

STOCK MARKET PRICE PREDICTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

SATHYASEELAN S	[513220205309]
ANANTH A	[513220205301]
SURENDHAR K	[513220205310]
ANISHKUMAR T	[513220205302]

*In partial fulfillment for the award of the degree
of*

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY



THIRUMALAI ENGINEERING COLLEGE, KANCHIPURAM

ANNA UNIVERSITY: CHENNAI – 600 025

MAY 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE



Certificate that this project report titled **“STOCK MARKET PRICE PREDICTION USING MACHINE LEARNING”** is the bonafide work of **“SATHYASEELAN S [513220205309], ANANTH A [513220205301], ANISHKUMAR T [513220205302], SURENDHAR K [513220205310]”** who Carried out the project work under my supervision.

SIGNATURE OF HOD

V. VIJAYABHASKAR M.C.A., M.Tech.,

HEAD OF THE DEPARTMENT,

Associate Professor,

Department of IT,

Thirumalai Engineering College,

Kanchipuram – 631 551.

SIGNATURE OF SUPERVISOR

A. PRIYANKA M.E.,

SUPERVISOR,

Assistant Professor,

Department of IT,

Thirumalai Engineering College,

Kanchipuram – 631 551.

Submitted for the Project Viva Voce held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I profoundly thank our **Chairman and trust members of Kanchipuram Educational Trust** for providing adequate facilities.

I would like to express my hearty thanks to our respectable Principal. Incharge **Mr.T.MohanRaj M.Tech.**, for allowing us to have the extensive use of our colleges facilities to our colleges facilities to have precious advice regarding the project.

I extend our thanks to Associate Professor **Mr.V.VIJAYABHASKAR M.C.A., M.Tech., Head of the Department, Information Technology** for this precious advice regarding the project.

I would like to express my deep and unbounded gratefulness to my project Guide **Mrs.A.PRIYANKA M.E.**, Department of Information Technology, for his valuable guidance and encouragement throughout the project. He has been a constant source of inspiration and has provided the precious suggestion throughout this project.

I thank all facilities and supporting staff for the help they extended in completing this project. I also express my sincere thanks to our parents, and all my friends for their continuous support.

TABLE OF CONTENTS

CHAPTER NO	LIST OF CONTENT	PAGE NO
	ABSTRACT	I
	LIST OF ABBREVIATION	II
	LIST OF FIGURES	III
	LIST OF TABLES	IV
1	INTRODUCTION	
	1.1 INTRODUCTION	1
	1.2 PROBLEM STATEMENT	2
	1.3 OBJECTIVE OF THE PROJECT	3
	1.4 STOCK MARKET	4
	1.5 SCOPE OF PROJECT	5
2	LITERATURE SURVEY	
	2.1 PAPER 1	7
	2.2 PAPER 2	7
	2.3 PAPER 3	8
	2.4 PAPER 4	9
	2.5 PAPER 5	9
	2.6 PAPER 6	10
	2.7 PAPER 7	10
	2.8 PAPER 8	11
	2.9 PAPER 9	12
	2.10 PAPER 10	13
3	PROPOSED METHODOLOGY	
	3.1 EXISTING SYSTEM	14
	3.2 PROPOSED SYSTEM	15

	3.3 PROPOSED TECHNIQUES	15
	3.4 PYTHON IN DATA SCIENCE	17
	3.5 TRAINING	20
	3.5.1 TRAINING DATASET	20
	3.6 TESTING	27
	3.6.1 TESTING DATASET	27
	3.7 SPLITTING, IMPUTATION AND INTERPOLATION	24
	3.8 DATA FRAMES	24
4	INTRODUCTION OF MACHINE LEARNING	
	4.1 GENERAL	26
	4.2 OVERVIEW OF MACHINE LEARNING	27
	4.3 MACHINE LEARNING-BASED APPROCHES	28
	4.3.1 DENSITY BASED DETECTION OF ANOMALY	28
	4.3.2 CLUSTERING BASED DETECTION OF ANOMALY	28
	4.3.3 SVM BASED DETECTION OF ANOMALY	29
	4.4 DATASET	29
5	SYSTEM REQUIREMENTS	
	5.1 GENERAL	31
	5.2 HARDWARE REQUIREMENTS	31
	5.3 SOFTWARE REQUIREMENTS	32
	5.4 SOFTWARE USED	32
	5.5 PYTHON PACKAGES	32
	5.5.1 NUMPY	33

