Stock Market Analysis using Stacked Ensemble Learning Method

MALHAR TAKLE

Applied project submitted in partial fulfilment of the

requirements for the degree of

MSc. In Data Analytics

at

Dublin Business School

Supervisor: Ms. Terri Hoare

Date: August 2020

DECLARATION

I, Malhar Ajit Takle, declare that this research is my original work and that it has never been presented to any institution or university for the award of Degree or Diploma. In addition, I have referenced correctly all literature and sources used in this work and this work is fully compliant with the Dublin Business School's academic honesty policy.

ACKNOWLEDGMENTS

I would like to express my special thanks to Ms. Terri Hoare, my teacher and research supervisor for her patient guidance from the beginning with constructive and valuable suggestions provided throughout the planning and improvement of my dissertation. She also helped me in doing a lot of research, and her ability to give her time generously has been particularly valued.
Secondly, I would like to thank my family and friends for their support and encouragement throughout my course.

Table of Contents

List of Figures

List of Tables

Abstract

Predicting stock market trends is a challenging problem. Technical and Fundamental analysis are traditionally used by traders to analyse the stock market. The decision as to whether to buy or sell a share in stock requires fast decision-making involving large investments. There are many competing and volatile contributing factors. Recent studies have shown that the use of ensemble modelling techniques enhances the performance of individual weak base learners in predicting stock movements. This research uses a supervised learning classification approach based on the percentage returns for the previous 5 days of a stock to predict trend directions. State-of-the art ensemble techniques including boosting, bagging and H2O stacked ensembles are compared. Empirical results show that the stacked ensemble with Gradient Boosting, XG Boosting and Random Forest achieves higher AUC scores than those of individual classifiers for all of the three indices: S&P 500, NASDAQ and NYSE.

List of important abbreviations

- CRISP DM – Cross-industry standard process for data mining
- S&P 500 – Standard & Poor's 500 Index
- NASDAQ – National Association of Securities Dealers Automated Quotations
- NYSE – New York Stock Exchange
- AdaBoost – Adaptive Boosting method
- GB – Gradient Boosting method
- XGBoost – eXtreme Gradient Boosting method
- DT – Decision Tree
- RF – Random Forest
- AUC – Area Under Curve
- TSF – Time Series Forecasting
- SMOTE – Synthetic Minority Over-sampling Technique

Chapter 1 - Introduction

1.1 Background

Predicting about the future has always been a fascination for human beings. Similarly, large number of masses are attracted to and try their luck in stock markets all over the world. Mainly, two ways are used for the prediction of stock markets called as Fundamental and Technical Analysis. Fundamental analysis is based on real world economic indicators whereas Technical analysis is purely based on statistical calculations & techniques. Analysing the Big Data related to stock markets is very difficult as the market prices are affected by multiple factors simultaneously (Bousono-Calzon *et al.*, 2019; Nti *et al.*, 2019; Nti *et al.*, 2020).

Many people rely on the trading signals and experts' advice for making their Buy or Sell decisions in the stock markets. Unfortunately, maximum number of traders fail to predict the right directions of the stock or overall index. This results into huge losses and damage to them. In the past, many attempts have been made to predict the unpredictable stock market prices and direction based on traditional analysis as well as machine learning models. But still there is a need of solutions to help the traders to take data-driven decisions.

1.2 Research Problem

Stock Market analysis is very crucial for those involved in the financial markets directly and indirectly. Also, it can impact the economy of a country. It aims to predict the future prices of the financial markets as well as the trends. Since the existing research suggests that forecasting of exact stock prices accurately is difficult, this research aims to focus on predicting the direction of the markets. Taking into consideration the advanced machine learning techniques, this research will be conducted on stacking ensemble classification methods. The research papers related to pattern recognition (Ho *et al.*, 1994; Frosyniotis *et al.*, 2003) indicate that ensembled classifiers work very well as compared to individual. But these techniques aren't explored much in the field of stock markets with the angle of classification problem.

Research Question – Comparison of stacked ensemble learning and other ensemble techniques for the trend prediction of US stock markets.

Aim – Accurately forecast the direction of stock market based on percentage change in the prices of previous 5 days.

Objective – To compare various machine learning classification models and stacked ensembles with an aim to identify efficient modelling technique for predicting the direction of US stock markets.

Hypothesis – State of the art H2O Stacked Ensemble models perform better than the individual machine learning algorithms and traditional ensemble learning techniques.

1.3 Scope

The scope of this research is to help build a trading software for accurately forecasting the stock markets trends based on efficient machine learning techniques. List of all the algorithms and methods used-

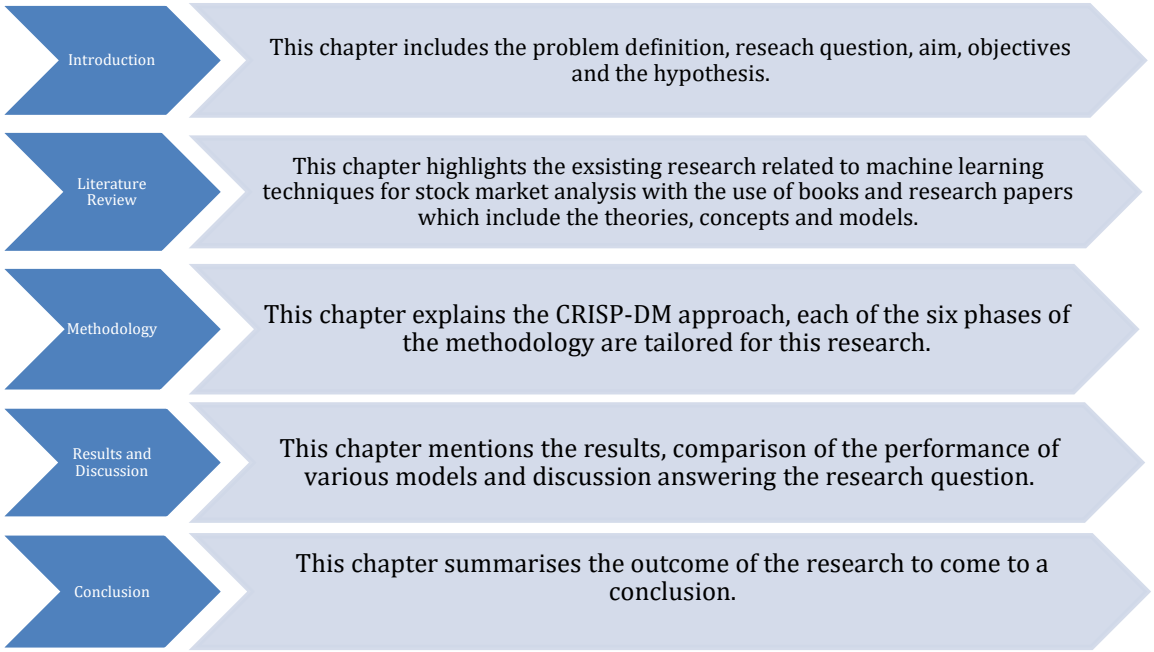
- Random Forest Classifier
- Decision Tree
- Adaptive Boosting
- Gradient Boosting
- XG Boosting
- Stacked Ensemble Learning
- H2O AutoML Feature

1.4 Limitation

The different machine learning models and ensemble learning techniques are applied to only 3 major stock indexes S&P 500, NYSE and NASDAQ for better computation and comparison among them. Also, there is a limitation of computing power as this research is conducted on HP Laptop with 8GB RAM and Core i3 processor. Thus, the Google Colab service was used to run the programs on Google Cloud.

