Evaluating the effectiveness of model of three level moving average strategy

Name: Jayasuryan Mutytala,
210962009
Computer Science and Engineering(Artificial Intelligence & Machine Learning)
Manipal Institute Of Technology,Manipal Academy of Higher Education,
Manipal,India
jsmu.dev@gmail.com

***Abstract*— This project involves creating a logical model that analyses stock prices, trading volumes, and technical indicators to provide automated buy or sell recommendations in a high-frequency trading environment. By leveraging machine learning techniques, this model aims to enhance profitability while eliminating the impact of psychological biases associated with manual trading execution.** **Keywords—High frequency trading, stock prices, trading indicators, automated trading.**

I. Introduction

Utilizing Machine Learning models in trading strategies benefited many day and passive traders as they make use of advanced learning algorithms and data analysis to make profitable investment decisions. Machine learning empowers traders to analyze large amounts of historical stock market data, enabling them to classify the best possible times for buying and selling stocks as well as predicting future stock prices.

This paper focuses on the implementation of the Three Level Moving Average Strategy. The strategy involves using multiple Exponential Moving Averages (Short (SMA) usually the window is 5-20 days, Intermediate/Middle (MMA) usually the window is 50-200 days and Long (LMA) usually the window is 200-250 days) to calculate the trend following the indicator that reacts to price change. The conditions for buy and selling stock is based on:

- Buy: When SMA crosses above MMA and LMA indicating a positive uptrend.
- Sell: When SMA crosses below MMA and LMA indicating a potential downtrend.

The machine learning model involves the comparison of three supervised learning algorithms:

- Logistic Regression
- SVM
- Decision Tree's

The goal is to determine the most efficient algorithm for classifying the buy and sell points. This would help traders reduce the tendency of psychological behavior affecting their strategy while making decisions.

II. Literature Review

Classifying the best possible buying and selling points in a trading strategy is critical of importance to traders and researchers. The capability of the machine learning model to classify the signals highly can contribute to significant gains. The objective of this implementation is to evaluate the model based on the profit/loss returns with respect to the Three Level Moving Average Strategy. This literature review evaluates the existing algorithms and methods such as LTSM (Long Short-Term Memory Networks), ANN (Artificial Neural Networks), SVM (Support Vector Machine), Decision Trees and Logistic Regression.

SVM [1] is widely used for solving classification and regression tasks in the financial time series domain [2]. It works on structural risk minimization principles for achieving a global optimum. SVM's are widely used as they for these problems as they have superior generalization performance and high resistance to overfitting make it very effective for stock market predictions tasks. SVM's can misclassify training data to minimize overall error across test data. The benefit of SVM is that it computes the global optimum that makes it superior to ANN.

ANN [2] is a very popular method for stock forecasting. ANN's consist of a series of interconnected nodes to emulate individual neurons and are organized in several layers. Weights are assigned randomly to inputs, computing the output is with respect to input and weights. During the training phase it detects patterns in training and reassignment of weights is done. However, it is accurate when the data does not have any sudden variations [3]. Its accuracy is slightly above 50% since stock market data is dynamic and nonlinear.[4]

Logistic Regression [5] is utilized in applications of banking, corporate finance, investments. Most researchers determined data mining applications such as LSTM's or CNN's is a better choice for predicting future stock predictions for determining the best stocks to invest for gaining the maximum returns.

Decision Tree’s [6] which are commonly used while working with unambiguous data which is favorable for stock market. But they often result in overfitting the training data leading to poor generalization of the new data which is not optimal for the stock market as it is continuous. A comparative analysis done between Linear Regression and Decision Tree regardless of dataset size. The analysis shows that Linear Regression consistently beats Decision Tree Regression in predicting stock market patterns. Decision Tree Regression exhibits considerable variability, especially in its underperformance in predictions with different dataset sizes, whereas the accuracy of Linear Regression varies modestly with changes in data volume. As a result, linear regression stands out as a more dependable option for supervised learning regression algorithms in stock market prediction for both large and small datasets.

LSTMs [7] are very popular for stock market data analysis; they are a type RNN used for processing sequential data and its effective of learning long term dependencies. LSTMs are very effective in processing and making predictions on sequential financial time series data. Additionally, it has additional memory that remembers information over long periods, this is very effective as market patterns influence future stock prices.

The results of the numerous studies [8] on machine learning models for stock market prediction are not all the same. Support Vector Machines (SVM) were discovered by R. Shah [9] (2007) to have an accuracy of 60%, which is consistent with Kim's findings. Classification and Regression Tree (CART), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA) were compared by Soni and Srivastava (2010). CART outperformed LDA and QDA and had the lowest misclassification rate (56.11%) in the Indian stock market. According to Yakup et al. (2011), SVM was not as good at forecasting the Istanbul Stock Exchange National 100 Index as Artificial Neural Networks (ANN). Finally, Yanshan Wang [10] (2014) forecasted the movement of the Hong Kong and Korean stock markets with remarkable accuracy using an integrated Principal Component Analysis and SVM (PCA-SVM) model.

The dataset used is historical Apple stock data taken from yahoo finance. Metrics such as opening and closing prices, highs and lows and trading volumes are essential for the classification of signals (Buy and Sell). The implemented machine learning algorithms are Logistic Regression, Support Vector Machines (SVM) and Decision Tree’s, the models to adapt to the dynamic environment of the stock market conditions classify the optimal buy and sell signals. Logistic Regression is efficient when there is a linear relationship between features and labels. This needs less computational power. SVM is very reliable for analyzing the complex relationships between data, especially when the decision boundary between buy and sell is very complicated. Decision Trees are very effective for analyzing non-linear relationships in the dataset.

