"In Pursuit of Technical Excellence"

A Project Report

On

"STOCK MARKET PRICE PREDICTION"

In partial fulfillment of for the degree of Bachelor of Technology

In

Computer Science and Engineering

Submitted By

Pranay S. Ingle (BE20F05F023)

Prathamesh S. Jadhav (BE20F05F024)

Rutvik A. Jitakar (BE20F05F026)

Vedang R. Kawalkar (BE20F05F037)

Under the Guidance of

Prof. S. G. Shikalpure

Department of Computer Science and Engineering

Government College of Engineering, Aurangabad

(Chhatrapati Sambhajinagar)

(An autonomous Institute of Government of Maharashtra)

2023-24

CERTIFICATE

It is certified that work carried out in the eighth semester by Pranay Ingle, Prathamesh Jadhav, Rutvik Jitakar, Vedang Kawalkar in fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering from Government College of Engineering, Aurangabad (Chhatrapati Sambhajinagar) during the academic year 2023-2024 is satisfactory and good.

Place: Chhatrapati Sambhajinagar
Date:

Prof. S. G. Shikalpure

Project Guide	Head Of Department
(CSE Department)	(CSE Department)

Principal

Government College of Engineering, Aurangabad
(Chhatrapati Sambhajinagar)

DECLARATION

We Pranay Ingle, Prathamesh Jadhav, Rutvik Jitakar, Vedang Kawalkar student of "Bachelor of Technology in Computer Science and Engineering" declare that the Project Report entitled "Stock Market Price Prediction" is an authenticated work done by us.

We further declare that to the best of our knowledge the report does not contain any part of any work which has been submitted elsewhere by any other person in any other University/Institute/Organization for the award of any degree or diploma.

Place: Chhatrapati Sambhajinagar

Date: / /2024

Your Faithfully

Pranay Ingle- BE20F05F023
Prathamesh Jadhav- BE20F05F024
Rutvik Jitakar - BE20F05F026
Vedang Kawalkar - BE20F05F037

ACKNOWLEDGEMENT

This is to acknowledge and thank all the individuals who played a role in shaping this project report. Without their constant support, guidance and assistance, this project report would not have been completed. Without their coordination, guidance, and reviewing this task, it could not be completed alone. First, we are extremely grateful to our project guide, **Prof. S. G. Shikalpure**, for the helpful guidance, suggestions and inspirations given by him. Without them, it would have been impossible. We would like to express our sincere gratitude to **Prof. S. G. Shikalpure**, Head of the department, Computer Science And Engineering, for endless support and motivation. Finally, we would also like to express our gratitude to our honorable principal. **Dr. Sanjay Dambhare** whose encouragement and assistance especially in terms of resources has helped in speedy and quality completion of projects.

Pranay Ingle (BE20F05F023)

Prathamesh Jadhav (BE20F05F024)

Rutvik Jitakar (BE20F05F026)

Vedang Kawalkar (BE20F05F037)

ABSTRACT

Stock price prediction plays a pivotal role in financial decision-making, aiding investors, traders, and financial institutions in managing risks and maximizing returns. In this project, we explore the effectiveness of machine learning algorithms for stock price prediction, focusing on Multi-Layer Perceptron (MLP) classification, Extra Trees Classifier, and Gradient Boosting Classifier. Our objective is to evaluate the performance of these algorithms across various metrics and datasets, providing insights into their strengths and limitations. The project begins with the acquisition and preprocessing of historical stock price data, followed by feature engineering to extract relevant information. We then develop and train machine learning models using the selected algorithms, optimizing hyperparameters to enhance predictive accuracy. Model evaluation is conducted using appropriate metrics, comparing performance across different algorithms and datasets. Our findings reveal nuanced differences in algorithm performance, with each algorithm exhibiting distinct strengths. The MLP classifier demonstrates flexibility and adaptability in capturing complex patterns, while the Extra Trees Classifier excels in handling high-dimensional data and mitigating overfitting. The Gradient Boosting Classifier showcases strong predictive performance and interpretability, leveraging boosting techniques to refine predictions.

LIST OF ABBREVIATIONS

ML: Machine learning

MLP: Multi Layer Perception

GBC: Gradient Boosting Classifier

ETC: Extra Trees Classifier

API: Application Programming Interface

ROC-AUC: Receiver Operating Characteristic Area Under the Curve

ARIMA: Autoregressive integrated moving average

SVM: Support Vector Machines

RNNs: Recurrent neural networks

LSTM: Long short-term memory

k-NN: k-Nearest Neighbors

EMA: Exponential Moving Average

MA: Moving Average

MAE: Mean Absolute Error

RMSE: Root Mean Squared Error

INDEX

Chapter 1

1. INTRODUCTION

In the dynamic landscape of financial markets, the ability to forecast stock prices accurately holds immense significance for investors, traders, and financial institutions alike. With the advent of machine learning (ML) algorithms and the availability of vast amounts of financial data, the quest for developing robust predictive models has intensified.This report presents an exploration into the realm of stock price prediction using multiple ML algorithms. Leveraging historical stock market data, we aim to develop models capable of forecasting future price movements with a high degree of accuracy. By employing various ML techniques, including regression, ensemble methods, and deep learning, we endeavor to uncover patterns and relationships within the data that can inform predictive insights.The advent of artificial intelligence and machine learning has revolutionized various industries, including finance. One of the prominent applications of machine learning in finance is stock price prediction, a task that has garnered significant attention due to its potential for financial gains and risk management. In this project, we delve into the realm of stock price prediction by employing multiple machine learning algorithms to forecast the future movements of stock prices.

The primary objective of this project is to explore the efficacy of different machine learning algorithms in predicting stock prices. Specifically, we focus on three popular algorithms: Multi-Layer Perceptron (MLP) classification, Extra Trees Classifier, and Gradient Boosting Classifier. These algorithms were chosen based on their proven track record in handling complex datasets and their potential to capture intricate patterns in stock price movements.Stock price prediction is inherently challenging due to the presence of various factors influencing market dynamics, including economic indicators, geopolitical events, and investor sentiment. Therefore, our project aims to assess the performance of the selected algorithms in capturing the inherent complexities of stock price data and generating accurate predictions.By testing these algorithms on historical stock price data, we aim to evaluate their predictive capabilities and identify the algorithm(s) that demonstrate superior performance in terms of accuracy, robustness, and scalability. Additionally, we seek to provide insights into the strengths and limitations of each algorithm, thus offering valuable guidance for practitioners and researchers alike in the field of financial forecasting.

1.1 Problem Statement

Stock price prediction is a crucial task in the field of finance, as it enables investors to make informed decisions, mitigate risks, and maximize returns on their investments. However, accurately forecasting the future movements of stock prices remains a formidable challenge due to the inherent complexity and volatility of financial markets. The primary problem addressed in this project is to assess the effectiveness of various machine learning algorithms in predicting stock prices. Despite the availability of numerous prediction models and methodologies, identifying the most suitable algorithm(s) for accurate and reliable stock price forecasting remains an ongoing concern.

Key challenges include:

- Complexity of Market Dynamics: Financial markets are influenced by a multitude of factors, including economic indicators, company performance, industry trends, geopolitical events, and investor sentiment. Capturing and interpreting these complex dynamics to make accurate predictions poses a significant challenge.
- Data Noise and Uncertainty: Stock price data is inherently noisy and subject to fluctuations caused by both predictable and unpredictable factors. Distinguishing between genuine patterns and random fluctuations is essential for building robust prediction models.
- Algorithm Selection and Performance Evaluation: With a plethora of machine learning algorithms available, selecting the most appropriate ones for stock price prediction requires careful consideration. Furthermore, accurately evaluating the performance of these algorithms in terms of prediction accuracy, stability, and scalability is crucial for informed decision-making.
- Overfitting and Generalization: Given the high dimensionality of stock price data and the potential for overfitting, ensuring that prediction models generalize well to unseen data is essential. Balancing model complexity with generalization capability is a critical aspect of building robust and reliable prediction models.

Addressing these challenges requires a systematic approach that involves rigorous experimentation, evaluation, and comparison of different machine learning algorithms. By tackling these issues, this project aims to support the creation of efficient and practical solutions for stock price prediction, thus benefiting investors, financial institutions, and researchers in the field of finance.

1.2 Goals and Objectives:

In the realm of finance, the ability to predict stock prices accurately is paramount for investors, traders, and financial institutions seeking to maximize returns and mitigate risks. Machine learning algorithms offer promising avenues for stock price prediction by leveraging historical data and complex patterns within financial markets. The goals and objectives of this project revolve around harnessing the power of machine learning to develop robust and accurate stock price prediction models.

Goals:

- Evaluate Algorithm Performance:

The primary goal of this project is to assess the performance of selected machine learning algorithms, including Multi-Layer Perceptron (MLP) classification, Extra Trees Classifier, and Gradient Boosting Classifier, in predicting stock prices. By rigorously testing these algorithms on historical stock price data, we aim to identify the most effective algorithm(s) for accurate and reliable prediction.

- Enhance Prediction Accuracy:

Another goal of this project is to enhance the accuracy of stock price prediction models. Through feature engineering, algorithm optimization, and model tuning, we seek to improve prediction accuracy and reduce prediction errors, thereby providing investors with more reliable forecasts of future stock price movements.

- Explore Model Interpretability:

In addition to accuracy, we aim to enhance the interpretability of prediction models. By analysing feature importance, understanding model decision boundaries, and providing intuitive explanations for predictions, we aim to make our models more transparent and accessible to stakeholders, fostering trust and understanding in the prediction process.

- Enable Real-World Application:

A crucial goal of this project is to develop prediction models that can be deployed in real-world financial applications. By ensuring scalability, robustness, and efficiency, we aim to create models that are suitable for deployment in trading platforms, investment tools, and financial decision support systems, enabling practitioners to leverage the power of machine learning for better decision-making.

Objectives:

- Data Acquisition and Preprocessing:

The first objective of the project is to acquire and preprocess historical stock price data from reliable sources. This involves collecting data for relevant financial instruments, cleaning the data to remove noise and inconsistencies, and transforming it into a format suitable for analysis and modelling.

- Feature Engineering:

The project aims to identify and engineer relevant features that may influence stock price movements. This involves extracting meaningful information from raw data, including historical prices, trading volumes, technical indicators, and external factors such as economic indicators and news sentiment, to enhance the predictive capability of the models.

- Model Development and Training:

The project seeks to develop and train machine learning models using the pre-processed data. This involves implementing selected algorithms, optimizing model hyperparameters, and training the models on historical data to learn patterns and relationships between input features and stock price movements.

- Model Evaluation and Comparison:

An important objective of the project is to evaluate and compare the performance of different machine learning algorithms. This involves assessing prediction accuracy, robustness, and scalability using appropriate evaluation metrics and statistical tests, enabling us to identify the algorithm(s) that demonstrate superior performance for stock price prediction.

- Analysis and Interpretation:

The project aims to analyse and interpret the results of model evaluations, providing insights into the strengths and weaknesses of each algorithm. This involves examining feature importance, understanding model decision-making processes, and explaining predictions in intuitive terms, enabling stakeholders to gain valuable insights from the prediction models.

- Documentation and Reporting:

The final objective of the project is to document and report the findings in a comprehensive project report. This involves documenting the methodology, experimental setup, results, analysis, and conclusions, providing a valuable resource for stakeholders interested in leveraging machine learning for stock price prediction.

1.3 Project Scope

The scope of this project encompasses the following aspects:

- Algorithm Testing: At first, algorithms need to be tested for the most effective accurate results. So, multiple algorithms were tested like gaussian naïve bayes, MLP classifier, K Neighbors Classifier, Logistic regression, Linear Discriminant Analysis, Decision Tree Classifier, Decision Tree regression, Random Forest Classifier, Gradient Boosting Classifier (GBC), Extra Trees Classifier (ETC) and Ada Boost Classifier.

