Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting

Working Paper · October 2016
DOI: 10.13140/RG.2.2.15294.48968

CITATION
1

READS
4,571

4 authors, including:

Shubharthi Dey
PES Institute of Technology
1 PUBLICATION **1** CITATION
SEE PROFILE

Snehanshu Saha
PES University
128 PUBLICATIONS **168** CITATIONS
SEE PROFILE

Suryoday Basak
PESIT South Campus
16 PUBLICATIONS **20** CITATIONS
SEE PROFILE

Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting

Shubharthi Dey*
PESIT South Campus
Bangalore
Karnataka, India
deysubharthi15@gmail.com

Yash Kumar†
PESIT South Campus
Bangalore
Karnataka, India
yash.kumar1396@gmail.com

Snehanshu Saha‡
PESIT South Campus
Bangalore
Karnataka, India
snehanshusaha@pes.edu

Suryoday Basak§
PESIT South Campus
Bangalore
Karnataka, India
suryodaybasak@gmail.com

ABSTRACT

Stock market prediction is the art of determining the future value of a company stock or other financial instrument traded on an exchange. It had been a real challenge for analysts and traders to predict the trends of stock market due to its uncertain nature. Stock prices are likely to be influenced by the factors like product demand, sale, manufacture, investor's sentiments, ruling government, recession etc. The successful prediction of a stock's future price could yield significant profit. The main aim of this paper is to design an efficient model which will accurately predict the trend of stock market using eXtreme Gradient Boosting(XGBoost) which has proved to be an efficient algorithm with over 87% of accuracy for 60 day and 90 day periods and it has proved to be much better when compared to traditional non-ensemble learning techniques. The prediction problem has been reconstructed as a classification problem and XGBoost turned out to be significantly better than the algorithms found in literature. The proposed model outperforms all existing forecasting models in literature and is able to forecast on long-term basis, an added feature absent in literature.

Keywords

XGBoost, ensemble learning, Exponential smoothing

*First Author and Second author share equal credit.

†First Author and Second author share equal credit.

‡Dr. Saha is the corresponding author.

§Suryoday is with the Center of Applied Mathematical Modeling.

1. INTRODUCTION

The stock market has always been the area of interest for people around the world. Perception and experience are put to good use by non-practitioners to invest in a particular company hoping for multiplied return. The stock market trend prediction had also been the area of keen interest of statisticians and computer scientists simply for the reason that the area throws complex modeling questions. There exist methods or algorithms which could predict the stock valuation with a fair degree of accuracy. However, a question still persists: if a person decides to buy shares from a particular company, what is the probability that it turns out to be a successful expedition or mere failure? An informed guess works on a broader basis i.e. considering the production sale and demand of the organization in the present scenario, it may be fine to invest in aparticular stock. However, it is too much to expect this to work in complex situations ignoring certain nunaced concepts and factors that govern the market. For example, the political situation in a country is inefficient and too volatile to handle the economy of the country. A fall in the economy triggers fall in the stock value of a company. Due to these minute and chaotic parameters, the prediction becomes increasingly difficult. The traders tend to invest in a firm which has potential or history of good return based on the current situation. However, there always exists a possibility that a company which appears to have incurred a loss may be the potential firm which the traders/investors may have continued faith in.

Over the past years, there had been enormous research in this field pivoted around statistical machine learning. Different predictive models and algorithms to a certain degree of accuracy, have been proposed and tested. Implementation of machine learning techniques is an evolving concept. This is somewhat different from traditional forecasting and diffusion methods. Early models used in stock forecasting involved statistical methods such as time series model and multivariate analysis (Gencay (1999), Timmermann and Granger(2004), Bao and Yang(2008)). Our paper mainly focuses on the machine learning approach of analysis as it is evident that the historical data set obtained (say from the date when the company existed) is impossible to analyze

without data mining methods. The prediction as a result of our proposed algorithm may help people to decide whether to invest in a particular company taking into consideration the chaos and volatility of stock. We adopted a machine learning approach different from the commonly practiced ones such as Support Vector Machine, Neural Network and Naive Bayesian Classifier, Linear Discriminant Analysis etc. Next section discusses related literature.

2. LITERATURE SURVEY

Recent years witnessed considerable traction in the field of Stock market prediction specially from the Machine Learning point of view. Prediction of stock market behavior is used to determine the future trends. Prediction is usually accomplished by analyzing historic time series data.

Several algorithms have been used in stock prediction such as SVM, Neural Network, Linear Discriminant Analysis, Logistic Regression, Linear Regression, KNN and Naive Bayesian Classifier. It was found that logistic regression was one of the best with a success rate of 55.65%. Dai and Zhang (2013) used the training data from 3M Stock data. Data contains daily stock information ranging from 1/9/2008 to 11/8/2013 (1471 data points). Multiple algorithms were chosen to train the prediction system. These algorithms include Logistic Regression, Quadratic Discriminant Analysis and SVM. The algorithms were applied to the next day model which predicted the outcome of the stock price on the next day and long term model, which predicted the outcome of the stock price for the next n days. The next day prediction model produced accuracy results ranging from 44.52% to 58.2%. SVM reported the highest accuracy of 79.3%, For the long term prediction where time window taken was 44.In one of the published paper **[11]** on ANN(artificial neural network),authors used for the forecast the direction of Japanese stock market gave an accuracy of 81.27%. In the published paper on **[10]** Random Forest(RF), an ensemble technique is used to predict the stock market prices(Trend up or Down) and returned the highest accuracy of 78.81% for the i^{th} day on the direction of movement in the daily TSE(Tehran Stock Exchange) index.

