

Short-term Stock Price Prediction Using Sentiment Analysis and Machine Learning



Requirements Analysis

COMP208 Group 4:

Killian McShane, James Harris, Will Payne, Matthew Blatherwick, James Murphy

1. Project Description

a. Target Audience

Our targeted audience is comprised of users who are considering trading, those who are active participants in the stock market and financially engaged individuals who could potentially become interested in trading.

b. Mission Statement

This project aims to produce a web application that enables users to view and interact with short-term stock price predictions by web-scraping Twitter using their developer API and Python's 'Natural Language Tool Kit' library to collect sentiment scores (a decimal score from 0 to 1, indicating consumer opinion from tweets) for given stock ticker symbols. Predictions can then be made by feeding historical stock data from Yahoo Finance into a recurrent neural network (such as an LSTM) to form a basic price prediction using Tensor Flow. A final, more accurate prediction can then be made using both consumer opinion (from sentiment scores) and our recurrent neural network's ability to predict.

c. Mission Objectives

Essential

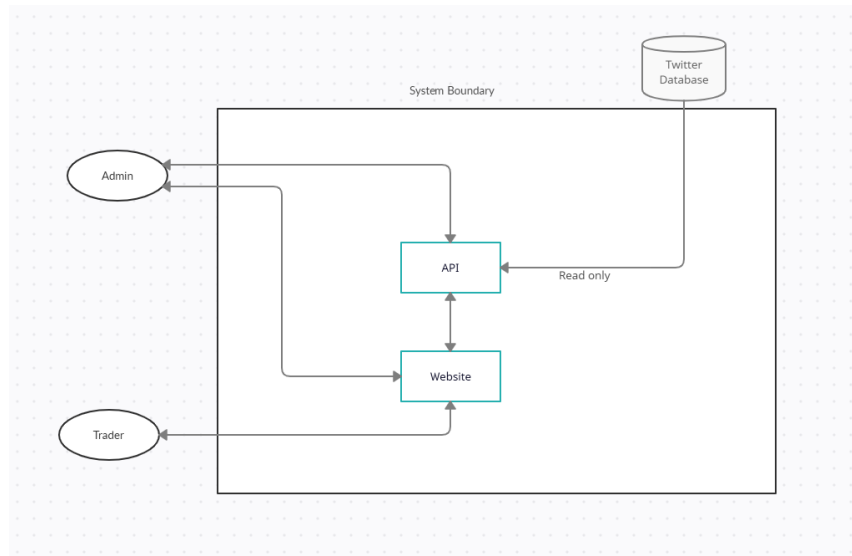
- A stock price prediction using a recurrent neural network (such as an LSTM) to form a basic level to gauge our sentiment data on.
- To collect sentiment data from Twitter's developer API using Python's NLTK library, combined with Yahoo Finance historical stock data to produce a more accurate prediction.
- A web application to graphically view these predictions.

Desirable

- Allow users to select their own choice of stock to form a prediction.
- Graphs to display stock movement.



d. System Boundary Diagram



e. User Views

The system will have two main user views. However as stated in the Target audience, our app isn't just targeted at one specific type of user. Our app is targeted at people who are both interested in getting involved into the world of trading, those who are also familiar with trading and even those who are actively engaged into stock market. This could mean we could have multiple User Views based on the Users familiarity with the stock market. A question could be asked once starting the app to try and determine the user's confidence with this field.

However, the main User views will be our core users (traders), and an administrator view for app optimisation. These are the functions each user type will need:

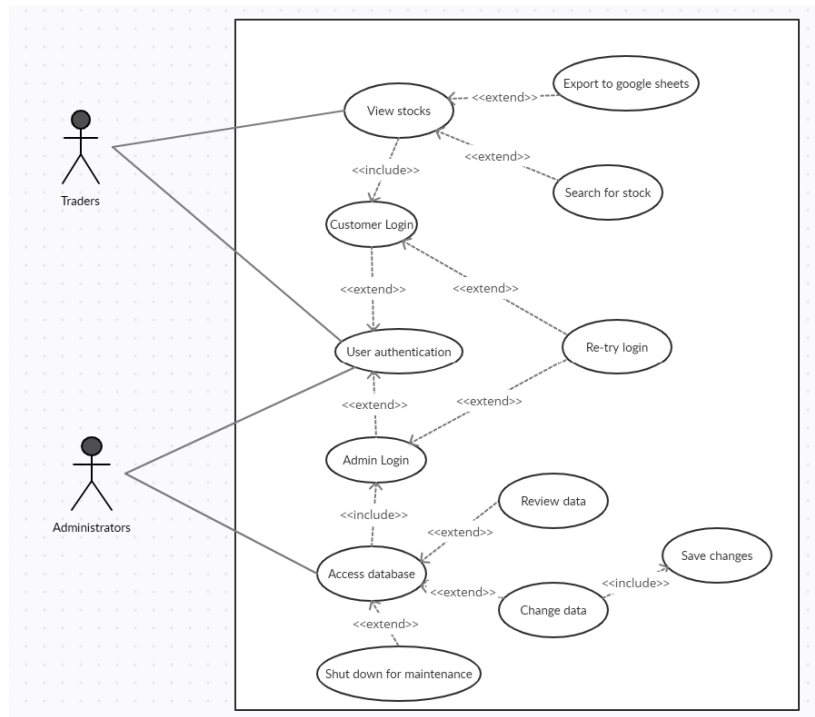
Traders:

