

# Going Viral

## *Using Machine Learning to identify what makes a tweet viral*

What does it mean to go viral? Generally speaking, going viral means that the content created is shared exponentially. In reality, most things end up going viral by accident, there is no real set formula to follow. Viral, by definition, means that the content is spread just like how a virus is spread by people. When you share something online, whether you get happy or sad or whatever emotion you have, you share it to invoke those same emotions on other people.

### **Defining Viral Tweets**

There's no real number that can identify or claim viral status. We can look at specifics like the number of shares or likes or some other measure of interaction, but at the end of the day that doesn't really matter. Everything is relative. For example, your favorite celebrity might have hundreds of thousands of followers and get thousands of retweets on every tweet, but that doesn't mean that tweet is viral. Meanwhile, if you get a few hundred retweets when you're only used to like 10 or 20, you could make the case that your tweet went viral.

For this, I'll be using a supervised machine learning algorithm. More specifically, I will be using a K-Nearest Neighbor classifier. In order for this to work, I will have flag each tweet in the dataset as viral or not viral. This means that I have to define viral status.

To get a good idea on what I should consider viral, I'm going to be looking at the number of retweets a given tweet has. A high number of retweets indicates that the tweet has been shared many times and has been exposed to a wide variety of people. But now the question is, how many retweets should I consider?

In this case, I believe that the median number of retweets for the entire dataset serves as a good baseline. The median is resistant to being heavily skewed by high and low numbers, which makes it perfect for this application. The median for this data is 13 retweets.

### **Features**

Now that these tweets have been flagged as viral or not, it's time to add in some features that might determine whether a tweet is viral or not. For this application, I will be looking at:

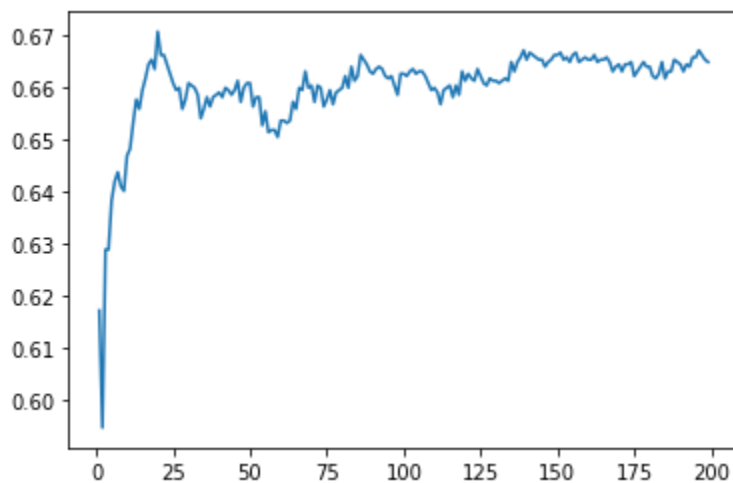
- Tweet Length – The Length of the Tweet
- Follower Count – The number of followers the person who sent the Tweet has
- Friends Count – The number users the person sending the Tweet is following
- Favorites Count – The number of “likes” the tweet received

- Number of Words – How many words are in the tweet

The difference between follower and friend count is a little confusing, but it is important to know the difference! This should make it a bit more clear: Followers are the users following me, friends are the users that I am following.

## The model and Score

Using K Nearest Neighbors method of classification, the data will be separated into two groups: One used to train the classifier, one used to test how well it performs. On the graph below, the x-axis represents  $k$ , the number of neighbors. The y-axis represents the *score* associated with that value of  $k$ .



For this model, using the features tweet length, followers count, friends count, favorites count, and number of words, the model is correctly classifying 67% of the time. It's better than random, but without any hard values to determine if something is viral or not, I don't think it will get much better.