Ensemble learning algorithms remains largely unexplored however. The focus of the is to implement the ensemble learning technique and to discuss its advantages over non-ensemble techniques. We will be using an ensemble learning method known as Xgboost to build our predictive model. Our model has been trained for 60, 90 and 120 days respectively and the results were impressive. Moreover, majority of the the related work focused on the time window of 10 to 44 days on an average as majority of the authors had preferred to use metric classifiers on time series data which is not smoothed. Therefore, those models are unable to learn from the data set when it comes to predicting for a long term window. Our model underscores that it firstly smooths the data to begin with being a non-metric classifier, it is capable of accurately predicting in a long term time window.

The paper will highlight certain critical aspects largely ignored by most of the literature. These include analyzing the non-linearity in features used for analysis and futility of employing linear classifiers, Long-run predictions running into 90 days where related manuscripts considered the time window up to 44 days. Significant improvement in accuracy obtained by our approach embellish the claims.

The remainder of the paper is organized as follows. Section 3 contains key definitions. Section 4 describes Data preprocessing and feature extraction in detail. Section 5 describes the methods and algorithm used. The results of the applied algorithm are obtained and analyzed in section 6. The next section is a comparative study establishing the superiority of our proposed algorithm. The authors conclude with a comprehensive summary of the work.

3. KEY DEFINITIONS

Time Series Data: A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data.

Note: We used the Apple,Yahoo data set. This is a time series data set which is further smoothed exponentially as discussed in section 4.

Gradient Boosting: Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Technical Indicators: Technical Indicators are important parameters that are calculated from time series stock data that aim to forecast financial market direction. They are tools which are widely used by investors to check for bearish or bullish signals.

Relative strength Index: Relative strength index(RSI) is calculated as

$$RSI = 100 - \frac{100}{(1 + RS)}$$

$$RS = \frac{average\ gain\ over\ past\ 14\ days}{average\ loss\ over\ past\ 14\ days}$$

The RSI is classified as a momentum oscillator, measuring the velocity and magnitude of directional price movements. Momentum is the rate of the rise or fall in price. The RSI computes momentum as the ratio of higher closes to lower closes: stocks which have had more or stronger positive changes have a higher RSI than stocks which have had more or stronger negative changes.

The RSI is most typically used on a 14-day timeframe, measured on a scale from 0 to 100, with high and low levels marked at 70 and 30, respectively. Shorter or longer timeframes are used for alternately shorter or longer outlooks. More extreme high and low levels-80 and 20, or 90 and 10 - occur less frequently but indicate stronger momentum.

Stochastic Oscillator: Stochastic oscillator is given by

$$\%k = 100 * \frac{C - L_{14}}{H_{14} - L_{14}}$$

where, C=current closing price; L_{14}= Lowest low over past 14 days; H_{14}= Highest high over past 14 days

The term stochastic refers to the point of a current price in relation to its price range over a period of time. This method attempts to predict price turning points by comparing the closing price of a security to its price range.

William % R: William's %R is give by

$$\%R = -100 * \frac{H_{14} - C}{H_{14} - L_{14}}$$

where, C = Current Closing Price; L_{14} = Lowest Low over the past 14 days; H_{14} = Highest High over the past 14 days. Williams %R ranges from -100 to 0. When its value is above -20, it indicates a sell signal and when its value is below -80, it indicates a buy signal.

Moving Average Convergence Divergence(MACD): The formula for calculating MACD is:

$$MACD = EMA_{12}(C)\text{-}EMA_{26}(C)$$

$$SignalLine = EMA_9(MACD)$$

where, MACD = Moving Average Convergence Divergence; C = Closing Price series; EMA_n = n day Exponential Moving Average.

When the MACD goes below the SignalLine, it indicates a sell signal. When it goes above the SignalLine, it indicates a buy signal.

Price Rate of Change: It is calculated as follows:

$$PROC(t) = \frac{C(t)\text{-}C(t\text{-}n)}{C(t\text{-}n)}$$

where, PROC(t) = Price Rate of Change at time t; C(t) = Closing price at time t. It measures the most recent change in price with respect to the price in n days ago.

On Balance Volume: This technical indicator is used to find buying and selling trends of a stock.

$$OBV(t) = \begin{cases} OBV(t\text{-}1)+Vol(t) & if C(t) > C(t\text{-}1) \\ OBV(t\text{-}1)\text{-}Vol(t) & if C(t) < C(t\text{-}1) \\ OBV(t\text{-}1) & if C(t) = C(t\text{-}1) \end{cases}$$

where, OBV(t) = On Balance Volume at time t; Vol(t) = Trading Volume at time t; C(t) = Closing price at time t.

Convex Hull: The convex hull of a set of points X is its subset which forms the smallest convex polygon that contains all the points in X. A polygon is said to be convex if a line joining any two points on the polygon also lies on the polygon. *The significance of Convex Hull is explained in section* 6.1

Linear Separability: Two sets of points X_0 and X_1 in n dimensional Euclidean space are said to be linearly separable if there exists an n dimensional normal vector W of a hyperplane and a scalar k, such that every point $x \in X_0$ gives $W^T x > k$ and every point $x \in X_1$ gives $W^T x < k$

Bagging: Bootstrap aggregating, also called bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach.

