Bungie Twitter Analysis

Capstone Project 2 Milestone Report
Sebastian Alvis

Problem Statement

For a company, public opinion is very important. The reach of their social media posts, or how many people see the posts, can often indicate how well a product or idea is being received. The changes in reach can often be correlated to announcements on social media, and can help drive business decisions related to advertising / marketing.

In this project, I am going to analyze the social media presence of Bungie and it's main game, Destiny 2. Part of the analysis will be using natural language processing to inspect the sentiment of each tweet: whether it is positive or negative, and whether that ties into other trends.

Dataset Description

The data was acquired from the Twitter API: https://developer.twitter.com/en/docs. I am using the standard, free version, which allows data acquisition from present day to 7 days prior. I acquired a weeks' worth of tweets on the topics "Bungie" and "Destiny 2" (topic data), and acquired the tweets made by the official Bungie (@Bungie) and Destiny 2 (@DestinyTheGame) Twitter accounts in the same time frame (timeline data).

This process required patience, as there was a significant problem with finding duplicate tweets. I could acquire 20k tweets, but only a few thousand of them would be unique. There is a request parameter that enforces a time constraint on the tweets returned in that request, but I was not using it correctly for some time. In the end, I managed to pass it the integer representing the earliest tweet in my data, and the next request would only find tweets as or less recent than that. This was a problem for the topic data, but not the timeline data.

The loop would run, gathering tweets until nothing was returned and I had exhausted the use of the standard search API. I dropped duplicate tweets and found I acquired 21.4k tweets about Destiny 2 and 24.7k about Bungie. The timelines for the official accounts in that week came back with 46 and 107 tweets.

Data cleaning had two main components: column reduction and missing data.

The dataframe reduction was about reducing the size of my data by removing columns. The topic data was originally a 320-column dataframe. The reason there were that many columns is because a lot of parameters about the user and the location from which they tweeted are

repeated. These columns for the tweet / user / location were repeated in the case of quoted tweets, retweets, and retweets of quoted tweets.

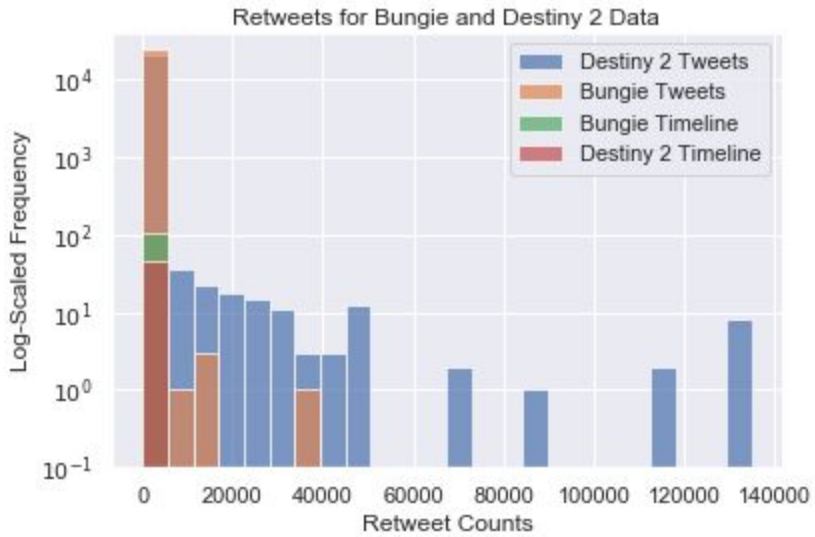
I went through all the column names, found about 50 columns that were the most relevant, and kept those. These 50 columns were present in the topic and timeline data both, so now my four dataframes all have the same columns (the timeline data originally had about 215 columns). I kept columns for some parameters of base tweets, quoted tweets, and retweets. There were less than 10 retweets of quoted tweets out of 40k+ tweets, so I deemed it irrelevant.