- Search engine (if the user wants to access certain stocks).
- User profile (to log into their account).
- A view of the stock's current value.
- A view of stock prediction.
- A graph view to display stock movement.
- Be able to access the Twitter API.
- Be able to save and store the predictions to google sheets.

Administrator:

- Access to backend tools (such as access to the server-side language).
- Be able to find data.
- Be able to save and change data.
- Be able to temporarily shut down the app for reasons such as maintenance and display a message conveying this.

Use Case Diagram:



f. Transaction Requirements

Trader:

- User needs to be able to create a sheet of projections.
- User needs to be able to save their projection sheets Be able to delete sheets.
- Projections also need to be able to be updated.

Administrator:

- Admin needs to be able to perform updates to the web app.
- Display messages if the app has any new errors to warn the user about.
- Have the ability to shut down the app temporarily and reboot it.
- View data on the number of users that are using the server.

g. Functional & Non-functional Requirements

- The system must allow users to view the predictions in a web application.
- The system must be able to access the Twitter API through a developer account.
- The system's back-end script for web scraping and the machine learning algorithm must be written in Python 3.
- The system must save and store (Google Sheets) predictions so that they can be accessed by the web application.
- Old predictions must not be overwritten by new predictions in Google Sheets.
- The system's web app must always request and show the latest stock predictions.
- Users of the system's web app should be able to select stocks from a pre-determined list of stocks on the web app.
- All stocks shown on the web app must have a corresponding stock price prediction.
- The system's script for returning stock price predictions must be run once every 24 hours.
- To load the web app the user must be connected to the internet.
- To run the script the admin must be connected to the internet.
- The system must allow new stocks to be added to the web app and to the script that returns stock price predictions.
- The system's web application must be responsive and work across different size devices.
- Initially the system must be capable of supporting 100 users at a time, this could then be scaled as and when required.

2. Conduct of Project and Plan

a. Background Research

Dr. Xiaowei Huang and Dr. Vitaliy Kurlin's modules in advanced artificial intelligence and data science (respectively) have provided us with the experience necessary to form the basis for this project. Our previous experience in web-scraping has allowed us to collect the sentiment (consumer opinion) data from Twitter user's tweets, and our skill with machine learning affords us the ability to create meaningful results from this raw data.

Some research will still need to be conducted in order to determine the optimal recurrent neural network architecture for this problem, however the LSTM is our current choice.

b. Data Required

We require up-to-date sentiment data from Twitter's users so that our predictions include consumer opinion and remain accurate. Financial data will also be taken from Yahoo Finance in order to train the machine learning model.

c. Design Stage

The project will be a web-based application using html for the front end, and python for the back-end coding. The back-end coding will handle the machine learning and perform calculations on the data collected that has been stored in a Google sheets file. This will then be passed on to the web API to be presented to the user.

The webpage will present the user with key information about current stocks in a ranking system, and the user can request more specific information about a particular stock which will present a graph with key information surrounding.

