Social Media based Stock Market Analysis using Big-Data Infrastructure

Shashanka Venkatesh	18 5001 145
Venkataraman Nagarajan	18 5001 192
Vishakan Subramanian	18 5001 196

BE CSE, Semester 8

Dr. N. Sujaudeen
Supervisor

Project Review: 1 (31 March 2022)
Department of Computer Science and Engineering
SSN College of Engineering

1 Abstract

There are several factors that influence the value of a stock apart from the typical quantitative and qualitative parameters seen in the fundamental analysis of stocks like balance sheets, income statements, cash flow statements etc. One of the most influential factors in recent years are social media trends.

The aim of this research is to analyze and understand the effect of such articles over the value of a stock at any given time. In our proposed methodology, we consider Twitter, a popularly used micro-blogging site as our social media source. We perform sentiment analysis on collected Tweets, and measure their correlation to stocks of companies within the sector the Tweet appeals to. To analyze such voluminous amounts of data in an efficient & effective manner, we propose to build an architecture that makes use of popular tools for analysing large-scale data at real time, like Apache Spark & Apache Kafka.

2 Proposed Work

Predicting how the stock market moves is a challenging issue due to many factors involved in it, like interest rates, economic growth, current trends, politics, etc.

Investors use various sources of information to determine whether or not to buy/sell stocks of any company. Two such sources that are widely used (especially in recent years) are news events & social media trends. Social media trends particularly seem to be influential in decision-making for the novice stock market investor, and thus its influence on present day stock market cannot be ignored.

Henceforth, we wish to understand and analyze the statistical correlation between social media trends and its effect on the relevant industry's stock price valuations.

To evaluate the correlation, we require a large sample space with periodically collected data points from various trustworthy sources of information. Stock data, current news & social media trends tend to be voluminous with a high degree of velocity. To appropriately wrangle such copious amounts of raw data to extract valuable information out of it requires the usage of a big-data architecture.

Our research works with the data collected from the time period from 30 March 2020 to 20 March 2022 - a reasonably wide sample space to help us obtain a fair perspective of the inherent correlation between social media trends and the stock market. We chose to work with sectors that are also considerably discussed about in online forums viz. Tech, Oil, Electric Vehicles (EVs), Gaming and and Cryptocurrency.

To appropriately relate the tweet trends of these sectors to their respective industries, we consider stocks of a few companies that have a high market share in their sectors. For example, to correlate the tweets made regarding the EV sector, we consider the stock prices of Tesla Motors (NASDAQ: TSLA) and Lucid Group (NASDAQ: LCID) for analysis. This leads to a more concrete understanding of the inherent correlation than compared to the work done by Lee et al.[1] which uses the Dow Jones Industrial Average (INDEXDJX) for their correlation analysis.