	5.5.2 PANDAS	33
	5.5.3 MATPLOTLIB	34
	5.5.4 SEABORN	35
	5.5.5 PLOTLY	36
	5.6 JUPYTER NOTEBOOK	38
6	DESIGN ENGINEERING	
	6.1 ARCHITECTURE DIAGRAM	39
	6.2 USE CASE DIAGRAM	42
	6.3 DATA FLOW DIAGRAM	43
7	IMPLEMENTATION	
	7.1 GENERAL	44
	7.2 PROCEDURE FOLLOWED DURING IMPLEMENTATION	44
	7.2.1 DATASET DESIGN	45
	7.2.2 PREPROCESSING	48
	7.2.3 PREDICTION	49
8	SOFTWARE TESTING	
	8.1 GENERAL	50
	8.2 TESTING	50
9	ALGORITHM	
	9.1 LOGISTIC REGRESSION ALGORITHM	53
	9.1.1 ADVANTAGES OF LOGISTIC ALGORITHM	54
	9.2 KNN	56
	9.3 RANDOM FOREST CLASSIFIER	57
	9.4 DECISION TREE ALGORITHM	59

10	IMPLEMENTATION	61
11	APPLICATION AND FUTURE ENHANCEMENT	
	11.1 APPLICATION	64
	11.2 FUTURE ENHANCEMENT	66
12	CODING	68
13	OUTPUT	
	13.1 SETTING THE DATA FRAME	74
	13.2 TRAINING THE MODEL	75
	13.3 GOOGLE STOCK PREDICTION	76
	13.4 STOCK PRICE TESLA	76
	13.4 PREDICTING THE VALUES	77
14	CONCLUSION	78
15	REFERENCES	80

ABSTRACT

Stock market has been constant fascinating topic but since last few years stockholders desire to hardback return on day to day basis got glorified, then with the support of machine learning stockholders initiates stress-free approaches to squint of forthcoming market trends. This paper benevolences LSTM grounded methodology to forecast the characteristics of a specific stock, our method put greater concentration on sets of hundred days moving averages. Hundred days moving average is a technical method follow by the stock market professionals forecast the forthcoming trends of market, which represents the current as well the characteristics which is stock going to show in upcoming days. As we are scraping data with the API and creating our own dataset hence our method is comfortable with ambiguous data besides it provides output with high accuracy. The results indicate that our method attained superior upshot than other approaches.

Keywords: LSTM (long- and short-term memory), API (Application program interface), machine learning, hundred days moving average, market trends.

LIST OF ABBREVIATIONS

ACRONYM	ABBREVIATIONS
WHO	WORLD HEALTH ORGANIZATION
NLP	NATURAL LANGUAGE PROCESS
DS	DATA SCIENCE
EDA	EXPLORATORY DATA ANALYSIS
CSV	COMMA SEPERATE VALUE
KNN	K-NEAREST NEIGHBOR
ROC	RECEIVER OPERATING CHARACTER
API	APPLICATION PROGRAMMABLE INTERFACE
NOSQL	NOT ONLY SQL
VOC	VARIENCES OF CONCERN
SVM	SUPPORT VECTOR MECHINE
BDV	BIG DATA VISUALIZATION

LIST OF FIGURES

FIGURE NO	FIGURES	PAGE NO
3.1.1	DATA CLEANING	15
3.1.2	NETWORK ANALYSIS	16
3.4.1	NEURAL NETWORK	18
3.4.2	STATISTICAL ANALYSIS	18
4.3.3.1	MATPLOTLIB	35
4.3.3.2	SEABORN	36
4.3.5.1	PLOTLY	37
6.1.1	ARCHITECTURE DIAGRAM	39
6.2.1	USE CASE DIAGRAM	42
6.3.1	DFD DIAGRAM	43
7.2.1.1	DATASET	45
7.2.3.1	ACCURACY	49
8.1.1	COMPARISION CHART	52
9.1.1	KNN ALGORITHM	57
9.2.1	RANDOM FOREST CLASSIFIER	58
9.3.1	DECISION TREE ALGORITHM	60
13.1.1	SETTING THE DATA FRAME	74
13.2.1	TRAINING THE MODEL	75
13.3.1	GOOGLE STOCK PREDICTION	76
13.4.1	STOCK PRICE TESLA	77
13.5.1	PREDICTING THE VALUES	77

LIST OF TABLES

TABLE NO	TABLES	PAGE NO
8.1.1	TABULATED RESULTS	50

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening new business or the need of high salary career.

Stock markets are affected by many factors causing the uncertainty and high volatility in the market. Although humans can take orders and submit them to the market, automated trading systems (ATS) that are operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human.

However, to evaluate and control the performance of ATSS, the implementation of risk strategies and safety measures applied based on human judgments are required. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analysed.

Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many time series prediction algorithms have shown their effectiveness in practice.

The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type - Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Stock market is a typical area that presents time-series data and many researchers study on it and proposed various models. In this project, LSTM model is used to predict the stock price.