1.5 Dissertation Roadmap

The dissertation project was implemented strategically as per the following roadmap-



Chapter 2 - Literature Review

The current research conducted for forecasting the trend of the stock markets is being investigated to identify the efficient machine learning algorithms and techniques. Many researchers have tried using different types of machine learning models and statistical algorithms for various stock market data. Drawing a conclusion regarding the best performing techniques and models for analysis of financial markets is not at all easy. In order to understand the concepts and machine learning models implemented for stock market prediction, the following literature was referred.

In the book "An Introduction to Statistical Learning", the authors have used the approach to predict the direction of the stock market instead of the actual prices. Whether the index will increase or decrease on a given day is predicted based on the lags of previous 5 days by considering it as a classification problem. Various models like Logistic regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis and K-Nearest Neighbours were implemented. Amongst all of them the best prediction accuracy of 60% was observed with QDA model for S&P 500 data between 2001-2005 (James, G., Witten, D., Hastie, T., Tibshirani, 2013).

In the research conducted by (Kumar *et al.*, 2018), five machine learning models are implemented to forecast the direction of stock prices. Various algorithms like Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Naive Bayes, SoftMax and Random Forest have been applied on the stocks of companies like Amazon, Bosch, Bata and Cipla, Eicher. Among which there are two different sizes of data sets large and small respectively. The models were trained with the values of different technical indicators calculated on the basis of prices. On comparing the performance of models, it was concluded that Random Forest works more accurately with large datasets while with small datasets the Naïve Bayes model gives highest accuracy. The results also showed that accuracy of prediction decreased with the decrease in number of inputs i.e. the technical indicators.

The research (Soni *et al.*, 2018) was aimed to understand the use of statistical algorithms and machine learning techniques for analysing the stock markets. It suggests that traditional methods used for analysis like technical, fundamental, time-series or statistical haven't worked well and were not accepted widely for prediction.

Models like Naïve Bayes, Decision Tree were implemented to analyse the trends of the markets and compared their performance. Further they proposed the method of using Ups and Downs in the daily change of markets by considering the means of change and predicting the next day. Thus, continuing further with the same approach for a large dataset by dividing it into train and test set, they observed progress in the accuracy. It helps to understand that the approach of using percentage change to predict the future days seems to be simple but proves useful if implemented with statistical algorithms. Also, the authors have stated a need of developing a technique to predict the trends to maximise the profits and reduce the risks out of the sudden rise or fall in the stock prices.

As a modern world technique, many attempted to automate the process of trading but prediction still remained an unsolved puzzle. Thus, a research (Nair *et al.*, 2010) was conducted to developing an automated system based on multiple algorithmic techniques. In this the hybrid prediction model based on traditional technical analysis, decision tree, dimensionality reduction and Adaptive neuro fuzzy inference system was proposed. On training and testing the model for daily time frame data from stock markets like BSE-SENSEX, FTSE 100, NASDAQ 100 and NIKKEI 225, it was observed that the hybrid model gives more accurate predictions than the individual techniques. Therefore, it indicates the future scope for implementing ensemble learning and mixed modelling.

Considering the fact that multiple external factors affect the prices of stock indices, the conference paper (Mehak Usmani,Syed Hasan Adil,Kamran Raza, 2016) proposed an approach based on different types of factors as inputs to the various machine learning models. The input variables were Oil rates, Gold Rates, Silver Rates, FEX, SMA, ARIMA, KIBOR, news & Twitter feed and the output for defined in two classes as Positive (Up) and Negative (Down). The machine learning algorithms like Single Layer Perceptron, Multi-Layer Perceptron, Radial Basis Function (RBF) and SVM were applied on the Karachi Stock Exchange data. Among the above, MLP performed better than others in terms of accuracy. The research also noted that oil prices had most impact on the KSE index price than other factors and the trend can be predicted with the machine learning models.

The research (Agarwal *et al.*, 2020) has dealt with the stock markets with two approaches of predicting price and trends. They have applied the artificial recurrent neural network architecture Long Short-Term Memory for Time Series analysis to forecast the price whereas Random Forest Classifier was used to understand the direction of the market. The RF model was implemented on the data of news headlines about State Bank of India to predict the trend of their stock value. The results show that the model performed with a 67% accuracy to predict the trends based on the sentimental analysis of the news. It helps us to understand that RF classifier can be used for prediction of trends in the stock market.

(Muhammad zulfiqar Umer,Muhammad Awais, 2019) research suggests the need of predicting the trends to avoid huge losses in highly volatile markets. The machine learning techniques algorithms like Linear Regression (LR), Three month Moving Average(3MMA), Exponential Smoothing (ES) and Time Series Forecasting (TSF) were implemented on the data of stocks of technology companies like Apple, Amazon and Google. After comparing the results of all the algorithms, it was found that ES performed best on all the stocks.

The research (Paliyawan, 2015) is aimed to predict the future trend of Stock Exchange of Thailand (SET) based on past data. The suitable time interval was chosen after time series analysis for the experiments. Decision Tree, Naïve Bayes and KNN were applied to the data with an approach to predict the patterns in the charts. It was observed that Time Series Forecasting doesn't perform well with the SET data and so the classification modelling techniques were proposed to be used with previous five days of index price to predict the next day. On training and testing the models, the DT model outperformed others with more accuracy in forecasting the fall of the market. The True Negative Rate was 51.60% whereas False Negative Rate was 14.10%. The researchers also deep dived into the model to find 18 different patterns in the charts which can be used in future to trade.

The approach of classification for prediction of stock market trend in order to reduce the error is proposed in (Basak *et al.*, 2019). The machine learning models are built on the values of traditional technical indicators like Price Rate of Change, On Balance Volume, Moving Average Convergence Divergence, Relative Strength Index, Stochastic oscillator which are derived from the stock prices data after exponential

smoothing. Ensemble learning technique was implemented based on RF classifiers & Gradient Boosted Trees using XG Boost to the stocks data of 10 different companies from different countries like Apple, Microsoft, Tata, Facebook, etc. Impressive accuracy was observed in the results with both the models and the robustness of the models was validated with Receiver Operating Characteristic curve & Out-of-bag error visualisation. On comparing the results, the accuracy of the ensemble learning techniques was much higher than the individual non-ensemble methods. Thus, it helps to understand the merits of ensemble machine learning models using random forests (Bagging)and XG Boost techniques.

(Ballings *et al.*, 2015) study was done with an objective of evaluating ensemble methods like RF, AdaBoost, Kernel Factory against individual classifier algorithms like Neural Networks, LR, SVM and KNN. The models were trained and tested on data of more than 5000 European companies. In order to compare the performance of different techniques, AUC was considered as the standard metric. Among all the models, Random Forest was the best performer with highest median AUC. Also, the other ensemble techniques stood in top four models in performance. As suggested by the researchers, this paper will explore further possibilities with ensemble learning techniques for the trend prediction of major stock market indices from United States of America. The researchers also indicate that a little progress in performance of prediction models with regards to stock markets will yield significant profits.

The research (Winkler *et al.*, 2016) aims to predict short term trend based on the data of 10 major stocks in each sector including the index of Spanish Stock Market between 2003 – 2013 with the approach of heterogenous ensembles. NN, KNN, SVM, genetic programming, DT and RF models were used as the learning classifiers. All the models were optimised with 10- fold cross validation technique. On applying heterogenous ensemble methods for short-term predictions, rise in the accuracy of classification was observed. Also, the accuracy for forecasting next day and next month trends was higher than next week. Thus, ensemble methods prove useful in increasing the accuracy than the single classifiers.

