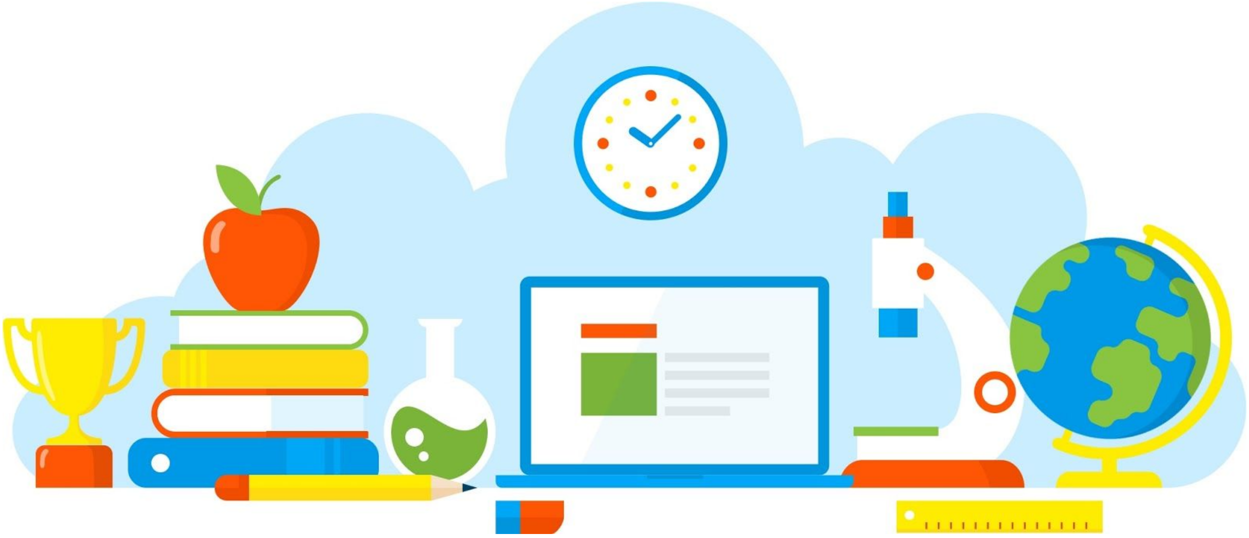


# ECE 568 Final Report

# RU Free Stock

Bowei Feng, Cong Deng
Chaoji Zuo, Xinyi Yang
05/09/2019

# 1 CONTRIBUTION BREAKDOWN

We all contribute equally in this project, and our project goes well under the strength of our efforts. Each one contributes equally and we have finished a great project. The work everyone did are listed as below:

Front-end: Chaoji Zuo

Back-end: Bowei Feng

Prediction Engine: Cong Deng

Testing & Document: Xinyi Yang

# 2 GLOSSARY OF TERMS

| | |
|---|---|
| High/Low | Highest and lowest prices of a stock that has been recorded during a period. |
| Average Volume | The amount of stocks that has been traded over duration of time. |
| Buy and Hold | The price of buying a good for a long term rather than trying to turn a  profit quickly. |
| Opening Price | The price of the first trade of any listed stock. |
| Closing Price | The price at the end of a trading day. |

| | |
|---|---|
| Dividend | A dividend is a payment made by a corporation to its shareholders, usually as a distribution of profits. When a corporation earns a profit or surplus, the corporation is able to reinvest the profit in the business (called retained earnings) and pay a proportion of the profit as a dividend to shareholders. |
| Earning per Share(EPS) | The portion of a company's profit allocated to each outstanding share of common stock. Earnings per share serves as an indicator of a company's profitability. It is calculated as (Net Income -Dividends on Preferred Stock) / divided by the Average Outstanding Shares |
| Forecast | A prediction of the future based on special knowledge. |
| Fundamental Trading | Fundamentalists trade companies based on fundamental analysis, which examines things like corporate events such as actual or anticipated earnings reports, stock splits, reorganizations or acquisitions. |
| Individual Investor | An investor who purchases relatively small lots of stocks for his or her own portfolio. |
| Institutional Investor | An entity with large amounts to invest, such as investment companies, brokerages, and investment banks. Institutional investors are covered by fewer protective regulations because it is assumed that they are more knowledgeable and better able to protect themselves. Institutional investors are usually a group of people, rather than individuals. |
| Market Trend | The tendency of a financial market to move in a particular direction over time. Bull market refers to an upward trend, and a bear market refers to a downward trend. |

| | |
|---|---|
| Momentum | The notion that an asset’s price is likely to move in one direction instead of changing directions |
| Support Vector Regression (SVR) | Support Vector Regression (SVR) uses the same principles as the SVM for classification. But its output is a real number not a class which is different from SVM. |
| Stock Symbol | A unique set of symbols that represent a particular company. Ex: GOOG is the stock symbol of Google. |
| Trading Day | The duration of time the stock market is open for buying and selling stock. Ex: For the New York Stock Exchange trading day is 9:30 AM Eastern Time to 4:00 PM Eastern Time, trading days never occur on weekends. |
| Technical Indicator | In technical analysis, a technical indicator is a mathematical calculation based on historic price, volume, or (in the case of futures contracts) open interest information that aims to forecast financial market direction |
| Moving Average Prediction Model | A way to predict the future price of stocks based on the assumption of constant underlying mean of given prices |
| Exponential Moving Average (EMA) | An exponential moving average (EMA), also known as an exponentially weighted moving average (EWMA), is a first-order infinite impulse response filter that applies weighting factors which decrease exponentially. The weighting for each older datum decreases exponentially, never reaching zero. |

# 3 SYSTEM REQUIREMENTS ( Chaoji Zuo)

## 3.1 ENUMERATED FUNCTIONAL REQUIREMENTS

| Identifier | Priority | Requirement |
|---|---|---|
| Data_1 | 5 | The server system would collect the historical data when user first search it. |
| Data_2 | 5 | The server system would collect the real time data when user first search it. |
| Data_3 | 4 | The server system would update the real time data when user search it again. |
| Data_4 | 4 | The server system would update the historical data when user search it again. |
| Data_5 | 5 | The system would store the data it collects. |
| Show_1 | 5 | The system would print the stock historical and real time prices with timeline as a graph. |
| Show_2 | 4 | The system would allow user to search a special stock. |
| Show_3 | 4 | The system would show the prediction value of stocks. |
| Show_4 | 3 | The system would have a index page, show the login information. |
| Prediction_1 | 4 | The system would predict the stock value base on the stock data. |
| Prediction_2 | 3 | The system would calculate the indicators of stock. |

| | | |
|---|---|---|
| Trade | 2 | The system allow user to do the virtual trades. |

