STOCK PRICE PREDICTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted By:

ANURAG KUMAR

University Roll Number:

20BCS4567

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(INTERNET OF THINGS)

Under the Supervision of:

MR. PULKIT DWIVEDI

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI – 140413, PUNJAB

NOVEMBER,2023

STOCK PRICE PREDICTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

ANURAG KUMAR (20BCS4567)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(INTERNET OF THINGS)

CHANDIGARH UNIVERSITY

NOVEMBER, 2023

BONAFIDE CERTIFICATE

Certified that this project report **"STOCK PRICE PREDICTION USING MACHINE LEARNING"** is the bonafide work of **"ANNU (20BCS4608), ANURAG KUMAR (20BCS4567) who carried out the project work under my/our supervision**

AMAN KAUSHIK

HEAD OF THE DEPARTMENT (AIT-CSE)

PULKIT DWIVEDI

SUPERVISOR

DECLARATION

I, **'ANURAG KUMAR', '** student of **'Bachelor of Engineering in the Internet of Things**, **session: 2020-2024**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, at this moment, declare that the work presented in this Project Work entitled **'STOCK PRICE PREDICTION USING MACHINE LEARNING'** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

CANDIDATE NAME & UID

ANURAG KUMAR (20BCS4567)

Place: Chandigarh University

Month & Year: NOVEMBER, 2023

TABLE OF CONTENTS

ABSTRACT

This study explores stock price prediction utilizing a machine learning approach, specifically employing the ARIMA (Auto Regressive Integrated Moving Average) model. Leveraging regression techniques and model evaluation through AIC (Akaike Information Criterion), the project research aims to enhance forecasting accuracy. The hypotheses testing assesses the significance of model parameters in predicting stock prices. By integrating statistical methodologies and machine learning, this study contributes to the advancement of robust and reliable stock price prediction models, offering valuable insights for investors and financial analysts.

ACKNOWLEDGEMENT

Apart from the efforts of all the crew members, the segment of this challenge document subject matter relies upon in large part the encouragement and steerage of our teachers. We take this possibility to specifically our gratitude to the academics who've been instrumental in the approval of this challenging subject matter. We would like to show our greatest appreciation to **Mr. Pulkit Dwivedi** and our Chancellor **Mr. Satnam Singh Sandhu** for giving us this golden opportunity to complete a great project on "**STOCK PRICE PREDICTION USING MACHINE LEARNING**" which also helped us do a lot of research and we came to new about so many new things and achieved a good amount of knowledge through this project. The contribution and support received from all the team members '**ANNU', 'ANURAG KUMAR', and 'GURLEEN KAUR'** including are high-spirited. The team spirit shown by all has made this project successful.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The financial markets have always been a subject of intense interest and scrutiny due to their impact on economies, businesses, and individual investors. In this era of rapid technological advancement, the ability to predict stock prices accurately has gained paramount importance. Accurate stock price predictions not only empower investors to make informed decisions but also offer valuable insights to financial analysts and policymakers.

Traditionally, stock price prediction has relied on fundamental analysis, technical analysis, and market sentiment analysis. However, the advent of machine learning techniques has ushered in a new era of predictive analytics. Machine learning algorithms, with their ability to analyze vast amounts of historical data and identify intricate patterns, have shown promise in improving the accuracy of stock price predictions.

Because of the inherent volatility and complexity of the stock market, predicting stock prices is a difficult task. Historically, conventional linear regression models were used to model stock price movements by taking various financial and economic indicators into account as independent variables. These models, however, frequently fail to capture the intricate and nonlinear relationships that exist in financial markets.

The availability of massive amounts of financial data, combined with advances in machine learning, has paved the way for more advanced models in recent years. To make predictions, these models can take into account not only financial metrics but also sentiment analysis of news articles, social media, and other alternative data sources.

This project endeavors to explore the application of machine learning methodologies in the domain of stock price prediction. We will delve into various aspects of this complex problem, including data preprocessing, feature engineering, model selection, and evaluation metrics. Our primary aim is to provide a comprehensive analysis of the strengths and limitations of machine learning models in forecasting stock prices.

1.1. STOCK MARKET

The stock market is a network of markets where investors buy and sell stocks and other securities. Shares of ownership in publicly listed corporations are available to the public, and those shares can be bought and sold on the stock market. Investors might profit by purchasing shares of a company at a cheap price and selling them at a higher price. The stock market is an important component of the global economy, providing finance for business growth and expansion. It is also a popular strategy for individuals to invest and develop their money over time.

A financial exchange where buyers and sellers exchange interests in publicly traded corporations is the stock market. It provides a marketplace for both individual and institutional stock buyers and sellers, representing ownership positions in companies. Because it makes it easier for businesses to get capital and gives investors a chance to invest in the expansion and success of these companies, the stock market is essential to the larger financial system. The dynamics of supply and demand, which are impacted by a number of variables including market mood, company performance, and economic conditions, drive stock prices. Investors have two options: they can purchase stocks at a discount and sell them for a profit, or they can take dividends and regular payments to shareholders issued by certain companies as income. An essential part of international finance, the stock market influences economies and reflects investor expectations and views regarding the prospects of individual companies and the economy as a whole.

1.2. IMPORTANCE OF STOCK PRICE PREDICTION

Predicting stock prices is important to many different financial market participants. These forecasts are used by traders, policymakers, investors, and financial experts to make well-informed judgments. Accurate stock price forecasts can provide traders and investors with information about possible investment opportunities, allowing them to manage risks, optimize their portfolios, and decide whether to purchase or sell. Stock price forecasts are used by financial analysts to analyze market trends, appraise company performance, and provide client recommendations. These forecasts are also essential for risk management plans, which enable participants to profit from market swings or protect themselves from future losses. Because the stock market frequently reflects broader economic conditions, policymakers utilize stock market knowledge to monitor the state of the economy. All things considered, stock price prediction is essential to the efficiency and openness of the financial system, providing players with important information to help them negotiate the challenges of investing and support the stability of the economy as a whole.

1.3. ROLE OF MACHINE LEARNING IN STOCK PRICE PREDICTION

Machine learning, by leveraging the power of data analysis and pattern recognition, plays a critical role in forecasting stock values. Economic statistics, company news, market mood, and other variables all have an impact on the stock market. Machine learning algorithms can filter through massive information, recognizing past pricing patterns and comprehending complicated correlations between various factors. They are capable of detecting tiny patterns and connections that human analysts may miss. To model and predict stock values, these algorithms employ numerous approaches such as regression analysis, time series forecasting, and deep learning. To generate more accurate forecasts, they take into account not only previous stock price movements but also external elements such as news mood, macroeconomic data, and geopolitical events.

Furthermore, machine learning models evolve and improve over time by learning from fresh data and so improving their predicting powers. While machine learning may give useful insights and forecasts, it's crucial to remember that stock markets are fundamentally unpredictable, and previous success does not guarantee future outcomes. Nonetheless, machine learning algorithms provide advanced risk management and decision-making capabilities to traders and investors in the fast-paced world of stock trading.

Machine learning algorithms can handle the vast amounts of financial data generated every day. These algorithms gather relevant data from a variety of sources, including historical stock prices, trade volumes, economic indicators, company reports, and news moods. The basis for predictive models is provided by feature extraction. One of the key advantages of machine learning is its capacity to recognize complicated patterns and correlations in financial data. Human analysts may struggle to grasp these patterns due to their intricacy and the massive quantity of data involved. Machine learning algorithms, which can discover correlations between multiple components and prior price movements, can help make better forecasts.

1.4. TIMELINE

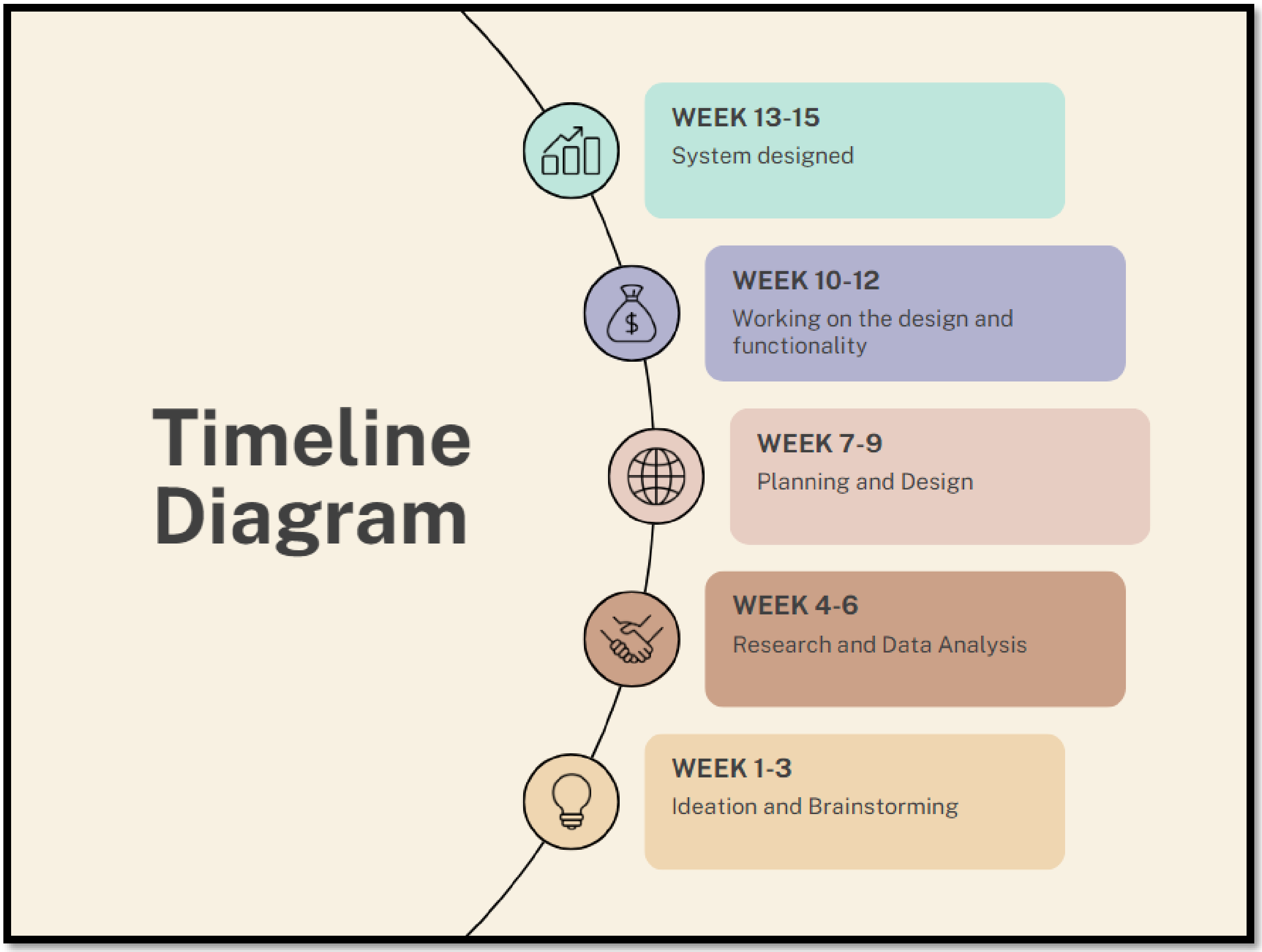

Timeline Diagram
WEEK 13-15
System designed
WEEK 10-12
Working on the design and functionality
WEEK 7-9
Planning and Design
WEEK 4-6
Research and Data Analysis
WEEK 1-3
Ideation and Brainstorming

