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

Design Stage

COMP208 Group 4:
Killian McShane, James Harris, Will Payne, Matthew Blatherwick, James Murphy

1. Summary of Proposal

a. Background, Aims & Objectives

The proposed stock price prediction system has four key components, each with their own unique challenges. This report aims to quell any confusion regarding the structure of the system and provide a framework for the project's implementation.

The following points outline our major system objectives.

i. Sentiment Analysis

- To create web-scraping algorithm using Twitter's Developer API allowing us to collect sentiment scores of Tweets relating to 5 pre-selected stocks.

ii. Machine Learning

- To create a machine learning model, trained on historical Yahoo Finance data and sentiment scores to output stock price predictions.

iii. Database

- To create a database on Google Sheets, using Google's 'Sheets API' to upload daily stock price predictions to, and allow the web application to pull data from.

iv. Web Application

- To create a web application displaying each of our pre-selected stocks' current price, along with our prediction for its closing price and a simple graph.

The following points elaborate on some initial system requirements.

Archive Old Predictions: This is to allow for historic graphing of the price predictions. The prediction line-graph should (hopefully) closely match the actual stock prices. Both the prediction, and real price lines should be graphically displayed on the same chart for an easy, user-friendly accuracy comparison and evaluation.

Responsive Web Design: Users should be able to view the predictions and corresponding graphs on any device they own without display errors. We want our design to respond to the user's behaviour based on screen size, platform and orientation. To achieve this, we will opt for a mobile first approach. We will handle the resizing of the website while designing in HTML and CSS to make sure that it looks appealing on all devices.

Graphing: We want the graphs to display data using contrasting colours so that users can easily distinguish between actual prices and predicted prices.

Disclaimer: A disclaimer on the web application's front page should notify users that we are not providing financial advice and that any losses made on their part are their own. We are not liable for any damages/costs incurred.

Desirable Features:

- A login system for users which could potentially be altered to enable a payment plan in the future depending on the success of the project.

b. Changes to Original Proposal

The system is ambitious. It therefore goes without saying that some changes had to be made for the success of the overall project. It was decided that users should not be allowed to select their own stocks to form a price prediction. As the system's userbase scales into the hundreds, the resources required to produce the prediction would be far too demanding for a project of this scale. There is a hard-coded limit on the number of requests one can make to the Twitter API every 15 minutes.

As you can see from the table, our application can produce 1800 tweet-based sentiment scores every hour (450 lots of 4). Initially, to train the machine learning model, each of the 5 stocks will require 1825 sentiment scores (relating to 5 years' worth of daily stock sentiment).

Response formats	JSON
Requires authentication?	Yes
Rate limited?	Yes
Requests / 15-min window (user auth)	180
Requests / 15-min window (app auth)	450

Once the model is trained however, we will only need 1 sentiment score for each stock, each day.

While this seems reasonable, when scaled up to hundreds of users, each requesting new stocks, our system would simply break down. It was therefore concluded that we would clearly need to limit the number of stocks.

c. Summary of Research & Analysis

Research was made into Natural Language Processing and the APIs available to us. We found that the process used a lot of fundamental machine learning knowledge that we already had. However, steps during the pre-processing stage of data were novel to us. Tokenization is a pre-processing step which splits longer strings of text into smaller word pieces, or tokens. It can also be referred to as text segmentation or lexical analysis. These tokens are then converted to numerical data through a process call binarization, so that they can be fed into the machine learning model for training.

We needed to research how data could be obtained from Yahoo Finance and how that data is structured. This was fairly a simple exercise and achieved by exploring the API and through trial and error. We needed to understand the different methods/algorithms that can be used for stock price prediction and how to evaluate them. This information has come from a mix of blog posts, research papers and taught modules at the University, noted at the end of the document.

2. Design

a. Preface

Our system is in some ways an empirical investigation into the effect of sentiment data on the predictive ability of a machine learning algorithm. Therefore, *we hypothesise that by training our model on this additional information, its predictive ability should improve.*

b. Anticipated Components

- **Sentiment Collection**

  Sentiment collection forms the foundation for the system. These scores will be the key attribute for our investigation of their improvements in the predictive analysis of stock prices.

  Using Twitter's Developer API, we will be able to collect 5 years-worth of daily sentiment scores for each of the 5 stocks every hour or so. This is due to the hard-coded limitations on requests to their website every 15 minutes.

