CIND 820 – Big Data Analytics Project

Krishnaveni Anantharaman

(# 501004847)

Supervisor: Tamer Abdou, PhD

Ryerson University

CIND820: Big Data Analytics Project
Final Report
Chang School of Continuing Education, Ryerson University

Title: How investors can use Machine learning techniques to analyse the stock market's trend?

Data Set: New York Stock Exchange data from Kaggle (https://www.kaggle.com/dgawlik/nyse)
Github link: https://github.com/krishrika/CIND820-Stock-Trend-Prediction

Krishnaveni Anantharaman
#501004847
Supervisor: Tamer Abdou, PhD (Email: tamer.abdou@ryerson.ca)

Table of Contents

Introduction

Stock market forecasting is a very important area in the financial sector. There has been a significant interest shown in this subject by academic researchers, investors and practitioners to aid in decision making. According to Bagheri et al., (2014), the ability to correctly predict future market trends is a prerequisite for successful financial market trading.

In the past stock selection relied heavily on the investors (or advisors) personal knowledge on the industry/company. Prediction methodologies were broadly classified into two categories. They are fundamental analysis and technical analysis. Fundamental analysis attempts to estimate a company's intrinsic value by analysing the company's financial statements. On the other hand, technical analysis analyses statistical trends gathered from past price movement and volumes. In the recent past, there has been a significant interest in using AI techniques to predict stock markets by examining patterns in massive amount of real time equity and economic data.

The Research question for this project is "How investors can use Machine learning techniques to analyse the stock market's trend?"

Literature Review

Many researches have been conducted in the past to predict stock performance in order to generate profitable trading opportunities. There are numerous literatures on this subject that have been published and these were consulted to ensure the project has a good foundation by understanding the strengths and weakness of these researches. The analysis of research papers referenced are outlined below. The reference section at the end of this document contains full citations

[1] *In the research "Predicting Stock Prices Using Technical Analysis and Machine Learning" submitted on June-2010 submitted by Jan Ivar Larsen(Norwegian University of Science and Technology),* the author attempts to build a model that can predict stock prices by applying ML algorithms on Technical indicators. The author has implemented two primary modules when building the model: a) Feature generation where technical indicators were constructed using domain knowledge, b) Feature aggregation where the output from (a) was fed into a machine learning classifier that can be used in stock selection. In the second module(ML classification) the author has developed an evolutionary algorithm called Agent Decision Tree Learning(ADTL). The author evaluates the feature generation module and the aggregation module and compares the results with the benchmark index by running simulations in a period of market uptrend and recession. Based on the results, the model has outperformed generally outperformed the benchmark index. The ideas of constructing technical indicators and then feeding into ML algorithms was referenced form this research

[2] *In the research "Predicting the direction of stock market prices using Random Forest"(* **(**arXiv:1605.00003 *[cs.LG])) submitted on May-2016, Luckyson Khaidem,Snehanshu Saha and Sudeepa Roy Dey*, the author researches ways to predict stock market returns using

random forest technique. Data was exponentially smoothed (applying more weightage to recent data) and then technical indicators (RSI,MACD etc) were computed.The entire data is then split into training and test data and before feeding to the model linear separability is tested. The author then proceeds to build a classifier based on random forest algorithm and evaluates the model using accuracy,precision,recall,specificity and ROC.The approach of constructing technical indicators and then feeding into the random forest model was adapted in our model with some structural changes(eg : feature selection). The evaluation criteria(precision,recall,specificity,accuracy) were also implemented in our model validation.

[3] *In the research "Equity forecast: Predicting long term stock price movement using machine learning"(arXiv:1603.00751 [cs.LG]) Nikola Milosevic predicts long term market movement based on company fundamentals*. The author applies feature selection techniques manually and comes up with the optimal set of features. The data with the optimal features is then fed into different classifiers and the resuls were evaluated using precision,recall and F-Score. While the classifier we are building is based on technical analysis (ie different type of data), the idea of evaluating different combination of features selection was reused in our model. Instead of manually selecting the features (as done in this paper), we used ML tools to do this. This way the feature selection is less error prone and we were able to increase the scope by implementing three different flavors of feature selection.

[4] *In the research, "PREDICTING AND BEATING THE STOCK MARKET WITH MACHINE LEARNING AND TECHNICAL ANALYSIS" Anthony Macchiarulo*. The author conducts a short comparison between machine learning and technical analysis to predict stocks market is provided. The machine learning algorithms used by the authors are Support vector Machines, Neural Network and Ensemble Learning. The authors have concluded that using machine learning as a trading strategy can positively impact the returns generated compared to using many technical indicators. It was concluded that in up markets ML outperforms technical analysis whereas in down markets technical analysis outperforms ML. The author uses cross validation to ensure the model does not do overfitting and we adopted this is idea in the model we built. One of the drawbacks in this research is that the down market had very few observations (<50) so the results may be biased. The author could have used smoothing techniques to balance the data before applying the ML classifiers.

[5] *In the paper "Machine Learning in Stock Price Trend Forecasting" by Dai and Zhang (2013),* the author uses logistic regression, quadratic discriminant analysis and SVM to train a prediction system on 3M Stock data. The aim of the research was to predict short term(next day) trend and long term(next n day) trend. While the type of research (prediction system) was different from our research(classification), we used the concept of trend in our research. The trend (class attribute in our research) is based on 1_day_return whereas in the review research it was based on closing price. The 1_day_return is based on closing price so conceptually it is very similar. Moreover, similar to this research the trend attribute in our research will use one of the two values (-1 or +1)

[6] *In the paper "Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market" L. Chen, Z. Qiao, M. Wang, C. Wang, R. Du and H. E. Stanley, "Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?" in IEEE Access, vol. 6, pp. 48625-48633, 2018, doi: 10.1109/ACCESS.2018.2859809* , the authors have used DL techniques

and compared the performance with traditional back propagation network(BP), Extreme learning machine(ELM) and Radical basis Function(RBF). The authors have run multiple simulations and have averaged out the results. While algorithm used in this paper is deep learning is different from ours (supervised learning), the idea of running multiple simulations and averaging the performance measures was utilized in our model. By doing this our model ensures there are no biases in the data.

[7] *In the paper "Machine Learning Techniques for Stock Prediction", Vatsal H.* Shah the author has experimented various algorithms like SVM,Linear regression,decision stumps for stock price prediction. While the author's approach to run multiple classifiers and evaluate them is very promising, he could have run each model multiple times and averaged since in his method there could be biases in the training data. This approach of running each multiple time was used in our research after identifying this draw back in the author's model.

Dataset

The source for this dataset is Kaggle New York Stock Exchange data (https://www.kaggle.com/dgawlik/nyse). This dataset is primarily intended for Fundamental and Technical analysis.

There are four files in the dataset: prices.csv, price-split-adjusted.csv, securities.csv and fundamentals.csv.

Prices.csv: raw, as-is daily prices. Most of data spans from 2010 to the end 2016, for companies new on stock market date range is shorter. There have been approx. 140 stock splits in that time, this set doesn't account for that. This file contains 7 columns and around 851k rows. Each stock has approximately 1800 rows of data, which will be filtered for our analysis. The attributes in this file are Date, symbol, open, close, high, low, volume. There are price data for 501 unique symbols.

S.no	Attribute	Description
1	Date	Stock trading date Data Type: Datetime
2.	Symbol	Ticker identifier for the stock of the company Date Type: string
3.	Open	NYSE opening trade price Date Type: float
4.	Close	NYSE closing trade price Date Type: float

S.no	Attribute	Description
5.	High	NYSE day highest trade price Date Type: float
6.	Low	NYSE day lowest trade price Date Type: float
7.	Volume	Total No. of the shares traded for the day Date Type: Integer

Table 1: Price data attribute and data type

Securities.csv: general description of each company with division on sectors. This file contains 8 columns and around 505 rows. The attributes in this file are Ticker symbol, security, SEC filings, GICS sector, GICS Sub Industry, Address of Headquarters, Date first added, CIK. There are securities data for 504 unique securities. (Note: Columns SEC filling, Address of Headquarter, Date first added and CIK are removed as part of data cleaning)

S.no	Attribute	Description
1.	Ticker symbol	Ticker identifier for the stock of the company Date Type: String
2.	Security	Company name Data Type: String
3.	SEC filings	Financial statement filing for the company Data Type: String
4.	GICS Sector	Sector of the company based on MSCI & S&P classification Date Type: String
5.	GICS Sub Industry	Sub-Industry of the company's sector Data Type: String
6.	Address of Headquarters	Address of the company Data Type: String
7.	Date first added	First filing date in SEC Data Type: Date

