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
- Implementation
- Results & Snapshots
- Performance Analysis
- Future Scope
- 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 metaheuristic 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

Implementation

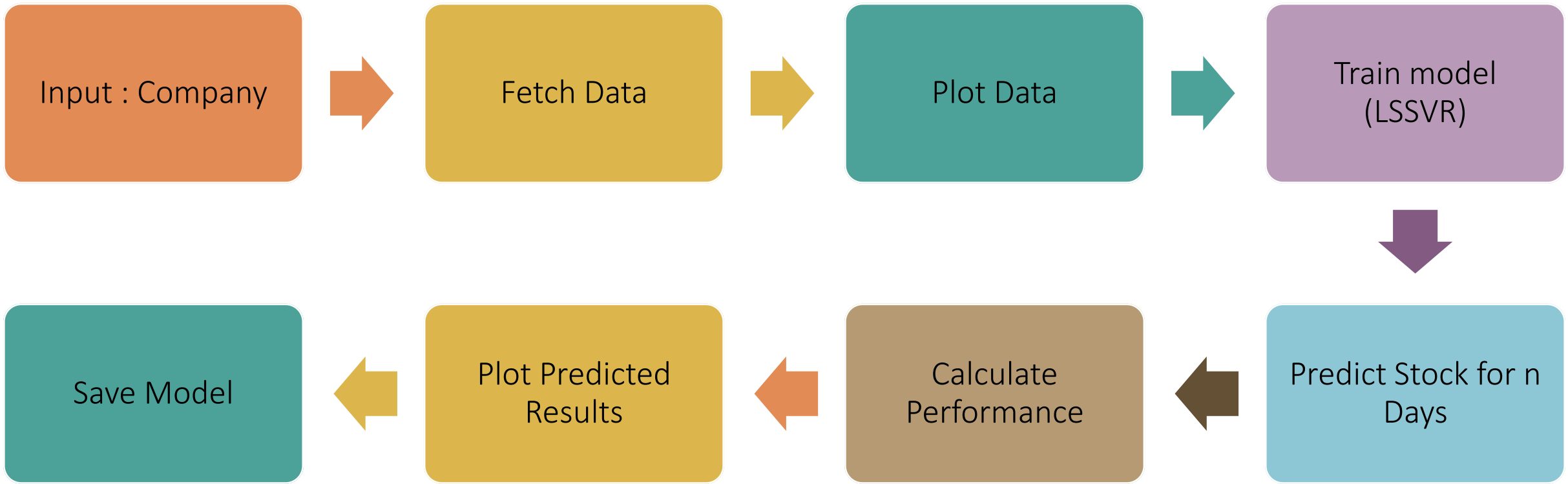
Algorithm

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

Algorithm Flow



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

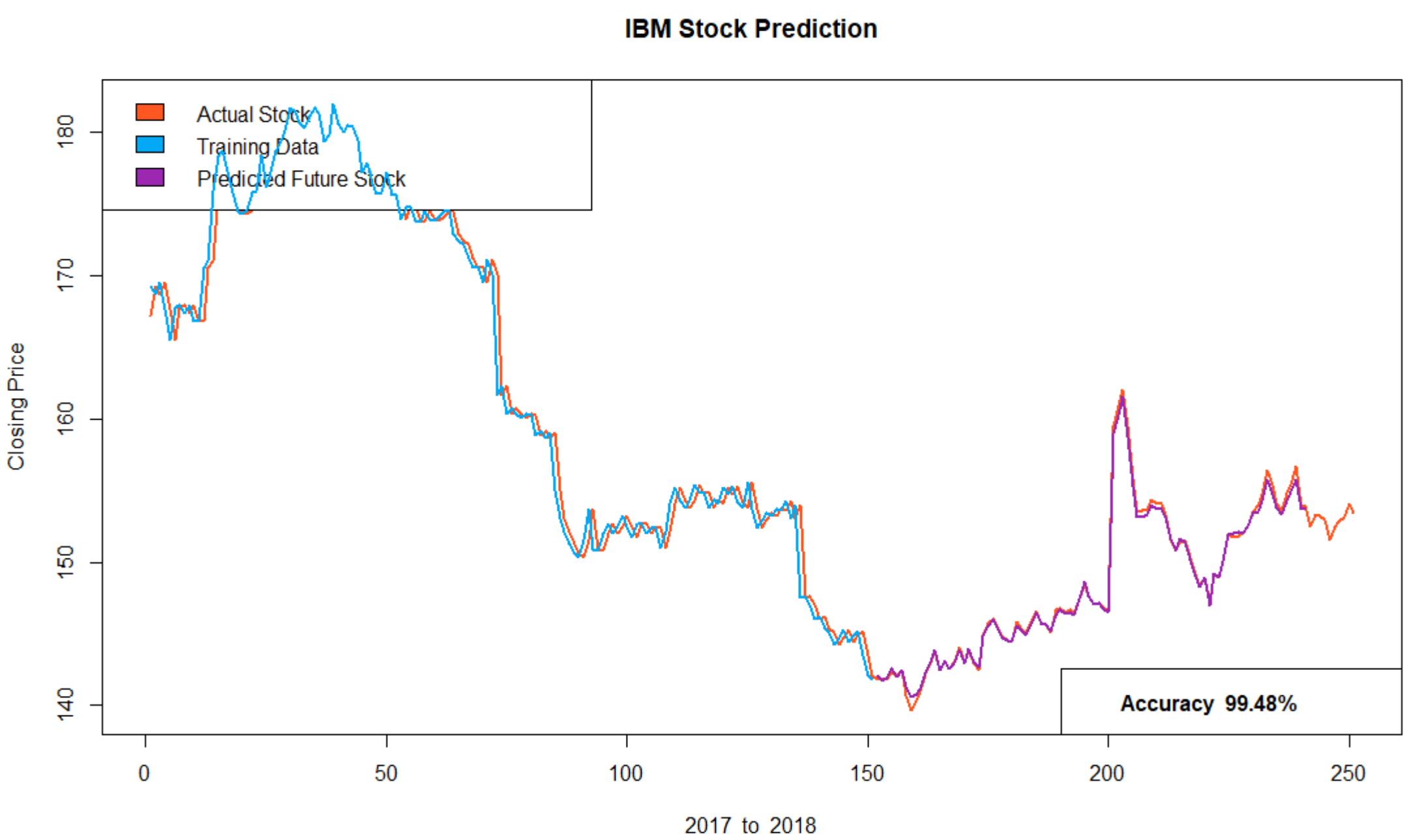
Sliding-Window