- Algorithm Selection: The project focuses on evaluating the performance of three specific machine learning algorithms for stock price prediction: Multi-Layer Perceptron (MLP) classification, Extra Trees Classifier, and Gradient Boosting Classifier. These algorithms were chosen based on their potential to handle the complexities of stock price data and their relevance in the context of financial forecasting.
- Data Collection and Preprocessing: Historical stock price data from relevant financial markets will be collected from reliable sources such as financial databases or APIs. The data will undergo preprocessing steps, including cleaning, normalization, feature selection, and splitting into training and testing sets.
- Feature Engineering: Relevant features that may influence stock price movements, such as historical prices, trading volumes, technical indicators, and external factors (e.g., economic indicators, news sentiment), will be identified and engineered to enhance the predictive capability of the models.
- Model Development and Training: Each selected algorithm will be implemented and trained using the preprocessed data. Hyperparameter tuning may be performed to optimize the performance of the models. Cross-validation techniques will be employed to assess the generalization ability of the models and mitigate overfitting.
- Evaluation Metrics: The performance of each model will be evaluated using appropriate metrics such as accuracy, precision, recall, F1-score, and ROC-AUC (Receiver Operating Characteristic Area Under the Curve). Additionally, visualizations such as confusion matrices and ROC curves may be utilized to provide deeper insights into model performance.
- Comparison and Analysis: A comparative analysis will be conducted to assess the strengths and weaknesses of each algorithm in terms of prediction accuracy, robustness, and computational efficiency. Insights gained from the analysis will be used to identify the algorithm(s) that demonstrate superior performance for stock price prediction.

- Limitations and Future Directions: The project will acknowledge any limitations encountered during the experimentation process, such as data availability constraints, model assumptions, and performance trade-offs. Additionally, potential avenues for future research and improvements will be discussed to guide further exploration in the field of stock price prediction.
- Documentation and Reporting: The findings of the project will be documented in a comprehensive project report, detailing the methodology, experimental setup, results, analysis, and conclusions. The report will serve as a valuable resource for stakeholders interested in leveraging machine learning for stock price prediction.

The project scope is limited to evaluating the performance of the specified machine learning algorithms on historical stock price data. It does not encompass real-time prediction or the integration of external data sources beyond the scope of traditional financial indicators.

Chapter 2

2. literature survey

Stock price prediction has long been a topic of interest in the field of finance, with researchers and practitioners continuously seeking more accurate and reliable forecasting methods. Machine learning algorithms have emerged as powerful tools for stock price prediction, leveraging historical data and complex patterns within financial markets. In this literature survey, we explore key studies and methodologies related to stock price prediction using machine learning algorithms, focusing on Multi-Layer Perceptron (MLP) classification, Extra Trees Classifier, and Gradient Boosting Classifier.

•**Traditional Approaches:**

Traditional approaches to stock price prediction often relied on statistical models such as autoregressive integrated moving average (ARIMA), linear regression, and time series analysis. While these methods provided valuable insights into historical trends, they often struggled to capture the non-linear and dynamic nature of financial markets.

•**Machine Learning Techniques:**

Machine learning techniques have gained traction in recent years due to their ability to handle complex datasets and extract meaningful patterns. Various studies have explored the application of machine learning algorithms for stock price prediction, including but not limited to:

•**Support Vector Machines (SVM):**

SVM has been widely used for binary classification tasks in stock price prediction, with studies demonstrating its effectiveness in capturing market trends and patterns.

•**Random Forest:**

Random Forest algorithms have been employed for stock price prediction, leveraging ensemble learning to improve prediction accuracy and robustness.

•Deep Learning Models:

Deep learning architectures, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, have shown promise in capturing temporal dependencies and complex relationships in stock price data.

Multi-Layer Perceptron (MLP) Classification:

MLP classification, a type of artificial neural network, has garnered attention for its ability to model non-linear relationships and handle complex datasets. Studies have demonstrated the effectiveness of MLP classifiers in predicting stock prices, particularly when dealing with high-dimensional data and intricate patterns.

•Extra Trees Classifier:

The Extra Trees Classifier, an extension of Random Forest, has been explored for stock price prediction due to its robustness to noise and ability to handle large feature spaces. Research has highlighted its efficacy in mitigating overfitting and improving prediction accuracy, particularly in noisy and high-dimensional datasets.

•Gradient Boosting Classifier:

Gradient Boosting algorithms, such as Gradient Boosting Classifier and XGBoost, have gained popularity for their superior predictive performance and interpretability. Studies have showcased their effectiveness in capturing both linear and non-linear relationships in stock price data, making them suitable choices for financial forecasting tasks.

•Challenges and Considerations:

Despite the advancements in machine learning techniques for stock price prediction, several challenges persist. These include data quality issues, model interpretability, overfitting, and the presence of market noise. Researchers continue to explore novel methodologies and techniques to address these challenges and improve prediction accuracy.

•**Future Directions:**

Future research directions in stock price prediction encompass the integration of alternative data sources, such as sentiment analysis, news articles, and social media data, to enhance prediction accuracy and robustness. Additionally, efforts to improve model interpretability, scalability, and real-time deployment capabilities will be crucial for advancing the practical applications of stock price prediction in real-world scenarios.

Chapter 3

3.Methodology

Our methodology combines machine learning methods like MLP classification, Gradient Boosting Classifier, and Extra Tree Classifier to construct our predictive model. Alongside these, we employ time series forecasting techniques such as Prophet and Autoregressive Integrated Moving Average (ARIMA), along with pattern recognition approaches. This comprehensive blend provides a detailed strategy for stock market analysis and prediction. Each step in our methodology is carefully designed and executed to ensure precision, reliability, and accuracy in forecasting market trends. By integrating these diverse techniques, we aim to capture the complexity of stock market behavior and deliver robust predictions that can assist investors and analysts in making informed decisions.

3.1Gathering and Preprocessing

Our process starts with gathering historical stock market data from reputable sources, ensuring a comprehensive view of market behavior over time. Next, we meticulously preprocess the data to remove anomalies like missing values or outliers, maintaining its quality and consistency. This cleansing step is crucial for the accuracy and reliability of our subsequent analysis.

Following preprocessing, we focus on feature extraction, identifying and selecting key variables from the raw dataset that influence stock market movements. These features could range from stock market indicators to economic factors. By extracting meaningful features, we aim to enhance the predictive power of our models and capture underlying trends effectively.

In essence, our approach combines thorough data collection, rigorous preprocessing, and strategic feature extraction to ensure accurate and reliable stock market analysis and prediction. Each step is designed to uphold the integrity of the data and improve the effectiveness of our predictive models.

3.2Creating model with ML Algos:

To develop our predictive model for stock price forecasting, we employed a combination of machine learning algorithms, specifically tailored to handle the complexities of financial markets.
The algorithms are :

Gaussian Naïve Bayes

Gaussian Naive Bayes (GNB) is a classification algorithm that applies Bayes' Theorem assuming independence among predictors (features) within each class. It's particularly useful for datasets with continuous features, assuming that each class is drawn from a Gaussian distribution. At its core, GNB relies on Bayes' Theorem, used to compute the probability of a hypothesis given the observed data.. In the context of classification, it predicts the probability of each class label given the input features and selects the label with the highest probability. The "naive" assumption in GNB is that the presence of a particular feature in a class is independent of the presence of any other feature. This simplifies the computation of probabilities and makes the algorithm computationally efficient. In summary, Gaussian Naive Bayes is a probabilistic classification algorithm that assumes feature independence and Gaussian distributions within classes. It calculates the conditional probability of each class given the input features and selects the class with the highest probability as the prediction.

MLP Classifier

The MLP (Multilayer Perceptron) Classifier is an artificial neural network designed for classification tasks, featuring multiple layers of interconnected nodes (perceptrons), organized in a feedforward manner. Each node in one layer is connected to every node in the subsequent layer, forming a network of interconnected nodes. The MLP Classifier typically comprises an input layer, one or more hidden layers, and an output layer. Each node in the hidden layers and the output layer employs an activation function to introduce non-linearity into the model, enabling it to learn complex patterns in the data.
During training, the MLP Classifier adjusts the weights of the connections between nodes through a process called backpropagation, where errors are propagated backward through the network and used to update the weights, aiming to minimize the difference between the predicted and actual outputs.
The flexibility of the MLP Classifier allows it to learn intricate relationships between features in the data, making it suitable for a wide range of classification tasks. However, training an MLP Classifier can be computationally intensive and may require careful tuning of hyperparameters to achieve optimal performance.

Logistic Regression

Logistic Regression is a statistical method used for binary classification tasks, where the target variable has only two possible outcomes. Although named 'regression', logistic regression is actually a classification algorithm. It works by modeling the probability that a given input belongs to a specific class using a logistic function. This function maps input features to a probability range of 0 to 1, indicating the likelihood of the input belonging to one of two classes..In logistic regression, the relationship between the input features and the output (probability of belonging to a class) is modeled using a linear equation, which is then transformed by the logistic function to constrain the output between 0 and 1. This transformation allows logistic regression to provide meaningful class probabilities.During training, the model's parameters (coefficients) are iteratively adjusted using optimization techniques like gradient descent to minimize the disparity between the predicted probabilities and the true class labels. Logistic regression is widely used due to its simplicity, interpretability, and efficiency, especially when dealing with linearly separable data or when the goal is to understand the impact of individual features on the probability of a particular outcome. However, it's important to note that logistic regression assumes a linear relationship between the features and the log odds of the outcome.

K-Nearest Neighbors

The k-Nearest Neighbors (k-NN) classifier is a simple yet effective algorithm used for both classification and regression tasks. It works by storing all available cases and classifying new cases based on a similarity measure (such as distance functions). In classification, the output is a class membership; for regression, it's the value of the object's attribute. The k-NN algorithm assigns a class to a new data point based on the majority class among its k nearest neighbors, where k is a user-defined parameter. This algorithm is intuitive, versatile, and easy to understand, making it a popular choice for many machine learning tasks. However, its performance can degrade with high-dimensional data or when dealing with imbalanced class distributions.

Decision tree Regression

Decision tree regression is a machine learning technique employed for predictive modeling in regression scenarios, where the goal is to predict continuous numerical outcomes. It works by recursively partitioning the input space into regions, each associated with a simple prediction model, typically the mean or median of the target variable within that region. The decision tree learns these partitions by selecting features that best split the data based on a criterion such as minimizing variance or maximizing information gain. During prediction, new data points traverse the tree, and their predicted values are obtained by averaging the target variable values within the corresponding leaf node. Decision tree regression is easy to interpret and can capture non-linear relationships in the data but may suffer from overfitting and instability with noisy data.

Random Forest Classifier

Random Forest Classifier is an ensemble learning method used for classification tasks. It operates by constructing a multitude of decision trees during training and outputs the mode (or mean) prediction of the individual trees for classification. Each tree in the forest is built using a random subset of the training data and a random subset of features, which enhances the model's robustness against overfitting and increases accuracy. When making predictions, each tree "votes" for the class label, and the class with the most votes across all trees is chosen as the final prediction. Random forests are known for their high accuracy, versatility, and ability to handle large datasets with high dimensionality, making them one of the most popular and powerful machine learning algorithms.

Gradient Boosting Classifier

Gradient Boosting Classifier is a powerful machine learning technique used for classification tasks. It works by sequentially adding weak learners (typically decision trees) to improve the model's predictive accuracy. During training, the algorithm focuses on instances where the preceding models have made errors, adjusting the subsequent models to correct these mistakes. It optimizes a loss function by minimizing the errors of the ensemble of weak learners. Gradient Boosting Classifier is highly effective, often outperforming other algorithms due to its ability to capture complex patterns in the data. However, it can be computationally expensive and sensitive to hyperparameters, requiring careful tuning for optimal performance.