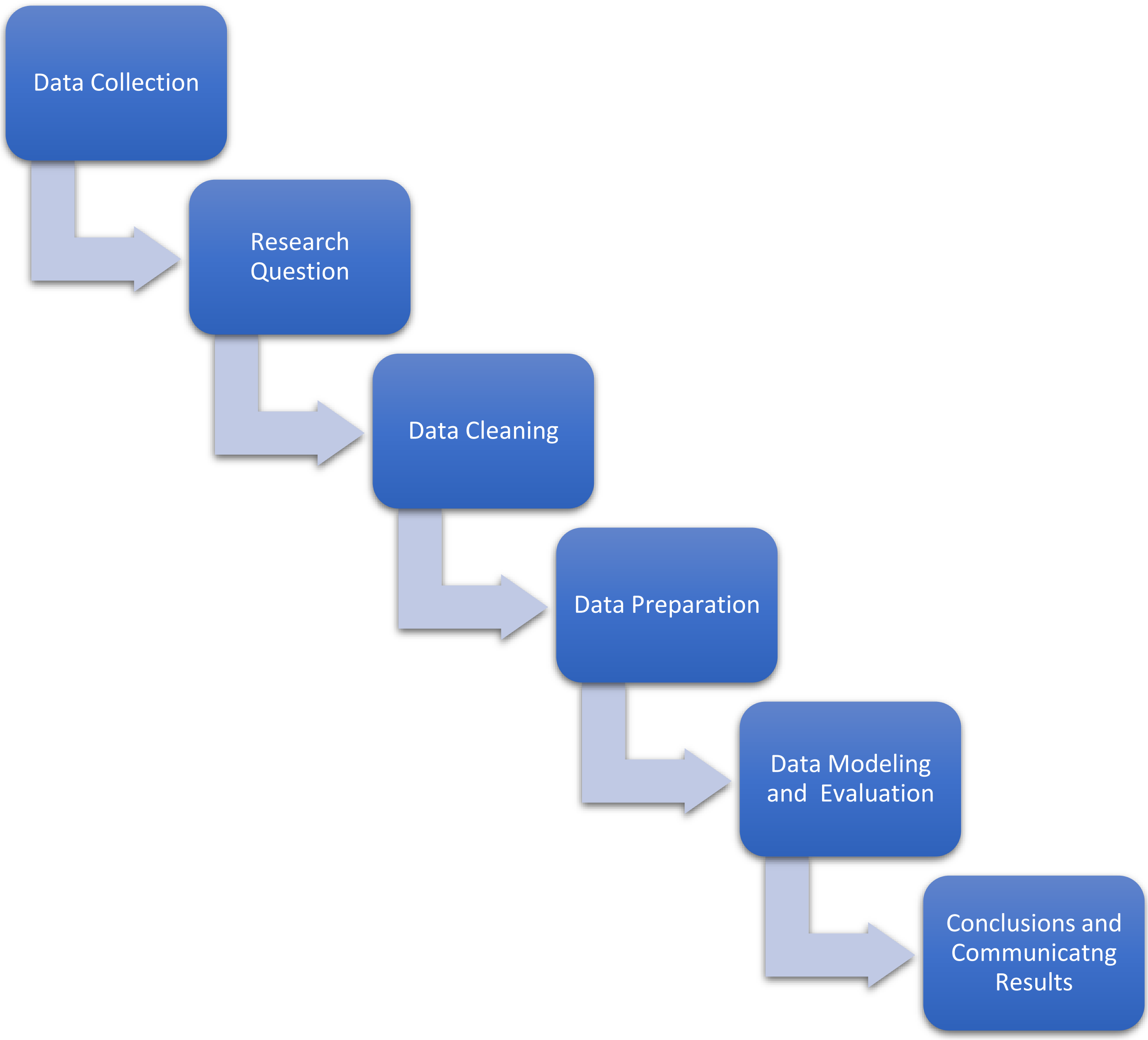
S.no	Attribute	Description
8.	CIK	Unique key to identify a corporation in SEC data base Data Type: Integer

Table 2: Security data attribute and data type

Note: For the purpose of the project we will not use price-split-adjusted.csv but use the non adjusted prices in the prices.csv. We will also not use fundamentals.csv as we are not incorporating Fundamental analysis in our model and limit our scope to technical analysis.

Approach

The files contain historical data for about 500 securities. For our illustrative purpose we will use a single stock (MSFT) but the program will be customisable so users can run the data preprocessing and model on any stock by passing a parameter. The following diagram outlines the approach we will use in our project


Data Collection
Research Question
Data Cleaning
Data Preparation
Data Modeling and Evaluation
Conclusions and Communicatng Results

Step 1: Data Collection

Participants in the stock market would need to get a sense of the direction in which a stock will move before trading on it. The New York Stock Exchange data from Kaggle (https://www.kaggle.com/dgawlik/nyse) has been downloaded in order to build a model for stock market prediction

Step 2. Research Question

The motivation for this project is to help users make better decisions when trading stocks by building a machine learning model for predicting the stock market and Classifying top performing industries. So, the research question for this project is How investors can use Machine learning techniques to analyse the stock market's trend?

Step 3. Data Cleaning

- There were no missing values in the Prices.csv file.
- After new features were added, we removed all rows which has nan values for feature extraction.
- From the Securities.csv we removed SEC filling, Address of head quarters, Date first added, CIK (Data in these columns will not have any impact on the model)
- The data type for Date column is loaded into the Pandas data frame as objects by default. For this data to be compatible, we convert the data type to Datetime. Please refer Table 1 & 2 for updated data types.
- In order to identify outliers for 1-day return, standard deviation method was used. Mean was calculated iteratively for 21 consecutive returns (rolling window) and any data in this set which was 3 standard deviation away from the mean was classified as an outlier. Outliers were removed for the training set only and the test set was not modified.

Step 4. Data Preparation

- The following features are extracted by performing simple transformations on the price and volume attributes. These features are some of the important ones used by industry practitioners for technical analysis. For better understanding, I have placed them into subcategories below.
- After the features were extracted, all rows (about 60 rows) which had NAN values were removed to improve model efficiency. The NAN values were expected for the first few time periods as some of the columns calculated moving average which will be missing for the first few time periods. For eg: the 60d volatility will be missing for the first 60 rows.
- The trend column which is calculated from 1-day return is the class attribute for the model. The column can hold either 1(if 1day return > 1) or -1(if 1-day return <1)
- Once the features are extracted, Correlation matrix between all the features (including the below) were determined using Pearson correlation method.
- The above data was split into training and test data set (70-30 split). Three types of feature selection techniques were then applied to the training set in order to improve the performance of the model. The three techniques-Filter method (Mutual Information classification), wrapper method (forward selection) and embedded method (tree based) where then applied separately to the training set.
- The price data was a balanced dataset which implies the data was equally distributed between two classes- up and down trend

1. Return Features

- **1-day return**: Ratio of today's close to yesterday's close for a stock
- **close_to_open:** Ratio of close to open for a day
- **close_to_high:** Ratio of close to high for a day
- **close_to_low:** Ratio of close to low for a day

2. Trend Indicator Features

- **Ma_50day:** Moving averages are used to smooth price data by calculating an average over a specific period. For our purpose we calculate 50 days simple moving average for each stock on a day
- **MACD_diff:** Moving average convergence difference is an indicator that shows the relationship between two moving averages of a security's price. It is calculated as the difference between 26 period exponentials moving average (EMA) and 12 period EMA.

3. Momentum Indicator Features

- **Stochastic_Oscillator:** Stochastic Oscillator is a momentum indicator that shows the location of the close relative to the high-low range over a set number of periods.
- **CCI:** The commodity channel indicator(cci) is a versatile indicator that can be used to identify a new trend or warn of extreme conditions. In general, CCI measures the current price level relative to an average price level over a given period of time.
- **RSI:** Relative Strength Index (RSI) is an extremely popular momentum oscillator that measures the speed and change of price movements. RSI oscillates between zero and 100.

4. Volatility Indicator Features

- **5d_volatility:** Calculated as standard deviation for 5-day(rolling) returns.
- **21d_volatility:** Calculated as standard deviation for 21-day(rolling) returns.
- **60d_volatility:** Calculated as standard deviation for 60-day(rolling) returns.
- **Bollinger_bands:** Bollinger Bands® are volatility bands placed above, middle and below a moving average. Volatility is based on the standard deviation which changes as volatility increases and decreases. The bands automatically widen when volatility increases and contract when volatility decreases. Their dynamic nature allows them to be used on different securities with the standard settings.

5. **Volume Features**

- **On_Balance_Volume:** On Balance Volume (OBV) measures buying and selling pressure as a cumulative indicator, adding volume on up days and subtracting it on down days.

Step 5. Data Modelling and Evaluation

The model will conceptually have two modules:

a) Technical analysis module: In this module technical indicators described above will be computed. The output from this module will feed the ML module (see below)
b) Machine learning module: This module will apply machine learning techniques for the data generated from the above module.
    - We will use simple sampling technique for the data in order to avoid biases introduced by randomised sampling.
    - 3 types of algorithms will be used on the resampled dataset to predict the stock market trend (Up or down). The algorithms are Random Forest, Decision Tree and Support Vector Machine.
    - Models were evaluated against the benchmark model (Random Forest) in order to identify which classifier gives better results in terms of classifying up and down trends in return.
    - The model will be evaluated using time series split validation method
    - To evaluate the performance of the model, Evaluation matrix like Accuracy, Precision, Recall, F1-score, ROC AUC score and confusion Matrix are used

Step 6. Conclusions and Future Work

Conclusion and Future recommendations will be provided based on the outcome of the research

Data Exploration

There are two datasets used in this analysis: prices and security. The security dataset contains indicative data like ticker, sector, sub sector, address etc. There are about 505 rows and 8 columns in the dataset of which we will use only 4 columns (Ticker symbol, Security, GICS Sector, GICS Sub Industry) and all of these fields contains string datatypes. The list of columns and their datatypes are in Table1. The price dataset holds raw daily prices from 2010 to end of 2016 of 540 stocks traded in NYSE. There are about 851k rows and 7 columns. The list of columns and datatypes for the security dataset is in Table 2. The price and security data set were merged, and analysis was done on return performance for each sector. Based on the output of the below figure

1, the top performing industry is Industrials followed by consumer discretionary and Information Technology.

S.no	GICS Sector	Industry Performance
1.	Industrials	5836
2.	Consumer Discretionary	5795
3.	Information Technology	5672
4.	Financials	5488
5.	Health Care	4850
6.	Real Estate	3608
7.	Consumer Staples	3498
8.	Utilities	3126
9.	Materials	1726
10.	Energy	746
11.	Telecommunications Services	420