### 3.2 ON-SCREEN APPEARANCE REQUIREMENTS

| **Identifier** | **Priority** | **Requirement** |
|---|---|---|
| Banner | 3 | A fixed banner on the main page shows our project name. |
| Footer | 3 | A fixed footer on the pages show our relatives information. |
| Flex | 4 | The page must adapt to a various size in order to maintain functionality, consistent design, and mobile devices. |
| Autofill | 3 | The search form can imagine the stock symbol the user would input and do the suggestion. |
| Buy_Sell | 2 | User can buy or sell a stock on the page, they can input the amount of trade too. |

## 4 FUNCTIONAL REQUIREMENTS SPECIFICATION

### 4.1 SPECIFYING THE DISCOVERED REQUIREMENTS:

Architectural style: Web-based.

The programs will run on a Web Browser and each user will have a virtual trade information. The virtual trade information includes their trade records and balance information. These information will be stored in our Mongo

Database for each user. There is no form of client communication, but if we feel as if it could be an added benefit to the website, it can be implemented in the future.

## 4.2 STAKEHOLDERS

Two Stakeholders can be identified:

1. User: each user could log in the system, get web services and do virtual trade.
2. Administrator: he will maintain and update website services.

## 4.3 ACTORS AND GOALS

Our system has both human and non-human actors.

1. User: The user will take the form of an investor that registers and uses the website.
2. Administrator: The manager that is in charge or keeping the system updated and in working order.
3. Prediction Algorithm: The algorithm(s) that will calculate the prediction.
4. Database: The database will hold all the user data and information as well as all the stock information. All the user information and their trade information will be stored in the database.
5. Yahoo! Finance API: The API is where we will pool all historical stock data from to store into the database to generate the prediction.
6. Alpha vantage API: The API is where we will pool all real-time stock data from to store into the database to generate the prediction.
7. Plotly: Will be used to graphically plot the stock data.

## 4.4 USE CASE

| Use Case | User priority | Function |
|---|---|---|
| **Case 1** | Administrator | Manage FREE STOCK system |
| **Case 2** | User | Stock search assistance |
| **Case 3** | User | Stock and price search |
| **Case 4** | User | Plot special graph |
| **Case 5** | User | Stock Indicator |
| **Case 6** | User | Price prediction |
| **Case 7** | User | Virtual trade |
| **Case 8** | User | Period switch |

Case 1: Manage FREE STOCK system

As the administrators, they need to analysis whether the website is work normally. And they also need to manage and maintain the web serving the customers continually. As an important part of the whole system, the administrator can create the database, update the data in the database (send a request to Google or Yahoo for the data and read them to import them to the database), and manage the users' information.

Case 2: Stock search assistance

Sometimes the customer can not remember the stock code very clear, and our assistance system could help user find their stock code quickly, when the user just type one or two characters of their stock code, our system will show all possible stock code below the search input for user to find the stock code they want. This assistance not only help user to find the stock code quickly and also make sure the code for searching is correct.

Case 3: Stock and price search

The customers can get a lot of information of these stocks, such as the real- time price, the historical price, the recent price graphs (days, months, one years, two years) and so on. This provides user with information about the company and the last year's all quarter results and the current years' quarter results.

Case 4: Plot special graph

As a presentation of history price and current price, the user can notice the flow and trend of the stocks in the graph. The user can click "Price Data" then "historical price" or "real time" to access the result page, then choose a stock to check the specific graph of the stock. The page will call a function to get the data from the database then show them on the screen as a chart.

Case 5: Stock indicator

After analyzing the history data and prediction, the server should return back a guiding suggestion to the user. Based on the prediction result, the server can analyze the user's performance and market situation to get some guiding result and help the user to do the decision.

Case 6: Price prediction

For the prediction of each stock, users do not care how you predict the prices, the only thing they care about is the accuracy of your prediction. Hence, it is necessary to provide the data accuracy for each stock's prediction to enable users know the whether they should trust a specific prediction.

Case 7: Virtual trade

For the virtual trade, each user will have a start balance. And based on the stock real-time data, the user can do the buy or sell operation for the real-time stock price, and the record will be stored in our database for this user, and the user could search their trade records and balance information. Based on the virtual trade, the user could be better in the real trade.

Case 8: Period switch

For historical and real-time stock data, we will show different information in two pages. The user could switch the period in our main page, and for the historical stock data, we will show the stock indicator and price prediction, and for the real-time stock data, we will have a virtual trade for the user.

# 5 USER INTERFACE SPECIFICATION

## 5.1 PRELIMINARY DESIGN

| **Actor Name** | **Description** | **Complexity** | **Weight** |
|---|---|---|---|
| **Plot Graph** | Plot the change of the stock price with the time change. | Complex | 5 |
| **Prediction Algorithm** | Provide prediction indicators and strategy. | Complex | 3 |
| **Database** | Records of stock information, user data, and system data. | complex | 3 |

| | | | |
|---|---|---|---|
| **User** | Login in and input the password. | average | 2 |
| **API** | Link the front-end and the back-end. Support several function in RESTful style. | average | 5 |
| **Trade** | Record the money. | average | 2 |

## 5.2 USER EFFORT ESTIMATION

| **Use Case** | **Function** | **Complexity** | **Weight** |
|---|---|---|---|
| Case 1 | Manage FREE STOCK system | Complex | 6 |
| Case 2 | Stock search assistance | Complex | 8 |
| Case 3 | Records of stock information, user data, and system data. | complex | 8 |
| Case 4 | Stock and price search | average | 5 |
| Case 5 | Plot special graph | average | 5 |
| Case 6 | Stock Indicator | simple | 4 |
| Case 7 | Price prediction | complex | 9 |
| Case 8 | Virtual trade | average | 3 |

## 5.3 EFFORT ESTIMATION USING USE CASE POINTS

UAW=5+3+3+2+5+2=20

UUCW=6+8+5+5+5+4+9+3+3=48

UUCP=UAW+UUCW=20+48=68

# 6 DOMAIN ANALYSIS

## 6.1 CONCEPT DEFINITIONS

In order to build the domain model, we will need to once again analyze the primary use cases. From there we will be able to derive the important concepts of the system. We will first look at the boundary concepts that directly interact with the actors and then afterwards analyze and find the internal concepts of the system.