AdaBoost Classifier

AdaBoost (Adaptive Boosting) Classifier is an ensemble learning method primarily used for classification tasks. It combines multiple weak learners, typically decision trees, to create a strong classifier. In each iteration, AdaBoost assigns weights to incorrectly classified data points, focusing on those instances in subsequent iterations to improve classification accuracy. It sequentially adjusts the weights of the weak learners based on their performance, allowing the model to learn from its mistakes. The final prediction is made by combining the predictions of all weak learners, weighted by their individual performance. AdaBoost is known for its ability to handle complex datasets and its resistance to overfitting. However, it can be sensitive to noisy data and outliers.

Data Collection: We sourcedto serve as the foundational dataset for our analysis. This data provides a comprehensive view of past market behavior, allowing our models to learn and identify patterns that influence stock price movements.

```
aapl_df.head()
```

Date	Open	High	Low	Close	Adj Close	Volume
2014-04-21	18.762142	19.004999	18.712856	18.970358	16.709381	182548800
2014-04-22	18.868214	18.993929	18.803572	18.989286	16.726055	202563200
2014-04-23	18.895000	18.968929	18.730356	18.741072	16.507425	394940000
2014-04-24	20.293215	20.357143	20.026072	20.277500	17.860731	759911600
2014-04-25	20.161785	20.428213	20.141430	20.426430	17.991917	390275200

Fig: Historical stock market data for Apple

```
msft_df.head()
```

Date	Open	High	Low	Close	Adj Close	Volume
2014-04-21	40.130001	40.150002	39.790001	39.939999	33.821377	22221200
2014-04-22	39.959999	40.139999	39.830002	39.990002	33.863712	27056700
2014-04-23	39.990002	39.990002	39.470001	39.689999	33.609688	24602800
2014-04-24	39.740002	39.970001	39.299999	39.860001	33.753639	42381600
2014-04-25	40.290001	40.680000	39.750000	39.910000	33.795979	56876800

Fig: Historical stock market data for Microsoft

Feature Engineering with Exponential Moving Average (EMA): We utilized Exponential Moving Average (EMA) calculations over different time periods (7, 14, and 28 days) to generate buy or sell signals for the stocks. EMA is a widely recognized technical indicator that helps in smoothing out price data to identify trends more effectively.

Formula: EMA=(Close−PreviousEMA)∗(2/(TimePeriod+1))+PreviousEMA

Target Variable Generation: To facilitate supervised learning, we created a target variable based on the actual stock returns. This variable indicates whether the actual returns are positive or negative over a specific time horizon. It serves as the ground truth for our models, enabling them to learn and make predictions based on historical performance.

```
aapl_technical_df
```

Date	Close	Actual Returns	7 Day EMA	14 Day EMA	28 Day EMA	RSI	MACD	MACD Signal	MACD Hist	Signals	Strategy Returns
2018-02-21	42.767502	-0.004539	42.264376	41.903443	42.543848	55.558223	-0.414255	-1.036335	0.622080	0.0	NaN
2018-02-22	43.125000	0.008359	42.479532	42.066318	42.583928	57.541513	-0.294401	-0.887948	0.593547	0.0	0.000000
2018-02-23	43.875000	0.017391	42.828399	42.307475	42.672967	61.430292	-0.137314	-0.737822	0.600507	1.0	0.000000
2018-02-26	44.742500	0.019772	43.306924	42.632145	42.815694	65.380033	0.056527	-0.578952	0.635479	1.0	0.019772
2018-02-27	44.597500	-0.003241	43.629568	42.894193	42.938577	64.196667	0.196185	-0.423924	0.620110	1.0	-0.003241
...	...	...	...	...	...	...	...	...	...	...	...
2024-04-12	176.550003	0.008627	172.118190	171.664141	173.150757	56.816148	-1.242213	-2.167750	0.925537	0.0	0.000000
2024-04-15	172.690002	-0.021863	172.261143	171.800923	173.118980	49.461093	-1.065775	-1.947355	0.881580	0.0	-0.000000
2024-04-16	169.380005	-0.019167	171.540859	171.478134	172.861120	44.179555	-1.179440	-1.793772	0.614332	0.0	-0.000000
2024-04-17	168.000000	-0.008147	170.655644	171.014383	172.525870	42.158316	-1.365138	-1.708045	0.342907	0.0	-0.000000
2024-04-18	167.039993	-0.005714	169.751731	170.484464	172.147534	40.761234	-1.571653	-1.680767	0.109113	0.0	-0.000000

Fig: Moving Averages And Signals of apple dataset

```
msft_technical_df
```

Date	Close	Actual Returns	7 Day EMA	14 Day EMA	28 Day EMA	RSI	MACD	MACD Signal	MACD Hist	Signals	Strategy Returns
2018-02-21	91.489998	-0.013266	91.358962	90.962512	90.646949	53.988129	0.789720	0.866126	-0.076406	1.0	NaN
2018-02-22	91.730003	0.002623	91.451722	91.064844	90.721642	54.581313	0.757299	0.844361	-0.087062	1.0	0.002623
2018-02-23	94.059998	0.025401	92.103791	91.464198	90.951874	59.975918	0.909136	0.857316	0.051820	1.0	0.025401
2018-02-26	95.419998	0.014459	92.932843	91.991638	91.260020	62.756553	1.126226	0.911098	0.215129	1.0	0.014459
2018-02-27	94.199997	-0.012786	93.249631	92.286086	91.462777	58.809475	1.186155	0.966109	0.220046	1.0	-0.012786
...	...	...	...	...	...	...	...	...	...	...	...
2024-04-12	421.899994	-0.014091	423.930927	422.885304	419.634356	52.186186	3.093923	3.555917	-0.461995	1.0	-0.014091
2024-04-15	413.640015	-0.019578	421.358199	421.652598	419.220953	44.220612	2.098447	3.264423	-1.165976	-1.0	-0.019578
2024-04-16	414.579987	0.002272	419.663646	420.709584	418.900886	45.244864	1.369586	2.885456	-1.515869	-1.0	-0.002272
2024-04-17	411.839996	-0.006609	417.707733	419.526972	418.413928	42.778927	0.564359	2.421236	-1.856878	0.0	0.006609
2024-04-18	404.269989	-0.018381	414.348297	417.492708	417.438484	36.809832	-0.676824	1.801624	-2.478448	0.0	-0.000000

Fig: Moving Averages And Signals of microsoft dataset

Date	RSI	MACD Signal	MACD
2018-02-22	0.196989	-0.674474	-0.377640
2018-02-23	0.501671	-0.605999	-0.310496
2018-02-26	0.811130	-0.533535	-0.227643
2018-02-27	0.718414	-0.462825	-0.167949
2018-02-28	0.673184	-0.396969	-0.124438
...	...	...	...
2024-04-12	0.140157	-1.258215	-0.782761
2024-04-15	-0.436105	-1.157689	-0.707347
2024-04-16	-0.849909	-1.087637	-0.755930
2024-04-17	-1.008271	-1.048535	-0.835303
2024-04-18	-1.117731	-1.036093	-0.923573

Date	RSI	MACD Signal	MACD
2018-02-22	0.196989	-0.674474	-0.377640
2018-02-23	0.501671	-0.605999	-0.310496
2018-02-26	0.811130	-0.533535	-0.227643
2018-02-27	0.718414	-0.462825	-0.167949
2018-02-28	0.673184	-0.396969	-0.124438

Date	RSI	MACD Signal	MACD
2018-08-22	1.389894	0.363877	0.389499
2018-08-23	1.430870	0.371807	0.378857
2018-08-24	1.494763	0.376057	0.369044
2018-08-27	1.657747	0.379544	0.369453
2018-08-28	1.804777	0.384093	0.377696

Fig: Moving Averages

In our analysis, we initially applied a Moving Average (MA) technique to both the Apple testing and training datasets to identify underlying trends. Following this, we developed a versatile function capable of executing multiple machine learning models and assessing their performance to identify the most suitable ones for our data.

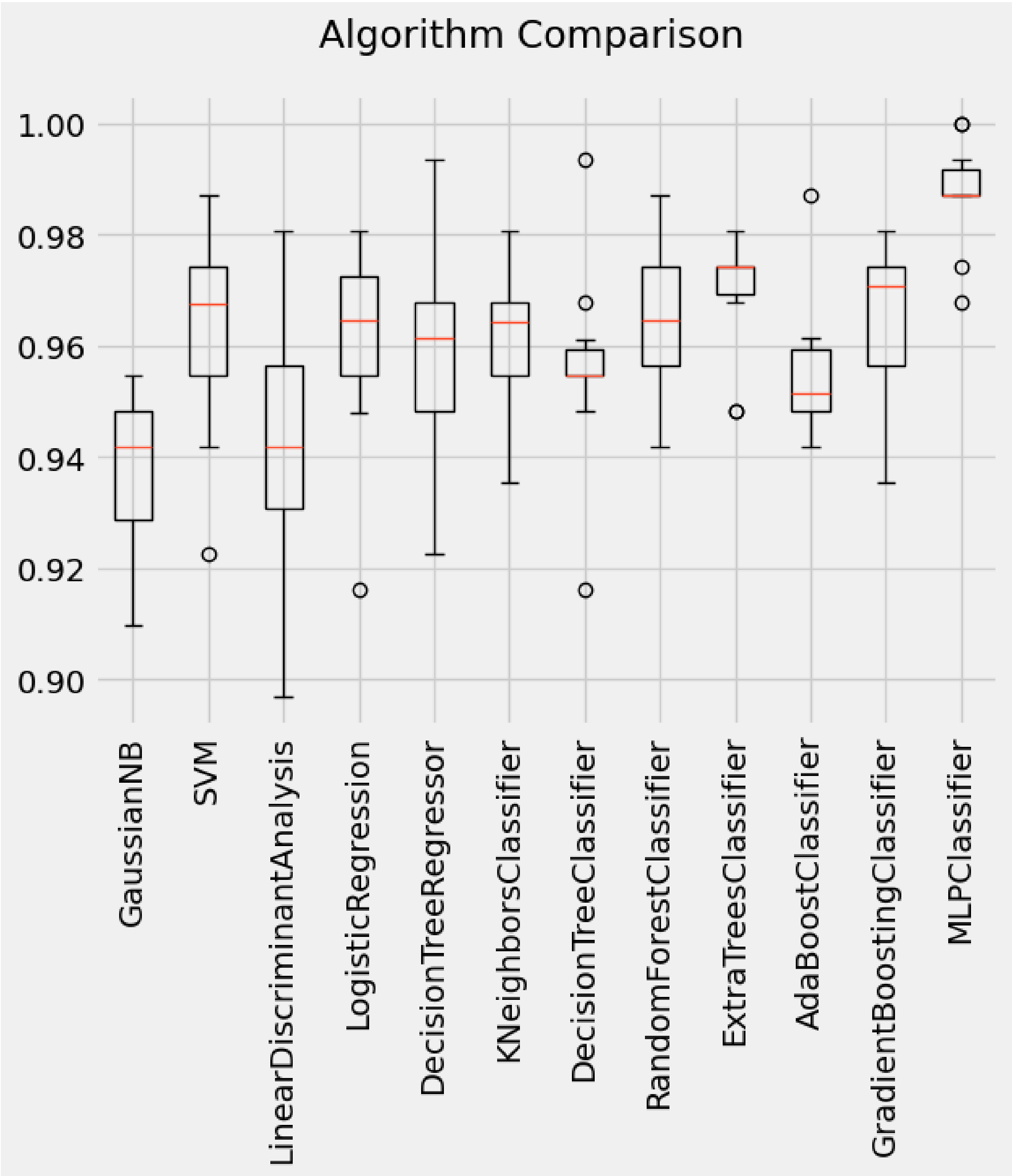

Fig: ML Algorithms Comparison

Upon evaluating various algorithms, we observed that the MLP Classifier, Gradient Boost Classifier, and Extra Tree Classifier demonstrated the highest accuracy levels. These algorithms were particularly effective in capturing the nuances of the data and producing reliable predictions.