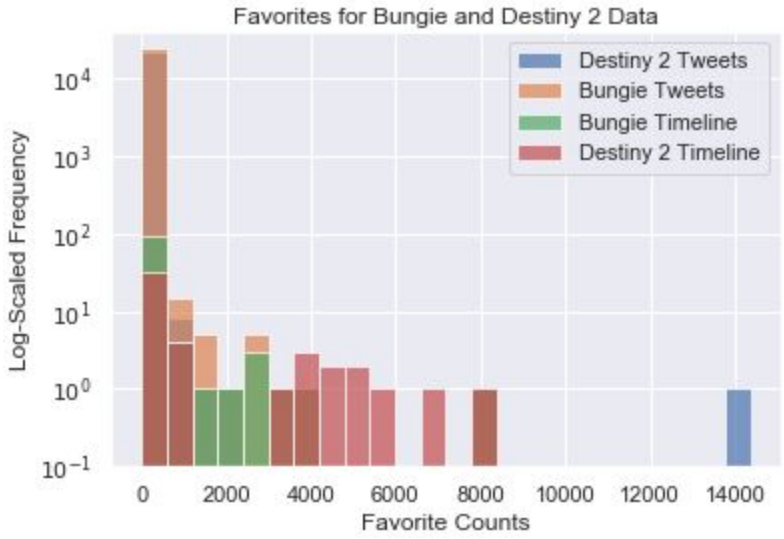
The missing data came in two forms. First was normal missing data: the user and text of the tweet were missing, so I could drop those incomplete rows. The second was misaligned data. I recognized the problem because I had some indices that were clearly text instead of an integer, and when digging further, the rest of the columns had improper values or datatypes. I'm unsure if this was a problem in the API or in my saving of the data, but the indices make me think the latter.

After reducing the column number and removing missing data, I saved the dataframes as new csv files for reproducibility, and the data wrangling / cleaning was complete.

Exploratory Data Analysis

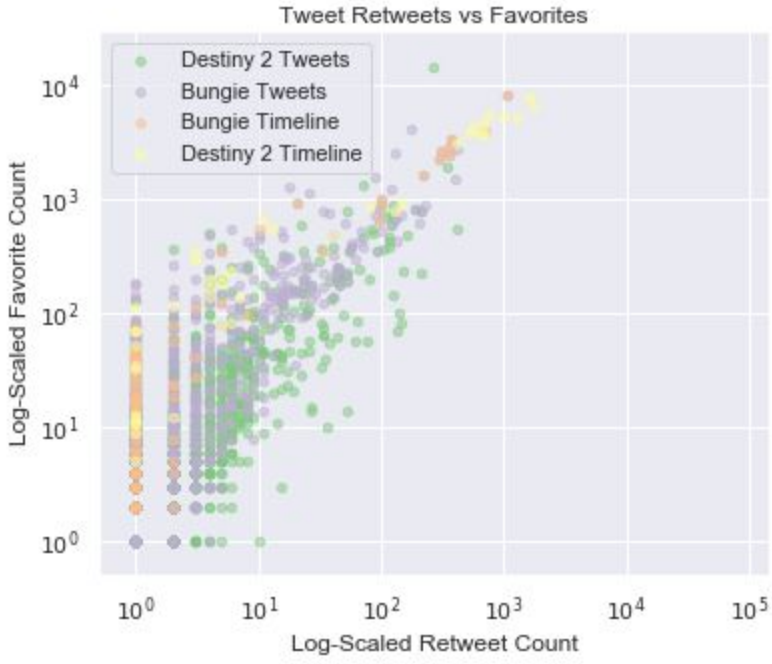
The first metrics that I can measure tweet engagement with are retweet and favorite counts. I have four dataframes, so I can plot these distributions for each one and overlay them all to get a sense of what the data looks like.





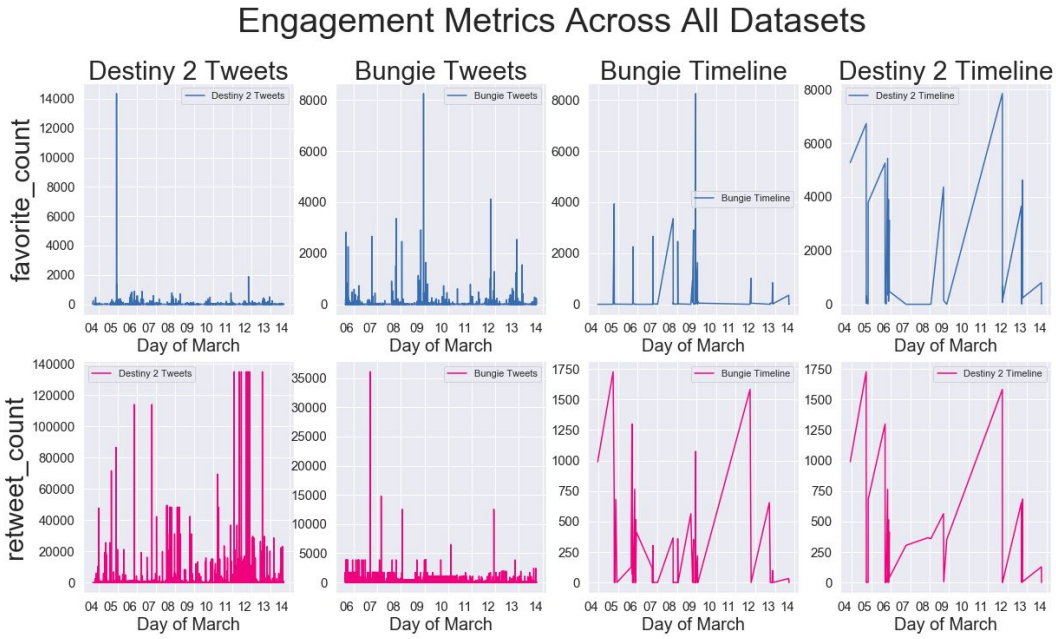
In these plots, the Destiny 2 Tweets and Bungie Tweets both have very similar values for the leftmost bin. If you look closely, you can see that the orange bar is slightly taller than the blue bar hiding behind it.

