Spark Near Real-Time Sentiment Analysis

*Mohan K. Patnam, CN Chen, Bruno Janota (CSCI E-63)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Motivation:** *This project examines whether social media feeds focused towards stock market and concerning companies, are a suitable data source for forecasting the direction and volatility of the underlying stock prices. In particular, we focus on StockTwits.com which is a popular social media platform where real investors, traders and general public alike express their opinion on companies and their performance in the form of tweets. We begin this project by providing a generic approach to download historical tweets for the chosen stocks over a chosen period. We then introduce a couple of models to calculate social sentiment scores using two variations of Bag-of-Words (dictionary) and a Naive-Bayes classifier. We apply basic text processing techniques using NLTK to preprocess tweets using the NLTK lemmatizer and stopwords. We further perform regression analysis to see if social sentiment can serve as a leading indicator to predict changes in underlying stock prices. We demonstrate an ability to sense the market pulse or sentiment using a near real-time stream of tweets. Our aim here is illustrate the use of modern machine-learning and real-time parallel processing technologies while answering a fundamental question of whether social sentiment matters for stock market performance.*.

**Data set**:   
1. Two months worth of tweets for three chosen stocks (**TSLA, FB, AAPL**) from [www.StockTwits.com](http://www.stocktwits.com)

2. Two months of historical stock prices from <http://finance.yahoo.com> for the same set of stocks.

3. Intraday stock prices from Bloomberg Finance for the same set of stocks.

**Technology**

Python and various API - StockTwits RESTful, NLTK, Spark Streaming, Tensorflow API to implement various phases of the project.

**Benefits**:

* Ease of use - python and its seamless integration with web downloads, text analysis, spark streaming and machine learning packages.
* Ease of web downloads using RESTful API.
* Ease of natural language processing using NLTK lemmatizer and stop-words packages.
* Parallel processing using spark streaming API by simultaneously running sentiment analysis on multiple stocks.

**Challenges**:

* StockTwits RESTful API has a limit of 200 requests per hour. Each request returns 30 tweets for a chosen stock in JSON format.
* Signal-to-noise ratio in the user tweets. Tweets are often highly unstructured in an informal language. A significant portion of noise can be filtered using appropriate dictionary of words and better processing of text (stop-words, n-grams, lemmatization). Pre-processing has implication on the accuracy of the sentiment prediction models.

**Demo Workflow:** We implemented various phases of sentiment analysis as a work-flow.

* Download tweets for a stock (TSLA, FB, AAPL) using StockTwits RESTful api and json processing. Output is a csv file with each record representing one tweet having following fields -   
  *TweetID, Stock, Date, CreateTime, Text, Sentiment*
* Preprocess tweets csv file. Remove stop-words first. Then perform lemmatization to reduce the dimensionality for different usages of the same word (organize, organizes, organizing).
* Perform Bag-of-Words analysis to calculate sentiment score using two dictionaries. Loughran McDonald (Financial) dictionary and the Harvard dictionary. Present confusion matrix for model accuracy using calculated vs presented sentiments in each case.
* Perform Naive-Bayes analysis to predict sentiment score by training the model first and verify using the test data. Present confusion matrix for model accuracy.
* Plot sentiment score using spark streaming of tweets in real-time.
* Perform Regression analysis (linear, cubic) using Tensorflow api to see if changes in sentiment score can predict changes in underlying stock prices.

**Results and Conclusions:**

1. Provide a generic approach to download raw tweets from a well known social media platform.
2. Predict daily sentiment score for a stock from its raw-tweets using multiple models.
3. Provide real time sentiment scores using spark streaming of raw tweets.
4. Understand the relationship between
5. sentiment score, tweet volume, trade volume and close prices.

**Documentation:**

Github repository for code, data and logs - <https://github.com/mpatnam/CSCIE63-Project>

YouTube URL of the full presentation video: [http://youtu.be/….](http://youtu.be/1MV0AgFIf9I)

YouTube URL of the 2min preview presentation video: [http://youtu.be/….](http://youtu.be/1MV0AgFIf9I)