Now we apply these algorithms on historical stock market data for Apple and Microsoft.

Model A: Applying Multilayer percepton (MLP) Classifier on apple dataset

	precision	recall	f1-score	support
-1.0	0.31	0.19	0.24	57
0.0	0.93	0.94	0.94	524
1.0	0.94	0.95	0.95	842
accuracy			0.92	1423
macro avg	0.73	0.70	0.71	1423
weighted avg	0.91	0.92	0.91	1423

Fig: MLP Classifier accurecy after applying on apple dataset

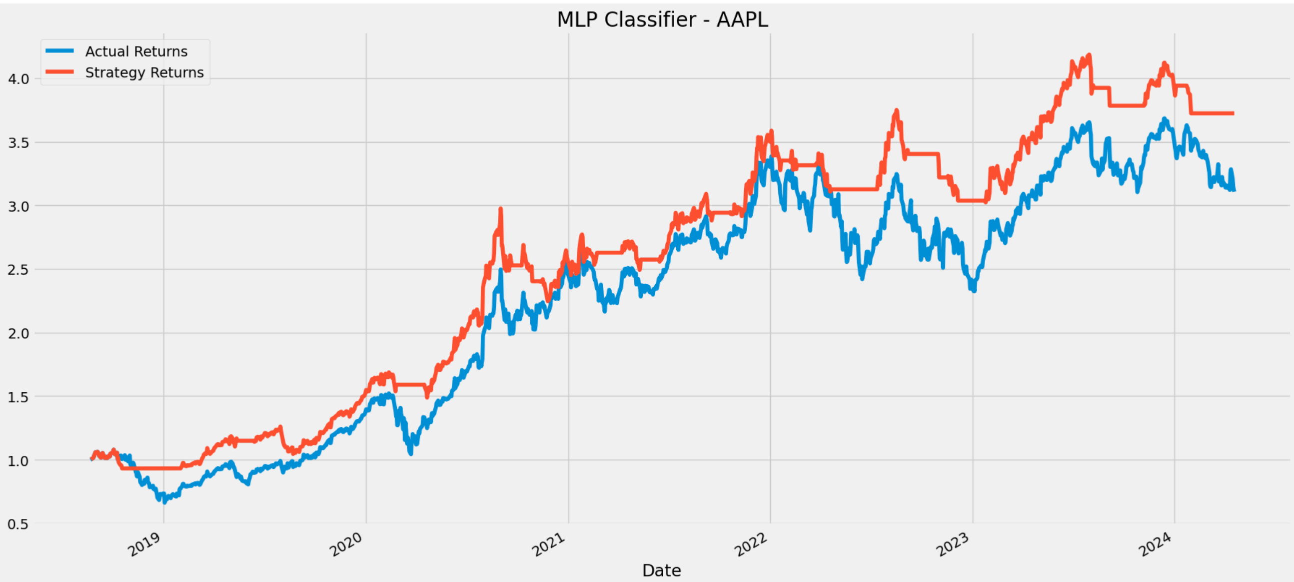

Fig: Actual vs Strategy Returns by MLP Classifier

Model B: Applying Extra trees Classifier on apple dataset

	precision	recall	f1-score	support
-1.0	0.42	0.09	0.14	57
0.0	0.95	0.97	0.96	524
1.0	0.93	0.97	0.95	842
accuracy			0.93	1423
macro avg	0.76	0.67	0.68	1423
weighted avg	0.92	0.93	0.92	1423

Fig: Extra trees Classifier accurecy after applying on apple dataset

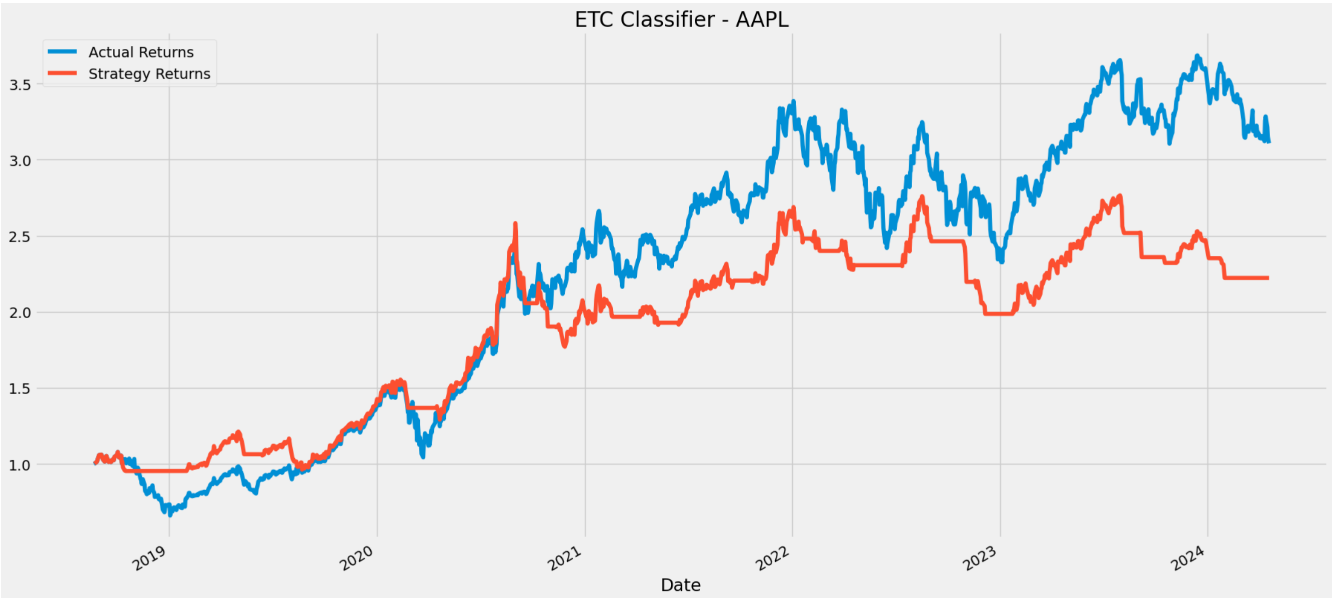

Fig: Actual vs Strategy Returns by Extra tree Classifier

Model C: Applying Gredient Boosting Classifier on apple dataset

	precision	recall	f1-score	support
-1.0	0.56	0.33	0.42	57
0.0	0.96	0.96	0.96	524
1.0	0.94	0.97	0.96	842
accuracy			0.94	1423
macro avg	0.82	0.75	0.78	1423
weighted avg	0.93	0.94	0.94	1423

Fig: Gredient Boosting Classifier accurecy after applying on apple dataset

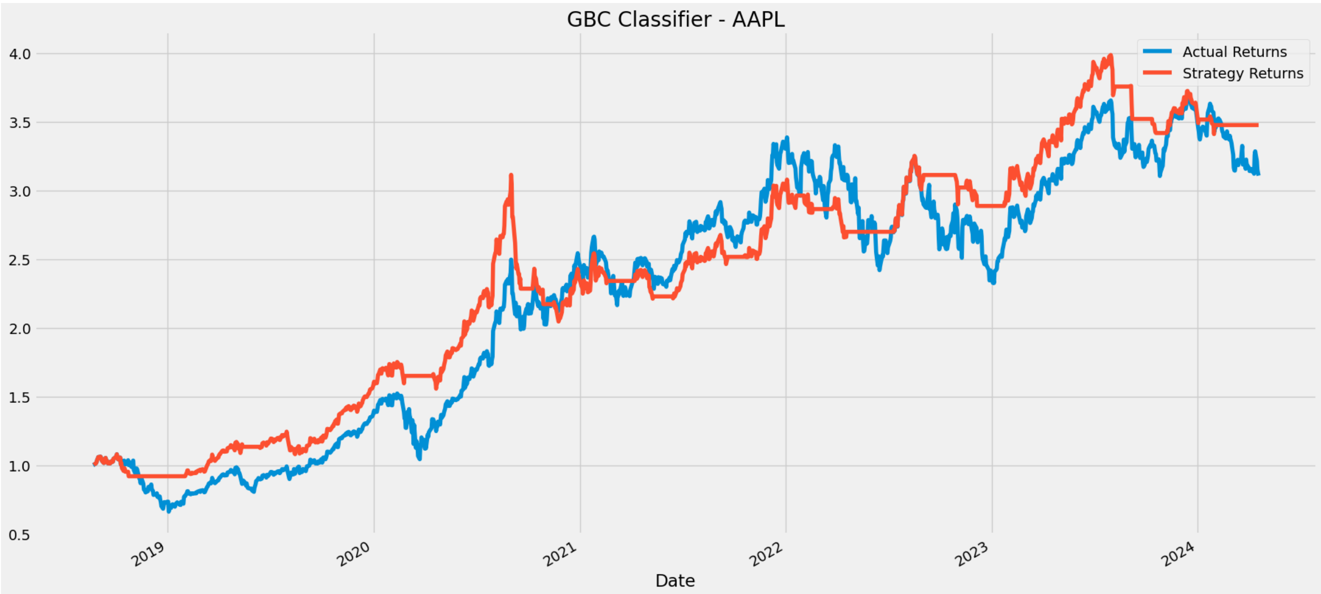

Fig: Actual vs Strategy Returns by Gredient Boosting Classifier

Model A: Applying Multilayer percepton (MLP) Classifier on microsoft dataset

	precision	recall	f1-score	support
-1.0	0.66	0.30	0.42	109
0.0	0.75	0.97	0.84	441
1.0	0.95	0.88	0.91	999
accuracy			0.86	1549
macro avg	0.78	0.72	0.72	1549
weighted avg	0.87	0.86	0.86	1549

Fig: MLP Classifier accurecy after applying on microsoft dataset

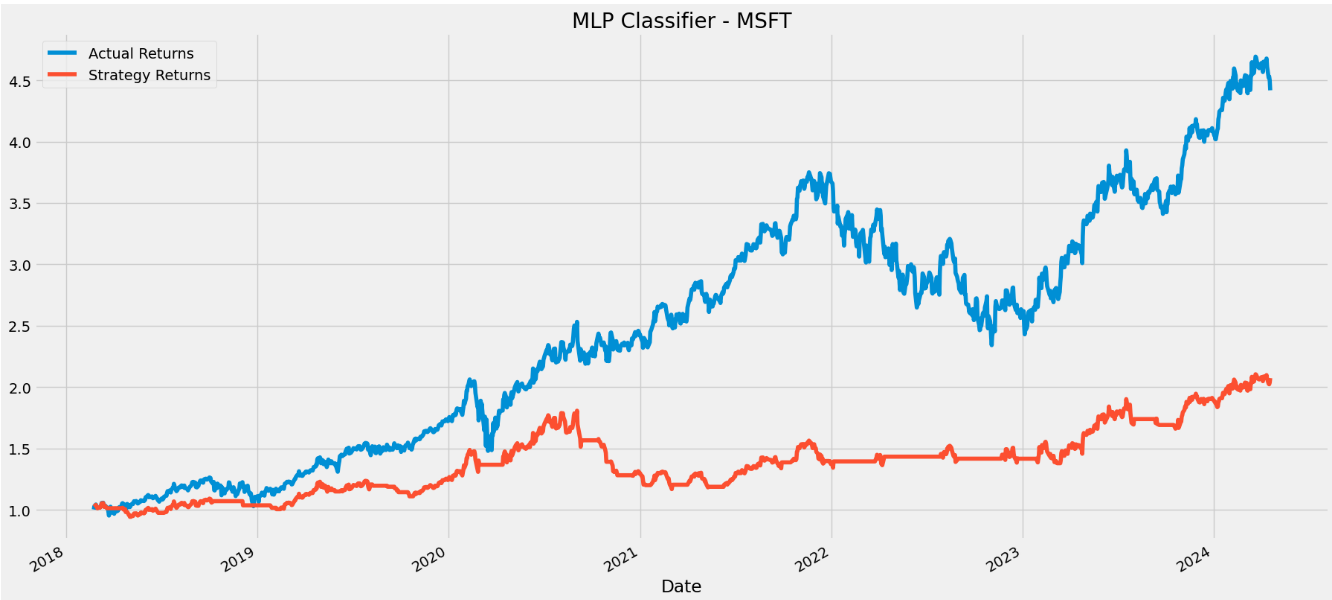

Fig: Actual vs Strategy Returns by MLP Classifier

Model B: Applying Extra trees Classifier on microsoft dataset

	precision	recall	f1-score	support
-1.0	0.65	0.10	0.17	109
0.0	0.76	1.00	0.87	441
1.0	0.93	0.89	0.91	999
accuracy			0.86	1549
macro avg	0.78	0.66	0.65	1549
weighted avg	0.86	0.86	0.84	1549