4. DATA PREPROCESSING AND FEATURE EXTRACTION

The data set is borrowed from yahoo finance, which included closing price, opening price, High, Low and Volume **(Apple,Yahoo)**. The time series historical stock data is exponentially smoothed. **Exponential smoothing** applies more weightage to the recent observation and exponentially decreasing weights to past observations. The exponentially smoothed statistic of a series Y can be recursively calculated as:

$$S_0 = Y_0; S_t = \alpha * Y_t + (1-\alpha) * S_{t-1}$$

where α is the smoothing factor, $0 < \alpha < 1$. Larger values of α reduce the level of smoothing. When $\alpha = 1$, the smoothed statistic becomes equal to the actual observation. The smoothed statistic, S_t can be calculated as soon as two observations are available. This smoothing removes random variation or noise from the historical data allowing the model to easily identify long term price trend in the stock price behavior. Technical indicators are then calculated from the exponentially smoothed time series data which are later organized into feature matrix. The target to be predicted in the i^{th} day is calculated as follows:

$$target_i = Sign(close_{i+d} - close_i)$$

where d is the number of days after which the prediction is to be made. When the value of $target_i$ is +1, it indicates that there is a positive shift in the price after d days and -1 indicates that there is a negative shift after d days. The $target_i$ values are assigned as labels to the i^{th} row in the feature matrix.

We typically expect to have a huge data set in order to enable the algorithm to recognize the pattern in the data set. This analysis becomes cumbersome with high possibility of error. Machine Learning proposes solutions for handling such huge data in an efficient manner. There are a range of methods classified into metric and non-metric classifer based on their working principle.

XGBoost is an ensemble ML technique which is based on the concept of Decision Tree with some advanced modifications that are designed to differentiate the performance of an XGBoost model from that of a simple Decision tree model. We present a brief overview of decision tree from the perspective of XGBoost.

Technical Indicators are important parameters that are calculated from time series stock data that aim to forecast financial market direction. They are tools which are widely used by investors to check for bearish or bullish signals. These technical indicators are calculated from time series stock market data available for predicting the direction of stock market. The technical indicators used are ***RSI indicator,Stocastic Oscillator, William % R, Moving Average Convergence Divergence(MACD), Price Rate Of Change,On Balance Volume*** and defined in the previous Section.

Technical indicators that are calculated using past observations, have been used as features. Thus, the order of the dates become irrelevant when bagging is performed. We use these indicators that were calculated with t-n data and reuse them to predict the t+1 event. An extreme example can be that we use features of day 3, features of day 30 to predict day 45. However,doing that does not ignore information embedded in the correlation of consecutive days. The features of ith day is calculated using OHCLV data of the past n days. For example, for the sake of simplicity, let us assume we have a technical indicator called F. This indicator is calculated using the closing price of the past n days i.e, Close(i-1), Close(i-2)... Close(i-n).

Feature extraction is a mechanism that computes numeric or symbolic information from the observation. The main task is to select or combine the features that preserve most of the information and remove the redundant components in

order to improve the efficiency of the subsequent classifiers without degrading their performances. It is the process of acquiring higher level information. The dimensionality of the feature space may be reduced by the selection of subsets of good features. Feature extraction plays an important role in a sense of improving classification performance and reducing the computational complexity. It also improves computational speed due to the fact that for less features, less parameters have to be estimated.

Feature Extraction Algorithm: According to the published paper[12] on Feature Selection, also known as the feature subset selection (FSS), or attribute selection (Attribute Selection) which is a method to select a feature subset from all the input features to make the constructed model better. In the practical application of machine learning, the quantity of features is normally very large, in which there may exist irrelevant features, or the features may have dependence on each other. Feature Selection can remove irrelevant or redundant features, and thus decrease the number of features to improve the accuracy of the model. Also selecting the really relevant features can simplify the model, and make the data generation process easy to understand for the researchers. The XgBoost algorithm is able to rank the various features based on their importance and based on the rank the high or low is accomplished.

5. METHOD

We begin by presenting a high level work flow diagram of the method employed.

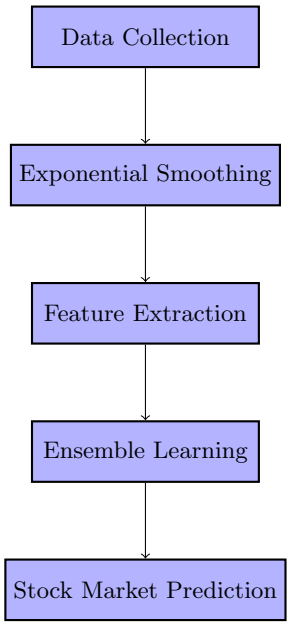

Fig.1:Illustration for proposed methodology

The first two blocks in the flowchart are Data Collection and Exponential Smoothing which has been discussed in section 4. The third block describes Feature Extraction is basically the selection of features from data set and building the derived values so that the data becomes more informative and non-redundant. Once, the feature is extracted and the feature matrix is prepared the data is subjected to train the algorithm which is done under the process of ensemble learning. After the model recognizes a pattern, 20% of the data set is used to test the robustness of the model and finally the prediction made is checked for accuracy, specificity, sensitivity which takes in the last block of the flowchart.

