**Progress Report**

**- Increment 2 -**

**Sentiment Analysis on Twitter**

# Team Members

Jacob Wharton; jtw18f; eyeCube

Andre Guiraud; alg17d; alguiraud

Oscar Kosar-Kosarewicz opk18 Kosaro

1. **Project Title and Description**

Our project is called SentimentAnalysis - Twitter. It will allow users to track the sentiments of tweets posted on Twitter, possibly over time and by location. The sentiments are regarding user-supplied keywords or phrases, the matching tweets for which are mapped to a floating point "quality" value from -1 to 1 using the machine learning model. A graph will display this output in a user-friendly way. We will use the TextBlob Python library to analyze the raw text from the tweets gathered using the StreamListener which accesses the Twitter API.

1. **Accomplishments and overall project status during this increment**

GitHub repo: [https://github.com/eyeCube/SentimentAnalysisTwitter/](https://github.com/eyeCube/SentimentAnalysisTwitter/tree/master/Model%20Training/machinegym)

We have working code that can analyze text such as a tweet and output a sentiment rating for how well it matches a sentiment like anger or happiness. The output can be either “analog,” anywhere from -1 to 1, or “digital,” being -1, 0, or 1 precisely.

A version of the website was successfully deployed on AWS with expected functionality. We are also able to send emails based on database parameters to alert users of queries they were interested in that have become available to view. This task is multithreaded to avoid disrupting requests made to the website by users, as it must constantly run to verify the database.

We have added a way to measure the performance of a model, allowing us to pinpoint sources of error and improve the model’s accuracy

1. **Challenges, changes in the plan and scope of the project and things that went wrong during this increment**

MachineGym (Jacob): One thing that was more difficult than I thought it would be was getting the grammar of the sentence to play a part in the sentiment quality output. It’s more complicated than I at first thought -- you cannot simply examine what word comes before a matched word to modify that word’s quality value. This would work to some degree but it would fail in perhaps just as many cases. Consider a more complicated grammatical sentence structure involving multiple adjectives before a noun, or any sequence of double-negatives, or any sarcasm, or other subtle conversational queues that would be difficult to pick up on without a machine learning algorithm to do that work for you. The best I can do is make a super-dumbed down English text parser that maybe just slightly more-often-than-not happens to output a useful value; if I can achieve that much, I’ll be happy.

I think my original scope may have been too prestigious. In the end my algorithm will probably be very poor at outputting the actual sentiments of a piece of text, just because making an English parser entirely by hand is far too vast a goal to aim for in such a limited time and with such limited resources. Basically, it will just do the best it can at increasing the amount of usable training data we have available, but as far as how efficiently it will be able to produce that data -- well, it probably won’t be winning any awards for performance.

ML Model (Oscar): I have found that despite attempts at running, an SGD Regressor, although functional, yields unsatisfactory performance and we will replace is with a Naive Bayes classifier, which should be more suited for the task. If it does not improve performance to our satisfaction, I will identify an alternative dataset that is better suited for tweet analysis than amazon reviews.

Website (Andre)

For the website, rather than styling everything ourselves we have opted to utilize Bootstrap and its pre-existing CSS and JS libraries. So, the website currently uses these libraries to present a professional appearance without the need for writing CSS/JS code from scratch.

For increment 2, getting the website to run in AWS was a pain because the default version of SQLite installed in AWS EC2 machines is older than the one required to run our Django version. After reading through the error logs and figuring out this was the problem I was able to update the SQLite version in our EC2 instance. After reloading the application it was good to go. However, this was an old version of the webapp, mostly for me to get familiar with AWS and elastic beanstalk.

The email scheduler was also a bit of a problem to get working. It requires modifying some of the Django setting files and overloading a function that is initialized every time the server is started. Once the scheduler was working, getting the email to format correctly was a problem that I was unable to figure out. I wanted to send the emails as HTML with a specific format, but the emails were being sent as plain text, so I decided to just leave it as plain text for this iteration.

1. **Team Member Contribution for this increment**

Oscar Kosar-Kosarewicz:

Progress report: IT sections 1 and 2, RD functional requirment section, spoke and edited the video

Source Code: Acquired amazon reviews training data, processed it and used to to train an SGD Regressor prediction model.

Jacob Wharton:

Progress report: contributed to sections 3,4,5,6. Worked on expanding machineGym.py

Requirements/Design doc: contributed to sections 2; formatting document.

Source code: all files within the ~/machinegym/ directory in GitHub repository (new files: angry.py, peaceful.py)

Video: everything relating to machineGym (recorded video/audio files)

Andre Guiraud:

Progress report: contributed to sections 3, 4 and 6

Requirements and design document: contributed to sections 2 and 6

IT document: contributed to sections 2, 3 ,4 and 5 plus formatting

Source code: everything contained in the website branch, within the project folder ‘SAT\_Website’

Video: all parts pertaining to the website

1. **Plans for the next increment**

Jacob plans to complete the basic sentiment analysis for more generic emotions perhaps like loneliness and connection. He also plans to continue to improve the algorithms for handling more complicated grammatical constructs, and to possibly add more words to the dictionaries. Finally, phrases need to be implemented, as the parser currently can only view words one at a time, but there are lots of phrases I plan to identify, too. I also want to add logic for the parser to identify emotes, both emojis and traditional-style emotes using punctuation e.g. “=)”.

Andre

Continue adding styling and making it consistent, right now it’s a blend of different things I am testing. Additionally, the plan is to combine everything into the webapp to get a preliminary version running with everyone's code. Hopefully, we can get this version of the website to run in AWS with all functionality.

Oscar

Replace SVB Regressor with Naive Baye’s classifier, possibly find alternative data set and integrate with Jacob’s Machine Gym

1. **Link to video**

https://youtu.be/b74ZMGvg7UU