Fig: Extra trees Classifier accurecy after applying on microsoft dataset

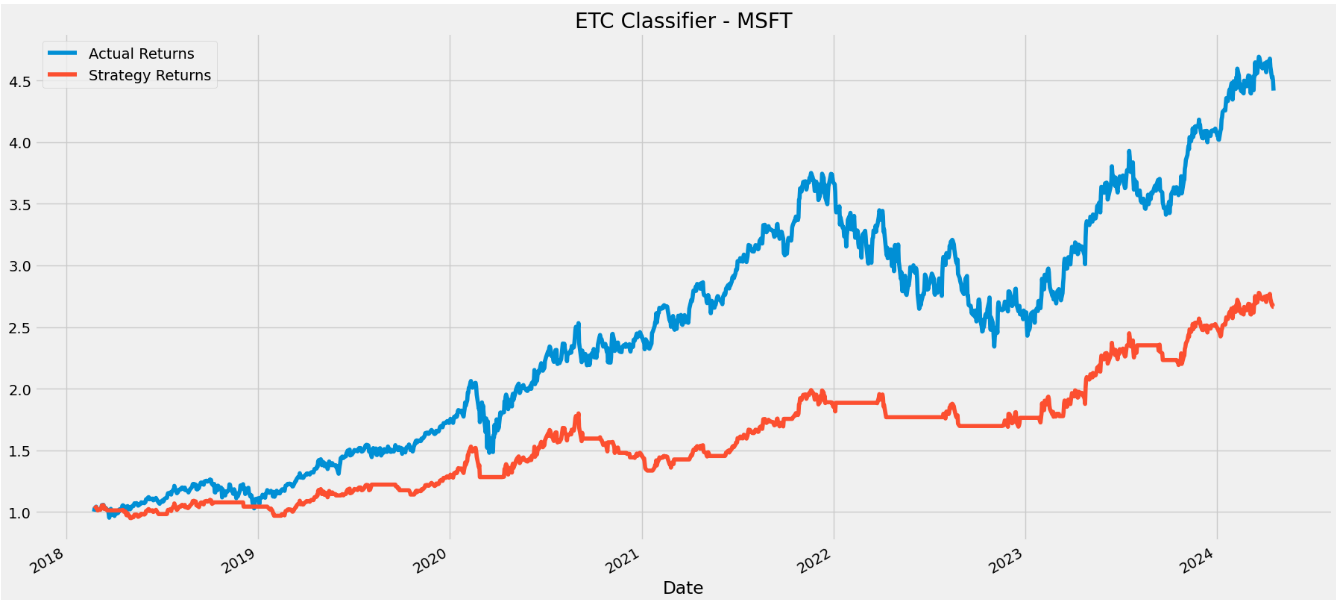

Fig: Actual vs Strategy Returns by Extra tree Classifier

Model C: Applying Gredient Boosting Classifier on microsoft dataset

	precision	recall	f1-score	support
-1.0	0.66	0.36	0.46	109
0.0	0.79	1.00	0.88	441
1.0	0.95	0.89	0.92	999
accuracy			0.88	1549
macro avg	0.80	0.75	0.75	1549
weighted avg	0.88	0.88	0.87	1549

Fig: Gredient Boosting Classifier accurecy after applying on microsoft dataset

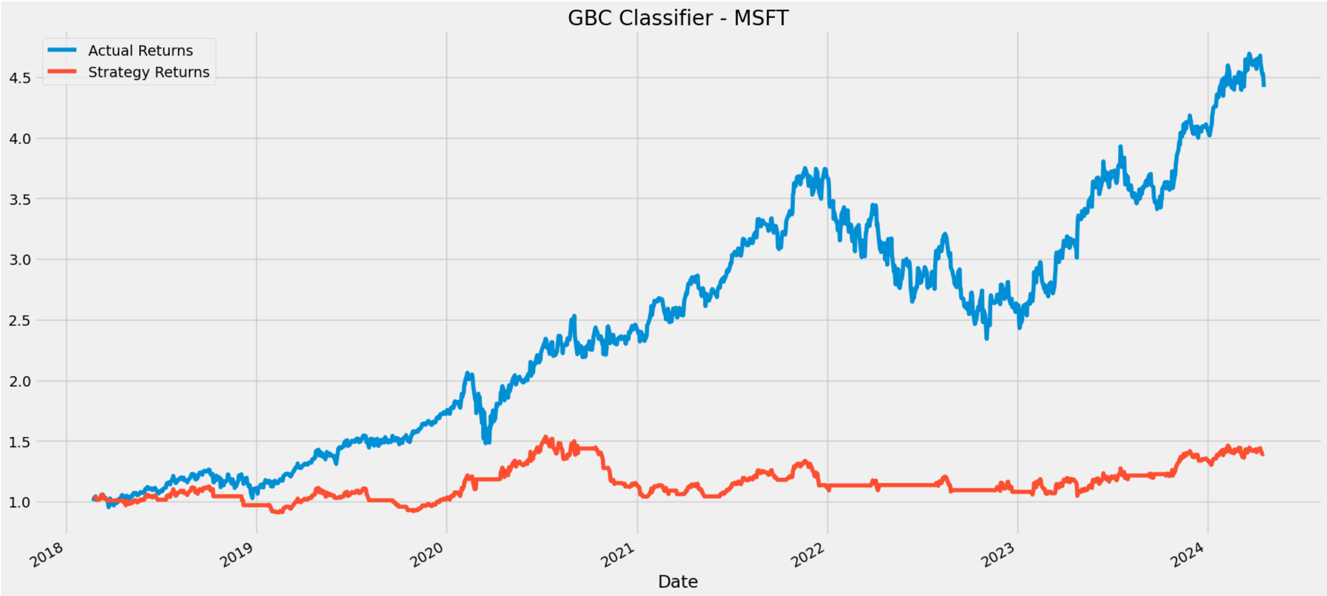

Fig: Actual vs Strategy Returns by Gredient Boosting Classifier

Findings: The algorithms yielded more accurate results when applied to the Apple dataset. The predicted returns generated by our model closely aligned with the actual returns for Apple, indicating the model's robustness and accuracy in predicting stock trends for this specific dataset.

.

Table:

	Apple Dataset	Microsoft Dataset
MLP Classifier	92%	86%
Extra trees Classifier	93%	86%
Gredient Boosting Classifier	94%	88%

Based on these findings, we finalized our model selection for stock market prediction. To further assess the model's applicability and robustness across different stocks, we tested it using the stock price dataset of "Head and Shoulders" stocks. This additional testing aimed to evaluate the model's performance on diverse datasets and ascertain its potential for broader applications in stock market forecasting.

```
[*********************100%%**********************]  1 of 1 completed
Date
2014-04-21   18.762142   19.004999   18.712856   18.970358   16.709377
2014-04-22   18.868214   18.993929   18.803572   18.989286   16.726051
2014-04-23   18.895000   18.968929   18.730356   18.741072   16.507420
2014-04-24   20.293215   20.357143   20.026072   20.277500   17.860731
2014-04-25   20.161785   20.428213   20.141430   20.426430   17.991915
...                ...         ...         ...         ...         ...
2024-04-11  168.339996  175.460007  168.160004  175.039993  175.039993
2024-04-12  174.259995  178.360001  174.210007  176.550003  176.550003
2024-04-15  175.360001  176.630005  172.500000  172.690002  172.690002
2024-04-16  171.750000  173.759995  168.270004  169.380005  169.380005
2024-04-17  169.610001  170.649994  168.000000  168.000000  168.000000

               Volume  high_roll_max  low_roll_min      head_shoulder_pattern
Date
2014-04-21  182548800            NaN           NaN                        NaN
2014-04-22  202563200            NaN           NaN                        NaN
2014-04-23  394940000      19.004999     18.712856                        NaN
2014-04-24  759911600      20.357143     18.730356                        NaN
2014-04-25  390275200      20.428213     18.730356                        NaN
...               ...            ...           ...                        ...
2024-04-11   91070300     175.460007    167.110001                        NaN
2024-04-12  101593300     178.360001    167.110001  Inverse Head and Shoulder
2024-04-15   73531800     178.360001    168.160004                        NaN
2024-04-16   73711200     178.360001    168.270004                        NaN
2024-04-17   50901200     176.630005    168.000000                        NaN
```

Fig: Head and Shoulder historical stock market dataset

After applying our finalized prediction model on head on shoulder dataset we got the following results:

Date	Open	High	Low	Close	Adj Close	Volume	high_roll_max	low_roll_min	head_shoulder_pattern
2014-05-09	20.876429	20.937500	20.726070	20.912144	18.522619	291597600	21.331785	20.726070	Head and Shoulder
2014-06-13	23.049999	23.110001	22.719999	22.820000	20.212465	218100000	23.690001	22.719999	Head and Shoulder
2014-06-18	23.067499	23.072500	22.837500	23.045000	20.411764	134056000	23.187500	22.837500	Head and Shoulder
2014-07-10	23.440001	23.887501	23.379999	23.760000	21.045067	158744000	24.200001	23.379999	Head and Shoulder
2014-07-18	23.405001	23.684999	23.254999	23.607500	20.909986	199952000	24.275000	23.142500	Head and Shoulder

In summary, our approach involved utilizing Moving Average techniques, developing a multifunctional model assessment tool, and selecting the most accurate algorithms based on their performance. The chosen model demonstrated high accuracy in predicting stock returns for Apple and exhibited promising results when tested on other stock datasets. This comprehensive testing and validation process enabled us to confidently select our model for stock market prediction, paving the way for its potential application in predicting trends across various stock markets.

3.3 Modeling with ARIMA:

ARIMA (Autoregressive Integrated Moving Average) is a widely used time series forecasting method that models the relationship between a variable and its past values, trends, and seasonality. It consists of three components: autoregression (AR), differencing (I), and moving average (MA). ARIMA models capture linear dependencies within a time series by incorporating lagged observations, differencing to stabilize non-stationary data, and smoothing out noise with moving average terms. This model is effective for predicting future values based on historical patterns and is commonly used in finance, economics, and other fields requiring time-dependent analysis.

We utilize the ARIMA model to analyze time series data, specifically tailored to identify both seasonality and linear trends present in the stock market. The model's key parameters (p, d, q) are carefully determined through either statistical methods or grid search techniques to ensure optimal performance. Once the model is trained on our meticulously cleaned dataset, we subject it to rigorous validation processes. Techniques such as cross-validation and out-of-sample testing are employed to assess the model's accuracy and reliability in forecasting.

After successfully validating the ARIMA model, we leverage its capabilities to predict future stock values. By extracting and utilizing patterns from historical data, the model generates forecasts that aim to capture the complex dynamics of the market. This comprehensive approach, combining precise parameter optimization with thorough validation, ensures that our forecasting tool is robust and effective. Ultimately, our goal is to provide accurate and reliable predictions that assist investors and analysts in making informed decisions in the ever-changing stock market landscape.