Surveying various machine learning algorithms was a key motivation even though some methods and algorithms could have been easily done with. This explains the reason for describing methods such as SVM or LDA even though the results are not very promising as they fall under the category of metric classifiers as mentioned in [1]. We reiterate that any learning method is as good as the data and without a balanced data set there could not exist any reasonable scrutiny of the efficiency of the methods used in the manuscript or elsewhere. Non-metric classifiers which include Decision Tree, Boosted Trees, would bolster the logic behind discouraging"Black-box" approaches to Data Analytics in the context of this problem or otherwise.

Before feeding the training data to the XgBoost model, the two classes of data are tested for **linear separability** by finding their convex hulls. Linear Separability is a property of two sets of data points where the two sets are said to be linearly separable if there exists a hyperplane such that all the points in one set lies on one side of the hyperplane and all the points in other set lies on the other side of the hyperplane. The separability of the training set determines whether the hypothesis space can solve a particular binary classification problem or not. The separability test can provide a set of hypothesis (initial solutions) which can be refined to minimize generalization error.

5.1 Test for linear separability

In order to check for Linear Separability, the convex hulls for the two classes are constructed. If the convex hulls intersect each other, then the classes are said to be linearly inseparable. Principle component analysis is performed to reduce the dimensionality of the extracted features into two dimensions. This is done so that the convex hull can be easily visualized in 2 dimensions. The convex hull test reveals that the classes are not linearly separable as the convex hulls almost overlap.

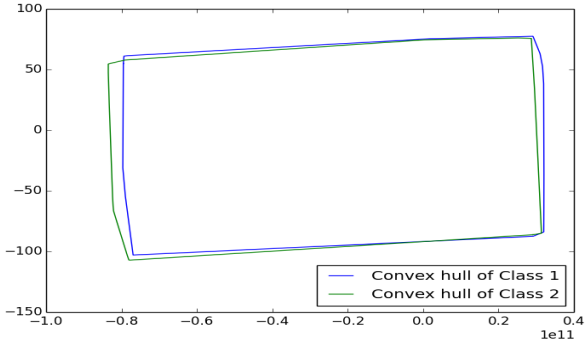

Fig.2: Convex Hull Test for Linear Separability

The observation concludes that Linear Discriminant Analysis cannot be applied to classify our data and hence, provid-

ing a stronger justification to why Xgboost is used. Another important reason is that since each decision trees in the algorithm operate on the random subspace of the feature space, it leads to the automatic selection of the most relevant subset of features.
After the analysis it is found that the data is not linearly separable, hence the metric methods are not preferred. Therefore the implementation of non-metric methods come into picture which is discussed in the next section.

5.2 Non-Metric Classifiers

A few significant and often used non-metric classifier include:
Decision Tree
Random Forest
Xtreme Gradient Boost

Decision Tree

Decision trees can be used for various machine learning applications. Decision tree constructs a tree that is used for classification and regression. But trees that are grown really deep to learn highly irregular patterns tend to over-fit the training sets. A slight noise in the data may cause the tree to grow in a completely different manner. This is because of the fact that decision trees have very low bias and high variance. Each of the nodes in tree is split on the basis of training set attributes. Every other node of the tree is then split into child nodes based on certain splitting criteria or decision rule, which determines the allegiance of the particular object (data) to the feature class. The leaf nodes must be pure nodes; when any feature vector that is to be classified reaches a leaf node. Splitting is done on the basis of highest importance of the attribute which is done using **Gini impurity** or **Shannon entropy**. Information gain is calculated and the attributes are selected. One significant advantage of decision trees is that both categorical and numerical data can be dealt with; a disadvantage is that decision trees tend to over-fit the training data. In order to prevent over fitting of the model pruning must be done, while constructing the tree or after the tree is constructed.
Gini impurity is used as the function to measure the quality of split in each node. Gini impurity at node N is given by

$$g(N) = \sum_{i \neq j} P(w_i)P(w_j)$$

where $P(w_i)$ is the proportion of the population with class label i. Another function which can be used to judge the quality of split is Shannon Entropy. It measures the disorder in the information content. In Decision trees, Shannon entropy is used to measure the unpredictability in the information contained in a particular node of a tree (In this context, it measures how mixed the population in a node is). The entropy in a node N can be calculated as follows.

$$H(N) = -\sum_{i=1}^{d} P(w_i) log_2(P(w_i))$$

where d is number of classes considered and $P(w_i)$ is the proportion of the population labeled as i. Entropy is the highest when all the classes are contained in equal proportion in the node. It is the lowest when there is only one class present in a node (when the node is pure).
The best split is characterized by the highest gain in information or the highest reduction in impurity. The information gain due to a split can be calculated as follows

