Stock Price Data Processing & Prediction System

16:332:568 Spring 2019

Software Engineering of Web Applications

Team #1

Yuhang Zhou (yz853)

Jiachen Ding (jd1287)

Lichuan Ren (lr629)

Haofan Zhang (hz332)

5/8/2019

1.Contribution Breakdown

All our group members make their efforts to contribute to our project, we consider this as an equal contribution

Yuhang Zhou:

Project management and web service development

Jiachen Ding:

Prediction algorithm and back-end development

Lichuan Ren:

Database processing and front-end development

Haofan Zhang:

Special feature development and back-end development

2.Costumer Statement of Requirements

2.1 Motivation

Investment stock market is getting more and more popular recently. Because of internet popularity, everyone has access to the investment stock market. Also, the investment market is pretty mature nowadays. With more and more people participating in investment areas, people's attitude to investment behavior is getting more and more positive.

With the improvement of life quality and increasing of incomes, people start considering taking advantage of idle funds they owned, so the stock market is one of the largest areas they would like to invest.

While most people are not familiar with the investment behaviors, all they can do is listening to the so-called expert's speech and follow others. So it's important for investors to own their own ideas about investment.

As we know, there are two basic investing analysis methods which are the fundamental analysis and technical analysis. The fundamental analysis is always suitable for investors who own massive funds and aim to obtain benefits in the long term. While most investors are small scale investors, and they aim to obtain benefits in the short term. So that the technical analysis is one of the most important methods for them to get benefits in the short term.

2.2 Project introduction

Our project aims to help those small-scale investors. Our project will provide efficient technical analysis tools to help them get access to the stock price and volume data. Neural network technology will also be used in our project to predict stock price in the future based on the history data. Also, our project can find a similar trend in the history data as a great reference for investors.

There are mainly three functions in our project. The first function is providing access to historical data and real-time data, also the graph based on the history data will be drawn for investors. The second function is using a neural network to predict the future price trend based on selected historical data. The last function is searching the time interval whose price trend is similar to the selected time interval, also the price trend after the found time interval will be provided for investors as reference.

3.Glossary of Terms

Artificial Neural network: Artificial neural networks (ANN) is computing systems vaguely inspired by the biological neural networks that constitute animal brains.

Database: A database is an organized collection of data, generally stored and accessed electronically from a computer system.

Front-end and back-end: the terms front end and back end refer to the separation of concerns between the presentation layer (front end), and the data access layer (back end) of a piece of software, or the physical infrastructure or hardware.

Flask: Flask is a micro web framework written in Python.

Jinja: Jinja is a web template engine for the Python programming language and is licensed under a BSD License created by Armin Ronacher.

Bootstrap: Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

Dense Layer: A dense layer represents a matrix vector multiplication. The values in the matrix are the trainable parameters which get updated during backpropagation.

Dropout layer: A dropout layer is used for regularization where you randomly set some of the dimensions of your input vector to be zero with probability $keepprob$. A dropout layer does not have any trainable parameters i.e. nothing gets updated during backward pass of backpropagation. To ensure that expected sum of vectors fed to this layer remains the same if no dropout was applied, the remaining dimensions which are not set to zero are scaled by $\frac{1}{keepprob}$.

Normalization: In another usage in statistics, normalization refers to the creation of shifted and scaled versions of statistics, where the intention is that these normalized values allow the comparison of corresponding normalized values for different datasets in a way that eliminates the effects of certain gross influences, as in an anomaly time series.

Activation Function: In artificial Neural Network, the activation function of a node defines the output of that node, or "neuron," given an input or set of inputs. This output is then used as input for the next node and so on until a desired solution to the original problem is found. It maps the resulting values into the desired range such as between 0 to 1 or -1 to 1 etc. (depending upon the choice of activation function).

4.System Requirements

4.1 Enumerated Functional Requirements

REQ1: The system can collect historical data and real-time data of stock price.

REQ2: The webpage should show the real-time data of the selected company.

REQ3: The webpage should show the historical data of the selected time interval and company.

REQ4: The system should using machine learning technology to predict the stock price of a selected company.

4.2 On-Screen Appearance Requirements

REQ1: There should a fixed navigation bar in the top of the webpage.

REQ2: The navigation bar should have the link to other functional pages including the home page.

REQ3: The link to the source code repository should be in the button of the web page.

REQ4: The link to the API documentation should be in the button of the web page.

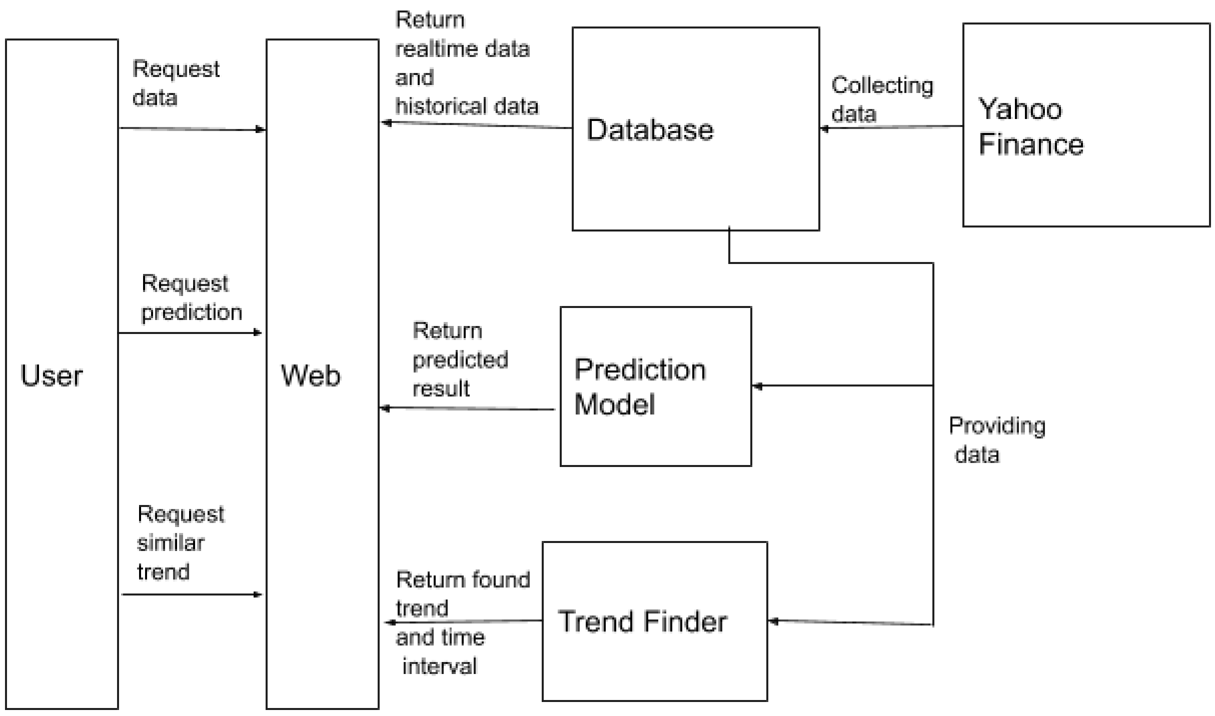
REQ5: The home page shows the basic information about this system.

REQ6: In the functional pages, there should be a form for the user to type in the time range and select company.

REQ7: After submitting the form, the results should show in graph and charts below the form.

5.System Architecture and Design

5.1 Basic Architecture

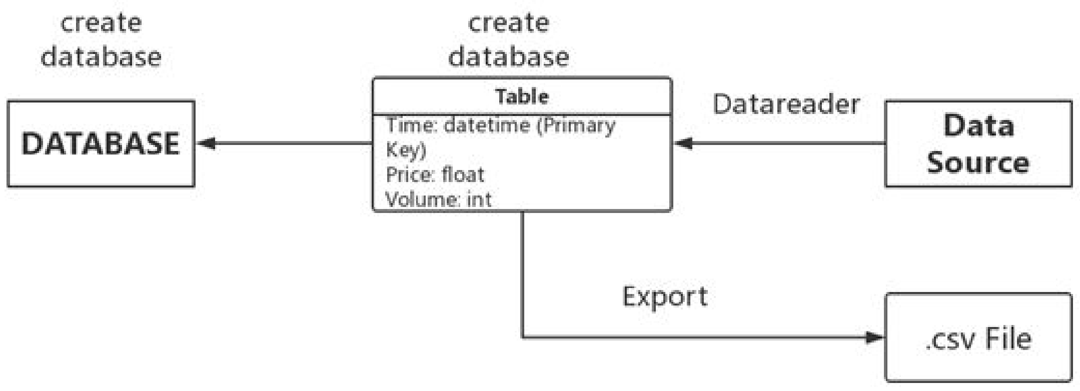


5.2 Use cases

Use case	Permission	Cases
Case 1	Administrator	System management
Case 2	Administrator	Update historical data of the database

Case 3	User	Get real-time the price and volume of the selected company
Case 4	User	Get historical data of the selected company
Case 5	User	Plot the historical data
Case 6	User	Get the prediction of the selected company
Case 7	User	Search time interval whose trend is similar to the selected trend
Case 8	User	Plot the similar trend, and the trend after the found time interval as reference

5.3 Data Collection



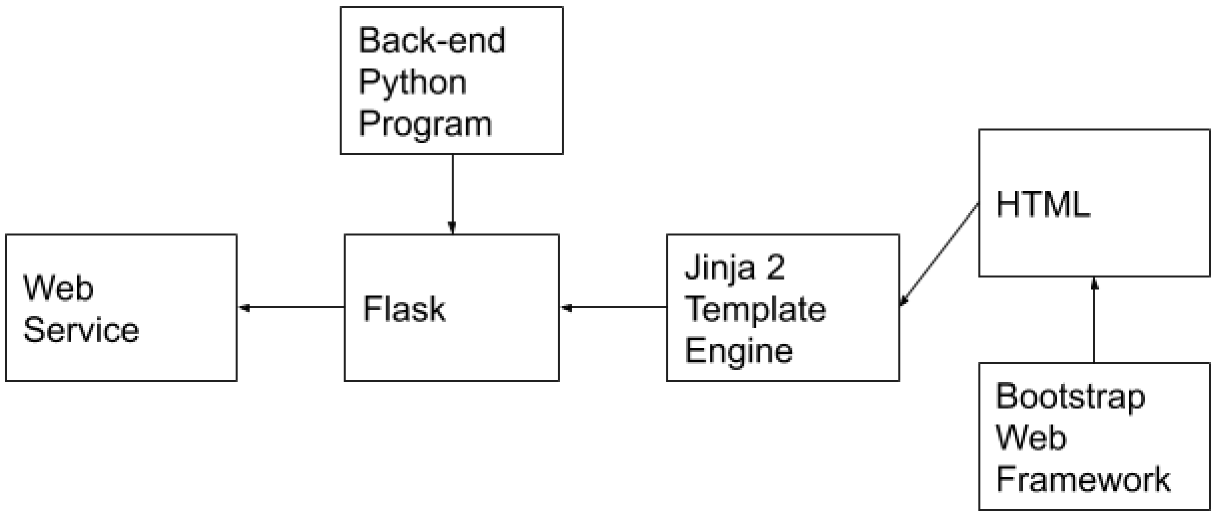
Our system consists of a local database which contains tables for storing stock prices, a data reader module which captures real-time data from the internet and an export module which exports data into a .csv file.

First, we used MySQL to create a local database and the tables in the database to record the data. For the real-time data, we embedded data with three keys, which are Time, Price and Volume. The Time stands for date time when the price is sampled, Volume stands for the volume of a specific stock. While there are more detailed keys for the price of history data like High, Low, Open, Close, Adj Close.

To seize history data, we use the DataReader function to get the data in the time range as we want. To seize real-time data from the internet, we have a timer to periodically start get_quote_yahoo function to get data from Yahoo and insert into the table as an update.

Another module is to export selected data from a table in the company domain. Any company can be added into the dictionary. The exported data is saved as a .csv file. History data as company.csv and real-time data as company+rt.csv.

5.4 Web Service



The web service of our project is based on the RESTful web API.