Preparing the highlights of records makes the knowledgeable model. The manner towards shutting inventory price forecast is depicted with inside the accompanying phase and the one of a kind exam did to discover the presentation of the fashions are demonstrated. The direct linear regression and support vector relapse calculations are carried out for making ready the dataset and foresee the destiny inventory cost.

Supervised learning is frequently delineated as assignment organized on these lines. it's deeply targeted around a selected assignment, taking care of associate ever increasing range of guides to the calculation till it will exactly perform on its task.

This is often the educational kind that you just can little doubt expertise. Stock marketplace prediction is largely characterized as trying to determine the inventory really well worth and provide a lively concept for the people to recognize and assume the marketplace and the inventory costs.

It is via way of means of and huge added using the quarterly financial percentage using the dataset. In this way, relying on a solitary dataset might not be good enough for the forecast and might supply a final result that is off base.

1.2 PROBLEM STATEMENTS

In today's dynamic financial landscape, investors are constantly seeking ways to make informed decisions amidst the inherent volatility of the stock market. Traditional methods of stock market analysis often fall short in accurately predicting price movements, leaving investors vulnerable to significant risks and missed opportunities. Thus, there is a growing demand for more sophisticated and data-driven approaches to stock market prediction.

This project aims to address this challenge by leveraging the power of machine learning (ML) techniques to develop predictive models capable of forecasting stock prices with a high degree of accuracy.

The primary objective is to build robust models that can effectively capture the complex patterns and relationships inherent in financial data, enabling investors to make more informed and profitable investment decisions.

1.3 OBJECTIVE

Businesses primarily run over customer's satisfaction, customer reviews about their products. Shifts in sentiment on social media have been shown to correlate with shifts in stock markets. Identifying customer grievances thereby resolving them leads to customer satisfaction as well as trustworthiness of an organization.

Hence there is a necessity of an un biased automated system to classify customer reviews regarding any problem. In today's environment where we're justifiably suffering from data overload (although this does not mean better or deeper insights), companies might have mountains of customer feedback collected; but for mere humans, it's still impossible to analyse it manually without any sort of error or bias. Oftentimes, companies with the best intentions find themselves in an insights vacuum.

You know you need insights to inform your decision making and you know that you're lacking them, but don't know how best to get them.

Sentiment analysis provides some answers into what the most important issues are, from the perspective of customers, at least.

Because sentiment analysis can be automated, decisions can be made based on a significant amount of data rather than plain intuition.

1.4 STOCK MARKET

A stock market, equity market or share market is the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or discrete entity) of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange as well as those only traded privately.

Examples of the latter include shares of private companies which are sold to investors through equity crowd funding platforms. Stock exchanges list shares of common equity as well as other security types, e.g. corporate bonds and convertible bonds. Stock price prediction is one of the most widely studied problem, attracting researchers from many fields.

The volatile nature of the stock market makes it really difficult to apply simple time-series or regression techniques. Financial institutions and active traders have created various proprietary models to beat the market for 2 themselves or their clients, but rarely did anyone achieve consistently higher than the average returns on investment.

The challenge of stock market price forecasting is so appealing because an improvement of just a few points of percentage can increase the profit by millions of dollars.

This paper discusses the application of Support Vector Machines and Linear Regression in detail along with the pros and cons of the given methods.

The paper introduces the parameters and variables which can be used to recognize the patterns in stock prices which can be helpful in future stock prediction and how boosting can be integrated with various other machine learning algorithms to improve the accuracy of our prediction systems.

1.5 SCOPE OF PROJECT

The scope of stock market prediction using machine learning encompasses a wide range of areas and activities. Here's an overview of the scope:

Data Collection and Preparation:

Gathering comprehensive datasets including historical stock prices, trading volumes, financial statements, market indices, economic indicators, and news sentiment analysis. Cleaning, preprocessing, and integrating diverse data sources to create a unified dataset suitable for analysis.

Feature Engineering:

Extracting relevant features from the raw data that have predictive power in determining future stock price movements. This involves selecting and transforming variables, creating technical indicators, incorporating market sentiment analysis, and engineering additional features that capture meaningful patterns in the data.

Model Development:

Implementing a variety of machine learning algorithms such as linear regression, decision trees, random forests, support vector machines, neural networks, and ensemble methods. Developing models capable of capturing complex relationships within the data and generating accurate predictions of stock prices.

Model Evaluation and Validation:

Assessing the performance of the trained models using appropriate evaluation metrics such as mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), accuracy, precision, recall, and F1-score. Conducting robust validation tests to ensure the models generalize well to unseen data and are not overfitting.

Deployment and Integration:

Implementing the trained models into a user-friendly interface or web application, allowing investors to access real-time predictions and insights. Integrating the predictive models into existing trading platforms, financial applications, or automated trading systems to facilitate informed decision-making and execution of trades.