$$\Delta I(N) = I(N) - P_L * I(P_L) - P_R * I(P_L)$$

where I(N) is the impurity measure (Gini or Shannon Entropy) of node N, P_L is the proportion of the population in node N that goes to the left child of N after the split and similarly, P_R is the proportion of the population in node N that goes to the right child after the split. $N_L$and N_R are the left and right child of N respectively.
Assume there are n data points $D = \{(x_i, y_i)\}_{i=1}^{n}$ and feature vectors $\{x_i\}_{i=1}^{n}$ with stated outcomes. Each feature vector is d- dimensional.
1: We define a classification tree where each node is endowed with a binary decision *if $x_i <= k$ or not; where k is some threshold.* The topmost node in the classification tree contains all the data points and the set of data is subdivided among the children of each node as defined by the classification . The process of subdivision continues until every node below has data belonging to one class only. Each node is characterized by the feature, x_i and threshold k chosen in such a way that minimizes diversity among the children nodes. This is often referred as gini impurity.
2: $X = (X_1, ..., X_d)$ is an array of random variables defined on probability space called as random vectors. The joint distribution of $X_1, ..., X_d$ is a measure on μ on R^d, $\mu(A) = P(X \in A)$, $A \in R^d$ where $d = 1, ..., m$. For example, Let $x = (x_i, ..., x_d)$ be an array of data points. Each feature x_i is defined as a random variable with some distribution. Then the random vector X has joint distribution identical to the data points, x.
3: Let us represent $h_k(x) = h(x|\theta_k)$ implying decision tree k leading to a classifier $h_k(x)$. Thus, a random forest is a classifier based on a family of classifiers $h(x|\theta_1), ..., h(x|\theta_k)$, built on a classification tree with model parameters θ_k randomly chosen from model random vector θ. Each classifier, $h_k(x) = h(x|\theta_k)$ is a predictor of the number of training samples. $y =^{+}_{-} 1$ is the outcome associated with input data, x for the final classification function, $f(x)$.
Next, we describe the working of the Xgboost learner by exploiting the key concepts defined above.

Xtreme Gradient Boost(XGBoost)

XGBoost is another method which comes under non-metric classifier family which is based on the concept of Decision Tree, but there are significant differences between the two. XGBoost is an ensemble of decision trees wherein weighted combinations of predictors is taken. XGBoost works on the same lines of Random Forest, but there is a difference in working procedures. The similarities are that the features extracted in both the cases is completely random in nature. If **n** is the total number of attributes in the feature matrix then lets say **m** is the number of attributes which are finally chosen to determine the split at each node. Here m is related to n in the following way

$$m = \frac{n}{3}$$

XGBoost basically is a collection of weak classifier decision trees and it primarily focuses to train the new decision tree to learn from the errors committed by the previous tree[s]. The learning trees are trained sequentially. Initially, a regression function is drawn which is fitted to the data set and due to random plotting of regression function errors occur which are referred to as **residual errors**. Subsequently, a

plot of all the residual errors is considered and another regression function is made to fit the model, the residual errors occurred in that case is taken care by the combination of the previous regression function and the current regression function. Hence continuing in this manner the regression function gets more and more complex in nature and the root mean squared error is observed to be significantly reduced. The following are the basic steps followed while executing XGBoost

5.3 Algorithm

The following steps are recursively carried throughout the process.
Step 1: Learn a regression Predictor.
Step 2: Compute the error residual.
Step 3: Learn to predict the residual. The error rate is calculated using the parametersmentioned below.
Error in prediction is given by

$$J = (y, \hat{y})$$

where

$$J(.) = \sum (y[i] - \hat{y}[i])^2$$

$\hat{y}$ can be adjusted in order to reduce the error by using the following formula

$$y[i] = y[i] + \alpha f[i]$$

where

$$f[i] \approx \nabla J(y, \hat{y})$$

Each learner estimates the gradient of the loss function. Gradient Descent is used to take sequence of steps to reduce sum of predictors weighted by step size α. We present the proposed algorithm for XGBoost below.

Algorithm 1 Xtreme Gradient Boosting

procedure XTREMEGRADIENTBOOST(D) ▷ D is the labeled training data
  `Initialize model with a constant value`
  $F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$.
  for do $m = 0$ to M
    `Compute the pseudo-residuals`
    `Fit base learner to pseudo residuals`
    T_i `= new DecisionTree()`
    $features_i$ `= RandomFeatureSelection(`D_i`)`
    T_i`. train(`D_i`,`$features_i$`)`
    `Compute multiplier` γ_m
    `Update the model`
  output $F_m(x)$

$L(y, \gamma)$ is the differentiable loss function and $h_m(x)$ is the base learner, connected by the following relation: $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$.

5.4 Analysis & Data Discovery

The following aspects deserve some discussion in the context of the data and algorithm applied to analyze the data.
Data Set Analysis: The data set doesn't contain categoriacal and ordinal variables. The column containing labels, "+1+ and "-1" has been removed as this is used for training. ID of the data set has been removed as it adds to the noise and is not significant at all. Gneralized Linear Models, for instance, assume that the features are uncorrelated. Assuming otherwise may sometimes make prediction less accurate, and most of the time make interpretation of the model almost impossible, under the GLM setting. Fortunately, boosted trees are very robust to these features. This eliminates the exercise of checking strong/weak correlation of the features.
Feature ranking & Feature correlation: Building a feature importance list is important as the ranks show the decreasing importance of features and may indicate/suggest omitting a few based on Gain, Cover and Frequency. In this case, the Gain values are the same for all the six features. Therefore pruning any of the features is not advised.
All the 6 technical indicators used are strongly correlated to each other. The correlation is confirmed by performing the *Chi-Squared* Test. Higher *Chi- Squared values* imply better correlation, p-values were always greater than 0.1.
One of the limitations of XgBoost is its limited flexibility in handling non-numeric data. If there are categorical or ordinal data, conversion to numeric data is necessary. The data set under investigation has only numeric data, therefore it did not affect the performance of XgBoost while pruning or splitting.
As figures 3 and 4 show (Please refer the next section, Results) train-RMSE error decreases. The model learns the data well without exhibiting random fluctuations in the error rate. Let us now proceed to the experimental results which will confirm the theory and expectations from the predictive model.