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

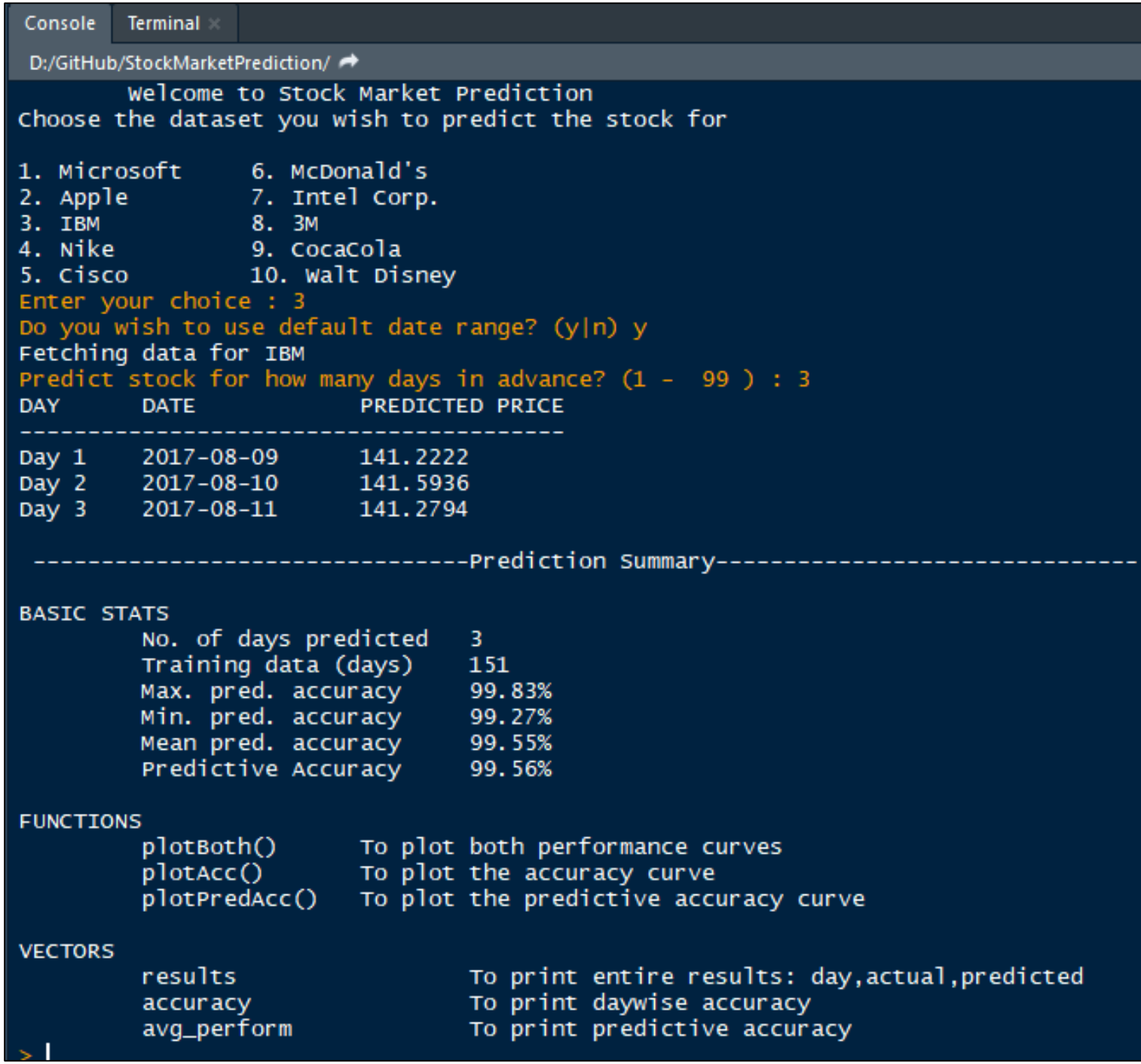
1. LAG ← `1`
2. *y* ← remove first LAG rows from data
3. reset row indices of *y*
4. *x* ← remove last LAG rows from data
5. *train* ← merge *(x,y)* into a dataframe
6. rename column name to *x* and *y*
7. **return** *train*

```
slidingWindow = function(data)
{
    y = data[-(1:LAG),]
    rownames(y) = NULL
    x= data[1:(DATA_SIZE-LAG),]
    train = data.frame(x[,2],y[,2])
    colnames(train) = c("x","y")
    return(train)
}
```