What is notable for these plots is that the overwhelming majority of the data sits in the first bin. The official accounts don't get a significant number of retweets ever (compared to the rest of the data), but they do get a competitive number of favorites. Retweets seem much higher than favorites.



I wondered if retweet and favorite counts were related, and found that it only appears that way if you log-scale both of them. I think there are a couple high outliers on each axis that make plotting these quantities difficult. Additionally, the outliers on the histograms made retweets seem more numerous than favorites, but plotting them together, it becomes obvious that most tweets have more favorites.

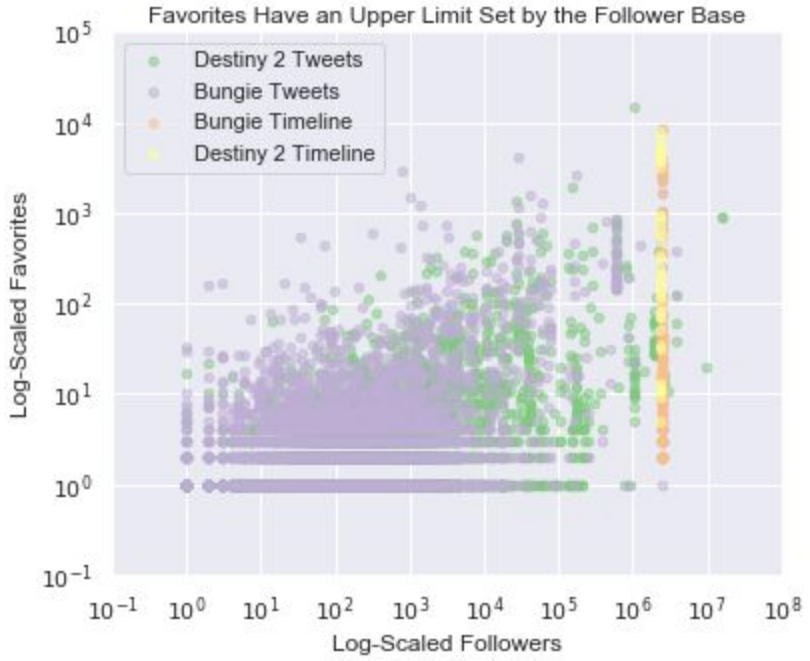
I was also interested in time correlations. I wanted to see if there is a trend in the number of retweets or favorites that tweets are getting, or if I could see that both rise and fall together over time.



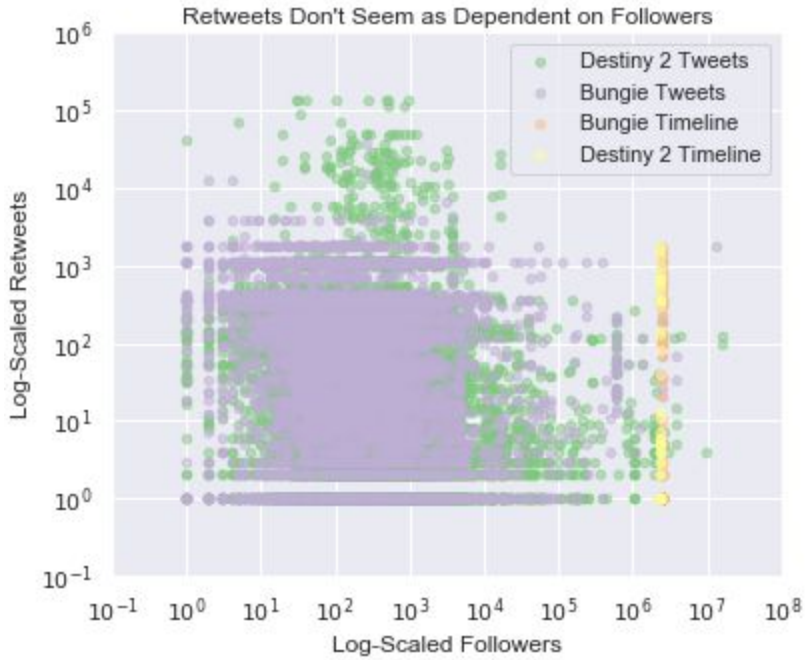
Each column is a different set of data (tweets about Destiny 2, tweets about Bungie, @Bungie's tweets, and @DestinyTheGame's tweets). The top row is favorite counts, and the bottom row is retweet counts.