A user register and log in system will be implemented for both Traders and Admins to create accounts. The web API will communicate with a DBMS where user information can be saved securely.

d. Implementation Stage

During the design phase we will be using a set of test data as a placeholder. Once the application has been created, we will then implement the real data from twitter to test the application. The machine learning section of the system can start and will gain experience to improve results over time.

e. Gantt Chart

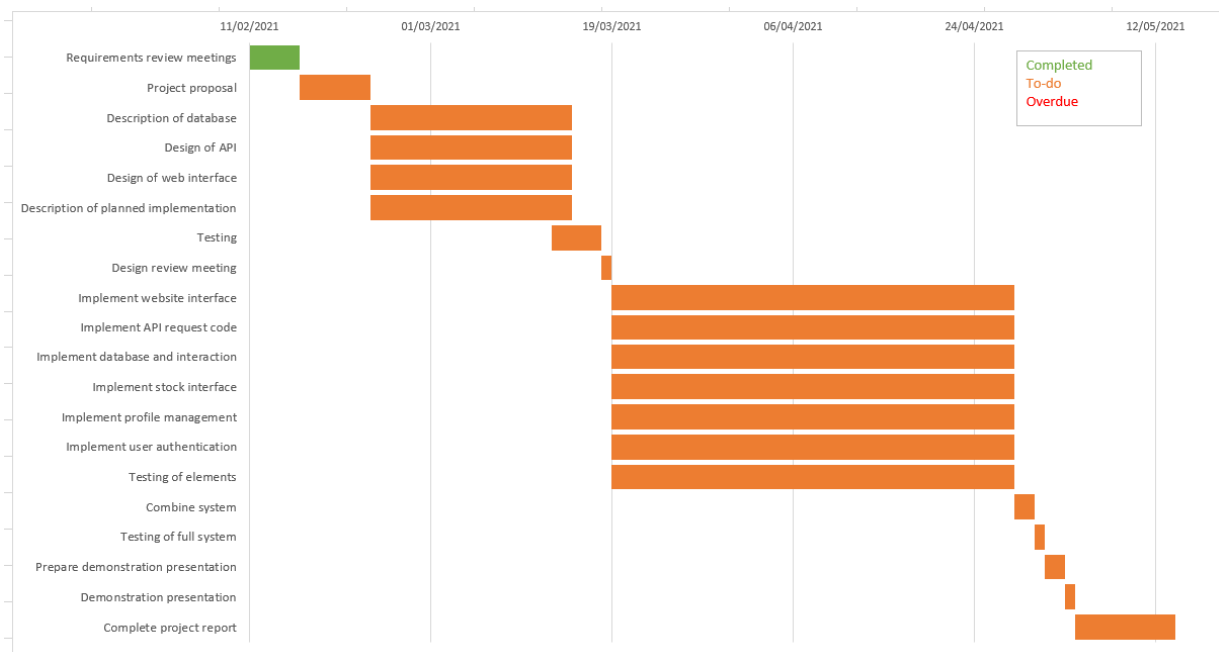


Chart to be regularly updated.

f. Hardware Requirements

- Each member of the project team will need to use their own or a University laptop/mobile device for the implementation and testing phase. The laptop/PC will need to be capable of running the software required for implementation.
- Each member of the project team will need access to the internet via WI-FI/cellular data.

g. Software Requirements

- Python 3 and IDE
 - o Natural Language Tool Kit Library
 - o TensorFlow
 - o NumPy
 - o Pandas
- Java and IDE
- Google Sheets
- Hosting platform for the web application

h. Risk Assessment

The major challenges in carrying out the project:

- The team needs to obtain a Twitter developer account to be able to use the Twitter API. This is crucial for the sentiment analysis.
- The team needs to create a working and effective machine learning algorithm for predictions. This is crucial to the project as it will be a key factor in the effectiveness and usefulness of the entire project.
- The team needs to source useful, clean data to use for the machine learning algorithm. The datasets should provide at least a 5-year history of desired stock.
- Google Sheets (or another cloud-based data processor) must provide adequate integration for scripting and web development. This is crucial to keeping the web app up to date with the latest stock price predictions.
- The web app created in Java must be responsive to users on different devices.

What new skills are required and how these will be acquired:

- Members of the project team will be using the Twitter API to web scrape for the first time. There is plenty of supporting documentation found on the Twitter API support page.
- Members of the project team will be using Google Sheets to store data and integrate it with code for the first time. There is supporting documentation found on the Sheets API page.

3. Bibliography

Python Machine Learning (Second Edition) by Sebastian Raschka & Vahid Mirjalili

<https://www.tensorflow.org>

<https://docs.tweepy.org/en/latest>

<https://python-twitter.readthedocs.io/en/latest>

<https://finance.yahoo.com>

<https://developer.twitter.com/en/support/twitter-api>

<https://developers.google.com/sheets/api>

<https://docs.oracle.com/javase/7/docs/technotes/guides/language>

<https://developer.mozilla.org/en-US/docs/Web/HTML>