Despite the model’s capabilities to classify the buy and sell signals, there are several various factors (geopolitical events, macroeconomic shifts) that influence the dynamic conditions of the stock market environment. Overfitting was a very regular issue that was encountered while trying compute the best possible parameters for improving the model accuracy, while implementing pipelining and regularization for the algorithms since the models trained with past data and they would perform poorly on new data, which is a significant concern. However, with the ongoing developments in Deep Learning models such as Long Short-Term Memory (LSTM) Networks and Convolutional Neural Networks (CNN), are more reliable as they detect patterns in stock price charts and process sequential data efficiently. However, the scope of the model only evaluates supervised learning techniques mentioned above.

In the world of stock markets determining the best strategy is the primary goal of every trader. The use of ML models provides an approach for improvising buy and sell decisions by using previous stock data. However, challenges arise due to the dynamic environment of the market which increases the complexity of attaining consistent profitability. Advanced deep learning algorithms such as LSTM and CNN provide better results with its ability to analyze stock chart patterns improving the overall model accuracy. The combination of data science and conventional trading insights may open the door to more complex and dependable trading strategies that strike a balance between computational efficiency and predictive accuracy. Machine learning and deep learning are becoming more and more important in the never-ending quest to optimize stock trading techniques.

III. Limitations and Goals

A. *Limitations*

This topic focuses evaluating the state-of-the-art supervised machine learning algorithms such as Logistic Regression, Support Vector Machines and Decision Tree’s. However, there are limitations this approach does not focus on unsupervised learning methods and deep learning techniques which provide higher accuracy and precision as algorithms such as LSTM’s can be utilized for taking in account of stock market patterns which is essential for making more reliable and profitable trades as the model will be able recall essential information and apply it to the volatile stock market data.

B. *Goals*

- *Evaluation of the Models:* The primary focus to evaluate the effectiveness of the strategy implemented with Logistic Regression, SVM and Decision Tree’s for classification of buy and sell signals. Additionally computing the overall loss/profit returned of each model.

- *Technical Indicator Analysis:* This method determines and examines Exponential Moving Averages (EMAs) over various time periods to use them as technical indicators and provide information about the momentum and trends of stock prices.

- *Trading Signal Generation:* The model should effectively generates buy and sell signals by utilizing these EMAs. These signals—which are based on the crossover of the short-, medium-, and long-term EMAs—are essential for determining possible trading entry and exit points.

- *Visualization of model performance on strategy*: To gain an understanding of effective trading signals, EMAs, and stock market behaviour and the influence of generated signals on trading decisions is facilitated by data visualization.

- *Performance Evaluation:* Plotting cumulative returns is used to evaluate the performance of trading strategies. This facilitates comprehension of the strategy's long-term efficacy.

- *Predictive Modelling:* The state-of-the-art machine learning models, such as Decision Tree Classifier, SVM, and Logistic Regression. These models demonstrate how your code can be used to leverage AI for financial analysis, as they have been trained to predict trends in the stock market.

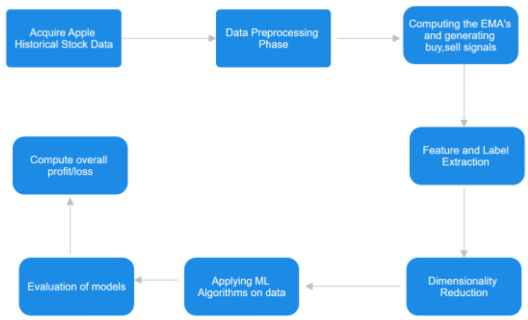

Figure 1: Model Architecture

IV. METHODOLOGY

A. Summary

The methodology used for classification of buy and sell signals following the three-level moving average strategy is given below:

- Data Preprocessing Phase
- Implementation of the Three Level Moving Average Strategy and computation of estimated Moving Averages (Long, Short and Medium)
- Identifying the technical indicators(features) and labels.
- Data Visualization of the strategy.
- Dimensionality Reduction through Principal Component Analysis (PCA)
- Training the Logistic Regression, Support Vector Machine and Decision Tree models.
- Computing the model's profit/loss returns for the strategy.
- Data Visualization of the results.
- Evaluation of the models.

The proposed methodology implements a wide scope on the prediction and classification of buy/sell signals. The procedure begins with loading the Apple (AAPL) historical stock data and involves computing daily returns, Exponential Moving Averages (EMA's) computed over short, medium, and long periods, that is essential to generating buy and sell signals at critical crossing points. These signals indicate potential opportunities for making a transaction to buy or sell.

Once data preprocessing, cleaning and determination of features and labels is completed. The dataset undergoes PCA (Principal Component Analysis), for dimensionality reduction, allowing more efficient training model training and visualization. Upon the completion of this phase, Logistic Regression, Support Vector Machines and Decision Tree models are trained on EMA's for the purpose of predicting the stock market trends.

The results obtained from the models are then visualised for evaluating the strategy. The cumulative returns are also used in analyse the computational effectiveness of model over the time. Data visualization is an integral component for the understanding of signal efficacy and stock market dynamics. The model is utilising classification reports to compare the accuracy of the Logistic Regression, SVM and Decision Tree in stock trend predictions.

But there are drawbacks, such as the possibility of overfitting, particularly when using the Decision Tree model. Prediction accuracy may be hampered by stock market volatility, unanticipated events, and reliance on EMAs as the main indicators. Although PCA simplifies data, it may result in information loss, which could impair model performance. Look-ahead bias is another issue that could distort the training process. These problems serve as a reminder of the difficulties and factors to be considered when using machine learning to make stock market predictions.