Table 3: GICS Sector Performance

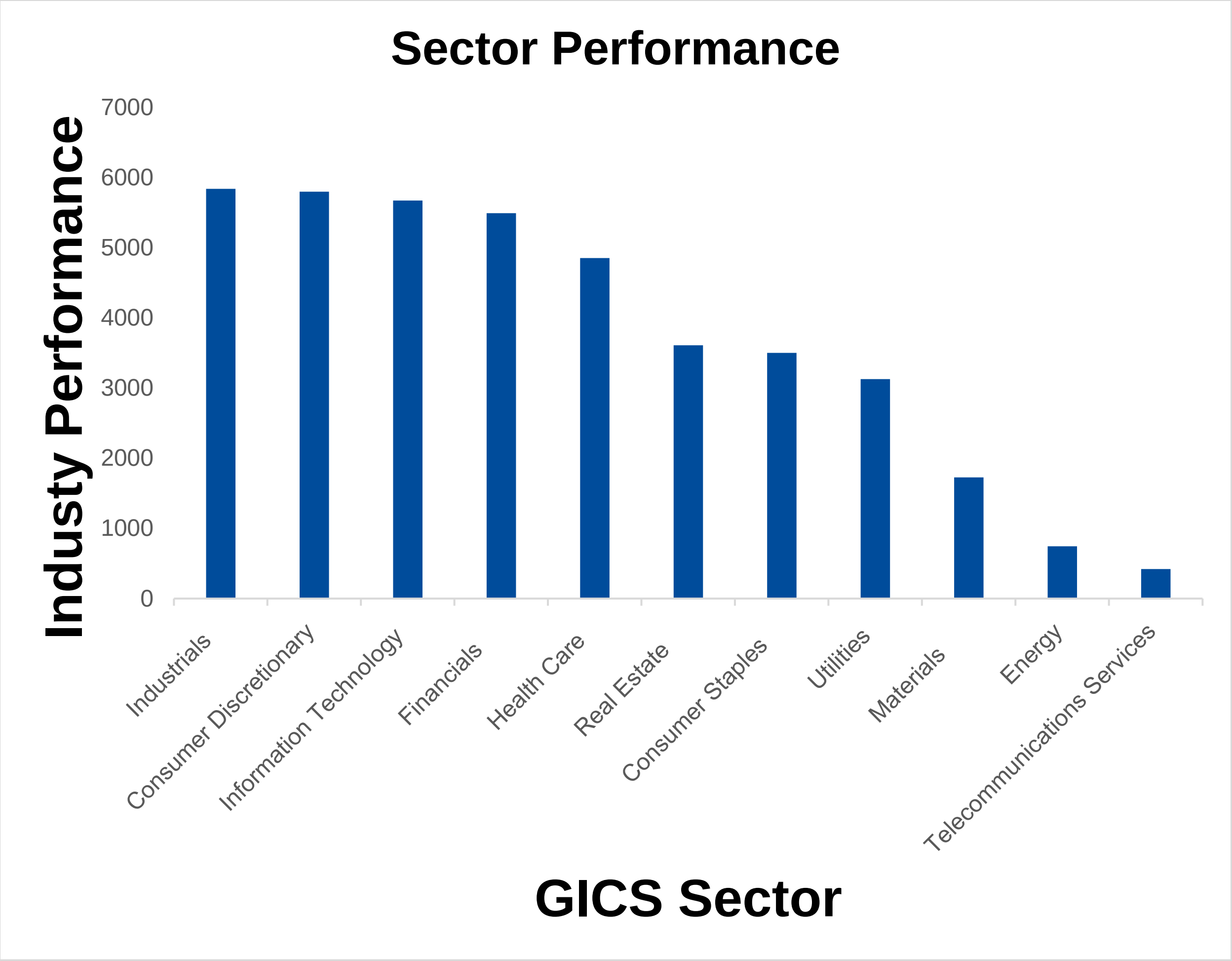


*Figure 1: **Industry performance***

To build and evaluate our model we will use the data for Microsoft (MSFT) stock, but the model can be easily used for any security in the dataset. For MSFT, there are 1762 rows

Feature Extraction

Technical analysis is a trading tool which attempts to employ various statistic methods on past market data to predict future price movements of a stock. The technical indicators are key inputs to our model and a very concise set of indicators are required to build an efficient model. The following technical indicators are additional features extracted from the price and volume fields in the price dataset and will be used in our model

6. 1-Day Return

1-Day return is the ratio of today's closing price to yesterday's closing price for a stock. This feature is a very important as it is used to derive our class attribute(trend) and also serves as input for calculating other technical indicators. This feature is also used for our outlier analysis.

7. Close_to_open

close_to_open calculates the ratio of closing price to opening price. The feature indicates how the stock has changed from the start of day to the end of day.

8. Close_to_low

close_to_low calculates the ratio of closing price to the lowest stock price on a particular day. This feature measures how the stock has progressed from the lowest point in the day to the end of day

9. Close_to_high

close_to_high calculates the ratio of closing price to the lowest stock price on a particular day. This feature measures how the stock has progressed from the highest point in the day to the end of day

10. Ma_50d

MA_50d (50 day moving average) is a trend indicator that calculates the simple average of closing prices for a stock for the past 50 days. While the calculation is simple, this is a very popular and an important indicator as it acts as a first line of support in an uptrend or first line of resistance in a downtrend.

11. MACD_Diff

The Moving average Convergence Divergence is a trend indicator that analyses the relationship between two different moving averages of a security's price. For the calculation we take the difference between 26 period and 12 period exponentials moving average (EMA) which is defined as below

EMA= (Today's closing price$*$2/1+period))+EMAYesterday $*$(1−(2/1+period))

The MACD is calculated using below formula

MACD = EMA (12 period) – EMA (26 period)

By calculating the difference between two trends (EMA), MACD attempts to offer the best of both the trend and momentum indicators A positive MACD implies the 12 period EMA is greater than the 26 period EMA and hence the upside momentum is increasing. Conversely a negative MACD implies the 26 period EMA is greater than 12 period EMA and hence the downside momentum is increasing

12. Stochastic Oscillator

The stochastic oscillator is a momentum indicator which aims to compare the closing price of a security to a range of past price. It takes a value between 0 and 100 and is very popular to generate overbought and oversold signals.

13. Commodity Channel Index (CCI)

The CCI is a momentum indicator which aims to compare the price over historic averages. CCI greater than zero indicates the price is above historic average and higher values signifies strength. Conversely a CCI less than zero indicates the price is below historic average and lower values indicates weakness.

14. Relative Strength Indicator (RSI)

RSI is a momentum indicator that measures the magnitude of recent price changes and identifies overbought or oversold conditions. Traditionally RSI greater than 70% is perceived to be an overbought (or overvalued) scenario and RSI less than 30% is perceived to be an oversold scenario (or undervalued) scenario.

15. 5_day_volatility

The 5-day volatility is calculated as standard deviation for 5-day(rolling) returns

16. 21_day_volatility

The 21-day volatility is calculated as standard deviation for 21-day(rolling) returns

17. 60_day_volatility

The 60 -day volatility is calculated as standard deviation for 21-day(rolling) returns

18. Bollinger Bands

Bollinger Band is a volatility indicator that composes of three lines- Simple Moving average which indicates middle band and an upper and lower band. The upper and lower bands are usually 2 standard deviations from the SMA. When the price of the security moves closer to the upper band the security is considered to be overbought and conversely when it moves closer to the lower band the security is considered to be oversold.

19. On Balance Volume

On balance volume is a volume indicator that uses flow in trading volume and predicts the stock's price.

Distribution of Class Attribute

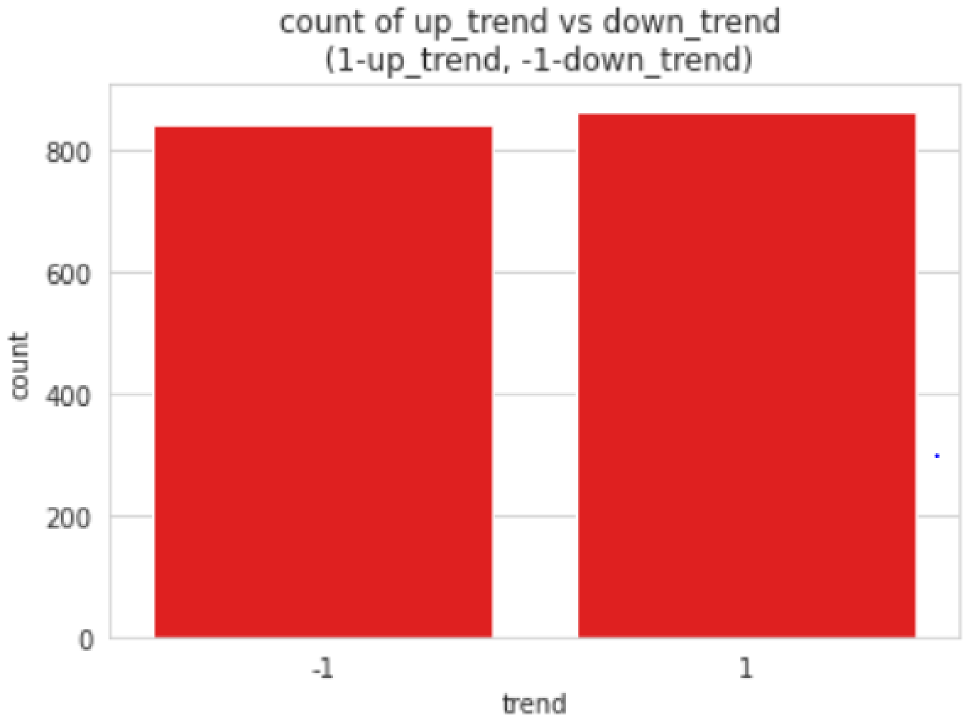


Figure 2: ***Distribution of class attribute***

The above figure show that there is equal distribution for the trend attribute (class attribute). The total no of up trend (+1) is 863 and down trend is 869(-1) out of 1732 and this shows that the data is balanced.

Correlation Matrix

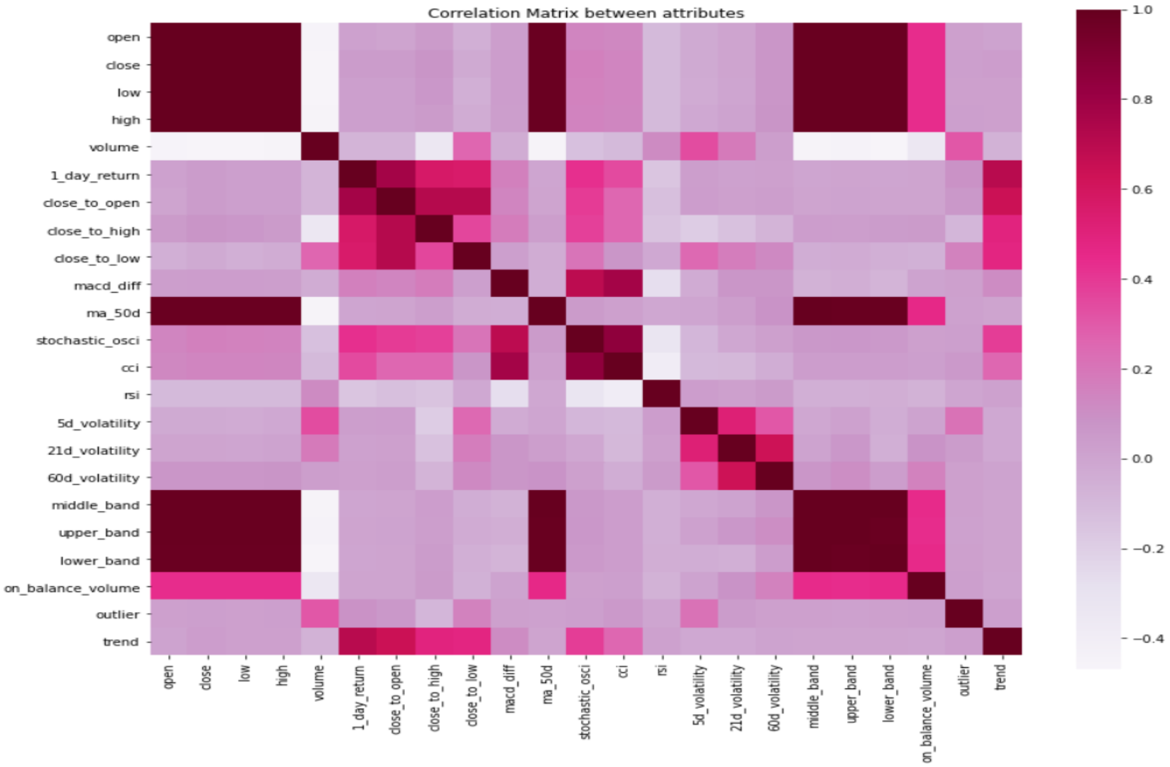


Figure 3: ***Heat map Correlation Matrix on the Price data set***

The above figure shows that the data is normally distributed (1_day_return/trend - dependent value). We used Pearson correlation for our analysis. Most of the attributes are calculated from the '1_day_return' column and therefore show high correlation between them. In order to determine an attribute strength, feature selection techniques were employed

