Stock Market Prediction

USING

MACHINE LEARNING

Akshay R	1ST14CS010
Aravind B	1ST14CS023
Arun Kumar	1ST14CS025
Ashok S	1ST14CS027

Under the guidance of

Dr. T John Peter

H.O.D, Dept. of CSE

Sambhram Institute of Technology

Department of Computer Science & Engineering

Agenda

- Abstract
- Introduction
- Existing Methods
- Proposed System
- System Design
- Implementation
- Results & Discussions
- Performance
- Future Enhancement
- Conclusion

Abstract

- Time series forecasting has been widely used to determine the future prices of stock, and the analysis and modelling of finance time series importantly guide investors' decisions and trades
- This work proposes an intelligent time series prediction system that uses sliding-window optimization for the purpose of predicting the stock prices
- The system has a graphical user interface and functions as a stand-alone application.
- The proposed model is a promising predictive technique for highly non-linear time series, whose patterns are difficult to capture by traditional models.

Introduction

- Financial markets are highly volatile and generate huge amounts of data daily
- It is the most popular financial market instrument and its value changes quickly
- Stock prices are predicted to determine the future value of companies' stock or other financial instruments that are marketed on financial exchanges
- However, the stock market is influenced by many factors such as political events, economic conditions and traders' expectation

What is Machine Learning

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed

More formally, it can defined as,

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E

Example: playing checkers.

*E = the **experience** of playing many games of checkers*

*T = the **task** of playing checkers.*

*P = the **probability** that the program will win the next game.*

Machine Learning in Stock Prediction

- The field of Machine Learning is vast and plays a key role in a wide range of critical applications.
- The concept of **Support Vector Machines (SVM)** have advanced features that are reflected in their good generalization capacity and fast computation.
- Predicting the stock market involves predicting the closing prices of a company's stock for any given number of days ahead.
- SVMs can be used to perform Linear Regression on previous stock data to predict the closing prices using Time series forecasting and other optimization algorithms

Existing Methods

- Time series forecasting consists of a research area designed to solve various problems, mainly in the financial area
- Support vector regression (SVR), a variant of the SVM, is typically used to solve nonlinear regression problems by constructing the input-output mapping function.
- The least squares support vector regression (LSSVR) algorithm is a further development of SVR and its use considerably reduces computational complexity and increases efficiency compared to standard SVR.
- The Firefly Algorithm (FA), which is a nature-inspired metaheuristic method, has recently performed extremely well in solving various optimization problems.

Disadvantages

- The existing system focuses on the stock price market in Taiwan, but does not generalize for other markets worldwide.
- The system does not allow the import of raw data directly
- The existing system cannot be used to analyze multi-variate time series
- Lastly, the system does not have a user-interface which can be distributed as a web app to users for personal use