Figure 1

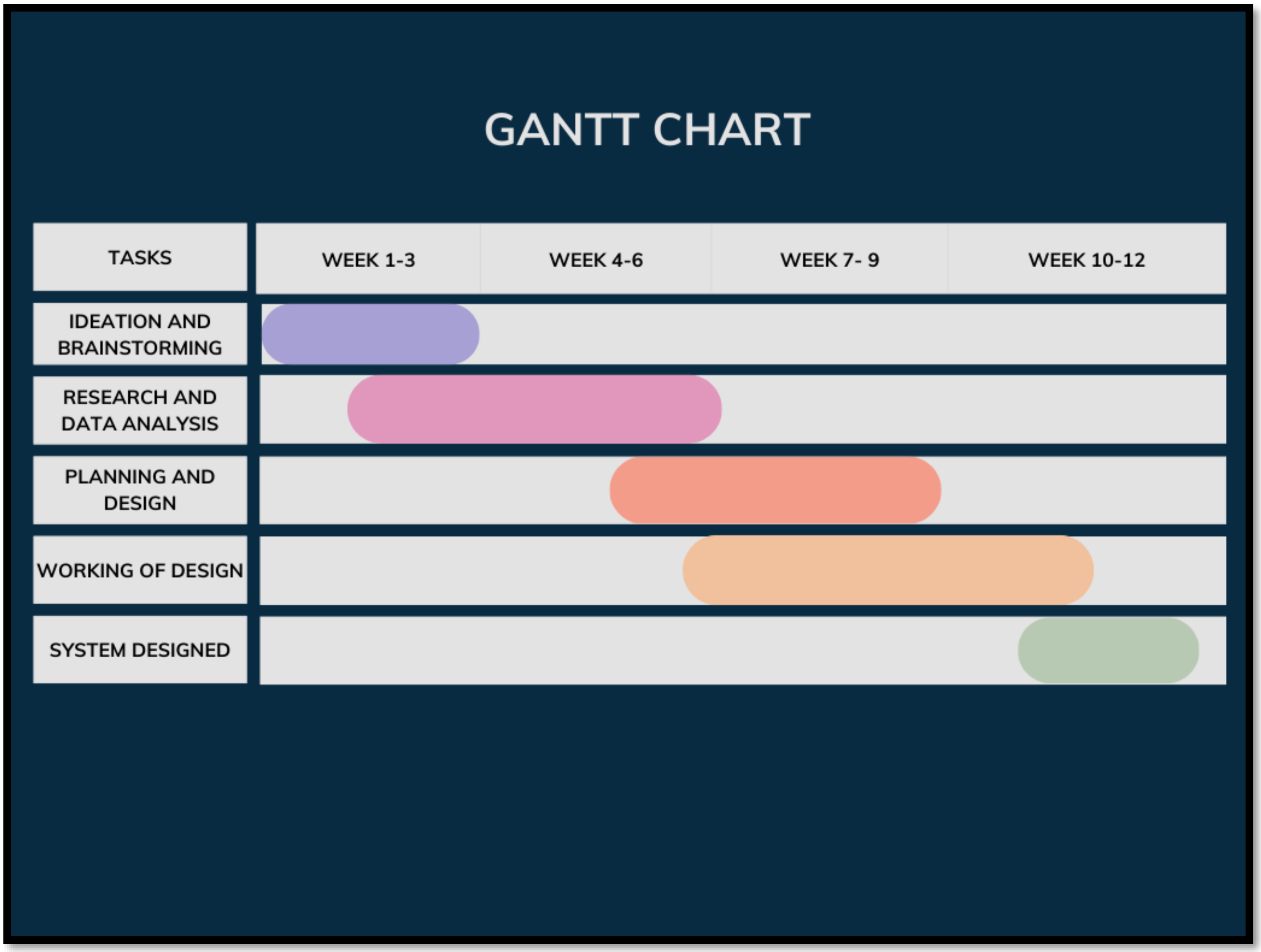


Figure 2

1.5. PROJECT SCOPE

The completion of this project opens up chances for further research and progress in the field of financial forecasting. Future research may include more advanced sentiment analysis approaches, research into cutting-edge deep learning models, and increased real-time prediction abilities. The study also emphasizes the importance of constantly altering and updating models to reflect changing market conditions. This project's purpose is to develop and test an advanced regression model for stock price prediction. The scope includes the following important items:

- The examination of a single stock or index.
- Utilization of past stock price data.

- Alternative data sources have been integrated, such as news sentiment analysis.
- Cutting-edge machine learning methods are utilized to improve forecast accuracy.
- The model's output is interpreted to give helpful decision-making information.

1.6. REQUIREMENTS

- Python IDE: An application known as the Python IDE (Integrated Development Environment) gives programmers all the tools they need to develop Python software. A code editor, a debugger, and a graphical user interface (GUI) for project management are commonly found in an IDE. Python IDEs may include Google Colab, Jupyter, PyCharm, and VS Code.

  PyCharm is an IDE for Python that is robust and packed with features, developed by JetBrains. It provides numerous productivity features, debugging support, and code analysis capabilities.

  Visual Studio Code (VSCode) is a lightweight, open-source code editor that supports Python through extensions, but it's not just an IDE for Python. It includes a big ecosystem of extensions and is quite flexible.

  With Jupyter, an open-source web program, you can create and distribute documents with narrative prose, equations, live code, and visualizations. In data science and research, it's commonly used.

  Google Colab, also known as Collaboratory, is an online platform that works similarly to a Jupyter Notebook and lets you write and run Python programs. In the context of data science, machine learning, and collaborative research, Google Colab is an extremely useful tool for executing Python code, even if it isn't an IDE like PyCharm or Visual Studio Code.

  Some of the Python libraries used in carrying out this project may include the following:

⇒ NumPy is a strong numerical computation package for the Python programming language. It supports massive, multi-dimensional arrays and matrices, as well as a set of mathematical functions for manipulating these arrays. NumPy is a core Python library for scientific computing that is extensively used in domains such as machine learning, data science, engineering, and scientific research.

⇒ Pandas is a well-known open-source data manipulation and analysis package for Python. It includes data structures for storing and processing huge datasets, as well as data cleaning, exploration, and analysis tools. Pandas' two main data structures are:

Series: Any data type can be stored in a one-dimensional labeled array. The same as a spreadsheet column or a single variable in statistics.

DataFrame: A labeled two-dimensional data structure containing columns of several data kinds. A spreadsheet or an SQL table is analogous. Structured data may be easily manipulated and analyzed with this tool.

⇒ Scikit-learn is a popular machine learning toolkit for the Python programming language. It offers easy and efficient data analysis and modeling tools, including as machine learning methods for classification, regression, clustering, dimensionality reduction, and more. The following are some of scikit-learn's core features and components:

API consistency: Scikit-learn provides a uniform and simple API for a variety of machine learning operations. This makes it more accessible to consumers and enables the transition between various algorithms.

Supervised Learning: Includes a wide range of supervised learning techniques such as linear and logistic regression, support vector machines, decision trees, and ensemble approaches such as random forests.

Unsupervised Learning: Provides unsupervised learning algorithms such as clustering approaches (KMeans, hierarchical clustering) and dimensionality reduction techniques (Principal Component Analysis - PCA).

Model Selection: Model selection and assessment methods are provided, including cross-validation procedures, grid search for hyperparameter tweaking, and model performance measures.

Preprocessing: Data preparation tools such as scaling, normalizing, encoding categorical variables, and managing missing values are included.

Feature Extraction: Provides feature extraction methods and tools for identifying key characteristics in a dataset.

Ensemble Methods: Ensemble techniques like as bagging and boosting are supported, as are random forests and AdaBoost implementations.

Pipeline: Allows the creation of machine learning pipelines, allowing for the seamless integration of preprocessing, feature extraction, and model training.

Integration with NumPy and Pandas: Integrates easily with NumPy arrays and Pandas DataFrames, simplifying data handling and interoperability with other Python data science packages.

Community and documentation: Scikit-learn has a large community and rich documentation, making it a great resource for both novice and professional machine learning practitioners.

⇒ Matplotlib is a comprehensive 2D charting package for Python that generates high-quality graphics. It offers a wide range of charts and graphs, making it an effective tool for data visualization and exploration. Matplotlib's primary features include the following:

Line Plots: Displaying data points with lines linking them, excellent for time series data or trends.

Scatter Plots: Individual data points are shown on a two-dimensional plane, which is useful for investigating correlations between variables.

Plots with Bars: Data is shown using rectangular bars, which are useful for comparing categories.

Histograms: The frequency of distinct values is shown in the distribution of a single variable.

Pie charts: Data is shown in a circular graph with each slice representing a percentage of the total.

- Machine Learning: Within the field of artificial intelligence (AI), machine learning focuses on creating models and algorithms that allow computers to learn and make decisions without explicit programming. Enabling machines to learn from data and gradually enhance their performance is the fundamental concept. Machine learning involves training computers on a dataset so they can find patterns, correlations, and trends in the data through an iterative learning process. After gaining this knowledge, judgments or predictions are based on fresh, unobserved data. Applications for machine learning are numerous and include recommendation systems, natural language processing, picture and audio recognition, and more. Three popular machine learning approaches—supervised learning, unsupervised learning, and reinforcement learning—are appropriate for various task kinds. The discipline keeps developing quickly, spurring advancements and discoveries across a range of sectors.
- Basic knowledge of linear programming.
- Knowledge about ARIMA Model.

CHAPTER 2

LITERATURE REVIEW/BACKGROUND STUDY

2.1. EXISTING SOLUTIONS

The existing systems include predicting stock prices using various Machine Learning algorithms. Some of these are:

- Recurrent Neural Networks (RNNs) and lengthy quick-term memory (LSTM): Many studies papers have explored the usage of RNNs and LSTM networks for stock price prediction. these networks are capable of taking pictures of temporal dependencies in inventory rate data
- Convolutional Neural Networks (CNNs): CNNs, which might be historically used in photo processing, have been adapted to research stock price facts inside the shape of pix (e.g., candlestick charts). This approach can seize styles and traits within the information.
- Ensemble strategies: Ensemble techniques, which include Random Forests and Gradient Boosting, have been applied to mix the predictions of a couple of system learning models, frequently resulting in extra accurate predictions.
- Reinforcement getting to know (RL): A few researchers have focused on using reinforcement getting-to-know techniques to make buying and selling decisions based on inventory charge predictions. RL retailers discover ways to take actions that maximize cumulative rewards through the years.
- Technical indicators and features: Research regularly includes the creation of features from technical indicators like transferring averages, Relative strength Index (RSI), and Bollinger Bands, which are then used as input to gadget studying fashions.

- Sentiment analysis: A few papers contain sentiment evaluation of information articles, social media, or monetary reports to gauge marketplace sentiment and include these statistics in stock price prediction models.
- Gaussian strategies: Gaussian tactics have been used for modeling inventory charge moves and uncertainties. They are mainly beneficial for modeling non-linear relationships.
- Attention Mechanisms: Interest mechanisms have been hired to weigh different time steps in the historical statistics in another way, permitting the version to attention to more applicable information.
- Time collection Forecasting techniques: Classical time collection forecasting methods like ARIMA (auto-regressive integrated shifting common) have also been used as benchmarks or components of more complicated models.
- Hybrid Models: A few papers advocate hybrid models that combine the strengths of more than one process, such as combining LSTM networks with technical signs.
- Quantitative models: A few studies in inventory prediction make use of quantitative fashions, which can be based totally on mathematical and statistical procedures instead of gadget studying. for example, the Black-Scholes model is a well-known quantitative model for alternative pricing.

It is crucial to remember that projecting stock prices is a difficult endeavor because of the complexity of financial markets, the prevalence of noise in price data, and the impact of unexpected occurrences. No model can predict stock prices with 100% confidence, and their performance can vary greatly based on a variety of factors, including the quality and quantity of data employed. Researchers are always working to improve the accuracy and dependability of stock price prediction models.

Challenges Faced by Existing Systems:

Many of the problems that currently exist in stock price prediction models are caused by the complexity and volatility of the financial markets. The precision and dependability of predictions may be impacted by these difficulties. The following are some typical difficulties that current stock price prediction models encounter:

- Financial markets are impacted by a wide range of elements, including news, events, mood, market noise, and random fluctuations. These marketplaces can display significant noise and erratic volatility, making it difficult to identify significant trends in the data.
- Stock prices frequently exhibit non-stationarity or a shift in their statistical characteristics over time. Because of this, it is challenging to directly apply conventional time series forecasting methods.
- Data quality: Errors, missing numbers, and inconsistencies in past financial data can have a big impact on how well predictive models work.
- Overfitting: Complex machine learning models are susceptible to overfitting, which causes them to perform well on training data but badly on untrained data. Due to the noisy and non-linear nature of financial data, overfitting is a serious risk in stock price prediction.
- Limited Data: The length of the available time series limits the historical stock price data, which might make it more difficult to train reliable and accurate models, especially for long-term forecasts.
- Market Behaviour Change: Changes in trading patterns, rules, or market dynamics are examples of structural changes that financial markets may experience. Models that do not modify to reflect these changes may eventually lose their effectiveness.
- Data Snooping and Survivorship Bias: When several models or strategies are tried on the same historical data, data snooping might happen, potentially selecting strategies that did well by accident. Survivorship bias happens when data is only used for stocks that are still trading and ignores stocks that have been delisted or gone out of business.