Visualisation of attribute in Price data set

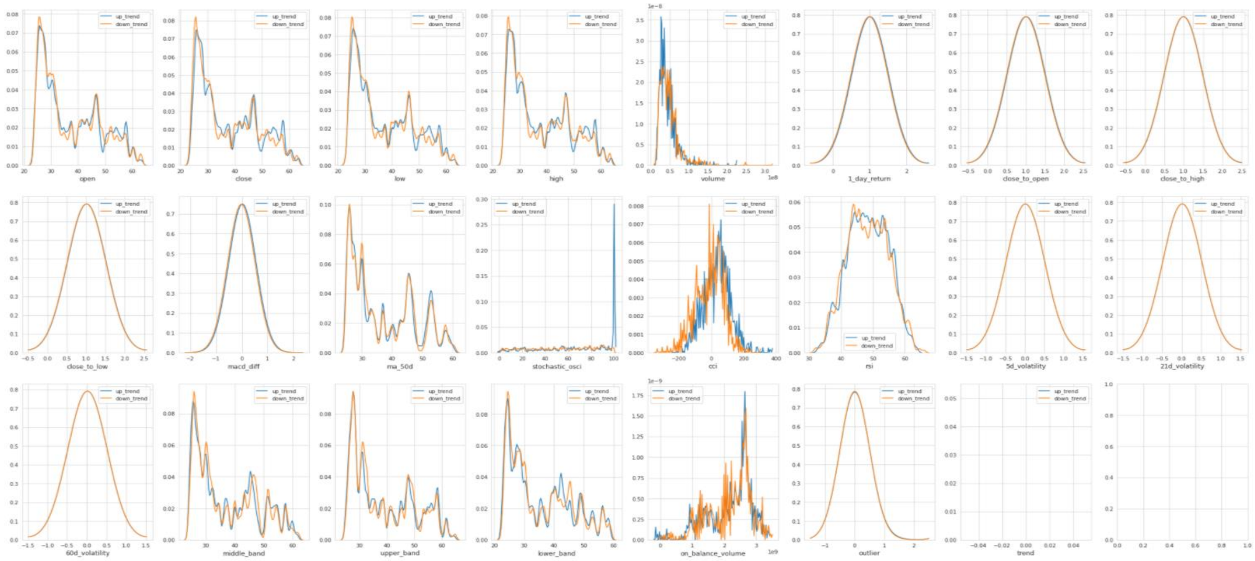


Figure 4: ***Kernel Density plot for all the features in Price data set***

The above Kernel Density plot shows normal distribution for Target attributes (trend derived from 1_day_return). Technical indicator columns are derived from 1_day_return. Outliers are identified and removed from training data set.

Data Preparation

Outlier

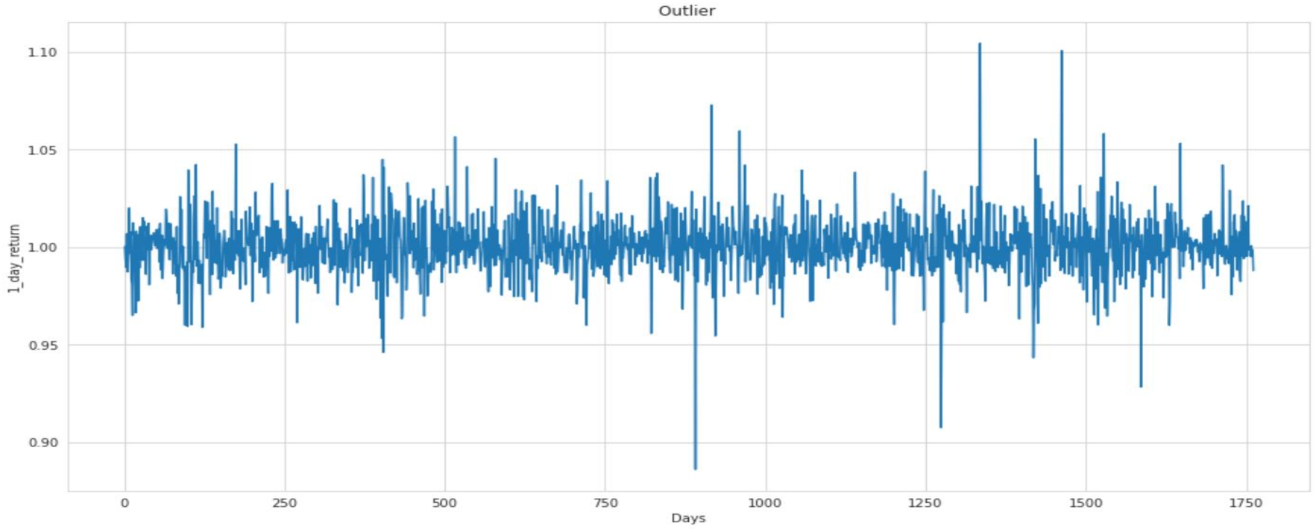


*Figure 5: **outlier for target variable(1_day_return)***

For outlier detection, a new column was added to flag a row as outlier or non outlier. Outliers were detected for the 1-day return attribute. Mean and standard deviation were calculated and compared. Mean was calculated on 21 days rolling window and if the value is 3 standard deviation away from the mean, that data point is identified as an outlier. The training data was then modified by eliminating the rows flagged as an outlier. The test data was preserved, and outliers were not removed from them.

Train-Test split

The price dataset was split into train and test using a 70-30 split. The train and test data were reused subsequently in all the classifiers. In order to ensure there are no overfitting bias, this split was performed 10 times and all the classifiers were run for each of the train data and tested again. The average of all the result (accuracy, precision etc.,) from each of these runs were then calculated to determine the model performance.

Feature Selection

Feature selection techniques were employed to reduce the dimensionality of our data and to improve the execution time for training the model. Three main feature selection techniques were applied, and accuracy of the model were analysed with each of the techniques. The following feature selection techniques were employed in our model

- Filter method – Mutual information
- Wrapper Method - Forward selection
- Embedded method – Tree based Algorithm

1. Filter Method – Mutual Information

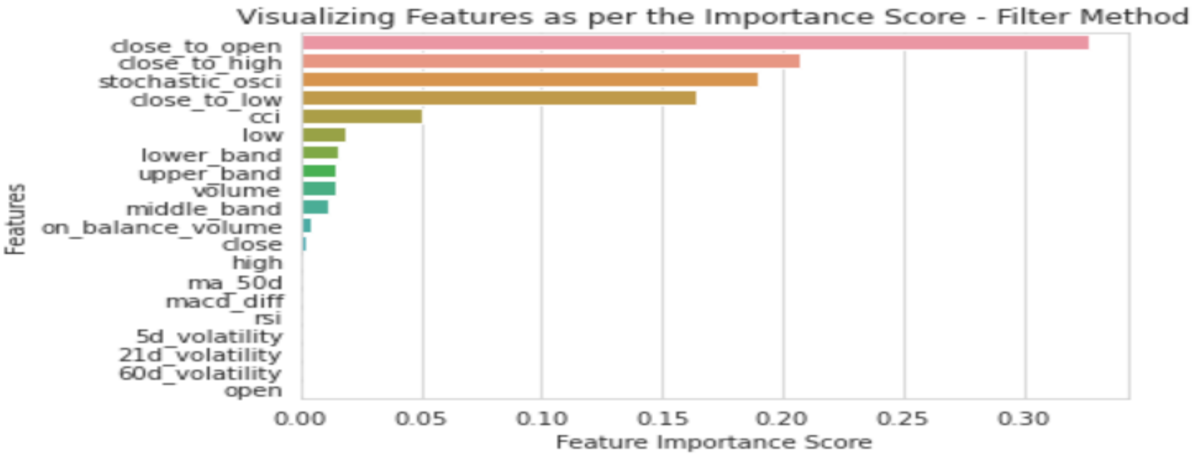


*Figure 6: **Visualizing Features as per Filter method***

The Mutual information (MI) is an entropy measure that assess which features should be included in the reduced dataset. This filter method is used because this is the recommended method when our target variable is a categorical variable. We compute the MI between each of our feature and the target variable(trend) for the training data set. Once this is calculated, we take the top 10 features based on the MI for our model. The below are the entropy score for each of our feature against the trend(target) column when we run this classifier. The results of this feature selection algorithm on our dataset is presented in the above figure.