- **Data Pre-processing**

  Historical stock price data from Yahoo Finance will be combined with sentiment scores taken from Twitter in a Panda's data-frame structure. The data will need to be pre-processed in order for the machine learning model to efficiently learn and predict. Pre-processing is done by a series of the Tokenizations, Binarizations that were discussed in the section above and finally appended to the daily adjusted stock closing prices.
  Once this final data-frame is formed for each stock, we should then be able to feed it into the machine learning model.

- **Machine Learning**

  Predicting stock price predictions depends on various factors such as the global economy, company financial reports and performance as well as the current political landscape. Typically, there are two approaches that can be used for predicting stock prices for organisations, technical analysis and fundamental analysis.

We will be focusing on technical analysis which is a method that uses historical prices of stocks to predict future prices. A machine learning algorithm uses large quantities of data to find patterns. It learns with experience and can be fine-tuned to suit the given data. There are multiple algorithms which are applicable to stock price prediction and these will be explored further in this document.

- **Graphing**

  In comparison, graphing is a relatively trivial component of the system however, these images will form the most appealing component of the web application.
  Several graphing libraries exist. Seaborn is our current choice for its improved visualisations over MatPlotLib.

- **Google Sheets Upload & Download Function**

  We decided to use Google Sheets as it provides a free easy to use API which allows us to upload and download with ease in both Python and Java.
  Data uploading will be done using Python, as this is the language the data will be produced in.
  Data downloads will be done using Java, as this is the language that the web application will be coded in

- **Web design**

  To design the website, we will need to use a few different languages. HTML will be used to provide the content displayed to the user, while CSS will give structure and style to that content. We will be using JavaScript to give the website interactive functionality and allow the user to interact with the interface. Lastly, we will need the website to be able to interact with the DBMS. We will use PHP to communicate between the two.

- **DBMS**

  We will be using a MySQL database to store user information. The DBMS will be hosted on an Apache HTTP Server.

- **User accounts**

  We want to give users the ability to have their own account when using our website. This will allow each user to choose

what they want to see when on the website. A favourites feature will be implemented so that users can keep track of the stocks they are interested in the most.

- **Creating stock graphs**

  To create graphs displayed on the website we will be using a JavaScript library called Chart.js. Values for the graph will be taken from google sheets. These graphs will give the most information to the user at just a glance and will give the website more visual appeal. We do not want the website to be cluttered with text, so a more clean and simple presentation of data will be more effective and less intimidating.

- **Ranking system**

  Our website will have a ranking system to display the top stocks at the current time. The user will also be able to change the time frame over which the rankings are decided, such as seeing the best stocks daily, weekly or monthly. This should give the user a quick look to see which stocks are doing well.

c. Data Descriptions

Initially the application will show five stock predictions with the view to expanding on this figure. For each stock prediction we will require a dataset of historical prices from Yahoo finance.
The structure of this data is shown in *Figure 1*.

Each column indicates the following:

- *Date*: The date of the stock prices.
- *Open*: The opening price of the stock on the given date.
- *High:* The maximum price of the stock on the given date.
- *Low:* The minimum price of the stock on the given date.
- *Close:* The close price of the stock on the given date.
- *Adj. close:* The close price of the stock adjusted to reflect corporate actions.
- *Volume:* The total amount traded on the given data

Date	Open	High	Low	Close*	Adj. close**	Volume
10 Mar 2021	121.69	122.17	119.79	120.00	120.00	34,085,526
09 Mar 2021	119.03	122.06	118.79	121.09	121.09	129,159,600
08 Mar 2021	120.93	121.00	116.21	116.36	116.36	153,918,600

Figure 1

The dataset will be split into training (60%), validation (20%) and test (20%) sets. Each partition will correspond to a block of time across the 5 years. For example, the training set will account for oldest data with the test set corresponding the latest 20% of data.

This data will be held in a Panda's data-frame along with an appended column for our Twitter sentiment scores for each day spanning the last 5 years.

d. Algorithms, Languages & Libraries

Core Machine Learning Algorithm

We aim to predict the daily adjusted closing prices of a number of stocks using data from the previous five years. There are a number of different machine learning algorithms that can be used to predict stock prices. Instead of picking one single algorithm we will be testing multiple prior to releasing the final version of the application. The different algorithms we will be training our data with are:

- *Last Value*: taking the previous day close price and using it as the stock price prediction.
- *Moving Average*: the predicted value will be the mean of the previous n values. This n value is arbitrary and is a parameter than can be changed throughout the training process.
- *Linear Regression*: to find a line of best fit, or the regression line.
- *LSTM*: Long short-term memory. Used to predict stocks an arbitrary number of steps into the future. LSTM has five essential components which allows it to model both long-term and short-term data. A diagram of the LSTM model is shown in Figure 2.