CHAPTER 2

LITERATURE SURVEY

2.1 PAPER 1

1. Survey of stock market prediction using machine learning approach

Authors: Ashish Sharma ; Dinesh Bhuriya ; Upendra Singh 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)

Stock market is basically nonlinear in nature and the research on stock market is one of the most important issues in recent years. People invest in stock market based on some prediction. For predict, the stock market prices people search such methods and tools which will increase their profits, while minimize their risks. Prediction plays a very important role in stock market business which is very complicated and challenging process. Employing traditional methods like fundamental and technical analysis may not ensure the reliability of the prediction. To make predictions regression analysis is used mostly. In this paper we survey of well-known efficient regression approach to predict the stock market price from stock market data based. In future the results of multiple regression approach could be improved using more number of variables.

2.2 PAPER 2

2. Short-term prediction for opening price of stock market based on

selfadapting variant PSO-Elman neural network Authors: Ze Zhang ; Yongjun Shen ; Guidong Zhang ; Yongqiang Song ; Yan Zhu, 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)

Stock price is one of intricate non-linear dynamic system. Typically, Elman neural network is a local recurrent neural network, having one context

layer that memorizes the past states, which is quite fit for resolving time series issues. Given this, this paper takes Elman network to predict the opening price of stock market. Considering that Elman network is limited, this paper adopts self-adapting variant PSO algorithm to optimize the weights and thresholds of network. Afterwards, the optimized data, regarded as initial weight and threshold value, is given to Elman network for training, accordingly the prediction model for opening price of stock market based on self- 4 adapting variant PSO-Elman network is formed. Finally, this paper verifies that model by some stock prices, and compares with BP network and Elman network, so as to draw the result that shows the precision and stability of this predication model both are superior to the traditional neural network.

2.3 PAPER 3

3. Combining of random forest estimates using LSboost for stock market index prediction Authors: Nonita Sharma ; Akanksha Juneja,2017 2nd International Conference for Convergence in Technology (I2CT)

This research work emphases on the prediction of future stock market index values based on historical data. The experimental evaluation is based on historical data of 10 years of two indices, namely, CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex from Indian stock markets. The predictions are made for 1-10, 15, 30, and 40 days in advance. This work proposes to combine the predictions/estimates of the ensemble of trees in a Random Forest using LSboost (i.e. LS-RF). The prediction performance of the proposed model is compared with that of well-known Support Vector Regression. Technical indicators are selected as inputs to each of the prediction models. The closing value of the stock price is the predicted variable. Results show that the proposed scheme outperforms Support Vector Regression and can be applied successfully for building predictive models for stock prices prediction.

2.4 PAPER 4

4. Using social media mining technology to assist in price prediction of stock market Authors: Yaojun Wang ; Yaoqing Wang,2016 IEEE International Conference on Big Data Analysis (ICBDA)

Price prediction in stock market is considered to be one of the most difficult tasks, because of the price dynamic. Previous study found that stock price volatility in a short term is closely related to the market sentiment; especially for small-cap stocks. This paper used the social media mining technology to quantitative evaluation market segment, and in combination with other factors to predict the stock price trend in short term. Experiment results show that by using social media mining combined with other information, the stock prices prediction model can forecast 5 more accurate.

2.5 PAPER 5

5. Stock market prediction using an improved training algorithm of neural network Authors: Mustain Billah ; Sajjad Waheed ; Abu Hanifa,2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)

Predicting closing stock price accurately is an challenging task. Computer aided systems have been proved to be helpful tool for stock prediction such as Artificial Neural Net-work (ANN), Adaptive Neuro Fuzzy Inference System (ANFIS) etc. Latest research works prove that Adaptive Neuro Fuzzy Inference System shows better results than Neural Network for stock prediction. In this paper, an improved Levenberg Marquardt (LM) training algorithm of artificial neural network has been proposed. Improved Levenberg Marquardt algorithm of neural network can predict the possible day-end closing stock price with less memory and time needed, provided previous historical stock market data of Dhaka Stock Exchange such as opening price, highest price, lowest price, total share traded. Moreover,

improved LM algorithm can predict day-end stock price with 53% less error than ANFIS and traditional LM algorithm. It also requires 30% less time, 54% less memory than traditional LM and 47% less time, 59% less memory than ANFIS.