The left two columns are the main bulk of tweets, and there isn't much time correlation between the retweet and favorite counts, probably because of outliers. The official accounts, however, have a decent time correlation. Part of this is probably because the official account have large follower bases that regularly interact with their tweets and that there aren't as many tweets in general. It also helps that since those columns are one Twitter account each, a lot of variables about the user posting are held constant.

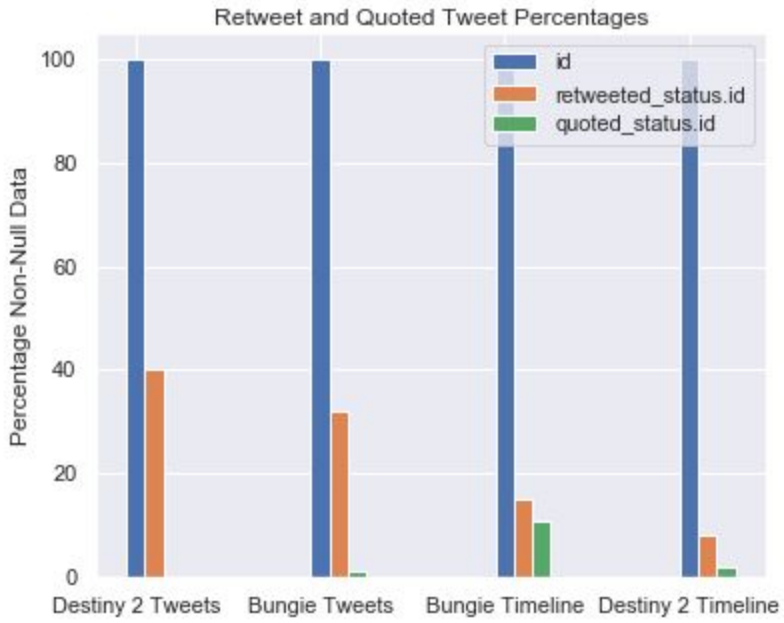
One of the biggest variables in the user is how many followers they have. It's not hard to imagine that users with a large base on Twitter will get more engagement. I wanted to check if follower counts correlated with retweet or favorite counts.



For favorites, pictured above, it seems that the follower base sets the upper limit of the favorite count, but an uninteresting tweet could always have fewer favorites that this limit.

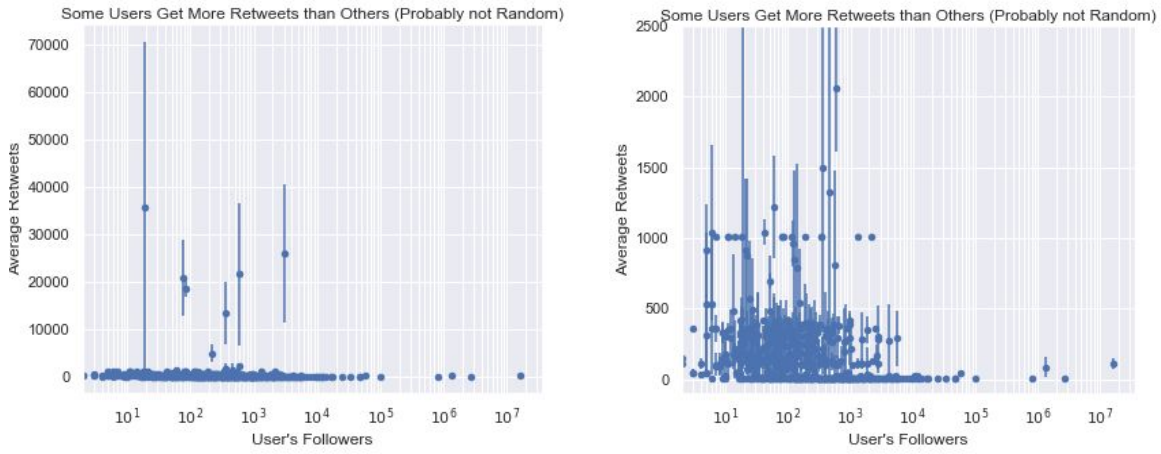


This trend does not hold for retweets. There doesn't appear to be a correlation between follower base and retweet counts, although this may be because a retweeting a retweet increments the retweet count on the original tweet.



Speaking of retweets, I realized I didn't have a sense of how many retweets were in my data. Each dataframe is plotted separately on the x-axis. Here, the blue bars are tweets, so 100% of my data is tweets. Retweets constitute roughly 30% of the overall data, differing depending on what dataset you look at. Quoted tweets are much rarer.