Website:

- Definition: A hypertext document connected to the World Wide Web.
- Responsibilities:
    - Display HTML document that shows the actor the current context.
    - Shows what actions can be taken through buttons.

Query:

- Definition: Search query.
- Responsibilities:
    - Hold a specific search query

Indicator:

- Definition: Generate stock indicator.

- Responsibilities:
    - Use different indicators to analysis the stock
    - Draw the indicator lines on the chart.

Predictor:

- Definition: Generate stock predictions.
- Responsibilities:
    - Use different algorithms to predict the stock
    - Plot the predictions in the table of the webpage.

Update Historical:

- Definition: Update historical stock data.
- Responsibilities:
    - Check weather the stock data is in the database
    - Download the latest stock data from the web source

Update Real-time:

- Definition: Update real-time stock data.
- Responsibilities:
    - Check weather the stock data is in the database
    - Download the latest stock data from the web source
    - Renew the real-time stock data every one minute

## 6.2 ATTRIBUTES

Website holds attributes related to display text, graph and related API to social media.

Query holds attributes related to search query for sector, industry and keyword. In our system, we can use keyword.

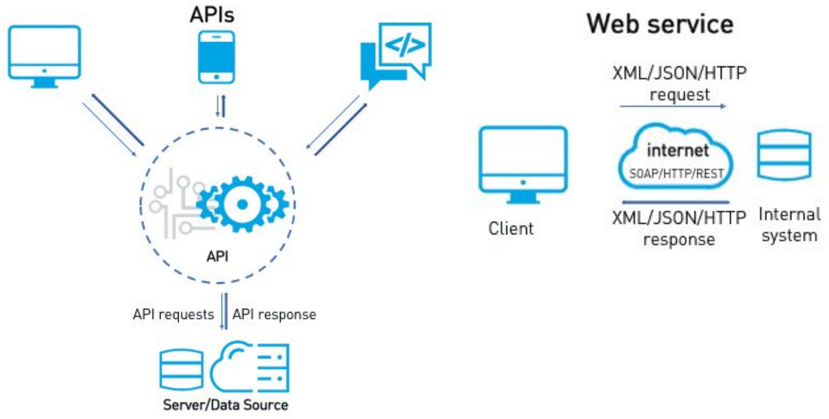
Database: Connection holds attributes to connect to the local database including address, port number, password and users' name.

Timer holds attributes related to when to update current stock prices as well as the time when to predict future stock prices.

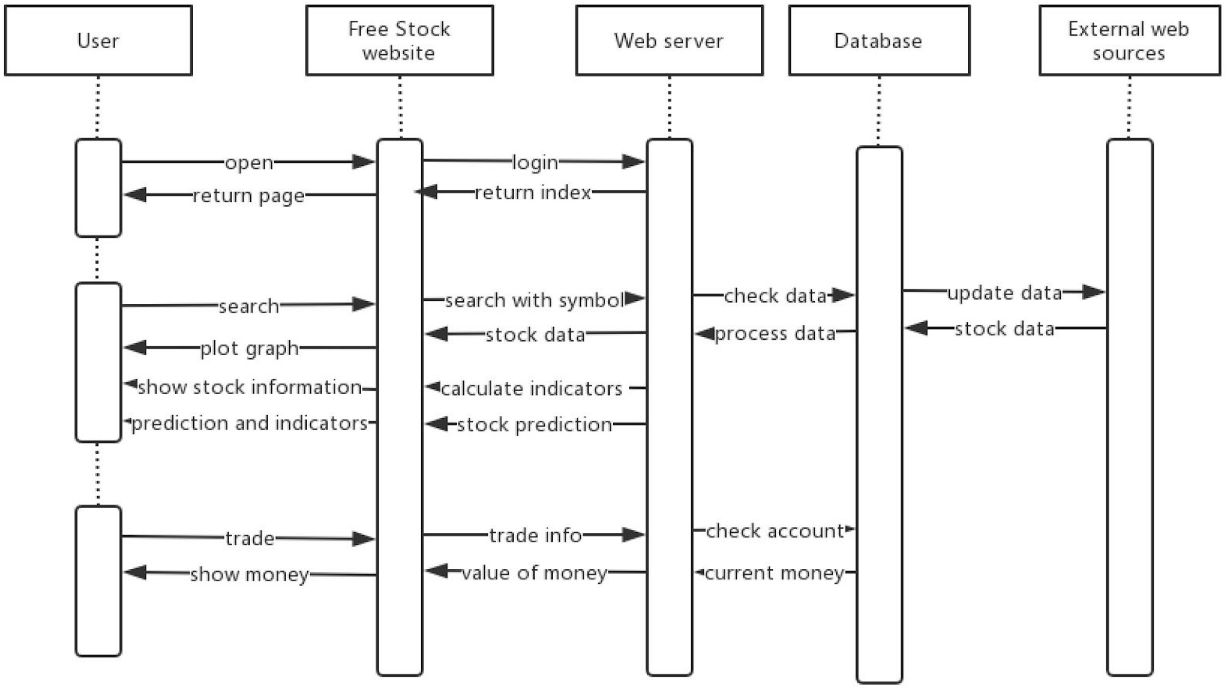
# 7 CLASS DIAGRAM AND INTERFACE SPECIFICATION

## 7.1 CLASS DIAGRAM

Overview:



Time sequence diagrams:



## 7.2 DATA TYPES AND OPERATION SIGNATURES

Stock Information:

- get_stock_highest(symbol: str): float

  Get the highest stock price of any company in the last one year

  Symbol: stock code

- get_stock_lowest(symbol: str): float

  Get the lowest stock price of any company in the last one year

  Symbol: stock code

- get_stock_average(symbol: str): float

  Get the average of stock price of any company in the last one year

  Symbol: stock code

Indicator:

- MA5(X: np.ndarray): float

  Calculate the value of MA5 indicator

  X: values of the indicator price

- MA10(X: np.ndarray): float

  Calculate the value of MA10 indicator

  X: values of the indicator price

- MA20(X: np.ndarray): float

  Calculate the value of MA20 indicator

  X: values of the indicator price

- EMA(X: np.ndarray): float

  Calculate the value of EMA indicator

  X: values of the indicator price

Prediction:

- predictBayes(symbol: str, period: int): float

  Help user predict the price of the stock they want with Bayesian Linear Regression

  Symbol: stock code

  Period: short-term or long-term

- predictDNN(symbol: str, period: int): float

Help user predict the price of the stock they want with Deep Neural Network

Symbol: stock code

Period: short-term or long-term

- predictSVR(symbol: str, period: int): float

  Help user predict the price of the stock they want with Support-Vector Regression

  Symbol: stock code

  Period: short-term or long-term

Virtual trade:

- buy_stock(price: float, volume: int): float

  Do the buy operation in the virtual trade

  Price: stock price of buy in

  Volume: the number of stock of buy in

- sell_stock(price: float, volume: int): float

  Do the sell operation in the virtual trade

  Price: stock price of sell out

  Volume: the number of stock of sell out

- search_trade(id: str): dict

  Help the user search his/her all trade information

# 8 SYSTEM ARCHITECTURE AND SYSTEM DESIGN

## 8.1 ARCHITECTURE STYLES

### 8.1.1 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

The microframework Flask is based on the Pocoo projects Werkzeug and Jinja2.

### 8.1.2 Jinja2

Jinja2 is a free and open-source front-end web framework for designing websites and web applications and a full featured template engine for Python. It has full unicode support, an optional integrated sandboxed execution environment, widely used and BSD licensed.

Jinja2 contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

## 8.2 Data Storage

In data storage part, we use MongoDB to create database and store the data. There are three tables in our database, they are historical price, realtime Prices and trading records.

### 8.2.1 Historical Price

Historical price is used to store the historical data of the stocks including symbol, high, low, close, open and timestamp. Here is a table to show the structure of the database.

| Name | Type | Description |
|---|---|---|
| Symbol | string | Name of stock |
| Timestamp | date | The date and time of record |
| Open | decimal128 | Open price |
| Close | decimal128 | Close price |
| Low | decimal128 | Low price |
| High | decimal128 | High price |
| Volume | int64 | Volume of stock |

### 8.2.2 Realtime Price

Realtime prices is a table which is used to store the real-time price and volume data of the stocks. It will be updated automatically every 30 seconds and insert a new value into the table. Below is the structure of the table.

| Name | Type | Description |
|---|---|---|
| Symbol | string | Name of stock |

| | | |
|---|---|---|
| Timestamp | date | The date and time of record |
| Price | decimal128 | Real time price of the stock |
| Volume | int64 | Volume of stock |

### 8.2.3 Trading Record

Trading record is a table to record the trade history of the virtual trading. Everytime the user trade a stock the system will store the trade detail. The structure of the table is as below.

| Name | Type | Description |
|---|---|---|
| User | string | Name of user |
| Symbol | string | Name of stock |
| Timestamp | date | The trade date and time of record |
| Price | decimal128 | Trade price of the stock |
| Number | int16 | Number of stock in trade |
| Buy/Sell | bool | True means buy while false means sell |

## 8.3 Web Service

### 8.3.1 Restful

REpresentational State Transfer (REST) is an architectural style that defines a set of constraints and properties based on HTTP. Web Services that conform to the REST architectural style, or RESTful web services, provide interoperability between computer systems on the Internet. RESTcompliant web services allow the requesting systems to access and manipulate textual representations of web resources by using a uniform and predefined set of stateless operations. Other kinds of web services, such as SOAP web services, expose their own arbitrary sets of operations.

"Web resources" were first defined on the World Wide Web as documents or files identified by their URLs. However, today they have a much more generic and abstract definition that encompasses every thing or entity that can be identified, named, addressed, or handled, in any way whatsoever, on the web. In a RESTful web service, requests made to a resource's URI will elicit a response that may be in XML, HTML, JSON, or some other format. The response may confirm that some alteration has been made to the stored resource, and the response may provide hypertext links to other related resources or collections of resources. When HTTP is used, as is most common, the operations available are GET, POST, PUT, DELETE, and other predefined CRUD HTTP methods.

By using a stateless protocol and standard operations, REST systems aim for fast performance, reliability, and the ability to grow, by re-using components that can be managed and updated without affecting the system as a whole, even while it is running.

### 8.3.2 RESTful API Design Definitions

**Resource**: A single instance of an object. For example, an animal. Collection: A collection of homogeneous objects. For example, animals. HTTP: A protocol for communicating over a network.

**Consumer**: A client computer application capable of making HTTP requests. Third Party Developer: A developer not a part of your project but who wishes to consume your data.

**Server**: An HTTP server/application accessible from a Consumer over a network.

**Endpoint**: An API URL on a Server which represents either a Resource or an entire Collection.

**Idempotent**: Side-effect free, can happen multiple times without penalty. URL Segment: A slash-separated piece of information in the URL.

### 8.3.3 Flask

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed.

The latest stable version of Flask is 0.12.2 as of May 2017. Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself.

Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

## 8.4 Library

### 8.4.1 Scikit-learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

**NumPy**: Base n-dimensional array package

**SciPy**: Fundamental library for scientific computing

**Matplotlib**: Comprehensive 2D/3D plotting

**IPython**: Enhanced interactive console

**Sympy**: Symbolic mathematics

Pandas: Data structures and analysis

Extensions or modules for SciPy care conventionally named SciKits. As such, the module provides learning algorithms and is named scikit-learn. The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as easy of use, code quality, collaboration, documentation and performance. Although the interface is Python, c-libraries are leverage for performance such as numpy for arrays and matrix operations, LAPACK, LibSVM and the careful use of cython.