All computation not directly involved with the construction of the web application will be done in Python 3. The web application itself will be completed in Java.

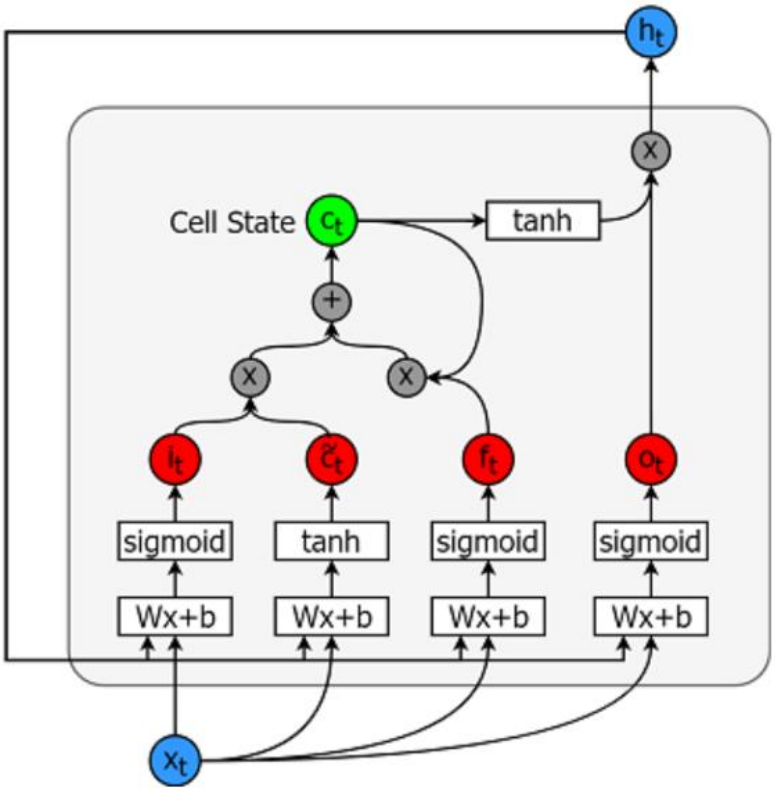


Figure 2
LSTM Architecture (our current favourite).

Current Library Selection:
Numpy, Pandas, TensorFlow, Keras, Seaborn, Google Sheets API, Twitter Developer API, NLTK Library, SKLearn (for evaluation metrics).

e. Interface Design

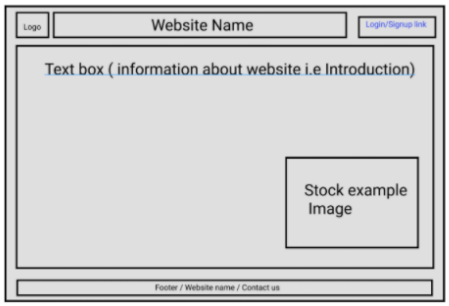
This is a storyboard of 6 of the pages the user will see. Starting with the welcome page which will mainly display information on what the site is about and what it can do. It will act as an introduction to the website, and also include buttons to enable the user to sign into an existing account or create a new one. The second page is the login page where the user will enter their account details. It also includes a button to sign up to the website for new users. It will also include buttons such as forgot password for users who have lost their login details.

The third page is for new users who need to sign up, it also includes a questionnaire on what type of user they are. The 4th page is a home page which will display some trending stocks the user may be interested in. This will be determined from the information retrieved from twitter. The bookmark page stores stocks that users have specifically saved or

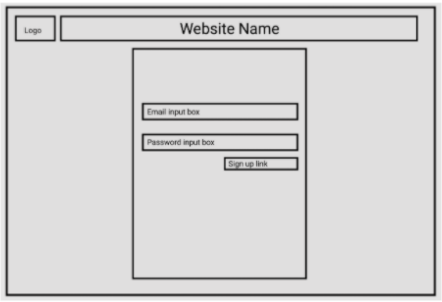
“bookmarked”. The final page is an example of what a single stock page will show. It also has a button to add or remove the stock to the previous bookmark tab.