2. Wrapper Method – Forward Selection

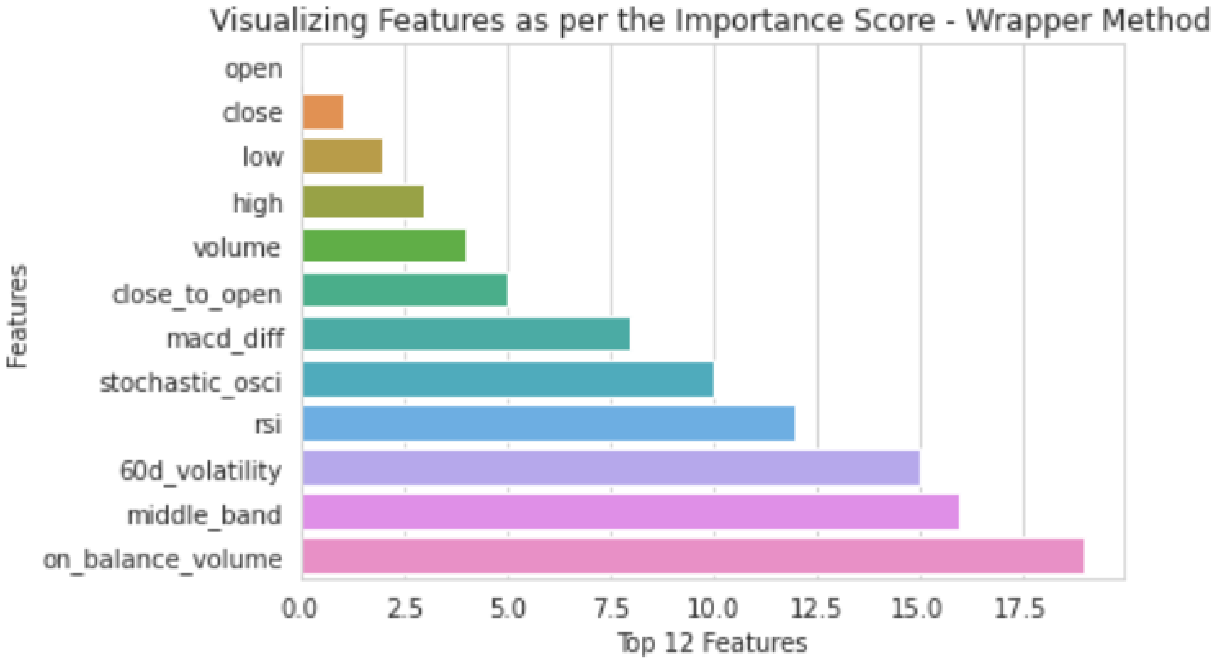


Figure 7: ***Visualizing Features as per Wrapper method***

The forward selection method is a greedy search algorithm that runs recursively to find the best set of features that improves the model. The search stops when the addition of a new variable does not improve the model. In this analysis Random Forest classifier were used with a desired number of features set to 12. The results of this feature selection algorithm on our dataset is presented in the above figure.

3. Embedded Method – Tree based Algorithm

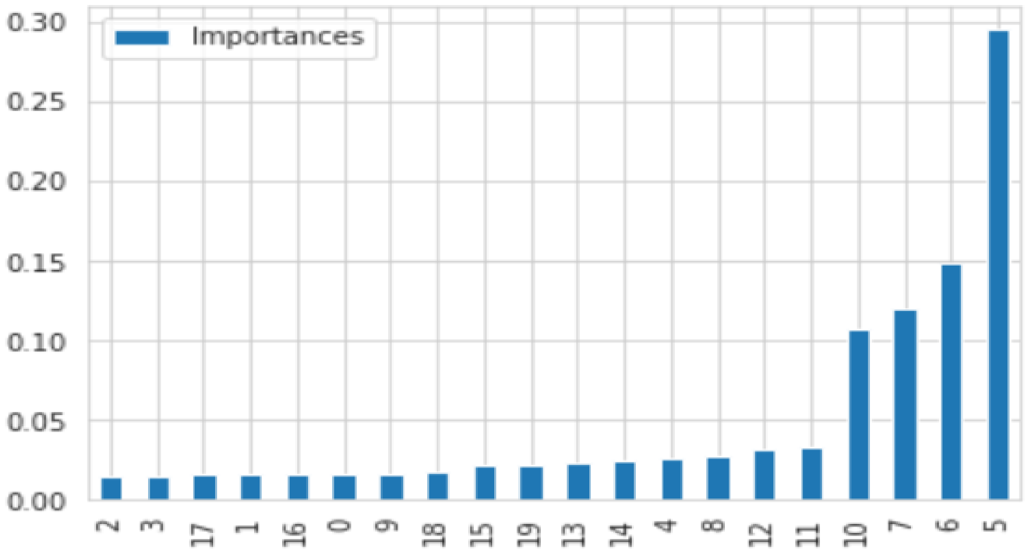


Figure 8: ***Visualizing Features as per Embedded method***

The feature selection algorithm is embedded in the random forest classifier and we attempt to use this in our model to sort the features based on importance. We then feed the top 10 features into our model at a later stage. The above figure represents the features and their importance from the Tree based method.

Data Modelling

Classifier on the Filter Method, Wrapper Method, Embedded Method and without Feature Selection

In the data modelling step, the features selected above from the three feature selection techniques: Filter, Wrapper and Embedded along with all the features (ie without feature selection) was fed into three different classifier models. The classifier models used were Random forest, Decision Tree and Single Vector Machine (SVM). This process was repeated 10 times with different training and testing data to ensure there are no biases. As a result, each model was run 40 times: 10 training sets * 3 feature selection techniques + 10 training sets without feature selection. The results were tabulated, and average scores were measured.

Random Forest Classifier

Random Forest classifier is a supervised machine learning algorithm that consist of large number of small decision trees which each produce their own classification. It then aggregates the classification from individual decision trees to arrive at the final classification in the model. The RF classifier was used in this model by running with 3 features and without feature selection as described above in the approach section. For each of the run the number of trees in the forest were set to 100 and to measure the quality of the split “Gini Impurity” criterion was used. The results of all the 40 runs were fed into model cross validator.

Decision Tree Classifier

Decision tree classifier is a supervised machine learning algorithm that constructs a tree like structure for classification. Each internal node represents a feature, a branch represents a decision rule and the leaf node represents outcome. The decision tree classifier was used in this model by running with 3 features and without feature selection as described above in the approach section. For each of the run maximum depth was set to 8 and minimum sample leaf was set to 5. To measure the quality of split “Entropy” was used as a criterion. The results of all the 40 runs were fed into model cross validator.

Support Vector Machine

Support Vector Machine is a supervised learning algorithm that plots each data as a point in n-dimensional space and then determines the hyperplane that differentiates the two class very well. The SVM classifier was used in this model by running with 3 features and without feature selection as described above in the approach section. Prior to running the SVM classifiers the data was scaled to improve the computational speed. For this research “linear kernel” was used for the training dataset.

The average results of the10 runs from three models are presented in the below table

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC Score	Cross Validation – Time Series Split (5 split)	Cross Validation – Time Series Split (10 split)
Random Forest	86.89	88.20	85.20	86.60	93.13	85.10	85.14
Decision Tree	84.93	86.10	83.50	84.70	90.91	79.59	79.99
Support Vector Machine	85.10	85.60	84.60	85.00	92.04	82.14	82.71

Table 4: Filter Method – Mutual Information Feature selection

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC Score	Cross Validation – Time Series Split (5 split)	Cross Validation – Time Series Split (10 split)
Random Forest	87.14	88.30	85.40	86.80	93.39	84.89	85.79
Decision Tree	84.95	86.00	83.40	84.50	90.14	79.08	79.53
Support Vector Machine	85.77	85.80	85.70	85.90	92.82	84.18	83.17

Table 5: Wrapper Method – Forward feature selection

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC Score	Cross Validation – Time Series Split (5 split)	Cross Validation – Time Series Split (10 split)
Random Forest	87.12	88.40	85.80	86.80	93.50	85.40	85.98
Decision Tree	85.60	86.90	84.00	85.50	91.24	79.59	78.13
Support Vector Machine	85.96	86.10	86.00	86.00	92.94	84.48	83.83

Table 6: Embedded Method – Tree based algorithm feature selection

Classifier	Accuracy	Precision	Recall	F1-Score	ROC AUC Score	Cross Validation – Time Series Split (5 split)	Cross Validation – Time Series Split (10 split)
Random Forest	85.94	88.20	83.50	85.80	90.78	84.89	84.67
Decision Tree	85.26	86.40	83.70	85.10	90.97	77.75	77.57
Support Vector Machine	86.00	85.90	86.30	86.00	93.28	83.46	83.55

Table 7: Without feature selection (with all attribute)

Data Model Evaluation and Result

Filter Method Feature Selection – Mutual Information

From the above results it was very evident that the Random Forest Classifier (benchmark Model) and SVM outperformed the Decision Tree classifier when classifying stock market trends. Random Forest shows 86.89 accuracy and SVM show 85.10. A good criterion to evaluate a better model is a high score of accuracy and F1 score. But when comparing Decision tree against our Random Forest classifier (benchmark model) results in terms of accuracy, precision, recall and F1 score were lesser than benchmark model.

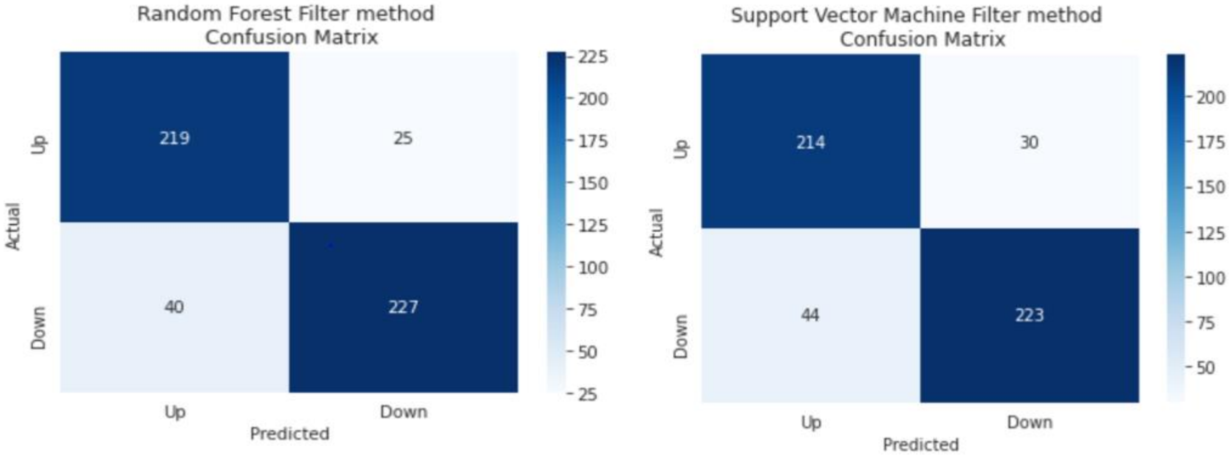


Figure 9: ***Random Forest Confustion Matrix vs SVM confusion Matrix***

From the above confusion matrix (Figure 9), Random Forest has captured 219 of the 244 up trends and 227 of the 267 down trends whereas SVM has captured 214 of the 244 up trends and 223 of the 267 down trends. Both models have predicted high true positive and low false positive rate which is very promising. Based on these results, it was clear that Random Forest and SVM classifiers are the best performing algorithms in Filter feature selection method (Mutual information). Therefore, these algorithms can help an investor to make the right trading decision, i.e. buy or sell a stock at the selected time.