(Dey *et al.*, 2016) research was conducted to build a model for forecasting the direction of stock market value with the use of XGBoost technique. The data sets of stock prices of Apple Inc. and Yahoo Inc were taken from Yahoo finance website. On

implementing the model for forecasting the direction of both the stocks for 28, 60 and 90 days it was observed that the value of Root Mean Squared Error (RMSE) was declining with the rise in iterations. The robustness of model was verified by plotting ROC curves and further from the accuracy metric it was found that the model proposed by using ensemble learning performed better than Logistic Regression, SVM, Artificial Neural Networks models.

(Weng *et al.*, 2018) research analyses and compares different ensemble models like Quantile Regression (QR) RF, QR Neural Network, Bootstrap Aggregating regression and boosting regression. Although it uses the regression approach, the outputs suggest that there is an improvement in performance with the use of ensemble techniques in comparison to traditional TSF.

(Tsai *et al.*, 2011) research was based on ensemble learning based on different classifiers for forecasting stock market returns. It also states the upliftment of accuracy and performance of ensemble learning over single classifiers. The researchers have focussed on comparing the homogenous and heterogenous ensemble with respect to stock returns prediction. The dataset was used for the research was taken from Taiwan Economic Journal related to electronic sector which constitutes the more than 70% of Taiwan Stock Market. It is observed that there is no cognizable variation between the performance of homogenous & heterogenous ensemble classification techniques considering Bootstrap Aggregating and Voting by majority. The research still concludes with betterment of accuracy with ensemble methods as compared to single classifiers.

(Gyamerah *et al.*, 2019) research compares the performance of the individual machine learning classification models – AdaBoost & KNN and stacked ensemble classifier. Gradient Boosting Machine was used as a meta learner. The stocks data from Nairobi Stock Exchange was used and classified into Buy or Sell classes based on the change in closing price. The features used for the prediction of stocks direction were the change in the price low & high, the change in the closing & opening price, the market cap i.e. value of the company and the volume of the market. The researchers have also mentioned the approach of using AUC to compare the performance of different classifiers. As stated, AUC in case of binomial classification should be higher than 0.50 otherwise it is not better than the probability of random chance. The accuracy of

Stacking, AdaBoost, KNN was 0.7810, 0.7150, 0.7770 respectively and the AUC was 0.8238, 0.7454, 0.8193 respectively. It clearly indicates that Stacking Ensemble model performed better than the individual classifiers.

(Jiang *et al.*, 2020) research was recently conducted on stock index movements with the data from 3 US stock markets and proposes stacking ensemble model by combination of tree based and deep learning ensemble techniques. It mentions about the better results given by their proposed stacked ensemble model than the individual ensemble models.

In the book (Kotu and Deshpande, 2018), the authors explain about the working of ensemble modelling. Ensemble models are simply a combination of multiple learners which perform independently with different approaches for the same data. Simplifying the method more they state the similarity with the wisdom of the crowd which is nothing but opinion of a group of individuals. It also explains the detailed working of ensemble by voting, bagging and boosting. As mentioned by the authors, ensemble learning techniques are widely used in the data science industry for building models. The famous machine learning models Random Forest and Gradient Boosting Machine are examples of ensemble learners based on Bagging & Boosting techniques respectively.

(Van Der Laan *et al.*, 2007) is a significant research which backs the concept of stacking. The researchers proved that the stacked ensemble method provides an optimisation in learning and also avoid over-fitting with cross validation technique.

H2O.ai is an open source software company which invented the H2O platform for the field of data science and machine learning. In the documentation on their website, they have mentioned about the inclusion of Stacked Ensemble models in the AutoML (Automatic Machine Learning) feature. Also, they have stated that their stacked ensemble models outperform the other models.

Taking the literature review into consideration, it is clear that most of the research are done based on implementation and comparison of traditional statistical algorithms and modern machine learning models for prediction of stock markets. The research also states that modern techniques are proving more accurate and robust for the unpredictable stock indices data from various stock markets all over the world. A research also mentions about no significant difference in the performance of

heterogenous and homogenous ensemble learning models. The stacked ensemble method is not exploited much for the prediction of stock market movement with approach of classification problem. Based on the existing research conducted in the field of stock markets, the following methods are selected to implement for this research-

1. Boosting Techniques- Adaptive Boosting, Gradient Boosting, Extreme Gradient Boosting
2. Random Forest
3. Stacked Ensemble Methods by H2O
4. AutoML (Automatic Machine Learning) by H2O

Chapter 3 - Methodology

The research was implemented as per the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology. It was proposed by the famous research (Shearer, 2000). It gives a standard step by step way of conducting research and implementing any Data mining project. The following phases were conducted in this research –

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modelling
5. Evaluation
6. Deployment

CRISP DM Methodology can be explained with the following diagram:

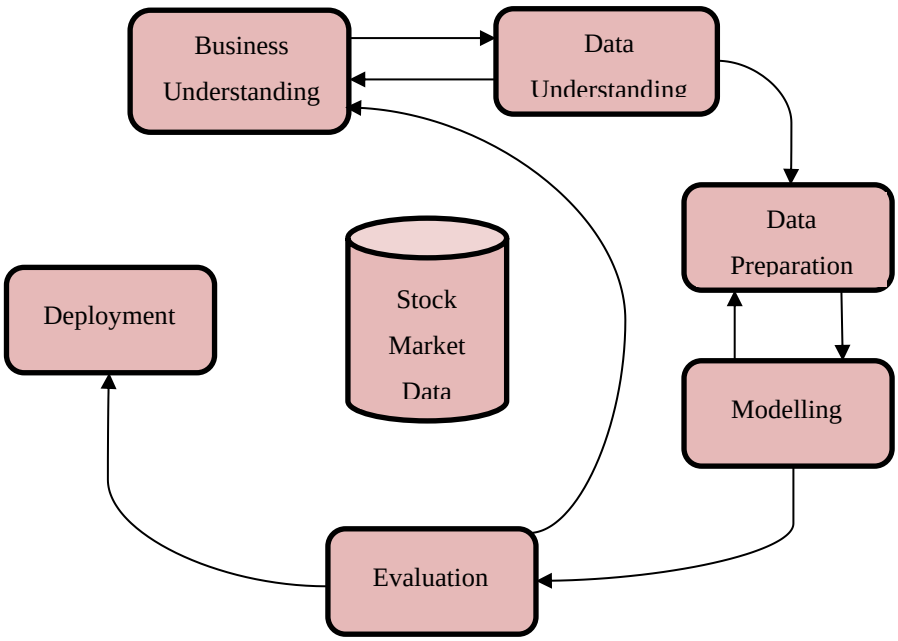


Figure 3.1. CRISP DM Methodology Diagram

All the phases of CRISP DM methodology are observed in detail with regards to the research.

1. Business Understanding:

It is an important and very first step of the research methodology. It focuses on studying in detail about the goals of a project in terms of business approach and also processing the business approach as a research problem. As the further steps are planned accordingly, it is very important to deep dive in the business problem during the business understanding phase.

The objective of this research is to forecast the direction of the stock market indices for making the investment or trading decisions. Generally, there are three options with the traders like Sell, Buy and Hold. But, in this research only Buy and Sell are considered as they are the main decisions in the stock markets. It is considered that even a small improvement in the prediction of the stock market direction can result into significant profits to the investors/traders. For predicting trend of stock index on the next day based on the percentage change in the price, the approach stated in the research (James, G., Witten, D., Hastie, T., Tibshirani, 2013) is used. As per the traditional methods used by traders, it is relevant to Price Action Trading Strategy. Various types of machine learning models are used to predict the stock market prices and the trends. As per the traditional methods the traders use technical analysis indicators like Moving Averages to predict the trends of the markets. With the progress of machine learning, it is really an interesting experience to apply modern techniques and algorithms for solving the real-world problem of stock market prediction. In simple words, the trader will decide to Buy if the market trend is predicted to be Up and Sell if the trend is predicted to be Down. As a business application, a successful modelling technique can be used to build an automated software which will predict and supply trading signals based on the forecast of direction in a real time environment. In this research, the business problem of forecasting the stock markets is approached as a classification problem with a focus on trends of the movement. Thus, the ensemble classifiers and stacked ensemble methods are chosen after the literature review. The aim of this research is to compare the different tree-based ensemble classifiers and use the stacked ensemble technique for different datasets of three major US Stock Market Indices including S& P 500 index, NASDAQ composite and NYSE composite. Also, the novel feature AutoML provided by H2O platform is implemented in this research

to find the top performing models for all three indices. This study is conducted to solve the research question “Can stacked ensemble learning techniques be used to predict stock market direction?”.

