Spark near real-time Sentiment Analysis

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

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

**Motivation:** *This project studies whether social media feeds focused towards stock market and concerning companies, are a suitable data source for forecasting the volatility in 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 for a chosen period. We then introduce couple of models to calculate social sentiment score using Bag-of-Words (dictionary) approach and Naive-Bayes classifier. We apply basic text processing techniques to preprocess tweets and help build these models. We further perform regression analysis to see if social sentiment can serve as a leading indicator to predict underlying changes in stock prices. We demonstrate an ability to sense the market pulse or sentiment using a near real-time stream of tweets. Our quest here is to help answer a fundamental question on whether social sentiment matters for stock market performance.*

# Project Phases

This project is divided into 4 phases. The first phase of the project focussed on data download and wrangling into a suitable format for further processing. We collected historical tweets for chosen stocks for a period of two months from [www.StockTwits.com](http://www.stocktwits.com) using their restful API. As per our research on investopedia.com, there are few stocks that are very popular on stocktwits platform ([Most followed stocks in stocktwits.com](http://www.investopedia.com/articles/markets/021316/most-followed-stocks-stocktwits-nflx-fb.asp)). We chose 3 such stocks for our sentiment analysis in this project - AAPL, FB, and TSLA. We also collected historical stock prices (close price, traded volume) for the same stocks for the same period from Yahoo! Finance (<http://finance.yahoo.com>).. The second phase of the project aims to build couple of models to predict the social sentiment in the form of a weighted score. We use this score to categorize the social sentiment of a tweet into one of the 3 categories - Bullish (positive), Neutral (none) and Bearish (negative). The first model is based on bag-of-words approach using couple of financial dictionaries to guide the lookup of keywords in the tweet and further derive the score. The second model is based on a probabilistic approach using Naive-Bayes Classifier. We train the model first using training-set data and use the test-set data to predict the score. We provide confusion matrix for both approaches to highlight the accuracy of respective models. The third phase of the project aims to build a regression model by using an aggregate daily sentiment score as a predictor of stock close price. We use GradientDescentOptimizer to train the regression model first and then use the test-data to predict stock close prices. We document model results and accuracy in the form of Tensor board summaries (loss) and computing graph.

**Terminology**

1. **Tweet** - a short message posted by an user on stocktwits platform with a max of 140 chars. See example below.  
   <https://www.stocktwits.com/symbol/AAPL?q=AAPL>  
   
2. **Social Sentiment** **Category** - The tweet (a.k.a message) can be optionally tagged with a category (Bullish or Bearish) indicating the sentiment towards a particular stock. For eg: above example is tagged with a bullish sentiment. A bullish sentiment indicates user is positive about a particular stock’s performance in the near future. A bearish sentiment indicates user is pessimistic about stock’s performance in the near future.
3. **Social Sentiment Score** - Besides a category, one can assign a score for each tweet using several approaches. A score is a numeric value and we normalize it to be within [-1 to +1] for this project purpose. Values in the range [-0.5 to 0.5] indicates a neutral sentiment. A value of 0.5 and above indicates a positive sentiment and likewise, a value of -0.5 and below indicates negative sentiment.
4. **Daily Sentiment Score** - An aggregate sentiment score assigned to a stock on a daily basis by processing the sentiment score of individual tweets for the same day. This is the raw score.
5. **Weighted Daily Sentiment Score** - An aggregate sentiment score for a day calculated using raw sentiment score weighted by number of tweets expressed as a percentage of total tweets for the chosen period (2 months in our case). A bullish or bearish overall sentiment on a given day should be weighted higher or lower depending on #tweets posted for that day when compared to the rest of the days.  
     