B. Data Preprocessing Phase

Dataset used is financial time series Stock data for Apple (AAPL). It consists of a broad range of technical indicators such as Open, High, Low, Close, Adjusted Close and Volume.

- Open: Determines the stock price value at the beginning of the day.
- High: The price at which trading activity peaks indicates the strength of buyers at their greatest.
- Low: Lowest stock price on the day. Displays minimum market value.

- Close: Final price which the stock trades occur during the business hours.
- Adjusted Close: A more accurate representation of the stock's value, adjusted for events such as rights offerings, stock splits, and dividends.

	Date	Open	High	Low	Close	Adj Close
2022-09-12	2022-09-12	159.589996	164.259995	159.300003	163.429993	162.466171
2022-09-13	2022-09-13	159.899994	160.539993	153.369995	153.839996	152.932739
2022-09-14	2022-09-14	154.789993	157.100006	153.610001	155.309998	154.394073
2022-09-15	2022-09-15	154.649994	155.240005	151.380005	152.369995	151.471420
2022-09-16	2022-09-16	151.210007	151.350006	148.369995	150.699997	149.811264

Figure 2: Apple stock Dataset

The date column is set as column index for time series analysis.

Actual Returns: it is the profit or loss of a trade over a specified period. This is expressed in form of percentage, when it is positive, this considered as a profit and if it is negative, it is considered negative.

Computing Actual Returns of a stock

$$Actual\ Returns = \frac{current\ price\ - original\ price}{original\ price}$$

Essentially computing the percentage change of the closing price will provide this information.

Strategy Returns: Profit or loss determined over time by using the following buy/sell signals or strategies to execute stock market transactions.

Computing Strategy Returns

$$strategy\ returns\ = actual\ returns\ \times signals$$

C. *Implementation of three level moving average strategy and computation of Exponential Moving Averages(EMA's)*

The moving average is a feature that accounts for the average change data over time. Traders use moving averages for the purpose of keeping note of price trends for stocks and understanding how to profit from price movements for stocks. EMA emphasizes more weight to recent trading days. This is reliable for short-term traders for whom longer term previous data would not be relied heavily on. This computed by averaging a series of prices providing equal weight to each of the prices.[11]

$$EMA = \ Price(t) \times 2/(N-1) \times EMA(y) \times \left(\frac{2}{N+1}\right)$$

t: Today
y: Yesterday
N: number of days in EMA

Three level moving average strategy:

The three-level moving average strategy is primarily used for observing the trend of the stock prices, Where the prices go up or down or stable. The strategy involves utilizing several EMA calculations (short, medium, and long) and subtracts the lag which is a delay in reacting to price changes.

$$Three\ level\ moving\ average = (3 \times EMA_1) - (3 \times EMA_2) - (3 \times EMA_3)$$

EMA_1: EMA
EMA_2: EMA of EMA_1
EMA_3: EMA of EMA_2

The model makes use of short moving average (SMA), medium moving average (MMA) and long moving average (LMA).

Short Moving Average (SMA):

- Usually computed over a brief time frame, between five to fifteen days. Since it responds swiftly to price fluctuations, it can be helpful in spotting transient trends.

Medium Moving Average (MMA):

- The medium moving average typically covers a range of periods, from 20 to 50 days.
- It's good for confirming trends and locating possible support or resistance levels because it strikes a balance between being sensitive to recent price changes and having the capacity to filter out market noise.

Long Moving Average (LMA):

- Computed over a 100–200 days.
- frequently used to spot long-term market trends and is less sensitive to daily price swings.

Generating Buying and Selling Signals

- Buy: When SMA crosses above MMA and LMA indicating a positive uptrend.
- Sell: When SMA crosses below MMA and LMA indicating a potential downward.

	Close	Actual_Returns	Short	Middle	Long	Buy	Sell
2022-09-13	153.839996	-0.058680	153.839996	153.839996	153.839996	NaN	NaN
2022-09-14	155.309998	0.009555	154.329997	153.973633	153.885934	155.309998	NaN
2022-09-15	152.369995	-0.018930	153.676663	153.827847	153.838560	152.369995	NaN
2022-09-16	150.699997	-0.010960	152.684441	153.543497	153.740480	NaN	NaN
2022-09-19	154.479996	0.025083	153.282959	153.628634	153.763590	NaN	NaN

Figure 3: Computing SMA, MMA, LMA and Generation of Buy/Sell signals.

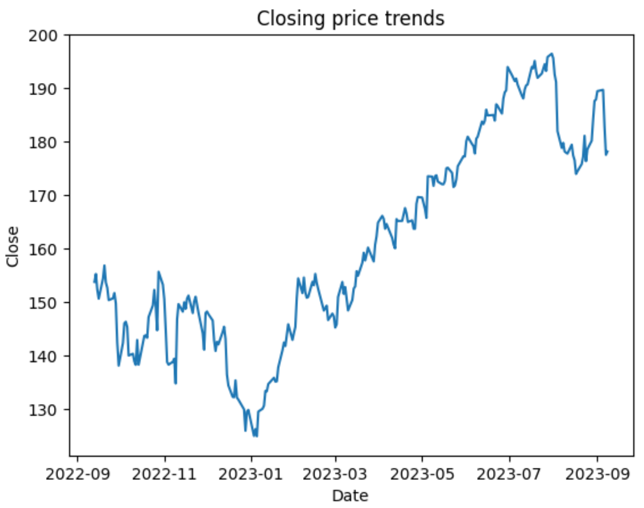

Figure 4: Visualizing the variation of the close price over time

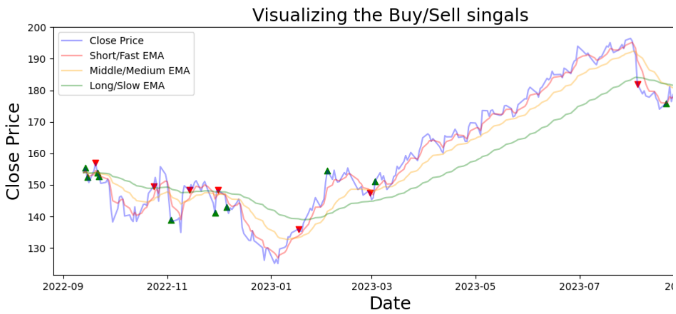

Figure 5: Visualizing SMA, MMA, LMA and generating buy and sell signals.