### 8.4.2 Tensorflow

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google’s AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

# 9 ALGORITHMS AND DATA STRUCTURES

## 9.1 Bayesian Theory

In probability theory and statistics, Bayes' theorem (alternatively Bayes' law or Bayes' rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

Bayes' theorem is stated mathematically as the following equation:

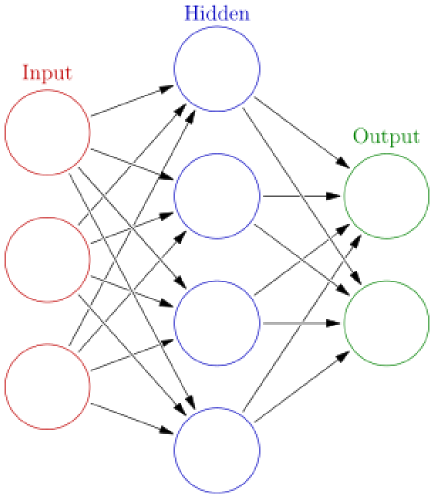
$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

## 9.2 Curve Fitting

Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points, possibly subject to constraints. Curve fitting can involve either interpolation, where an exact fit to the data is required, or smoothing, in which a "smooth" function is constructed that approximately fits the data. A related topic is regression analysis, which focuses more on questions of statistical inference such as how much uncertainty is present in a curve that is fit to data observed with random errors. Fitted curves can be used as an aid for data visualization, to infer values of a function where no data are available, and to summarize the relationships among two or more variables. Extrapolation refers to the use of a fitted curve beyond the range of the observed data, and is subject to a degree of uncertainty since it may reflect the method used to construct the curve as much as it reflects the observed data.

## 9.3 Deep Neural Network

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks. The goal is that eventually, the network will be trained to decompose an image into features, identify trends that exist across all samples and classify new images by their similarities without requiring human input.



We totally have 4 layers which means we set two hidden layer together with input layer and output layer.

Short term: 50 days’ prices as 50 inputs, 20 hidden neurons for each hidden layer and one output neuron.

Long term: 250 days’ prices as 250 inputs, 20 hidden neurons for each hidden layer and one output neuron.

Backpropagation is our way to adjust weights. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function.

Training Set:

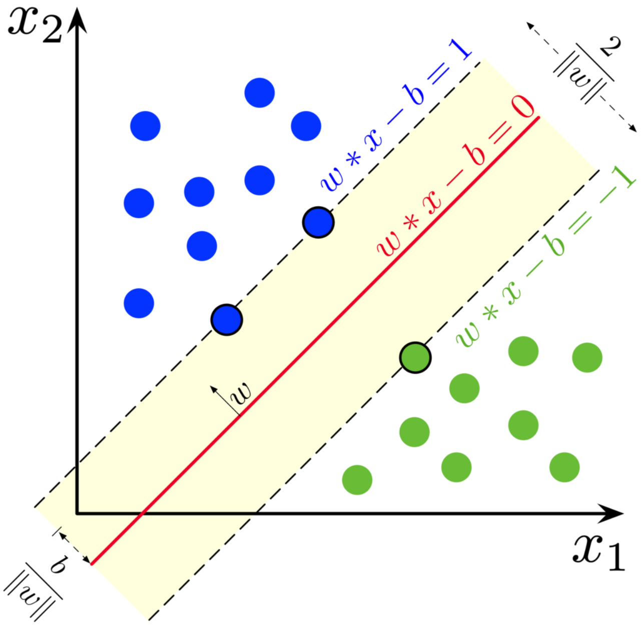
50 days’ prices before wanted days in short term

250 days’ prices before wanted days in long term

After training, we will choose 50 days’ price to predict price in short term and 250 days’ price to predict price in long term.

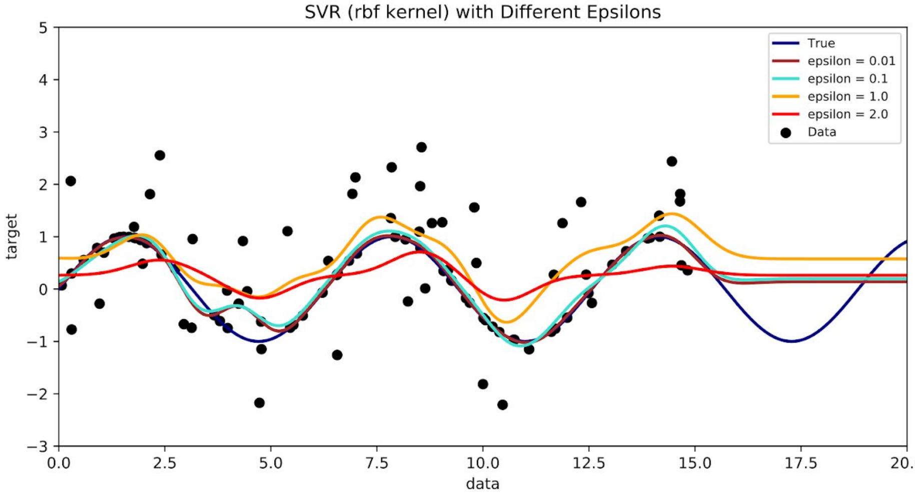
## 9.4 Support-Vector Machine

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.



## 9.5 Support-Vector Regression

A version of SVM for regression is called support-vector regression (SVR). The model produced by support-vector classification depends only on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction.



## 9.6 Moving Average

In statistics, a moving average (rolling average or running average) is a calculation to analyze data points by creating a series of averages of different subsets of the full data set. It is also called a moving mean (MM) or rolling mean and is a type of finite impulse response filter. Variations include: simple, and cumulative, or weighted forms.

Given a series of numbers and a fixed subset size, the first element of the moving average is obtained by taking the average of the initial fixed subset of the number series. Then the subset is modified by "shifting forward"; that is, excluding the first number of the series and including the next value in the subset.

A moving average is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles. The threshold between short-term and long-term depends on the application, and the parameters of the moving average will be set accordingly. For example, it is often used in technical analysis of financial data, like stock prices, returns or trading volumes. It is also used in economics to examine gross domestic product, employment or other macroeconomic time series. Mathematically, a moving average is a type of convolution and so it can be viewed as an example of a low-pass filter used in signal processing. When used with non-time series data, a moving average filters higher frequency components without any specific connection to time, although typically some kind of ordering is implied. Viewed simplistically it can be regarded as smoothing the data.

$$CMA_n = \frac{x_1 + \cdots + x_n}{n}\,.$$

In our system, we use MA5, MA10 and MA20 to see the trend of the stock price and compare different MA lines, we could get more information to help user understand the stock better.

## 9.7 Exponential moving average

An exponential moving average (EMA), also known as an exponentially weighted moving average (EWMA), is a first-order infinite impulse response filter that applies weighting factors which decrease exponentially. The weighting for each older datum decreases exponentially, never reaching zero. The graph at right shows an example of the weight decrease.