Looking at things from a single tweet perspective is natural, since each row of data is a tweet, but it would be good to see if certain users get more or less engagement than others, and if there are factors that help. I found the average and standard deviation of retweets for each user, then plotted it against the user's follower base.

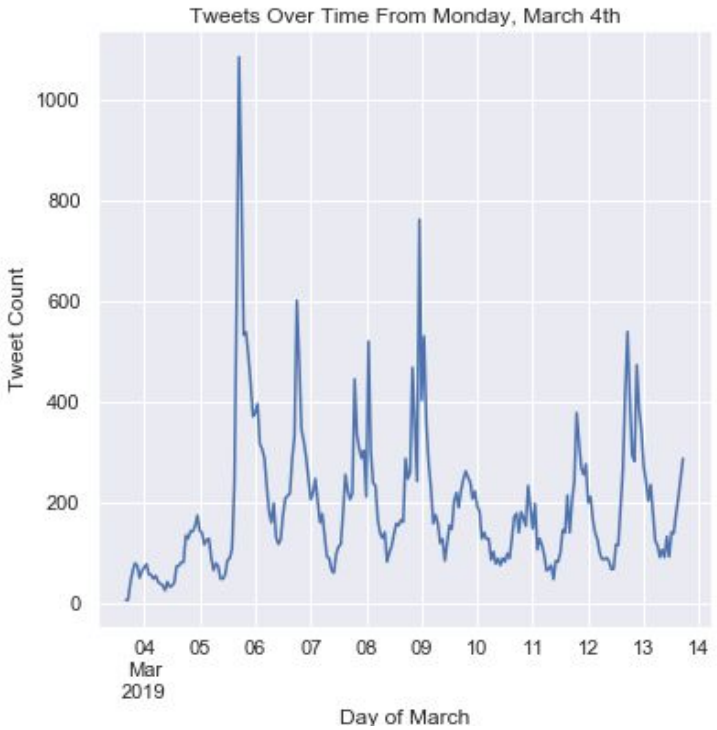


The left plot has a larger scale on the y-axis. The error bars are the 95% Confidence Intervals on the average number of retweets for the given user. Similar plots can be found for favorite counts. Overlapping error bars indicate that the difference in the average retweets for a pair of

users could just be random. Since there are many sets of error bars that do not overlap, I can tell that there is less than a 5% chance for those differences to be random.

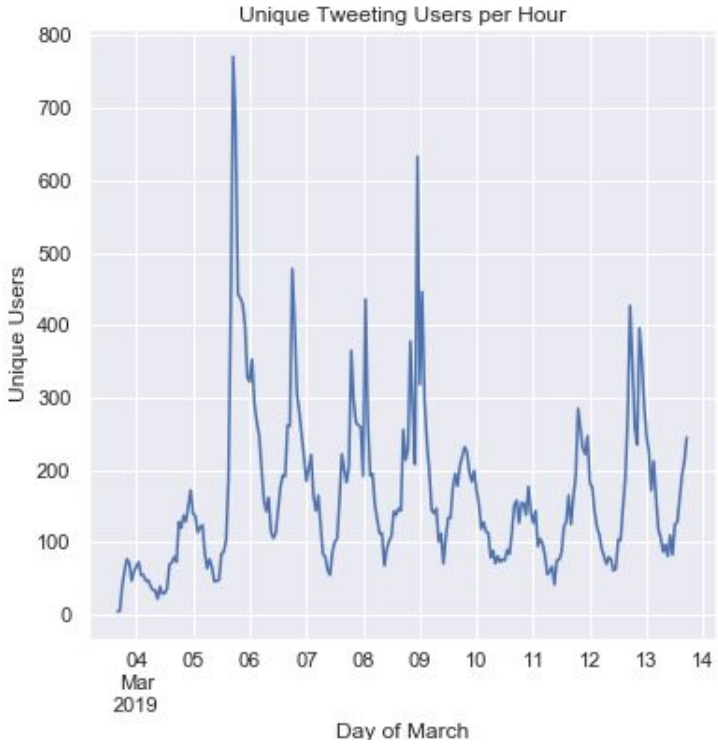
Since I can find that there is a (probably) non-random difference in retweets for some users, this begs the question: why is there a difference? My guess for now is at least follower base, posting frequency, the text of each tweet.

Now, since this is time-ordered data, I am interested in when tweets generally get posted.