2.6 PAPER 6

6. Efficacy of News Sentiment for Stock Market Prediction Authors: Sneha Kalra ; Jay Shankar Prasad, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)

Stock Market trend prediction will always remain a challenging task due to stochastic nature. The enormous amount of data generated by the news, blogs, reviews, financial reports and social media are considered a treasure of knowledge for researchers and investors. The present work focuses to observe fluctuations in stock prices with respect to the relevant news articles of a company. In this paper, a daily prediction model is proposed using historical data and news articles to predict the Indian stock market movements. Classifier Naïve Bayes is used to categorize the 6-news text having negative or positive sentiment. The count of the positive and negative sentiment of news articles for each day and variance of adjacent days close price along with historical data is used for prediction purpose and an accuracy ranging from 65.30 to 91.2 % achieved with various machine learning techniques.

2.7 PAPER 7

7. Literature review on Artificial Neural Networks Techniques Application for Stock Market Prediction and as Decision Support Tools Authors: Muhammad Firdaus ; Swelandiah Endah Pratiwi ; Dionysia Kowanda ; Anacostia Kowanda

This literature review is aiming to explore the use Artificial Neural Network (ANN) techniques in the field of stock market prediction. Design: Content analysis research technique. Data sources: Information retrieved from

ProQuest electronic databases. Review methods: Utilizing key terms and phrases associated with Artificial Neural Network Stock Market Prediction from 2013-2018. Out of the 129 scholarly journals reviewed, there are 4 stock market studies met the inclusion criteria. The analysis and the evaluation includes 6 ANN derivatives techniques used to predict. Results: Findings from the reviewed studies revealed that all studies show consistency that the accuracy rate of ANN stock market prediction is high. 2 Studies shows accuracy above 90%, 2 studies show accuracy above 50%. Conclusion: This study reveals that the ability of ANN shows consistency of an accuracy rate of stock market prediction. Four method in predicting stock market had an accuracy above 95%. The highest accuracy achieved by using Signal Processing/Gaussian Zero-Phase Filter (GZ-Filter) with 98.7% prediction accuracy. 2018 Third International Conference on Informatics and Computing (ICIC).

2.8 PAPER 8

8. Stock Market Movement Prediction using LDA-Online Learning Model Authors: Tanapon Tantisripreecha ; Nuanwan Soonthomphisaj, 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)

In this paper, an online learning method namely LDA-Online algorithm is proposed to predict the stock movement. The feature set which are the opening price, the closing price, the highest price and the lowest price are applied to fit the Linear Discriminant Analysis (LDA). Experiments on the four well known NASDAQ stocks (APPLE, FACEBOOK, GOOGLE, and AMAZON) show that our model provide the best performance in stock prediction. We compare LDA-online to ANN, KNN and Decision Tree in both Batch and Online learning scheme. We found that LDA-Online provided the best performance. The highest performances measured on GOOGLE, AMAZON, APPLE FACEBOOK stocks are 97.81%, 97.64%, 95.58% and 95.18% respectively.

2.9 PAPER 9

Stock Market Prediction Analysis by Incorporating Social and News Opinion and Sentiment Authors: Zhaoxia Wang ; Seng-Beng Ho ; Zhiping Lin, 2018 IEEE International Conference on Data Mining Workshops (ICDMW)

The price of the stocks is an important indicator for a company and many factors can affect their values. Different events may affect public sentiments and emotions differently, which may have an effect on the trend of stock market prices. Because of dependency on various factors, the stock prices are not static, but are instead dynamic, highly noisy and nonlinear time series data. Due to its great learning capability for solving the nonlinear time series prediction problems, machine learning has been applied to this research area. Learning-based methods for stock price prediction are very popular and a lot of enhanced strategies have been used to improve the performance of the learning based predictors. However, performing successful stock market prediction is still a challenge. News articles and social media data are also very useful and important in financial prediction, but currently no good method exists that can take these social media into consideration to provide better analysis of the financial market. This paper aims to successfully predict stock price through analysing the relationship between the stock price and the news sentiments. A novel enhanced learning-based method for stock price prediction is proposed that considers the effect of news sentiments. Compared with existing learning-based methods, the effectiveness of this new enhanced learning-based method is demonstrated by using the real stock price data set with an improvement of performance in terms of reducing the Mean Square Error (MSE).

2.10 PAPER 10

**10.Stock Price Prediction Using News Sentiment Analysis
Authors:Vijayvergia ; David C. Anastasiu,2019 IEEE Fifth International
Conference on Big Data Computing Service and Applications
(BigDataService).**

