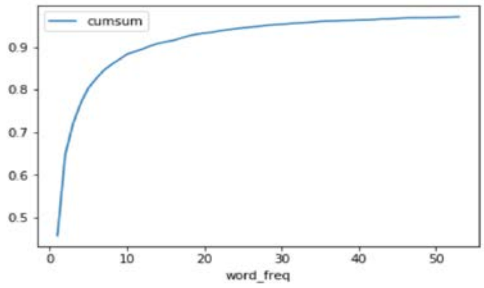
The CNN model is introduced since it has advantages in imbalanced data classification. The doc2vec and word2vec is used to train a large amount of unlabeled reviews to generate a fixed weights word matrix to map the word in the embedding phrase. The doc2vec is difference in predict the word by concatenating the paragraph vector D (shared within the paragraph). The words vector trained by Doc2Vec has a better performance when check the words similarities.

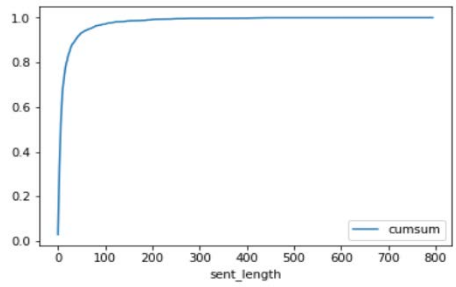
The figure describes the top 5 most similar word with 'play':





The CNN model is using three sizes of filters (bigram, trigram and quadrigram), each size has 64 filters in convolutional layer.





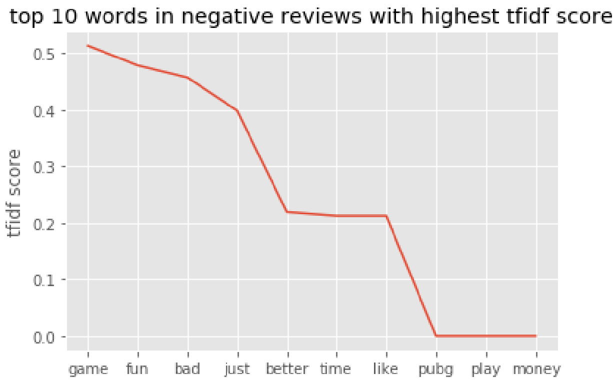
	precision	recall	f1-score	support
0	0.76	0.62	0.68	138
1	0.89	0.94	0.92	489
avg / total	0.87	0.87	0.87	627

	precision	recall	f1-score	support
0	0.78	0.71	0.75	138
1	0.92	0.95	0.93	489
avg / total	0.89	0.90	0.89	627

In summary, The CNN using pretrained matrix has a better performance as the picture show above.

8. Word Interpretation

To get some insights from negative sentiment reviews, we use TFIDF weights and Word2Vec to dig out contents from reviews.



As the picture shows, we picked top ten words which have highest frequency. Some words such as bad, time, money are key words might point to potential problem.

Next step, we use Word2Vec to find most similar words with these top 10 words, trying to find correlation between targeted words.

As a result, the problems found are as follows:

1. A lot of game issues
2. Serious time delay for server lags
3. Devs(Developer) fix the bug
4. Vehicle issue in game
5. Waste money
6. A lot of cheaters

9. Conclusion

In unsupervised learning we found that the label in dataset cannot accurately describe the attitude of players, but the purpose of our project is to excavate valuable information from critics and thus it’s very important for us to train a model to find those critics with frim stand. And then we trained several classification models including Multinomial Naïve Bayes, SVM, Logistic Regression and CNN, and found that CNN has the best performance among these models. Finally, we interpret the words meaning after classification, and then give the suggestion based on what we got from the result and raise the suggestion to company. Wipe out cheaters from game, it hugely impacts users game experience. Give more support to users when they meet with game issues. Developers need to put more efforts to optimize game and fix those existed bugs. Cautiously deal with the micro-transaction, make it reasonably to users.

10. Future Improvement

There are still some improvements that we can make in the future. For example, we can try Steam Game Platform API, and thus we can get more access to the text data we needed, and we can learn huge amount of knowledge from interacting with API. And we can improve VADER by updating word list. Since VADER is defined as Sentiment metrics techniques, we expect to improve model performance by updating the word list and re-training the model.

Reference:

https://www.sciencedirect.com/science/article/pii/S0893608014002135

https://en.wikipedia.org/wiki/PlayerUnknown%27s_Battlegrounds