The home and bookmark page also include a search bar for the user to navigate the site and search a specific stock that they have in mind which will then take them to the stock page. You will also notice that for most of the pages there is a contact us section at the bottom for users to contact a help team.

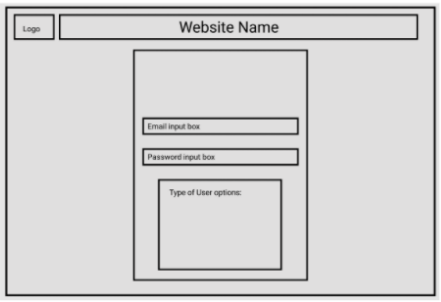
Welcome page



Log in / sign up page

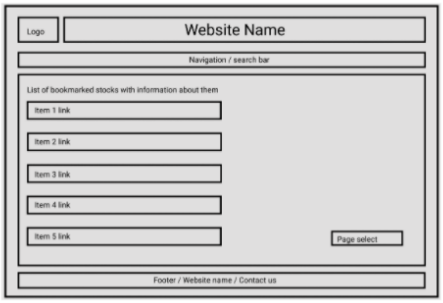


New user sign up page

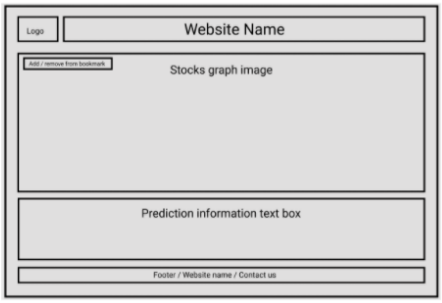


Home page
(navigation bar and trending stocks)

Bookmark page



Stock page



Here is a rough mock-up of the general layout of one of the stock pages. The actual website will include a navigation bar to navigate between pages. However, the main layout of the stock pages will hopefully be similar to this, you can see that the main information on the stock is shown on the left, with a graph to accompany this information to its left. The information below will be information on the company so the user can gain a better understanding about what type of company the page is displaying.

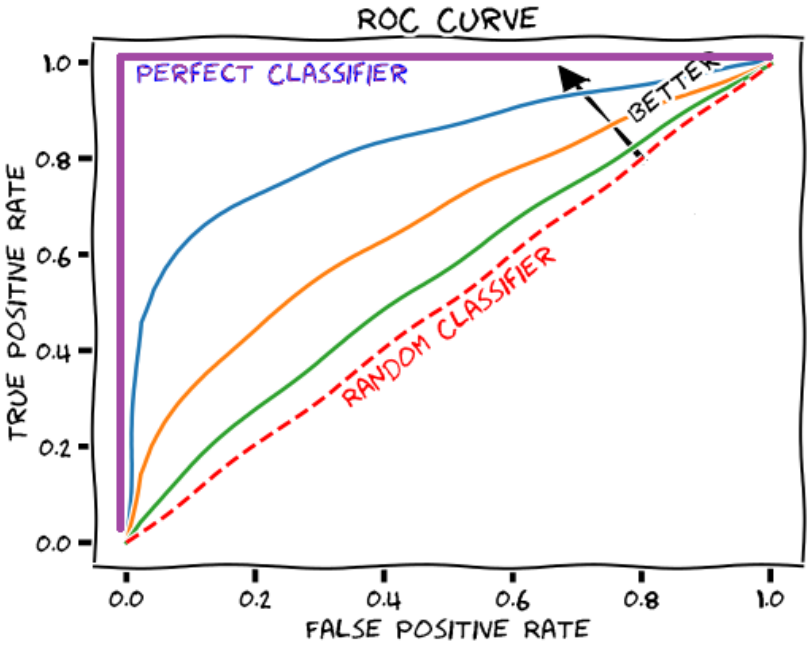
f. Evaluation of Results

To evaluate the effectiveness of the methods we can use the RMSE method, the root mean squared error. The lower the value, the better our prediction. We can then compare our predictions against one another to find the best algorithm for stock price prediction.

The SKLearn library contains a swathe of useful evaluation metrics. Our current favourites include:

- **ROC Curves**

  A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
  It enables us to produce line graphs, which allow us to visually evaluate the predictive ability of several models against one another.