3 Architectural Design

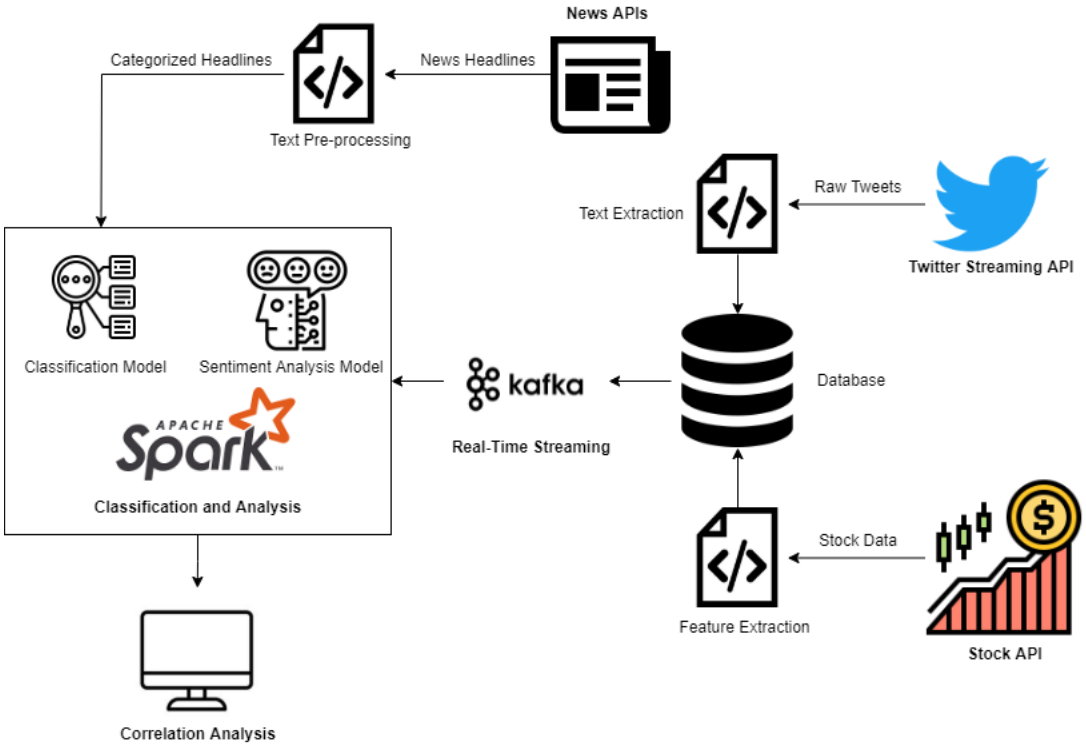


Figure 1: Proposed Architecture

3.1 Twitter Data Pre-Processing

The Twitter Streaming API allows to query tweets with certain keywords or hashtags in them. The collected tweets are processed to get the raw text in them, and are then stored in a database. But this method is prone to having false positives, i.e., tweets containing the keyword we're looking for, but are not actually relevant to the sector itself.

In order to remove the false positives from the tweets collected using keywords, news headlines data is used to train a machine learning model to classify text phrases onto their relevant sectors. For example, a Tweet: *"Fly anakin has made its way to my apple music. Gotta go on the discovery now"* was identified as a tweet regarding the tech industry, since it includes the name of one of the giants - Apple. But it is clearly not a tweet that has anything to do with the company itself.

The next step is to analyze the sentiment of the tweet. This can then be used to see if there is any correlation between tweets of about a certain sector, and the stock market changes of the companies that come under that sector.

An NLP model BERTweet (Nguyen et al. [12]), can be used to analyze the sentiment of the tweets. BERTweet, having the same architecture as BERT-base (Devlin et al. [10]), is trained using the RoBERTa pre-training procedure (Liu et al.[11]). Experiments show that BERTweet outperforms strong baselines RoBERTa-base and XLM-R-base (Conneau et al., 2020), producing better performance results than the previous state-of-the-art models on three Tweet NLP tasks: Part-of-speech tagging, Named-entity recognition and text classification.

The output of BERTweet is a 3-tuple of confidence values in the format of [negative_sentiment, neutral_sentiment, positive_sentiment]. This output is then converted to a single integer based on whatever sentiment the model has maximum confidence in (-1: negative; 0: neutral; 1: positive).

The category classifier and the sentiment analyzer can be set-up on the Spark Lambda Architecture. The Tweets can then be fed to Spark using Kafka, from the database, in order to simulate real-time analysis.

3.2 The BERT NLP Model

BERT is a multi-layered encoder. In Devlin et al.'s [10] paper, two models were introduced, BERT base and BERT large. The BERT large has double the layers compared to the base model. By layers, we indicate transformer blocks. The BERT encoder expects a sequence of tokens. The below image shows how tokens are processed and converted. [CLS] is a special token inserted at the beginning of the first sentence. [SEP] is inserted at the end of each sentence. We created segment embeddings by adding a segment 'A' or 'B' to distinguish between the sentences. We also add the position of each token in the sequence to get position embeddings.
Our classification model and sentiment analysis model makes use of the BERT base architecture, considering significant overheads associated with employing BERT large, which utilizes more computing resources.