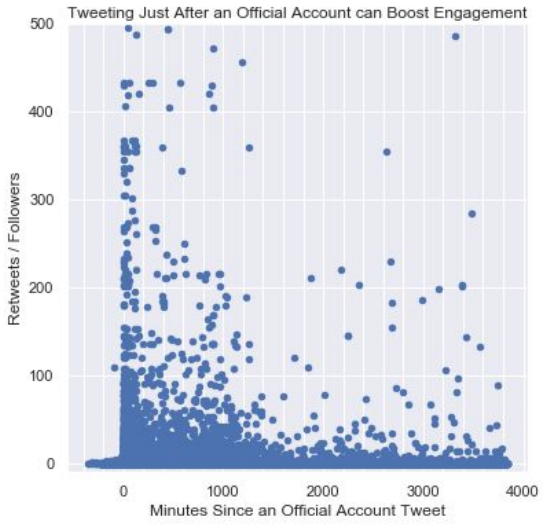
The first tweets I have are from Monday, March 3rd, 2019. The first couple peaks are artificially small. The Destiny 2 topic tweets I acquired go back to March 3th, but the Bungie topic tweets only go back to March 5th. This is just what I was able to get from the API.

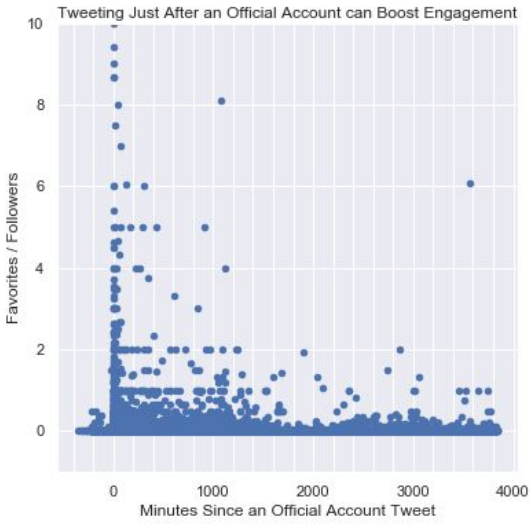
The grid lines are midnight, and we can see that most tweets are posted in the evening hours. The lower peaks on March 10-11 are Saturday and Sunday evenings. The official accounts tend not to post tweets on the weekends, which may be part of the reason why there are fewer tweets going out then.



The number of unique users who post in a given hour strongly matches the pattern of tweets in a given hour. Most people probably post 1-2 times in an hour, making the distributions very similar.

There is a thought that the official Bungie and Destiny 2 accounts can generate Twitter chatter by posting, but we can check this visually. I constructed a column for each tweet to represent the time since an official tweet was posted. I would expect that a smaller time difference will result in more engagement (retweets / favorites). The following plots support this theory.





In-Depth Analysis / Machine Learning

The first bit of analysis I was interested in was sentiment scores for each tweets' text. After doing some digging, I found that with the NLTK package, there is a library called VADER, which is an experimental NLP model focused on social media text. The main function that I used from VADER generated polarity scores for positive, negative, neutral, and overall sentiment.

The scores seemed very accurate for positive text, decently accurate for negative text, and decently accurate for neutral text (which involved lots of Destiny 2 - related proper nouns). The overall sentiment score was the most accurate.I went through and acquired these four sentiment scores for each tweet, then proceeded on to more data cleaning and feature engineering.

Data cleaning was simply removing official account tweets from the rest (since one of the features I am using is the time since the most recent official account tweet). The only other feature generation was to test if I could get more accurate results using the time since the most recent official tweet in different forms. These forms were linear (t), inverse (1 / t), and exponential decay (e^(-t)). I then rescaled the data to be between 0 and 1 and ran a train-test-split to keep 30% of the data for testing.

The first model I chose to use was a linear regression, attempting to predict how many retweets a tweet would have. I tried basic linear regressions, with scaled and unscaled data, with all three time formats, and after removing problematic negative time values. I never found a R-squared over 0.001, and many of the scores were negative. Evidently, Twitter responses are somewhat random since they rely on human interactions, so this does not follow the relatively simple model I have set up. Predicting retweet counts is very difficult.

I also tried a random regression forest, which did not work.

Since predicting the exact number of retweets was too difficult, I decided to instead put the retweet counts into three groups, and predict which group the tweet would be in based on my features. These groups were retweet counts of 0 - 99 (class 0), 100 - 9999 (class 1), and 10000+ (class 2). The problem was then not a regression, but a classification.

I used the K-Nearest Neighbors classifier, which immediately returned a 90% accuracy predicting the class of retweets. Hyperparameter tuning brought that up to 91.5%. The main source of error was predicting class 1 tweets to be class 0. However, looking at the confusion matrix, I saw that since class 2 has significantly fewer data than the other two classes, the algorithm put it at a low priority, thinking that all classes are equivalent. The class 2 predictions had a precision of 0.69 and a recall of 0.26. Only 9 of the 34 class 2 tweets in the test set were correctly predicted.

