Final Project: Twitter Sentiment Analysis

Video Demo: <https://youtu.be/akyg4JRMh7M>

**Problem**

I did my final project on the sentiment analysis of twitter data. Before getting into my topic around sentiment analysis. Sentiment analysis is the processing of text to determine its emotional tone: positive, negative, or neutral. It is often used in businesses to determine how well a product is being received. But what was the problem that I was trying to solve through doing this sentiment analysis? A new game had just come out called Star Wars Jedi Survivor and I wanted to see how the game was being perceived around launch day. What were some of the good things about the game and what were some things that needed to be improved on.

**Experiments**

There were two things I needed to do to determine this. First, I had to train a machine learning model on an existing dataset that already was labeled based on its sentiment and then I had to use that model to predict on the sentiment of my new data or in this case the twitter data. So, I used two different approaches to do this. First, I used a pre-trained machine learning model called Text Blob and then I made my own model/trained my own model using scit-learn. The benefit of the pretrained model Textblob was that I did not need to train on the model beforehand which saved me some effort and time. But I wanted to create my own model to compare the two accuracies of the model and see how they both compared. For my own model I trained it on the Steam Reviews Dataset from Kaggle. It was a large data set of 6,000,000 data points that had steam reviews from the user and had the labeled sentiment. Because my computer could not handle reading that many data points I subset the data to look at only 50,000 which would be more than enough to train my model. But before training my model for sentiment analysis there is a lot of pre-processing that must be done to get the best accuracy. You want to only look at a small unique subset of words to get the best results. To do this I removed any words called stop words which are often redundant and unnecessary words that cannot be perceived as a positive or negative sentiment. Words like (the, of, a) are some examples of those words. To do this I use NLTK’s library to download all these stop words and delete them from the tweet. Then I used tokenization to break down the tweet into a bag of words to analyze each word. When looking at each word in the tweets I then used lemmatization which reduced the word to its root form. This is done in sentiment analysis to reduce the number of unique words and focus analysis on a small subset of words. First to train my model I used a TfidfVectorizer on my which converts a collection of raw documents into a matrix of TF-DF features. These features are the most frequently used words which are converted into feature indices that are computed into a word occurrence frequency matrix. It converts each word into a document with a number that the computer can understand that shows how relevant that word is in the document, in our case tweet. I then split my data into testing and training data. Then I used different classifiers (Linear SVC and Logistic Regression) to fit my data, which essentially returns the best fit hyper that categorizes your data, which in our case is predicting the sentiments. To figure out the hyperparameters of the data I used what is known as Grid Search CV which based on the provided features gives the combination of different hyperparameters that result in the highest accuracy. After training my model I predicted the sentiments to identify the accuracy of the models. Using the Linear SVC classifier, the accuracy score was 87.1% and the Logistic Regression was 86.9%. Both give similar but high accuracies. I also calculated the accuracy for the pre-trained Text blob model to see how well that model did and that gave me an accuracy of 76.1%. So fairly good but not as good as the model we trained. So now for my experiments and results. To analyze how well this product was being received I collected Twitter data. I did this by using the Twitter API to collect the tweets that were written around the release of the game. After pre-processing and cleaning the data like I did with the Steam Review data set I applied both my custom machine learning model that used Linear SVC(didn’t use the logistic regression one because it gave worse results) and the Textblob to predict the overall A blue circle with a triangle in the middle

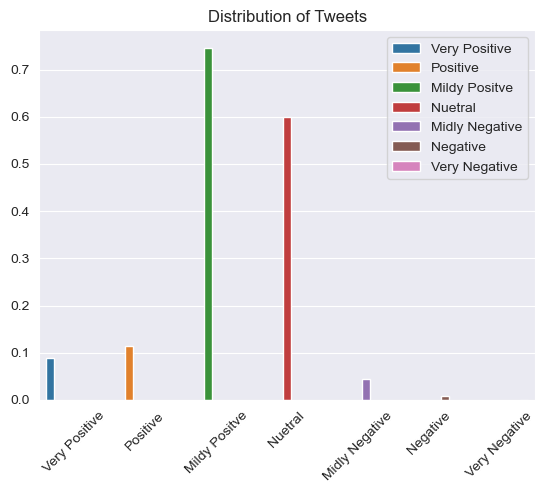
Description automatically generated with low confidenceA blue green and orange pie chart

Description automatically generated with low confidencesentiment of the game around release day.