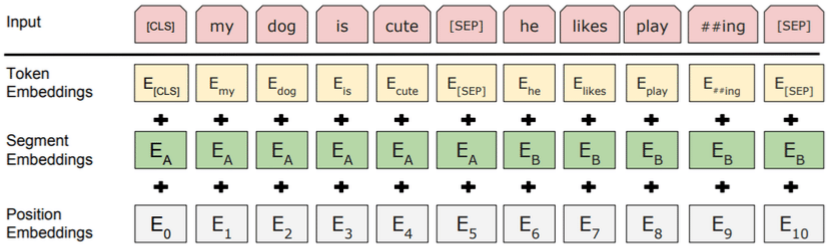


Figure 2: BERT Embeddings

The sum of the above three embeddings is the final input to the BERT Encoder. BERT takes an input sequence, and it keeps traveling up the stack. At each block, it is first passed through a Self Attention layer and then to a feed-forward neural network.

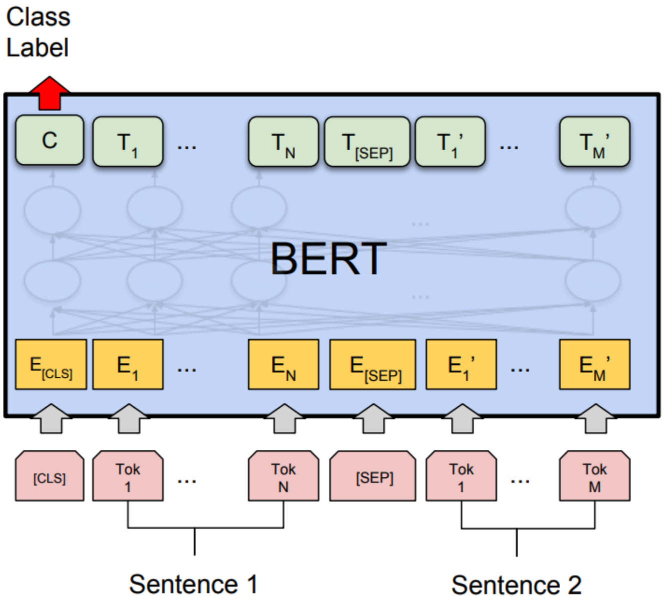


Figure 3: The BERT Sequence Classifier

3.3 The Classification Model

The classification model, which is used to strip false-positive tweets, is built using the BERT Sequence Classifier.

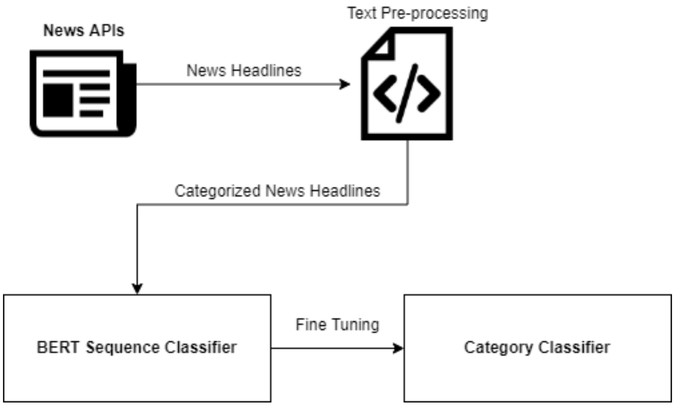
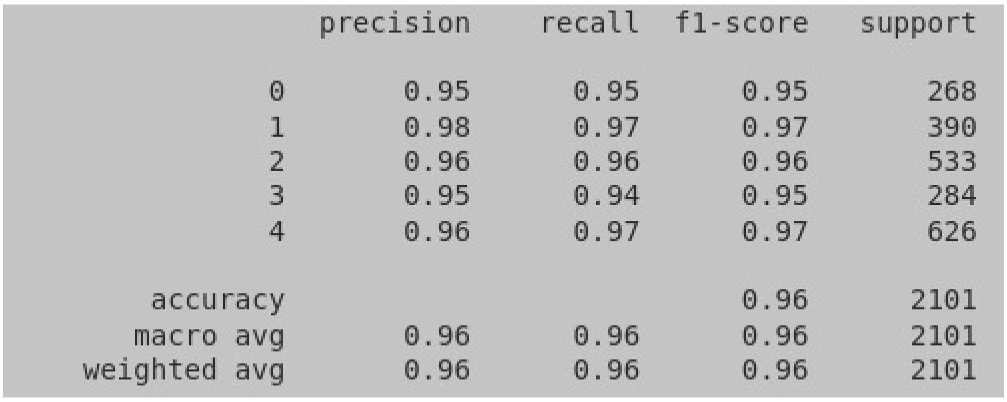