Considering our project is mostly based on python, so we choose to use the Flask as our web service framework. Flask is a micro web framework written in Python. Flask is a microframework based on Werkzeug, Jinja 2 and good intentions. With the Flask framework, we can make these back-end python functions as web service, also provide users to interact with the HTML pages we made.

The Jinja 2 template engine is included in the Flask framework. We are using this engine to render our HTML webpages. This engine can help us to make our webpages more flexible and powerful.

To make our webpages look more beautiful, we use bootstrap to beautify our webpages. The framework we are using is the “Bootstrap_Flask” which can work with the Flask framework. The bootstrap is an open source front-end framework, contains frameworks for the HTML, CSS, and Javascript, and provide lots of extensions.

5.5 Prediction

To make prediction, we implemented 3 different strategies which are SVM, Bayesian Curve Fitting and Long-Short Term Memory.

SVM is known as a tool for classification and it basically project higher dimensional vectors into lower dimensional plane to make division of the plane so that it can make classification. But in our project we utilize it as a predicting tool since we believe that it can also be capable of predicting. Being fed with large quantity of historical data in a long period, it can learn the trend of price changing and be able to predict. And the outcome can be great when validating.

Bayesian Curve Fitting can be used to cultivate the trend which can be curved into linear regression and polynomial regression. Stock price changes can be curved into relatively precise polynomial regression but the order can be found only by testing. In our project we follow through 10 stocks and use historical data to find out a polynomial expression that fit the data most likely and with this expression predictions of both short-term and long-term can be made.

LSTM is an advanced version of Recurrent Neural Network. It has all the components that normal RNN has and what is more, it has a cell gate which can decide information to be either remembered or forgot. Since stock price can be influenced by history and the possible change can be characterized from the past, it is useful to apply LSTM to learn the potential trend. We take historical data from the past 3 years to train the model. Since the input data is fed as batch and model is trained batch by batch, to make prediction, we use price of the latest 50 days to generate the prediction of the price of tomorrow. And the next step is we use the new latest data to concurrently obtain prediction. Via this we can get both the short term and long term prediction. But this functionality requires high accuracy of the model, so we trained the model for 500 epochs.

5.6 Trend Finder

The trend finder is another key function of our system, we use this function to find the old similar trend of each. After, the old similar trend has been found. The program will store the trend data into pictures and upload to the web service.
The trend finds programming was developed by python and use the Pearson correlation coefficient to find a similar trend. The Pearson correlation coefficient is a measure of the linear correlation between two variables X and Y. According to the Cauchy–Schwarz inequality, it has a value between +1 and −1, where 1 is a total positive linear correlation, 0 is no linear correlation, and −1 is a total negative linear correlation. It is widely used in the sciences. It was developed by Karl Pearson from a related idea introduced by Francis Galton in the 1880s and for which the mathematical formula was derived and published by Auguste Bravais in 1844.

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \qquad (Eq.1)$$

where:

- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

The formula for ρ can be expressed in terms of mean and expectation. Since

$$\operatorname{cov}(X,Y) = \operatorname{E}[(X-\mu_X)(Y-\mu_Y)],$$ [7]

the formula for ρ can also be written as

$$\boxed{\rho_{X,Y} = \frac{\operatorname{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y} \qquad \textit{(Eq.2)}}$$

where:

- σ_Y and σ_X are defined as above
- μ_X is the mean of X
- μ_Y is the mean of Y
- E is the expectation.

The formula for ρ can be expressed in terms of uncentered moments. Since

- $\mu_X = \operatorname{E}[X]$
- $\mu_Y = \operatorname{E}[Y]$
- $\sigma_X^2 = \operatorname{E}[(X-\operatorname{E}[X])^2] = \operatorname{E}[X^2] - [\operatorname{E}[X]]^2$
- $\sigma_Y^2 = \operatorname{E}[(Y-\operatorname{E}[Y])^2] = \operatorname{E}[Y^2] - [\operatorname{E}[Y]]^2$
- $\operatorname{E}[(X-\mu_X)(Y-\mu_Y)] = \operatorname{E}[(X-\operatorname{E}[X])(Y-\operatorname{E}[Y])] = \operatorname{E}[XY] - \operatorname{E}[X]\operatorname{E}[Y]$

the formula for ρ can also be written as

$$\rho_{X,Y} = \frac{\operatorname{E}[XY] - \operatorname{E}[X]\operatorname{E}[Y]}{\sqrt{\operatorname{E}[X^2] - [\operatorname{E}[X]]^2}\ \sqrt{\operatorname{E}[Y^2] - [\operatorname{E}[Y]]^2}}.$$

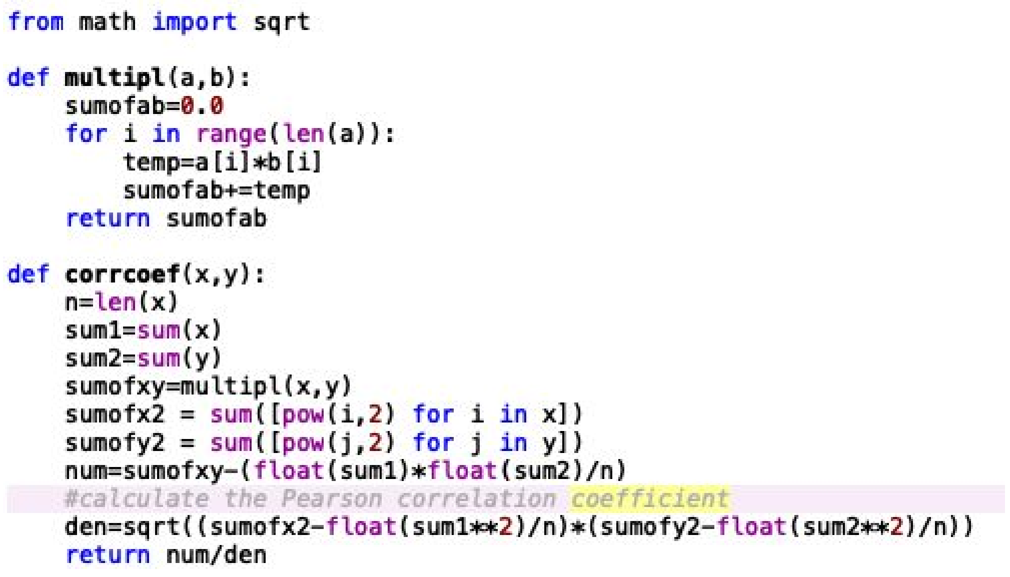
Pearson's correlation coefficient when applied to a sample is commonly represented by rxy and may be referred to as the sample correlation coefficient or the sample Pearson correlation coefficient. We can obtain a formula for rxy by substituting estimates of the covariances and variances based on a sample into the formula above.

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

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 $\bar{y}$

The absolute values of both the sample and population Pearson correlation coefficients are less than or equal to 1. Correlations equal to 1 or −1 correspond to data points lying exactly on a line (in the case of the sample correlation), or to a bivariate distribution entirely supported on a line (in the case of the population correlation). The Pearson correlation coefficient is symmetric: corr(X,Y) = corr(Y,X). A key mathematical property of the Pearson correlation coefficient is that it is invariant under separate changes in location and scale in the two variables. That is, we may transform X to a + bX and transform Y to c + dY, where a, b, c, and d are constants with b, d > 0, without changing the correlation coefficient. (This holds for both the population and sample Pearson correlation coefficients.) Note that more general linear transformations do change the correlation: see Decorrelation of n random variables for an application of this.
The application of Pearson correlation coefficient in our program is set a the eventual value of Pearson correlation coefficient result and input a list of recent data. After the data was input, we could find the other list of data which is history data. The history data should have similar trend compared with history data. The specific python code is implemented as follows：



```
from math import sqrt

def multipl(a,b):
    sumofab=0.0
    for i in range(len(a)):
        temp=a[i]*b[i]
        sumofab+=temp
    return sumofab

def corrcoef(x,y):
    n=len(x)
    sum1=sum(x)
    sum2=sum(y)
    sumofxy=multipl(x,y)
    sumofx2 = sum([pow(i,2) for i in x])
    sumofy2 = sum([pow(j,2) for j in y])
    num=sumofxy-(float(sum1)*float(sum2)/n)
    #calculate the Pearson correlation coefficient
    den=sqrt((sumofx2-float(sum1**2)/n)*(sumofy2-float(sum2**2)/n))
    return num/den
```

6.Algorithms

6.1 Support Vector Machine

SVM is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. It can also be used to do regression which indicates prediction. So in our project we implement SVM to do prediction. In this project, we focus on the prediction of the trend of stock market (either increase or decrease). Therefore, the change of a feature over time is more important than the absolute value of each feature.

We define $x_i(t)$, where $i \in \{1, 2, ...16\}$ to be feature i at time t. The feature matrix is given by

$$F = (X1,\ X2,\ ...,\ Xn)^T\ ,$$

where

$$Xt = (x_1(t), x_2(t), ..., x_{16}(t)) .$$

The new feature which is the difference between two daily prices can be calculated by

$$\nabla\delta x_i(t) = x_i(t) - x_i(t - \delta)$$

$$\nabla\delta X(t) = X(t) - X(t - \delta) = (\nabla\delta x_1(t),\ \nabla\delta x_2(t),\ \cdots \nabla\delta x_{16}(t))^T$$

$$\nabla\delta F = (\nabla\delta X(\delta + 1),\ \nabla\delta X(\delta + 2),\ ...,\ \nabla\delta X(n))$$

Due to the difference in market value and basis of each market, the differential values calculated above can vary in a wide range. To make them comparable, the features are normalized as following:

$$\begin{aligned}
\mathcal{N}(\nabla_\delta x_i(t)) &= \frac{x_i(t) - x_i(t-\delta)}{x_i(t-\delta)} \\
\mathcal{N}(\nabla_\delta X(t)) &= (\mathcal{N}(\nabla_\delta x_1(t)), \cdots, \mathcal{N}(\nabla_\delta x_{16}(t)))^T \\
\mathcal{N}(\nabla_\delta(\mathcal{F})) &= (\mathcal{N}(\nabla_\delta X(\delta+1)), \cdots, \mathcal{N}(\nabla_\delta X(n)))^T
\end{aligned}$$

and the normalization can be implemented as:

$$normal(X(t)) = \frac{\mathcal{N}(\nabla_\delta X(t))}{|\mathcal{N}(\nabla_\delta X(t))|}$$

By calculating the $X(t)$ we can get prediction of the price.

6.2 Bayesian Curve Fitting

Given the training dataset χ and t, along with a new test point x ,by using polynomial regression method to predict the value of t, i.e. to evaluate the predictive distribution $p(t \mid x, \chi, t)$. Assume that the parameters α and β are fixed and known in advance. We have

$$p(t \mid x, \chi, t) = \int p(t \mid x, w) p(w \mid \chi, t) dw$$

where

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right)$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) p(\mathbf{w}|\alpha)$$

by some calculus we have

$$p(t|x,\mathbf{x},\mathbf{t}) = \int p(t|x,\mathbf{w})p(\mathbf{w}|\mathbf{x},\mathbf{t})\,\mathrm{d}\mathbf{w} = \mathcal{N}\left(t|m(x),s^2(x)\right)$$

$$m(x) = \beta\boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)t_n \qquad s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\boldsymbol{\phi}(x)$$

$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^{\mathrm{T}} \qquad \boldsymbol{\phi}(x_n) = \left(x_n^0,\ldots,x_n^M\right)^{\mathrm{T}}$$

thus

$$p(t|x,\mathbf{x},\mathbf{t}) = \mathcal{N}\left(t|m(x),s^2(x)\right)$$

This is the predictive distribution resulting from the Bayesian treatment of polynomial curve fitting. In our project we set the polynomial order as 6 which has been verified as the best value of this feature to gain the best performance.

