Mini Project 3: Text Mining
Uma Desai
SofDes 16 Spring Section 2

Project Overview

For this project I chose to use Twitter as my data source. I utilized the web-mining tool Pattern to analyze tweets for their subjectivity/polarity in order to gauge the general degree of positivity or negativity and make predictions centered around each of the main contenders of the upcoming March Madness tournament. I hoped to use the results of the data analysis to make predictions about the outcome of the tournament by the level of positivity of tweets about each team, weeding out tweets that were outputted to be significantly subjective by Pattern.

Implementation

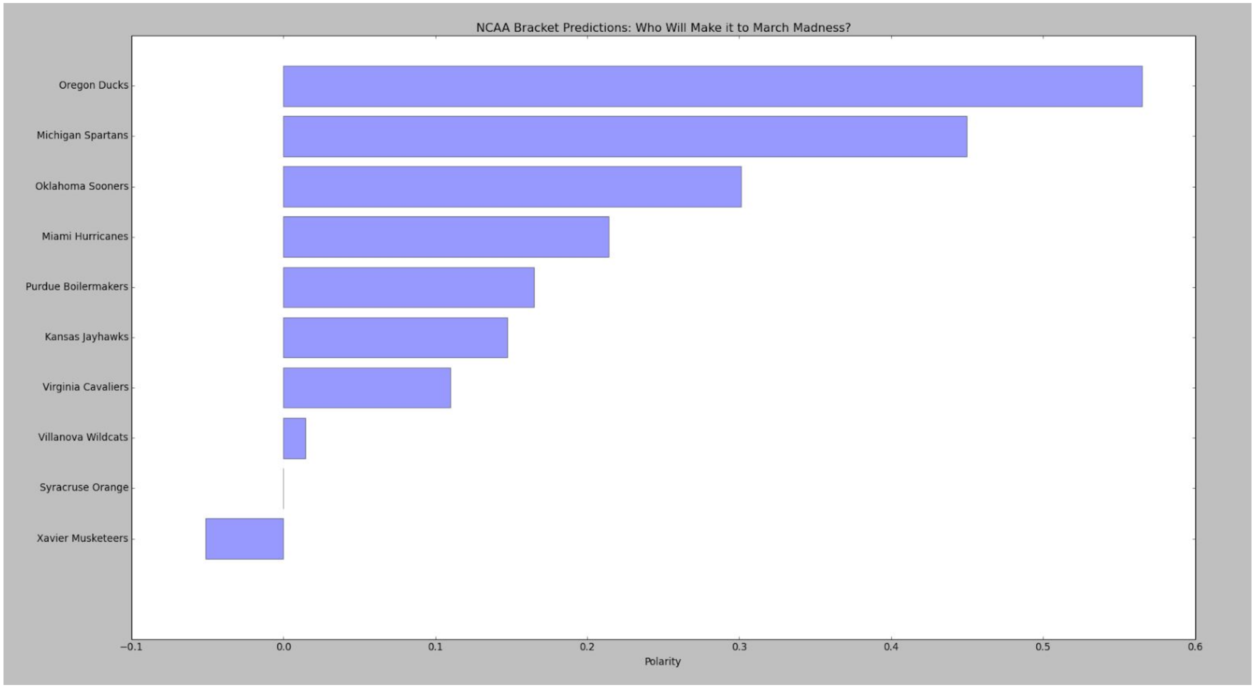
The basic idea behind this Twitter Predictions program is that the user inputs a number of keywords or hashtags which are used to collect a list of a specified number of tweets that can then be analyzed for polarity and subjectivity using Pattern's sentiment command and returned as a tuple. The average polarity for each search is then combined with the list of original inputs into a tuple which is sorted from greatest polarity (positivity in recent tweets) to least polarity. The key players in this process were Twitter as the data source, Pattern for its sentiment analysis tools, a simple algorithm that organized the results into a tuple and sorted them for the ranking, and Matplotlib for its 2d plotting tools.

An interesting choice that I had to make about the program was when I was unit testing and realized that many of the tweets were outputting polarity that was incredibly opinion-based and had no factual evidence. I realized that I needed a way to filter out some of the tweets that were completely objective and shouldn't be accounted for in making actual predictions. To do so, I implemented the subjectivity measure, which I had originally left out, back into the measure_sentiment function and used an if statement to filter out tweets that were significantly objective. After another few rounds of unit testing (and getting blocked when I ran the search for tweets function too many times), I realized that the tweets being used to measure the predictions were significantly more reliable and garnering more accurate results.