2. Data Understanding:

In order to understand the data well, first it needs to be collected. After the data are gathered, exploratory analysis is conducted to identify problems and features of the data. The data used for training and testing the different models is acquired from Yahoo Finance Website https://finance.yahoo.com/. This study uses the historical data for daily timeframe of all 3 United States stock market indices- NASDAQ, S&P500 and NYSE between the period - Jan 1985 to June 2020.

The datasets contain 8946 rows of stock index values recorded on daily basis and 7 columns. The columns in the raw data are as follows:

- Date – Date of the record
- Open – Opening price of the given period
- High – Highest price of the given period
- Low – Lowest price of the given period
- Close – Closing price of the given period
- Adj Close – Adjusted closing price after the corporate actions
- Volume – Total number of shares traded

As this research is not based on Time Series Analysis, the Date column is dropped. Also, the Adjusted Closing price is used as the main value of the stock index. The Adj Close gives the most accurate value of the stock as it is calculated after the alterations caused due to events like stock splits, dividend distribution or new stock offerings. Considering the business logic and the approach given in (James, G., Witten, D., Hastie, T., Tibshirani, 2013), all the columns except the Adj Close are dropped. The inputs given to the models will be derived from the selected column.

3. Data Preparation:

The third phase is about making the data ready to build the models. There are various actions performed like transformation, feature selection, data cleaning, etc. As per the approach chosen for this research, the following pre-processing steps are done:

- Similar to the Smarket Dataset used in the (James, G., Witten, D., Hastie, T., Tibshirani, 2013), the daily percentage returns are calculated by using pct_change() function from Pandas library in Python. It is stored in a new column named 'Today'.
- Considering the classification method used, the target variable is derived from the percentage returns by dividing positive percentage returns as 1 or Up and negative percentage returns as 0 or Down. It simply means increase or decrease in the value of the stock index. A new column is created to store the target variable called as Direction.
- During first trials of the code, it was observed that the models were getting trained with a bias. After counting the records of both the classes, a difference was found. In order to avoid biased training of the model, Synthetic Minority Over-sampling Technique (SMOTE) was implemented by using SMOTE() function from imblearn.over_sampling library. After over sampling, equal number of values were counted for both classes ensuring unbiased training of the model. Oversampling techniques create artificial values in the dataset and can cause impact on the performance of the model. In order to avoid the adverse effect of over sampling, it has been done before creating the lag features.
- Similar to the Smarket dataset, the lag features are created for previous five days by using the shift() function in Pandas library of Python. To store the values, Lag_1 to Lag_5 columns are created.

	Direction	lag_1	lag_2	lag_3	lag_4	lag_5
5	1.0	0.007256	-0.001522	0.003421	-0.005408	-0.004838
6	0.0	0.018949	0.007256	-0.001522	0.003421	-0.005408
7	1.0	-0.002377	0.018949	0.007256	-0.001522	0.003421
8	1.0	0.015484	-0.002377	0.018949	0.007256	-0.001522
9	1.0	0.001759	0.015484	-0.002377	0.018949	0.007256
...	...	...	...	...	...	...
9643	0.0	-0.005232	-0.002182	-0.006695	-0.001711	-0.000137
9644	0.0	-0.002398	-0.005232	-0.002182	-0.006695	-0.001711
9645	0.0	-0.003066	-0.002398	-0.005232	-0.002182	-0.006695
9646	0.0	-0.005806	-0.003066	-0.002398	-0.005232	-0.002182
9647	0.0	-0.003208	-0.005806	-0.003066	-0.002398	-0.005232

9643 rows × 6 columns

Figure 3.2. Pre-processed S&P500 Dataset

	Direction	lag_1	lag_2	lag_3	lag_4	lag_5
5	1.0	0.005285	0.000407	-0.000813	-0.001217	0.002033
6	1.0	0.013748	0.005285	0.000407	-0.000813	-0.001217
7	1.0	0.005983	0.013748	0.005285	0.000407	-0.000813
8	1.0	0.013085	0.005983	0.013748	0.005285	0.000407
9	1.0	0.009002	0.013085	0.005983	0.013748	0.005285
...	...	...	...	...	...	...
9951	0.0	-0.004607	-0.030463	-0.022942	-0.006020	-0.020981
9952	0.0	-0.014612	-0.004607	-0.030463	-0.022942	-0.006020
9953	0.0	-0.000977	-0.014612	-0.004607	-0.030463	-0.022942
9954	0.0	-0.004812	-0.000977	-0.014612	-0.004607	-0.030463
9955	0.0	-0.003921	-0.004812	-0.000977	-0.014612	-0.004607

9951 rows × 6 columns

Figure 3.3. Pre-processed NASDAQ Data

	Direction	lag_1	lag_2	lag_3	lag_4	lag_5
5	1.0	0.006643	-0.000837	0.003069	-0.004736	-0.003984
6	0.0	0.017708	0.006643	-0.000837	0.003069	-0.004736
7	1.0	-0.001548	0.017708	0.006643	-0.000837	0.003069
8	1.0	0.014433	-0.001548	0.017708	0.006643	-0.000837
9	1.0	0.002134	0.014433	-0.001548	0.017708	0.006643
...	...	...	...	...	...	...
9627	0.0	-0.005769	-0.003614	-0.004824	-0.004535	-0.012731
9628	0.0	-0.000926	-0.005769	-0.003614	-0.004824	-0.004535
9629	0.0	-0.007235	-0.000926	-0.005769	-0.003614	-0.004824
9630	0.0	-0.002291	-0.007235	-0.000926	-0.005769	-0.003614
9631	0.0	-0.002301	-0.002291	-0.007235	-0.000926	-0.005769

9627 rows × 6 columns

Figure 3.4. Pre-processed NYSE Data

- After the generation of Lag features, the dropna() function is used to remove the null values as a data cleaning step. Null values can adversely affect the performance of the model and some algorithms don't even work with null values.
- Correlation plot is used to analyse the correlation between the features and the highly correlated column 'Today' is removed for better model building.