   (n is the day index)  
   *TweetMagnitude(n) = (#Tweets(n) \* 100) / Sum(#Tweets for the chosen period)  
   TweetMagnitudeEqual = Assuming equal # tweets for each day, this is the tweet contribution on a given day.*

*WeightedDailySentimentScore(n) = (RawDailySentimentScore(n)) \* (TweetMagnitude(n) / TweetMagnitudeEqual)*In our case, we collected historical tweets for a period of 2 months from Mar 1st to Apr 30th. In that period, there were 42 trading days excluding holidays and weekends. That makes TweetMagnitudeEqual = 2.38%

1. **Stock Close Price** - stock price of a chosen company as indicated by the close price on a given trading day.
2. << **definitions for bag-of-words model** >>
3. << **definitions for naive-bayes model**>>

**Scope and Assumptions**

1. Historical tweets - Collected for 3 stocks (AAPL, FB, TSLA) for a time period of 2 months starting from March 1st to April 30th. This period had 42 trading days excluding holidays and weekends.
2. Each tweet collected from StockTwits.com can have an optional sentiment category attached by user (bearish or bullish). A majority of tweets are classified as none with the assumption that underlying tweets doesn't have any keywords that strongly indicate either bullish or bearish sentiment. In other words, these tweets are classified under 'neutral' sentiment for this project.

**Sources**

1. Investopedia - [Most followed stocks in stocktwits.com](http://www.investopedia.com/articles/markets/021316/most-followed-stocks-stocktwits-nflx-fb.asp)
2. StockTwits API - <https://stocktwits.com/developers/docs>
3. Yahoo! Finance for historical stock prices - [AAPL Historical Prices](https://finance.yahoo.com/quote/AAPL/history?period1=1488344400&period2=1493524800&interval=1d&filter=history&frequency=1d)
4. Bloomberg for Intraday prices - ??
5. Dictionary sources - ??
6. Naive Bayes Classifier - ??

**API and Tools**

1. Python 3.5 on Windows-10 and Linux (Cloudera VM) platform
2. IDE - PyCharm and IPython Notebook
3. Packages - numpy, tensorflow, matplotlib, urllib, json, ……
4. StockTwits RESTful API
5. Tensorboard API
6. Spark
7. …
8. ...

# Data Collection

Stock Tweets (source StockTwits.com)

Show url for chosen stocks

Show sample table

Historical prices (source Yahoo Finance)

Show url for chosen stocks

Show sample table

Intraday prices (source Bloomberg)

Show sample table

Tweets processing - json/restful api

Python code details

Results: table showing #tweets collected for 3 stocks.

# Sentiment Analysis

Basic analysis of tweets.

Describe input data format.

Basic stats for #tweets each day - SD/mean/median.

Show various plots.

#tweets vs close price,

#tweets vs traded volume

# Bag-Of-Words Model

Describe Approach

Describe input data format

Any assumptions?

Show code snippets and logs

Results: Confusion Matrix

Daily Sentiment Score vs Close Price changes

# Naive-Bayes Classifier

Describe approach

Describe input data format

Any assumptions?

Show code snippets and logs

Results: Confusion Matrix

Daily Sentiment Score vs Close Price

# Regression Model

Describe approach

Describe input data format

Any assumptions?

Show code snippets and logs

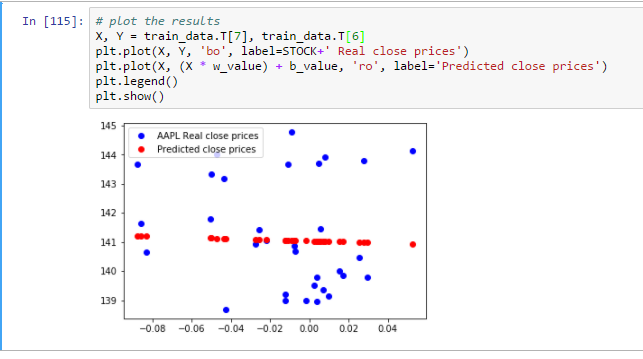
Results: Linear regression plot

Tensorboard summaries and graph

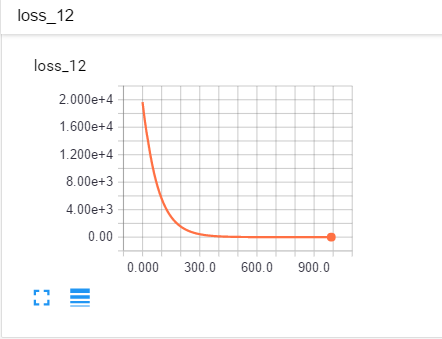
# Real time analysis

# Future work

Regression plots for AAPL (Daily sentiment score vs close price)



Loss function



Tensorboard graph for GradientDescentOptimizer training.