6.3 Long-Short Term Memory

There are several architectures of LSTM units. A common architecture is composed of a cell (the memory part of the LSTM unit) and three "regulators", usually called gates, of the flow of information inside the LSTM unit: an input gate, an output gate and a forget gate. Some variations of the LSTM unit do not have one or more of these gates or maybe have other gates. Intuitively, the cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The activation function of the LSTM gates which is sigmoid. There are connections into and out of the LSTM gates, a few of which are recurrent. The weights of these connections, which need to be learned during training, determine how the gates operate.

The compact forms of the equations for the forward pass of an LSTM unit with a forget gate are:

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

where

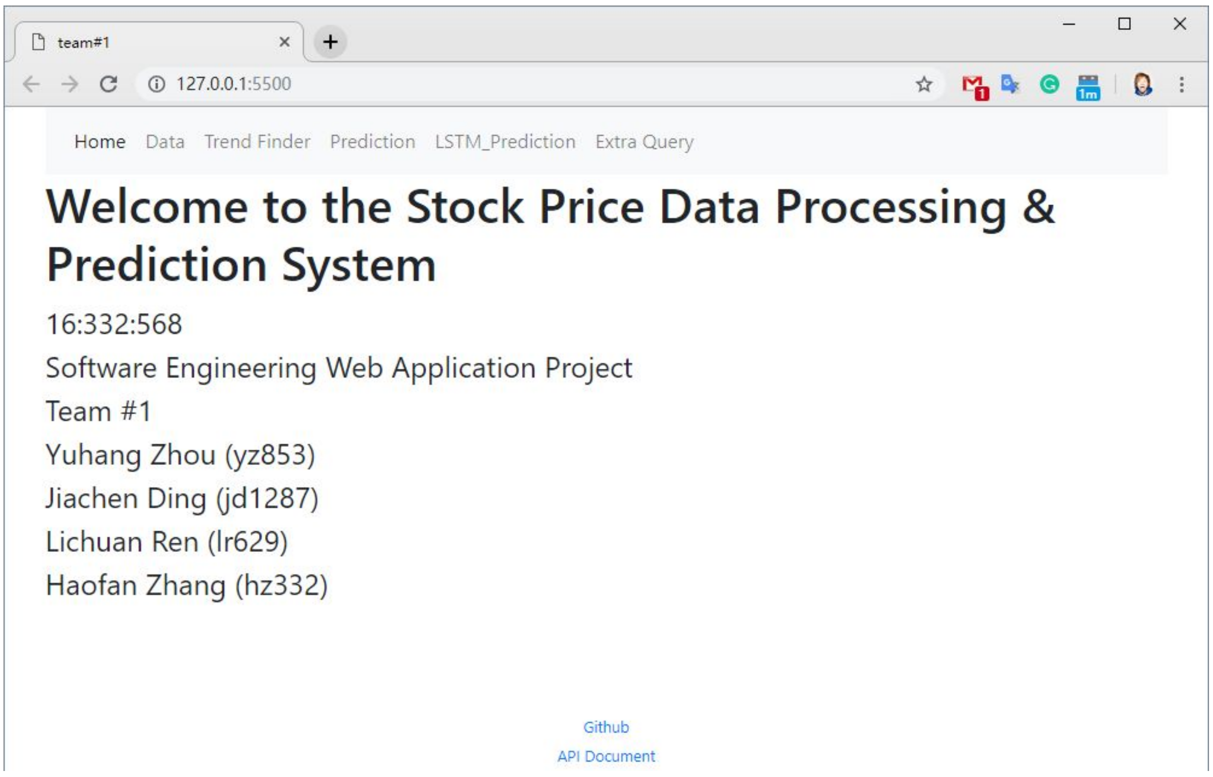
$x_t \in \mathbb{R}^d$: input vector to the LSTM unit
$f_t \in \mathbb{R}^h$: forget gate's activation vector
$i_t \in \mathbb{R}^h$: input gate's activation vector

$o_t \in \mathbb{R}^h$: output gate's activation vector
$h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit
$c_t \in \mathbb{R}^h$: cell state vector

In our project, we first have data preprocessing module to do data normalization which aims to normalize input data into $[-1, 1]$ to relieve the burden of the model. Then we set up 2 LSTM connected with 2 dropout layers between each to make feature extraction which refers to learning information from history. The purpose of adding dropout layers is to prevent overfitting since the size of training dataset is relatively small considering the size of the model. Once the features are extracted we feed them into dense layer which is a fully-connected layer to make decision, after making comparison with the true result, the predictive value will be propagate back as the BP algorithm suggests to renew the weights of the model. That is basically how this whole model works.

7.User Interface and Implementation

7.1 Home Page



Welcome to the Stock Price Data Processing & Prediction System

16:332:568

Software Engineering Web Application Project

Team #1

Yuhang Zhou (yz853)

Jiachen Ding (jd1287)

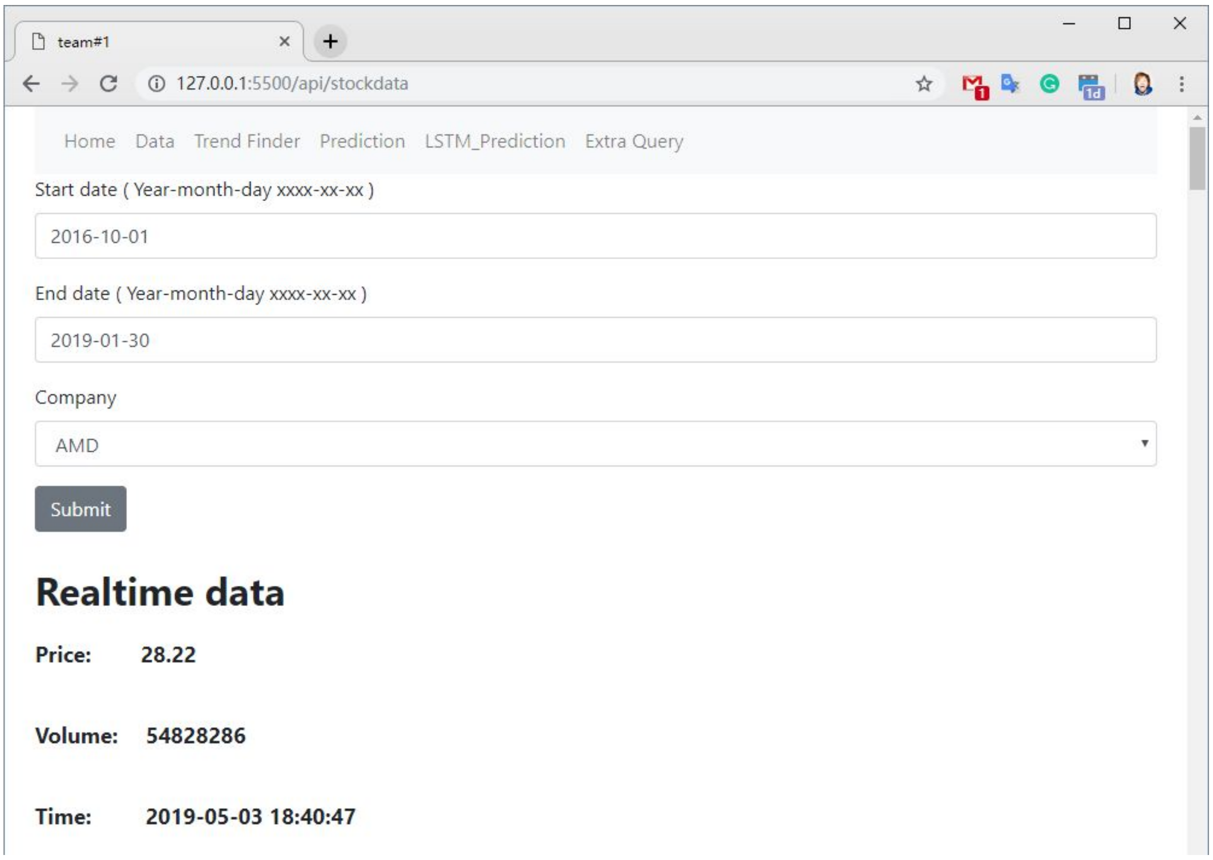
Lichuan Ren (lr629)

Haofan Zhang (hz332)

The screenshot above shows the home page of our user interface, which displays our project name and our team member information, as well as the program navigation bars. By mouse clicks the navigation fields, our users can access their desired functionalities. We have six different functionalities in our navigation fields.

7.2 Data Access

By a mouse click at Data, our user can access our first function, where they can access real-time stock price data for a particular stock, or they can look up historical stock data.



Start date (Year-month-day xxxx-xx-xx)

2016-10-01

End date (Year-month-day xxxx-xx-xx)