	Today	Direction	lag_1	lag_2	lag_3	lag_4	lag_5
Today	1.000000	0.645155	-0.025492	0.014837	0.030640	0.011870	0.030740
Direction	0.645155	1.000000	0.017742	0.041456	0.042990	0.054852	0.046160
lag_1	-0.025492	0.017742	1.000000	-0.025513	0.014858	0.030588	0.011834
lag_2	0.014837	0.041456	-0.025513	1.000000	-0.025517	0.014786	0.030562
lag_3	0.030640	0.042990	0.014858	-0.025517	1.000000	-0.025533	0.014773
lag_4	0.011870	0.054852	0.030588	0.014786	-0.025533	1.000000	-0.025569
lag_5	0.030740	0.046160	0.011834	0.030562	0.014773	-0.025569	1.000000

Figure 3.5. Correlation Plot for pre-processed S&P500 Dataset

	Today	Direction	lag_1	lag_2	lag_3	lag_4	lag_5
Today	1.000000	0.661827	0.042889	0.033274	0.056213	0.051723	0.047640
Direction	0.661827	1.000000	0.110274	0.079446	0.078732	0.097523	0.085159
lag_1	0.042889	0.110274	1.000000	0.042870	0.033245	0.056211	0.051638
lag_2	0.033274	0.079446	0.042870	1.000000	0.042864	0.033245	0.056191
lag_3	0.056213	0.078732	0.033245	0.042864	1.000000	0.042863	0.033216
lag_4	0.051723	0.097523	0.056211	0.033245	0.042863	1.000000	0.042861
lag_5	0.047640	0.085159	0.051638	0.056191	0.033216	0.042861	1.000000

Figure 3.6. Correlation Plot for pre-processed NASDAQ Dataset

	Today	Direction	lag_1	lag_2	lag_3	lag_4	lag_5
Today	1.000000	0.636010	-0.002026	0.021275	0.037790	0.005699	0.029698
Direction	0.636010	1.000000	0.041821	0.044766	0.043080	0.045490	0.036776
lag_1	-0.002026	0.041821	1.000000	-0.002029	0.021293	0.037766	0.005678
lag_2	0.021275	0.044766	-0.002029	1.000000	-0.002034	0.021284	0.037766
lag_3	0.037790	0.043080	0.021293	-0.002034	1.000000	-0.002057	0.021273
lag_4	0.005699	0.045490	0.037766	0.021284	-0.002057	1.000000	-0.002048
lag_5	0.029698	0.036776	0.005678	0.037766	0.021273	-0.002048	1.000000

Figure 3.7. Correlation Plot for pre-processed NYSE Dataset

- The data is divided into train and test sets by using train_test_split function from Scikit Learn. Ratio of 80:20 was used for splitting the data which means 80% data is training and 20% of the data is used for testing.
- Hyper Parameter Optimization:

It is a very important step to identify the optimal parameters for all the models during training stage. Selecting the perfect parameters while building the model impact the accuracy and speed of training. This research uses Grid Search for tuning the hyper parameters. The GridSearchCV function is used from the sklearn library for Python. The best possible parameters are identified by performing the cross-validation process. Once the optimal parameters are found, the models can be built with them and trained with the training data. This research uses the Grid Search method to build the classifiers in order to compare them in terms of performance. The Grid Search technique is time consuming but gives good results and thus helpful for building the models with better performance as compared to manual selection of parameters. In a way, it automates the process of parameters selection.

Best Parameters identified were as follows:

S&P 500 –

Adaptive Boosting: {'learning_rate': 0.001, 'n_estimators': 100}

Gradient Boosting: {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 100}

XG Boosting: {'colsample_bytree': 0.7,'gamma': 0.2,'learning_rate': 0.05,'max_depth': 12,'min_child_weight': 7}
NASDAQ –
Adaptive Boosting: {'learning_rate': 0.001, 'n_estimators': 100}
Gradient Boosting: {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 5}
XG Boosting: {'colsample_bytree': 0.7, 'gamma': 0.2,'learning_rate': 0.05,'max_depth': 6,'min_child_weight': 3}
NYSE –
Adaptive Boosting: {'learning_rate': 0.001, 'n_estimators': 100}
Gradient Boosting: {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 50}
XG Boosting: {'colsample_bytree': 0.7,'gamma': 0.4,'learning_rate': 0.05,'max_depth': 6,'min_child_weight': 5}

- In order to perform Binary classification, the response variable is converted into 'category' type.
- In case of H2O models, the H2O frames for train and test data are created with the H2Oframe() function. H2OFrame is identical to DataFrame in Pandas library of Python or data.frame in R studio. The only main difference is that the data is usually not stored in memory, rather it is contained on a H2O cluster. Therefore, a H2OFrame is nothing but an instance to the data. Various operations like can be performed with it for data manipulation. It is the basic storage of data in H2O. The train and test frames were created for further modelling using H2O techniques.

4. Modelling:

- **Rapid Miner Auto Model**:

At the beginning, the Rapid Miner software was used to understand the data and modelling possibilities. The relevant datasets for all three stock markets were prepared and uploaded to the software. The Auto model feature was run for all the data and results were observed based on given performance metrics. It implements all the

models after selecting three requirements of target column, the prediction to be done and type of the problem whether classification or regression. Once all the models are built, various performance measures are presented in a chart which makes is easier to compare the different models based on that particular measure. The Auto Model feature provided by Rapid Miner is really very easy to use and requires no coding skills. However, to deep dive into the research topic the python code was developed for all three datasets by importing the required libraries.

- **Decision Tree Classifier**:

It is the simple Decision Tree algorithm for classification problem. The DT classifier works as an iterative process of partitioning the data till the end. Also, it works on multiple iterations, for practical use the depth of the tree is set to avoid the problem of overfitting. This research uses DT classifier algorithm as the base learner for implementing the ensemble learning methods. Decision Trees are cheaper to build and work very speedily with unknown data.

- **AdaBoost**:

AdaBoost stands for Adaptive Boosting which was first formulated by an award winning research (Freund and Schapire, 1997). Boosting is mainly practiced with an aim to improve accuracy. AdaBoost technique is used to develop a strong algorithm for better performance from a weak base learning algorithm. In this research, DT is used as the base learner. The final function of the Adaboost algorithm is given by equation (1) (Freund and Schapire, 1997).

$$f(x) = sign(\sum_{m=1}^{N} \theta_m f_m(x)) \quad (1)$$

Where, f_m stands for the m^{th} weak classifier and is the corresponding weight. It forms the weighted combination for N different weak learning models.

This research uses AdaBoost Classifier from the Scikit Learn library in Python.

- **Gradient Boosting**:

Gradient Boosting is an ensemble method of Boosting type which works on multiple iterations to develop a strong learning model from weak learner algorithms. It is considered to improve the performance in both types of problems- classification as well as regression. The basic function was statistically formulated by (Breiman, 1997) and further improvised by (Hastie *et al.*, 2001) for regression. After the foundations

were laid, many researchers worked on the Gradient Boosting Algorithm for various applications in the field of statistics.

For this research, Gradient Boosting Classifier is used from the Scikit Learn library and Gradient Boosting Machine (GBM) from H2O platform. The H2O GBM is claimed to be an advanced classification algorithm as per the documentation on their website (H2O.ai, n.d.) . The GBM is an ensemble technique based on forward learning and it can be used as regressor as well as classifier. It is considered to give good results if updated properly with assumptions.

GBM works on following algorithm stated by (Hastie *et al.*, 2009)

Initialise $f_{k0} = 0, k = 1,2, \dots, K$

for m = 1 to M:

Set $p_k(x) = \frac{e^{f_k(x)}}{\sum_{l=1}^{K} e^{f_l(x)}}$, $k = 1,2, \dots, K$

For $k = 1\ to\ K$:

a) $r_{ikm} = y_{ik} - p_k(x_i), i = 1,2,..,N$
b) fit a regression tree to the targets r_{ikm} , $i = 1,2,..N$, giving terminal regions R_{jim},j=1,2,…,Jm $R_{jim}, j = 1,2, \dots, J_m$
c) compute $\gamma_{jkm} = \frac{K-1}{K} \frac{\sum_{x_i \in R_{jkm}} (r_{ikm})}{\sum_{x_i \in R_{jkm}} |r_{ikm}|(1-|r_{ikm})}, j = 1,2, \dots, J_m.$
d) Update $f_{km}(x) = f_{k,m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jkm} I(x \epsilon R_{jkm})$

Output $\hat{f}_k(x) = f_{kM}(x), k = 1, 2, \dots, K$

The performance of GBM is very much affected by some specific parameters and number of columns. It is mainly based on the idea that a best model is built after combining different models and the combination causes reduction in the error. Due to the risk of overfitting the performance and accuracy is measured with respect to the test dataset instead of training data. It works sequentially on the base classifiers. Bias and variance are reduced after gradient boosting is implemented. It is a supervised machine learning. It repetitively works on the model improvement based on the errors by previous predictors. H2O makes it easy to make distribute and parallelize GBM. It works on the distributed trees and each row of the data is assigned with a node.