6. RESULT

The main aim of this paper is to predict the rise and fall of the stock market. Hence as a measuring parameter we have used +1 for indicating the rise in stock valuation in the future and -1 to indicate the fall in the prices. The following results were observed after the computation of the data set by using XGBoost. We obtain the root mean squared error(RMSE) for the **60 day prediction and 90 day prediction for Apple Inc. Data set** as

Fig. 3: RMSE plot for Apple Inc. data set

Fig.4 shows the reduction in RMSE for the **28 day, 60 day and 90 day prediction respectively for Yahoo!Inc. Data set** for each iteration:

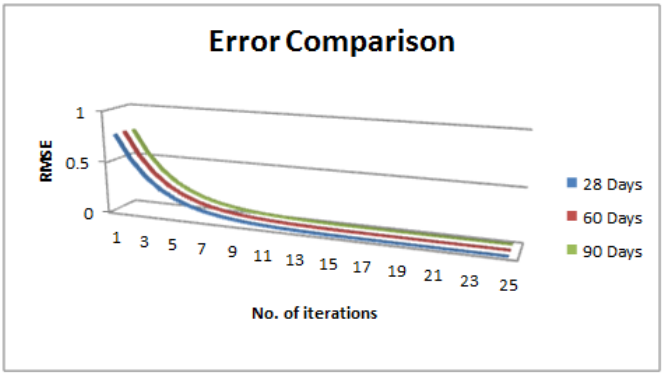

Fig.4: RMSE plot for Yahoo! Inc. data set

It is clear from these graphs that there is a decreasing trend of RMSE value as the number of iterations increases.

The parameters that are used to evaluate the robustness of a binary classifier are accuracy, precision and recall (also known as sensitivity and specificity). These parameters are calculated by **Confusion Matrix**. The formula to calculate these parameters are given below:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

$$specificity = \frac{tn}{tn + fp}$$

where,
tp = Number of true positive values
tn = Number of true negative values
fp = Number of false positive values
fn = Number of false negative values; the accuracy results for the two data sets has been tabulated below:

Results				
Days	accuracy	precision	recall	specificity
60	0.879918	0.773997	0.856182	0.890330
90	0.897095	0.756569	0.888198	0.901730

Table 1: Accuracy results for 60 & 90 days for Apple Inc.

Results				
Days	accuracy	precision	recall	specificity
28	0.9995	1.0	0.99916	1.0
60	0.99918	0.99915	0.99915	0.99920
90	0.99917	1.0	0.9982	1.0

Table 2: Accuracy Results for 28, 60 & 90 days for Yahoo! Inc.

6.1 Area Under ROC Curve

In statistics, a **receiver operating characteristic(ROC)**, or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection[1] in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm[1] and can be calculated as (1 - specificity). The ROC curve is thus the sensitivity as a function of fall-out. In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the cumulative distribution function (area under the probability distribution from $-\infty$ to the discrimination threshold) of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability in x-axis.

There are four possible outcomes from a binary classifier. If the outcome from a prediction is p and the actual value is also p, then it is called a true positive (TP); however if the actual value is n then it is said to be a false positive (FP). Conversely, a true negative (TN) has occurred when both the prediction outcome and the actual value are n, and false negative (FN) is when the prediction outcome is n while the actual value is p. To draw a ROC curve, only the true positive rate (TPR) and false positive rate (FPR) are needed (as functions of some classifier parameter). The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.

A ROC space is defined by FPR and TPR as x and y axes respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). Since TPR is equivalent to sensitivity and FPR is equal to 1 - specificity, the ROC graph is sometimes called the sensitivity vs (1 - specificity) plot. Each prediction result or instance of a confusion matrix represents one point in the ROC space.The diagonal divides the ROC space. Points above the diagonal represent good classification results (better than random), points below the line represent poor results (worse than random).

We can infer that our model performed significantly well as compared to non-ensemble techniques as well as other ensemble techniques for a short term window of 28 days as well as long term window of 60 and 90 days respectively.

For the Apple data set used, the time window taken was 60 and 90 days for predicting the results which gave better results compared to non-ensemble techniques and other ensemble techniques. The above four curves showcase the performance measure of the algorithm. The diagonal line in the graph is the threshold line which basically shows that if the performance curve is above the diagonal line or in other words higher is the curve above the diagonal line more is the predicted accuracy and vice versa.

7. DISCUSSION AND CONCLUSION

The algorithm used here has to be checked for its robustness. this is accomplished by checking the accuracy in predicting the result or it can be achieved by analyzing the receiver operating characteristics or the ROC curve. It is evident that Xgboost has outperformed the metric as well as other

non-metric classifiers in accuracy. From Fig.9 and 10 it is clear our model gave higher accuracy.

According to the area under ROC curve obtained for 60 and 90 day prediction from Fig.5 and Fig.6, results are clearly better than the previously used machine learning methods(*AUC for 60 day is 0.8435109 and for 90 day prediction XGboost gives AUC of 0.7127071*). Fig 7and Fig 8 (*AUC for 28 day is 0.94 and for 60 day prediction XGboost gives an AUC of 1.0*) show the same evidence. Also the 90 day prediction on Yahoo! data set gave AUC of 1.0. Logistic Regression poorly performs as compared to SVM and Xgboost. Also from the bar graph, it is evident that SVM performs well with an accuracy of nearly 80% but Xgboost however outrules SVM in terms of Accuracy(≈88%).