Figure 4: Fine tuning the BERT model using News Data

The model obtained a 96% accuracy on the test data of approximately 2000 samples with the following Confusion Matrix (The range 0 - 4 represents the encoded values for the 5 different sectors):



```
              precision    recall  f1-score   support

           0       0.95      0.95      0.95       268
           1       0.98      0.97      0.97       390
           2       0.96      0.96      0.96       533
           3       0.95      0.94      0.95       284
           4       0.96      0.97      0.97       626

    accuracy                           0.96      2101
   macro avg       0.96      0.96      0.96      2101
weighted avg       0.96      0.96      0.96      2101
```

Figure 5: Classification Report

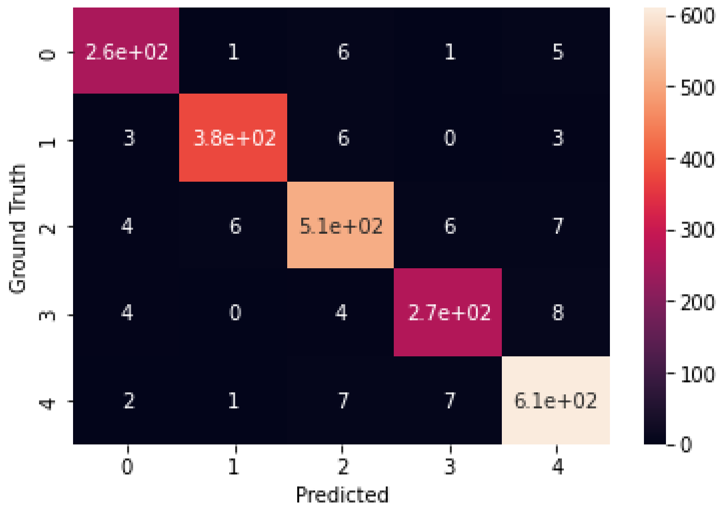
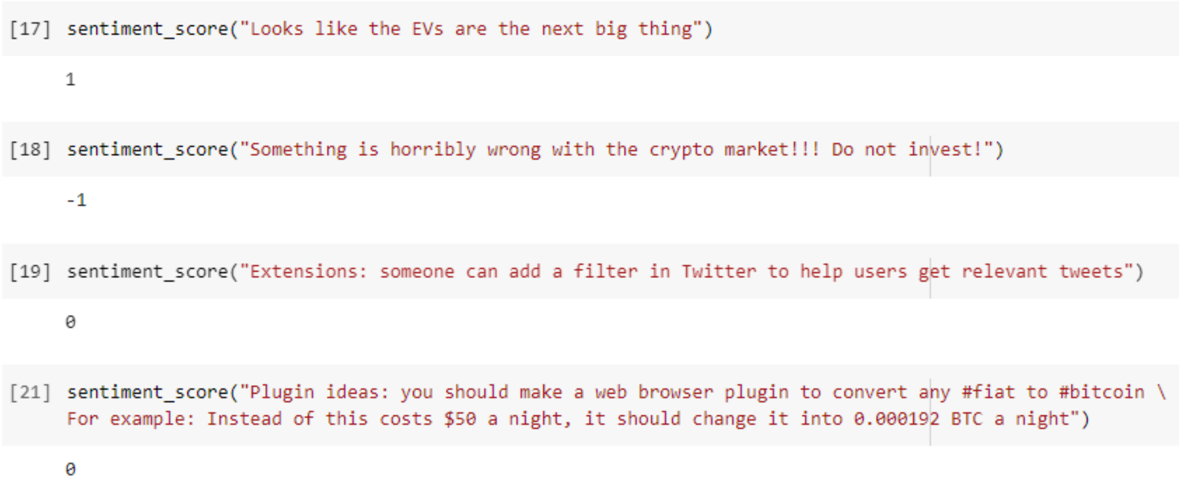