- High Frequency and Real-Time Data: In today's markets, real-time data and high-frequency trading can present problems for models that need quick and effective processing.
- Global Financial Markets are Interconnected: Because of this, developments in one market can quickly affect another, making projections much more difficult.

Advanced modeling methods, feature engineering, data pre/treatment, continuous model monitoring, and model adaptation are frequently combined to address these issues. Additionally, stakeholders should be appropriately informed of the constraints and uncertainties that come with stock price prediction by scholars and practitioners in this field.

2.2. PROPOSED SYSTEM

The proposed system for the project "Stock Price Prediction Using Machine Learning" integrates an adaptive regression model to enhance the accuracy and adaptability of stock price predictions. This adaptive approach aims to capture the dynamic nature of financial markets and adjust to changing conditions, leading to more reliable forecasts, which include Time series analysis along with sentiment analysis. Incorporate sentiment analysis of news articles, social media, or financial reports as additional features. Sentiment analysis can provide valuable insights into market sentiment, which can impact stock prices.

Feature selection methods Employ feature selection techniques to identify the most relevant features for your linear regression model. The performance of the model can be enhanced by removing unnecessary or duplicate features. Finally, evaluate the algorithm by RMS or R2 method.

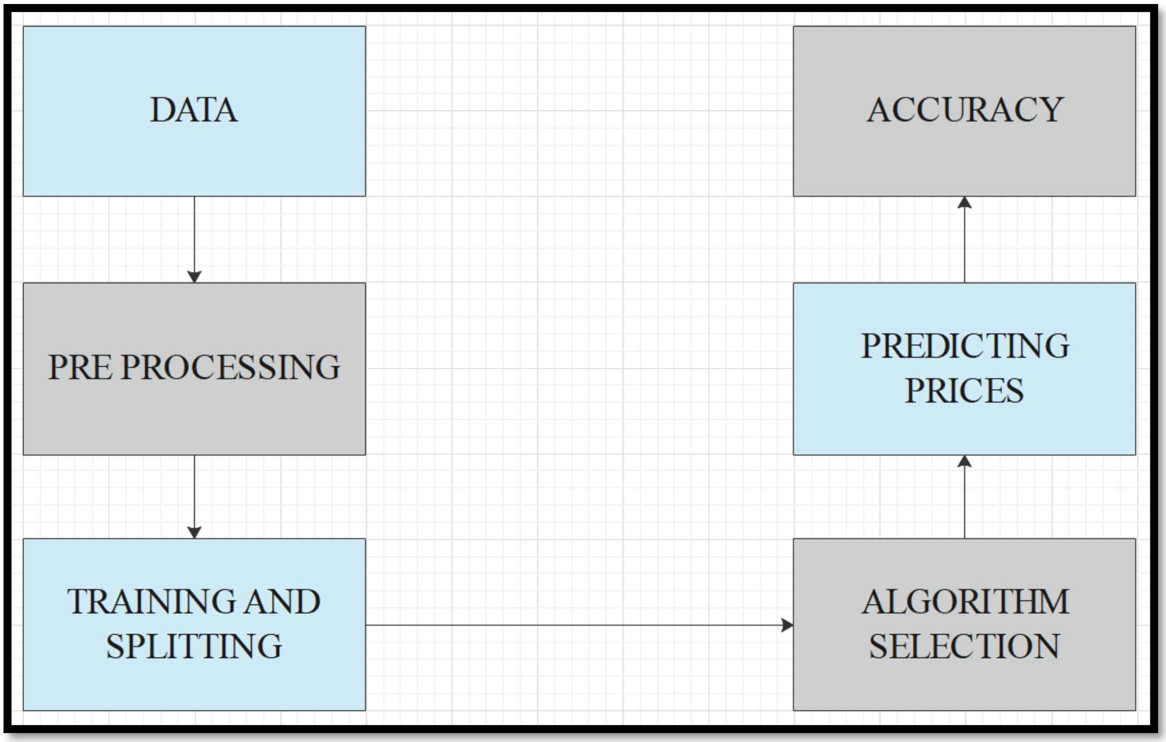


Figure 3

2.3. RELATED WORK

a. Stock Market Prediction System Using Machine Learning Approach.

Faisal Momin, Sunny Patel, Kuldeep Shinde, and Prof. Reena Sahane of the Sandip Institute of Engineering and Management in Nashik, India, carried out the research. The research uses a machine learning technique to predict stock prices using a neural network backpropagation algorithm and gradient descent optimization on stock price data. The back propagation neural network approach seeks to select the parameter of learning rate, and training cycle adaptively to acquire the best value throughout the process of stock data training to obtain accuracy in prediction. The values of the training cycle and learning rate were determined using each of these methods.

b. Stock Market Prediction Using Machine Learning.

V Kranthi Sai Reddy's research was conducted at the Hyderabad, India-based Sreenidhi Institute of Science and Technology. One of the most significant activities in the world of finance is stock trading. Trying to anticipate the future value of a stock or other financial instrument traded on a financial exchange is known as the stock market prediction. This paper explains how machine learning can be used to predict stocks. Most stockbrokers employ technical, fundamental, or time series analysis when making stock predictions. The Python programming language was used to make stock market predictions using machine learning. In this research, the authors suggest a machine learning (ML) method that will be taught using the stock data already accessible to gather intelligence and then apply the learned information to a precise prediction. This study employs pricing with both daily and up-to-the-minute frequencies and a machine learning technique known as Support Vector Machine (SVM) to forecast stock prices for both large and small capitalizations in the three separate marketplaces.

c. Stock Market Forecasting Using Machine Learning Algorithms.

Shunrong Shen, Haomiao Jiang, and Tongda Zhang from Stanford University collaborated on this study project. In this study, they have suggested using global stock data along with data from other financial products as the input features to machine learning algorithms like SVM. A new prediction method that uses SVM to predict the next day's stock trend by making use of the temporal link between various financial products and the various stock markets around the world. The numerical statistics show a forecast accuracy of 74.4% for NASDAQ, 76% for S&P500, and 77.6% for DJIA. It also tracked the actual increase in the markets, the same approach is also used with other regression algorithms. They compared the performance of the suggested prediction algorithm to existing benchmarks, a straightforward trading model is created.

d. Stock Price Prediction Using Machine Learning and Deep Learning Frameworks.

Jaydip Sen from the Praxis Business School in Kolkata completed this paper. Python is the programming language used in this study, and LSTM networks were implemented using the TensorFlow deep learning framework. In a multivariate time series framework, those networks were used to forecast the prices of Tata Steel and Hero Motocorp. The implementation also made use of a variety of machine-learning methods, such as logistic regression, KNN, decision trees, bagging, boosting, random forest trees, ANN, SVM, multivariate regression, etc. Using the Metastock tool throughout two years, from January 2013 to December 2014, stock price data was gathered at five-minute intervals.

e. Stock Closing Price Prediction using Machine Learning Techniques.

This research paper is a contribution by Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar from the Jaypee Institute of Information Technology, Noida. Due to the financial stock markets' volatility and non-linearity, accurately predicting stock market returns is an extremely difficult undertaking. The development of artificial intelligence and improved processing power have made programmed methods of prediction more effective at forecasting stock prices. In this study, five businesses from various industry sectors had their closing prices predicted using artificial neural networks and random forest techniques. The stock's open, high, low, and close prices are used to create new variables that are then used as model inputs. RMSE and MAPE, two common strategic indicators, are used to evaluate the models. The models are effective at forecasting stock closing prices, as evidenced by the low levels of these two indicators.

f. A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing.

Different Machine Learning models are used in the research conducted by Sidra Mehtab and Jaydip Sen from the NSHM Knowledge Campus in Kolkata, India. They have used eight regression and eight classification algorithms to give several strategies for stock price and movement prediction on a weekly forecast horizon. These models were built using deep learning and machine learning techniques. These models were created, adjusted, and tested using daily historical NIFTY 50 data from January 2, 2015, to June 28, 2019. Metrics like sensitivity, specificity, positive predictive value, negative predictive value, and classification accuracy were used to evaluate the model in this study. The capacity of this sentiment analysis-enhanced model to predict the movement of the NIFTY 50 stock price with accuracy has been determined to be the best among all models.

g. Machine Learning Approaches in Stock Price Prediction: A Systematic Review.

This paper was produced in part by Payal Soni, Yogya Tewari, and Prof. Deepa Krishnan of the Mukesh Patel School of Technology Management and Engineering at NMIMS University in Mumbai. This study examines the many methods for predicting stock prices, including classical machine learning, deep learning, neural networks, and graph-based algorithms. It provides a thorough review of the methods used to forecast stock values and looks at the difficulties involved as well as the direction of future research in the field. The study was expanded to include deep learning and neural network models such as convolutional neural networks, artificial neural short-term memory, long-short-term memory, etc. in addition to classic ML methods like RF, KNN, SVM, Naive Bayes, etc. The study examines the outcomes of these algorithms to forecast the stock values of various firms and also includes several additional methodologies, including sentiment analysis, time series analysis, and graph-based algorithms.

h. Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions.

The following individuals from BGSB University (Rajouri, India), NIT Srinagar, Dongguk University (Seoul, Korea), and Inje University (Gimhae, Korea) contributed to this paper: Nusrat Rouf, Majid Bashir Malik, Tasleem Arif, Sparsh Sharma, Saurabh Singh, Satyabrata Aich, and Hee-Cheol Kim Several machine-learning models that were employed in the research are included in this paper. They have determined the distribution of the SMP approaches and the number of publications every year, and they have compared the accuracy to various forms of data, such as market data and textual data. In this piece, research utilizing a general SMP paradigm was reviewed. It largely concentrated on research from the most recent ten years (2011–2021). Based on the types of data utilized as input, the methods for pre-processing the data, and the machine learning strategies applied for the predictions, the research was examined and contrasted. It also reviewed the various evaluation measures applied to performance measurement across various studies.

2.4. REVIEW SUMMARY

Sr.No.	Author	Article	Tools/Techniques
1.	Faisal Momin, Sunny Patel, Kuldeep Shinde, Prof.A.C.Taskar	Stock market prediction system using machine learning approach	Machine Learning, Neural Networks
2.	Shunrong Shen, Haomiao Jiang, Tongda Zhang	Stock Market Forecasting Using Machine Learning Algorithms	Machine Learning algorithms, SVM
3.	Jaydip Sen	Stock Price Prediction Using	ML Techniques- Logistic Regression,

		Machine Learning and Deep Learning Frameworks	KNN, Decision Tree, ANN, SVM
4.	Mehar Vijh , Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar	Stock Closing Price Prediction Using Machine Learning Techniques	ML Techniques- ANN, Random Forest Model
5.	Sidra Mehtab, Jaydip Sen	A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing	Classification, Regression, LSTM
6.	Payal Soni, Yogya Tewari, and Prof. Deepa Krishnan	Machine Learning Approaches in Stock Price Prediction: A Systematic Review	Machine Learning, Deep Learning, and Neural Networks
7.	Nusrat Rouf, Majid Bashir Malik, Tasleem	Stock Market Prediction Using Machine	ANN, SVM, DNN, Regression Algorithms, Naïve

	Arif, Sparsh Sharma, Saurabh Singh, Satyabrata Aich, and Hee-Cheol Kim	Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions	Bayes, Genetic Algorithms, etc.

2.5. PROBLEM DEFINITION:

The future worth of business stock and other financial assets traded on an exchange can be found using machine learning stock price prediction. Gaining significant profits is the entire point of stock price forecasting. It might be challenging to forecast the direction of the stock market. Other elements including physical and psychological aspects, reasonable and irrational conduct, and so forth are also taken into account while making a forecast. Share prices are dynamic and unstable as a result of the interaction of all these elements. Because of this, very accurate stock price predictions are quite challenging to create. This is why a stock price prediction model needs to be there in the market.