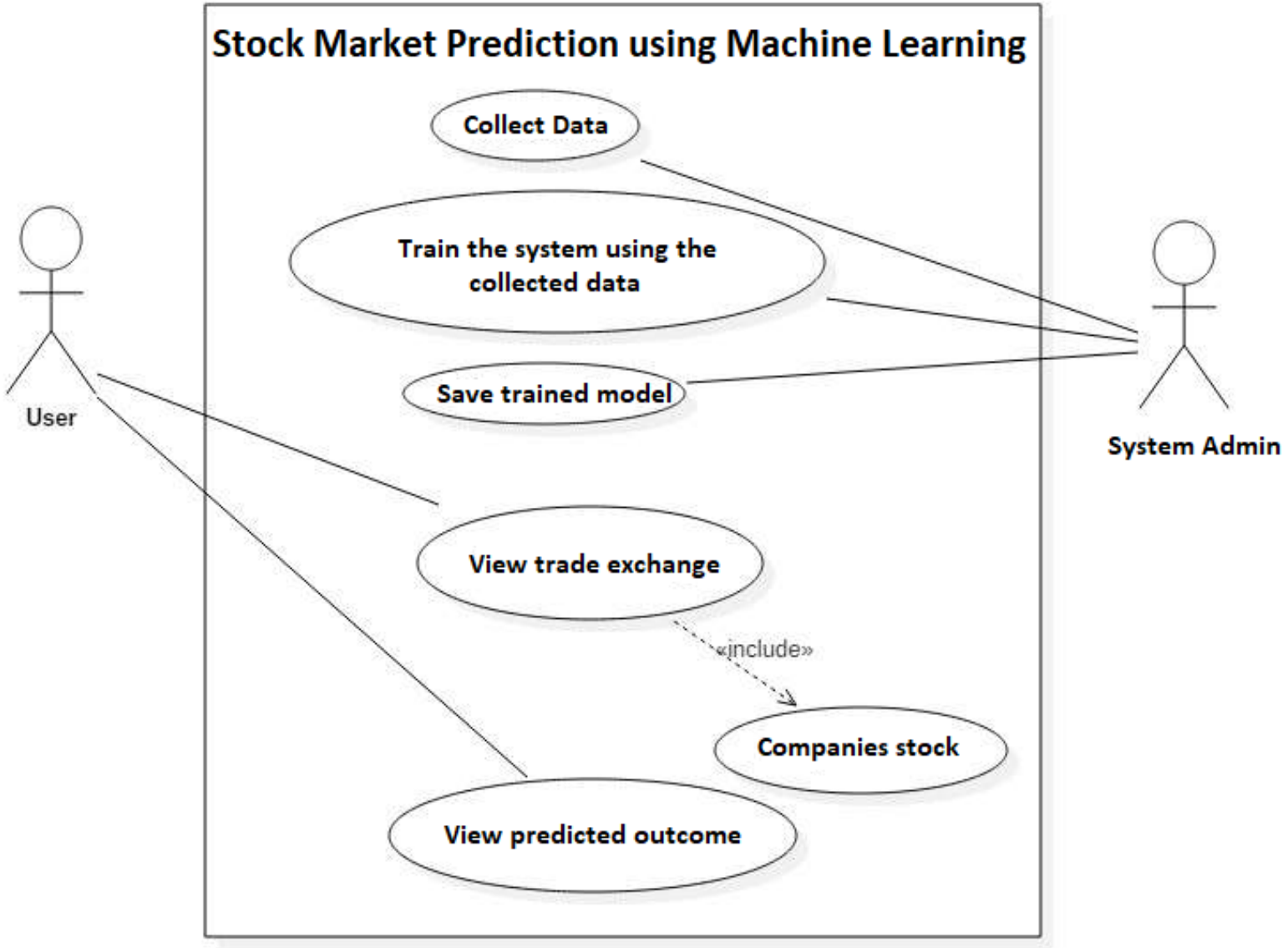
Proposed System

- To generalize the application of the existing system, our work uses the system to estimate other stocks in similar emerging markets and mature markets
- The system can be extended to analyze multivariate time series data and import raw dataset directly
- Profit can be maximized even when the corporate stock market is has lower value
- The development of a web-based application has been considered to improve the user-friendliness and usability of the expert system.

