PSM Project Status Report: 4
Stock Price Trend Forecasting with Machine Learning Algorithms

Shane Bari
University of Wisconsin-Stout
baris0074@my.uwstout.edu

Russell Chamberlain
University of Wisconsin-Stout
chamberlainr0202.my.uwstout.edu

Cassie Dale
University of Wisconsin-Stout
dalec0190@my.uwstout.edu

Jared Siverling
University of Wisconsin-Stout
siverlingj@uwstout.edu

Keith Wojciechowski
Department of Mathematics
University of Wisconsin, Stout
wojciechowskik@uwstout.edu

Submitted: 12/17/15; Accepted: 12/17/15; Published: Date 3.

Abstract

The following report is a summary of work completed over the course of the Fall 2015 semester at University of Wisconsin-Stout. The summary includes an overview of the preliminary model and a future plans section that explains how the Project will progress next semester. Information from previous reports is included, where needed, for understanding of current and future pursuits.

Overview of Prelimanary Model

The progress of the Project is on schedule as planned. Currently, there is a working model in R that utilizes three predictor functions to return a suggestion of buy, sell, or hold. The R code also contains functions that measure the return on investment (ROI) if the predictions from the program are followed.

Model Input and Output

The model is a set of functions that perform specified tasks outlined in this section. First, the user sets the inputs for the model:

1. data is input from Yahoo Finance using the quantmod package in R
2. downloaded data is selected: open, closing, high, low, or adjusted
3. user may set two time periods for moving averages
4. user provides weights for Simple Moving Average (SMA) or Exponential Moving Average (EMA) – default is General Moving Average (GMA)

Once the above inputs are set, the function

1. calculates the desired moving average with the preferred time limit
2. prints plots showing the data and moving average

3. runs a prediction function that compares the position of the newest data point to the previous moving average and returns a suggestion: buy, sell, or hold

Model Function List and Description

The following defines each function currently being used in the program as well as provides a short description of its purpose:

getSymbols(): Grabs information from Yahoo Finance on selected company's stock history.

WMA(): *Weighted Moving Averages.* Allows moving averages to be weighted, so user can choose SMA, EMA, GMA, or a combination of SMA and EMA.

PlotMAs(): *Plot Moving Averages.* Plots two moving averages against data retrieved from getSymbols() function. Two graphs are shown on the plot, upper and lower, and titles list what moving averages were used.

mAvgsAll(): *Moving Averages All.* Takes in data matrix from getSymbols() and returns a matrix that contains selected data (open, high, low, close, volume, or adjusted) as well as two columns with the information from WMA(). Function calls WMA() and PlotMAs().

gPredDataMA(): *Give Prediction Based on Data vs Moving Average.* Gives a prediction based on matrix from mAvgsAll() and compares selected data with two moving averages (labeled short and long).

gPredDataMABuy(): *Give Prediction on Data vs Moving Average for Buy.* Function returns a true boolean if gPredDataMA() would predict a buy point and returns a false boolean otherwise. This is used in calculating profitably (ROI).

gPredDataMASell(): *Give Prediction Based on Data vs Moving Average for Sell.* Function returns a true boolean if gPredDataMA() would predict a buy point and returns a false boolean otherwise. This is used in calculating profitably (ROI).

gPredCrossMA(): *Give Prediction Based on Comparing Moving Averages.* Gives a prediction based on matrix from mAvgsAll() and compares the two moving averages selected.

gPredCrossMABuy(): *Give Prediction Based on Comparing Moving Averages for Buy.* Function returns a true boolean if gPredCrossMA() would predict a buy point and returns a false boolean otherwise. This is used in calculating profitably (ROI).

gPredCrossMASell(): *Give Prediction Based on Comparing Moving Averages for Sell.* Function returns a true boolean if gPredCrossMA() would predict a sell point and returns a false boolean otherwise. This is used in calculating profitably (ROI).

lreg(): *Linear Regression.* Returns slope of least-squares regression line of last data points. The number of data points used is selected by the user when function is called.

fProfitDataMA(): *Find Profit if gPredDataMA() is Followed.* Calculates ROI if user had used gPredDataMA() to decide when to buy, sell, and hold stock. Function returns information on total amount spent, total profit, and ROI for two moving averages. WARNING: function assumes stock will increase and will only sell stock if the purchase amount of the stock is lower than the current selling point.

fProfitCrossMA(): *Find Profit if gPredCrossMA() is Followed* Calculates ROI if user had used gPredCrossMA() to decide when to buy, sell, and hold stock. Function returns information on total amount spent, total profit, and ROI for two moving averages.