2.6. GOALS / OBJECTIVES

The primary objectives of this project are as follows:

- Develop an Advanced Regression Model: To create a sophisticated regression model that can capture complex patterns and relationships in stock price data.
- Incorporate Alternative Data: To integrate non-traditional data sources, such as news sentiment analysis and social media sentiment, to improve prediction accuracy.

- Enhance Model Interpretability: To ensure that the model is interpretable and can provide insights into the factors driving stock price movements.
- Evaluate Model Performance: To rigorously assess the model's performance using appropriate evaluation metrics and statistical tests.
- Provide Insights for Investment Decisions: To deliver actionable insights that can guide trading and investment strategies.

CHAPTER 3

DESIGN FLOW/PROCESS

3.1. EVALUATION & SELECTION OF SPECIFICATIONS/FEATURES

The proposed architectural design may include specifications/features such as:

- Data preprocessing
- Feature Selection
- Building and Training Model.

Data Preprocessing: It is a crucial step in every model. It started with data gathering which consists of gathering historical data that typically includes information like open, high, low, and close price entries present in the dataset. Missing values were imputed and outliers were removed from the data as in financial data, missing values are common due to market holidays or errors in data feeds. The null values were removed using a particular function. The categorical values are converted into numerical values. Relevant features were created that helped the model capture patterns in the data. Common features include moving averages, volatility measures, and technical indicators.

Feature Selection: This was done to build the model. Identified and retained features that were highly correlated with the target variable. Different Machine Learning algorithms and statistical tests were used to determine the features used for prediction.

Building and Training Model: Different Machine Learning algorithms were employed, including adaptive algorithms, linear programming, etc. For this, the appropriate model needs to be selected first. Split the data into training and testing sets. Cross-validation techniques are used to prevent overfitting and train the selected model on the training data using features derived from data preprocessing and sentiment analysis. Evaluated the model performance using appropriate metrics on the testing data. They optimized the model to improve its accuracy.

3.2. DESIGN CONSTRAINTS

There are numerous data restrictions that you may meet in the context of the ARIMA (Auto Regressive Integrated Moving Average) model for time series analysis. Addressing these limits is critical to achieving trustworthy and relevant results from your ARIMA model. Here are some examples of frequent data limitations connected with ARIMA modeling:

- Stationarity: ARIMA models assume that time series data is stationary, which means that its statistical features do not fluctuate over time. Non-stationary data might lead to inaccurate model findings.
- Seasonality: ARIMA algorithms may fail to capture seasonality trends in data.
- Data Gaps and Missing Data: ARIMA models demand a continuous time series with no gaps or missing data.
- Outliers: Outliers can have a substantial influence on the performance of ARIMA models because they can alter the underlying patterns in the data.
- Data Frequency: ARIMA models are generally developed for evenly spaced time intervals.
- Model Ordering: Choosing the correct order of differencing (d), autoregressive (p), and moving average (q) terms may be difficult.
- Limited history Data: ARIMA models may require a significant amount of history data to find trends and generate good forecasts.
- ARIMA presupposes a linear connection between previous observations and future projections.
- Model Sensitivity to Parameter Changes: Small changes in model parameters might result in unexpected results.

- External Factors: ARIMA models may fail to account for external factors that impact the time series but are not included in the model

3.3. ANALYSIS OF FEATURES AND FINALIZATION SUBJECT TO CONSTRAINTS

- Stationarity: Using differencing techniques, make the data steady. This is done by calculating the difference between two successive observations.
- Seasonality: If seasonality is present, consider adopting a Seasonal ARIMA (SARIMA) model, which extends ARIMA to account for seasonal variations.
- Data Gaps and Missing Values: Make up or interpolate missing numbers, and make sure there are no gaps in the time series data.
- Outliers: Outliers should be identified and handled correctly utilizing techniques such as outlier detection or robust modeling methodologies.
- Data Frequency: Ensure that your time series data is regularly spaced. If not, consider resampling or interpolation to get a constant frequency.
- Model Order Selection: To guide model order selection, use statistical tests such as the Augmented Dickey-Fuller (ADF) test and visual evaluation of autocorrelation and partial autocorrelation functions. If historical data is scarce, examine alternate models or models using a portion of the available data. Consider the ramifications of this assumption and, if required, seek more flexible models.
- Parameter Sensitivity of the Model: Conduct sensitivity analysis to determine how parameter selections affect model performance and stability.
- External Variables: Think about including external variables or looking into more complex models that can handle external variables.

3.4. DESIGN FLOW

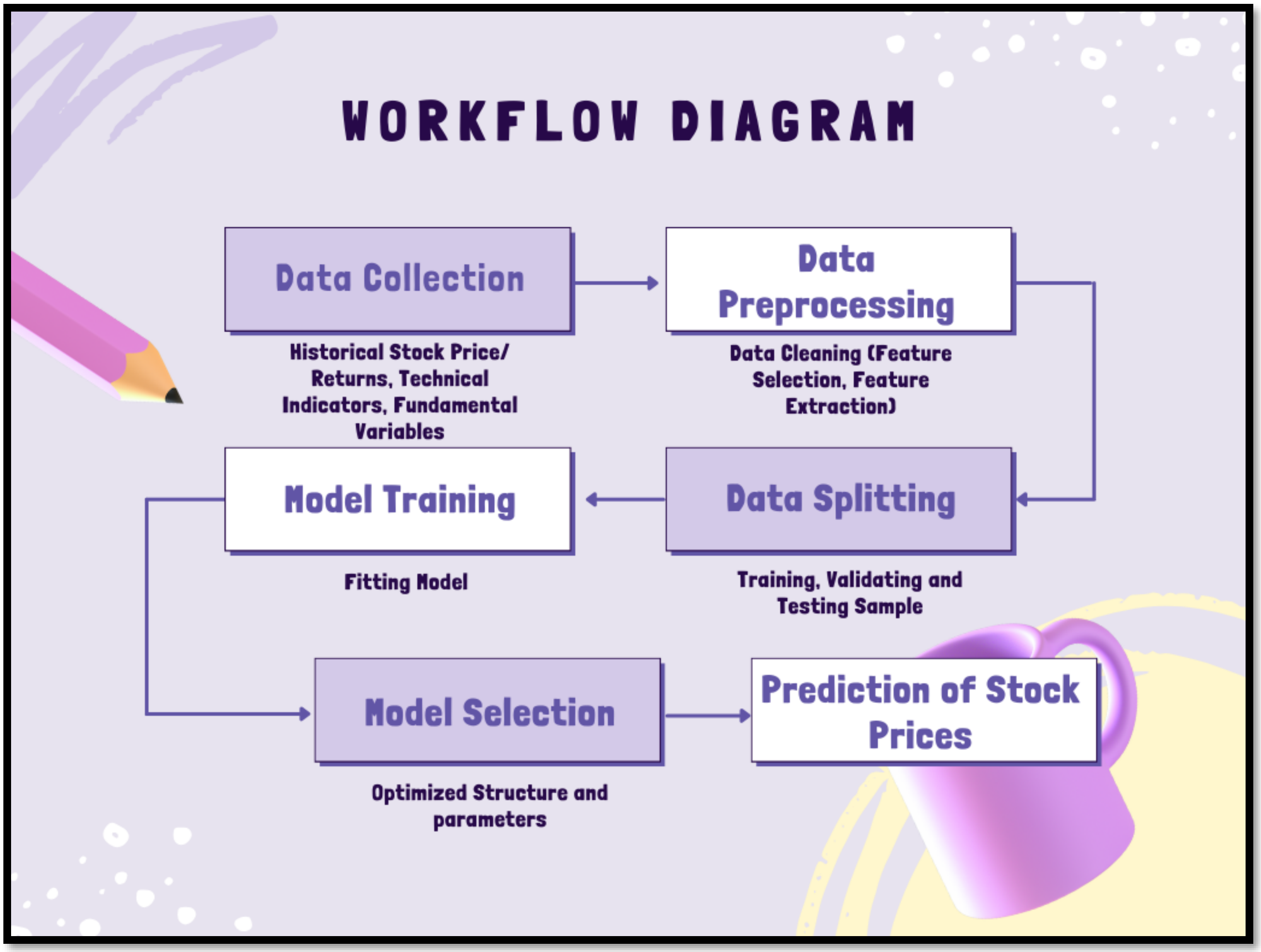


Figure 4

3.5. DESIGN SELECTION

The ARIMA (AutoRegressive Integrated Moving Average) model excels in detecting patterns in time series data, notably in the context of stock prices. Its applicability is due to the following factors:

- Handling Trends and Seasonality: ARIMA uses differencing to handle trends, ensuring that the data remains stable and the model can capture underlying patterns. Furthermore, seasonal ARIMA (SARIMA) expands its capabilities to account for repeating trends, such as seasonality in stock prices.
- Components of Autoregressive and Moving Average: The autoregressive component evaluates the effect of previous values on current values, reflecting

the historical dependency commonly found in stock prices. The moving average component filters out short-term volatility, allowing long-term patterns to be seen.

- Adaptability of parameters: ARIMA's parameter selection flexibility (p, d, q) enables for customization to unique time series features.
- Simplicity and interpretability: ARIMA's simplicity improves interpretability, making it an appealing option for stakeholders. This trait is useful when expressing and interpreting model results, especially in financial circumstances.

Considerations for restrictions and assumptions include:

- Stationarity Assumption: ARIMA presupposes stationarity, which means that the statistical features of the time series stay constant across time. Achieving stationarity is critical, and differencing is used to resolve non-stationarity in stock prices.
- External Factors: ARIMA may not account for external factors impacting stock prices, such as economic events or geopolitical developments. Users should be aware of the model's limits in capturing unplanned occurrences.

