MIS41110 – Programming for Analytics

Stock Symbols – Group Project

Icon

Description automatically generated

Group 4

|  |  |  |
| --- | --- | --- |
| **Name** | **Student ID** | **Project Contribution** |
| Odhrán McDonnell | 17335221 | 25% |
| Tristan Fluechter | 21201568 | 25% |
| Kunal Gupta | 21200525 | 25% |
| Anirudh Bhatia | 21201428 | 25% |

**UCD Michael Smurfit Graduate Business School**

**Darren Redmond & Dr Miguel Nicolau**

Table of Contents

[Section 1: User Manual 3](#_Toc89091282)

[1. Introduction 3](#_Toc89091283)

[1.1 - Project Code Source 3](#_Toc89091284)

[1.2 - Glossary of Terms 3](#_Toc89091285)

[2. Program Modules 3](#_Toc89091286)

[3. Operating Instructions 4](#_Toc89091287)

[3.1 Streamlit 4](#_Toc89091288)

[3.2 Command Line 4](#_Toc89091289)

[4. Explicit Error Handling 4](#_Toc89091290)

[Section 2: Upper Level UML Activity Diagram 6](#_Toc89091291)

[Section 3: References 8](#_Toc89091292)

[Section 4: Appendix 9](#_Toc89091293)

[4.1 Program Modules 9](#_Toc89091294)

[4.2 Required Packages 10](#_Toc89091295)

# Section 1: User Manual

## Introduction

This user manual provides all the required information for the user to successfully run our stock information and prediction program. The manual outlines the procedure for using the program from both a command line interface and a graphical user interface (GUI) hosted on www.Streamlit.io. The purpose of our program is to provide the user with descriptive statistics, future price predictions and associated investment recommendations for a chosen company over a user-selected time series.

### 1.1 - Project Code Source

The sources for the codes for both the Streamlit and the command line programs are provided below:

Streamlit: <https://github.com/tristanfluechter/streamlit-fa>

Command Line: <https://github.com/tristanfluechter/commandline-fa>

### 1.2 - Glossary of Terms

Following are the list of acronyms used in the code and the manual:

* MA = Moving Average
* GUI = Graphical User Interface
* RMSE = Root Mean Squared Error
* MACD = Moving Average Convergence Divergence
* LSTM = Long Short-Term Memory

## Program Modules

The program is split into several modules, all of which perform different aspects of the program. For detailed information please refer the *Appendix* section at the end.

* **Financial\_data.py** - This module scrapes financial websites to gather and display useful financial data such as analyst predictions, market sentiments, and critical financial KPIs.
* **Google\_news.py** - This module scrapes Google News for headlines about the user’s selected stock.
* **Descriptive\_stats.py** - The module uses a Pandas dataframe to conduct descriptive analytics on the user’s chosen stock in the chosen time frame. The descriptive statistical measures are about the stock closing price (mean, standard deviation, quartiles), a linear trend line, a simple MA plot, a weighted MA plot, and a MACD plot.
* **Linear\_regression.py -** This module performs a simple linear regression on the stock data selected by the user. The user is prompted for a target future prediction date and their desired number of past days to consider for the regression model.
* **LSTM\_prediction.py -** This module uses the dataframe of the user’s selected stock to train a Long Short Term Memory neural network model to make 15-day future stock price predictions.
* **Facebook\_prophet.py -** Based on the previously inputted stock data timeframe, the model returns a price prediction, a related trend line, and relevant lower and upper confidence bounds.
* **Sentiment\_analysis.py -** This module provides a sentiment analysis for the user’s selected stock based on Google News headlines. A recommendation to either buy or sell the stock based on the sentiment is then provided.
* **Stock\_evaluator\_GUI.py -** This module is used to run the entire program. The above modules are all imported and a Stock Prediction class is created. The module runs the streamlit GUI on the local host.
* **Stock\_evaluator\_CL.py -** This module is used to run the entire program. The above modules are all imported and a Stock Prediction class is created. The module runs the command line menu on the terminal.

## Operating Instructions

For both the Streamlit and Command Line UI’s, a complete list of required packages for the program to successfully run is provided in ‘requirements.txt’.

* Download the source code from section 1.1 of the user manual. Place all the files and modules in a single folder.
* Check if python is installed on the system. Open terminal and type “Python3”. If the terminal does not return the version information, please install python through the following link: [Install Python](https://docs.python.org/3/using/mac.html)
* Install requirements/packages mentioned in the *Appendix* section. Please use the following link to learn about downloading the required libraries: [Install packages](https://packaging.python.org/tutorials/installing-packages/)
  + OSX: Installation of Facebook Prophet module (prophet) should work seamlessly
  + Windows: Installation of Facebook Prophet module might not work with pip install. Use conda environment instead (follow [these](https://stackoverflow.com/questions/49889404/fbprophet-installation-error-failed-building-wheel-for-fbprophet) steps).
* Set the terminal present working directory to the folder where code has been saved.
* To run the GUI or command line version of the program, type the following from terminal (make sure you use the environment to which you installed all modules!):
* **Streamlit**: streamlit run stock\_evaluator\_gui.py
* **Command Line**: python stock\_evaluator\_cl.py

### 3.1 Streamlit

Streamlit generates a web application on the local host where the user needs to enter the required stock and date information. The homepage displays the basic information about using the application.