2019-01-30

Company

AMD

Realtime data

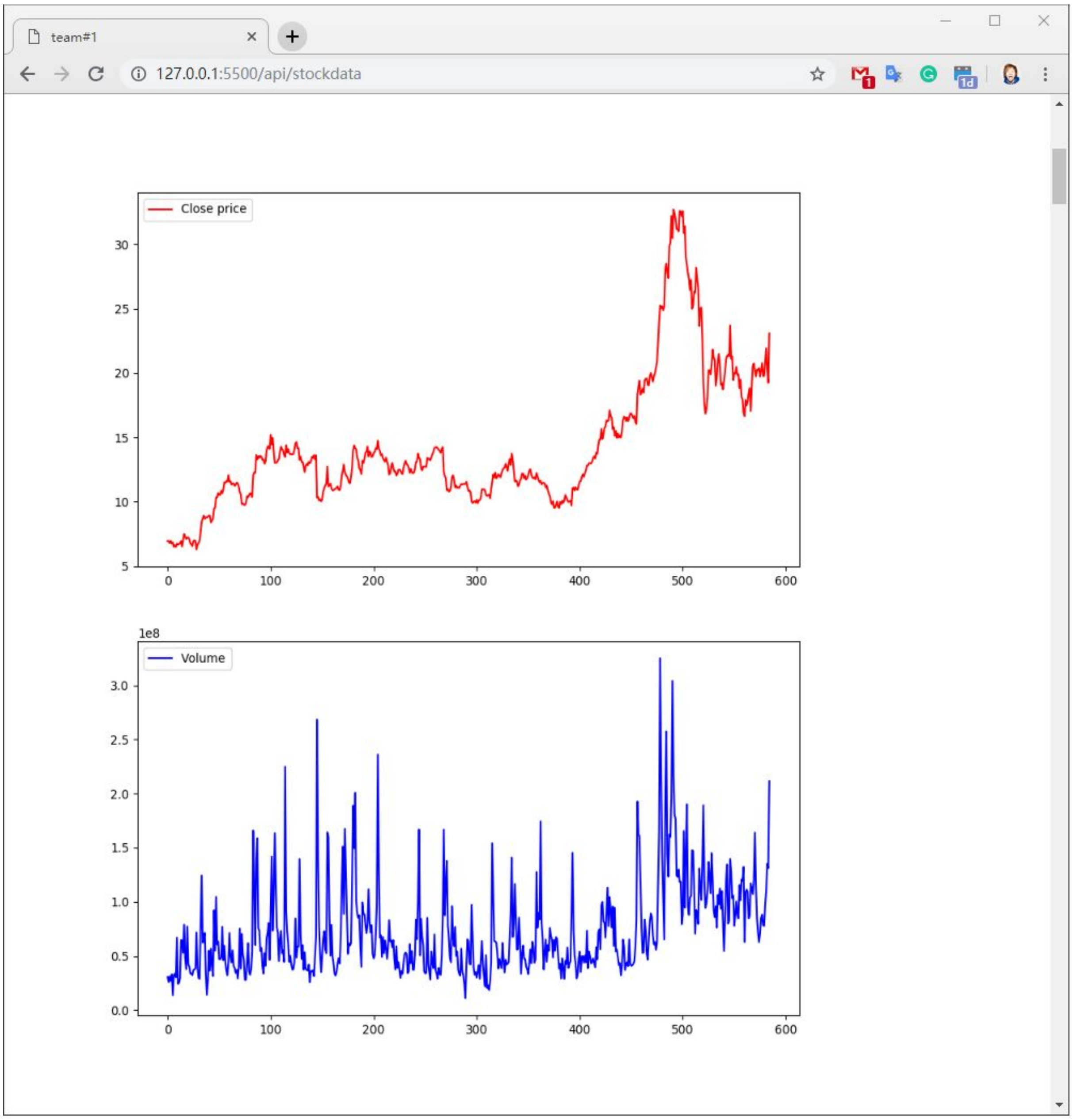
Price: 28.22

Volume: 54828286

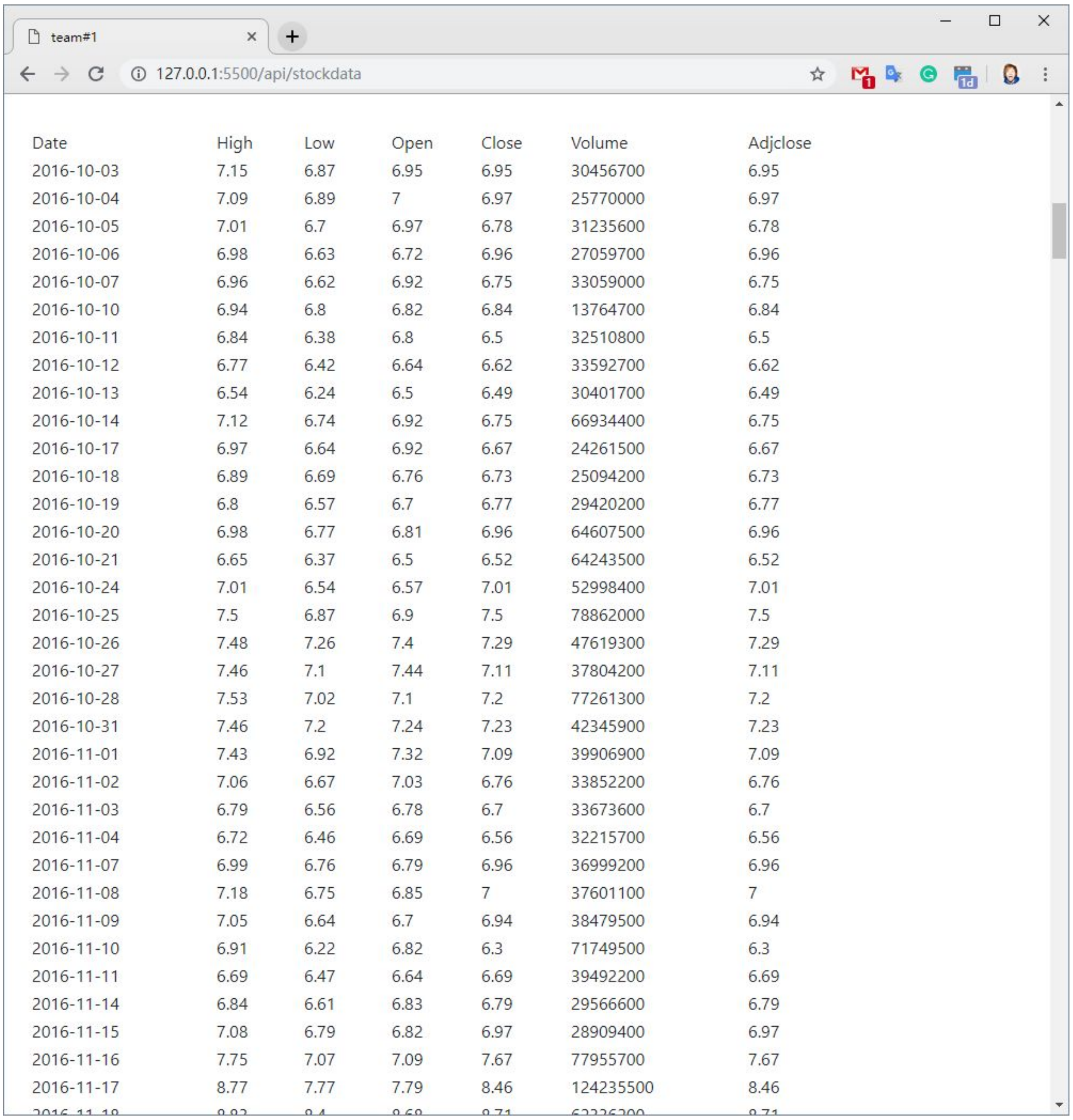
Time: 2019-05-03 18:40:47

If the user requests to access historical data of a stock, then the user could enter a start date and an end date in the input field, in a format of Year-month-day. The user also needs to enter the stock symbol to specify which stock he/she would like to look into.

After clicking at the button says Submit, our website will show a time plot of the close price and another time plot of the volume of the chosen stock in the chosen time period.



Besides the time plots, our website also shows textual stock information, from the start date to the end date chosen by the user.



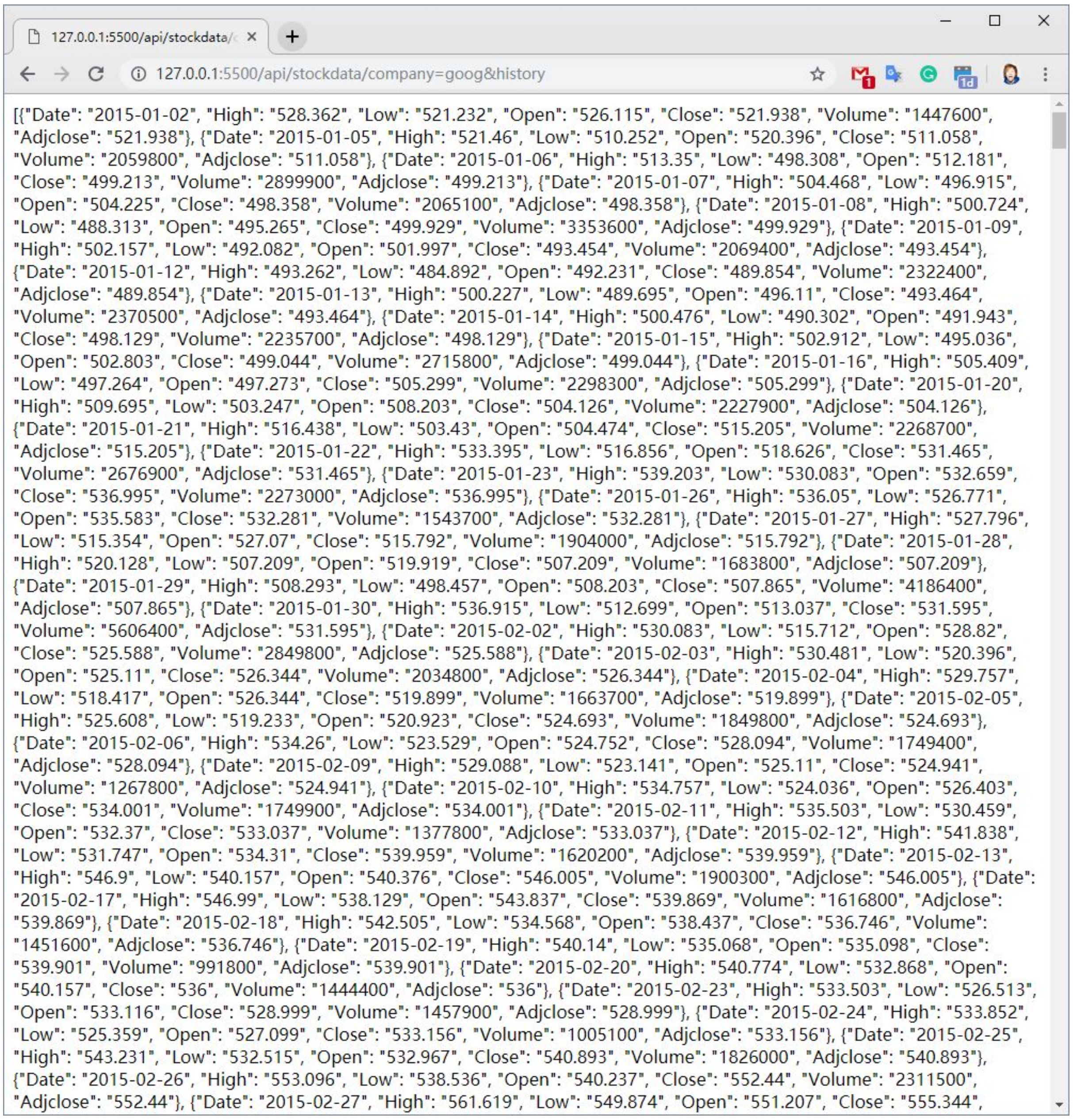
team#1

127.0.0.1:5500/api/stockdata

Date	High	Low	Open	Close	Volume	Adjclose
2016-10-03	7.15	6.87	6.95	6.95	30456700	6.95
2016-10-04	7.09	6.89	7	6.97	25770000	6.97
2016-10-05	7.01	6.7	6.97	6.78	31235600	6.78
2016-10-06	6.98	6.63	6.72	6.96	27059700	6.96
2016-10-07	6.96	6.62	6.92	6.75	33059000	6.75
2016-10-10	6.94	6.8	6.82	6.84	13764700	6.84
2016-10-11	6.84	6.38	6.8	6.5	32510800	6.5
2016-10-12	6.77	6.42	6.64	6.62	33592700	6.62
2016-10-13	6.54	6.24	6.5	6.49	30401700	6.49
2016-10-14	7.12	6.74	6.92	6.75	66934400	6.75
2016-10-17	6.97	6.64	6.92	6.67	24261500	6.67
2016-10-18	6.89	6.69	6.76	6.73	25094200	6.73
2016-10-19	6.8	6.57	6.7	6.77	29420200	6.77
2016-10-20	6.98	6.77	6.81	6.96	64607500	6.96
2016-10-21	6.65	6.37	6.5	6.52	64243500	6.52
2016-10-24	7.01	6.54	6.57	7.01	52998400	7.01
2016-10-25	7.5	6.87	6.9	7.5	78862000	7.5
2016-10-26	7.48	7.26	7.4	7.29	47619300	7.29
2016-10-27	7.46	7.1	7.44	7.11	37804200	7.11
2016-10-28	7.53	7.02	7.1	7.2	77261300	7.2
2016-10-31	7.46	7.2	7.24	7.23	42345900	7.23
2016-11-01	7.43	6.92	7.32	7.09	39906900	7.09
2016-11-02	7.06	6.67	7.03	6.76	33852200	6.76
2016-11-03	6.79	6.56	6.78	6.7	33673600	6.7
2016-11-04	6.72	6.46	6.69	6.56	32215700	6.56
2016-11-07	6.99	6.76	6.79	6.96	36999200	6.96
2016-11-08	7.18	6.75	6.85	7	37601100	7
2016-11-09	7.05	6.64	6.7	6.94	38479500	6.94
2016-11-10	6.91	6.22	6.82	6.3	71749500	6.3
2016-11-11	6.69	6.47	6.64	6.69	39492200	6.69
2016-11-14	6.84	6.61	6.83	6.79	29566600	6.79
2016-11-15	7.08	6.79	6.82	6.97	28909400	6.97
2016-11-16	7.75	7.07	7.09	7.67	77955700	7.67
2016-11-17	8.77	7.77	7.79	8.46	124235500	8.46

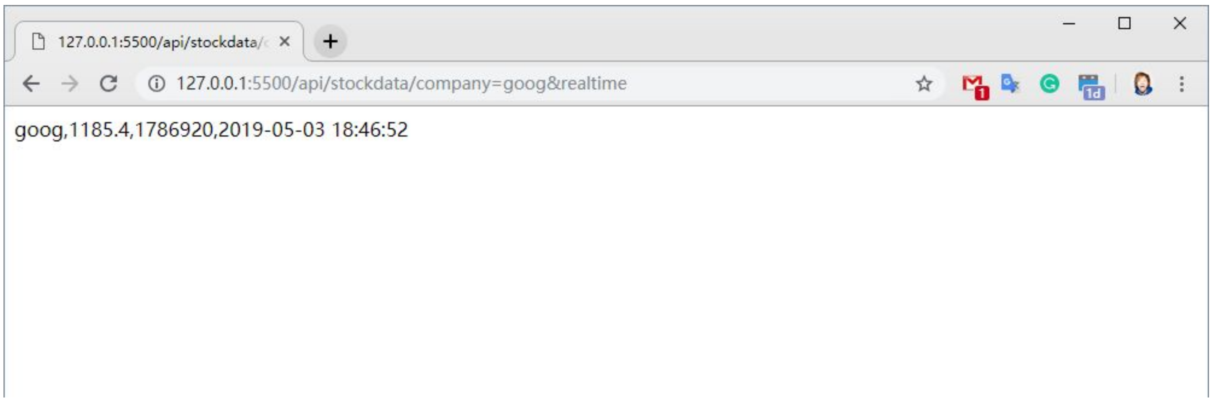
When the user type in an URL in a browser in a form of:

localhoset:port/api/stockdata/company=<symbol>&history, our website will return historical data in the format of JSON. For example, if a user type in 127.0.0.1:5500/api/stockdata/company=goog&history, our web service will provide historical data of GOOG in the format of JSON as the below figure shows.