Please see the attachment FILE NAME:"R Code Walkthrough.pdf" TITLE:"Stock Price Trend Forecasting with Machine Learning Algorithms" for a walk-through of the code that explains functions in greater detail, shows the flow of the code and when functions are called, displays outputs of some of the functions, and displays relative code.

Relative R Packages

The Quantmod package used in an R environment provides a number of benefits in terms of importing, manipulating, and analyzing stock market data. Quantmod is free to access and allows us to import data from a number of sources including Yahoo, Google, and FRED as well as local data sources such as an SQL database or .csv file. Since quantmod is available on the Comprehensive R Archive Network (CRAN), extensive documentation and examples are available.[3]

Technical Trading Rules (TTR) is another package available to us and is designed to enhance quantmod by offering specific indicator functions to construct datasets that can be used for prediction. A full list of functions available in TTR is given in [6]. Currently, the TTR package is not being utilized but may be used in the upcoming semester depending on the needs of the Project.

Explanation of Plots

Plots printed from the model, using PlotMAs(), will show the chosen data as a red line and the selected moving average as a green line. The titles used in the plots will change based on the time periods chosen for consideration.

Short Comings of Model

Though functioning, the preliminary model is still basic. As such, it has short comings. The model currently has two major prediction functions: gPredDataMA() and gPredCrossMA(). There is an important assumption made in gPredDataMA(): the function will only sell stocks if their is a profit to be made. This means if there is a catastrophic or significant loss in the value of the stock, the function does not have way to "cut-losses" and sell before the value of the stock is above the purchased amount. Both prediction functions only allow for the single purchase of a stock, the program will not allow multiple purchases if there is another "buy" return without their being a "sell" command between "buy" commands.

The model also does not consider any outside influences and is solely making a prediction based on previous data. This means that outside forces are not considered by the model, currently. So, if a catastrophic event happens in the US market, global economy, or a change in regulations, the model will not take these factors into consideration.

The model only uses SMA and EMA as moving averages. It does give more functionality by allowing weights or using SMA and EMA to calculate GMA, but it can only use these two moving averages for the prediction. We have been researching additional moving averages to add to the model if we can show a significant improvement to the model from their addition.

The model is currently set to daily rates and cannot be used for weekly or monthly rates in its current condition. However, it is believed that simple modifications can be made to the primary function in the model as well as how data is pulled form Yahoo Finance to resolve this short coming.

Future Plans and Options

The preliminary model is a solid foundation for the final model that will meet the goals of the Project. The profit functions will be continually improved to allow for the purchase of more than one stock and additional prediction functions may be added. We have reached the point where we are considering different Machine Learning(ML) algorithms to implement next semester as well as preparing to utilize data mining techniques.

Data Mining

Data mining techniques will be utilized to gather information on the count of significant words. These counts will be used as indicators for special interests, like market sentiment. There will be a massive amount of data collected so the team will be learning to program in Structured Query Language(SQL) to manage, sort, and store the large amounts of data for the Project.

Learning Algorithm

Machine Learning (ML) algorithms recognize patterns in past data generate predictive models and have been used in financial time series and stock market applications with moderate-to-high levels of accuracy.[1][2][4][5][7] Examples of ML algorithms include logistic regression, support vector machines, and artificial neural networks; future research will be required to identify the ideal algorithm/combination of algorithms.

References

[1] Shunrong Shen Haomiao Jiang and Tongda Zhang. Stock market forecasting using machine learning algorigthms.

[2] Kyoung jae Kim. Financial time series forecasting using support vector machines.

[3] Joshua M. Ulrich Jeffrey A. Ryan and Wouter Thielen. Quantitative financial modelling framework, 2015.

[4] Neil F. Riley Jerry K. Bilbrey, Jr. and Caitlin L. Sams. Short-term prediction of exchange traded funds (etfs) using logistic regression generated client risk profiles.

[5] Ismail Hmeidi Khalid Alkhatib, Hassan Najadat and Mohammed K. Ali Shatnawi. Stock price prediction using k-nearest neighbor (knn) algorithm. 3(3), March 2013.

[6] Joshua Ulrich. Technical trading rules, 2015.

[7] Marijana Zekie. Neural network applications in stock market predictions.