3.4Predictive forecasting:

The Prophet forecasting model is set up to identify seasonality and account for holidays from the outset. To streamline the model fitting and evaluation process, we partition the preprocessed data into separate training and validation sets. Once trained on the training data, the Prophet model adaptively learns and adjusts to recognize patterns and seasonality within the dataset. To gauge the model's accuracy and reliability, we compute performance metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) using the validation set.

After evaluating the Prophet model's performance, we leverage its learned patterns and trends to generate future stock value projections. By harnessing the insights from the trained Prophet model, we aim to provide forecasts that capture the inherent complexities and nuances of the stock market. This approach ensures that our forecasting tool is not only adaptive and accurate but also capable of generating reliable predictions to assist investors and analysts in navigating the dynamic landscape of the stock market effectively.

3.5 Identification of Patterns:

Pattern recognition algorithms play a pivotal role in uncovering recurring patterns and trends within historical stock price data. These algorithms systematically analyze the dataset to identify common patterns such as triangles, head and shoulders, and double tops. Recognizing these patterns is vital as they often serve as indicators for potential future market movements.

By identifying and understanding these patterns, traders and investors can gain valuable insights into the market's behavior. This knowledge enables them to make well-informed decisions when developing or adjusting their trading and investment strategies. Utilizing pattern recognition algorithms as part of our methodology enhances our ability to anticipate market trends and movements, thereby empowering individuals to navigate the stock market with greater confidence and precision.

3.6 Assessment of Performance:

An essential part of our methodology is assessing performance by computing various indicators to measure the effectiveness of our integrated approach. We conduct a comparative analysis between the combined methodology and the standalone performance of individual models, such as Prophet and ARIMA (Arimax). Additionally, we perform sensitivity analysis to evaluate the integrated framework's robustness against variations in data and input parameters. This evaluation process ensures that our integrated approach not only enhances predictive accuracy but also demonstrates resilience and reliability across different scenarios, validating its effectiveness in forecasting stock market trends.

3.7 Deciphering and Making Decisions:

The integrated platform equips traders, investors, and financial analysts with valuable insights to navigate the intricacies of the stock market. By analyzing forecasted trends and patterns, stakeholders gain the knowledge needed for informed decision-making. This empowers them to effectively manage risks and fine-tune investment portfolios, maximizing potential returns while minimizing uncertainties.

Chapter 4

4.Screenshots and Project Flow:

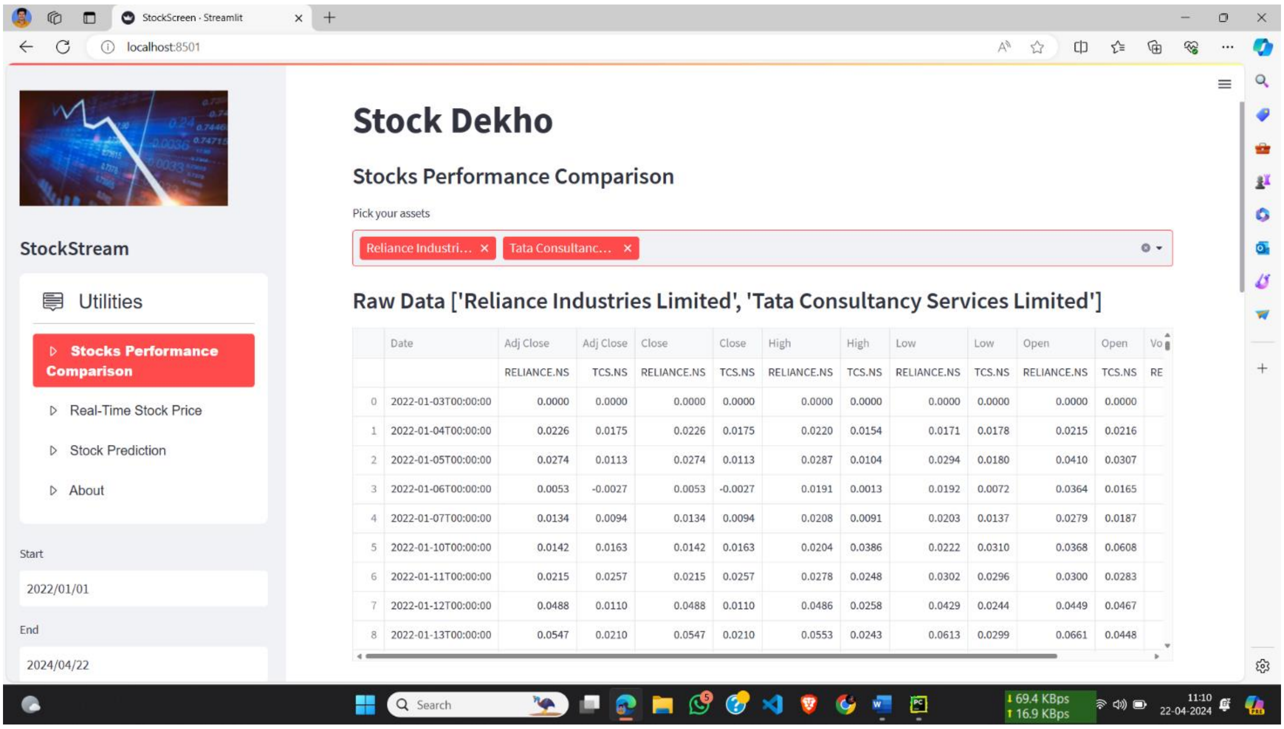

Fig: Stock Performance Comparison 1

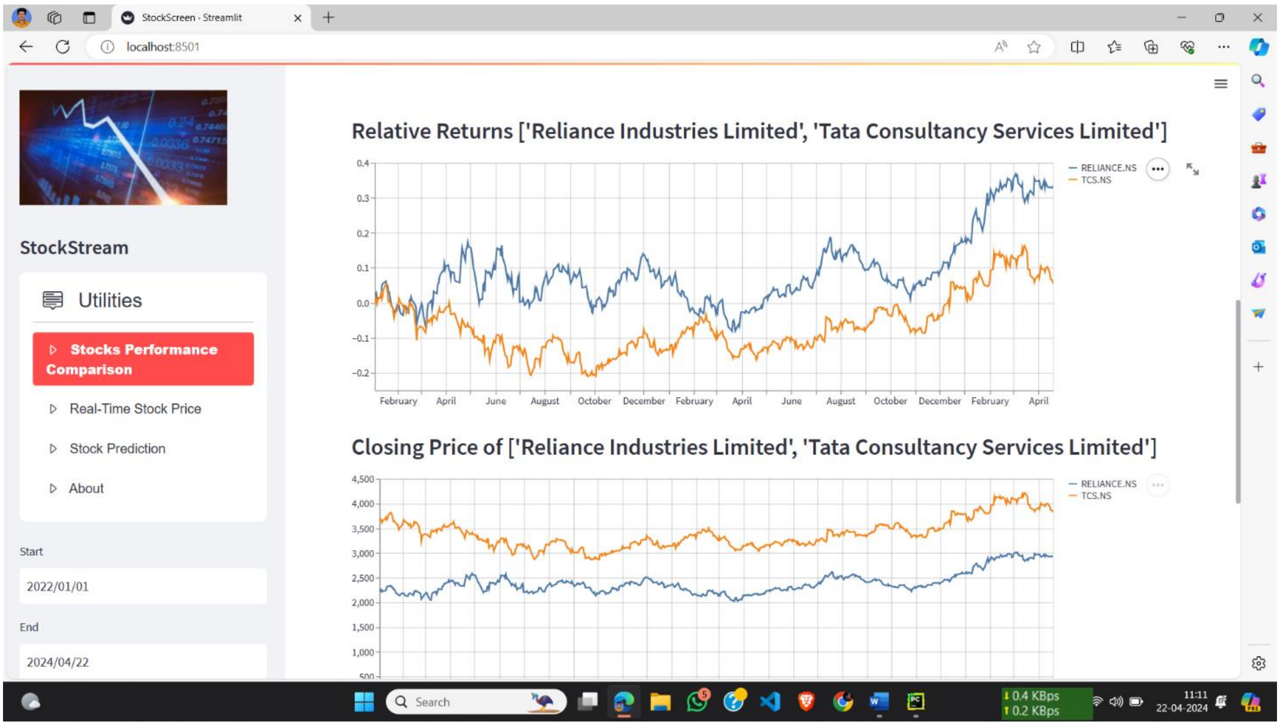

Fig: Stock Performance Comparison 2

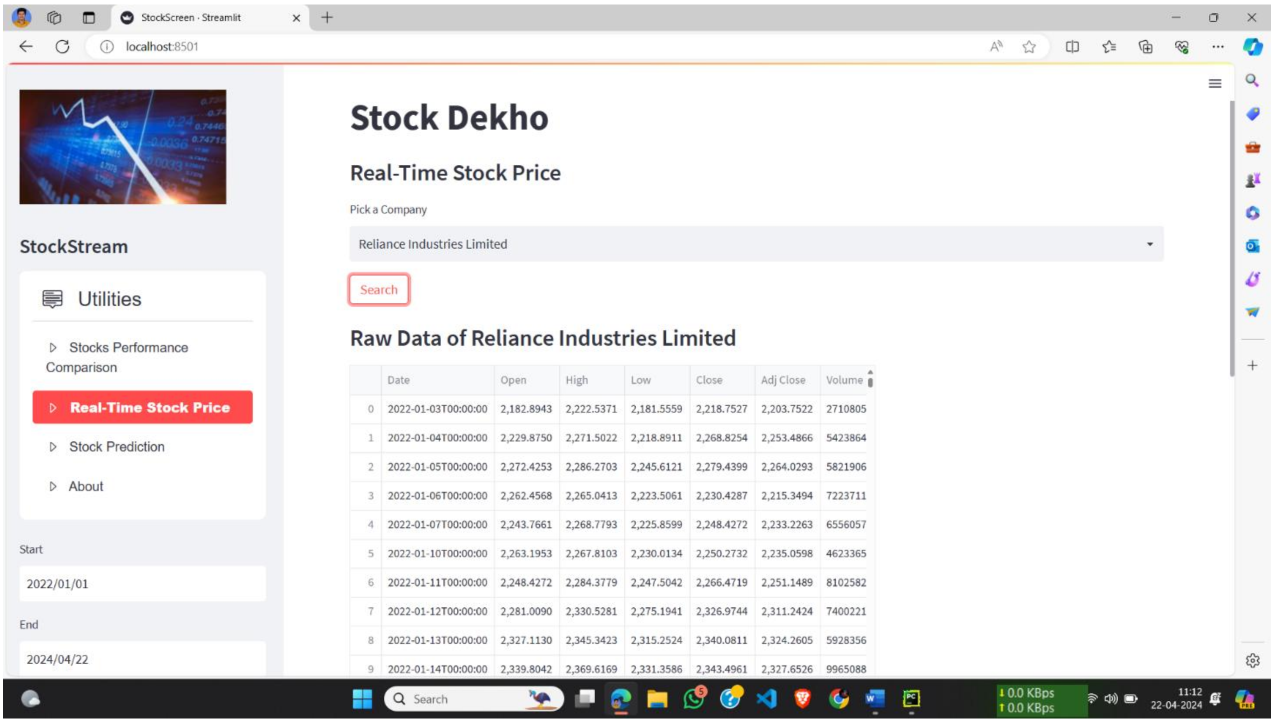

Fig: Real-Time Stock Prize (1)