D. *Identifying features and labels*

The features used are the calculated SMA, MMA and LMA. Labels are Buy (1) and Sell Signals (0), 2 conditions are used to generate these signals. The Signal is set 1 when the actual returns is greater than 1 and if there exists a sell price for that column. Alternatively, the signal is set 0 and if there exists a sell price for that column if actual returns are less than 0.

This could be improvised by considering features such as Relative Strength Index (RSI) which identifies when a stock is overbought (RSI > 70) this is indicates a sell signal or oversold (RSI < 30) which indicates a buy signal.

E. *Data Visualization of the strategy*

The daily cumulative returns is also computed by calculating the cumulative product of the daily percentage change.

$$Cumulative\ return\ =\ \prod_{i=1}^{t}(1\ +\ Strategy\ Returns)$$

This section involves the analysis of the plots obtained (Figure 4,5,6).

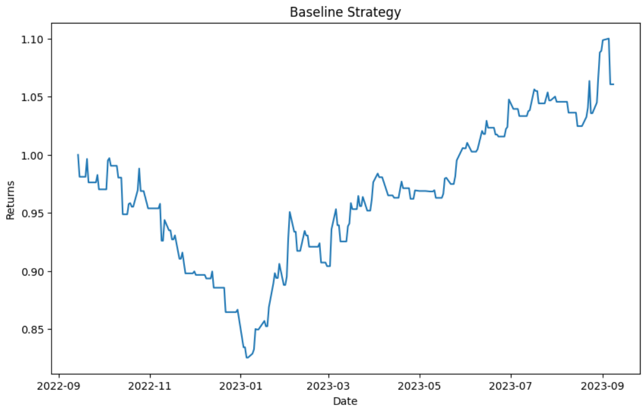

Figure 6: Cumulative returns of the Three Level Moving Average strategy

Figure 6 depicts the performance of the strategy over time. The returns axis indicates that the strategy is evaluated in terms of cumulative returns starting from a normalised value 1. The graph shows volatility as it is evident with the declining trend followed by a period of recovery and general uptrend. The sudden increase at the end indicates that the strategy worked well or that the market had recently moved in the right direction.

Figure 4 shows the stock's closing price trends without the use of any additional indicators. It indicates the movement to the stock price movement over time, highlighting volatility and the general trend of the stock for short-term traders, these periods of high volatility, marked by sharp peaks and troughs, may prove to be critical.

Figure 5 In order to indicate possible buy and sell opportunities, this graph combines the stock's closing price with short, medium, and long Exponential Moving Averages (EMAs). The points where the strategy would recommend buying the stock—usually when the short EMA crosses above the longer EMAs—are indicated by the green upward-pointing triangles or buy signals. On the other hand, the red triangles pointing downward (sell signals) indicate when it is best to sell or steer clear of buying, and they typically appear when the shorter EMA crosses below the longer EMAs. The overall trend is upward, indicating a bullish market; however, the model has detected multiple points at which action could be advantageous.

F. *Dimensionality Reduction using PCA(Principal Component Analysis)*

Principal Component Analysis (PCA) is statistical technique for dimensionality reduction and is widely used for analyzing large datasets containing many dimensions or features while preserving the maximum amount of information, which allows us visualize data.

The feature set used in the code contains SMA, MMA and LMA. PCA converts the set of correlated EMA's into a set of linearly uncorrelated variables known as principal components. PCA is used to simplify the stock market data. Ensuring that features contribute equally to the analysis, the data is first standardized before this reduction is carried out. Machine learning models such as Support Vector Machines (SVM) and Logistic Regression can be applied more accurately and efficiently when Principal Component Analysis (PCA) is used to identify the principal components that account for most of the variance in the data. In addition to making the process of training the model easier, the resulting lower-dimensional space helps to visualize the data and the decision boundaries of the model, offering distinct insights into the market trends that the EMAs are tracking.

G. *Training the Logistic Regression, Support Vector Machine and Decision Tree models.*

- *Logistic Regression:* This model classifies data into two categories (buy and sell) based on the probability of outcomes. The model is fitted with the data after scaling and performing PCA, reducing dimensionality and multicollinearity which makes the model very robust. The model understands the relationship between features (such as EMAs) and the likelihood of each trading signal (buy/sell). It assesses how changes in the values of EMAs are related to the likelihood of a stock being a good buy. This evaluated based accuracy, precision, recall and F1-score.

- *Support Vector Machines (SVM):* The SVM model is utilized for its capability to identify the optimal boundary (hyperplane) that divides distinct classes (buy and sell signals). Much like Logistic Regression, SVM is trained using PCA-reduced features. The focus is to maximize the distance between the data points belonging to the two categories. The key to its success in predicting stock movements lies in its ability to distinguish between profitable and unprofitable trades, even when dealing with intricate or subtly varied data.