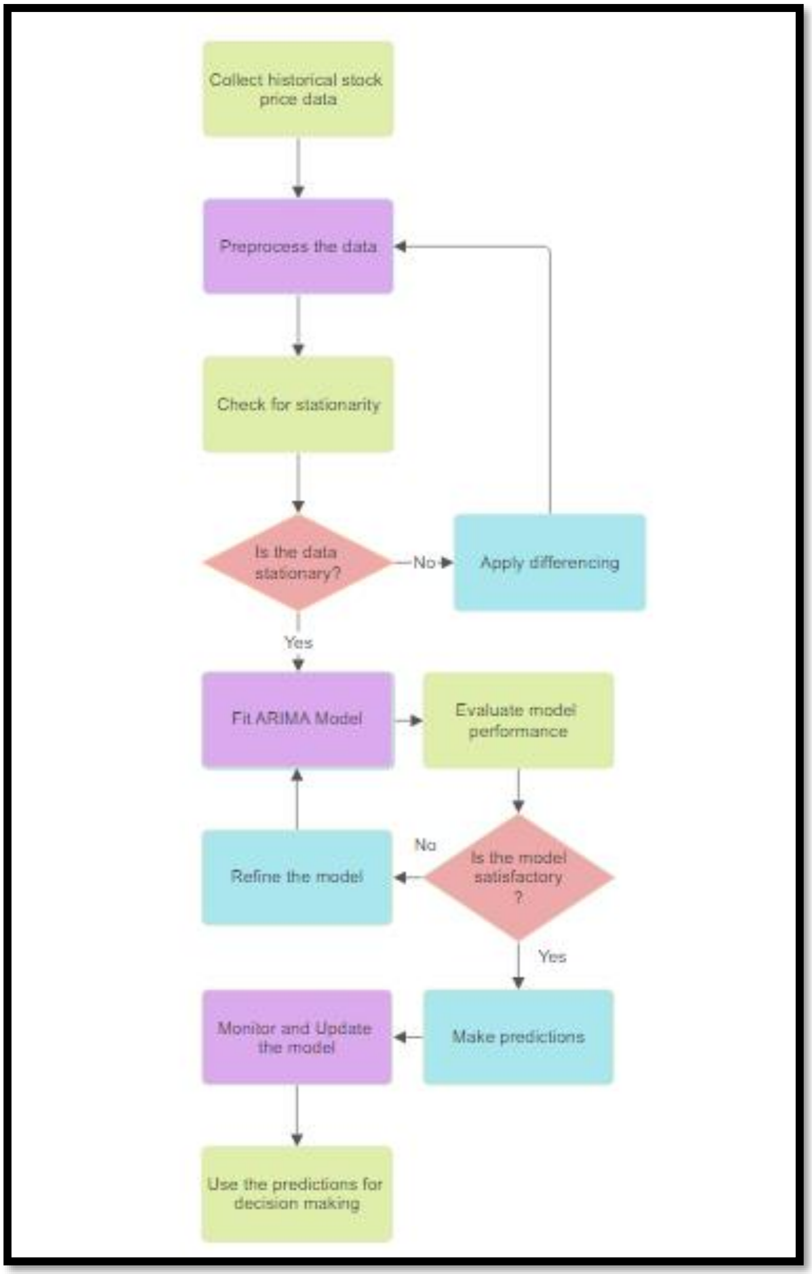


Figure 5

3.6. IMPLEMENTATION PLAN/METHODOLOGY

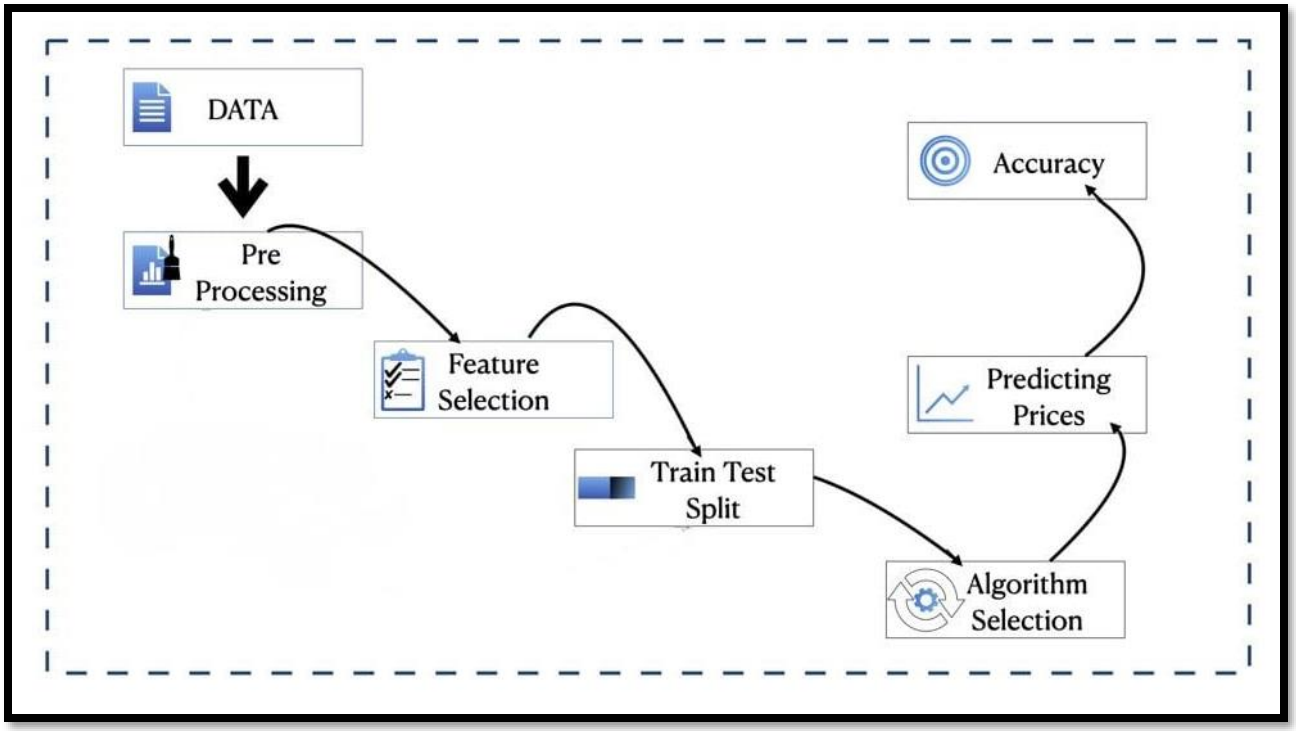


Figure 6

LINEAR REGRESSION IN DEPLOYING THE MODEL:

Linear regression is the most fundamental machine learning approach that can be applied to a set of data. An equation indicating the relationship between the independent and dependent variables is returned by the linear regression model.

$$Y = \theta 0 + \theta 1X1 + \theta 2X2 + ..\theta nXn$$

The following is the formula for linear regression: the weights are represented by the coefficients θ1, θ2, …. θn, while the independent variables are represented by x1, x2,….xn.

DEPLOYING OF ARIMA(Auto-Regressive Integrated MovingAverage) MODEL:

The most preferred statistical technique for time series forecasting is ARIMA. In order to forecast future values, ARIMA models consider historical values. Three crucial ARIMA parameters are as follows:

- p (previous values applied to the upcoming value forecast)
- q (the future values are predicted using the past forecast mistakes)

- d (the differencing order)

ARIMA parameter tweaking takes a long time. Thus, we'll employ auto ARIMA, which chooses the optimal set of (p,q,d) values that produces the least amount of error automatically.

$$Yt = c + \emptyset 1\, Yt - 1 + \emptyset 2\, Yt - 2 + \theta 1\, et - 1 + \ \theta 2\, et - 2 + \ \theta 3\, et - 3 + et$$

The time series value at time t is denoted by Yt. The word c is constant. The autoregressive coefficients are ϕ1, ϕ2. At time t, et is the white noise error term. The moving average coefficients are θ1, θ2, and θ3. Three moving average terms (q=3) and two autoregressive components (p=2) are present in our model of differenced time series (d=1). During the model fitting phase, the values of the parameters (c,ϕ1, ϕ2,θ1, θ2, θ3) would be computed using the historical data that is now available.

CHAPTER 4

RESULTS ANALYSIS AND VALIDATION

4.1. CODE

```
#!pip install chart-studio
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import plot
from plotly.offline import download_plotlyjs,
init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)

df=pd.read_csv("ADANIPORTS.CSV")

df.head()

df.info()

adani=df.loc[:,["Date","Open","High","Low","Close","Volum
e"]]

adani.head()

adani['Date']=pd.to_datetime(adani['Date'])

print(f'range of date {adani.Date.min()}
{adani.Date.max()}')
print(f'total days={(adani.Date.max()-
adani.Date.min()).days} days')

adani.describe()

adani.boxplot(column=['Open','Close','High','Low'],grid=T
rue)
```

```
layout = go. Layout (
    title='Stock Prices of adani',
    xaxis=dict (
        title= 'Date',
        titlefont=dict(
        family='Courier New, monospace',
            size=18,

        )
    ),
    yaxis=dict(
       title='Price',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,

        )
    )
)

adani_d=[{'x':adani['Date'],'y':adani['Close']}]
plot=go.Figure(data=adani_d,layout=layout)

iplot(plot)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

X=np.array(adani.index).reshape(-1,1)
y=adani["Close"]
X_train,X_test,y_train,y_test=train_test_split(X,y,test_s
ize=0.2,random_state=101)

scaler=StandardScaler().fit(X_train)

lr =LinearRegression()

lr.fit(X_train,y_train)

prediction =lr.predict(X_test)
```

```
plt.grid(True)
plt.xlabel('Date')
plt.ylabel('Close Prices')
plt.plot(adani['Date'],adani['Close'])
plt.plot(X_test,prediction )
plt.title('Adani closing price')
plt.show()

from sklearn.metrics import mean_squared_error,
mean_absolute_error,r2_score

print('mean_squared_error : ', mean_squared_error(y_test,
prediction))
print('r2 : ', r2_score(y_test, prediction))

#!pip install pmdarima
import os
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA
from pmdarima.arima import auto_arima
from sklearn.metrics import mean_squared_error,
mean_absolute_error
import math

plt.figure(figsize=(10,6))
plt.grid(True)
plt.xlabel('Date')
plt.ylabel('Close Prices')
plt.plot(adani['Date'],adani['Close'])
plt.title('Adani closing price')
plt.show()

#Distribution of the dataset
df_close = adani['Close']
df_close.plot(kind='kde')
```

```
#Test for staionarity
def test_stationarity(timeseries):
    #Determing rolling statistics
    rolmean = timeseries.rolling(14).mean()
    rolstd = timeseries.rolling(14).std()

    #Plot rolling statistics:
    plt.plot(timeseries, color='blue',label='Original')
    plt.plot(rolmean, color='red', label='Rolling Mean')
    plt.plot(rolstd, color='black', label = 'Rolling
Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean and Standard Deviation')
    plt.show(block=False)
    print("Results of dickey fuller test")
    adft = adfuller(timeseries,autolag='AIC')
    # output for dft will give us without defining what
the values are.
    #hence we manually write what values does it explains
using a for loop
    output = pd.Series(adft[0:4],index=['Test
Statistics','p-value','No. of lags used','Number of
observations used'])
    for key,values in adft[4].items():
        output['critical value (%s)'%key] =  values
    print(output)
test_stationarity(df_close)

from pylab import rcParams
rcParams['figure.figsize'] = 10, 6
df_log = np.log(adani['Close'])
moving_avg = df_log.rolling(12).mean()
std_dev = df_log.rolling(12).std()
plt.legend(loc='best')
plt.title('Moving Average')
x=adani['Date']
plt.plot(x,std_dev, color ="black", label = "Standard
Deviation")
plt.plot(x,moving_avg, color="red", label = "Mean")
plt.legend()
plt.show()
```

```
#split data into train and training set
train_data, test_data = df_log[1:int(len(df_log)*0.9)],
df_log[int(len(df_log)*0.9):]
plt.figure(figsize=(10,6))
plt.grid(True)


plt.plot(df_log, color='green', label='Train data')
plt.plot(test_data,color= 'blue', label='Test data')
plt.legend()
test_data

model_autoARIMA = auto_arima(train_data, start_p=0,
start_q=0,
                      test='adf',       # use adftest to
find optimal 'd'
                      max_p=4, max_q=4, # maximum p and q
                      m=10,              # frequency of
series
                      d=None,           # let model
determine 'd'
                      seasonal=False,   # No Seasonality
                      start_P=0,
                      D=0,
                      trace=True,
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True)
print(model_autoARIMA.summary())
model_autoARIMA.plot_diagnostics(figsize=(15,8))
plt.show()

import statsmodels.api as sm
#Modeling
# Build Model
model = sm.tsa.arima.ARIMA(train_data, order=(2,1,3))
fitted = model.fit()
print(fitted.summary())

result=fitted.forecast(333, alpha=0.05)
```

```
conf_ins = fitted.get_forecast(333).summary_frame()
hist=2989
# Plot
plt.figure(figsize=(10,5), dpi=100)
plt.plot(train_data, label='training data')
plt.plot(test_data, color = 'blue', label='Actual Stock
Price')
plt.plot(result, color = 'orange',label='Predicted Stock
Price')
plt.fill_between(test_data.index,conf_ins['mean_ci_lower'
],conf_ins['mean_ci_upper'],color='k', alpha=.10)

plt.title(' Adani Ports Stock Price Prediction')
plt.xlabel('Days')
plt.ylabel(' Stock Price')
plt.legend(loc='upper left', fontsize=8)
plt.show()

mse = mean_squared_error(test_data, result)
print('MSE: '+str(mse))
mae = mean_absolute_error(test_data, result)
print('MAE: '+str(mae))
rmse = math.sqrt(mean_squared_error(test_data, result))
print('RMSE: '+str(rmse))
mape = np.mean(np.abs(result -
test_data)/np.abs(test_data))
print('MAPE: '+str(mape))
```

4.2. IMPLEMENTATION OF SOLUTION

Our dataset was obtained from Adani Ports stocks on Kaggle. It has 15 columns overall 3322 rows of which we have used 5 called open, close, low, high, and date columns. Following that, we used the log function to normalize the data and attempted to forecast the closing stock price for a specific day. After that, the data was divided into testing and training cases. 333 rows were utilized to validate the ARIMA model's data, whereas 2989 rows were used for training.

Evaluation metrics make use of mean squared error. The main goal of this model is to predict the closing price on a given date almost exactly based on the moving average's prior closing price.

