Sentiment Analysis of Smart Devices Topics from Twitter Data

Allan Lo, Eric Yang, Prajakta Pandharkar

Problem Definition

Social media platforms offer a tremendous opportunity for marketing professionals and data scientists to analyze and discover valuable insights from customer sentiments and opinions associated with products and services, current market trends, and real time evaluations of marketing campaigns that can help inform better business decisions. For this project, we will focus on developing a sentiment analysis system that collects and extracts twitter data on topics related to smart devices and provide analytical results on customer sentiments. Specially, we will address the following questions:

1. What is the general perception, opinion, or sentiment on twitter toward smart handheld devices, including smart phones, tablets, and smartwatches across different brands?
2. What categories of audience is the above smart devices attracting or turning away? Does it have any impact on the brand of the company?
3. Which companies have the best or worst sentiments associated with the products and how do they change over time with specific events (product launches)?

Data source

As part of initial analysis, we narrowed the sources to Twitter API. Twitter APIs will provide us with options of Streaming API for live streaming to capture customer sentiments during product launches by companies, and, events such as conferences and REST API for collecting tweets with little latency.

In order to classify the tweets, we plan to use a list of devices such as cell phones (iPhone, iPads, apple watch, tablets and their branded companies. We plan to collect only those tweets that contain our narrowed list of keywords or hashtags. This will make sure that we are pulling only reasonable amount of data from the tweets. We will also use some keywords and phrases for sentiment analysis. In addition, we plan to check whether we can capture tweets and retweets by companies' executives and reaction from public.

For scraping the tweets, we plan to use “Tweepy” for Twitter APIs with Python scripts to download high volume of data in real time quickly. Tweepy establishes a session and it uses Twitter’s much required ‘Oauth’ authentication for the streaming session. It also provides an interface to iterate through different types of objects that Twitter offers. It allows us to limit the number of tweets by downloading them one tweet per line and to stay within limits of the Twitter agreement through Python scripting. In addition, we will restrict the keywords using Python scripts in order improve the performance later in the stage. Twitter API provides a JSON response in the form of dictionary with _json which provides structured way to collect the text over just raw text. This will provide us ability to collect tweets and store them in JSON format and can be easily converted to required models for further use.

High Level Architecture

In the next step we plan to build a data lake using HDFS on Amazon Web Services to store the data. We will plan to use hadoop utilities to copy the files to HDFS. We will utilize a Hive Database as the backend and Spark as the computing framework. We're also planning to explore using NoSql DB such as Mongo DB. Our understanding is using Mongo works well with JSON files and could provide a better solution for Twitter data.

We will have an AWS EC2 instance that performs the sentiment analysis and store the results in the database. By utilizing a data lake architecture with an ELT framework we'll have more flexibility for our analysis when we move to the modeling/analysis phase. We will use Spark/PySpark to perform our ELT tasks. We will rely on a schema on read framework when we need to define entity relationships. For visualization, we will present the results through a HTML portal that enables python API calls as our primary tool and R/Shiny as our secondary option for generating summary and visualization of queries from the database.

Model/Method

The main output of our model will be to determine the sentiment related to the product launch. In addition we will track sentiment over time and determine if the sentiment we find on the platform is consistent with the overall success of the product. Sentiments will be segmented into three broad categories; positive, neutral, or negative. In order to determine these three categories we will develop keyword lists that are associated with each. For example words such as good, great, best, etc. occur in a tweet with a specific product we will categorize the tweet as positive sentiment related to a specific product. To accomplish this and also account for context we will rely on existing datasets that have already categorized sentiment as a training dataset. It will also be important for us to focus on text analysis. For this, we plan to use tokenization using some out-of-the box libraries such as NLTK to split the stream of tweets into smaller units called tokens of words or phrases. Using these methods we should be able to directionally determine sentiment for product launches. One of the biggest challenges we will face is understanding the context of tweets. For example if two negative words are used in a tweet this could actually be positive or neutral sentiment. In this case relying on our existing datasets/libraries could become more important. In addition we will look into utilizing emoji sentiment analysis which could be more straightforward and it supplements the text sentiment analysis.

Project Plan/Summary

Data Acquisition

- The first phase of our project will begin with data acquisition. We will first obtain a sample dataset from Twitter to see what data is publically available through their API. **(10/21) -All Team Members**
- We will then build a python script to reliably acquire the data, with a planned daily cadence of appending tweet data to our database. **(11/4) - Eric**

Data Architecture

- Once we have setup our data acquisition through the twitter API we will focus on building out the architecture of our data lake on AWS. This will include setting up HDFS and query applications such as Hive/Mongo and Spark. **(11/11) - Prajakta and Allan**

Testing

- With the initial acquisition and architecture framework built we will test our system ensuring that we could reliably pull in data and store it in our data lake. **(11/18) - All Team Members**

Modeling/Visualization

- The final phase will be focus on building our sentiment modeling capabilities and utilizing querying applications/visualization layers to output insightful sentiment analysis.   **(12/1) - Allan**