Results

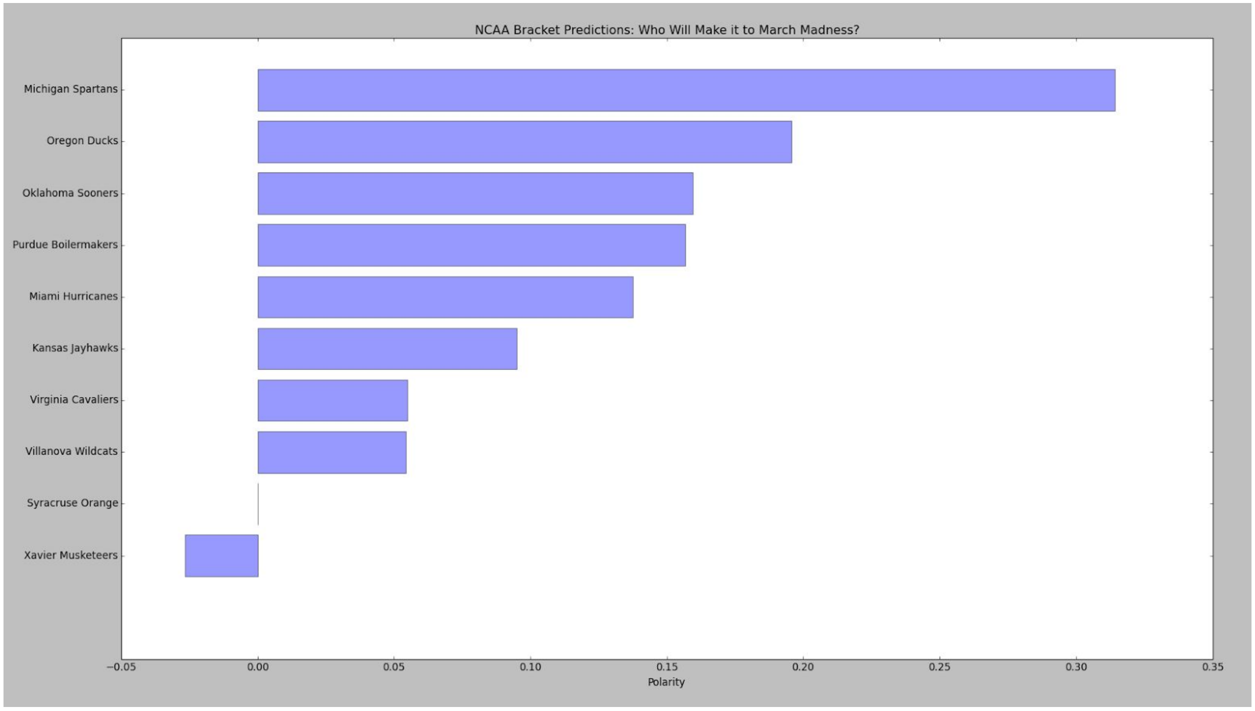
My program allows the user to input keywords or hashtags from Twitter and returns the inputs in a horizontal bar graph ranked from most to least positive prospects based on the level of positivity being seen in the most recent tweets about that topic. My program also seeks to filter out tweets that are determined to have a greater level of subjectivity than 0.5 by Pattern in an attempt to analyze the results that are more factual than opinionated and thus get more accurate results. One of my first ideas when brainstorming implementation of this program was to analyze tweets about some of the top NCAA teams and make predictions about who would make it to March Madness. I got some interested results.

