NATURAL LANGUAGE PROCESSING (SE 316)

STOCK MARKET TRENDS PREDICTION USING STOCK TWEETS ANALYSIS

MADE BY:-

Ayush Gupta
2K15/CO/040

Jatin Thareja
2K15/CO/069

Rishabh Kumar
2K15/CO/104

PROBLEM DESCRIPTION

- Our project aims to explore the impact of factors such as social media tweets, change in volume of tweets and previous prices on the daily price movements of stock market.

- We intend to apply sentiment analysis on tweets collected from StockTwits(a social media platform dedicated to share markets and investments) spanning a period of last 4-5 years to gather meaningful information. Using them, trading volumes and price data over the same period, we predict daily bearish/bullish trends for three major stocks, Google, Apple and Amazon.

Data extraction – I
Tweets data extraction

- StockTwits data was collected and downloaded in raw JSON format totalling over 900,000 messages for three major stocks Amazon(AMZN), Apple(AAPL) and Google(GOOGL). API provided by StockTwits was used to collect tweets related to a specific company.

- The tweets were sorted according to time and stored in 3 separate csv files along with the stock symbol, date and time of tweet, user Id* and Message Id*.

* StockTwits gives a unique User Id to every user and a unique Message Id to every tweet in its database.

Cleaning and pre-processing of tweets data

- Significant pre-processing was necessary to generate "bag-of-words" feature vectors The pre-processing steps included :-

- Aggregation of tweets data according to date
- Removing company tickers, tags and digits
- Removing html entities and links
- Tokenizing the tweets
- Lemmatization of tokens to get root words
- POS Tagging using TreeBank corpus

Data extraction – II
price and volume data

- Daily split-adjusted stock price data was collected via the Intrinio API.
- The data included Opening, Closing and Highest Prices along with trading volumes of last 1000 days counting backwards from 31, March, 2018.
- The data was stored in three separate csv files for Amazon, Apple and Google respectively.

Proposed method - overview

- Our aim, and hence our method, was divided into two parts :-

- **Sentiment Analysis Phase** :- We tried lexicon based approach namely SentiWordNet scoring and supervised learning techniques RNN, DNN and Naïve Bayes classifiers.

- **Prediction Phase** :- We used a multi layer perceptron using 'lbfgs' optimizer to classify trend as bullish or bearish.

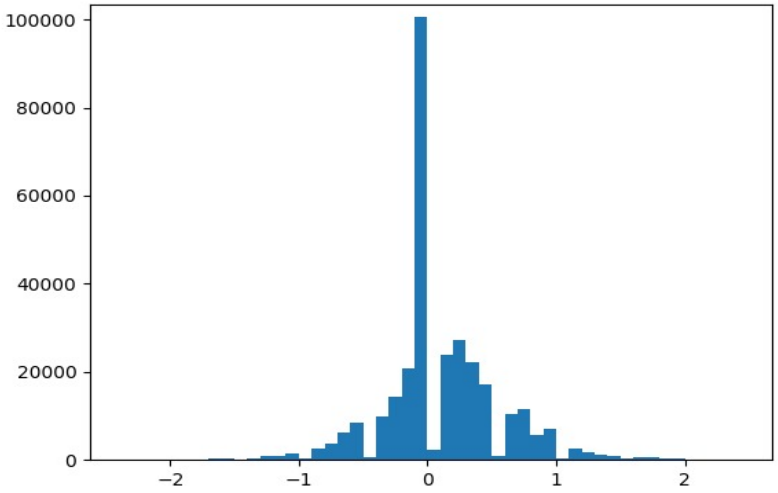
SENTIMENT ANALYSIS METHOD – I
SENTI-WORD-NET

- Each word in WordNet is assigned an objectivity, a negativity and a positivity score, sum of which comes out to be 1.

- Using NLTK, we tokenized, POS tagged, lemmatized the tweets. We used lesk algorithm for word sense disambiguation in case of multiple sense.

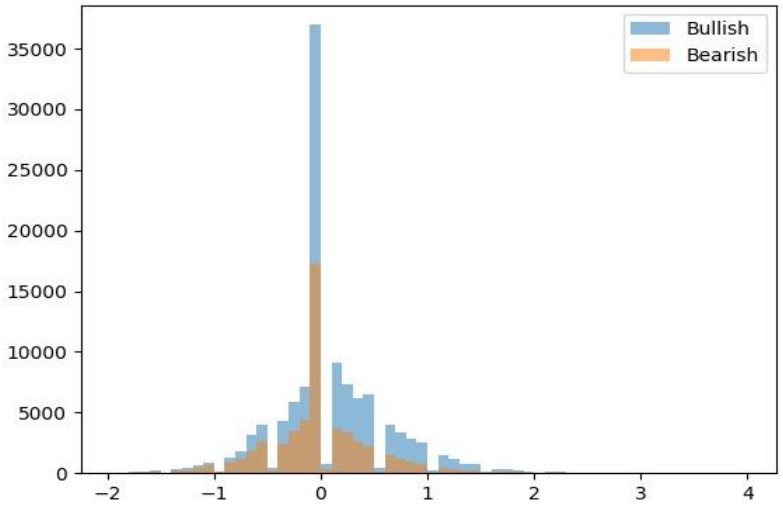
- A list of negation words was gathered from WordStat financial dictionary. Whenever a negation word was encountered, scores were reversed for subsequent words.