In the published paper[**10**] on ANN(Artificial Neural Network) which predicted for i^{th} day, results showed an accuracy of 81% which is less than the accuaracy obtained from our given model. Likewise, in [**11**], Random Forest (RF) model could predict stock market movement direction with an accuracy of 78.81% which is still lesser than Xgboost's predicted accuracy.

The remarkably high accuracy of prediction in the case of Yahoo! Inc. data set could be a matter of concern. Natural suspicion about inherent bias in training data could arise. However, we have checked the data set and confirm the non-existence of heavy bias of data set. The proportion of positive and negative data are in range of 45:55.

The literature survey helps us conclude that Ensemble learning algorithms have remained unexploited in the problem of stock market prediction. We have used an ensemble learning method known as XGBoost to build our predictive model. The comparative analysis testifies the efficacy of our model as it outperforms the models discussed in the literature survey. We believe that this is due to the lack of proper data processing in **[07][15][16]**. In this paper, we have performed exponential smoothing which is a rule of thumb technique for smoothing time series data. Exponential smoothing removes random variation in the data and makes the learning process easier. To our surprise, very few papers found in the literature survey exploited the technique of smoothing. Another important reason could be the inherent non linearity in data. This fact discourages the use of linear classifiers. However in **[15]**, the authors have used linear classifier algorithm as the supervised learning algorithm which yielded a highest accuracy of 55.65%. We believe that the use of SVM in **[7]** and **[13]** is ill-advised. Due to that fact that the two classes in consideration (rise or fall) are linearly inseparable, researchers are compelled to use SVM with non linear kernels such as Gaussian kernel or Radial Basis Function. Despite many advantages of SVMs, from a practical point of view, they have some drawbacks.Considering these arguments we can also conclude why the prediction by these classifiers were limited to a maximum of 44 day time window which qualifies our learning model to surpass all these metric classifiers in terms of long term prediction. An important practical question that is not entirely solved, is the selection of the kernel function parameters - for Gaussian kernels the width parameter σ - and the value of ϵ in the ϵ loss insensitive function **(Horvath (2003) in Suykens et al.)**.

Predicting stock market due to its non linear, dynamic and complex nature is really difficult. However in the recent years, machine learning techniques have proved effective in stock forecasting. Many algorithms such as SVM, ANN etc. have been studied for robustness in predicting stock market. However, ensemble learning methods have remained unexploited in this field. In this paper, we have used XGBoost classifier to build our predictive model and our model has produced really impressive results. The model is found to be robust in predicting future direction of stock movement. The robustness of our model has been evaluated by calculating various parameters such as accuracy, precision, recall and specificity. For all the datasets we have used i.e, Apple and Yahoo, we were able to achieve accuracy in the range 87-99% for long term prediction. ROC curves were also plotted to evaluate our model. The curves demonstrate the fidelity of our model graphically.

Our model can be used for devising new strategies for trading or to perform stock portfolio management, changing stocks according to trends prediction. In future, we could build boosted tree models to predict trends for really short time window in terms of hours or minutes. Ensembles of different machine learning algorithms can also be checked for its robustness in stock prediction. We also recommend exploration of the application of Deep Learning practices in Stock Forecasting involving learning weight coefficients on large, directed and layered graphs.

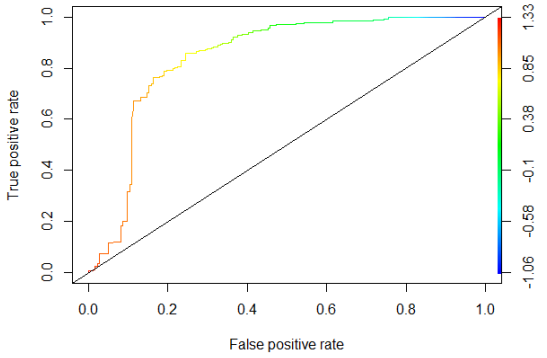

Fig.5: ROC Curve for 60 days(Apple Inc.) **AUC is 0.8435**

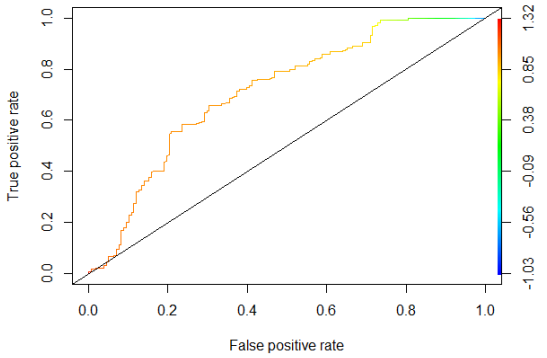

Fig.6: ROC curve for 90 days(Apple Inc). **AUC is 0.7127**

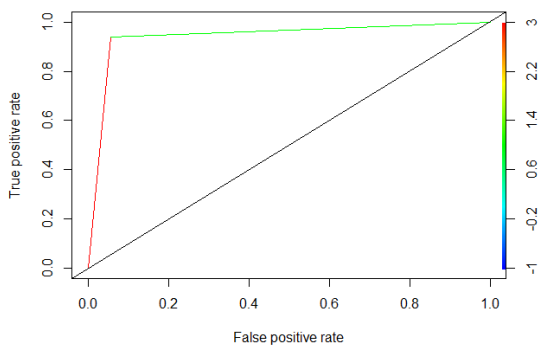

Fig.7: ROC curve for short term prediction of 28 days(Yahoo! Inc.). **AUC is 0.94**