- *Decision Tree:* Decision Trees classify from the rules inferred from features. It's trained on PCA dimensionality reduced data. The model splits data into subsets with respect to the value of chosen features effectively learning where the outcome is bought or sell. The effectiveness of the model is measured by its ability to comprehend nonlinear relationships between features and label without resulting in overfitting of data.

H. *Computing the model profit/loss returns*

The profit/loss returns calculation of the model cumulative returns. The model performs similar computational procedure as cumulative returns were calculated previously. The exception is that cumulative returns with respect to model predictions are calculated.

$$\begin{aligned} &profit\ or\ loss \\ &= model\_cumulative\_returns.iloc[-1] - 1 \end{aligned}$$

If the result is positive, it indicates the model generates profit and if the result is negative, as it indicates loss.

V. RESULT ANALYSIS

A. *Logistic Regression Analysis:*

	Model_Predictions	Actual_Returns	Strategy_Returns
2022-10-31	1	-0.015410	-0.015410
2022-09-22	1	-0.006375	-0.006375
2023-06-06	1	-0.002060	-0.002060
2023-01-18	1	-0.005370	-0.005370
2023-04-28	1	0.007541	0.007541
...	...	...	...
2023-02-17	1	-0.007547	-0.007547
2023-02-27	1	0.008248	0.008248
2022-12-28	1	-0.030685	-0.030685
2023-04-04	1	-0.003250	-0.003250
2022-10-25	1	0.019338	0.019338

Figure 7: Logistic Regression predictions data frame.

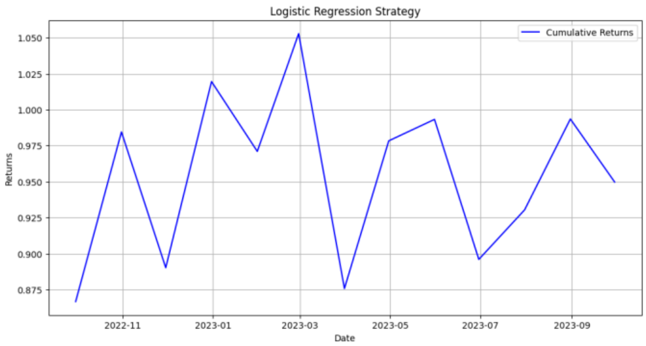

Figure 8: Cumulative strategy returns for logistic regression model.

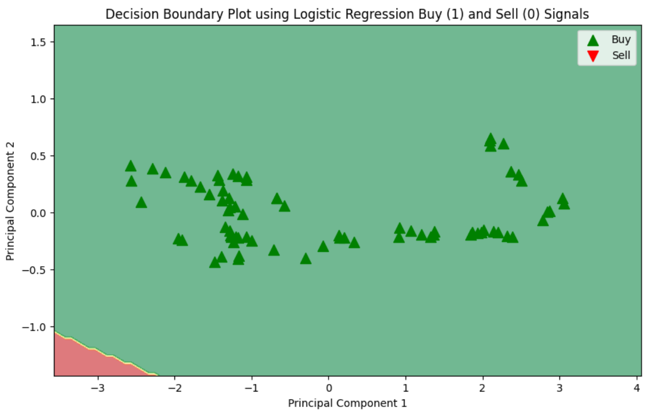

Figure 9: Decision Boundary returns for logistic regression model.

Figure 8 indicates the performance of the investment over time is depicted in the cumulative returns plot for the Logistic

Regression strategy. The relative growth or shrinkage of the investment is represented by the y-axis, where values above and below 1.0 denote gains and losses, respectively. The sharp swings in the graph indicate that the model's predictions result in a volatile trading strategy, with bursts of quick profits interspersed with equally rapid losses. This volatility in the investment's value suggests that the strategy may be highly risky, necessitating additional analysis and perhaps even model improvement. It also emphasizes how important it is to have a thorough risk management strategy in place to guarantee that the strategy's long-term returns are steady and predictable.

A logistic regression model on PCA-reduced data produced the decision boundary plot shown in Figure 9, which shows the model's attempt to classify 'buy' and 'sell' signals using PC1 and PC2, the two principal components, as its axes. The boundary marks the point at which the model shifts from predicting a "buy" signal to a "sell" signal; it is most likely set at a logistic regression threshold of 0.5. According to this linear decision boundary, the model distinguishes between the two classes using a simple rule that was deduced from the training set.

```
Profit/loss returns from the logistic regression model 0.03574767803595469
```

Figure 9: Gain/Loss returns from the Logistic regression model.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	37
1	0.51	1.00	0.67	38
accuracy			0.51	75
macro avg	0.25	0.50	0.34	75
weighted avg	0.26	0.51	0.34	75

Figure 10: Classification report for the Logistic regression model

The model's precision score in predicting buy opportunities suggests that the model can distinguish buy signals from sell signals. Still, the overall performance in differentiating between buy and sell signals is not ideal. The model's poor performance for both classes is evident from the precision and recall metrics, which also point to difficulties in correctly classifying the two kinds of trading signals. There may be a variety of underlying causes for this, such as complex or noisy data patterns, a lack of distinct features, or a need for more intricate model calibration. Essentially, this limits the predictive power of the model at this point, indicating that it would be necessary to improve the feature set, model complexity, or data preprocessing in order to improve the model's ability to classify.