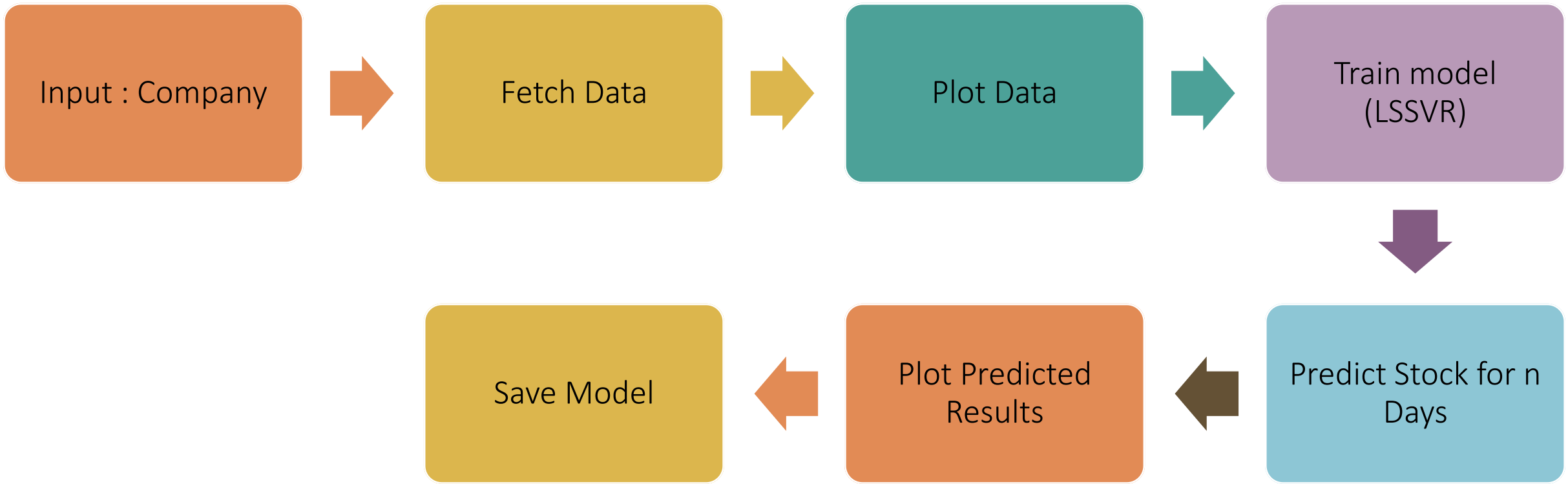
System Design

USE CASE DIAGRAM

1. Data is initially collected from online sources or the stock exchange
2. The data is then used to train the system
3. Trained model is saved
4. User views the trade exchange and stock of a company
5. Using the model, closing prices are predicted