127.0.0.1:5500/api/stockdata/c
127.0.0.1:5500/api/stockdata/company=goog&history

```
[{"Date": "2015-01-02", "High": "528.362", "Low": "521.232", "Open": "526.115", "Close": "521.938", "Volume": "1447600",
"Adjclose": "521.938"}, {"Date": "2015-01-05", "High": "521.46", "Low": "510.252", "Open": "520.396", "Close": "511.058",
"Volume": "2059800", "Adjclose": "511.058"}, {"Date": "2015-01-06", "High": "513.35", "Low": "498.308", "Open": "512.181",
"Close": "499.213", "Volume": "2899900", "Adjclose": "499.213"}, {"Date": "2015-01-07", "High": "504.468", "Low": "496.915",
"Open": "504.225", "Close": "498.358", "Volume": "2065100", "Adjclose": "498.358"}, {"Date": "2015-01-08", "High": "500.724",
"Low": "488.313", "Open": "495.265", "Close": "499.929", "Volume": "3353600", "Adjclose": "499.929"}, {"Date": "2015-01-09",
"High": "502.157", "Low": "492.082", "Open": "501.997", "Close": "493.454", "Volume": "2069400", "Adjclose": "493.454"},
{"Date": "2015-01-12", "High": "493.262", "Low": "484.892", "Open": "492.231", "Close": "489.854", "Volume": "2322400",
"Adjclose": "489.854"}, {"Date": "2015-01-13", "High": "500.227", "Low": "489.695", "Open": "496.11", "Close": "493.464",
"Volume": "2370500", "Adjclose": "493.464"}, {"Date": "2015-01-14", "High": "500.476", "Low": "490.302", "Open": "491.943",
"Close": "498.129", "Volume": "2235700", "Adjclose": "498.129"}, {"Date": "2015-01-15", "High": "502.912", "Low": "495.036",
"Open": "502.803", "Close": "499.044", "Volume": "2715800", "Adjclose": "499.044"}, {"Date": "2015-01-16", "High": "505.409",
"Low": "497.264", "Open": "497.273", "Close": "505.299", "Volume": "2298300", "Adjclose": "505.299"}, {"Date": "2015-01-20",
"High": "509.695", "Low": "503.247", "Open": "508.203", "Close": "504.126", "Volume": "2227900", "Adjclose": "504.126"},
{"Date": "2015-01-21", "High": "516.438", "Low": "503.43", "Open": "504.474", "Close": "515.205", "Volume": "2268700",
"Adjclose": "515.205"}, {"Date": "2015-01-22", "High": "533.395", "Low": "516.856", "Open": "518.626", "Close": "531.465",
"Volume": "2676900", "Adjclose": "531.465"}, {"Date": "2015-01-23", "High": "539.203", "Low": "530.083", "Open": "532.659",
"Close": "536.995", "Volume": "2273000", "Adjclose": "536.995"}, {"Date": "2015-01-26", "High": "536.05", "Low": "526.771",
"Open": "535.583", "Close": "532.281", "Volume": "1543700", "Adjclose": "532.281"}, {"Date": "2015-01-27", "High": "527.796",
"Low": "515.354", "Open": "527.07", "Close": "515.792", "Volume": "1904000", "Adjclose": "515.792"}, {"Date": "2015-01-28",
"High": "520.128", "Low": "507.209", "Open": "519.919", "Close": "507.209", "Volume": "1683800", "Adjclose": "507.209"},
{"Date": "2015-01-29", "High": "508.293", "Low": "498.457", "Open": "508.203", "Close": "507.865", "Volume": "4186400",
"Adjclose": "507.865"}, {"Date": "2015-01-30", "High": "536.915", "Low": "512.699", "Open": "513.037", "Close": "531.595",
"Volume": "5606400", "Adjclose": "531.595"}, {"Date": "2015-02-02", "High": "530.083", "Low": "515.712", "Open": "528.82",
"Close": "525.588", "Volume": "2849800", "Adjclose": "525.588"}, {"Date": "2015-02-03", "High": "530.481", "Low": "520.396",
"Open": "525.11", "Close": "526.344", "Volume": "2034800", "Adjclose": "526.344"}, {"Date": "2015-02-04", "High": "529.757",
"Low": "518.417", "Open": "526.344", "Close": "519.899", "Volume": "1663700", "Adjclose": "519.899"}, {"Date": "2015-02-05",
"High": "525.608", "Low": "519.233", "Open": "520.923", "Close": "524.693", "Volume": "1849800", "Adjclose": "524.693"},
{"Date": "2015-02-06", "High": "534.26", "Low": "523.529", "Open": "524.752", "Close": "528.094", "Volume": "1749400",
"Adjclose": "528.094"}, {"Date": "2015-02-09", "High": "529.088", "Low": "523.141", "Open": "525.11", "Close": "524.941",
"Volume": "1267800", "Adjclose": "524.941"}, {"Date": "2015-02-10", "High": "534.757", "Low": "524.036", "Open": "526.403",
"Close": "534.001", "Volume": "1749900", "Adjclose": "534.001"}, {"Date": "2015-02-11", "High": "535.503", "Low": "530.459",
"Open": "532.37", "Close": "533.037", "Volume": "1377800", "Adjclose": "533.037"}, {"Date": "2015-02-12", "High": "541.838",
"Low": "531.747", "Open": "534.31", "Close": "539.959", "Volume": "1620200", "Adjclose": "539.959"}, {"Date": "2015-02-13",
"High": "546.9", "Low": "540.157", "Open": "540.376", "Close": "546.005", "Volume": "1900300", "Adjclose": "546.005"}, {"Date":
"2015-02-17", "High": "546.99", "Low": "538.129", "Open": "543.837", "Close": "539.869", "Volume": "1616800", "Adjclose":
"539.869"}, {"Date": "2015-02-18", "High": "542.505", "Low": "534.568", "Open": "538.437", "Close": "536.746", "Volume":
"1451600", "Adjclose": "536.746"}, {"Date": "2015-02-19", "High": "540.14", "Low": "535.068", "Open": "535.098", "Close":
"539.901", "Volume": "991800", "Adjclose": "539.901"}, {"Date": "2015-02-20", "High": "540.774", "Low": "532.868", "Open":
"540.157", "Close": "536", "Volume": "1444400", "Adjclose": "536"}, {"Date": "2015-02-23", "High": "533.503", "Low": "526.513",
"Open": "533.116", "Close": "528.999", "Volume": "1457900", "Adjclose": "528.999"}, {"Date": "2015-02-24", "High": "533.852",
"Low": "525.359", "Open": "527.099", "Close": "533.156", "Volume": "1005100", "Adjclose": "533.156"}, {"Date": "2015-02-25",
"High": "543.231", "Low": "532.515", "Open": "532.967", "Close": "540.893", "Volume": "1826000", "Adjclose": "540.893"},
{"Date": "2015-02-26", "High": "553.096", "Low": "538.536", "Open": "540.237", "Close": "552.44", "Volume": "2311500",
"Adjclose": "552.44"}, {"Date": "2015-02-27", "High": "561.619", "Low": "549.874", "Open": "551.207", "Close": "555.344",
```

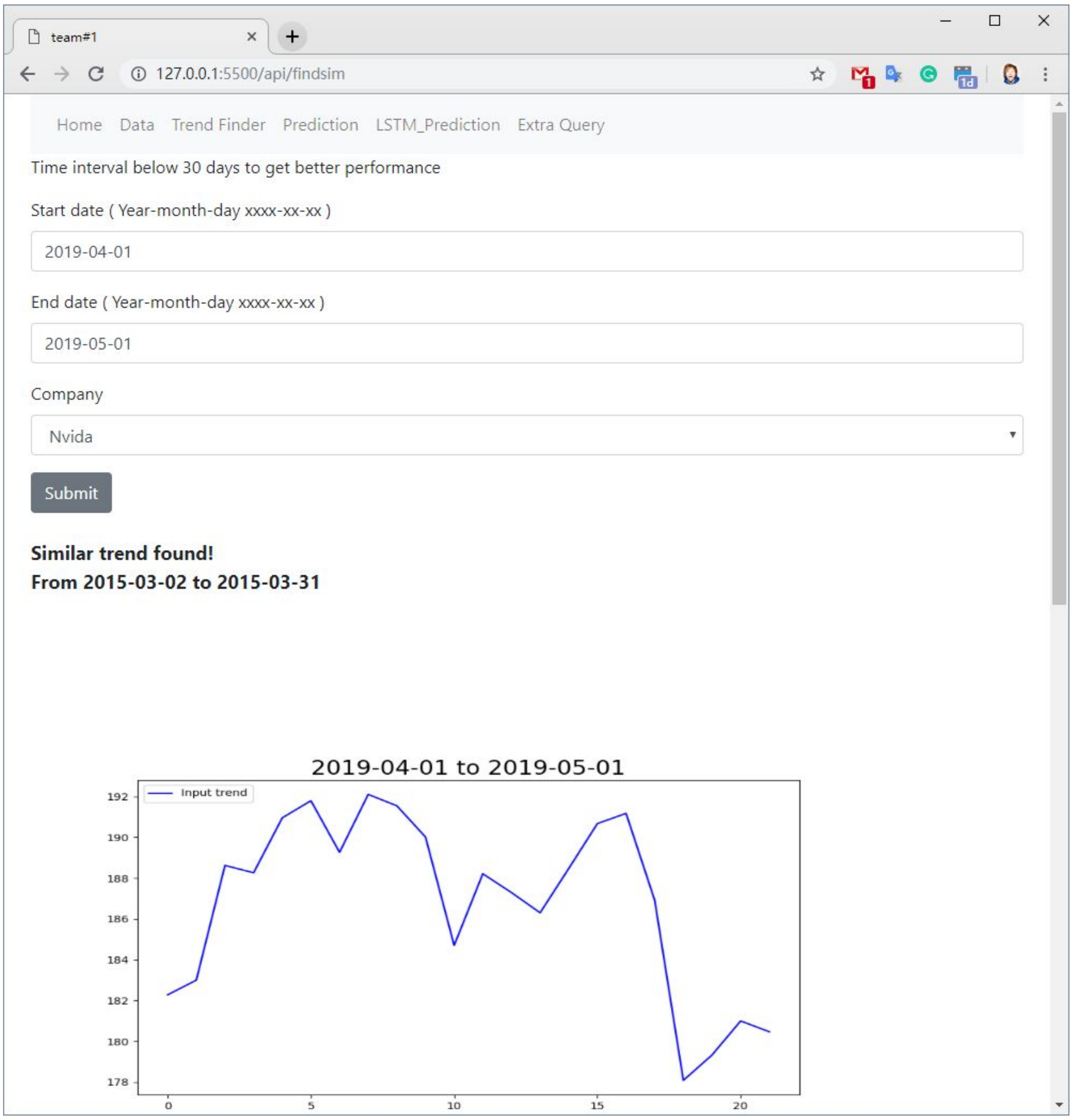
Similarly, the user can enters a stock symbol and request its real-time information through an URL, which is in the form of: localhoset:port/api/stockdata/company=<symbol>&realtime. For example, if a user type in 127.0.0.1:5500/api/stockdata/company=goog&realtime, our website will shows the stock symbol, the price per share, the total volume and the date and time.