- **XGBoost**:

XGBoost stands for eXtreme Gradient Boosting and is a powerful tree-based ensemble learning algorithm developed by (Chen and Guestrin, 2016). It has been applied in a wide range of solutions since it was proposed. It follows the gradient boosting framework to implement the machine learning algorithms. The objective function of the XGBoost algorithm is given by equation (2).

$$Obj(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (2)$$

Where, the first part represents the training loss of the model, either logistic or squared, and the second part represents the complexity of each tree.

The main benefits of using XG Boost are that it is very fast as compared to other gradient boosted tree algorithms and the model performance is very good for classification problems.

This research uses the XG boost classifier by importing xgboost library in python and XG Boost Estimator from the H2O platform.

- **Random Forest**:

Random forests were developed by (Breiman, 2001) and are an improved version of decision tree bagging. In this method, several tree predictors are combined based on sampling of a random vector and ensuring that each tree in the forest has the same distribution. The original data is divided into subsets by using the bootstrap method with sample sampling and feature sampling. The optimal split of every tree is created by searching the feature subset randomly. The Gini impurity index is used as a criterion to generate the split. For the purpose of final prediction, the majority vote method is applied on the constructed decision trees. Random forest method is not based on the assumption of a specific distribution and is simple to construct, thus making it suitable for financial classification problems.

This research uses Random Forest Estimator function from the libraries provided by H2O platform.

- **H2O Stacked Ensembles**:

The concept of stacking was originally proposed by the research (Wolpert, 1992) with the title "Stacked Generalisation" and further improved with k-fold cross validation in (Breiman, 1996) which is used currently. The main aim of stacking is to reduce the generalisation error rate and combine the strong & diverse learning models. H2O platform provides a stacked ensemble algorithm which is also called Super Learning. The concept of Super Learner is explained in detail by (Van Der Laan *et al.*, 2007). The following steps are implemented to build a stacked ensemble-

- Frame an ensemble by specifying the list of n base learners and the meta learner.
- Training –

  Each of the base learning algorithm is trained with the training dataset and cross-validated by using k-fold cross validation method. The level 1 data is generated by combining the predicted values by all the base learners for all the rows of training data and the actual response values. The level one data is used to train the meta learner algorithm and further the model is used to predict the response values for test dataset.
- Prediction – The predicted values by base learners are given as input to the meta learner and then the prediction is done by stacked ensemble.

This research uses stacked ensemble (H2O.ai, n.d.) with 3 H2O estimator models namely, Random Forest Estimator, Gradient Boosting Estimator and XG Boost Estimator. The default meta learner algorithm GLM with non-negative weights & no standardisation is used to build the stacked ensemble model (H2O.ai, n.d.).

- **H2O AutoML**:

The H2O AutoML interface automates the process of training a large selection of models to a desired dataset by specifying only a few parameters. The required parameters for AutoML are the response column name, y, and the training set, training_frame. Additionally, one of the stopping parameters are required in order to stop AutoML from running once the parameter value is reached. These parameters are max_runtime_secs, which specifies the maximum time that AutoML will run for, and max_models, which specifies the maximum number of models to build. There are

several optional parameters that can be specified for the AutoML run. The output of the AutoML run shows a leader board of models that were trained during the run. The process of model training using AutoML includes k-fold cross validation by default. All the trained models are presented in order of their ranks which is calculated based on metrics such as AUC for binary classification problems, mean per-class error for multiclass classification problem and deviance for regression problems. Thus, AutoML enables in training a wide variety of models for any given dataset and presents metrics to select the best model. This research uses the AutoML feature from H2O by importing H2OAutoML library in python. Due to the computing limitations of the laptop used for the research, the AutoML feature was practiced by using Google Colaboratory platform. It speeds up the implementation of AutoML and helps to avoid any interruptions caused due to limits of computing power. After the implementation of it, the leader board consisting of top performing models is printed along with the general performance metrics. It helps to compare the different machine learning techniques for the dataset (H2O.ai, n.d.).

5. Evaluation:

As the name suggests, this phase of CRISP DM methodology involves the evaluation of the models built with regards to technical performance as well as business perspective. Only after evaluating the models, the next phase of deployment is done. Evaluation is an important step in terms of application of models to satisfy the business requirements. Especially, in case of stock market trend prediction the evaluation plays a significant role as it can cause huge losses or profits based on the accuracy of the model. Accuracy, AUC, Precision, Specificity and Sensitivity are different types of performance measures used to evaluate the model. As all the metrics have their own advantages and demerits, this research has considered a combination of multiple metrics instead of one.

6. Deployment:

Deployment is the last phase of the CRISP DM methodology and very important for a machine learning project. It is the actual delivery of the models to the end users or the

stake holders. This research project only aims to compare the different models and the stacked ensemble learning method. But as a future scope the models built can be deployed for the real-world use. The goal of the deployment will be to build an automated trading software which will provide real-time signals to the investors and the traders of stock market and thus will help them to make their decisions whether to Buy or Sell. As the deployment step is very important in terms of direct impact on the end users, it is very important to test the stacked ensemble models and approach with different parameters, datasets and combinations of algorithms. Even a minute improvement in the prediction of stock market trends can contribute significantly with profits for the traders and investors in real world.

Chapter 4 - Results and Discussion

This section shows the results of all the applied ensemble techniques for the selected stock market datasets.

- **Results for Boosting Methods:**

Table 4.1. Performance comparison of Boosting Models for S&P 500

	AdaBoost	**Gradient Boost**	**XG Boost**	**Best Score**
Accuracy	0.525677	0.525780	0.518935	Gradient Boost
Precision	0.549132	0.548411	0.532359	Ada Boost
Recall	0.724169	0.615296	0.574657	Ada Boost
F1 Score	0.604435	0.568671	0.546090	Ada Boost
AUC	0.525661	0.525768	0.518931	Gradient Boost

The above table shows that Gradient Boosting Classifier has resulted with highest accuracy for S&P 500 data. Also, in terms of Area Under Curve (AUC) score the Gradient Boosting method outperforms other models. The performance of AdaBoost, Gradient Boost & XG Boost classifiers slightly varies in terms of accuracy, precision and AUC.

Table 4.2. Performance comparison of Boosting Models for NASDAQ

	AdaBoost	**Gradient Boost**	**XG Boost**	**Best Score**
Accuracy	0.541456	0.533717	0.542761	XG Boost
Precision	0.583371	0.584785	0.582143	Gradient Boost
Recall	0.708104	0.706479	0.682394	Ada Boost
F1 Score	0.615339	0.606843	0.605584	Ada Boost
AUC	0.541792	0.533635	0.542701	XG Boost

The above table shows that XG Boost classifier model works better than Gradient Boost and Ada boost Classifiers in terms of accuracy and AUC score for NASDAQ data.

Table 4.3. Performance comparison of Boosting Models for NYSE

	AdaBoost	**Gradient Boost**	**XG Boost**	**Best Score**
Accuracy	0.538725	0.531764	0.512226	Ada Boost
Precision	0.553778	0.556857	0.534250	Gradient Boost
Recall	0.749650	0.750059	0.582852	Gradient Boost
F1 Score	0.624437	0.616908	0.545914	Ada Boost
AUC	0.538276	0.531717	0.512209	Ada Boost

For the NYSE dataset, the Ada Boosting Classifier has outperformed other models in terms of Accuracy and AUC.
Thus, all the three classifier models have performed differently for the three different stock markets. Also, the accuracy is higher than the random guess probability which is 0.50 considering the stock market directions Up and Down.