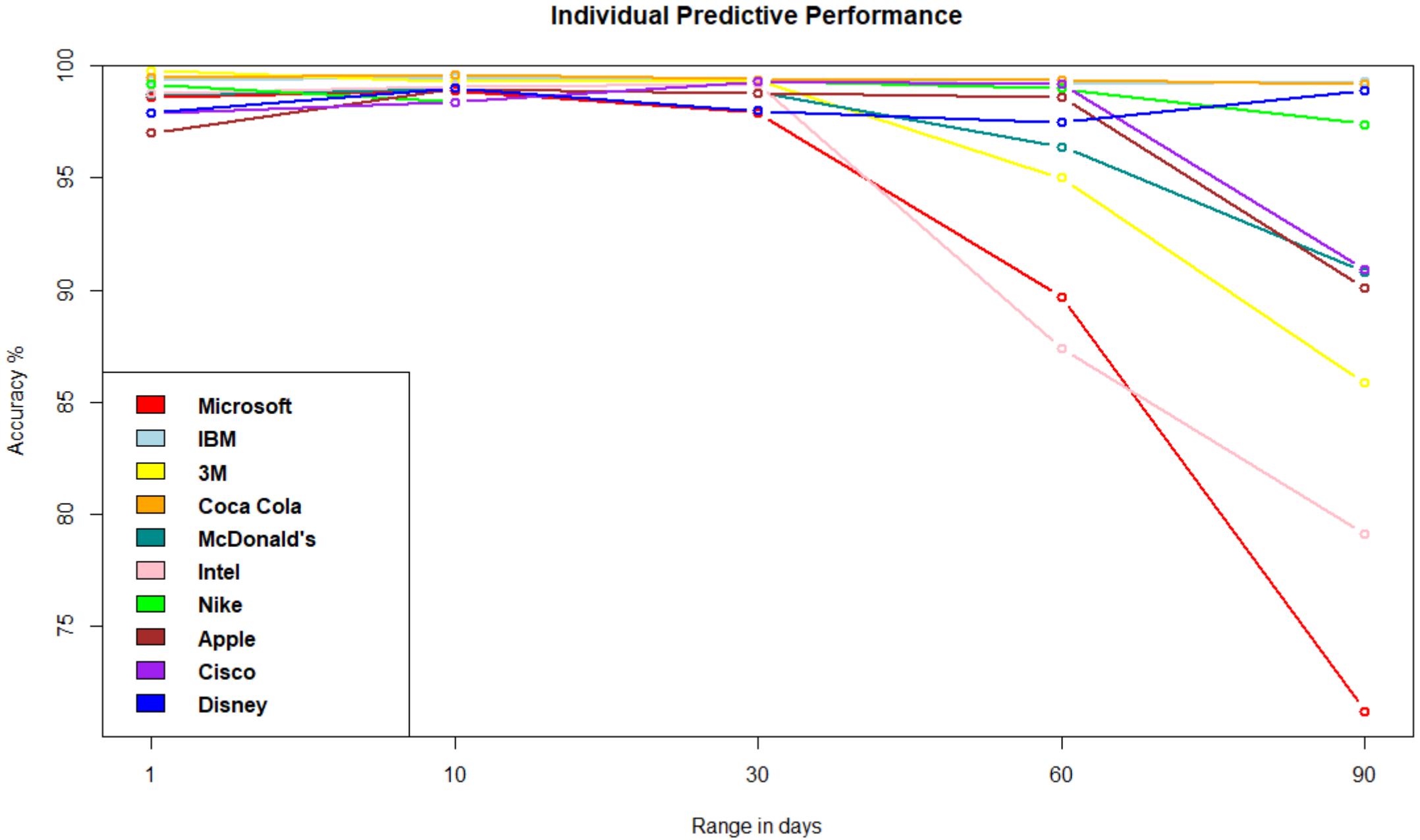
Results


IBM Stock Prediction
Actual Stock
Training Data
Predicted Future Stock
Closing Price
180
170
160
150
140
0
50
100
150
200
250
2017 to 2018
Accuracy 99.48%

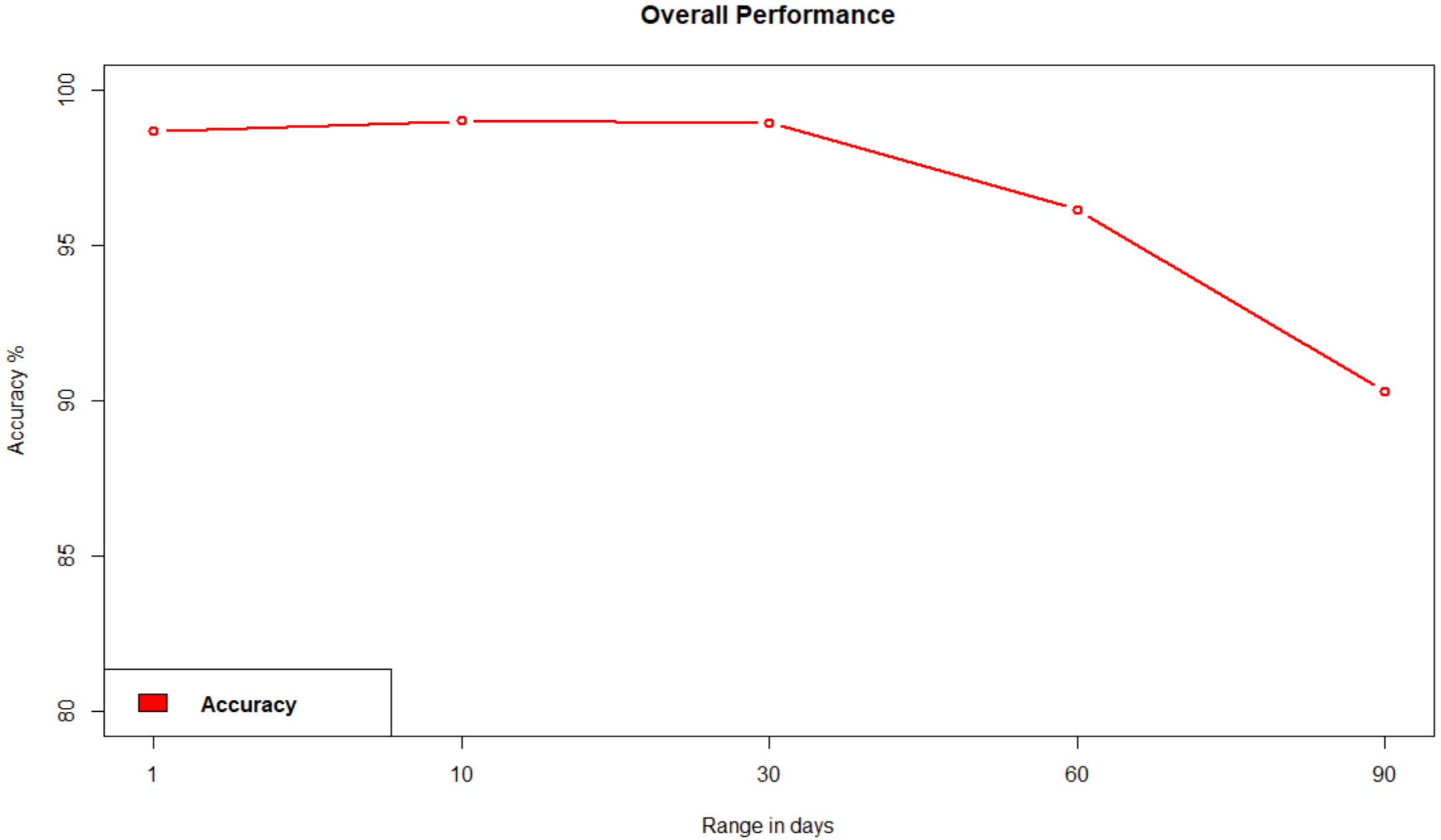
Snapshot


Console
Terminal
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

Performance



Performance



Limitations & Future Scope

Limitations :

- 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.

Future Scope :

- 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

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