The EMA for a series Y may be calculated recursively:

$$S_t = \begin{cases} Y_1, & t = 1 \\ \alpha \cdot Y_t + (1 - \alpha) \cdot S_{t-1}, & t > 1 \end{cases}$$

Where:

- The coefficient α represents the degree of weighting decrease, a constant smoothing factor between 0 and 1. A higher α discounts older observations faster.
- Yt is the value at a time period t.
- St is the value of the EMA at any time period t.

Moving averages smooth the price data to form a trend following indicator. They do not predict price direction, but rather define the current direction with a lag. Moving averages lag because they are based on past prices. Despite this lag, moving averages help smooth price action and filter out the noise. They also form the building blocks for many other technical indicators and overlays, such as Bollinger Bands, MACD and the McClellan Oscillator. The two most popular types of moving averages are the Simple Moving Average (SMA) and the Exponential Moving Average (EMA). These moving averages can be used to identify the direction of the trend or define potential support and resistance levels.

## 9.8 MACD

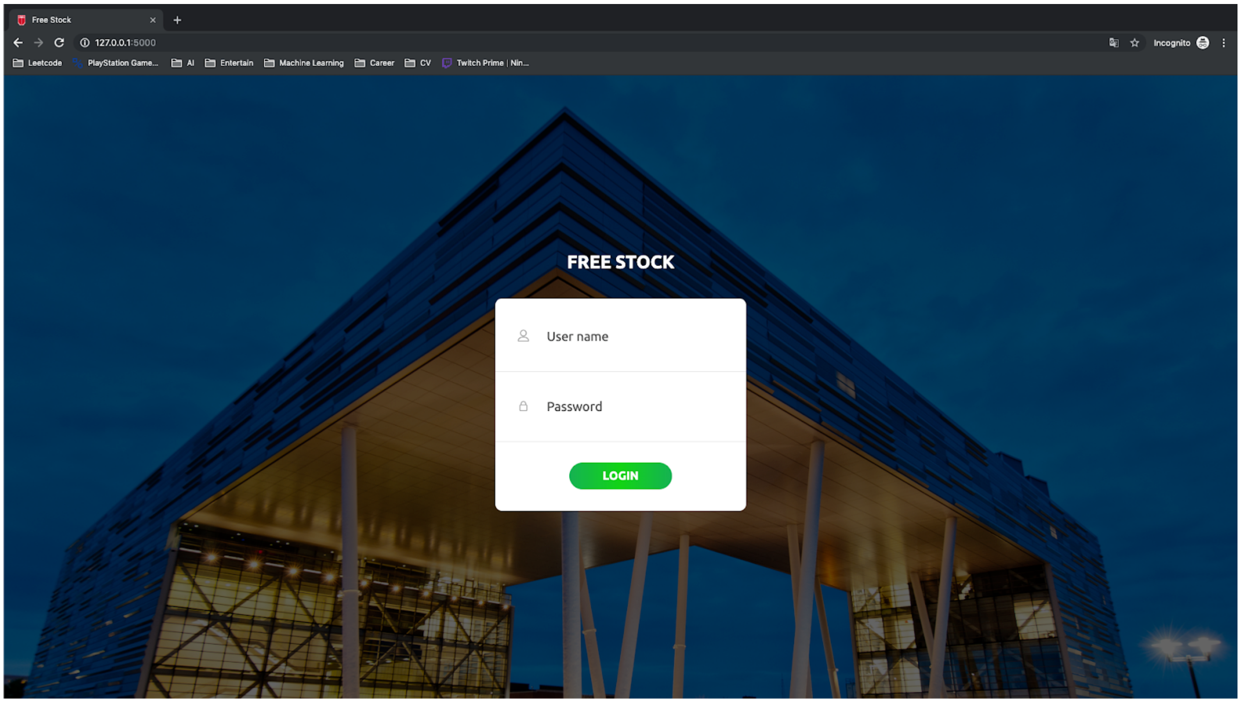
MACD, short for moving average convergence/divergence, is a trading indicator used in technical analysis of stock prices, created by Gerald Appel in the late 1970s. It is designed to reveal changes in the strength, direction, momentum, and duration of a trend in a stock's price.

The MACD indicator is a collection of three time series calculated from historical price data, most often the closing price. These three series are: the MACD series proper, the "signal" or "average" series, and the "divergence" series which is the difference between the two. The MACD series is the difference between a "fast" (short period) exponential moving average (EMA), and a "slow" (longer period) EMA of the price series. The average series is an EMA of the MACD series itself.

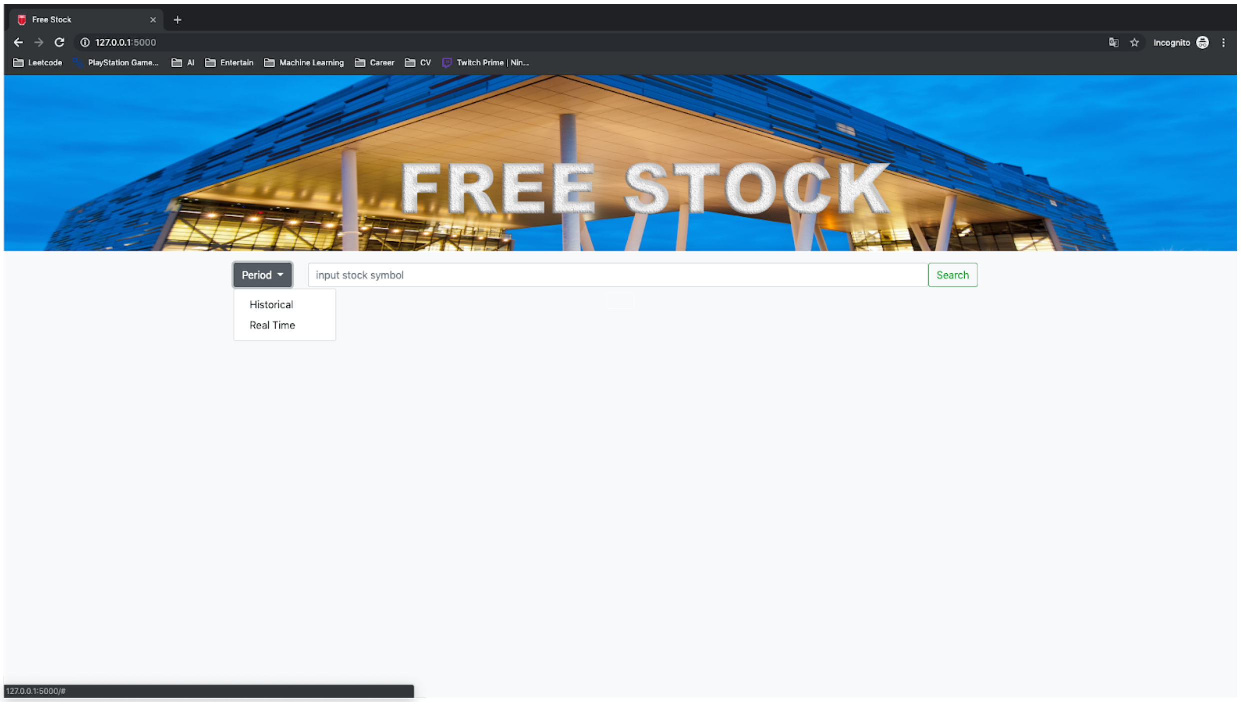
The MACD indicator thus depends on three time parameters, namely the time constants of the three EMAs. The notation "MACD(a,b,c)" usually denotes the indicator where the MACD series is the difference of EMAs with characteristic times a and b, and the average series is an EMA of the MACD series with characteristic time c. These parameters are usually measured in days. The most commonly used values are 12, 26, and 9 days, that is, MACD(12,26,9). As true with most of the technical indicators, MACD also finds its period settings from the old days when technical analysis used to be mainly based on the daily charts. The reason was the lack of the modern trading platforms which show the changing prices every moment. As the working week used to be 6-days, the period settings of (12, 26, 9) represent 2 weeks, 1 month and one and a half week. Now when the trading weeks have only 5 days, possibilities of changing the period settings cannot be overruled. However, it is always better to stick to the period settings which are used by the majority of traders as the buying and selling decisions based on the standard settings further push the prices in that direction.

