**Software Implementation and Testing Document**

**For**

**Group Sentiment Analysis Twitter**

Version 1.1

**Authors**:

Jacob Wharton

Andre Guiraud

Oscar Kosar-Kosarewicz

# **Programming Languages (5 points)**

Python - we are using this language due to its flexibility. It has excellent data science libraries which we need as well as mature web development libraries, allowing us to integrate the components of the project more easily. In addition, all of our group members are experienced in Python.

# **Platforms, APIs, Databases, and other technologies used (5 points)**

* Github for version control.
* Django for web application development
* AWS to acquire training data
* AWS (hopefully) to host the web app
* Tweepy to gather tweets
* TextBlob to parse text and write algorithms for language comprehension
* Kibana (possibly) for browser-based visualization of output data
* Schedule to schedule email sender for continuous run
* Bootstrap library for web styling

# **Execution-based Functional Testing (10 points)**

*Describe how/if you performed functional testing for your project (i.e., tested for the* ***f unctional requirements*** *listed in your RD).*

MachineGym: (Jacob) tested the output of the text analyzers by inputting simple text designed to match one or two hashtags or words in the text, where I already knew what the output should be. I did this repeatedly for various sentiments and for different combinations of words and hashtags.

Website (Andre): Website was tested running on a local server, along with gmail for sending emails. AWS was also tested with a simpler version of the website.

# **Execution-based Non-Functional Testing (10 points)**

*Describe how/if you performed non-functional testing for your project (i.e., tested for the* ***non-functional requirements*** *listed in your RD).*

Website (Andre): Database integrity was tested by attempting to load blank data into the database. It was rejected, maintaining the database integrity. Also, if an email that already existed is entered a duplicate is not created.

ML Model (Oscar): Developed scripts that generate test data and test model’s performance for improved analysis

# **Non-Execution-based Testing (10 points)**

*Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*

MachineGym: (Jacob) I always do a code-walkthrough where I re-read everything I wrote, looking for any logic errors, errors in the flow of the program, or any areas where I made a note to go back and change something.

Website (Andre): For the email scheduler a lot of code-walkthrough to ensure that everything was being called where necessary, arguments were being passed correctly, etc.

I tested the HTML code on the browser before running the website too, to try out different bootstrap styling options.