B. Support Vector Machine Analysis:

	Model_Predictions	Actual_Returns	Strategy_Returns
2022-10-31	0	-0.015410	-0.000000
2022-09-22	0	-0.006375	-0.000000
2023-06-06	1	-0.002060	-0.002060
2023-01-18	1	-0.005370	-0.005370
2023-04-28	0	0.007541	0.000000
...	...	...	...
2023-02-17	0	-0.007547	-0.000000
2023-02-27	0	0.008248	0.000000
2022-12-28	1	-0.030685	-0.030685
2023-04-04	0	-0.003250	-0.000000
2022-10-25	0	0.019338	0.000000

Figure 11: SVM model predictions data frame

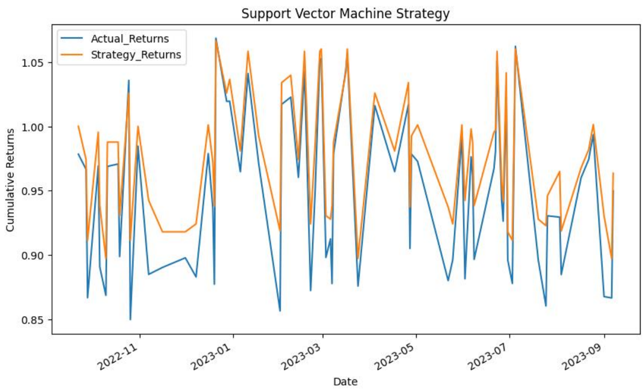

Figure 11: SVM strategy returns and actual returns.

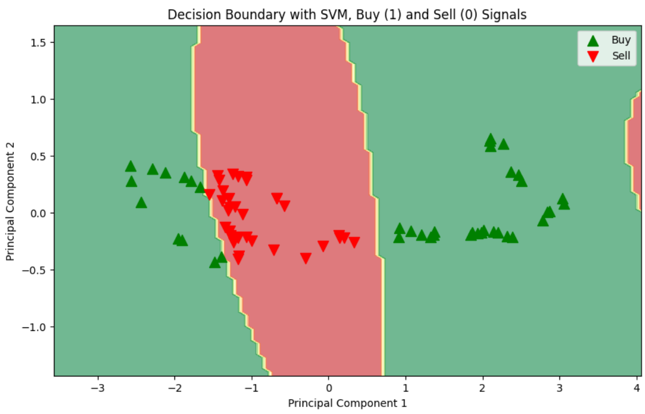

Figure 12: SVM decision boundary plot

Figure 11 indicates the cumulative returns of the actual stock prices over time are plotted against the returns of the SVM strategy in the second plot. The returns that would result from holding the stock are represented by the blue line (Actual Returns), and the returns that would result from adhering to the buy and sell signals of the SVM model are represented by the orange line (Strategy Returns).

Line Closeness: The SVM strategy closely mimics the stock's performance where the lines move in close proximity to one another. This shows that the model's buy signals correspond well with periods of rising stock price and its sell signals correspond with periods of falling stock price.

Divergence Points: The strategy is either outperforming or underperforming the straightforward buy-and-hold method at the points where the lines diverge. More so than in the blue line, steep dips in the orange line indicate times when the SVM model may have sent out sell signals during rising market trends, which would have prevented the strategy from realizing gains or even from selling at a loss.

Underperformance of the Strategy: The SVM strategy is generally underperforming when compared to the buy-and-hold strategy if the orange line is continuously below the blue line or has more noticeable dips. It suggests that there may be inefficiencies in the model's buy and sell signal timing, which could result in losses during periods of erratic market conditions or the loss of possible profits.

The decision boundary plot of the SVM model in Figure 12 illustrates how two principal components from PCA were used to classify market signals into buy and sell. The decision boundary of the SVM is represented by the curved line, which shows the best possible distance in feature space between the two classes. The model's balance between identifying intricate market patterns (preventing underfitting) and guaranteeing generalizability to new data without being unduly sensitive to noise (preventing overfitting) is reflected in the curve's shape. The model's accuracy is indicated by the points' placement in relation to the boundary; misclassifications are visible when points are placed on the wrong side of the boundary.

The SVM classifier's classification report displays performance metrics that are close to the threshold of chance, with both classes' precision, recall, and F1-scores ranging from 0.49 to 0.51. With an accuracy of 0.49, the model indicates that it makes accurate predictions less than half of the time. Although the near-equal macro and weighted averages across the board suggest that the model's ability to distinguish between the classes is marginal and not significantly better than flipping a coin, both classes have similar support, indicating a balanced dataset. This emphasizes how important it is to improve the model—possibly by investigating different modelling approaches, hyperparameter optimization, or more advanced feature engineering.

```
Profit/loss returns from the SVM model 0.02586289920461371
```

Figure 13: Profit return SVM strategy

	precision	recall	f1-score	support
0	0.49	0.46	0.47	37
1	0.50	0.53	0.51	38
accuracy			0.49	75
macro avg	0.49	0.49	0.49	75
weighted avg	0.49	0.49	0.49	75

Figure 14: Classification report from the SVM model.

The report indicates that the higher recall and F1-score for class 1 suggest that the SVM model is more successful in identifying "buy" signals than "sell" signals. But the precision and accuracy are only marginally better than a wild guess, suggesting that the model's predictive power is still somewhat limited. The moderate F1-scores also point to a trade-off between recall and precision, especially when it comes to the model's inability to sell predictions.