- **Results for Stacked Ensembles with H2O:**

Table 4.4. Performance of H2O Stacked Ensembles

Stock Index Data	**AUC**
S&P 500	0.5818
NASDAQ	0.6245
NYSE	0.5822

The stacked ensembles were formed with three models-
1. Gradient Boosting Estimator
2. XG Boost Estimator
3. Random Forest Estimator

Considering the AUC scores obtained for each model individually as mentioned in Tables 4.1, 4.2 and 4.3, the performance of stacked ensembles shown in Table 4.4 is better for all three datasets.
These results are in line with the previous research in the area of stacked ensembles.

- **Results for H2O AutoML feature:**

Table 4.5. H2O AutoML Leader board for S&P 500

Model ID	**AUC**
DeepLearning_grid__2_AutoML_20200830_151501_model_1	0.591236367
StackedEnsemble_BestOfFamily_AutoML_20200830_151501	0.582272547
StackedEnsemble_AllModels_AutoML_20200830_151501	0.578606385
XRT_1_AutoML_20200830_151501	0.5739538
XGBoost_grid__1_AutoML_20200830_151501_model_3	0.573930685
XGBoost_grid__1_AutoML_20200830_151501_model_1	0.573881767
GBM_3_AutoML_20200830_151501	0.571225143
DRF_1_AutoML_20200830_151501	0.56942109
XGBoost_2_AutoML_20200830_151501	0.569115219
XGBoost_grid__1_AutoML_20200830_151501_model_2	0.568900194

As per the list of models in table 4.5, the Stacked Ensemble Model from Best of Family and Stacked Ensemble Model from All Models are among the top performers as per the H2O AutoML results. These stacked ensembles are formed with the best performing and all the individual models, respectively. The model

based on Deep Learning algorithm with grid built by AutoML has given the best performance in terms of AUC score. Thus, it is the best model for the S&P 500 dataset as per the above results.

Table 4.6. H2O AutoML Leader board for NASDAQ

Model ID	**AUC**
StackedEnsemble_BestOfFamily_AutoML_20200830_190241	0.631976938
GBM_1_AutoML_20200830_190241	0.630584024
GBM_5_AutoML_20200830_190241	0.630077143
StackedEnsemble_AllModels_AutoML_20200830_190241	0.628803881
GBM_grid__1_AutoML_20200830_190241_model_1	0.628290437
DRF_1_AutoML_20200830_190241	0.626708705
XGBoost_grid__1_AutoML_20200830_190241_model_4	0.622457263
GBM_2_AutoML_20200830_190241	0.620941668
DeepLearning_1_AutoML_20200830_190241	0.619681533
XGBoost_grid__1_AutoML_20200830_190241_model_1	0.618255299

As mentioned in table 4.6, the Stacked Ensemble Model built with the Best of Family models has outperformed all the other models for the NASDAQ dataset with AUC score 0.6319. The Gradient Boosting Machine models have performed slightly lesser in terms of AUC than the leading stacked ensemble. Also, the Stacked Ensemble of all models stood top 4th in the leader board with an AUC score 0.6288.

Table 4.7. H2O AutoML Leader board for NYSE

Model ID	**AUC**
XGBoost_grid__1_AutoML_20200830_202804_model_4	0.584799923
DeepLearning_1_AutoML_20200830_202804	0.583504143
XGBoost_grid__1_AutoML_20200830_202804_model_3	0.580967631
XGBoost_3_AutoML_20200830_202804	0.577946483
StackedEnsemble_BestOfFamily_AutoML_20200830_202804	0.577895753
XGBoost_2_AutoML_20200830_202804	0.575011684
DRF_1_AutoML_20200830_202804	0.573418539
GBM_5_AutoML_20200830_202804	0.572134092
GBM_1_AutoML_20200830_202804	0.572032092
XRT_1_AutoML_20200830_202804	0.571929552

In the table 4.7 the results for NYSE dataset show that the XGBoost Grid 1 Model has performed best in terms of AUC followed by Deep Learning 1, XG Boost Grid 1 and XG Boost 3 models. The Stacked Ensemble model has stood 5th in the leader board but the XG Boost 3 model has slightly performed better than it.

Usually the stacked ensemble models lead the scoreboard and outperform the individual models. But, the above tables 4.5,4.6,4.7 give us little different

insights regarding the performance of the models. As per the AutoML documentation provided by H2O.ai, it can happen when the data is small like the datasets which are used in this research. It will be interesting to check the performance of stacked ensemble models with larger data for the same US stock market indices.

It is evident that the H2O stacked ensemble models perform better than the traditional ensemble techniques and machine learning classifiers for the trend prediction of US stock markets. Thus, using the stacked ensemble learning techniques will improve the profitability in trading the US stock indices by predicting the trends.

Chapter 5 - Conclusion

The comparison of results of machine learning techniques implemented for all the three stock market indices show that the stacked ensemble learning models perform better than individual classifiers. In case of the individual boosting classifiers, the Gradient Boosting Classifier works best for S&P 500, XG Boost works best for NASDAQ and Ada Boost works best for NYSE in terms of Accuracy and AUC scores. The H2O stacked ensembles built with Gradient Boosting Estimator, Random Forest Estimator and XG Boost Estimator achieved AUC scores of 0.5818, 0.6245 and 0.5822 for S&P 500, NASDAQ and NYSE, respectively. Considering the accuracy and AUC scores for all the classifiers and stacked ensembles it is clear that the performance of each technique varies for the different stock indices. The leader board of H2O AutoML function also indicates that the performance of various machine learning models varies for the stock indices but the stacked ensembles are among the leading performers.

Future Research Work

The future work of this research will be varying the number of base learner algorithms with different combinations and parameters for building the stacked ensembles. Conducting a research with similar approach on large datasets for same US stock markets will also be a future step. Also, it will be interesting to observe the performance of stacked ensembles for the trend prediction of different stock markets from various parts of the world. Finally, the development of an automated trading expert advisor software based on the stacked ensembles will be an important future scope.

Plagiarism and Referencing

5.1 Referencing

Agarwal, H., Jariwala, G. and Shah, A. (2020) *Analysis and Prediction of Stock Market Trends Using Deep Learning*. Springer Singapore DOI: 10.1007/978-981-15-3369-3_39.

Ballings, M. *et al.* (2015) 'Evaluating Multiple Classifiers for Stock Price Direction Prediction'. *Expert Systems with Applications*, 42(20), pp. 7046–7056. DOI: 10.1016/j.eswa.2015.05.013.

Basak, S. *et al.* (2019) 'Predicting the Direction of Stock Market Prices Using Tree-Based Classifiers'. *North American Journal of Economics and Finance*, 47(June 2018), pp. 552–567. DOI: 10.1016/j.najef.2018.06.013.

Bousono-Calzon, C. *et al.* (2019) 'On the Economic Significance of Stock Market Prediction and the No Free Lunch Theorem'. *IEEE Access*, 7, pp. 75177–75188. DOI: 10.1109/ACCESS.2019.2921092.

Breiman, L. (1997) 'Arcing the Edge'. *Statistics*, 4, pp. 1–14.

Breiman, L. (2001) 'Random Forests'. *Random Forests*.

Breiman, L. (1996) 'Stacked Regressions'. *Machine Learning*, 24(1), pp. 49–64. DOI: 10.1007/bf00117832.

Chen, T. and Guestrin, C. (2016) 'XGBoost: A Scalable Tree Boosting System'. DOI: 10.1145/2939672.2939785.

