



Twitter Sentiment Analysis:  
Nintendo E3 2018 Final Report


Jason Zhou


## Introduction


Every year at the E3 (Electronic Entertainment Expo), major developers and publishers in the gaming industry come together to present and preview new and upcoming games. The big three players at this annual event are Sony, Microsoft, and Nintendo. Excited gamers look forward to seeing new titles exclusive to each of the three companies' flagship consoles being announced. Nintendo has announced the newest Super Smash Bros game, a classic fan favorite, among other awaited titles for release in late 2018 and 2019.

 | E3 2018 | Livestreaming begins Monday, June 11, 3:30 p.m. PT | 6:30 p.m. ET

**Video presentation**  
June 12, 9 a.m. PT | 12 p.m. ET

  
June 11 – 12

  
June 12

  
June 12 – 14

During Nintendo's presentation and conference, fans and viewers made sure to post their reactions to Twitter as Nintendo revealed their upcoming games. Analyzing tweets is an excellent way to gauge customer reaction to newly announced products, and we'll be doing just that.

### **Business Problem**

As the data science team at Nintendo, we're in charge of aiding the marketing department in determining demand for its up-and-coming major titles. Nintendo wants to have an idea of how many physical copies of the new games it will need to distribute throughout stores in order to meet demand. One way to do this is by analyzing the tweets that were posted during Nintendo's conference at E3, where they announced the release of upcoming major games. While the data science team won't be determining the demand itself, we'll be building models that will be trained to label the tweets as positive or negative. If our models are able to perform acceptably well, we will be able to accurately determine overall sentiment of the announcement of these games.

In addition to the highly anticipated Super Smash Bros. Ultimate, other major titles are also being scheduled to release in late 2018 or 2019. The other titles we will be looking into are Fire Emblem: Three Houses and Super Mario Party.

## **Working Data Set**

Our data set was found on Kaggle, and in total there were more than 100,000 tweets that were related to Nintendo's E3 2018 presentation. The tweets were compiled together using the `filterStream()` function, using keywords `#NintendoE3` and `#NintendoDirect`. Each tweet is represented using Twitter's own Tweet JSON formatting. More specifics can be found on Twitter's official Tweet documentation:

<https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/intro-to-tweet-json>

A tweet object has a variety of attributes, but we're only really interested in exactly two of them, the 'text' and 'entities' attribute. The 'text' attribute contains the text body of the tweet, and the 'entities' attribute contains information such as hashtags used in the tweet. The text bodies of these tweets will be the main feature of interest, while the hashtags will be used to categorize the tweets into separate data sets.

## **Data Science Approach Summary**

The data science approach I will be using will be a binary classification one. The goal is to be able to train models to accurately label tweets as "positive" or "negative", which in this problem are our two classes. While the tweets are not prelabeled, the labeling will be done by using a Python package known as TextBlob. TextBlob is able to assign a document a polarity score between -1.0 and 1.0. A positive value indicates positive sentiment, and a negative value indicates negative sentiment. Using the bag-of-

words matrix representation as our X, and the positive/negative labels as our y, I will then build, test, and refine a few different classification models to determine the best one. I will start with Logistic Regression and Random Forest algorithms, and then potentially try one or two more classification algorithms that might be able to produce better models. Performance will be defined by accuracy, precision, f1, and recall.

After these models are built, I will then need to test them against my own judgment. I will take a sample of the tweets and label them all myself as positive or negative. After doing so, I can then have each of the best performing models label these tweets and see how accurately the labeling was done compared to my own judgment. Their performance will be scored based on classification report metrics.

If it turns out that the models are able to label tweets accurately to an acceptable degree of performance, then they can be used to evaluate overall sentiment regarding each of the three main games that Nintendo is about to release.

## **Data Wrangling**

The text bodies of a Tweet often contain strings that aren't helpful and irrelevant to natural language processing. Here is an example printed below:

Time for Nintendo to beam excitement directly to your eyes! #NintendoE3 #E3

<https://t.co/OkVfIAIHbV>

What needs to be done to this tweet specifically would be the following: lowercase everything, remove the URL, remove the punctuation, and ignore stopwords and words

that aren't present in the dictionary such as "Nintendo". After this tweet's text specifically would be cleaned, it would look like this:

time beam excitement directly eyes

Some other methods of text processing and cleaning include filtering out numbers and emojis, and lemmatizing words. Foreign tweets will also be filtered by checking to see if they're present in the English dictionary. Some tweets will dodge this check, so further filtering was done by checking for foreign articles such as "de, el, un, au".