C. Decision Tree

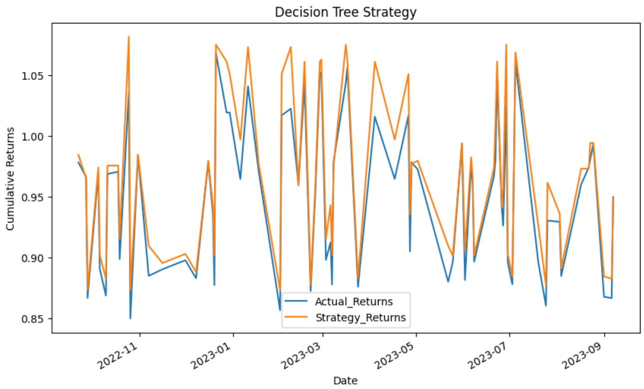

Figure 14: Decision Tree cumulative returns

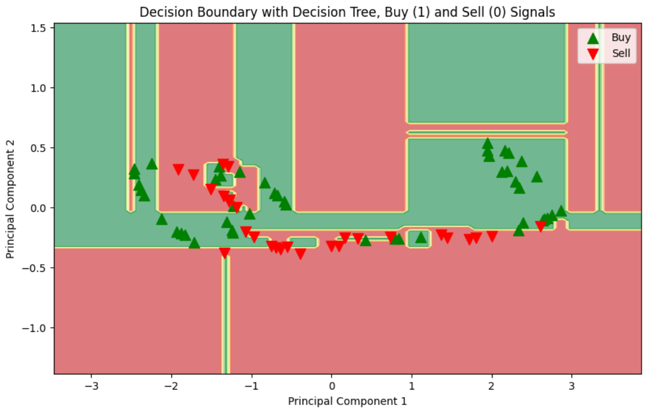

Figure 15: Decision Boundary plot

Figure 14 compares the cumulative returns of a trading strategy by Decision Tree model to the stock's actual returns over time. The blue line indicates the actual returns, the performance of the stock if it had been held without trading, which represents the actual returns. The orange line, on the other hand, indicates the returns of a strategy that follows the buy and sell signals that the Decision Tree recommends. By examining the two lines, it is possible to determine how volatile they are as well as the times when the strategy outperforms or underperforms the actual returns. The strategy's performance and the actual stock returns show a clear correlation, but the strategy does not consistently outperform the buy-and-hold strategy, as shown by the orange line that frequently crosses below the blue line.

The decision tree classifiers, decision boundary plot results in orthogonal, step-like divisions between the 'Buy' and 'Sell' regions, as seen in the Decision Tree's decision boundary plot. When 'Buy' and 'Sell' signals are present in the same regions, it may be a sign of overfitting caused by noise in the data, misclassifications, or the result of closely spaced decision thresholds. A model's complexity can be inferred from the number and size of its steps in the decision boundary; a model with more steps and smaller steps may be more complex and may not generalize very far beyond the training set.

	Model_Predictions	Actual_Returns	Strategy_Returns
2022-10-31	1	-0.015410	-0.015410
2022-09-22	0	-0.006375	-0.000000
2023-06-06	1	-0.002060	-0.002060
2023-01-18	1	-0.005370	-0.005370
2023-04-28	0	0.007541	0.000000
...	...	...	...
2023-02-17	0	-0.007547	-0.000000
2023-02-27	0	0.008248	0.000000
2022-12-28	0	-0.030685	-0.000000
2023-04-04	0	-0.003250	-0.000000
2022-10-25	1	0.019338	0.019338

Figure 15: Decision Tree predictions data frame

```
Profit/loss returns from the decision tree model 0.08195375841803876
```

Figure 16: Decision Tree model returns loss.

The decision tree model makes the biggest profit indicates that it is reliable at identifying the complex feature interactions and patterns in stock data. Even though its predictive accuracy isn't always high, its hierarchical decision-making can effectively capitalize on market trends and possibly provide a few decisions that have a significant financial impact. However, since Decision Trees are prone to learning noise, which can inflate performance on training data, one must exercise caution to avoid overfitting. To ensure that the observed profits are due to the model's ability to generalize rather than the model being learned from the training dataset, robustness must be ensured through additional validation. Moreover, the model's risk strategy, whether conservative or aggressive, can have a big impact on profitability, especially when it comes to controlling and capturing big market movements while minimizing losses.

	precision	recall	f1-score	support
0	0.54	0.41	0.46	37
1	0.53	0.66	0.59	38
accuracy			0.53	75
macro avg	0.53	0.53	0.52	75
weighted avg	0.53	0.53	0.53	75

Figure 17: Classification report for the decision tree

The higher precision, recall, and F1-score scores for class 1 indicate that the Decision Tree model is marginally better at predicting 'Buy' signals (class 1) than 'Sell' signals (class 0), according to the classification report. Nonetheless, the model's overall accuracy of 0.53 means that it can only predict outcomes marginally better than random guessing, which is thought to have an accuracy of roughly 0.50. The model shows some predictive power, particularly in 'Buy' signal identification; however, as evidenced by the modest F1-scores for both classes, there is significant room for improvement in the model's performance.

D. *Hyperparameter tuning:* This process involves determination of the most optimal hyperparameters(parameter values that control the learning process and compute the model parameters that the algorithm utilises for learning) for the machine learning algorithm. In the machine learning pipeline, hyperparameter tuning is a crucial step that improves trading strategies by fine-tuning models to better capture market patterns and maximize profits. To avoid overfitting and preserve the capacity to detect underlying trends, it entails experimenting with a variety of hyperparameter values. This procedure not only increases a model's predictive accuracy, which results in more dependable trading signals, but it also makes it possible to compare various models fairly because robust performance metrics are used to evaluate them under the same circumstances. In the end, hyperparameter tuning aims to identify the model that yields the maximum profit in real-world trading applications while also functioning well in terms of accuracy.

```
Profit/loss returns from the logistic regression model: 0.051958737210726635
Profit/loss returns from the SVM model: 0.21673706227157186
Profit/loss returns from the decision tree model: 0.06792207161052444
The best model based on profit is: SVM with a profit of 0.21673706227157186
```

Figure 17: Results after hyperparameter tuning

After hyperparameter tuning, the Support Vector Machine (SVM) proved to be the best option for financial analysis because of its strong data separation capabilities, adaptability to non-linear relationships, resilience to outliers, efficiency in high-dimensional spaces, good generalization to new data, and precisely adjusted hyperparameters. All of these qualities allowed SVM to perform better than other models in terms of precisely forecasting financial trends and yields, which is why it was chosen for this particular financial analysis task.