Figure 6: Fine tuning the BERT model using News Data

3.4 Sentiment Analysis

According to Nyugen et al.[12], the BERTweet model presented by them performs much better than existing models like RoBERTa and XLM-R which were tested on the SemEval2017-Task4A dataset. It has an accuracy of 72% and an F1-score of 72.5%.



```
[17] sentiment_score("Looks like the EVs are the next big thing")

     1

[18] sentiment_score("Something is horribly wrong with the crypto market!!! Do not invest!")

     -1

[19] sentiment_score("Extensions: someone can add a filter in Twitter to help users get relevant tweets")

     0

[21] sentiment_score("Plugin ideas: you should make a web browser plugin to convert any #fiat to #bitcoin \
     For example: Instead of this costs $50 a night, it should change it into 0.000192 BTC a night")

     0
```

Figure 7: Sentiment analysis done on sample tweet data

3.5 Stock Data Analysis

Raw stock data is collected from public stock APIs, and is then pre-processed in a predefined format, to be stored in the database. Using Apache Kafka, the stock data can then be streamed to the Spark Architecture, where the data trend-analyzed current mood towards that sector's stock.

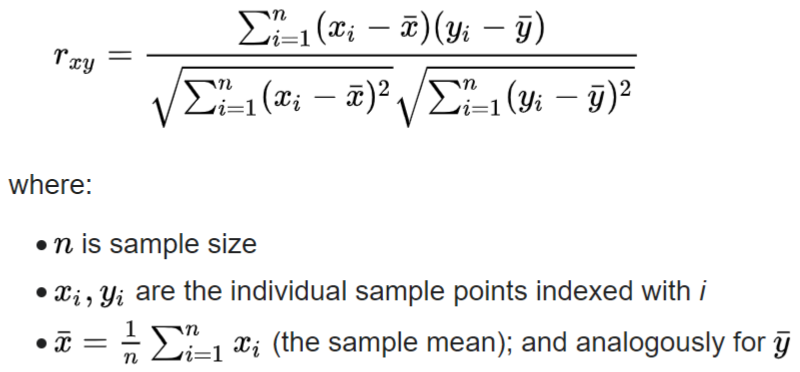
This trend analysis is then compared with the trends & public moods seen from the Twitter sentiment analysis, for further statistical studies.

3.6 Statistical Correlation Analysis

The main objective of our research is to identify whether there exists any correlation between Twitter data & stock market data in real-time, and if so, quantify the correlation between them in different sectors.

Correlation analysis needs to be done sector-wise, since social media might have a varying degree of influence over each sector - a logical example would be the cryptocurrency market, which is highly volatile with respect to social media sentiments.

The Pearson correlation coefficient is a measure of linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations. Thus it is essentially a normalized measurement of the covariance, such that the result always has a value between 1 and 1.