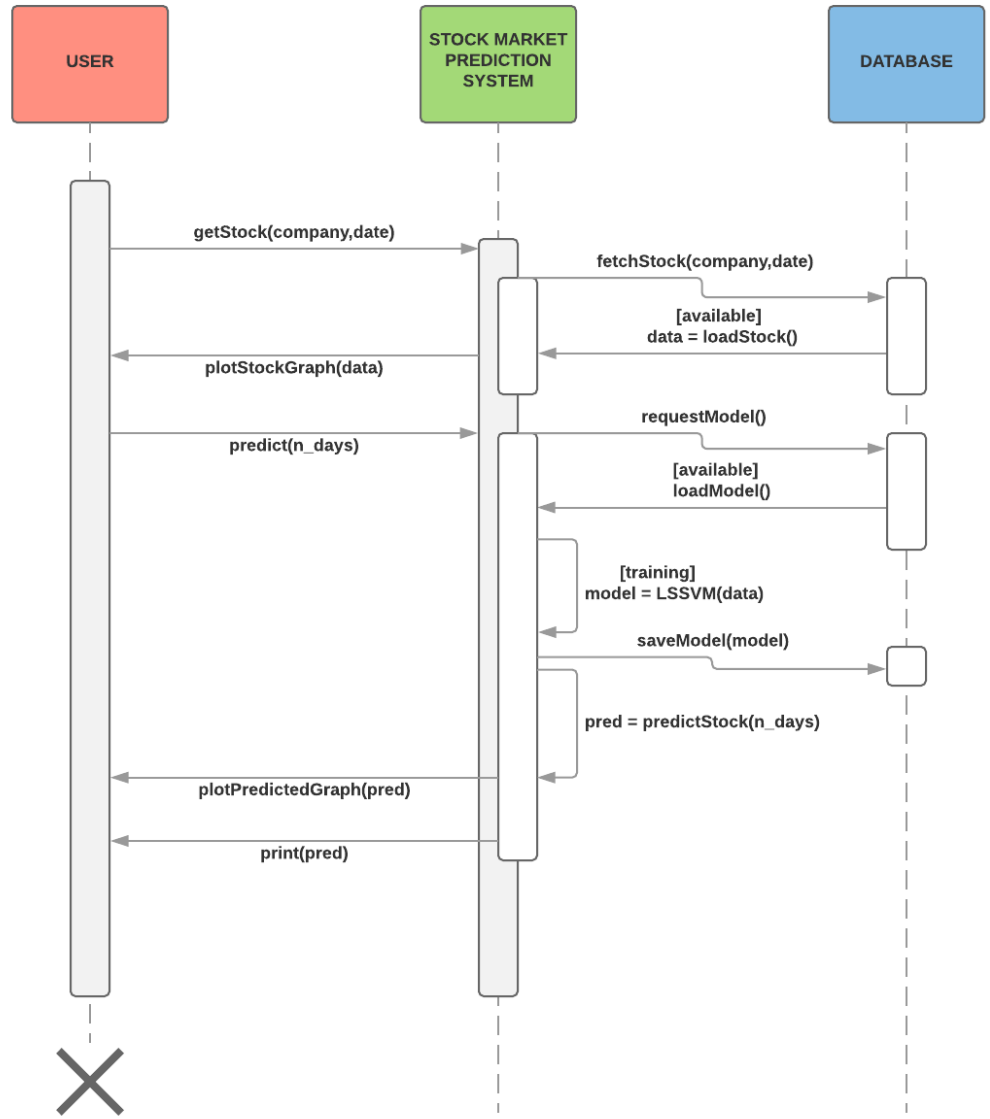
Data Flow Diagram



Sequence Diagram

SEQUENCE DIAGRAM

1. User visits the website/webapp
2. Previously saved model is loaded
3. User requests for a company's stock data
4. He requests for prediction to be made
5. The Stock Market Prediction System trains a model using the data from the database
6. The model is saved for further use and closing price is predicted
7. Result is displayed along with graph



Implementation

Algorithm

Algorithm : StockPrediction
Input : *COMP, D_RANGE, N_PRED* [company, date range, n-day predictions]
Output : A vector of predicted prices and graph, RESULTS

1. *data* ← fetch stock for COMP in date range D_RANGE
2. ***plot****(data)*
3. *train_data* ← ***slidingWindow***(*data*) *//perform sliding window operation on data*
4. RESULTS ← 0 *//set accuracy vector to zeroes*
5. **for each** day **in** *N_PRED* **:**
    1. *model* ← ***LSSVM***(*train_data*) *//pass the training data to LSSVM*
    2. *pred* ← ***predict***(*model*,*day*) *//predict the price given model and day*
    3. remove first item from *train_data* *//removing last item and adding predicted value*
    4. *train_data* ← add *pred* to *train_data* *//Last In, First Out.*
    5. RESULTS ← add *pred* to RESULTS
6. **end for**
7. ***print***(RESULTS)
8. ***plot***(RESULTS)
9. **return**

Implementation

LSSVR Formulation

Least Squares Support Vector Regression

The least squares version of the SVM classifier is obtained by reformulating the minimization problem as

$$\min J_2(w, b, e) = \frac{\mu}{2} w^T w + \frac{\zeta}{2} \sum_{i=1}^{N} e_{c,i}^2,$$

For the kernel function $K(x, x_i)$ one typically has the Radial Basis Function

$$K(x, x_i) = \exp\left(-\|x - x_i\|^2 / \sigma^2\right),$$

- The LSSVR algorithm is a further development of SVR by Suykens (2001) and involves equality instead of inequality constraints, and works with a least squares objective function.
- The LSSVR approach considerably reduces computational complexity and increases efficiency compared to standard SVM.
- LSSVR solves linear equations instead of a quadratic programming problem

Implementation

Sliding-Window

Algorithm : slidingWindow
Input : *data* [stock data]
Output : A data frame of a lagged stock data

1. LAG ← 1
2. *y* ← remove first LAG rows from data
3. reset row indices of *y*
4. *x* ← remove last LAG rows from data
5. *train* ← merge *(x,y)* into a dataframe
6. rename column name to *x* and *y*
7. **return** *train*