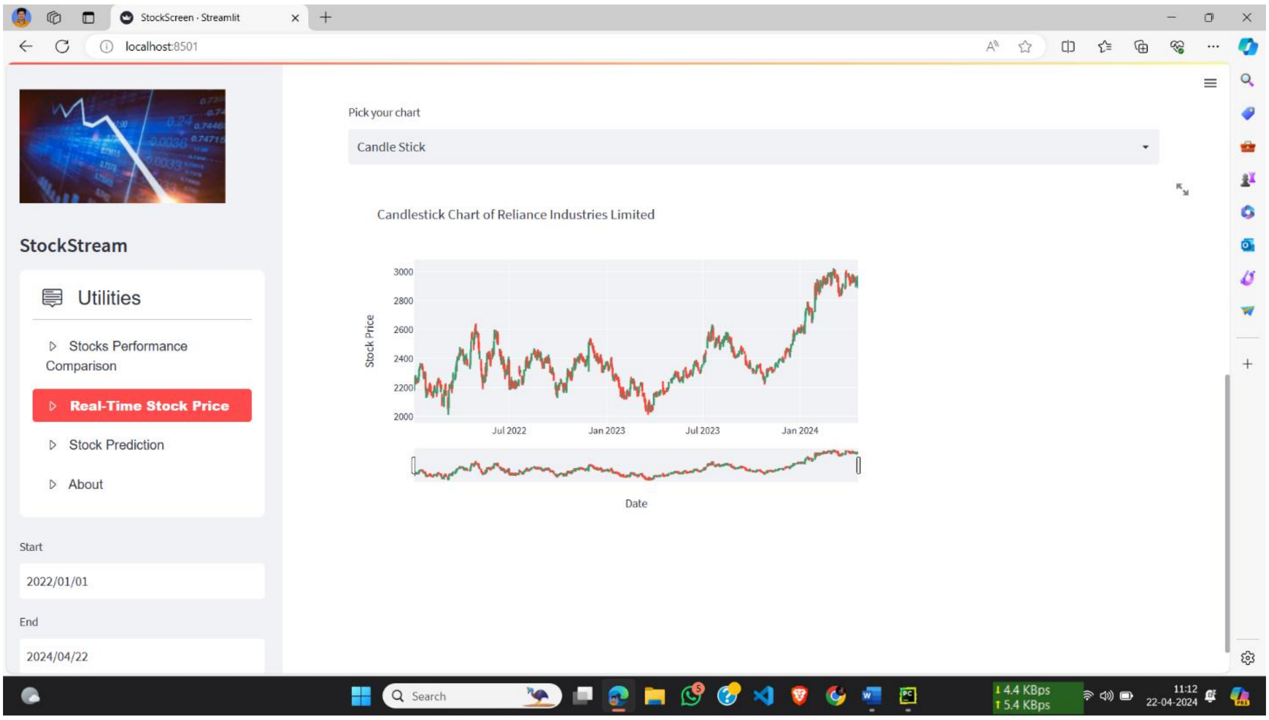

Fig: Real-Time Stock Prize (2)

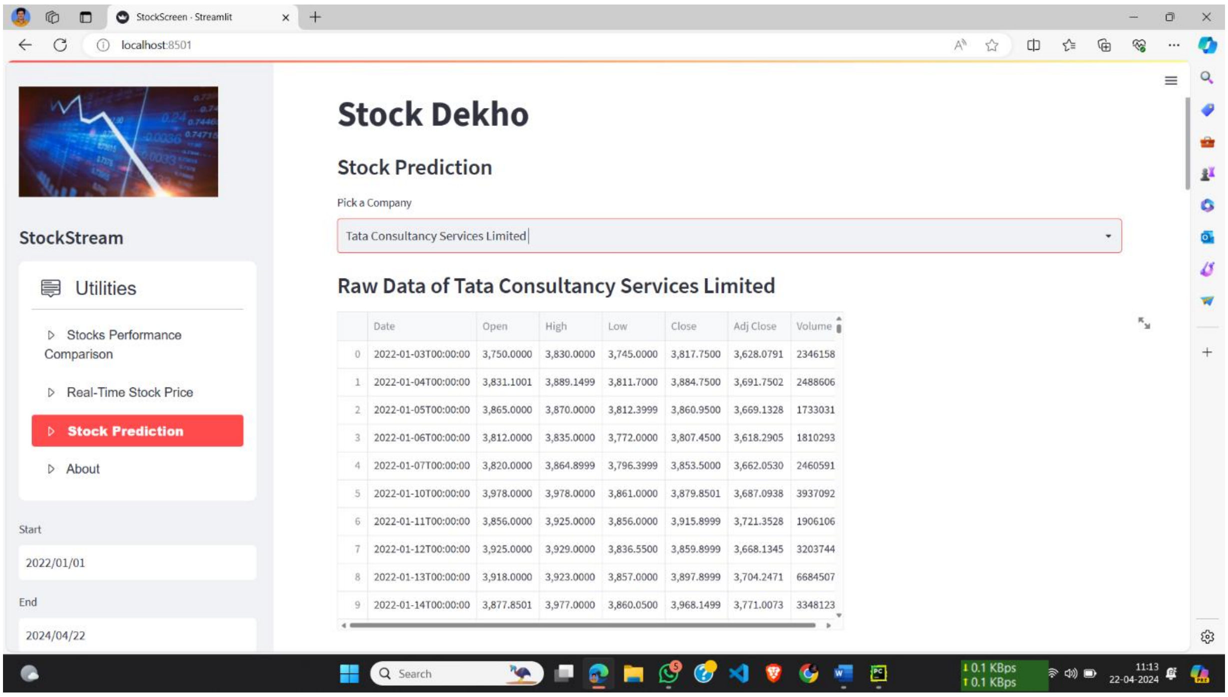

Fig: Stock Prediction (1)

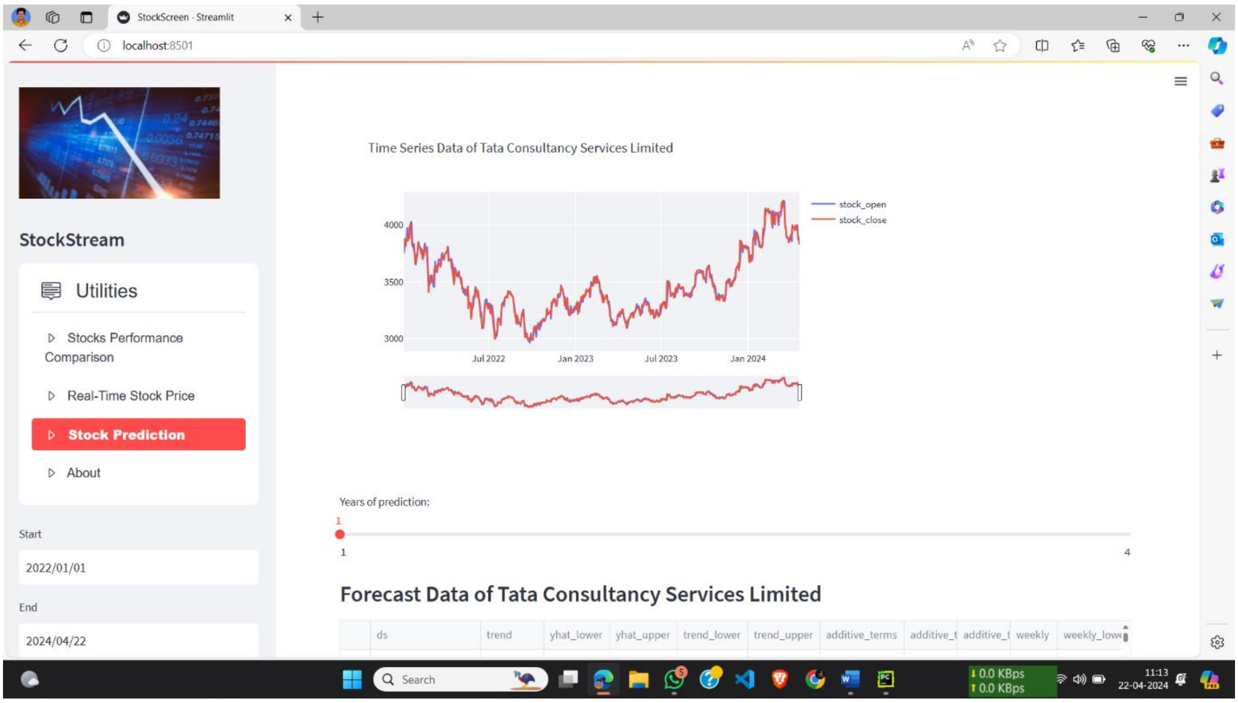

Fig: Stock Prediction (2)

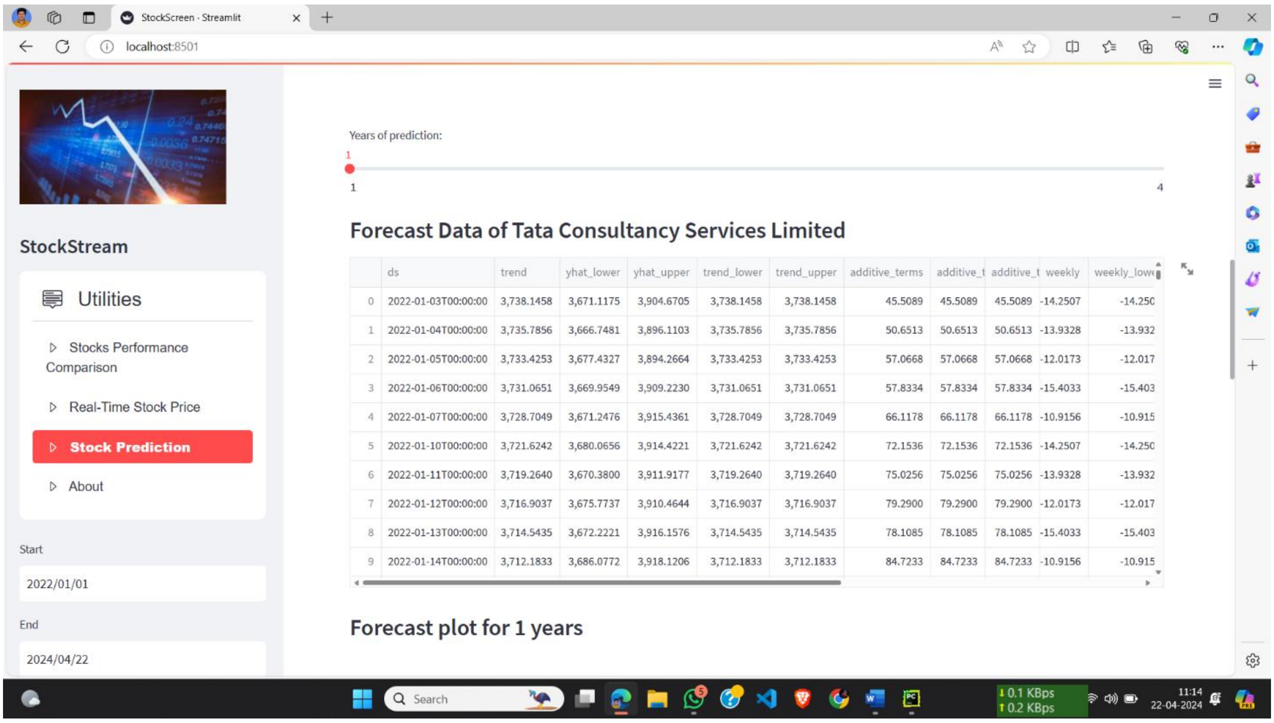

Fig: Stock Prediction (3)

Fig: Stock Prediction (4)