$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where:

- n is sample size
- x_i, y_i are the individual sample points indexed with *i*
- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean); and analogously for $\bar{y}$

Figure 8: The Pearson Coefficient for a sample

4 Techniques Used

4.1 News Data Collection and Pre-processing

We used 3 APIs to collect News Data:

1. **NewsData**: `GET https://newsdata.io/api/1`
2. **NewsCatcher**: `GET https://api.newscatcherapi.com/v2/search`
3. **New York Times**: `GET https://api.nytimes.com/svc/search/v2/articlesearch.json`

Three Python scripts were used to collect data from each of the APIs. A congregating script was used to call the other scripts and collect the combined data in a CSV file. The Headlines were labelled based on the the category they were extracted as.

4.2 Tweet Data collection and Pre-Processing

The Twitter API (`GET https://api.twitter.com/2/tweets/search/recent`) along with keywords for each category of analysis was used to collect tweets.

A Python script was written to collect the data, and generate a dataset of tweets labelled according to their category.

4.3 Stock Data collection and Pre-Processing

The Polygon API (`GET https://api.polygon.io/v1/open-close` and `GET https://api.polygon.io/v1/open-close/crypto`) was used to collect stock ticker data. The API accepted the type of ticker data required, and the date to get the data from. A Python script was used to collect historic Stock data for the past two years, and was stored in a dataset categorized according to the sector of the companies queried.

4.4 Storage of Tweet and Stock data in database

In order to facilitate real-time inflow of data, Apache Kafka needs to be used to stream the collected data to Apache Spark. Kafka needs the data to be stored on a database, either SQL or NoSQL. Since the data collected has a very well-defined schema, we have chosen the SQLite database to store the collected data. The Python scripts have generated two datasets in the CSV format.

A Python script utilizing the Pandas library was used to populate the datasets into two relational tables within the SQLite database.

5 Expected Outcomes

The main purpose of the entire analysis, is to unearth a correlation between Tweets and Stock market changes, if any. The expected outcome, therefore, is an accurate correlation analysis between tweet sentiments, and the stock market variations in the related sectors.

6 Further work

1. **Building a Kafka pipeline:** The data from the SQLite Database must be streamed using Apache Kafka in order to be consumed by Apache Spark.

2. **Configuring Apache Spark:** The Spark Lambda system needs to be configured to run both the models, and also perform the correlation analysis.

3. **Conducting the analysis:** The correlation analysis performed by Spark must be conducted and presented using appropriate visualization tools, in order to draw conclusions.

References

[1] Chungho Lee, Incheon Paik, *Stock Market Analysis from Twitter and News Based on Streaming Big Data Infrastructure*

[2] Vaanchitha Kalyanaraman, Sarah Kazi, Rohan Tondulkar, Sangeeta Oswal, *Sentiment Analysis on News Articles for Stocks*

[3] Zhihao Peng, *Stocks Analysis and Prediction Using Big Data Analytics*, 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)

[4] Gurcan, Fatih & Berigel, Muhammet, *Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges*, 2018 2^{nd} International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)

[5] Aparna Nayak, M. M. Manohara Pai and Radhika M. Pa, *Prediction Models for Indian Stock Market*

[6] Dharmaraja Selvamuthu, Vineet Kumar and Abhishek Mishra, *Indian stock market prediction using artificial neural networks on tick data*

[7] Kanavos, Andreas & Vonitsanos, Gerasimos & Mohasseb, Alaa & Mylonas, Phivos. (2020). *An Entropy-based Evaluation for Sentiment Analysis of Stock Market Prices using Twitter Data*

[8] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, *Efficient Estimation of Word Representations in Vector Space*

[9] Marsland, Stephen, *Machine Learning: An Algorithmic Perspective*, 2014

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*

[12] Dat Quoc Nguyen, Thanh Vu, Anh Tuan Nguyen, *BERTweet: A pre-trained language model for English Tweets*