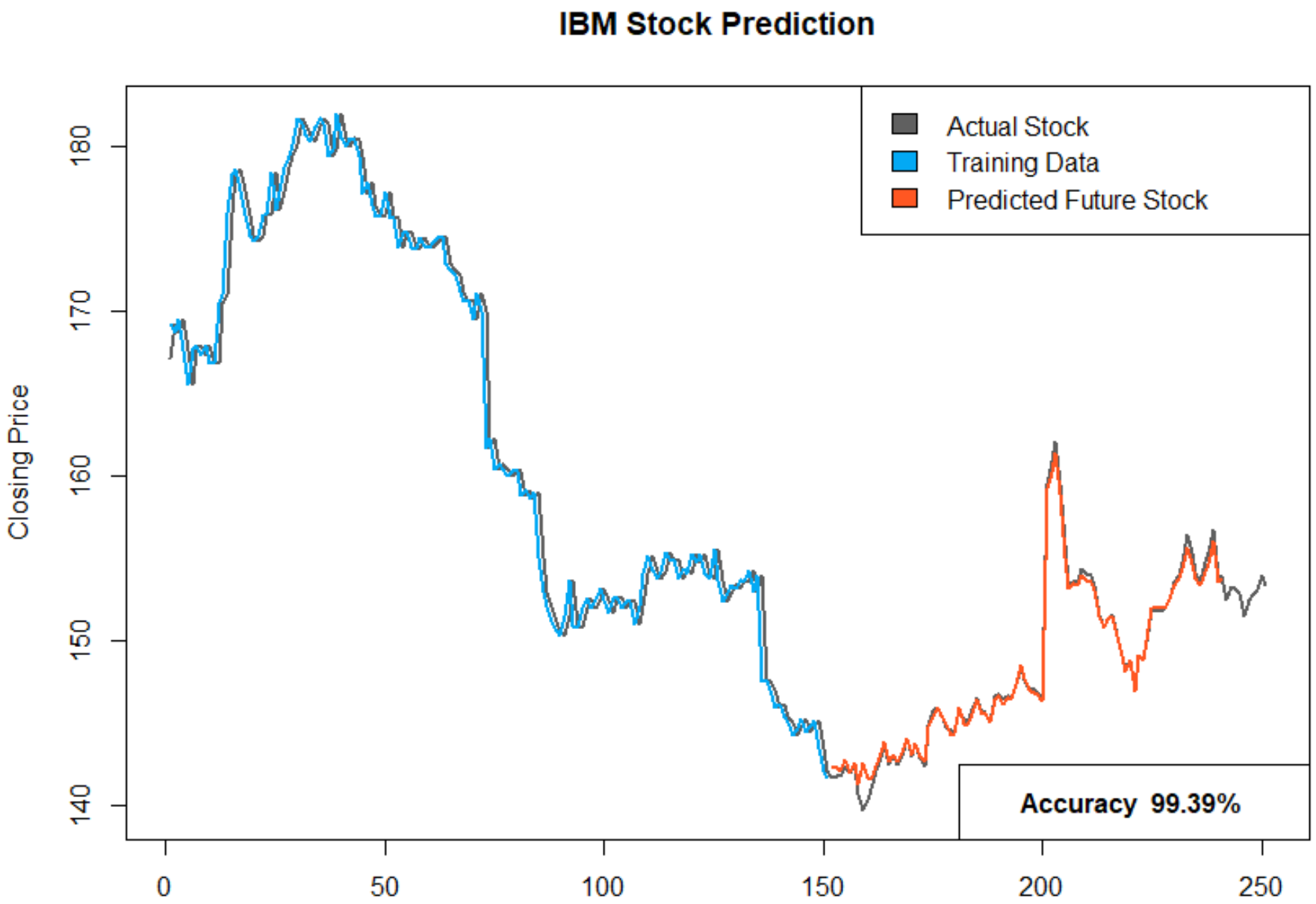
Using previous time steps as input variables and use the next time step as the output variable, we obtain the following table on the left