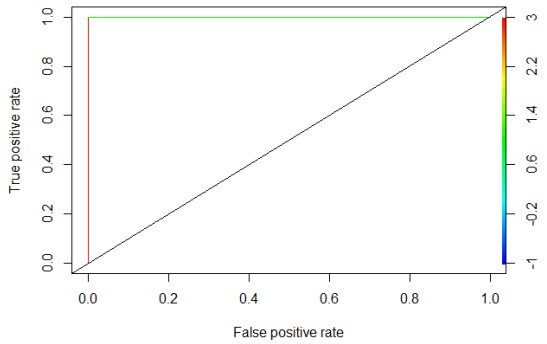

Fig.8: ROC curve for 60 days(Yahoo! Inc.). **AUC is 1.0**

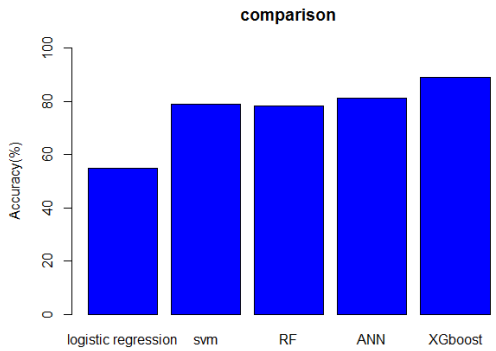

Fig.9: Result for Long term Prediction on Apple Inc. : XGBoost beats all other predictive algorithms reported in literature.

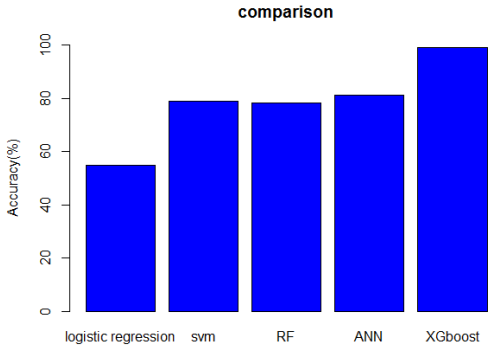

Fig.10: Result for Long term Prediction on Yahoo! Inc. : XGBoost beats all other predictive algorithms reported in literature by quite a margin.

The proposed model indicates, to the best of our knowledge, the nonlinear nature of the problem and the futility of using linear discriminant type machine learning algorithms. The accuracy reported is not pure chance but is based solidly on the understanding that the problem is not linearly separable and hence the entire suite of SVM type classifiers or related machine learning algorithms should not work very well. The solution approach adopted is a paradigm shift in this class of problems and minor modifications may work very well for slight variations in the problem statement.

8. REFERENCES

[1]Das Shom Prasad,Padhy Sudersan, Support Vector Machines for Prediction of Future Prices in indian Stock Market.*International Journal of Computer Applications(0975-8887)*,March 2012.

[2]Chauhan Bhagwant,Bidave Umesh, Gangathade Ajit, Kale Sachin Stock Market Prediction Using Artificial Neural Networks.*International Journal of Computer Science and Information Technology,Vol 5(1),2014,904-907*

[3]Kar Abhishek Stock Market Prediction using Artificial Neural NetworkS. *Y8021*

[4]S. R. Y. Mayankkumar B Patel, Stock prediction using artificial neural network, International Journal of Innovative Research in Science, Engineering, and Technology, 2014.

[5]Mingyue iu and u Song .Predicting the Direction of Stock-Market Index Movement Using an Optimized Artificial Neural Network Model .*Published online 2016 May 19. doi: 10.1371/ journal.pone.0155133*

[6]Elisa Siqueira, Thiago Otuki,Newton da Costa Jr Stock Return and Fundamental Variables:A Discriminant Analysis Approach.*Applied Mathematical Sciences, Vol. 6, 2012, no. 115, 5719 - 5733.*

[7]Yuqing Dai, Yuning Zhang (2013). Machine Learning in Stock Price Trend Forecasting. Stanford University.

[8]Hellstrom, T., Holmstromm, K. (1998). Predictable Patterns in Stock Returns. Technical Report Series IMa-TOM-1997-09 .

[9]R. Gencay, Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules, *Journal of International Economics, vol. 47,no., pp. 91-107,19.*

[10]A. Timmermann and C. W Granger, Efficient market hypothesis and forecasting, *International Journal of Forecasting, vol. 20,no., pp. 15- 27,2004.*

[11] Sadegh Bafandeh Imandoust and Mohammad Bolandraftar. Forecasting the direction of stock market index movement using three data mining techniques: the case of Tehran Stock Exchange*S. Bafandeh Imandoust Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 4, Issue 6(Version 2), June 2014, pp.106-117*

[12]Yuqinq He, Kamaladdin Fataliyev, and Lipo Wang, Feature Selection for Stock Market Analysis.

[13]Phichhang Ou,Hengshan Wang. Prediction of Stock Market Index Movement by Ten Data Mining Techniques.

[14]Khan,W., Ghazanfar,M.A., Asam,M., Iqbal,A., Ahmed,S., Javed Ali Khan. Predicting Trend In Stock Market Exchange Using Machine Learning Classifiers. *Sci.Int(Lahore), 28(2), 1363-1367, 2016*

[15]Haoming Li, Zhijun Yang and Tianlun Li (2014). Algorithmic Trading Strategy Based On Massive Data Mining. Stanford University.

[16]Xinjie (2014). Stock Trend Prediction With Technical Indicators using SVM. Stanford University.