MSE is a popular statistic for assessing the performance of a prediction model, particularly ARIMA models. MSE is calculated as the average squared difference between anticipated and actual values. Here's how MSE is useful in time series forecasting using ARIMA:

- Model Development: The parameters (p, d, q) for fitting an ARIMA model to historical time series data are determined to minimize the MSE on a training dataset. This approach entails determining the ARIMA order combination that best reflects the patterns in the training data.
- Evaluation of the Model: It's critical to test the ARIMA model's performance on unseen data once it's been trained. This is frequently accomplished by making predictions on a test dataset and comparing them to actual values using metrics like MSE.
- MSE Determination: The MSE for ARIMA-based time series forecasting is determined as the average of the squared discrepancies between the projected and actual values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Yi - \widehat{Yi}\right)^2$$

⇒ Yi is the actual value at time i,

⇒ $\widehat{Yi}$ is the predicted value at time i,

⇒ N is the number of observations.

- Interpretation: A lower MSE suggests that the predictions are more accurate, implying a better-fitting model. However, MSE must be interpreted within the context of the particular time series and situation at hand.
- Model Comparisons: MSE quantifies the accuracy of the ARIMA model. It may be used to compare the performance of ARIMA with other forecasting approaches or to compare the performance of multiple ARIMA models with variable orders.

When our model is compared to the linear regression model, the outcome shows two values that are listed in the following table.

	Linear Regression	ARIMA
MEAN SQUARED ERROR	35472.819	0.1032

Table 1

Test Results for AIC dataset using Dickey-Fuller Test:

```
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-10764.514, Time=0.33 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-10762.808, Time=0.66 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-10762.804, Time=0.28 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-10766.346, Time=0.41 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-10760.788, Time=0.34 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 2.038 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 2988
Model:               SARIMAX(0, 1, 0)   Log Likelihood                5384.173
Date:                Thu, 02 Nov 2023   AIC                         -10766.346
Time:                        12:39:19   BIC                         -10760.344
Sample:                             0   HQIC                        -10764.186
                               - 2988
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2         0.0016   1.93e-06    825.108      0.000       0.002       0.002
===================================================================================
Ljung-Box (L1) (Q):                   0.29   Jarque-Bera (JB):         103015855.48
Prob(Q):                              0.59   Prob(JB):                         0.00
Heteroskedasticity (H):               0.12   Skew:                           -22.34
Prob(H) (two-sided):                  0.00   Kurtosis:                       911.69
===================================================================================
```

Figure 7

```
                              SARIMAX Results
==============================================================================
Dep. Variable:                  Close   No. Observations:                 2988
Model:                 ARIMA(2, 1, 1)   Log Likelihood                5384.360
Date:                Fri, 03 Nov 2023   AIC                         -10760.720
Time:                        10:36:11   BIC                         -10736.712
Sample:                             0   HQIC                        -10752.082
                               - 2988
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.0050      5.225      0.001      0.999     -10.236      10.246
ar.L2          0.0051      0.057      0.090      0.928      -0.107       0.117
ma.L1          0.0050      5.227      0.001      0.999     -10.239      10.249
sigma2         0.0016   2.44e-06    651.807      0.000       0.002       0.002
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):         103015513.19
Prob(Q):                              1.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.12   Skew:                           -22.34
Prob(H) (two-sided):                  0.00   Kurtosis:                       911.69
===================================================================================
```