SENTI-WORD-NET GRAPHS



This graph above represents SentiWordNet scores for Apple stocks.

This graph below represents SentiWordNet score for labelled data.



SENTIMENT ANALYSIS METHOD – II
RNN/DNN CLASSIFIER

- A deep neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers between the input and output layers.

- To run sentiment analysis on the data we first need to convert words to their respective dense vector representation. We used Word Embedding, Word2Vec and GloVe Embedding for the same.

- Both RNN and DNN were implemented using TensorFlow and gave an accuracy of 70% on our labelled training data.

SENTIMENT ANALYSIS METHOD – III
MULTINOMIAL NAÏVE BAYES

- We created a vocabulary of more than 65000 words from our labelled data and use the frequency of appearance of each word as the feature vector for our classification.

- We used both unigrams and bigrams of words for our classification.

- It was implemented using scikit-learn and gave us an accuracy of 72% on the test data. we combined the lexicon based positive and negative scoring of a word to get accuracy of 73.15%.

Trends prediction phase

- The second stage of our project was to use results of sentiment analysis results and price and volumes data to predict bullish or bearish nature of the stocks.

- We used a Multi-level Perceptron of two hidden layers of size 2, using a ‘lbfgs’ optimizer which is an optimizer in the family of quasi-Newton methods.

- Before classifying we scaled our data to have a unit standard deviation

Trends prediction phase

- The input for the neural network consisted of total number of bullish and bearish messages for a particular stock, the actual previous day trend, change in volume of messages about the stock, and cash flow in the market for that particular stock.

- The output was either a bullish or a bearish label for next day closing price.

- The classifier was implemented using scikit-learn and gave accuracy of 57% on the test data.

Implementation example

- Consider a list of tweets for the stock of Apple.

  ['$APPL is bound to increase!!!!!!',
  '@rK buy buy :) buy buy :) $APPL',
  'I am bearish for *** $APPL ***',
  'Unpredictable, for $APPL see https://www.apple.com']

- Pre-processing phase changes them to :-

  [' is bound to increase',
  'buy buy :) buy buy :) ',
  'I am bearish for ',
  'Unpredictable, for see this ']

Implementation example

- Each sentence is given a positivity and negativity score. Objectivity score is equal to 1 - (positive + negative) score. It is represented by a tuple:-

[(0.125, 0.125), (0.875, 0.0), (0.0, 0.5), (0.0, 0.625)]

- Sentiment of each sentence is calculated using a combination of lexicon and supervised learning approach given by

[‘Bearish’, ‘Bullish’, ‘Bearish’, ‘Bullish’]

Implementation example

- Now the above results, along with other features are fed to MLP classifier. Let us say that the previous day actual sentiment for Apple was Bullish, the previous day tweets volume was 7, and the cash flow for today is 7000000. Thus the feature vector will become

  [Cash Flow = 7000000,

  Previous actual sentiment = 1,

  Calculated bearish sentiment = 2,

  Calculated bullish sentiment = 2,

  Change in tweet volume = -3]

The above vector scaled for unit standard deviation and fed to MLP to get Bullish prediction.

Results and accuracy tweets classification

- Our approach for sentiment analysis using a combination of lexicon based and supervised learning using naive bayes gave an accuracy of 73.15%.

	Precision	Recall	F1-score	Support
Bearish	0.53	0.78	0.63	1559
Bullish	0.89	0.71	0.79	3738
Avg/Total	0.78	0.73	0.74	5297

Results and accuracy trends prediction

- The results for predicting the actual trend of market, using multi layer perceptron model gave an accuracy of 57%. The results are :-

	Precision	Recall	F1-score	Support
Bearish	0.59	0.12	0.20	82
Bullish	0.59	0.94	0.72	109
Avg/Total	0.59	0.59	0.50	191

Shortcomings

- Model has a high recall for Bullish trends whereas quite low recall for Bearish, which is a direct consequence of less data available for bearish sentiment as well as bearish movement of prices.

- We gave equal weights to the tweets of all users, while some users can be more influential over others.

- The lexicon we followed as one of the features of our classification is not optimized for stock related messages.

REFERENCES

1. Predicting Stock Price Movement Using Social Media Analysis, Derek Tsui
2. SENTIWORD NET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, Stefano Bacci
3. Opinion Mining Using SentiWordNet, Julia Kreutzer & Neele Witte
4. Predicting Stock Movement through Executive Tweets, Michael Jermann
5. Stock Market Trends Prediction after Earning Release, Ran An, Chen Qian, Wenjie Zheng