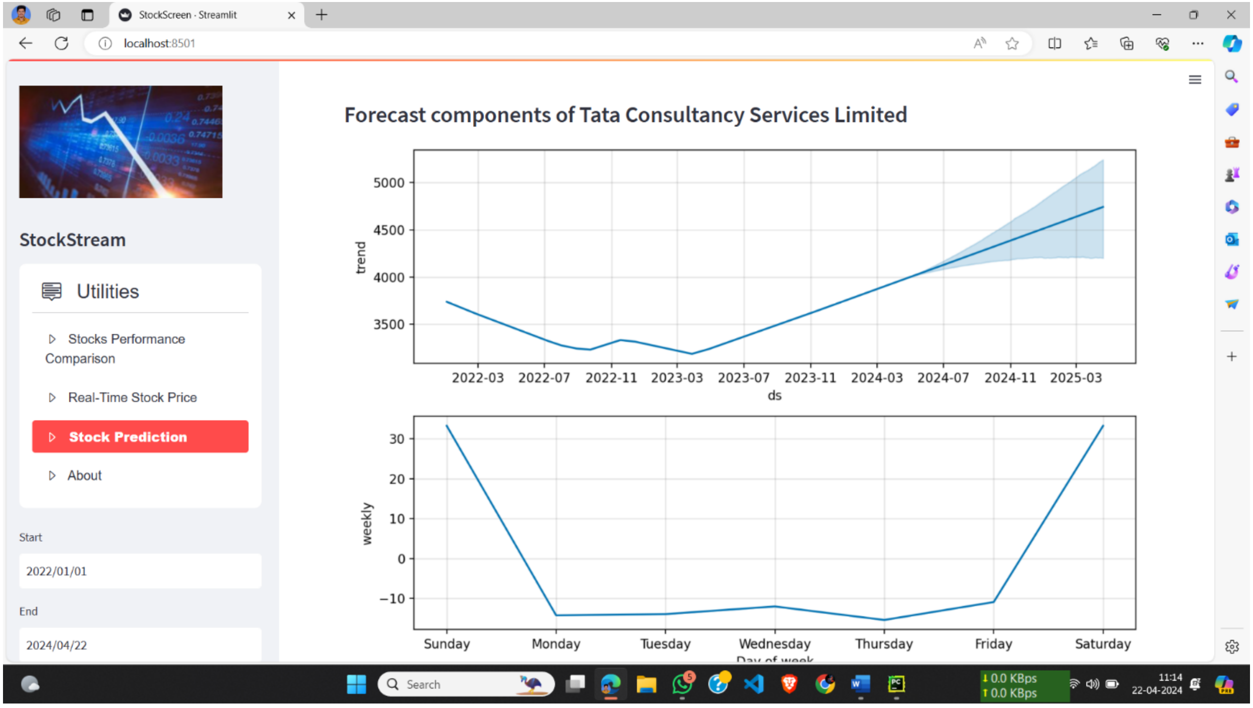

Fig: Stock Prediction (5)

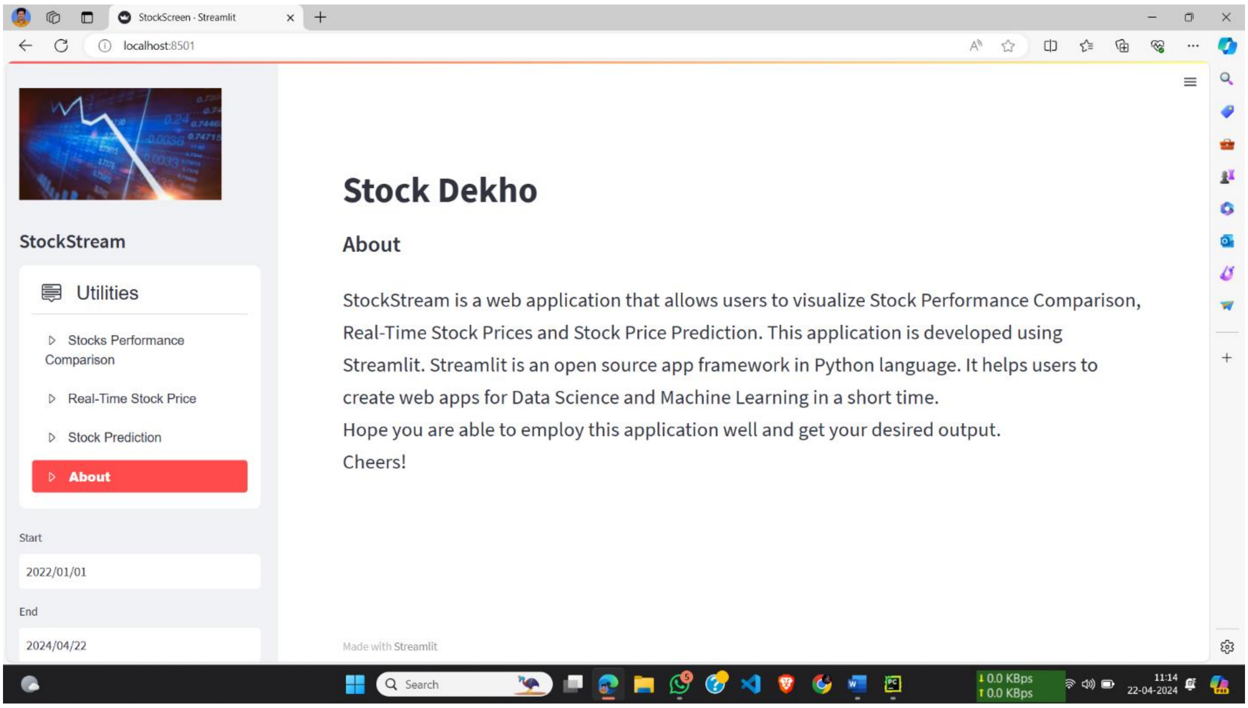

Fig: About

Chapter 5

5.1 FUTURE SCOPE

While this project focuses on evaluating the performance of selected machine learning algorithms for stock price prediction, there are several avenues for further exploration and enhancement. The future scope of this project includes:

Integration of Alternative Data Sources:

Incorporating alternative data sources such as social media sentiment, news articles, satellite imagery, and macroeconomic indicators can provide valuable insights into market sentiment and trends. Future iterations of the project could explore the integration of such data to improve prediction accuracy and robustness.

Ensemble Methods and Model Stacking:

Investigating ensemble learning techniques, such as model stacking and blending, can further enhance prediction performance by combining the strengths of multiple base models. By leveraging the diversity of different algorithms, ensemble methods can mitigate individual model biases and improve overall prediction quality.

Deep Learning Architectures:

Exploring deep learning architectures, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, for stock price prediction represents an intriguing avenue for future research. These architectures are well-suited for capturing temporal dependencies in sequential data, making them potentially effective for modelling the dynamic nature of financial markets.

Feature Engineering and Selection Strategies:

Refining feature engineering techniques and exploring automated feature selection methods can help identify the most relevant predictors for stock price prediction. Future research could focus on leveraging domain knowledge and advanced feature engineering algorithms to extract informative features from diverse data sources.

Model Interpretability and Explainability:

Enhancing the interpretability and explainability of prediction models is crucial for building trust and understanding in financial applications. Future iterations of the project could explore techniques for model interpretability, such as feature importance analysis, SHAP (Shapley Additive explanations) values, and model-agnostic explanation methods.

Real-Time Prediction and Deployment:

Transitioning from offline batch prediction to real-time prediction systems can enable timely decision-making for traders and investors. Future work could focus on developing scalable, low-latency prediction pipelines and deploying models in production environments for real-world applications.

Market Regime Detection and Adaptive Models:

Incorporating techniques for detecting changes in market regimes and adapting prediction models accordingly can enhance model robustness in dynamic market environments. Future research could explore adaptive learning algorithms and regime-switching models to capture changing market conditions effectively.

Risk Management and Portfolio Optimization:

Expanding the scope of the project to include risk management and portfolio optimization strategies can provide a holistic approach to financial decision-making. Future work could explore incorporating risk measures, diversification techniques, and portfolio rebalancing strategies into the prediction framework to maximize risk-adjusted returns.

By addressing these future avenues, the project can evolve into a more comprehensive and sophisticated framework for stock price prediction, offering valuable insights and practical solutions for stakeholders in the finance industry.

Chapter 6

6.1 CONCLUSION

In conclusion, our project explored the effectiveness of various machine learning algorithms in predicting stock prices. We tested Multi-Layer Perceptron (MLP) classification, Extra Trees Classifier, and Gradient Boosting Classifier, revealing nuanced differences in their performance across metrics and datasets.

MLP excelled in capturing non-linear relationships and adapting to market changes, while Extra Trees Classifier handled high-dimensional data well, mitigating overfitting. Gradient Boosting Classifier demonstrated strong predictive performance and interpretability.

No single algorithm emerged as the unequivocal "best" choice, emphasizing the importance of considering dataset characteristics and computational constraints. Future research could explore ensemble methods, deep learning, and alternative data integration to enhance accuracy and interpretability.

This project contributes to stock price prediction dialogue, informing practitioners and researchers for informed decision-making. Collaboration and interdisciplinary research can advance prediction accuracy and empower stakeholders with actionable insights in finance.

Stock price prediction is complex due to market volatility. Machine learning offers promise but presents challenges. MLP captured non-linear dynamics effectively but requires careful tuning. Extra Trees handled high-dimensional data well, while Gradient Boosting provided strong predictions and interpretability.

However, no algorithm is universally superior. Choice should consider dataset characteristics, resources, and interpretability needs.

Future research can explore ensemble methods and deep learning for improved accuracy. Integrating alternative data sources like sentiment analysis could enhance insights.

In summary, our project contributes to understanding algorithm effectiveness in stock price prediction. Through careful evaluation and collaboration, we can navigate financial complexities and drive innovation in prediction methodologies.

Chapter 7

7.1 REFERENCE

[1] T.T. Ho and Y. Huang, "Stock price movement prediction using sentiment analysis and CandleStick chart representation," Sensors, vol. 21, no. 23, p. 7957, 2021.

[2] M. Ananthi and K. Vijayakumar, "Retraction note to: stock market analysis using candlestick regression and market trend prediction (CKRM)," J Ambient Intell Humaniz Comput, vol. 68–73, 2022.

[3] C.C. Hung, Y.J. Chen, S.J. Guo, and F.C. Hsu, "Predicting the price movement from candlestick charts: a CNN-based approach," Int J Ad Hoc Ubiqui Comput, 2020, p. 111.

[4] R.M. Dhokane and S. Agarwal, "A Predictive Model of the Stock Market Using the LSTM Algorithm with a Combination of Exponential Moving Average (EMA) and Relative Strength Index (RSI) Indicators," J Inst Eng India Ser B, 2024.

[5] W. Long, Z. Lu, and L. Cui, "Deep learning-based feature engineering for stock price movement prediction," Knowl Based Syst, pp. 163–173, 2019.

[6] R.K. Behera, M. Jena, S.K. Rath, and S. Misra, "Co-LSTM: convolutional LSTM model for sentiment analysis in social big data," Inf Process Manage, 2021, p. 102435.

[7] B. Dhyani, M. Kumar, P. Verma, and A. Jain, "Stock market forecasting technique using ARIMA model," Int J Recent Technol Eng (IJRTE), vol. 8, no. 6, 2020.

[8] A. Andriyanto, "Sectoral stock prediction using convolutional neural networks with candlestick patterns as input images," Int J Emerg Trends Eng Res, pp. 2249–2252, 2020.

[9] A. Bansal, "Prediction and Analysis of Stock Market using ARIMA Model and Machine Learning Techniques," NeuroQuantology, pp. 3804-3810, 2022.

[10] B. Jin, S. Gao, and Z. Tao, "ARIMA and Facebook Prophet Model in Google Stock Price Prediction," Proceedings of Business and Economic Studies, pp. 60-66, 2022.

[11] D. Bathla, A. Garg, and Sarika, "Stock Trend Prediction Using Candlestick Pattern," Springer, Singapore, 2023.

turnitin

Similarity Report ID: oid:28541:58890517

PAPER NAME

PROJECT REPORT FINAL-2.pdf

WORD COUNT	CHARACTER COUNT
6869 Words	**42922 Characters**
PAGE COUNT	FILE SIZE
45 Pages	**2.3MB**
SUBMISSION DATE	REPORT DATE
May 8, 2024 4:05 PM GMT+5:30	**May 8, 2024 4:05 PM GMT+5:30**

20% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 20% Internet database
- 7% Publications database
- Crossref database
- Crossref Posted Content database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)