Figure 8

Figure 9

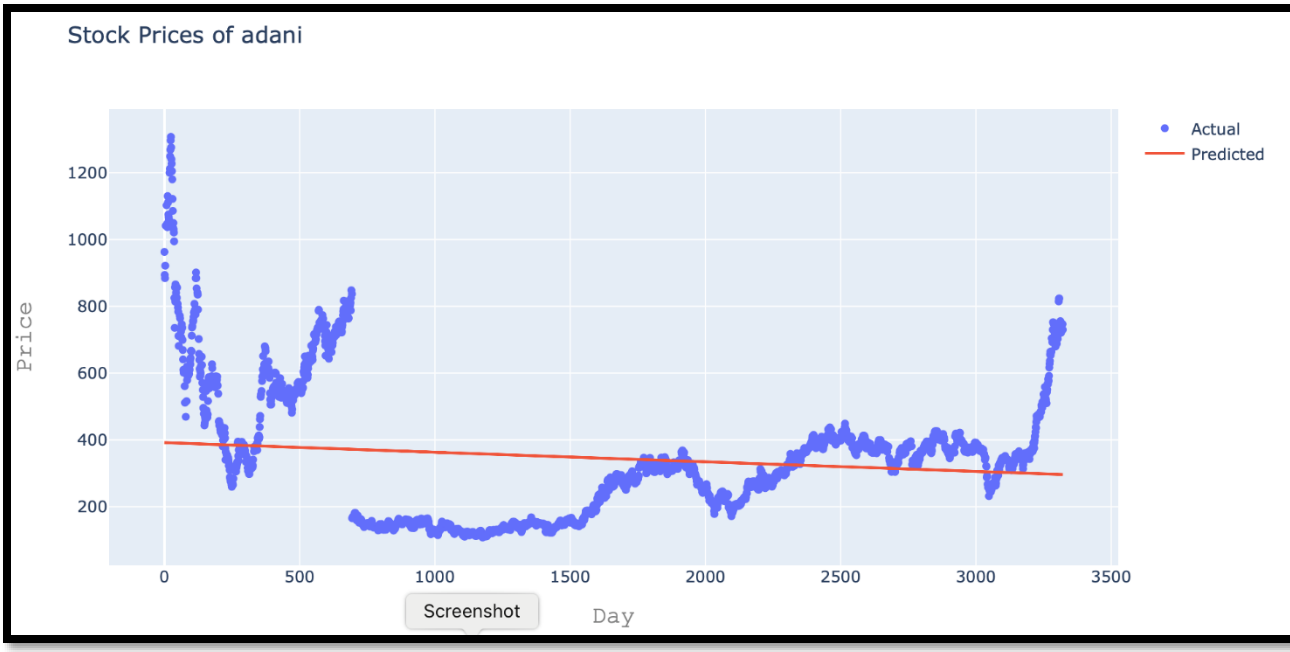


Figure 10

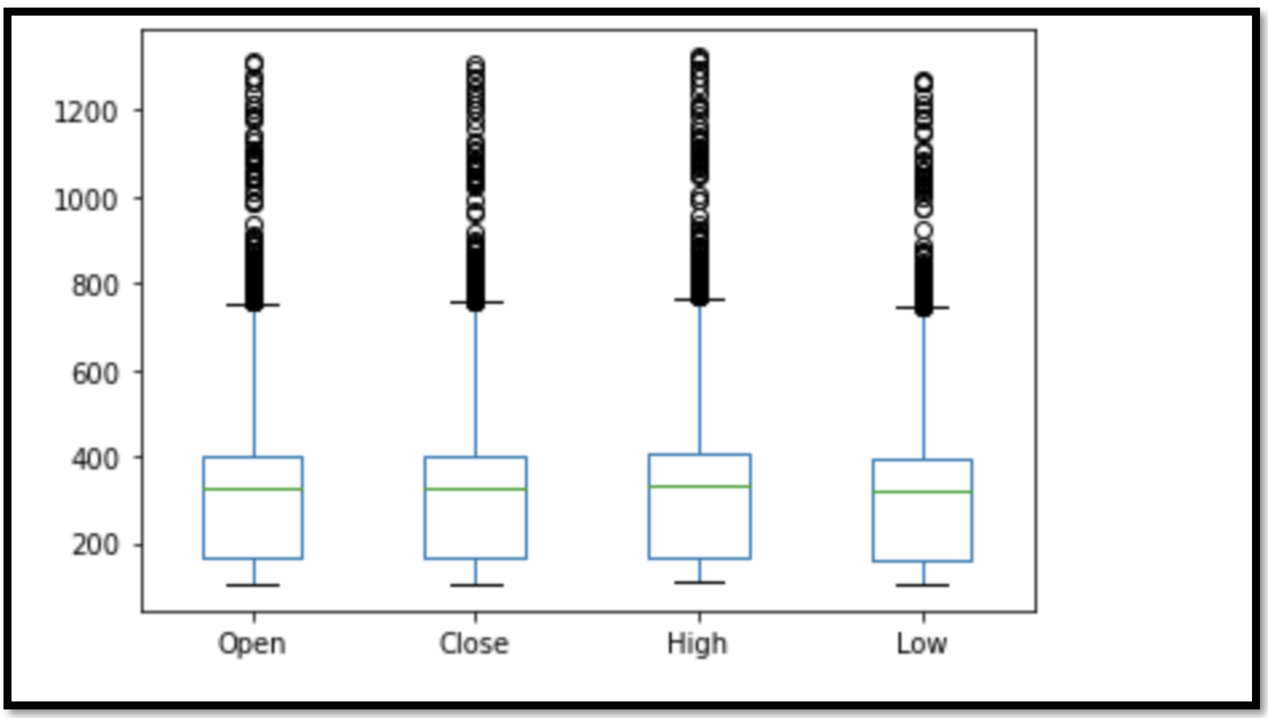


Figure 11

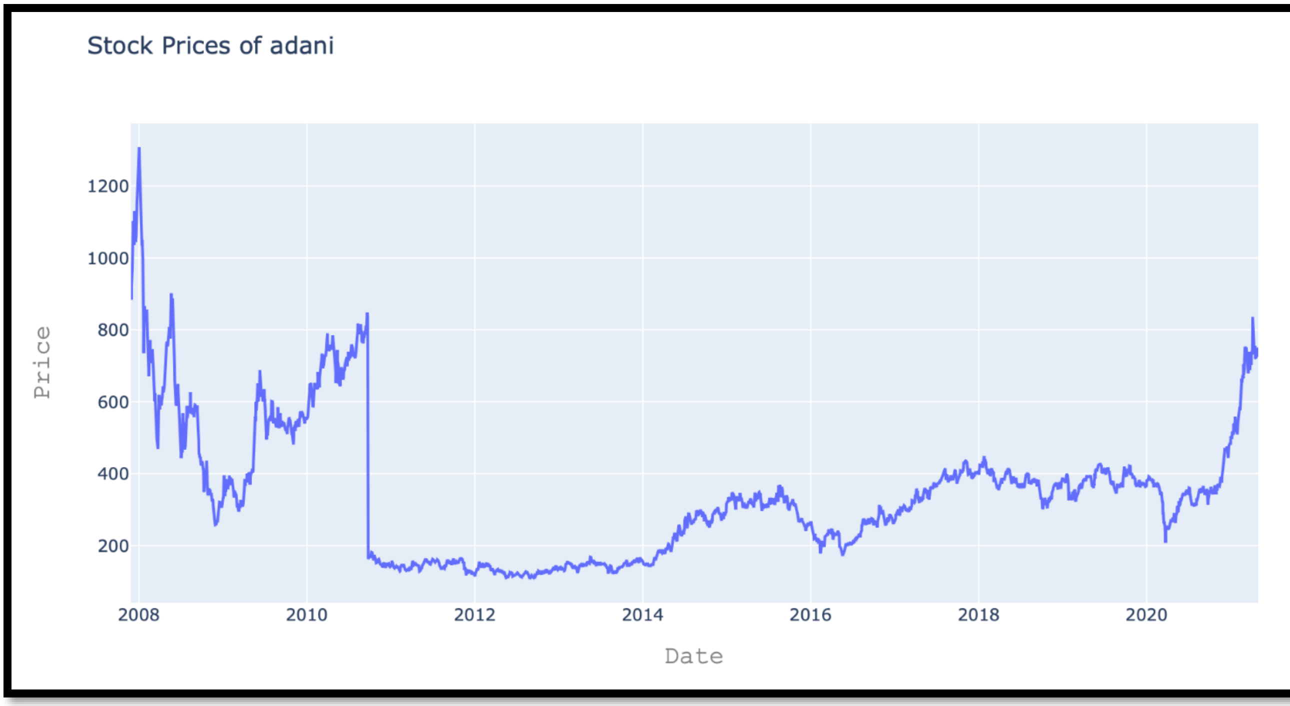


Figure 12

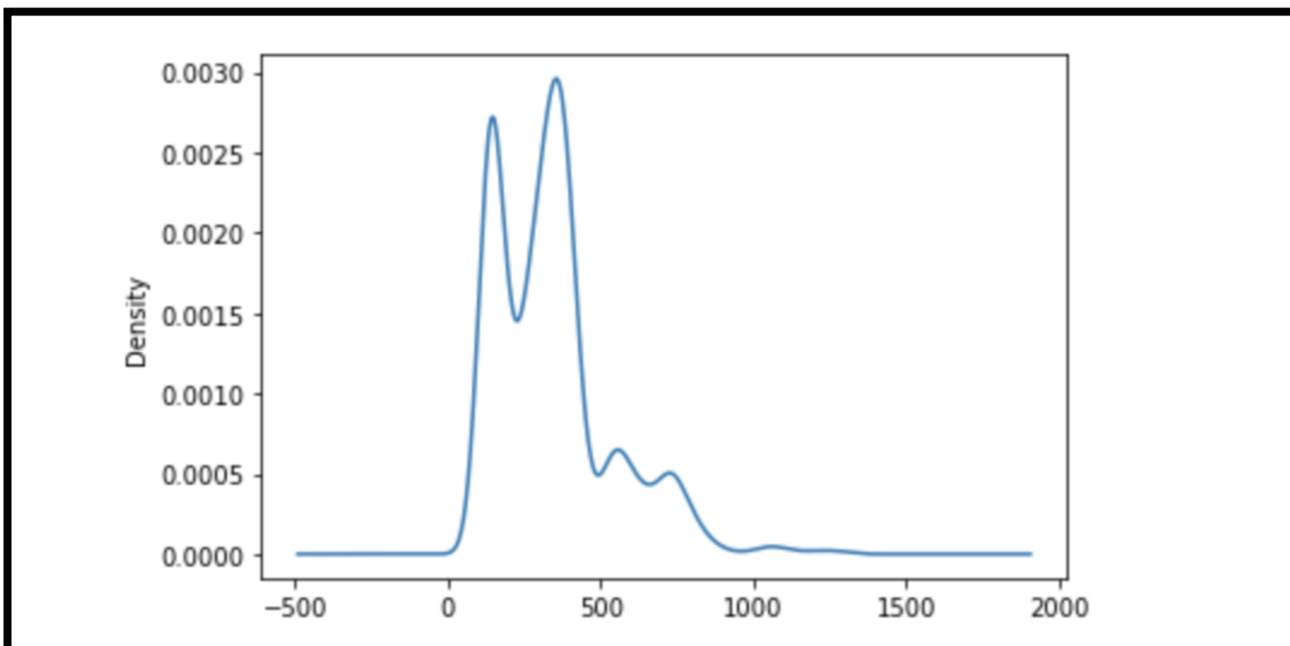
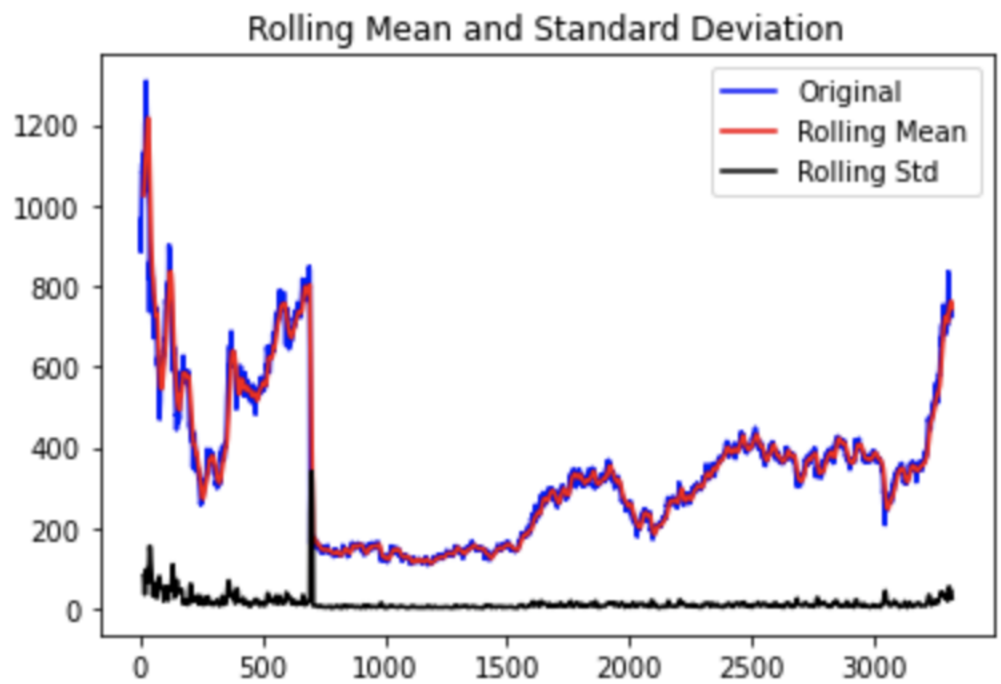


Figure 13

Test Results for a stationary dataset:




```
Results of dickey fuller test
Test Statistics                  -3.473098
p-value                           0.008705
No. of lags used                  5.000000
Number of observations used    3316.000000
critical value (1%)              -3.432324
critical value (5%)              -2.862412
critical value (10%)             -2.567234
dtype: float64
```