Predicting stock market prices has been a topic of interest among both analysts and researchers for a long time. Stock prices are hard to predict because of their high volatile nature which depends on diverse political and economic factors, change of leadership, investor sentiment, and many other factors. Predicting stock prices based on either historical data or textual information alone has proven to be insufficient. Existing studies in sentiment analysis have found that there is a strong correlation between the movement of stock prices and the publication of news articles. Several sentiment analysis studies have been attempted at various levels using algorithms such as support vector machines, naive Bayes regression, and deep learning. The accuracy of deep learning algorithms depends upon the amount of training data provided. However, the amount of textual data collected and analyzed during the past studies has been insufficient and thus has resulted in predictions with low accuracy. In our paper, we improve the accuracy of stock price predictions by gathering a large amount of time series data and analysing it in relation to related news articles, using deep learning models. The dataset we have gathered includes daily stock prices for S&P500 companies for five years, along with more than 265,000 financial news articles related to these companies. Given the large size of the dataset, we use cloud computing as an invaluable resource for training prediction models and performing inference for a given stock in real time. Index Terms-stock market prediction, cloud, big data, machine learning, regression.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 EXISTING SYSTEM

Nowadays, as the connections between worldwide economies are tightened by globalization, external perturbations to the financial markets are no longer domestic. With evolving capital markets, more and more data are being created daily.

The intrinsic value of a company's stock is the value determined by estimating the expected future cash flows of a stock and discounting them to the present, which is known as the book value.

This is distinct from the market value of the stock, that is determined by the company's stock price. This market value of a stock can deviates from the intrinsic value due to reasons unrelated to the company's fundamental operations, such as market sentiment. The fluctuation of stock market is violent and there are many complicated financial indicators.

Only few people with extensive experience and knowledge can understand the meaning of the indicators and use them to make good prediction to get fortune. Most people have to rely solely on luck to earn money from stock trading. However, the advancement in technology, provides an opportunity to gain steady fortune from stock market and also can help experts to find out the most informative indicators to make better prediction.

The prediction of the market value is 10 of paramount importance to help in maximizing the profit of stock option purchase while keeping the risk low.

3.2 PROPOSED SYSTEM

Linear Regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression.

Advantages

- Space complexity is very low it just needs to save the weights at the end of training. Hence, it's a high latency algorithm.
- It's very simple to understand
- Good interpretability Feature importance is generated at the time model building. With the help of hyperparameter lamb, you can handle features selection hence we can achieve dimensionality reduction.

3.3 PROPOSED TECHNIQUES

Analysing and visualizing COVID-19 data can provide valuable insights into the spread and impact of the virus. Here are some proposed techniques for such a project:

1. Data Collection: Gather COVID-19 data from reliable sources such as government health departments, the World Health Organization (WHO), or reputable datasets like Johns Hopkins University's COVID-19 Data Repository.

2. Data Cleaning: Clean the data by removing duplicates, handling missing values, and ensuring consistency in data formats. This step is crucial for accurate analysis.

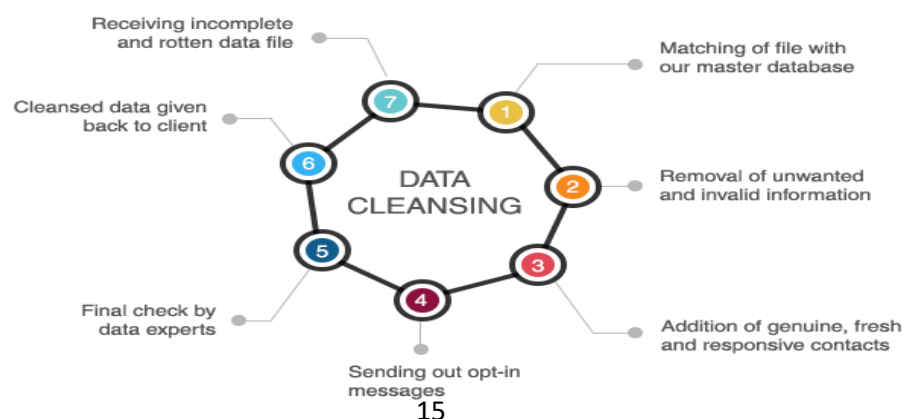


Fig 3.3.1 DATA CLEANING

3. Exploratory Data Analysis (EDA): Conduct EDA to understand the characteristics of the data, such as trends over time, geographical distribution, and demographic patterns. Techniques like summary statistics, histograms, and time series analysis can be helpful.

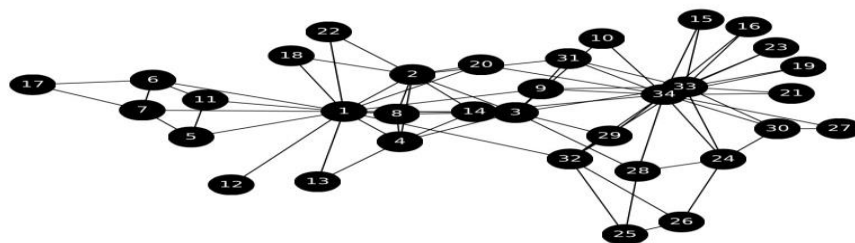
4. Time Series Analysis: Analyze the temporal trends in COVID-19 cases, deaths, and other relevant metrics using time series techniques such as decomposition, autocorrelation, and forecasting models like ARIMA or Prophet.