Time	Price
1	100
2	110
3	108
4	115
5	120

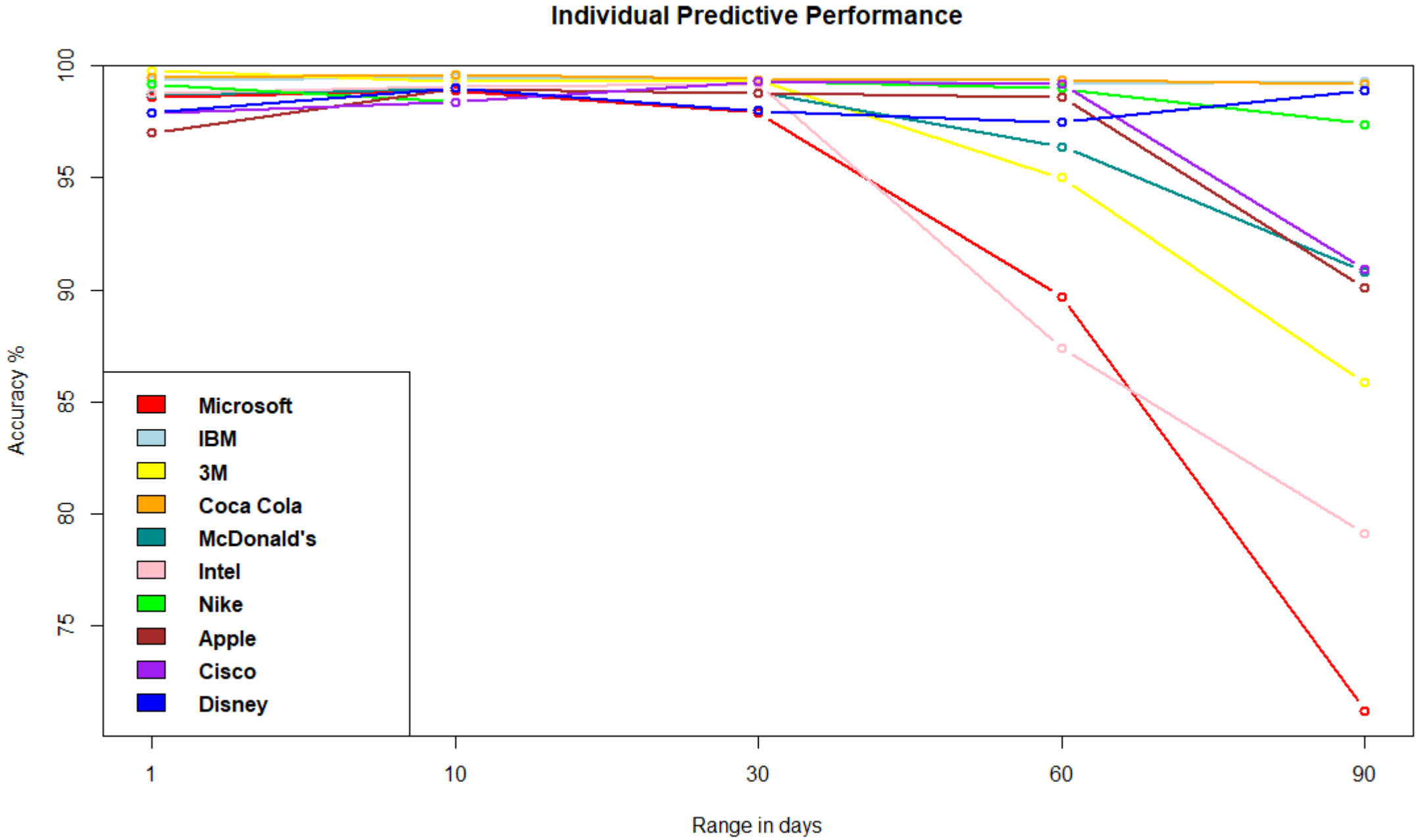
X	Y
?	100
100	110
110	108
108	115
115	120
120	?

Re-organizing the time series dataset this way, we obtain the table as above, on the right