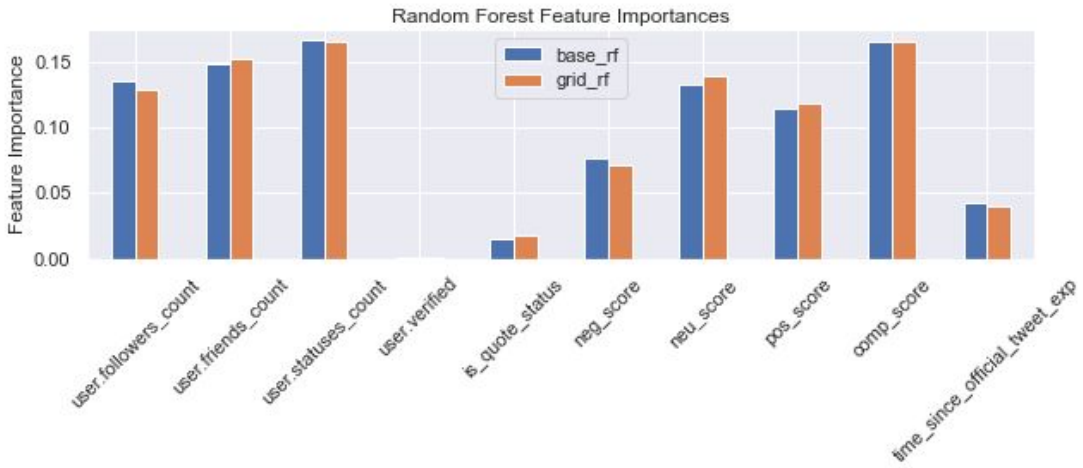
This isn't what I wanted, so I decided to upsample my data. I copied the class 1 data 4 times and copied the class 2 data 331 times so that each class then had roughly the same amount of data (~37k). Rescaling and splitting the data into training and testing sets, the base k-nearest neighbors classifier returned a 92% test accuracy! Hyperparameter tuning found a 95% test accuracy. The class 2 tweets now had the highest precision and recall, and the main error was predicting class 0 tweets as class 1.

The next problem was discovering that k-nearest-neighbors operates as a black box: I can't see how it makes its decisions and extract insights from that process. The algorithm is accurate, but I can't use it except to predict. I moved to a random forest classifier, since I knew that algorithm has feature importances.

The stock random forest had a test accuracy of 92% on normally sampled data, and hyperparameter tuning brought that up a few tenths of a percent. As with the previous algorithm, the main source of error was predicting class 1 tweets to be class 0, and class 2 tweet prediction performance was terrible (precision 1, recall 0.09).

Training and testing the random forest on upsampled data, the test accuracy was a whopping 97.3%! Hyperparameter tuning added about a tenth of a percent. This algorithm had an incredibly good class 2 prediction rate, just like the k-nearest-neighbors algorithm, and the same main source of error (predicting class 0 tweets to be class 1).

Both algorithms performed quite well, and with similar prediction and misprediction styles. This gave me confidence that both algorithms use similar features to make predictions. I then looked at the feature importances of the random forests to glean some conclusions about what makes tweets popular.

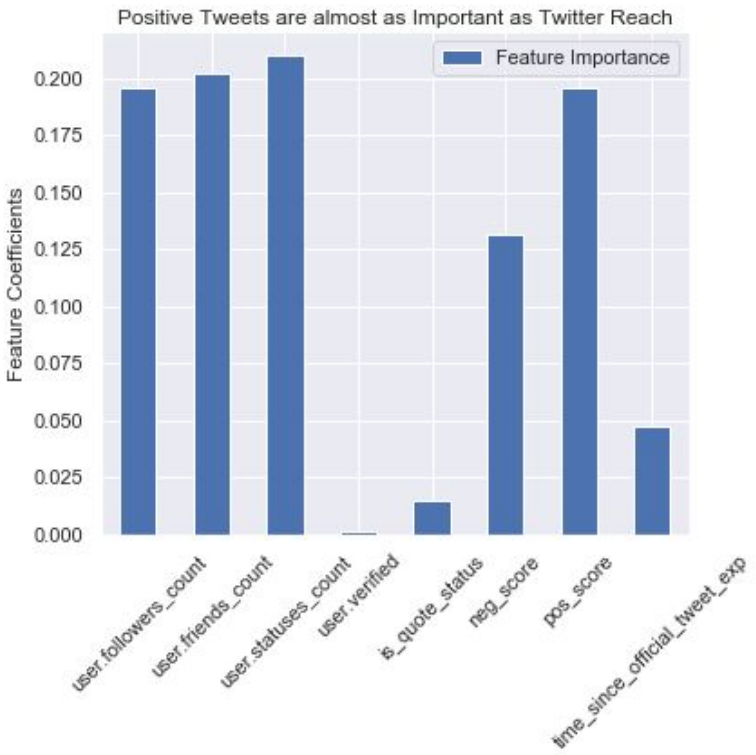


Addressing the lowest bars, we can see being a verified user doesn't matter on which class of retweets your tweet is in. Being a quote status is also unimportant. The time since the last official tweet is only a little important. What is important is the base amount of reach your Twitter account has: your number of followers, number of those you follow (friends), and the number of statuses you have posted.