Wrapper Feature Selection - Forward Feature Selection

From the above results it was very evident that the Random Forest Classifier (benchmark Model) and SVM outperformed the Decision Tree classifier when classifying stock market trends. Random Forest shows 87.14 accuracy and SVM show 85.77. A good criterion to evaluate a better model is a high score of accuracy and F1 score. But when comparing Decision tree against our Random Forest classifier (benchmark model), results in terms of accuracy, precision, recall and F1 score were lesser than benchmark model.

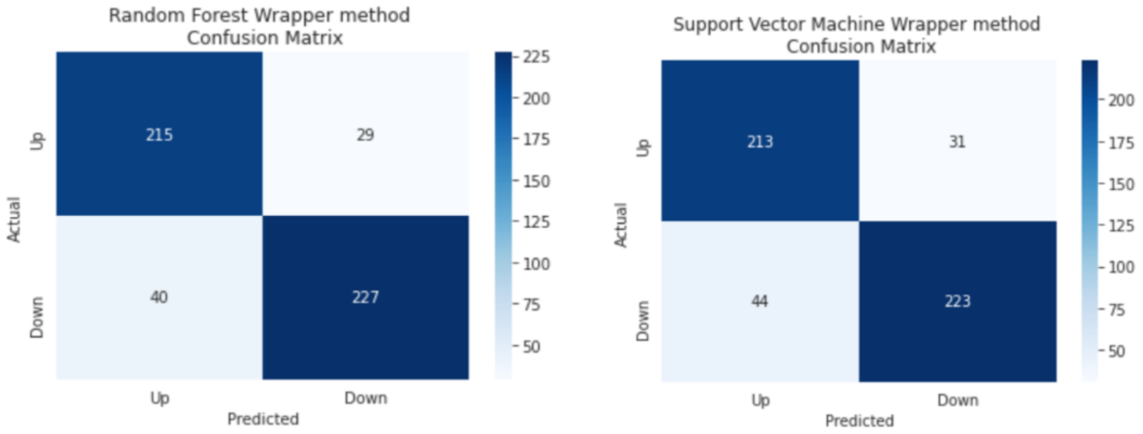
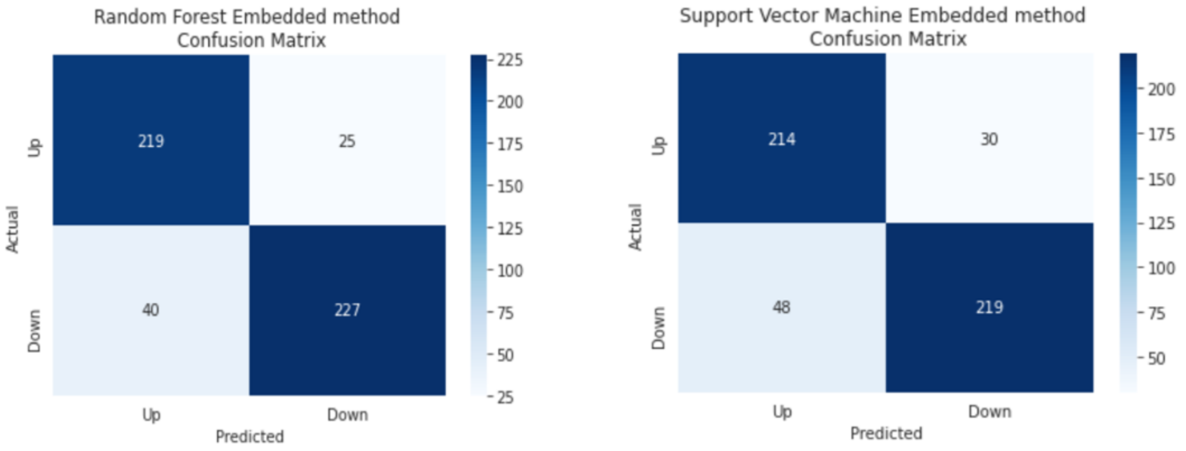


Figure 10: ***Random Forest Confustion Matrix vs SVM confusion Matrix***

From the above confusion matrix (Figure 10), Random Forest has captured 215 of the 244 up trends and 227 of the 267 down trends. Random Forest predicted high true positive and low false positive rate. Based on these results, it was clearly evident that Random Forest is the best performing algorithms in Wrapper feature selection method(Forward feature selection).Therefore, this algorithm can help an investor to make the right trading decision, i.e. buy or sell a stock at the selected time.

Embedded Feature Selection – Tree based Feature Selection

From the above results it was very evident that the Random Forest Classifier (benchmark Model) and SVM outperformed the Decision Tree classifier when classifying stock market trends. Random Forest shows 87.12 accuracy and SVM show 85.96. A good criterion to evaluate a better model is a high score of accuracy and F1 score. But when comparing Decision tree against our Random Forest classifier (benchmark model), results in terms of accuracy, precision, recall and F1 score were lesser than benchmark model.


Random Forest Embedded method Confusion Matrix

Support Vector Machine Embedded method Confusion Matrix

Figure 11: ***Random Forest Confustion Matrix vs SVM confusion Matrix***

From the above confusion matrix (Figure 11), Random Forest has captured 219 of the 244 up trends and 227 of the 267 down trends. Random Forest predicted high true positive and low false positive rate. Based on these results, it was clearly evident that Random Forest is the best performing algorithms in Embedded feature selection method(Tree based)Therefore, this algorithm can help an investor to make the right trading decision, i.e. buy or sell a stock at the selected time.

Without Feature Selection

From the above results it was very evident that the Random Forest Classifier (benchmark Model) and SVM outperformed the Decision Tree classifier when classifying stock market trends. Random Forest shows 85.94 accuracy and SVM show 86. A good criterion to evaluate a better model is a high score of accuracy and F1 score. But when comparing Decision tree against our Random Forest classifier (benchmark model) results in terms of accuracy, precision, recall and F1 score were lesser than benchmark model. Moreover, SVM outperformed the Benchmark model when no feature selection methods were applied.

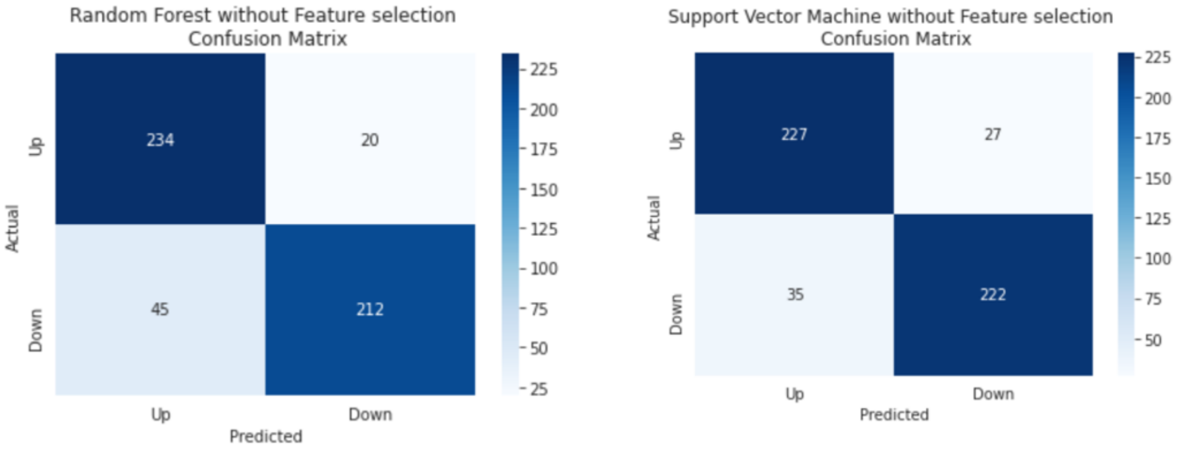


Figure 12: **Random Forest Confustion Matrix vs SVM confusion Matrix**

From the above confusion matrix (Figure 12), Random Forest has captured 234 of the 254 up trends and 212 of the 257 down trends whereas SVM has captured 237 of the 254 up trends and 222 of the 257 down trends. Both models have predicted high true positive and low false positive rate which is very promising. Based on these results, it was clearly evident that Random Forest and SVM classifiers are the best performing algorithms when all features were included. Therefore, these algorithms can help an investor to make the right trading decision, i.e. buy or sell a stock at the selected time.

Performance Measures:

The evaluation Metrics such as Precision, Recall, and F1 score were used to identify the best classifier from the algorithms.

Recall is the proportion of correct positive classifications from cases that are positive. Precision is the proportion of correct positive classifications from cases that are predicted as positive. The F1-score considers both precision and Recall computing the score. The higher the F1-score, the better the model. Embedded and Wrapper produce the highest F1- score (table 5&6) in our model. The following are the formulas of F1-score, Precision and Recall.

$$Precision = \frac{TP}{TP + FP}$$

$$TP = \text{True positive}$$

$$TN = \text{True negative}$$

$$FP = \text{False positive}$$

$$FN = \text{False negative}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

ROC- AUC curve

ROC AUC measures are one of the most important performance measures for the classification problem. ROC (Receiver Operating Characteristics) is the probability curve AUC (Area Under the Curve) measures the ability of the model to classify the classes. Higher the AUC better the Model. The below figure shows the ROC AUC curves for Random Forest Classifier with Wrapper method (left side) and Embedded method (Right side). Embedded method produces higher AUC (0.94) as displayed.