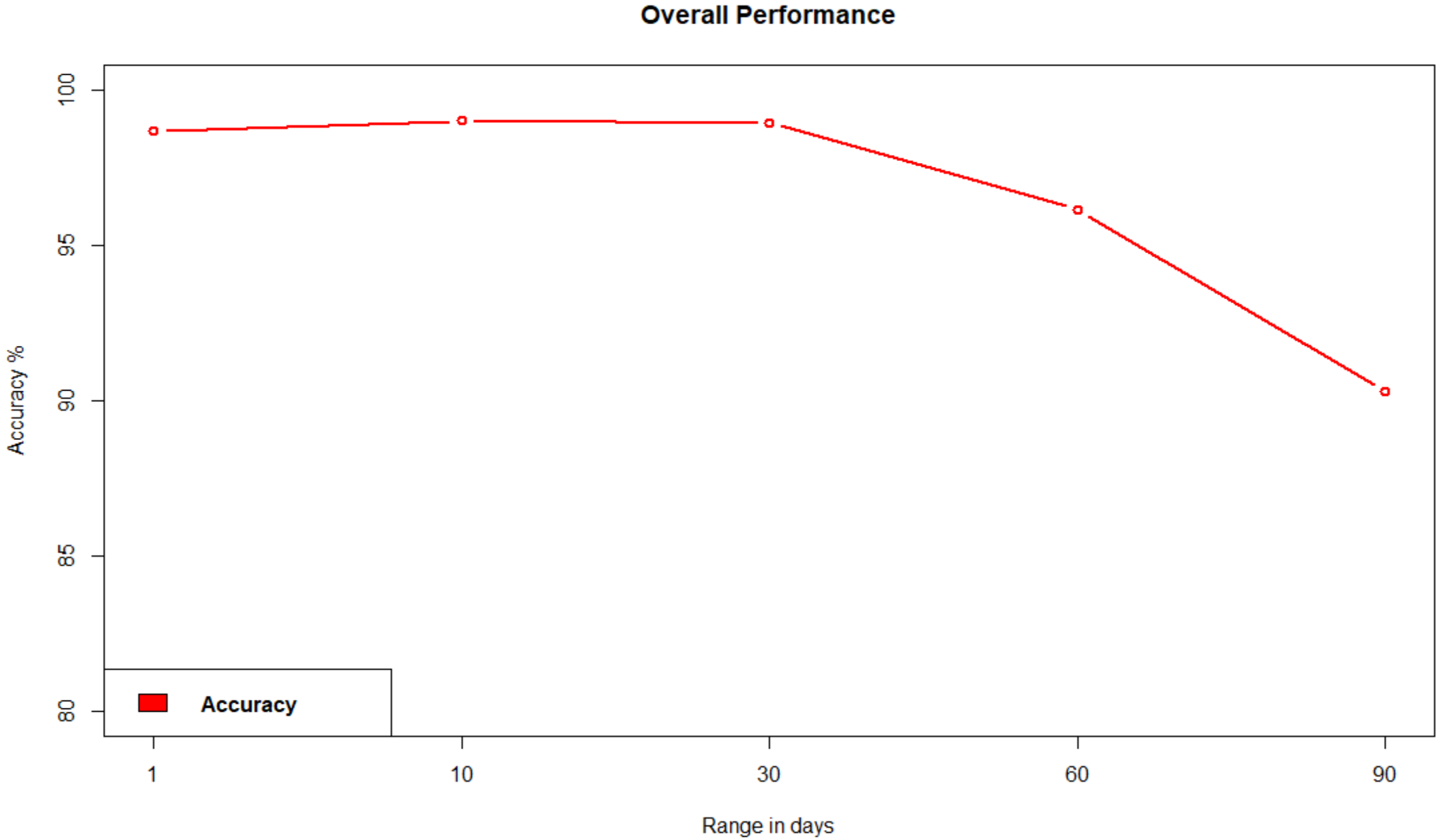
Results & Discussions



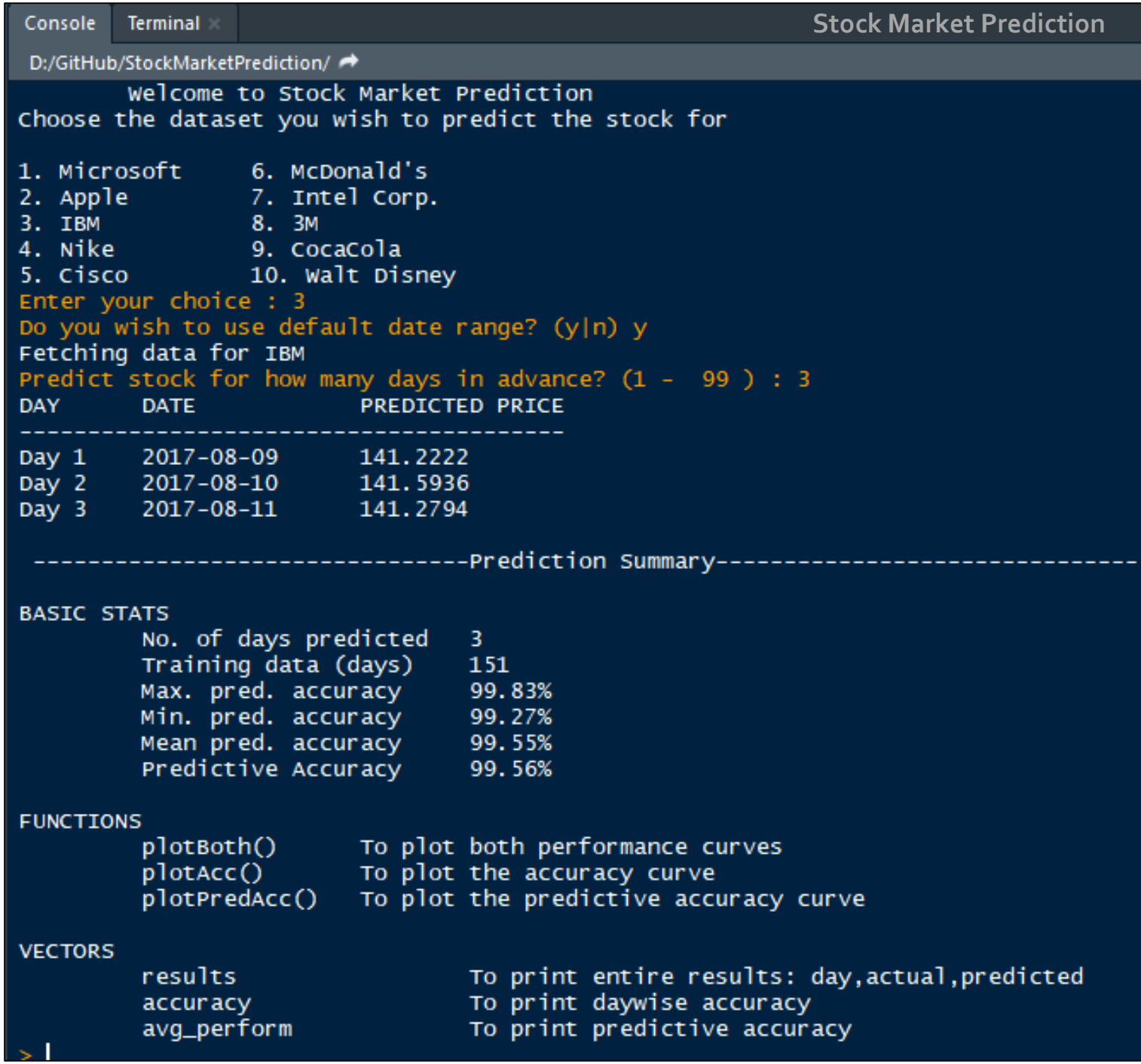
Performance



Performance



Screenshot


Console
Terminal
Stock Market Prediction
D:/GitHub/StockMarketPrediction/
Welcome to Stock Market Prediction
Choose the dataset you wish to predict the stock for
1. Microsoft
2. Apple
3. IBM
4. Nike
5. Cisco
6. McDonald's
7. Intel Corp.
8. 3M
9. CocaCola
10. Walt Disney
Enter your choice : 3
Do you wish to use default date range? (y|n) y
Fetching data for IBM
Predict stock for how many days in advance? (1 - 99) : 3
DAY DATE PREDICTED PRICE
Day 1 2017-08-09 141.2222
Day 2 2017-08-10 141.5936
Day 3 2017-08-11 141.2794
Prediction Summary
BASIC STATS
No. of days predicted 3
Training data (days) 151
Max. pred. accuracy 99.83%
Min. pred. accuracy 99.27%
Mean pred. accuracy 99.55%
Predictive Accuracy 99.56%
FUNCTIONS
plotBoth() To plot both performance curves
plotAcc() To plot the accuracy curve
plotPredAcc() To plot the predictive accuracy curve
VECTORS
results To print entire results: day,actual,predicted
accuracy To print daywise accuracy
avg_perform To print predictive accuracy

Future Enhancement

- The limitation of the proposed system is its computational speed, especially with respect to sliding-window validation as the computational cost increases with the number of forward day predictions.
- The proposed model does not predict well for sudden changes in the trend of stock data.
- This occurs due to external factors and real-world changes affecting the stock market.
- We can overcome this by implementing Sentiment Analysis and Neural Networks to enhance the proposed model.
- We can modify the same system to an online-learning system that adapts in real-time.

Conclusion

Thus, as we can see above in our proposed method, we train the data using existing stock dataset that is available. We use this data to predict and forecast the stock price of n-days into the future.

The average performance of the model decreases with increase in number of days, due to unpredictable changes in trend.

The current system can update its training set as each day passes so as to detect newer trends and behave like an online-learning system that predicts stock in real-time.

Thank You