Figure 14

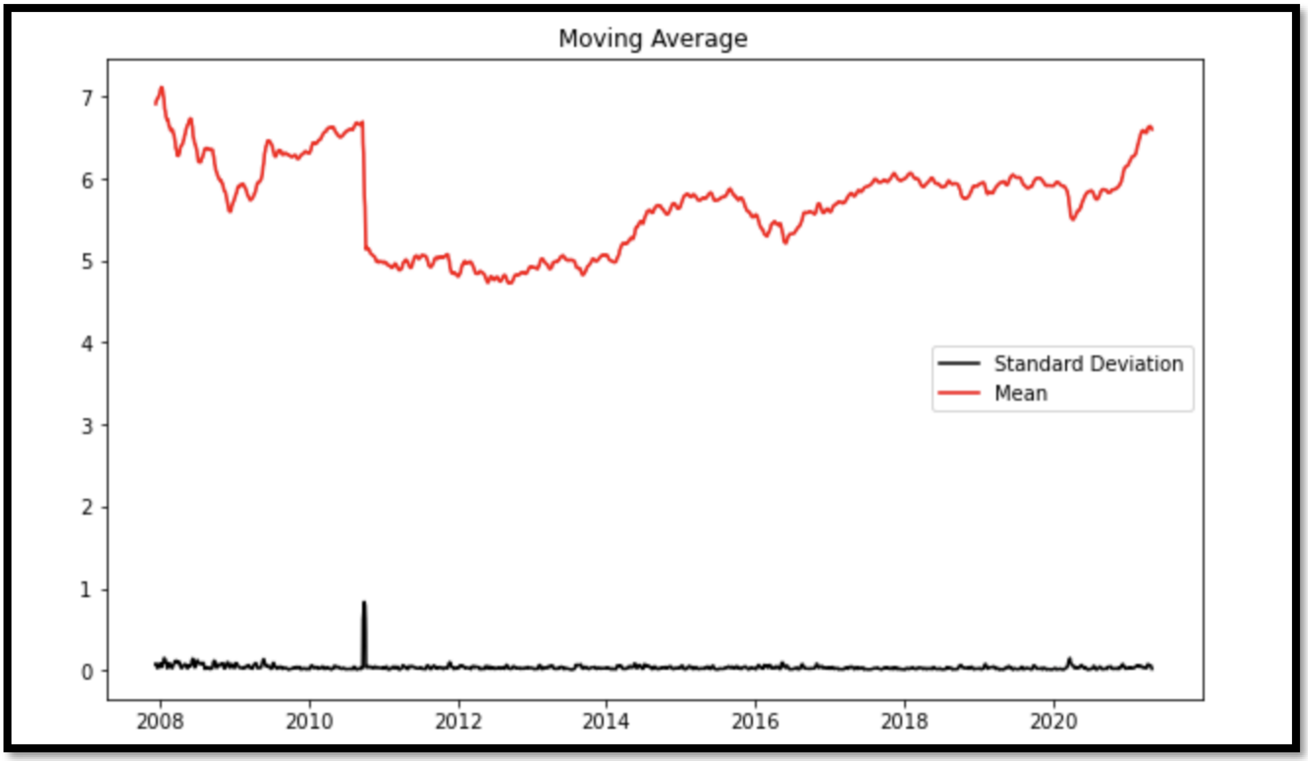


Figure 15

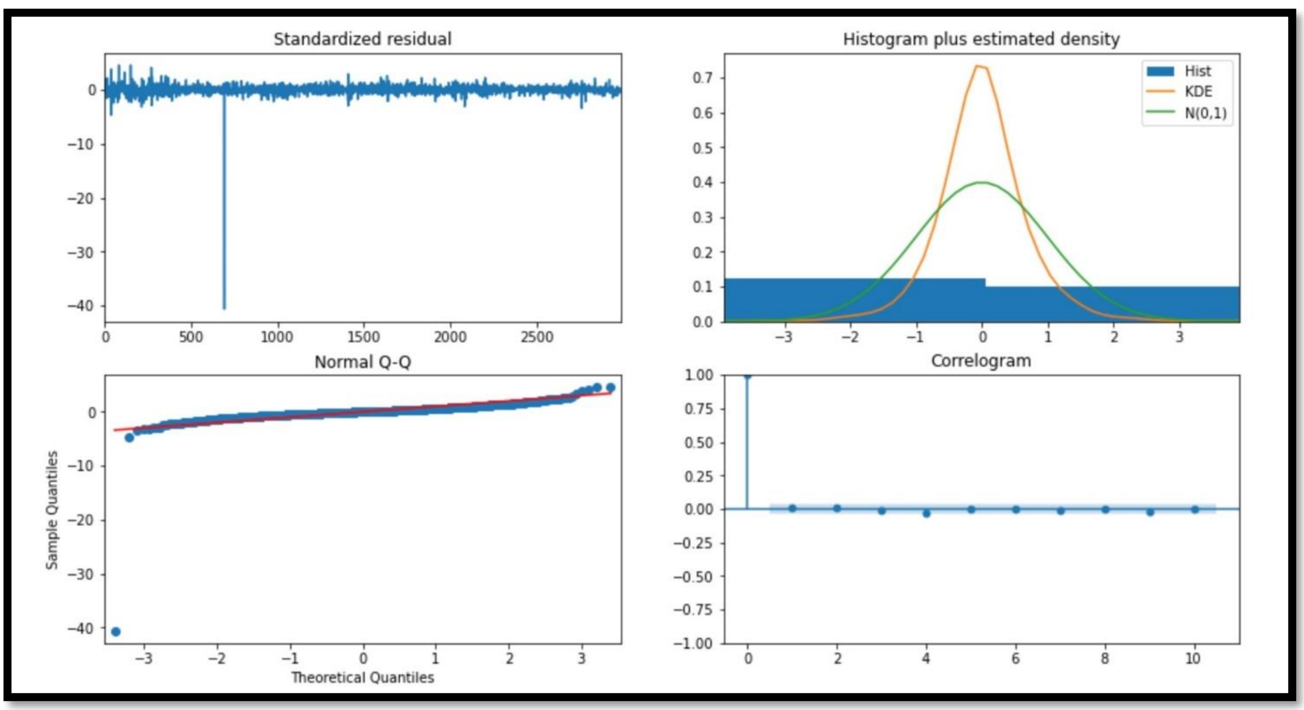


Figure 16

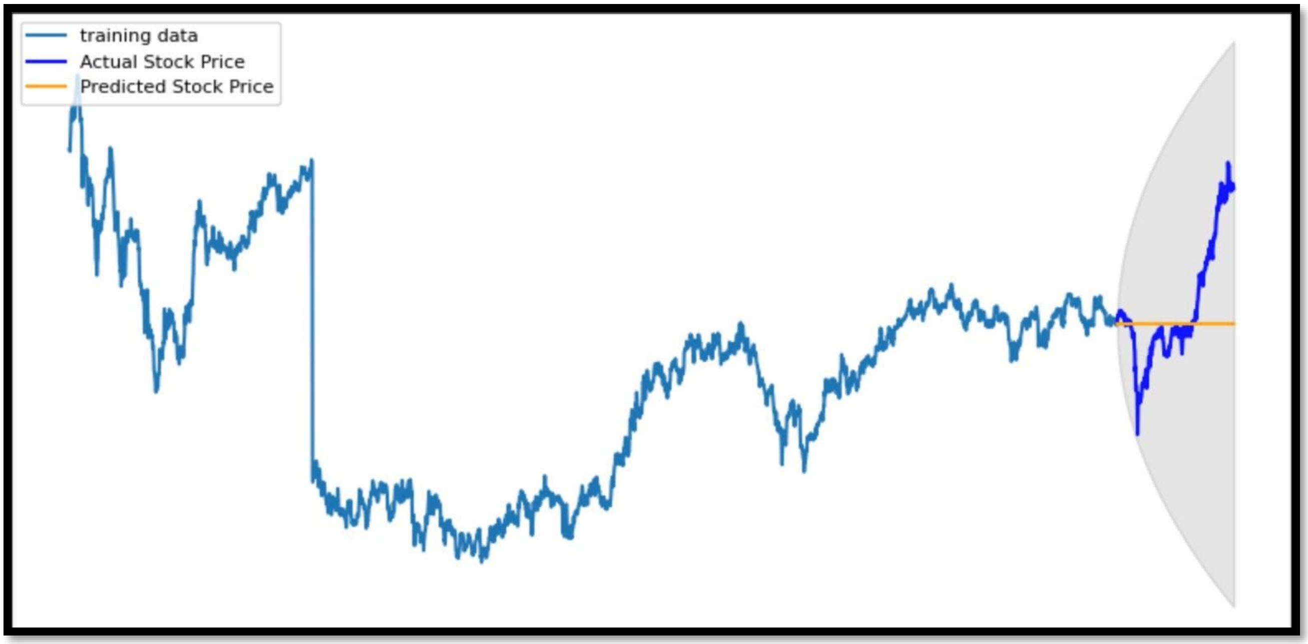


Figure 17

```
MSE: 0.10329954926128003
MAE: 0.22599451485264857
RMSE: 0.32140247239447306
MAPE: 0.036518593442856996
```

Figure 18

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1. CONCLUSION

We can infer from our research on stock price prediction using machine learning, namely linear regression and ARIMA models, that both techniques have potential and limits in terms of financial forecasting.

While linear regression is simple and easy to understand, ARIMA models provide a powerful foundation for time series analysis. This study underlines the need for a diverse approach to stock price prediction, integrating a range of machine learning algorithms and domain expertise, in order to improve forecasting accuracy. In order to fully exploit the promise of machine learning in this industry, future studies should address the complexity and uncertainty prevalent in financial markets while studying more advanced models and data sources.

The pursuit of more accurate and trustworthy stock price predictions is a continual and dynamic adventure powered by innovation and multidisciplinary cooperation.

5.2. FUTURE WORK

- The primary focus of ARIMA models is historical price data, and they may not easily take into account outside variables that can affect price movements, such as news events, economic indicators, or market sentiment.
- Incorporating sentiment analysis into a machine learning project that uses methods such as ARIMA models and linear regression is a promising direction for future research. The impact of market sentiment on stock prices can be captured by the model by integrating sentiment data from financial news, social media, and other pertinent sources. By analyzing and quantifying the sentiment portrayed in textual data, Natural Language Processing (NLP) approaches can provide another level of information to prediction models.

With regard to investor sentiment, market perception, and outside variables impacting stock movements, this sentiment-driven feature may provide insightful information. Understanding and adding the emotional context surrounding financial assets should help the model become more predictive and more able to adjust to shifting market conditions. This shift into sentiment analysis fits well with the way machine learning applications are developing in the finance industry, providing a deeper comprehension of the variables affecting stock price fluctuations.

- Normality and Stationarity: When these presumptions are broken, the accuracy and dependability of the model may suffer.

TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK

CHAPTER 1: INTRODUCTION

This chapter will cover the overview of the system.

CHAPTER 2: LITERATURE REVIEW/BACKGROUND STUDY

This chapter includes the literature available for our system and the findings of the researchers will be highlighted which will become the basis of the current implementation.

CHAPTER 3: DESIGN FLOW/PROCESS/METHODOLOGY

This chapter will cover the technical details of the proposed approach and the techniques used for the implementation.

CHAPTER 4: RESULT ANALYSIS AND VALIDATION

This chapter will include the code and output or a glimpse of the system.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

The major findings of the work will be presented in this chapter. Also, directions for extending the current study will be discussed.

REFERENCES

1. Stock Market prediction System using Machine Learning approach FaisalL Momin, Sunny Patel, Kuldeep Shinde, Prof.A.C.TASKAR BE COMPUTER ENGINEERING, Sandip Institute of Engineering and Management, Nashik, India.
2. Stock Market Forecasting Using Machine Learning Algorithms Shunrong Shen, Haomiao Jiang Department of Electrical Engineering Stanford University Tongda Zhang Department of Electrical Engineering Stanford University.
3. Stock Price Prediction Using Machine Learning and Deep Learning Frameworks Jaydip Sen Department of Analytics and Information Technology Praxis Business School Kolkata, INDIA.
4. International Conference on Computational Intelligence and Data Science (ICCIDS 2019) Stock Closing Price Prediction using Machine Learning Techniques Mehar Vijh a, Deeksha Chandolab, Vinay Anand Tikkiwalb, Arun Kumarc a Jaypee Institute Of Information Technology, Noida 201304, India bDepartment of Electronics and Communication Engineering, Jaypee Institute Of Information Technology, Noida 201304, India c Department of Computer Science and Engineering, National Institute of Technology, Rourkela, 769008, India.
5. A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing Sidra Mehtab School of Computing and Analytics NSHM Knowledge Campus Kolkata,INDIA Jaydip Sen School of Computing and Analytics NSHM Knowledge Campus Kolkata, INDIA.
6. Machine Learning Approaches in Stock Price Prediction: A Systematic Review Payal Soni 1, Yogya Tewari 1 and Prof. Deepa Krishnan 1 1 Department of Computer Engineering, Mukesh Patel School of Technology Management and Engineering, NMIMS University(Deemed-to-be), Mumbai, India.
7. Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions Nusrat Rouf 1,

Majid Bashir Malik 2 , Tasleem Arif 3 , Sparsh Sharma 4 , Saurabh Singh 5 , Satyabrata Aich 6,* and Hee-Cheol Kim 7.

8. Momin, Faisal & Patel, Sunny & Shinde, Kuldeep &Sahane, Prof & Sabri, Mustafa. (2020). Stock Market Prediction System Using Machine Learning Approach. SSRN Electronic Journal. 7. 190-194.

9. Vanukuru, Kranthi. (2018). Stock Market Prediction Using Machine Learning. 10.13140/RG.2.2.12300.77448.

10. Shen, Shunrong, Haomiao Jiang, and Tongda Zhang. “Stock Market Forecasting Using Machine Learning Algorithms.” (2012).

11. Sen, Jaydip. (2018). Stock Price Prediction Using Machine Learning and Deep Learning Frameworks.

12. Vijh, Mehar Chandola, Deeksha Tikkiwal, Vinay Kumar, Arun. (2020). Stock Closing Price Prediction using Machine Learning Techniques. Procedia Computer Science. 167. 599-606. 10.1016/j.procs.2020.03.326.

13. Mehtab, Sidra Sen, Jaydip. (2019). A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing. SSRN Electronic Journal. 10.2139/ssrn.3502624.

14. Soni, Payal Tewari, Yogya Krishnan, Deepa. (2022). Machine Learning Approaches in Stock Price Prediction: A Systematic Review. Journal of Physics: Conference Series. 2161. 012065. 10.1088/1742- 6596/2161/1/012065.

15. Rouf, Nusrat Malik, Majid Arif, Tasleem Sharma, Sparsh Singh, Saurabh Aich, Satyabrata Kim, Hee-Cheol. (2021). Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions. Electronics. 10. 2717. 10.3390/electronics10212717.

16. Singh, Shubha Gutta, Sreedevi Hadaegh, Ahmad. (2021). Stock Prediction Using Machine Learning. WSEAS TRANSACTIONS ON COMPUTER RESEARCH. 9. 152-158. 10.37394/232018.2021.9.17.

17. Deshmukh, Yogita Saratkar, Deepmala Tiwari, Yash. (2019). Stock Market Prediction Using Machine Learning. IJARCCE. 8. 31-35. 10.17148/IJARCCE.2019.8107.

18. C, Karthikeyan A, Sahaya Perumal, Anandan Prabha, Ramadoss D, Mohan Babu D, Vijendra. (2021). Predicting Stock Prices Using Machine Learning Techniques. 1-5. 10.1109/ICICT50816.2021.9358537.

19. Amineni, Sriya. (2023). Using Machine Learning To Predict Future Stock Prices. Journal of Student Research. 12. 10.47611/jsrhs.v12i2.4439.

20. Sahoo, Prasanta Charlapally, Mr. (2019). Stock Price Prediction Using Regression Analysis.

21. Chakraborty, Sanjay Kairi, Animesh Roy, Nilanjana Maity, Saikat Pandey, Saroj De, Indrajit. (2023). Market-Based Stock Allocation Using a Hybrid Regression Model. SN Computer Science. 4. 10.1007/s42979-023-01883-1.

22. Yu, Minhui. (2022). Linear Regression Model for Stock Price of Pfizer. 10.1007/978-981-19-0564-354.

23. Adebiyi, Ayodele Adewumi, Aderemi Ayo, Charles. (2014). Stock price prediction using the ARIMA model. Proceedings - UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, UKSim 2014. 10.1109/UKSim.2014.67.

24. Quadri, S. Fatima, Sumayya. (2023). EMBOT: STOCK PREDICTION, CRYPTO PREDICTION AND ROBOT TRADING SYSTEM USING ML AND RPA. 38. 182-189. 10.5281/zenodo.7628822.

25. Kara, Yakup Boyacioglu, Melek Baykan, Omer. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Systems with Applications. 38. 5311-5319. 10.1016/j.eswa.2010.10.027.

26. Ayyıldız, Nazif ˙Iskenderoglu, ˘ Omer. (2023). Prediction Of Stock Index ¨ Movement Using Machine Learning Methods: An Application On Bist 100 Index.

27. Li, Hongzheng Huang, Shaohang. (2021). Research on the Prediction Method of Stock Price Based on RBF Neural Network Optimization Algorithm. E3S Web of Conferences. 235. 03088. 10.1051/e3sconf/202123503088.

28. Kaur, Ranjeet Sharma, Dr. Yogesh Bhatt, Devershi. (2021). Measuring Accuracy of Stock Price Prediction Using Machine Learning Based Classifiers. IOP Conference Series: Materials Science and Engineering. 1099. 012049. 10.1088/1757-899X/1099/1/012049.

29. Tang, Xiaobin Lei, Nuo Dong, Manru Ma, Dan. (2022). Stock Price Prediction Based on Natural Language Processing1. Complexity. 2022. 1-15. 10.1155/2022/9031900.

30. Waduge, Nisal Ganegoda, Upeksha. (2018). Forecasting Stock Price of a Company Considering Macroeconomic Effect from News Events. 1-5. 10.1109/ICITR.2018.8736133.

31. Elbousty, H. Salah-ddine, Krit. (2021). Stock Market Forecasting Model From Multi News Data Source Using a Two-Level Learning Algorithm. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 12. 1022-1030. 10.17762/turcomat.v12i5.1746.

32. Ding, Guangyu Qin, Liangxi. (2020). Study on the prediction of stock price based on the associated network model of LSTM. International Journal of Machine Learning and Cybernetics. 11. 10.1007/s13042-019- 01041-1.

33. Vargas, Manuel Anjos, Carlos Bichara, Gustavo Evsukoff, Alexandre. (2018). Deep Learning for Stock Market Prediction Using Technical Indicators and Financial News Articles. 1-8. 10.1109/IJCNN.2018.8489208.

34. Roondiwala, Murtaza Patel, Harshal Varma, Shraddha. (2017). Predicting Stock Prices Using LSTM. International Journal of Science and Research (IJSR). 6. 10.21275/ART20172755.

35. Juairiah, Fatima Mahatabe, Mostafa Jamal, Hasan Shiddika, Aysha Shawon, Md Mandal, Nibir Chandra. (2022). Stock Price Prediction: A Time Series Analysis.