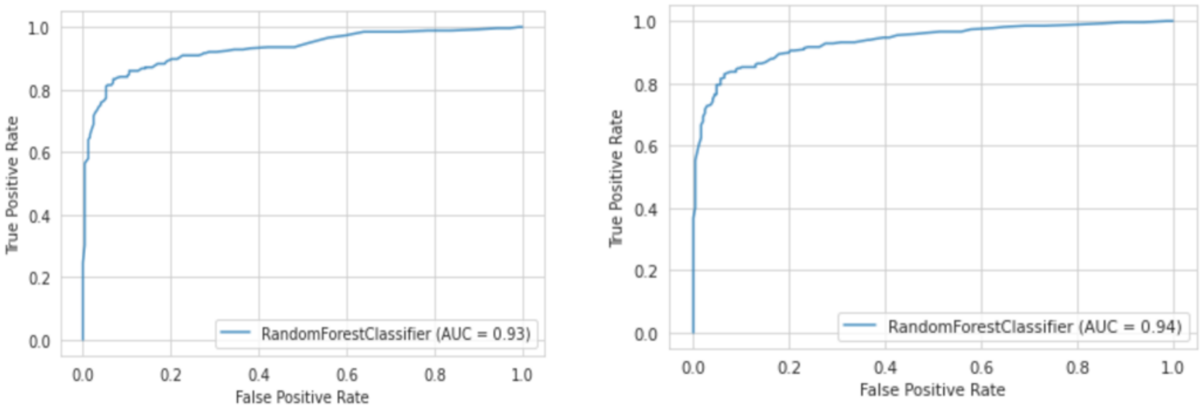


Figure 13: ***ROC AUC Score for Wrapper (left side) & Embedded Method (Right side)***

Cross Validation – Time Series split

Cross validation is a very powerful method to evaluate the performance of a model. The approach employed by cross validation is to split the data using one of the various methods and validate the training set to arrive at an accurate estimate. Cross validation aims to prevent various problems in a model like overfitting, selection bias and reduce noise.

The learning classifiers were validated using the time series split method which uses a "sliding window approach" for the k-fold split. The validator generates train and test indices to split the data and each split generates test indices that will be higher than before. Conceptually for the k'th split, the first k folds will be train set and (k+1) fold will be test sets.

This approach is suited to validate our model as the data contains time series data and other methods like simple split may not take into account the time component in the data. The model is validated comprehensively by running both 5 and 10 fold cross validation for the 3 classifiers with and without feature selection. As illustrated in Table 4 to 6, embedded method with 10-fold split generates the highest accuracy (85.98%) when using random forest classifier. Similarly, for 5-fold split, Random forest generates the highest accuracy (85.40%). The accuracy numbers imply that the models used were not overfitted and there is no selection bias

Conclusion and Future Work

Stock price trend prediction is a very complex and a challenging field as there are numerous factors that can affect the price of a stock which includes macro economic factors like inflation, GDP and microeconomic factors like substitution products, allocation of funds within a company, management structure etc. Apart from that investor sentiments and unexpected events like natural calamity can also have an influence on the company stock price. To provide a system which can reliably predict stock price trends is very important for investors and for the general health of the economy.

The high-level objective of this research was to analyze and build a strong model for classifying, testing and validating whether the price of a stock will go up or down on a particular day. The model attempts to construct technical indicators from the historic price data and runs different supervised learning algorithm on the technical indicators to arrive at an optimal model.

Identifying the appropriate technical indicators was one of the challenges encountered as there are 100s of technical indicators used in the industry and require a very specialized domain knowledge to master them. In this research, representative of each category of technical indicators (eg: Trend, Momentum, Volume, Volatility) was understood and used. Feature selection was then performed on the newly added attributes to identify the optimal set of attributes. For thoroughness one method each of Filter method (Mutual Information), Wrapper method (Forward selection), embedded method (Tree based) was used to extract the appropriate features. The model also tried to run using without feature selection and the results were compared. This process was repeated for all the machine learning algorithms and evaluated.

The machine learning based classifiers when ran against various permutations of feature selection (as described in the above paragraph) reveals that Random Forest (benchmark model) when used with Wrapper and Embedded method yield the best performance. When comparing Decision tree and Support vector machine against Random forest (Benchmark model) it was observed that SVM performs better than Decision tree. The performance was evaluated using accuracy, precision and recall. Therefore, the machine learning models with superior performance used in this research can be used by investors to predict market trends and make favorable investment decisions. Stock price data is added daily and hence the model can be retrained periodically and fine-tuned as required.

Future Work

The ideas and process used by this research to predict stock price trends have a great prospect for further development.

For the purpose of this research only technical analysis(indicators) was used as an input model. Future researchers are encouraged to implement other areas like Fundamental analysis and sentimental analysis and tune the models accordingly.

In this research various flavors of supervised learning were experimented and evaluated. It is recommended that unsupervised and reinforcement learning be explored in the future.

A natural extension of this paper will be to build a model to predict the trend for a portfolio of stocks. This will require further analysis like determining the correlation between the stocks and eliminating noise generated due to sector influences.

Appendix

List of Figures:

List of Tables:

Sources for Technical Indicators:

1. https://school.stockcharts.com/doku.php?id=technical_indicators:moving_averages
2. https://school.stockcharts.com/doku.php?id=technical_indicators:moving_average_convergence_divergence_macd
3. https://school.stockcharts.com/doku.php?id=technical_indicators:stochastic_oscillator_fast_slow_and_full
4. https://school.stockcharts.com/doku.php?id=technical_indicators:commodity_channel_index_cci
5. https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_band_width
6. https://school.stockcharts.com/doku.php?id=technical_indicators:on_balance_volume_obv

Literature Review References:

[1] In the research "Predicting Stock Prices Using Technical Analysis and Machine Learning" submitted on June-2010 submitted by Jan Ivar Larsen (Norwegian University of Science and Technology)

[2] In the research "Predicting the direction of stock market prices using Random Forest"(**(*arXiv:1605.00003*** [cs.LG]) *) submitted on May-2016, Luckyson Khaidem,Snehanshu Saha and Sudeepa Roy Dey*

*[3] In the research "Equity forecast: Predicting long term stock price movement using machine learning"(*arXiv:1603.00751 [cs.LG]) *Nikola Milosevic predicts long term market movement based on company fundamentals.*

[4] In the research, “PREDICTING AND BEATING THE STOCK MARKET WITH MACHINE LEARNING AND TECHNICAL ANALYSIS” Anthony Macchiarulo.

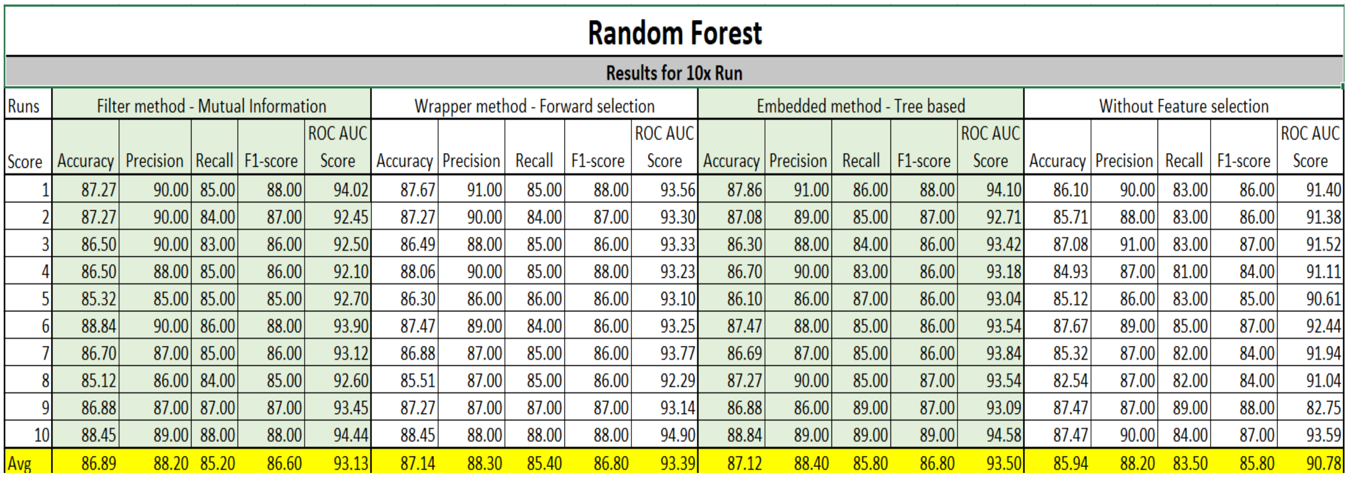
[5] In the paper "Machine Learning in Stock Price Trend Forecasting" by Dai and Zhang(2013)

[6] In the paper “Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market” L. Chen, Z. Qiao, M. Wang, C. Wang, R. Du and H. E. Stanley, "Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?" in IEEE Access, vol. 6, pp. 48625-48633, 2018, doi: 10.1109/ACCESS.2018.2859809

[7] *In the paper "Machine Learning Techniques for Stock Prediction" , Vatsal H. https://bigquant.com/community/uploads/default/original/1X/5c6d3b9959a8556a533a58e0ac4568dfc63d6ff4.pdf*

Model Simulation Results:

1. Random Forest Classifier Results

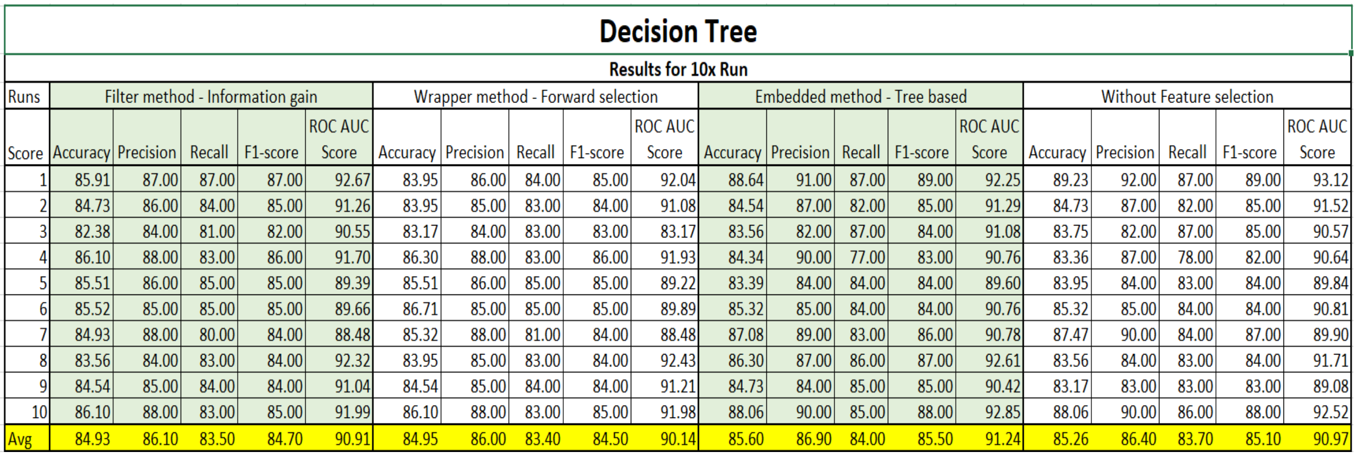


Random Forest

Results for 10x Run

Runs	Filter method - Mutual Information					Wrapper method - Forward selection					Embedded method - Tree based					Without Feature selection				
Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score
1	87.27	90.00	85.00	88.00	94.02	87.67	91.00	85.00	88.00	93.56	87.86	91.00	86.00	88.00	94.10	86.10	90.00	83.00	86.00	91.40
2	87.27	90.00	84.00	87.00	92.45	87.27	90.00	84.00	87.00	93.30	87.08	89.00	85.00	87.00	92.71	85.71	88.00	83.00	86.00	91.38
3	86.50	90.00	83.00	86.00	92.50	86.49	88.00	85.00	86.00	93.33	86.30	88.00	84.00	86.00	93.42	87.08	91.00	83.00	87.00	91.52
4	86.50	88.00	85.00	86.00	92.10	88.06	90.00	85.00	88.00	93.23	86.70	90.00	83.00	86.00	93.18	84.93	87.00	81.00	84.00	91.11
5	85.32	85.00	85.00	85.00	92.70	86.30	86.00	86.00	86.00	93.10	86.10	86.00	87.00	86.00	93.04	85.12	86.00	83.00	85.00	90.61
6	88.84	90.00	86.00	88.00	93.90	87.47	89.00	84.00	86.00	93.25	87.47	88.00	85.00	86.00	93.54	87.67	89.00	85.00	87.00	92.44
7	86.70	87.00	85.00	86.00	93.12	86.88	87.00	85.00	86.00	93.77	86.69	87.00	85.00	86.00	93.84	85.32	87.00	82.00	84.00	91.94
8	85.12	86.00	84.00	85.00	92.60	85.51	87.00	85.00	86.00	92.29	87.27	90.00	85.00	87.00	93.54	82.54	87.00	82.00	84.00	91.04
9	86.88	87.00	87.00	87.00	93.45	87.27	87.00	87.00	87.00	93.14	86.88	86.00	89.00	87.00	93.09	87.47	87.00	89.00	88.00	82.75
10	88.45	89.00	88.00	88.00	94.44	88.45	88.00	88.00	88.00	94.90	88.84	89.00	89.00	89.00	94.58	87.47	90.00	84.00	87.00	93.59
Avg	86.89	88.20	85.20	86.60	93.13	87.14	88.30	85.40	86.80	93.39	87.12	88.40	85.80	86.80	93.50	85.94	88.20	83.50	85.80	90.78

2. Decision Tree Classifier Results

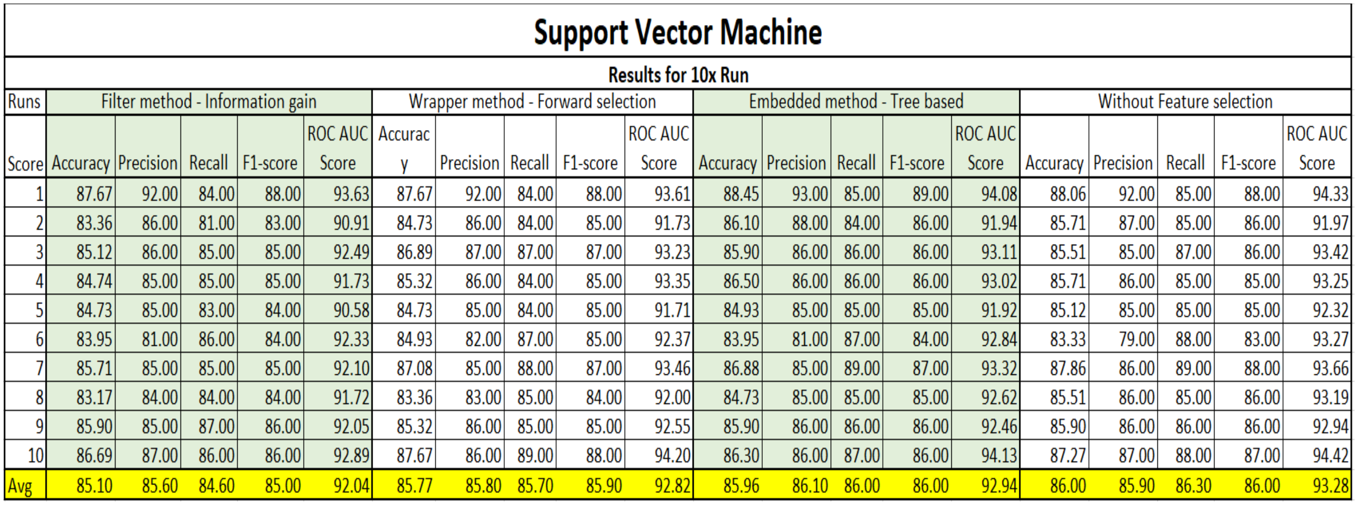


Decision Tree

Results for 10x Run

Runs	Filter method - Information gain					Wrapper method - Forward selection					Embedded method - Tree based					Without Feature selection				
Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score
1	85.91	87.00	87.00	87.00	92.67	83.95	86.00	84.00	85.00	92.04	88.64	91.00	87.00	89.00	92.25	89.23	92.00	87.00	89.00	93.12
2	84.73	86.00	84.00	85.00	91.26	83.95	85.00	83.00	84.00	91.08	84.54	87.00	82.00	85.00	91.29	84.73	87.00	82.00	85.00	91.52
3	82.38	84.00	81.00	82.00	90.55	83.17	84.00	83.00	83.00	83.17	83.56	82.00	87.00	84.00	91.08	83.75	82.00	87.00	85.00	90.57
4	86.10	88.00	83.00	86.00	91.70	86.30	88.00	83.00	86.00	91.93	84.34	90.00	77.00	83.00	90.76	83.36	87.00	78.00	82.00	90.64
5	85.51	86.00	85.00	85.00	89.39	85.51	86.00	85.00	85.00	89.22	83.39	84.00	84.00	84.00	89.60	83.95	84.00	83.00	84.00	89.84
6	85.52	85.00	85.00	85.00	89.66	86.71	85.00	85.00	85.00	89.89	85.32	85.00	84.00	84.00	90.76	85.32	85.00	84.00	84.00	90.81
7	84.93	88.00	80.00	84.00	88.48	85.32	88.00	81.00	84.00	88.48	87.08	89.00	83.00	86.00	90.78	87.47	90.00	84.00	87.00	89.90
8	83.56	84.00	83.00	84.00	92.32	83.95	85.00	83.00	84.00	92.43	86.30	87.00	86.00	87.00	92.61	83.56	84.00	83.00	84.00	91.71
9	84.54	85.00	84.00	84.00	91.04	84.54	85.00	84.00	84.00	91.21	84.73	84.00	85.00	85.00	90.42	83.17	83.00	83.00	83.00	89.08
10	86.10	88.00	83.00	85.00	91.99	86.10	88.00	83.00	85.00	91.98	88.06	90.00	85.00	88.00	92.85	88.06	90.00	86.00	88.00	92.52
Avg	84.93	86.10	83.50	84.70	90.91	84.95	86.00	83.40	84.50	90.14	85.60	86.90	84.00	85.50	91.24	85.26	86.40	83.70	85.10	90.97

3. Support Vector Machin Results



Support Vector Machine

Results for 10x Run

Runs	Filter method - Information gain					Wrapper method - Forward selection					Embedded method - Tree based					Without Feature selection				
Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score	Accuracy	Precision	Recall	F1-score	ROC AUC Score
1	87.67	92.00	84.00	88.00	93.63	87.67	92.00	84.00	88.00	93.61	88.45	93.00	85.00	89.00	94.08	88.06	92.00	85.00	88.00	94.33
2	83.36	86.00	81.00	83.00	90.91	84.73	86.00	84.00	85.00	91.73	86.10	88.00	84.00	86.00	91.94	85.71	87.00	85.00	86.00	91.97
3	85.12	86.00	85.00	85.00	92.49	86.89	87.00	87.00	87.00	93.23	85.90	86.00	86.00	86.00	93.11	85.51	85.00	87.00	86.00	93.42
4	84.74	85.00	85.00	85.00	91.73	85.32	86.00	84.00	85.00	93.35	86.50	86.00	86.00	86.00	93.02	85.71	86.00	85.00	85.00	93.25
5	84.73	85.00	83.00	84.00	90.58	84.73	85.00	84.00	85.00	91.71	84.93	85.00	85.00	85.00	91.92	85.12	85.00	85.00	85.00	92.32
6	83.95	81.00	86.00	84.00	92.33	84.93	82.00	87.00	85.00	92.37	83.95	81.00	87.00	84.00	92.84	83.33	79.00	88.00	83.00	93.27
7	85.71	85.00	85.00	85.00	92.10	87.08	85.00	88.00	87.00	93.46	86.88	85.00	89.00	87.00	93.32	87.86	86.00	89.00	88.00	93.66
8	83.17	84.00	84.00	84.00	91.72	83.36	83.00	85.00	84.00	92.00	84.73	85.00	85.00	85.00	92.62	85.51	86.00	85.00	86.00	93.19
9	85.90	85.00	87.00	86.00	92.05	85.32	86.00	85.00	85.00	92.55	85.90	86.00	86.00	86.00	92.46	85.90	86.00	86.00	86.00	92.94
10	86.69	87.00	86.00	86.00	92.89	87.67	86.00	89.00	88.00	94.20	86.30	86.00	87.00	86.00	94.13	87.27	87.00	88.00	87.00	94.42
Avg	85.10	85.60	84.60	85.00	92.04	85.77	85.80	85.70	85.90	92.82	85.96	86.10	86.00	86.00	92.94	86.00	85.90	86.30	86.00	93.28