After this is done, the tweets are to be categorized by topics, which in our case is the games that they are referencing. Hashtags will be used for this classification but there are a few complications with doing this. Misspelled or differently worded hashtags will need to be accounted for. For example, if we want to look at every tweet related to Fire Emblem, the most common hashtag present in the data that we would use for this would be #fireemblem. However, there are also other tweets that are tagged as #fireemblemthreehouses, #fireeemblem, #fe, #fireemblemswitch, and so forth. I want to include every tweet that's tagged as a variation of #fireemblem, and the challenge is making sure I can identify as many of these alternate hashtags as possible without accidentally picking up unrelated ones. The end result of the data cleaning will be three collections of cleaned tweets, one for each of the games that are to be analyzed.

### **Exploratory Data Analysis**

Since our data is all textual, an effective visual representation of the data would be to create a word cloud of it. A word cloud presents the most frequently occurring

words in the corpus, and highlights their relative frequencies by displaying the word with larger or smaller size. An example, using our Smash Bros word cloud is shown below:

## Super Smash Bros. Ultimate Tweets word cloud

In a very general sense, a word cloud like this can somewhat summarize the corpus of documents. While not every word on its own necessarily conveys positive or negative sentiment, a bit of searching reveals some words that do. In this word cloud for example, we can see positive words such as excited, good, happy, wanted, amazing. Cool, this tells us that we can perhaps expect the majority sentiment of this collection of tweets to be more positive.

After looking at word clouds for each of the other two games, feature extraction was performed on each of the data sets by using vectorizers. Using both count vectorizers and TF-IDF vectorizers, we were able to gain some insight into which token frequencies were more correlated with each other.

## Preprocessing

As mentioned earlier, a focus of this project is to determine the accuracy of TextBlob's sentiment labeling. To do this, some data will need to be sampled and manually evaluated by myself using my own judgment. Then, I can compare the results of the labeling done by models built based on TextBlob's labeling, to my own labeling. This will ultimately determine if the models are practical enough to be used.

For starters, this is where every tweet in each data set will be labeled by TextBlob. TextBlob assigns each string a polarity score between -1.0 and 1.0. I decided that a polarity score greater than or equal to 0 would be labeled 'positive', and a score less than 0 would be labeled 'negative'. Positive would be labeled as 0, and negative would be labeled as 1. After this was done, I wanted to see the distribution of positive to negative labels for each game. A table summarizing the labels depicted here:

Labels by TextBlob	Smash Bros	Fire Emblem	Mario Party
Positive	11861	1534	829
Negative	674	29	53

As we can see, the data is quite imbalanced. This will need to be accounted for during the modeling. We also know that samples will have to be taken using stratification, in

order to preserve a minimum number of negative tweets in the sample. So here, a stratified sample was taken for each game, to be labeled by myself later on.

## **Modeling**

For our binary classification problem, I decided to move forward with four different classification algorithms: logistic regression, random forest, gradient boost, and SVM.

We're starting with our data sets being a column representing the processed text of the tweet, and the column representing the sentiment label they were assigned using TextBlob. Because the text strings themselves cannot be used to train models, we instead used count vectorizers to transform and represent them. Now, each tweet or row is represented as a bag-of-words. The data sets were split 70/30 train/test, and from there it was time to see how baseline models performed. Because we're working with 3 data sets and we're experimenting with 4 different types of models, there are in total 12 sets of results.

Classification reports were generated for each baseline model for each data set. While the scoring on the positive tweets were generally quite high, they were very low when it came to the negative tweets. Example shown:

Performance on Testing Data				
	precision	recall	f1-score	support
positive tweets	0.95	1.00	0.97	211
negativetweets	1.00	0.21	0.35	14
accuracy			0.95	225
macro avg	0.98	0.61	0.66	225
weighted avg	0.95	0.95	0.94	225



This is most certainly caused by our imbalanced data, with negative tweets being the minority class.