5. Geospatial Analysis: Visualize the geographical spread of COVID-19 using maps and explore spatial patterns using techniques like choropleth maps, heatmaps, and spatial autocorrelation analysis.

6. Machine Learning Models: Develop machine learning models to predict COVID-19 outcomes, such as future case counts or mortality rates. Common algorithms include regression, random forests, and neural networks.

7. Network Analysis: Investigate the spread of COVID-19 through networks of interactions, such as social networks, transportation networks, or contact tracing data. Network analysis techniques like centrality measures and community detection

164 J. ROGEL-SALAZAR



We can see the connections among members in the network depicted in Figure 3.7. Node number 1 is Mr. Hi (the

Figure 3.7: Zachary's karate club. 34 individuals at the verge of a club split. Edges correspond to friendship relationships among club members.

FIG 3.3.2 NETWORK ANALYSIS

By employing these techniques, you can gain a comprehensive understanding of the COVID-19 pandemic and contribute to efforts in monitoring, mitigation, and decision-making.

3.4 PYTHON IN DATA SCIENCE

Python is one of the most popular programming languages for data science due to its simplicity, versatility, and extensive ecosystem of libraries. Here's how Python is commonly used in data science:

1. Data Manipulation: Python libraries like Pandas provide powerful tools for data manipulation, including reading and writing various file formats, handling missing data, reshaping datasets, and performing operations like filtering, sorting, and aggregation.

2. Data Visualization: Libraries like Matplotlib, Seaborn, and Plotly enable data visualization in Python, allowing you to create a wide range of plots, charts, and graphs to explore data distributions, relationships, and trends.

3. Machine Learning: Python offers rich libraries for machine learning, such as Scikit-learn, TensorFlow, and PyTorch. These libraries provide implementations of various machine learning algorithms, including classification, neural networking, regression, clustering, and dimensionality reduction, as well as tools for model evaluation and hyperparameter tuning.

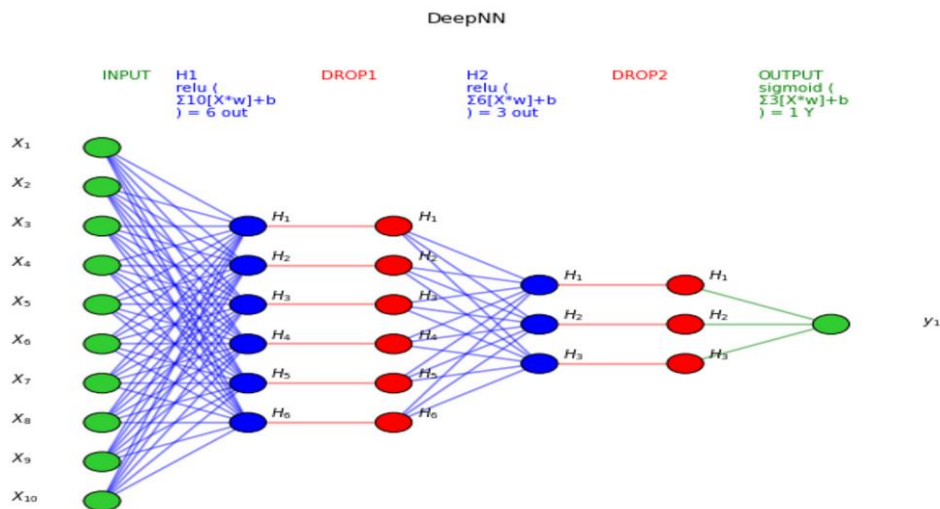


FIG 3.4.1 NEURAL NETWORK

4. Statistical Analysis: Python's stats models library provides tools for statistical modelling, hypothesis testing, and time series analysis, allowing data scientists to conduct rigorous statistical analyses and make data-driven decisions.

