Twitter Sentiment Analysis of Apple Products by Region [Literature Review]

# Introduction

It is crucial for companies to consistently identify customer feedback, sentiments towards upcoming products and measure brand loyalty. Apple Inc‘s reputation has declined over the past few years. Users generally have had a negative response to their previous few product releases, as products fail to achieve any new breakthrough in technology or design, iteration after iteration. I am interested in classifying tweets related to four main apple products: ipad, iphone, Mabook, and Applewatch by measuring the tweets polarity by region to derive insights about product affinities across North America and how the customers feel about each product.

In this paper, two datasets are used, one with tweets with sentiments pre-assigned to each tweet and the second curated by myself from Twitters search API. The preassigned data to used train five classifiers: Naive Bayes, MultinomialNB, BernoulliNB, Logistic regression and Linear SVC using Scikit learn in Python. Both datasets are first pre-processed using techniques such as tokenization, stop-words removal and other techniques which will be detailed in the dataset section. Once the classifiers are trained, the data from the Twitter API is pre-processed using the same techniques and feed into the classifiers. The data is visualized in a dynamic graph using Tableau to show the sentiments towards each of the products, with the ability to toggle between various cities and filter by polarity of the tweet.

# Literature Review

In recent years, sentiment analysis of data, generated from microblogging platforms has become very valuable. NLP techniques are constantly improving and are able to derive insights from consumer opinions with greater accuracy. Sentiment Analysis can be a challenging task because natural language is complex. The challenge is that when speaking naturally, in a casual environment such as Twitter, we do not need to follow strict language rules to express ourselves and to be understood. We use slang, sarcasm and more to express ourselves.

In the past few years, the best method of sentiment analysis has proven to be a lexicon-based approach. A lexicon-based approach splits up the individual words in the text and assigns a sentiment value to each word based on a bag of words to compare with, and the computes the sentiment polarity of the entire tweet. This approach is quite effective but only if each tweet is known to be an opinion. In other words, if the tweet has text that is descriptive or is a quote for example, then this approach is not entirely accurate. *Mudinas, Zhang and Levene [1]* propose a solution that combines lexicon based approaches with Learning based approaches. Learning based approaches use unsupervised learning done on a massive corpus of content, many billions of sentences to train the models by example. They conclude that a pure learning based system is more effective than a pure lexicon-based system, and that a combined Lexicon and learning system achieves the highest accuracy.

In a study by *Xie, Vovsha, Rambow and Passonnau [2]*, instead of using a bag-of-words model, they used a bigram model which store sets of words and stores the frequency of those terms together. They conclude that this is more accurate that a unigram approach but point out the absence of any semantic analysis in their study to check for words that may have multiple meanings depending on context.

Another recent study on this topic by *Kouloumpis, Wilson and Moore [3]*, looked specifically at hashtags and emoticon data parsed from the tweets to predict sentiment. They use lexicons and n-gram for features along with Part-of-speech features in their tests. They conclude that lexicons and n-grams are still effective in sentiment analysis for hashtag and emoticon data but including the Part-of-speech features actually reduced accuracy significantly.

Dataset (Source: https://data.world/crowdflower/apple-twitter-sentiment)

There are two datasets being used in this study. The first dataset, which contains 3125 tweets, is from CrowdFlower called Apple computers Twitter sentiment takes a look into the sentiment around Apple, based on tweets containing #AAPL, @apple, etc. It has sentiment values pre-assigned to each tweet by contributors that were given a tweet and asked whether the user was positive, negative, or neutral. The second dataset is curated by myself using Twitter’s search API and contains 10,000 tweets for each product. A python script that searches for phrases related to each of the four apple products is used to create four separate json files with the Twitter data. The json files contain various attributes of the tweet. For this study, I have extracted the text and location (city, state) from each tweet.

## Approach

## Step 1 (Split dataset)

I used the data from CrowdFlower as the training dataset for the classifiers. The first preprocessed the tweets by reading the files into Python and splitting the dataset into train and test csv files.

## Step 2 (Preprocessing tweets using NLP techniques)

I then imported these files into Python dataframes using Pandas. A very useful module in Python library called scikit-learn has functions that convert a collection of text documents to a matrix of token counts, which are used as our bag-of-words for our linear classifier. I removed remove punctuations, lowercase, remove stop words, and stem words from the text, which can be directly performed by CountVectorizer function. I also use a simple regular expression to remove non-letter characters. I define PorterStemmer function for each item and use function (nltk.word\_tokenize) to tokenize the data. I intend to process the data further to remove url’s and hashtags, as well as lexicons and n-grams to further improve the model.

Github: https://github.com/aryankaushik89/Capstone\_Ryerson/tree/master

## Step 3 (Train classifiers) + Step 4 (Improve Accuracy)

I have fed the data through a logistic regression model as of yet and recorded a 72% accuracy at this stage. I intend to train Naive Bayes, MultinomialNB, BernoulliNB and Linear SVC classifiers as well.

## Step 5 (gather secondary dataset using Twitter search API)

To use the twitter API, I first created an account where they provide you with access keys required search Twitter. In Python, I started with installing a library called Weepy which allows you to communicate with Twitter. From there I fed the access keys and wrote an algorithm to save the tweets into a json file. The API has a limitation of a hundred tweets per 15 mins, so the script repeats the process until stopped.

Github: https://github.com/aryankaushik89/Capstone\_Ryerson/tree/master

## Step 6 (Preprocessing Tweets from API)

In this step, I will use the same approach to preprocess the text for analysis as in step 2.

## Step 7 (Feed data through classifiers)

Once I have applied all of the NLP techniques and trained the classifiers with the training data, I will then feed the data collected for each of the four products through the classifiers and predict sentiment values.

## Step 8 (Visualization)

In this final step, I will use Tableau to visualize the data gathered from the API to graph the sentiments of twitter users in each city and for each product. Additionally, I will create word clouds to visualize the positive and negative words used to describe each product.