127.0.0.1:5500/api/stockdata/c
127.0.0.1:5500/api/stockdata/company=goog&realtime

```
goog,1185.4,1786920,2019-05-03 18:46:52
```

7.3 Trend Finder

By a mouse click on the Trend Finder, our user can enter the functional page where they can find time intervals in the historical data with a similar trend of a given time period. The user will be informed that the time interval should better be below 30 days.



team#1

127.0.0.1:5500/api/findsim

Home Data Trend Finder Prediction LSTM_Prediction Extra Query

Time interval below 30 days to get better performance

Start date (Year-month-day xxxx-xx-xx)

2019-04-01

End date (Year-month-day xxxx-xx-xx)

2019-05-01

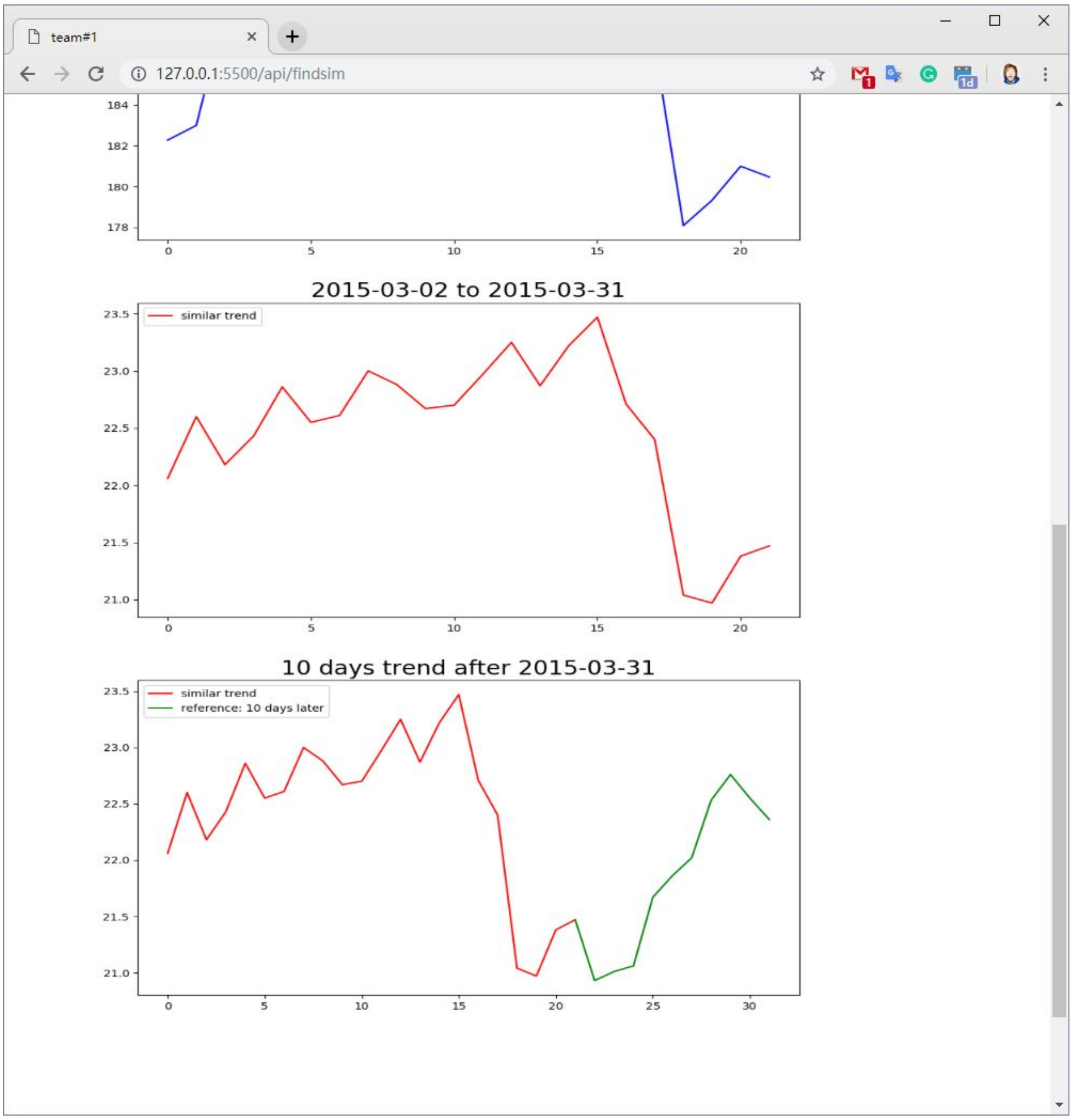
Company

Nvida

Submit

Similar trend found!
From 2015-03-02 to 2015-03-31

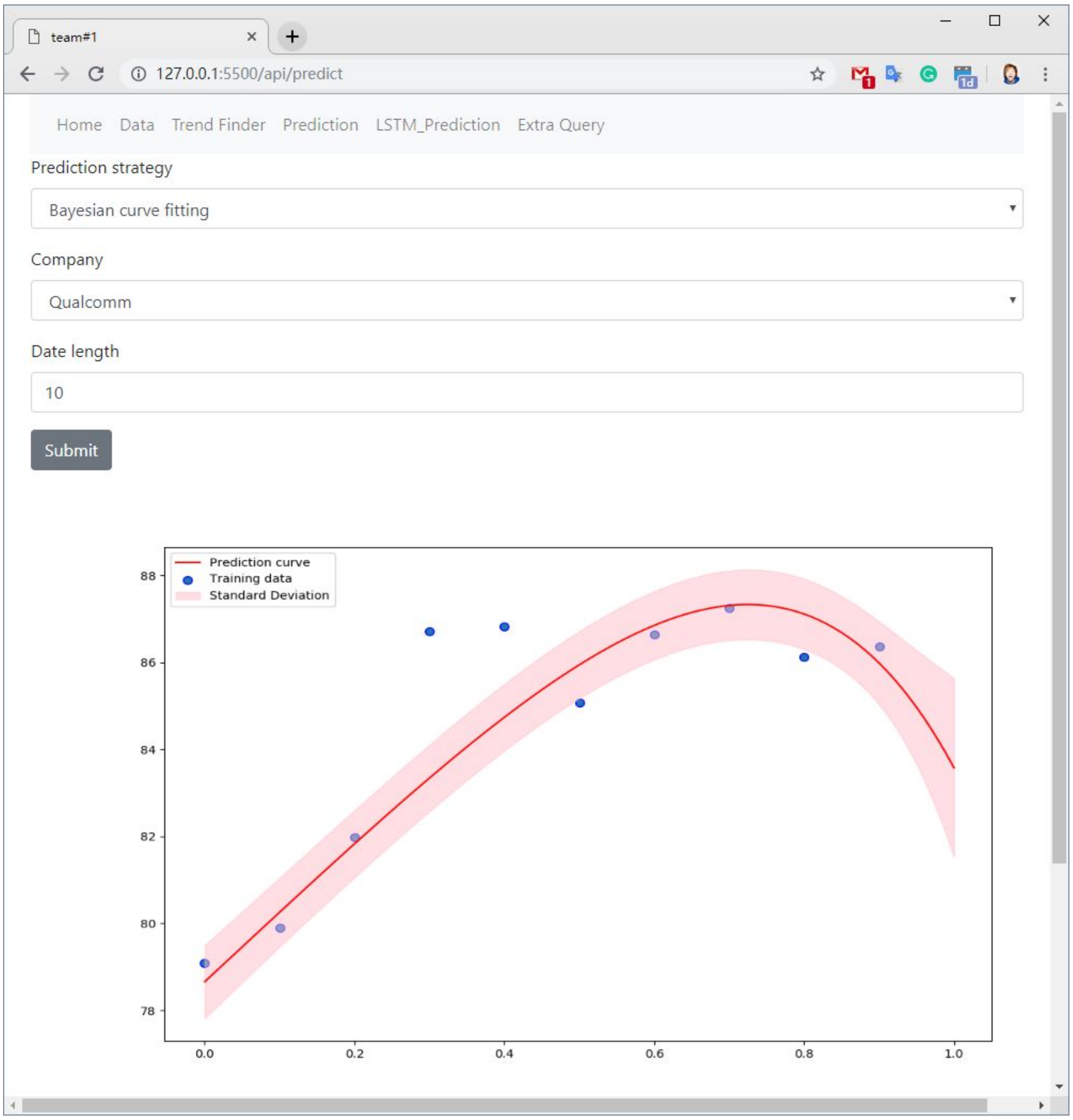
Once our users enter this functional page, they can request to find time intervals with a similar trend as the desired time interval. For example, here we enter the stock symbol NVIDIA into the company field and enter the start date and the end date in the format of Year-month-day. After a click on the submit button, if similar trends are found in the historical data, our website will say similar trend found and display the most correlated time intervals.



After we find the most correlated time interval for our user, we also display the 10 days trend after that similar trend and we provide this as a reference to our user.

7.4 Prediction

Clicking at the Prediction bar in the navigation bars, our users can enter the functional page for prediction.