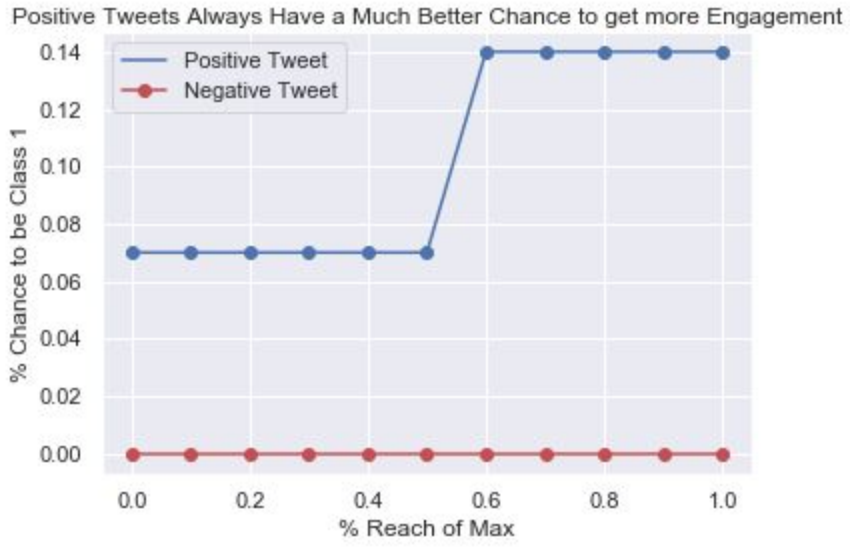
In terms of the sentiment scores, we see that the compound score has the most importance, followed by neutral, then positive, then negative. It is interesting that neutral tweets seem more important than positive tweets, but then again, I did see that a lot of proper nouns related to the game are classified as neutral, so that may explain it. The other big takeaway here is that positive tweets are more popular than negative tweets: a good result for Bungie.

One of the problems I found at this point is that the compound sentiment score uses the other three scores in its calculation. This correlation may cause inconsistent results, so I wanted to remove it and try again. I decided to stick with only the positive and negative scores, so that I could see which is more important.

I made another random forest with the hyperparameters that performed best from earlier and got a test accuracy of 97%. Looking at these feature importances below, I find that not only is a positive sentiment more important than a negative sentiment, but a positive sentiment is almost as important as the users' Twitter reach! That is very good news for Bungie, as it means people want to praise their game and will boost other tweets that do so, even if they are from small accounts.



Technically, I know that positive sentiment is more important than negative sentiment, but I don't know which direction the predictions tend toward. So, I made a function to show the predicted probabilities of being in class 0 or class 1 for data that I would simulate. This data had 2 forms: either a positive or negative score of 0.75, and the rest of the data was set as the same. I then varied the base reach numbers of follower count, friend count, and status count, holding the positive and negative scores the same. This plot is below.



Negative tweets always ended up as class 0, and positive tweets generally had a good chance to be in class 1, boosted when the reach was at least 50% of the max in my data.

Thus, I concluded that Bungie and Destiny 2 are viewed very well by their community, and they should keep up the good work.

Conclusions

Early on, I found that most tweets about Bungie and Destiny are posted on weekday evenings. There is also a slight trend that tweets right after Bungie and Destiny post on Twitter get more engagement. If Bungie and Destiny want to increase their Twitter chatter, they should post mostly on weekday evenings.

Machine learning models can predict if the retweets for a tweet will fall into certain ranges based on certain factors. One of the best insights from the random forest models is that positive sentiments attract more attention to tweets than negative sentiments. This indicates that more of the visible Twitter chatter about Bungie and Destiny is positive. The company is doing well!

These models could be repurposed for Bungie to predict if it's own tweets will exceed certain engagement thresholds and help them hit these thresholds.

Future Work

I can recollect data with a stronger emphasis on cleaning, making sure that all the tweets I keep are actually about Destiny or Bungie.

I'd want to reclassify the retweets into more groups, then seeing if the accuracy is maintained. Alternatively, I could log-scale the retweet counts and run another linear regression, since it would be less sensitive to outliers.

I should run one of the classification models on a longer list of Bungie's official tweets to see if there are certain factors that they should consider when posting on Twitter.