Dey, S. *et al.* (2016) 'Forecasting to Classification : Predicting the Direction of Stock Market Price Using Xtreme Gradient Boosting Forecasting to Classification : Predicting the Direction of Stock Market Price Using Xtreme Gradient Boosting'. (October), pp. 1–10. DOI: 10.13140/RG.2.2.15294.48968.

Freund, Y. and Schapire, R.E. (1997) 'A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting'. *Journal of Computer and System Sciences*, 55(1), pp. 119–139. DOI: 10.1006/jcss.1997.1504.

Frosyniotis, D., Stafylopatis, A. and Likas, A. (2003) 'A Divide-and-Conquer Method for Multi-Net Classifiers'. *Pattern Analysis and Applications*, 6(1), pp. 32–40. DOI: 10.1007/s10044-002-0174-6.

Gyamerah, S.A., Ngare, P. and Ikpe, D. (2019) 'On Stock Market Movement Prediction Via Stacking Ensemble Learning Method'. *CIFEr 2019 - IEEE Conference on Computational Intelligence for Financial Engineering and Economics*, (June 2020). DOI: 10.1109/CIFEr.2019.8759062.

H2O.ai. *AutoML*. *Documentation*. Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html.

H2O.ai. *Stacked Ensembles*. Available at: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/stacked-ensembles.html (Accessed: 15 August 2020b).

Hastie, T., Tibshirani, R. and Friedman, J. (2001) *Greedy Function Approximation: Gradient Boosting Machine*. second. Available at: https://statweb.stanford.edu/~jhf/ftp/trebst.pdf.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) 'The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition by Trevor Hastie, Robert Tibshirani, Jerome Friedman'. *International Statistical Review*, 77(3), pp. 482–482. DOI: 10.1111/j.1751-5823.2009.00095_18.x.

Ho, T. kam., Hull, J.J. and Srihari, S.N. (1994) 'Decision Combination in Multiple Classifier Systems'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1), pp. 66–75. DOI: 10.1109/34.273716.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013) *An Introduction to Statistical Learning - with Applications in R | Gareth James | Springer*. Available at: https://www.springer.com/gp/book/9781461471370%0Ahttp://www.springer.com/us/book/9781461471370.

Jiang, M. *et al.* (2020) 'An Improved Stacking Framework for Stock Index Prediction by Leveraging Tree-Based Ensemble Models and Deep Learning Algorithms'. *Physica A: Statistical Mechanics and Its Applications*, 541(258), p. 122272. DOI: 10.1016/j.physa.2019.122272.

Kotu, V. and Deshpande, B. (2018) 'Data Science Concepts and Practice - Ensemble Modelling (002).Pdf'. In *Data Science Concepts and Practice*. p. 568.

Kumar, I. *et al.* (2018) 'A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction'. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, (Icicct), pp. 1003–1007. DOI: 10.1109/ICICCT.2018.8473214.

Van Der Laan, M.J., Polley, E.C. and Hubbard, A.E. (2007) 'Super Learner'. *Statistical Applications in Genetics and Molecular Biology*, 6(1). DOI: 10.2202/1544-6115.1309.

Mehak Usmani,Syed Hasan Adil,Kamran Raza, S.S.A.A. (2016) 'Stock Market Prediction Using Machine Learning Techniques'. *2016 3rd International Conference On Computer And Information Sciences (ICCOINS)*, pp. 322–327. DOI: 10.1109/ICAC49085.2019.9103381.

Muhammad zulfiqar Umer,Muhammad Awais, M.M. (2019) 'Stock Market Prediction Using Machine Learning(ML) Techniques'. *September 2019Advances in Distributed Computing and Artificial Intelligence Journal 8(4):97*, 8, pp. 97–116. DOI: DOI: 10.14201/ADCAIJ20198497116.

Nair, B.B., Dharini, N.M. and Mohandas, V.P. (2010) 'A Stock Market Trend Prediction System Using a Hybrid Decision Tree-Neuro-Fuzzy System'. *Proceedings - 2nd International Conference on Advances in Recent Technologies in Communication and Computing, ARTCom 2010*, pp. 381–385. DOI: 10.1109/ARTCom.2010.75.

Nti, I.K., Adekoya, A.F. and Weyori, B.A. (2020) *A Systematic Review of Fundamental and Technical Analysis of Stock Market Predictions*. Springer Netherlands DOI: 10.1007/s10462-019-09754-z.

Nti, I.K., Adekoya, A.F. and Weyori, B.A. (2019) 'Random Forest Based Feature Selection of Macroeconomic Variables for Stock Market Prediction'. *American Journal of Applied Sciences*, 16(7), pp. 200–212. DOI: 10.3844/ajassp.2019.200.212.

Paliyawan, P. (2015) 'Stock Market Direction Prediction Using Data Mining Classification'. *ARPN Journal of Engineering and Applied Sciences*, 10(3), pp. 1302–1310. DOI: 10.13140/RG.2.2.26523.46882.

Shearer, C. (2000).

Soni, D. *et al.* (2018) 'Optimised Prediction Model for Stock Market Trend Analysis'. *2018 11th International Conference on Contemporary Computing, IC3 2018*, pp. 2–4. DOI: 10.1109/IC3.2018.8530457.

Tsai, C.F. *et al.* (2011) 'Predicting Stock Returns by Classifier Ensembles'. *Applied Soft Computing Journal*, 11(2), pp. 2452–2459. DOI: 10.1016/j.asoc.2010.10.001.

Weng, B. *et al.* (2018) 'Macroeconomic Indicators Alone Can Predict the Monthly

Closing Price of Major U.S. Indices: Insights from Artificial Intelligence, Time-Series Analysis and Hybrid Models'. *Applied Soft Computing Journal*, 71, pp. 685–697. DOI: 10.1016/j.asoc.2018.07.024.

Winkler, S.M. *et al.* (2016) 'Heterogeneous Model Ensembles for Short-Term Prediction of Stock Market Trends'. *International Journal of Simulation and Process Modelling*, 11(6), pp. 504–513. DOI: 10.1504/IJSPM.2016.082914.

Wolpert, D.H. (1992) 'Original Contribution: Stacked Generalization'. *Neural Netw.*, 5(2), pp. 241–259. DOI: 10.1016/S0893-6080(05)80023-1.

Appendices

This document will guide you through the contents of the Artifacts and the necessary steps to implement the Python code for dissertation project titled "**STOCK MARKET ANALYSIS USING STACKED ENSEMBLE LEARNING METHOD**".

1) Contents of Artifacts-
   a) Datasets –
      - S&P 500.csv – Raw dataset of S&P 500 stock index
      - NASDAQ.csv – Raw dataset of NASDAQ stock index
      - NYSE.csv – Raw dataset of NYSE stock index
      - Pre-processed S&P 500.csv – The pre-processed dataset of S&P 500 index created for modelling and further operations.
      - Pre-processed NASDAQ.csv – The pre-processed dataset of NASDAQ index created for modelling and further operations.
      - Pre-processed NYSE.csv – The pre-processed dataset of NYSE index created for modelling and further operations.
   b) Results –
      - S&P 500 AutoML.csv – The H2O AutoML leader board for S&P 500 data.
      - NASDAQ AutoML.csv – The H2O AutoML leader board for NASDAQ data
      - NYSE AutoML.csv – The H2O AutoML leader board for NYSE data.
   c) Code –
      - Final_S&P 500.ipynb – Python code to run all the operations for S&P 500
      - Final_NASDAQ.ipynb – Python code to run all the operations for NASDAQ
      - Final_NYSE.ipynb – Python code to run all the operations for NYSE

2) During the implementation of this research, while using the H2O technology, there were queries on the implementation of models and the requirements of computing power. For this reason, the researcher communicated with the developers of H2O.ai, including the Chief Machine Learning Scientist.