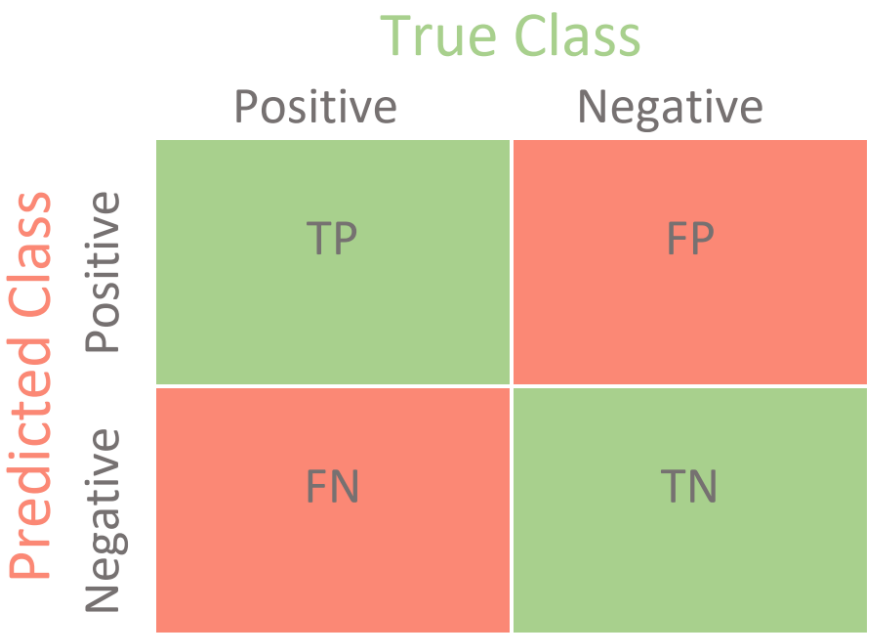
- ***k*-Fold Cross Validation**

  Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called *k* that refers to the number of groups that a given data sample is to be split into. Of the *k* subsamples, a single subsample is retained as the validation data for testing the model,

and the remaining $k - 1$ subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data.
This offers us a holistic evaluation of the model's predictive ability over unseen data.

- **Confusion Matrices.**

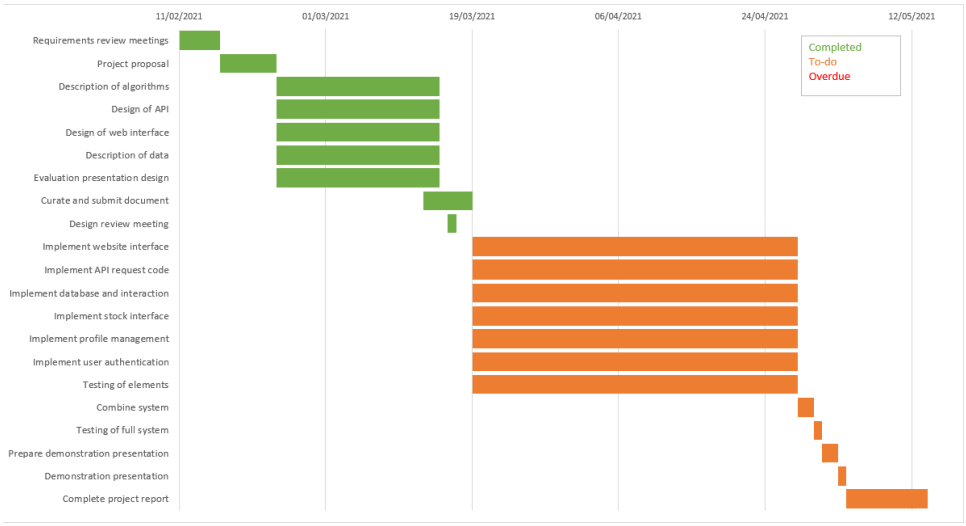
  Confusion Matrices allow us to visually evaluate how well our model classifies its data and how badly it has mis-classified.
  A key advantage is that it enables us to pin point which class of data the model is struggling with.



g. Anticipated Conclusion

In conclusion, we anticipate that the model's predictive ability will improve with the addition of sentiment data from Twitter proving our hypothesis.

3. Review of Plan



So far, we are happy with our productivity. We have been holding at least one meeting per week and believe that these have been helpful in assigning roles during the design process and ensuring that this part of the project was done in time. We are confident that if this continues, we will be able to produce our project in the given time frame.

4. References

COMP 226: Trading and Financial Markets with *Prof. Rahul Savani*,
Machine Learning techniques applied to stock price prediction,
Stock Closing Price Prediction using Machine Learning Techniques – Research Paper.
https://developers.google.com/sheets/api/quickstart/python