The user has the option to choose from a drop down menu, that will navigate through different functionalities like – basic information, descriptive statistics, advanced analytical charts, and predictive models. The user can run the application on terminal or directly on the Streamlit cloud through the following link: <https://share.streamlit.io/tristanfluechter/streamlit-fa/main/stock_evaluator.py>

### 3.2 Command Line

The program generates a menu for the user to make different selections. The user needs to input the stock ticker and the required date information. After the inputs, a menu appears asking the user to make a selection. The main menu has the following options:

1. Plot Stock Price
2. Show News Headlines
3. Show Analyst Predictions
4. Show Descriptive Statistics Menu
5. Show Predictive Statistics Menu
6. Pick New Stock and Timeframe
7. Exit App

The user can see sub menus for descriptive and predictive statistics and has the option to jump back to the main menu or query another stock.

## Explicit Error Handling

**Stock Ticker input**

* The user should verify the letter representation of the stock ticker that they wish to analyse on Yahoo Finance before inputting it.

**Start & end analysis date**

* Dates must inputted in the format YYYY-MM-DD.
* The target date must be a date in the future.
* The difference between the end date and the current date must be negative.

**Day numbers for ‘x’ day MA’s**

* The desired moving average number must be inputted as a positive integer value and cannot exceed the dataframe length.

**Weights for weighted MA’s**

* The inputted weights must be entered as a positive float value between 0 and 1.
* The sum of all the weights must be equal to exactly 1.

**Timeframe for MACD Analysis**

* The timeframe (number of past days to be considered) must be at least 27 days.

**Number of past days for Linear Regression**

* The number of past days to be used for the regression model must be expressed as a positive integer. It must also not exceed the amount of available stock data.

**Target date for Linear Regression**

* The target date must be expressed in the form YYYY-MM-DD. The date must be a date in the future.

**LSTM train-test split**

* The LSTM train-test split must be expressed as a positive float between 0.6 and 0.8.

# Section 2: Upper Level UML Activity Diagram

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

# Section 3: References

Alpharithms, n.d. *Calculating the MACD in Python for Algorithmic Trading.* [Online]   
Available at: https://www.alpharithms.com/calculate-macd-python-272222/

Kaggle, 2018. *FbProphet and Plotly Example.* [Online]   
Available at: https://www.kaggle.com/ahmetax/fbprophet-and-plotly-example

Patil, R., 2021. *Stock Price Movement Based On News Headline.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2021/05/stock-price-movement-based-on-news-headline/

Patil, R., 2021. *Stock Sentiment Analysis using NLP.* [Online]   
Available at: https://github.com/ronylpatil/Stock-Sentiment-Analysis

Towards Data Science, 2019. *Time Series Forecasting with Recurrent Neural Networks.* [Online]   
Available at: https://towardsdatascience.com/time-series-forecasting-with-recurrent-neural-networks-74674e289816

Section 4: Appendix

## 4.1 Program Modules

Detailed information about the modules used in the program, both command line and Streamlit.

##### Financial\_data.py

This module scrapes financial websites to gather and display useful financial data such as analyst predictions, market sentiments, and critical financial KPIs. ‘BeautifulSoup’ is imported as a HTML parser, and the module regex is deployed to search for strings on each site. Data is scraped from CNN Money and MarketWatch.

##### Google\_news.py

This module scrapes Google News for headlines about the user’s selected stock. The headlines are outputted in the descriptive statistics section and are also used to formulate the stock sentiment analysis results.

##### Descriptive\_stats.py

This module uses a Pandas dataframe to conduct descriptive analytics on the user’s chosen stock in the chosen time frame. The principal outputs are basic descriptive statistical measures about the stock closing price (mean, standard deviation, quartiles), a linear trend line, a simple MA plot, a weighted MA plot, and a MA convergence/divergence plot. The simple MA is computed with two day numbers chosen by the user. The user can select six weights for the weighted MA plot.

##### Linear\_regression.py

This module performs a simple linear regression on the stock data selected by the user. The user is prompted for a target future prediction date and their desired number of past days to consider for the regression model. The resulting model is displayed in a plot. An interpretation of the reliability of the regression model results based on its RMSE value is provided once the model has been returned. An RMSE value of <= 0.4 is considered too low for any weight to be placed on the result of the model, while RMSE values of > 0.4 indicate that the model output can be relied on to inform stock investment decisions.

##### LSTM\_prediction.py

This module uses the dataframe of the user’s selected stock to train a Long Short Term Memory neural network model to make 15-day future stock price predictions. The user is prompted for a train-test split of the data. It is necessary to segregate the data into training and testing data to avoid overfitting of the model. Once the model has been trained, a 15-day stock price prediction for the chosen stock is provided along with the RMSE of the model.

##### Facebook\_prophet.py

This module provides the user with another predictive analytic tool for their chosen stock. Based on the previously inputted stock data timeframe, the model returns a price prediction, a related trend line, and relevant lower and upper confidence bounds. The model gives a long-term prediction of the desired stocks.

##### Sentiment\_analysis.py

This module provides a sentiment analysis for the user’s selected stock based on Google News headlines. A random forest classifier is used to classify a dataset of headlines into a binary prediction of whether the selected stock’s value will increase (1) or decrease (0) on the following day. An associated recommendation to either buy or sell the stock based on the sentiment is then provided.

##### Stock\_evaluator.py

This module is used to run the entire program. The above modules are all imported and a Stock Prediction class is created.

## 4.2 Required Packages

Full list of required packages along with the version:

* beautifulsoup4==4.10.0
* gnews==0.2.0
* keras==2.6.0
* matplotlib==3.3.4
* numpy==1.19.5
* pandas==1.2.4
* pandas\_datareader==0.10.0
* pandas\_ta==0.3.14b0
* plotly==5.3.1
* prophet==1.0.1
* regex==2021.4.4
* requests==2.25.1
* scikit\_learn==1.0.1
* statsmodels==0.12.2
* streamlit==0.88.0
* yahoo\_fin==0.8.9.1
* tensorflow