Figure 1:



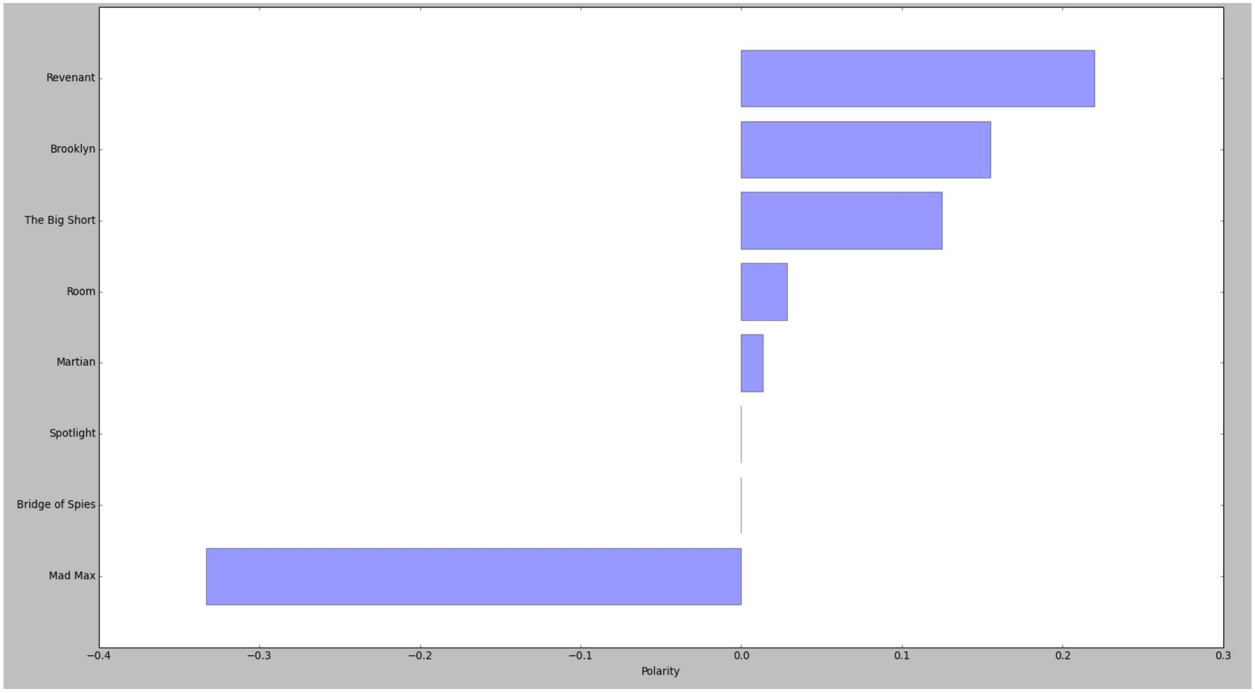
This is the chart that my program outputted the first time I ran it. The y-axis shows the keywords for various NCAA teams that I inputted into the program, and the x-axis shows the average polarity that was outputted from the tweets that were collected. I ran the program to look at the 15 latest tweets for each of the inputs, so 150 total tweets were analyzed. To get more accurate results, this number should be increased greatly, but because of Twitter's search engine limit, I ran it with these numbers.

Figure 2:



I ran this same program again just fifteen minutes later and got different results as seen in figure 2. Many of the teams seemed to experience a decrease in polarity, including the Michigan Spartans, and yet they increased in ranking. These changes may be mostly due to the relatively small number of tweets the program is collecting, but it could also be due to recent events that may changed a team's prospects. These results are exciting because they are indicative of the program's ability to track events and how they affect perceived positivity surrounding a team's prospects. This program is also really cool because it can be applied to many other scenarios and still get meaningful results, such as the presidential race or Oscar predictions. Here's an example of the results I got when I inputted the nominees for the Oscar award for Best Picture.

Figure 3:



Reflection:

Thinking back on the project, I realize that there were some aspects that were really hard to unit test. I had to print the tweets that were being analyzed and see if they made sense with the subjectivity and polarity ratings that were being outputted by Pattern, which is pretty hard to do considering that Pattern also has significant limitations in its ability to pick up sarcasm and emotions. I regret not using a program other than Pattern such as Indico to get more accurate readings on the polarity/subjectivity of tweets. The limitation on the number of tweets that could be collected at once was also a huge obstacle that significantly limited the accuracy of the results. Going forward I will look into making sure there are not significant limitations on my data source before centering an entire project idea around it. I am, however, very happy about how many exciting new tools I learned through this project as well as the many applications this project has. It was also very useful to be able to create plots of my results through Python, which is something I've never done before. Moving forward, I know to do more research on the possible limitations of an idea before deciding my project topic.