# 10 USER INTERFACE DESIGN AND IMPLEMENTATION

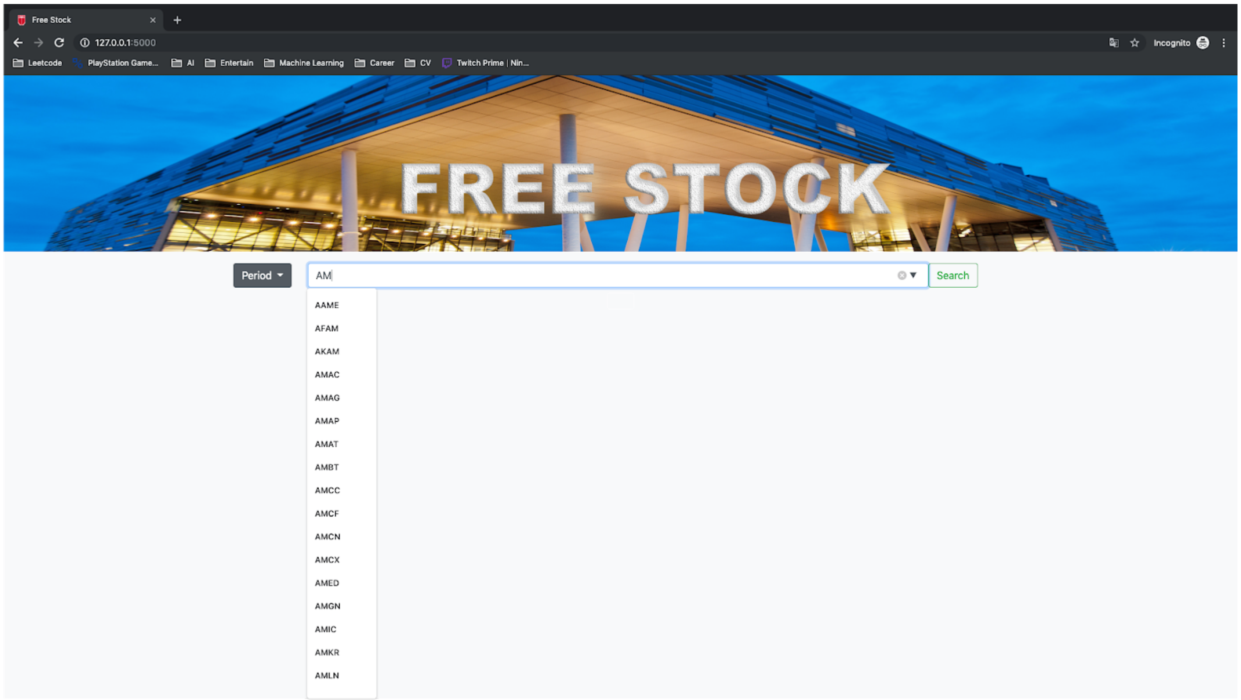
## 10.1 Login Page



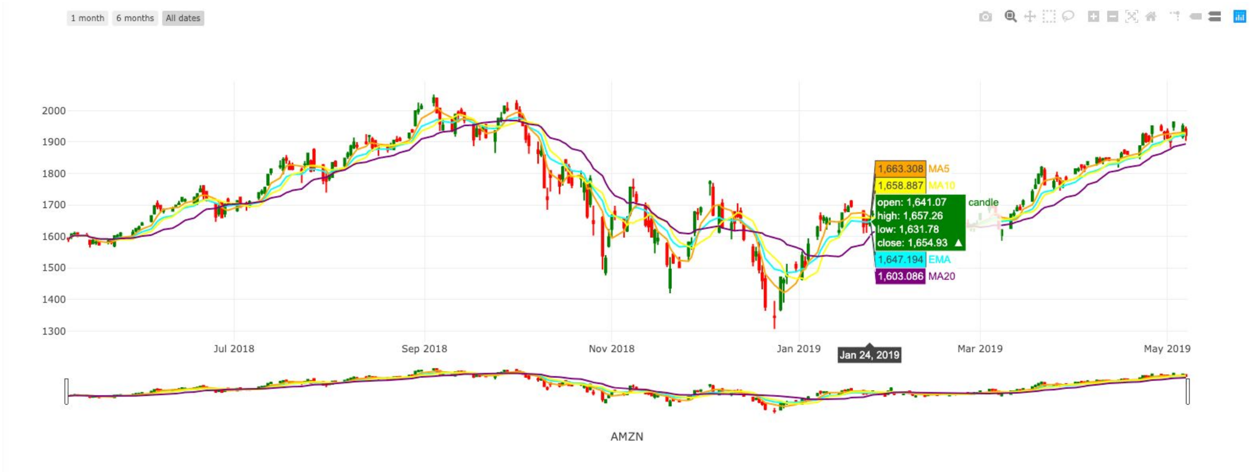
## 10.2 Period Switch



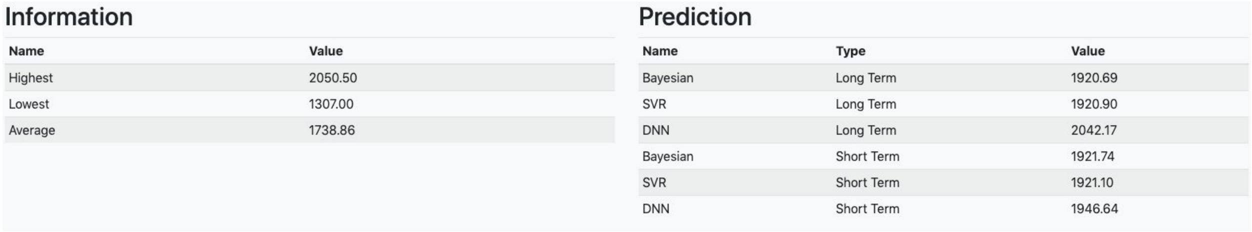
## 10.3 Stock Code Assistance



## 10.4 Historical Data



## 10.5 Indicator And Prediction



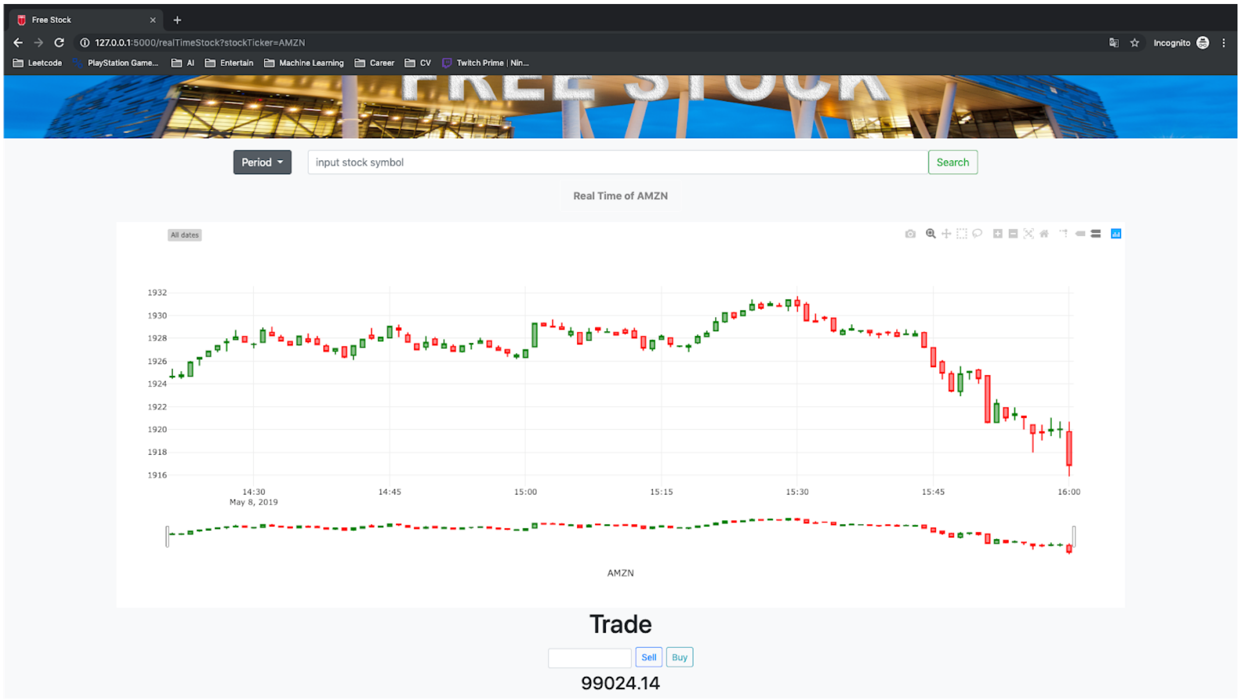
### Information

| Name | Value |
|---|---|
| Highest | 2050.50 |
| Lowest | 1307.00 |
| Average | 1738.86 |

### Prediction

| Name | Type | Value |
|---|---|---|
| Bayesian | Long Term | 1920.69 |
| SVR | Long Term | 1920.90 |
| DNN | Long Term | 2042.17 |
| Bayesian | Short Term | 1921.74 |
| SVR | Short Term | 1921.10 |
| DNN | Short Term | 1946.64 |

## 10.6 Real-time Data And Virtual Trade



# 11 RELATED WORK AND FUTURE WORK

## 11.1 Related work

There are many common used existing systems like Yahoo Finance, Google Finance, Bloomberg. They provide users to visit different stocks and view their summary. Also, users can query different types of data of the specific stock, offer the figure of the stock for a certain time period and predict the future stock price to help making investment decisions. In their indicator and prediction, they provide more methods than us and could create document report to help user analysis the stock and more helpful in real trade. And these work not only focuses on the stock price,  but also provide more information in financiance, world event and etc..

## 11.2 Future Work

First and most important would be to integrate the prediction models into the website. We ran into some technical issues which prevented us from actually doing this, mainly the fact that they were written in different languages. We are also not proficient in all of the technologies that could have allowed us to do this.

Second, we have to improve our front-end to provide a more user-friendly interface because our front-end is simple and crude now. The website will be more useful and easier to use.

Then, more functions will be added to the website. For example, users can search a certain stock by typing in its symbol into a search box. At the same time, login and sign up functions will be added, so we can send notification emails to users then.

Another idea we had toyed with but never got the chance to implement was even more technical indicators but that would complicate the system. In order to reduce this complication, we could use all of these indicators, along with a neural network, using the back propagation method, to generate states, this way we have a prediction algorithm using indicators that would all analyze different areas and generate one prediction, rather than just look at each indicator individually.

Last but not least, we would perfect our trade system, the database would record more information of each trade, and add more restriction on the trade. What’s more, we would integrate the prediction system into the trade system, support a new and amazing portfolio backtesting function.