VI. CONCLUSION

The study of Decision Tree, Support Vector Machine (SVM), and Logistic Regression models has yielded valuable insights into the possibilities and constraints of using machine learning on financial time series data. Overall, the results were mixed, even though each model showed distinct characteristics regarding strategy performance and decision boundaries.

The Logistic Regression model had moderate accuracy provided a probabilistic method for classifying buy and sell signals. The performance metrics showed that even though its decision boundary was linear, it was not able to fully capture the intricate patterns present in stock price movements.

Among the models tested, the SVM model was the most profitable, indicating that it may be a more dependable model overall. Given the complexity of financial market structures, the ability to form non-linear decision boundaries enabled it to capture more complex patterns in the data. Despite not outperforming the market on a consistent basis, the SVM strategy's evaluation period profitability supports further research and application.

This observation does not, however, lessen the need for thorough back testing against out-of-sample data in order to verify the model's resilience and ability to generalize beyond the particular market conditions that were encountered during the testing stage. Furthermore, it will be essential to include transaction costs, market impact, and risk management considerations in the real world that will require considering risk management, market impact, and transaction costs.

However, due to limitations and the scope of my knowledge I only utilised supervised machine learning algorithms. This model will be effective while executing short term trades over long term trades. The most optimal solution for using this strategy for using it as a long-term trading strategy hourly time series data over 3-4 years would have been very ideal. The investigation was limited to the using 1 year of daily stock data. Additionally, more technical indicators such as [RSI, Moving Average Convergence Divergence (MACD), Bollinger Bands and Volume Weighted Average Price (VWAP)] features for improvising the accuracy and precision for computing buy and sell signals.

The scope of this project can be improvised with deep learning algorithms. Long Short-Term Memory (LSTM) networks, have the potential to significantly improve stock market prediction because of their ability to identify high-dimensional non-linear relationships and temporal patterns present in financial time series. With their ability to incorporate a variety of data sources, including unstructured and alternative data, LSTMs provide real-time analysis and the ability to adjust to fluctuations in the market. The creation of hybrid models that combine LSTMs with other methods, improving model interpretability, integrating risk management into deep learning architectures, and guaranteeing adherence to changing regulatory standards are some of the future research areas. By using deep learning, this research path aims to provide sophisticated, flexible, and transparent algorithmic trading solutions.

In conclusion, the use of machine learning for classification of signals is intriguing, the intricacy of the financial markets poses serious difficulties that call for cautious model tuning, risk management, and reasonable performance expectations. In order to adapt to shifting market conditions, future work should concentrate on improving feature engineering, investigating more complex algorithms, and adding adaptive mechanisms. The objective will always be to find strategies that can withstand the test of new and unseen data while striking a balance between predictive accuracy and the practical realities of trading.

VII. References

[1] Grigoryan, H. (2017). "Stock Market Trend Prediction Using Support Vector Machines and Variable Selection Methods." In Proceedings of the [International Conference on Applied Mathematics], (pp. 210), DOI: 10.2991/ammsa-17.2017.45.

[2] G. F. Bjoern Krollner, Bruce Vanstone, “Financial time series forecasting with machine learning techniques: A survey,” in European Symposium on Artificial Neural Networks: Computational and Machine Learning, Bruges, Belgium, April 2010. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] A. Kar, “Stock prediction using artificial neural networks,” Dept. of Computer Science and Engineering, IIT KanpurR. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] S. R. Y. Mayankkumar B Patel, “Stock prediction using artificial neural network,” International Journal of Innovative Research in Science, Engineering, and Technology, 2014.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[5] Taran Rishi,” Stock Market Analysis Using Linear Regression”, Proceedings of the Jepson Undergraduate Conference on International Economics, Vol. 4 [2022], Art. 4,Department of Economics,University of Northern Iowa. Pp. 2

[6] R. Karim, M. Alam, and R. Hossain, "Stock Market Analysis Using Linear Regression and Decision Tree Regression," in Proc. eSmarTA, 2021, pp. 1-6, doi: 10.1109/eSmarTA52612.2021.9515762.

[7] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM”, April 2017

[8] N. Tripathy, "Stock Price Prediction Using Support Vector Machine Approach," in [International Academic Conference on Management & Economics,Oxford,United Kingdom], [2019].

[9] Shah H.V., 2007.Machine Learning Techniques for Stock Prediction, Foundations of Machine Learning, Spring Soni S.and Shrivastava, S., 2010.Classification of Indian Stock Market Data Using Machine Learning Algorithms, International Journal on Computer Science and Engineering, 02(09):2942-2946

[10] Huang, W., Nakamori, Y. and Wang, S.Y. 2005. Forecasting Stock Market Movement Direction with Support Vector Machine, Computers and Operations Research, 32(10):2513–2522

[11] J. Fernando, “Moving Average (MA): purpose, uses, formula, and examples,” *Investopedia*, Mar. 31, 2023. https://www.investopedia.com/terms/m/movingaverage.asp

[12] C. Mitchell, “Triple Exponential Moving Average (TEMA): definition and formula,” *Investopedia*, Aug. 15, 2022. https://www.investopedia.com/terms/t/triple-exponential-moving-average.asp#:~:text=The%20TEMA%20reduces%20lag%20more,EMA%20of%20the%20original%20EMA.