team#1

127.0.0.1:5500/api/predict

Prediction strategy

Bayesian curve fitting

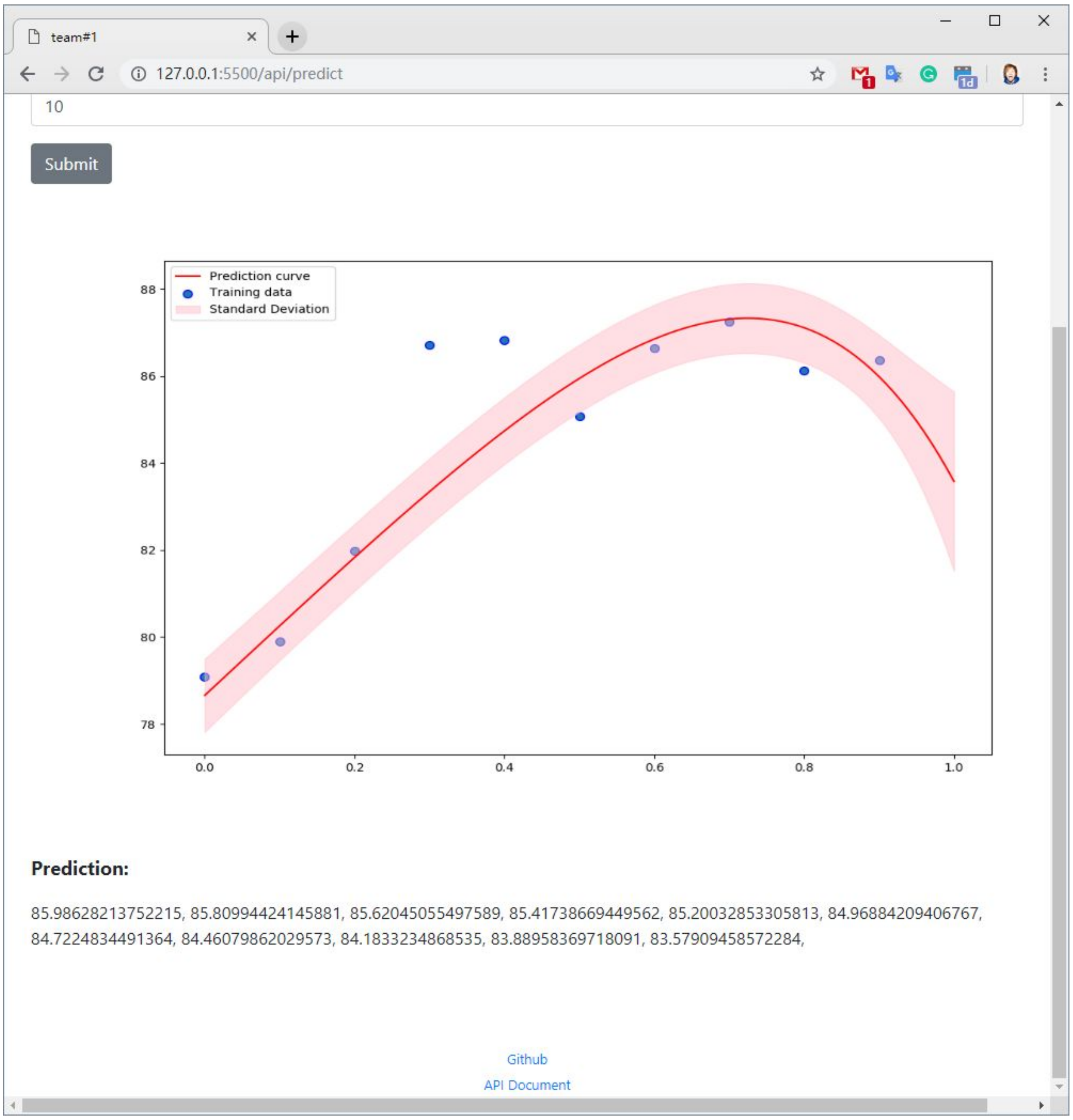
Company

Qualcomm

Date length

10

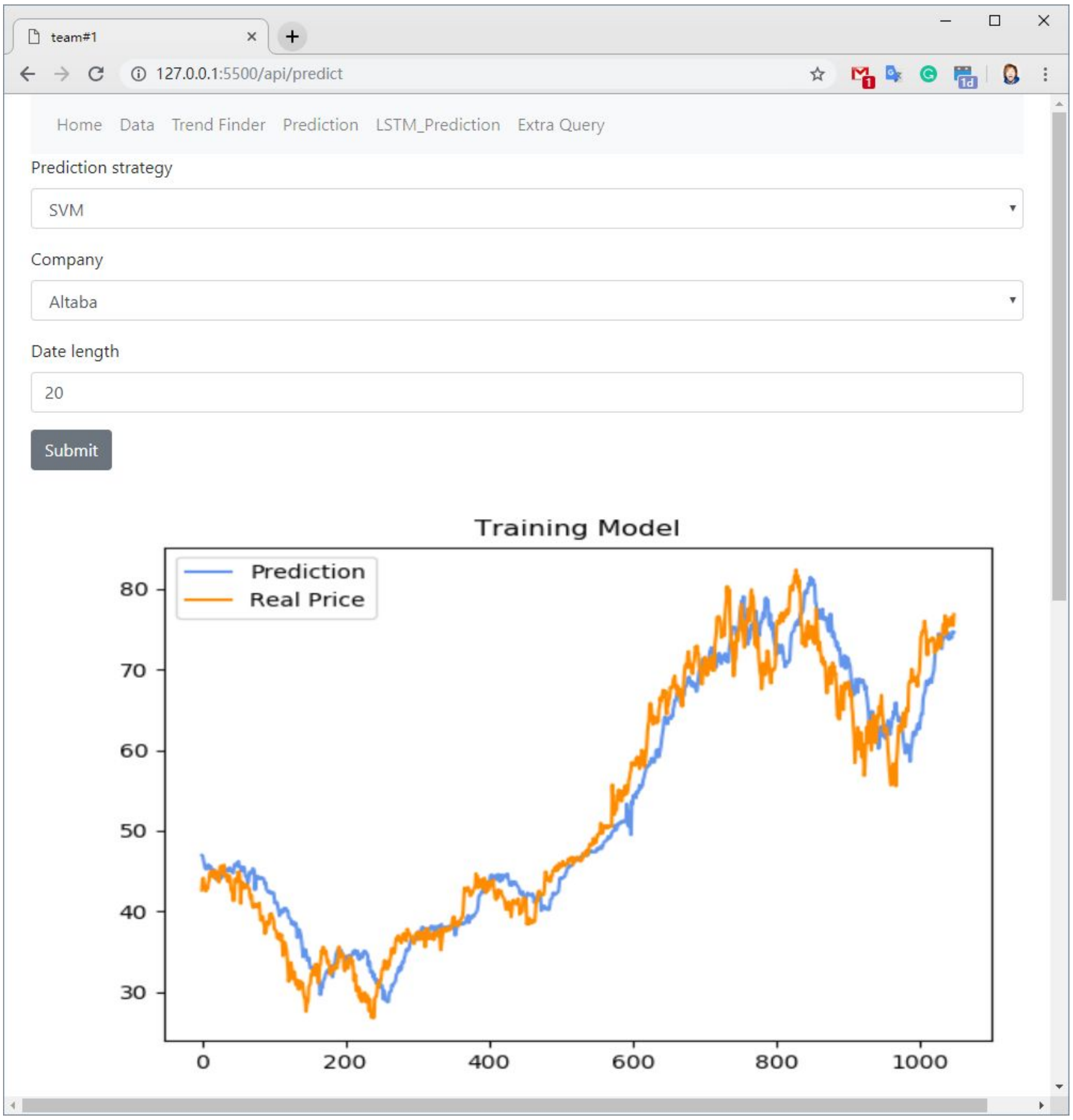
In the field of prediction strategy, the user can choose which strategy they would like to choose for the prediction. Here our users can choose either Bayesian curve fitting or SVM. Also, our users need to enter the stock symbol to specify which stock they would like to look into. Moreover, they can enter a positive integer to specify their desired data length using for prediction. For example, here we choose Bayesian curve fitting as our prediction strategy. After that, we enter Qualcomm in the company field and enter 10 as the data length.



Prediction:

85.98628213752215, 85.80994424145881, 85.62045055497589, 85.41738669449562, 85.20032853305813, 84.96884209406767, 84.7224834491364, 84.46079862029573, 84.1833234868535, 83.88958369718091, 83.57909458572284,

After a mouse click on the submit button, our website will display a plot showing the prediction curve, the training data, and the standard deviation. Besides, our website will also give a predictive numerical value for the stock price.



Prediction strategy

SVM

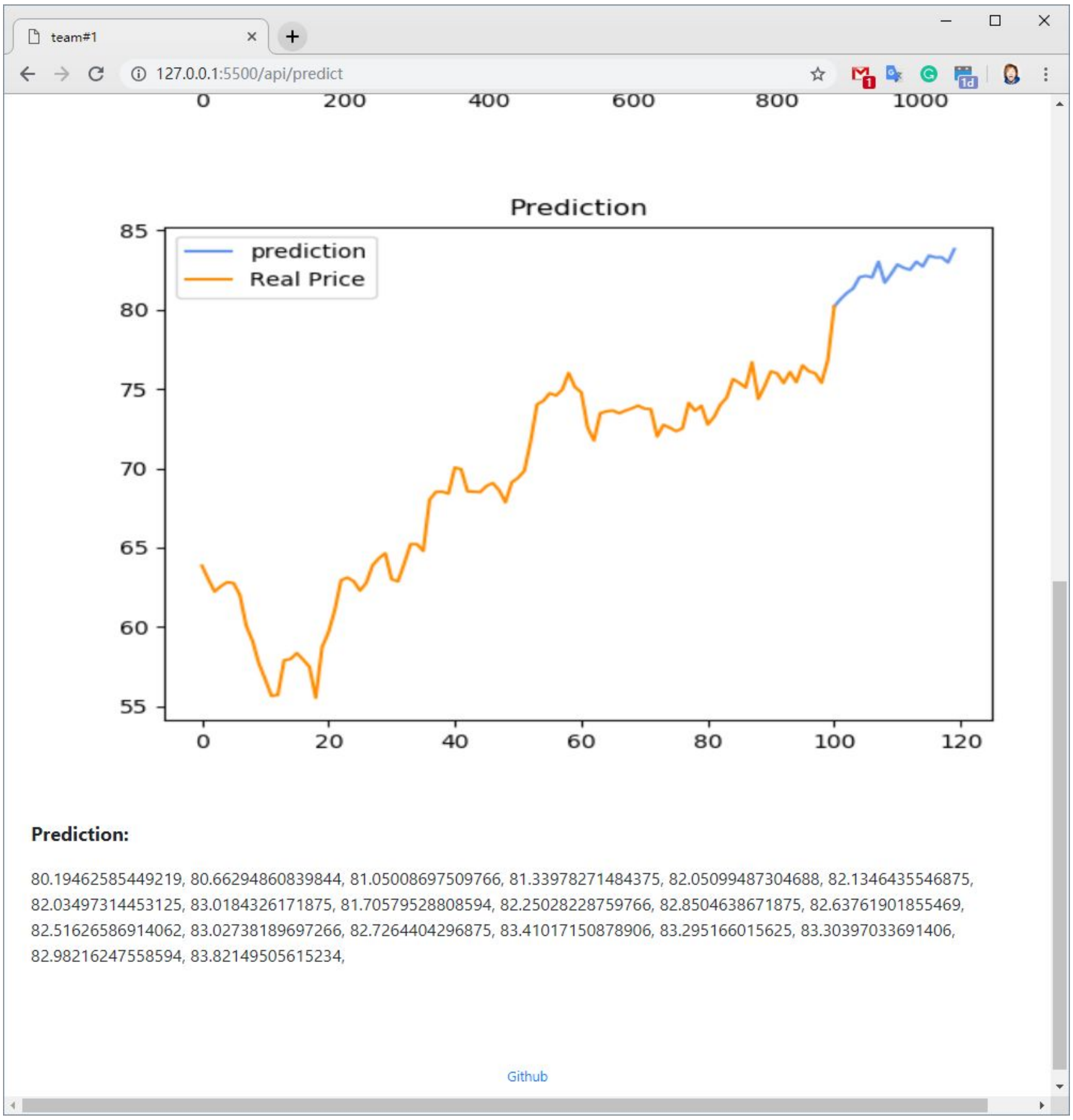
Company

Altaba

Date length

20

We could also choose SVM as our prediction strategy. For example, here we choose SVM in the prediction strategy field and enter Altaba in the company field and 20 for the data length.



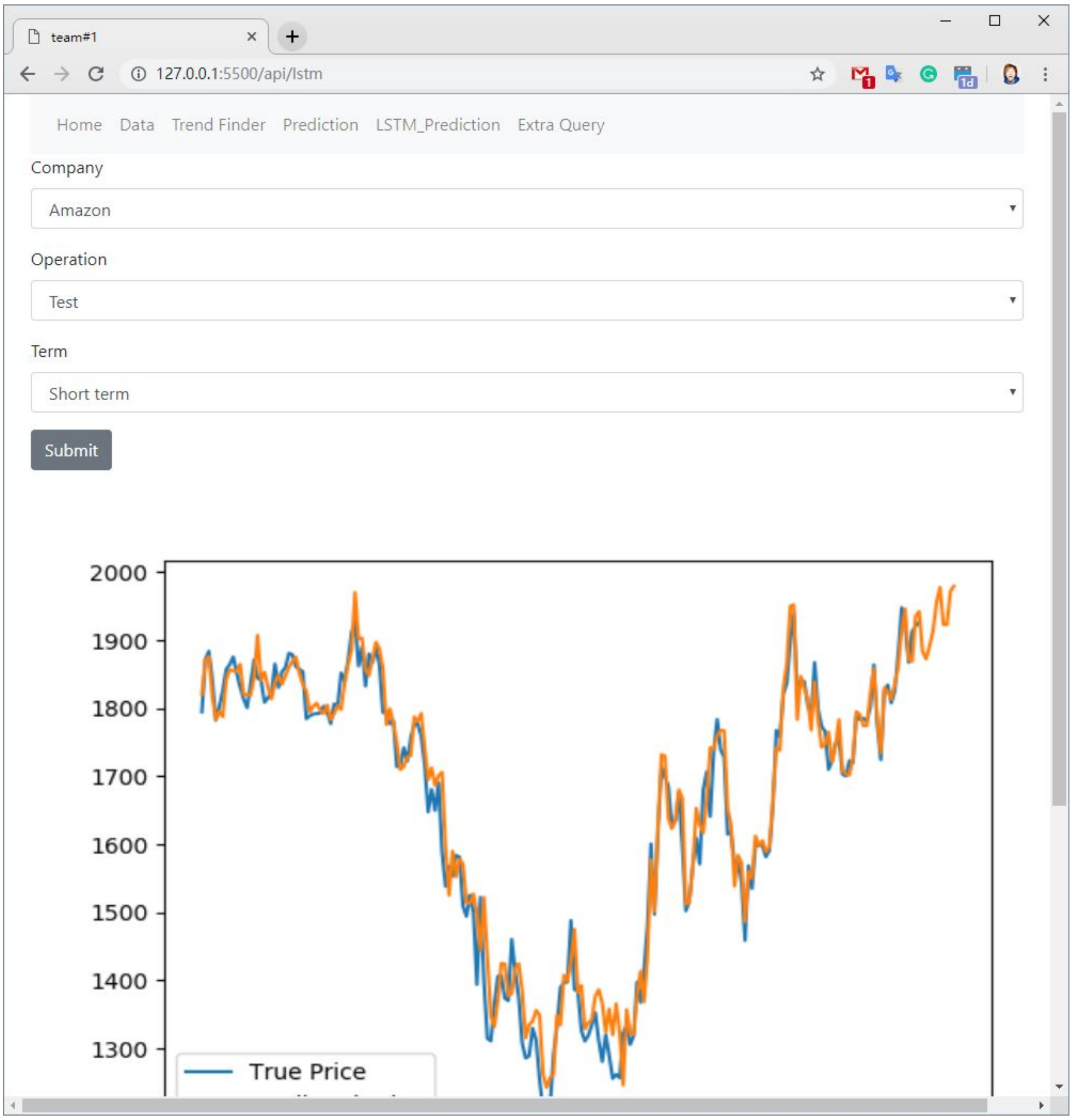
Prediction:

80.19462585449219, 80.66294860839844, 81.05008697509766, 81.33978271484375, 82.05099487304688, 82.1346435546875, 82.03497314453125, 83.0184326171875, 81.70579528808594, 82.25028228759766, 82.8504638671875, 82.63761901855469, 82.51626586914062, 83.02738189697266, 82.7264404296875, 83.41017150878906, 83.295166015625, 83.30397033691406, 82.98216247558594, 83.82149505615234,

Similarly, after a mouse click on the submit button, our website will display a plot showing training model, the true price and the predicted price. Besides, our website will also give a predictive numerical value for the stock price.

7.5 LSTM Prediction

Clicking on the LSTM_Prediction bar in the navigation bars, our users can enter the functional page for LSTM prediction.



Company

Amazon

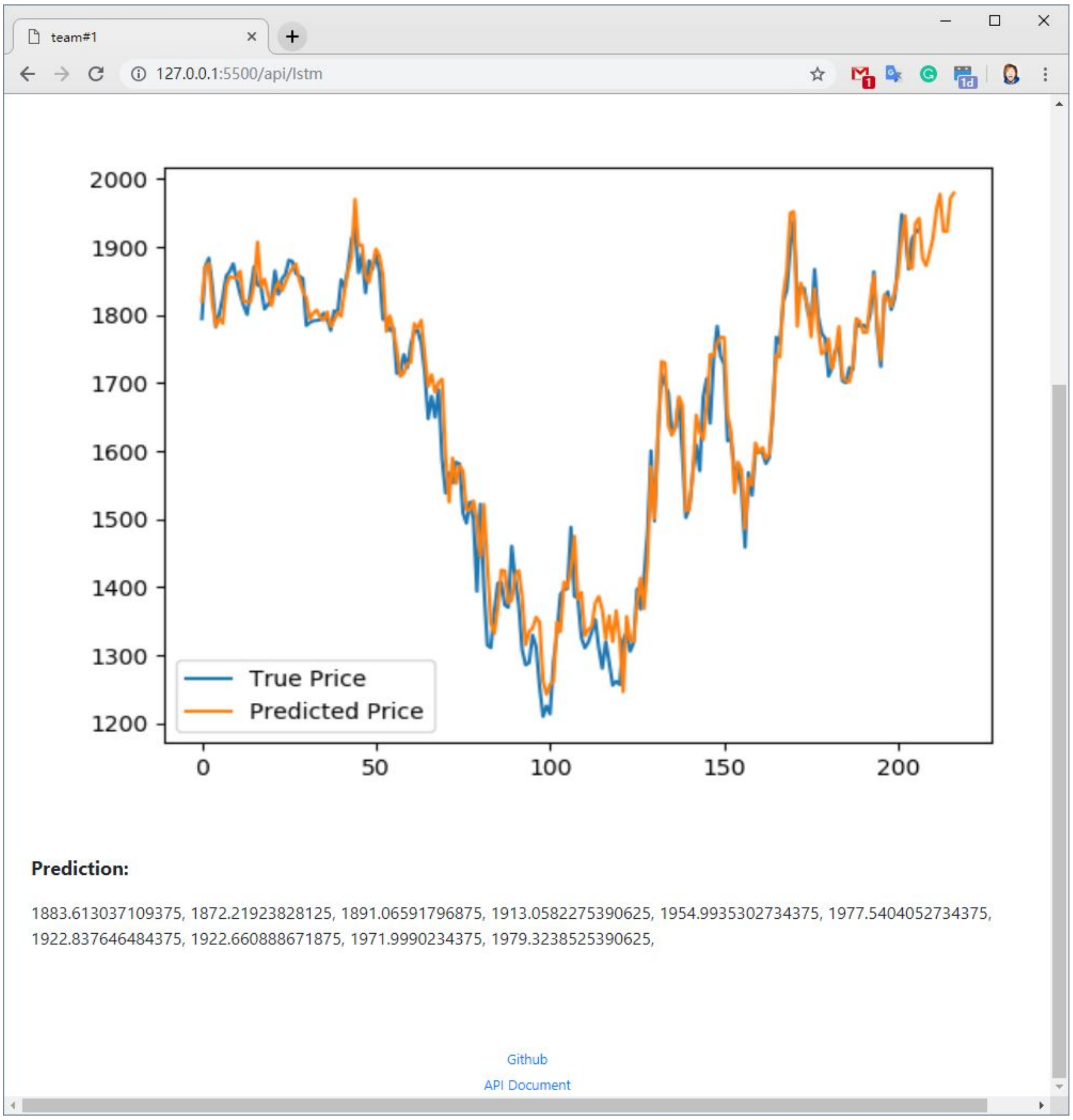
Operation

Test

Term

Short term

For this page, the user needs to specify the company he/she would like to look into, the operation, and the term, which can be either short term or long term.



Prediction:

```
1883.613037109375, 1872.21923828125, 1891.06591796875, 1913.0582275390625, 1954.9935302734375, 1977.5404052734375,
1922.837646484375, 1922.660888671875, 1971.9990234375, 1979.3238525390625,
```

Similarly to the prediction page, after a mouse click on the submit button, our website will display a plot showing the true price and the predicted price. Besides, our website will also give a predictive numerical value for the stock price.

7.6 Extra Query

In this part, we implement some extra queries for our program as bellows.

1. Show the list of all companies in the database along with their latest stock price (real time latest stock price).

2. Get the highest stock price of any company in the last ten days.

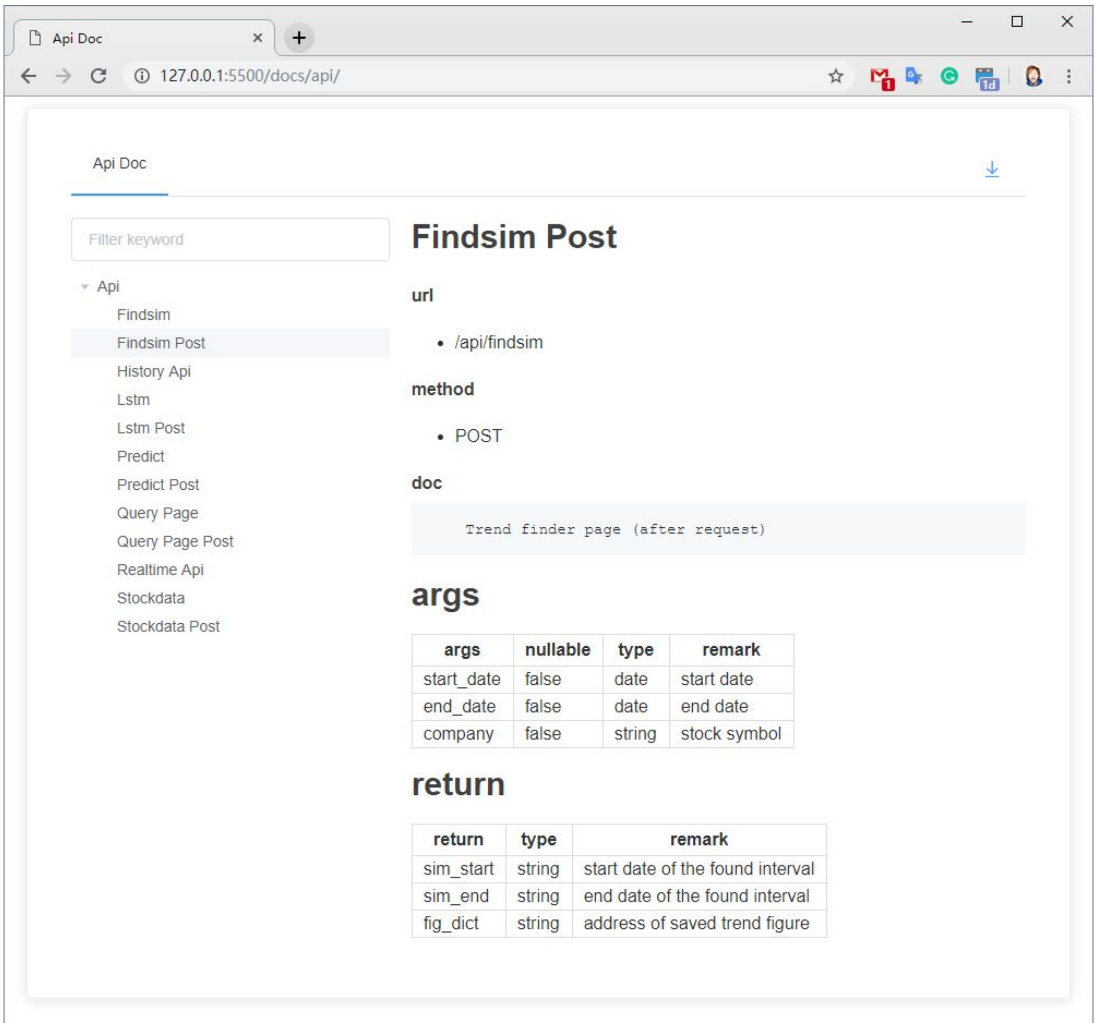
3. The average stock price of any company in the latest year.

4. The lowest stock price for any company in the latest year.

5. List the ids of companies along with their name who have the average stock price lesser than the lowest of any of the Selected Company in the latest one year.

7.7 API documentation

Our website can create API documentation as our users requested.



Findsim Post

url

- /api/findsim

method

- POST

doc

```
Trend finder page (after request)
```

args

args	nullable	type	remark
start_date	false	date	start date
end_date	false	date	end date
company	false	string	stock symbol

return

return	type	remark
sim_start	string	start date of the found interval
sim_end	string	end date of the found interval
fig_dict	string	address of saved trend figure

8. Future Work

Our potential services that will be provided in the future include:

1. Given a stock, suggest an action, such as “buy,” “sell,” “hold,” or “sit-out”;
2. Recommend a stock to buy, from all stocks that are being tracked, or from all in a given industry/sector;
3. Provide confidence level of the prediction;
4. Risk analysis to analyze “what if” scenarios.

9. Reference

Representational state transfer From Wikipedia, the free encyclopedia

https://en.wikipedia.org/wiki/Representational_state_transfer

Welcome to Flask

http://flask.pocoo.org/docs/1.0/

Designing a RESTful API with Python and Flask

https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask

The Flask Mega-Tutorial

https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world

LSTM

https://en.wikipedia.org/wiki/Long_short-term_memory

Keras document

https://keras.io/zh/

Matplotlib

https://matplotlib.org/

Polynomial Regression

https://en.wikipedia.org/wiki/Polynomial_regression

SVM Theory

https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-th
eory-f0812effc72