(Left TextBlob and Right Custom Model)

**Results**

As you can see from the graphs above the overall sentiment was pretty positive. While the custom model was only trained on predicting positive or negative the pretrained model Textblob gave positive, neutral, and negative responses. But both had the majority positive sentiments around the game with 47% positive for Text blob and 97% positive for the custom machine learning model.

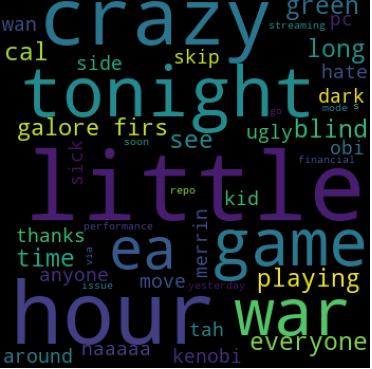
A picture containing text, screenshot, line, plot

Description automatically generated

A picture containing text, font, screenshot, design

Description automatically generatedA picture containing text, font, screenshot, typography

Description automatically generatedThe reason I included the Textblob model despite giving worse accuracies because it provided a level of sentiment 0.0 – 1.0 which gave a lot more information. Thus, I was able to create the graphs above which showed the varying category of sentiment of the tweets. As you can see the overall sentiment is mildly positive and neutral.

A close-up of words

Description automatically generated with low confidenceI also created these word clouds for both the Textblob and the custom model to see which words were most commonly used in good reviews, to determine what were some of the good things about the game. Some words I noticed were the exploration, how the game looked, and the characters in the game.

A picture containing text, screenshot, diagram, square

Description automatically generatedI also created a word cloud for the negative reviews and the most commonly used words in the negative reviews were about the performance, the bugs/crashes, and the FPS.

I also analyzed key words that are normally used to review games i.e., the performance of the game, quality, gameplay, and journey/story. I did this by looking at tweets that contained key words like performance, quality, etc. and looked at the sentiment towards those tweets. Overall, you can see the metric that most stands out is performance, which has the highest negative sentiment out of the other categories. While the quality of the game has the most positive sentiment.

**Reflection**

Overall, reflecting on this assignment I was successful in determining the overall reaction to the game. I was able to determine aspects of the game that were good: characters, exploration, quality, and parts of the game that were not: performance, bugs, etc. Some improvements I can make is to explore more complex and complicated machine learning models. Another improvement is to figure out a way to detect sarcasm in the tweets to better calculate the sentiment. I do plan to continue to work on a similar project to this over the summer, I want to do more sentiment analysis projects and see if I can do more in-depth analysis into it.

**Justification**

This project is related to our class because it looks at building/loading machine learning models based on a feature, the reviews from the Steam Review dataset, and predicts on a response variable i.e., the sentiment of the review/tweet. My project explores analyzing twitter data, using machine learning to calculate sentiment, and then performing data visualization to look at the results. The level of effort spent for this project can be divided into two parts: learning about sentiment analysis and then applying it. I know I wanted to do a project looking at a game that I was looking forward to how Star Wars Jedi Survivor was being perceived. I knew I could do this by exploring sentiment analysis. I spent 3 hours looking at what sentiment analysis was and how people were doing it on different topics. For example, I looked at a video that performed sentiment analysis of Presidential candidates and how each candidate was being perceived. After getting a basic understanding of it I did my own sentiment analysis of the games in the steam reviews dataset, which took around 1-2 hours. Then, I wanted to get twitter data, so I used the twitter api, which was difficult to get at first because I kept a forbidden access error, though eventually with a video and some documentation I was able to get the data (30 mins). Performing sentiment analysis on the twitter data, analyzing the results, and visualizing the data took around 2 hours. And then finally applying those results and plots into a presentation took another 1-2 hours of work. Overall, as compared to homework 4 where we were only modifying code and generating and analyzing plots this project consisted of learning a new concept, applying that new concept, creating a presentation around the project, and finally creating this report it took 1.5 times the effort of homework 4.

**Word Count**

1518