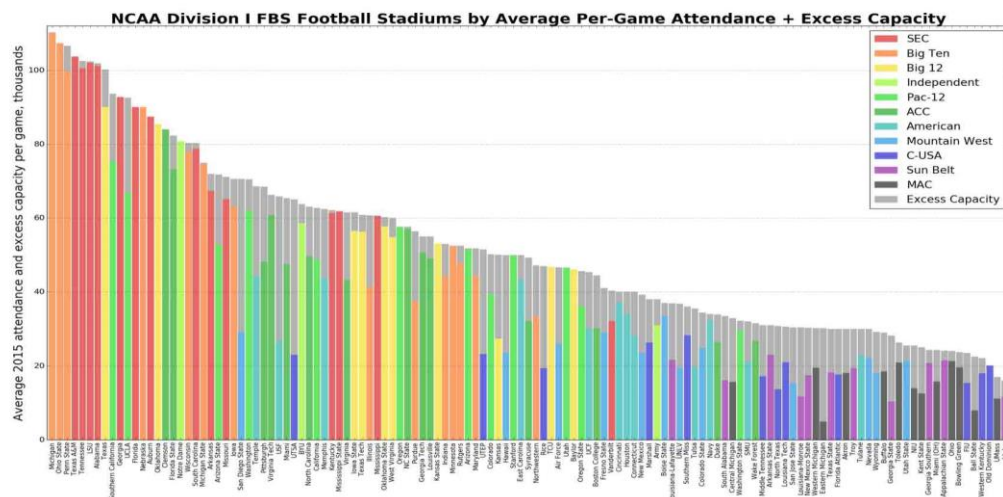


FIG 3.4.2 STATISTICAL ANALYSIS

5. Web Scraping: Python's BeautifulSoup and Scrapy libraries are commonly used for web scraping, enabling data scientists to extract data from websites and APIs for analysis and modelling.

6. Data Cleaning and Preprocessing: Python offers libraries like NumPy and SciPy for numerical computing and advanced mathematical functions, which are essential for data cleaning, Preprocessing, and transformation tasks.

7. Big Data Processing: Python interfaces with big data processing frameworks like Apache Spark and Disk, allowing data scientists to analyse large-scale datasets distributed across clusters of machines.

8. Deep Learning: Python libraries like TensorFlow and PyTorch are widely used for deep learning tasks, including neural network design, training, and deployment, enabling data scientists to build complex models for tasks like image recognition, natural language processing, and reinforcement learning.

9. Interactive Computing: Python's Jupyter Notebook and JupyterLab provide interactive computing environments that combine code, visualizations, and narrative text, making it easy for data scientists to explore data, experiment with algorithms, and communicate their findings.

10. Integration with Other Tools: Python seamlessly integrates with other tools and technologies commonly used in data science, such as databases (e.g., SQL databases, NoSQL databases), cloud services (e.g., AWS, Google Cloud), and visualization tools (e.g., Tableau, Power BI).

Overall, Python's rich ecosystem of libraries, combined with its simplicity and flexibility, makes it an ideal choice for data scientists to tackle a wide range of data science tasks effectively.

3.5 TRAINING

Training data is a large dataset used to teach a machine learning model how to recognize outcomes. It can include text, images, video, or audio, and can be structured in many ways.

For example, for sequential decision trees, the training data would be raw text or alphanumerical data. For supervised ML models, the training data is labeled, while the data used to train unsupervised ML models is not labeled. The quality of training data is important when creating reliable algorithms.

Training involves learning good values for all the weights and the bias from labeled examples. For example, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss, a process called empirical risk minimization.

Here are some skills that data science training programs can help fill out: Critical thinking, Coding for data engineering and analysis, Generating valuable insights from data, and Predictive analytics and data mining.

3.5.1 TRAINING DATASET

The training dataset is a fundamental component in the process of training a machine learning model. It's essentially the data that the model uses to learn patterns, associations, and relationships between input features and their corresponding target outputs. Here's a detailed breakdown of the training dataset:

1. Input Data (Features): The training dataset consists of a set of input data points, also known as features. These features are the variables or attributes that

the model will use to make predictions or classifications. Features can be of various types, including numerical, categorical, or textual data, depending on the nature of the problem being solved. Each data point in the training dataset is represented by a set of features, where each feature provides specific information about the input.

2. Output Labels or Targets: Along with input features, the training dataset also includes corresponding output labels or targets. These labels represent the desired output or prediction that the model should learn to approximate based on the input features. In supervised learning tasks, where the model learns from labeled data, the training dataset contains both input features and their corresponding output labels.

3. Size and Quantity: The size and quantity of the training dataset can significantly impact the performance and generalization ability of the trained model. Typically, a larger training dataset provides more diverse examples for the model to learn from, potentially leading to better generalization on unseen data. However, collecting and labeling large datasets can be time-consuming and resource-intensive, so practitioners often strive to strike a balance between dataset size and model performance.

4. Data Preprocessing: Before feeding the data into the model for training, it often undergoes preprocessing steps to clean, normalize, and transform the features. Data preprocessing may involve tasks such as handling missing values, scaling numerical features, encoding categorical variables, and splitting the dataset into training and validation subsets.

5. Training Process: During the training process, the model iteratively adjusts its internal parameters to minimize the difference between its predictions and the actual target labels in the training dataset.

6. Evaluation: Once the model is trained using the training dataset, it is evaluated on a separate validation or test dataset to assess its performance and generalization ability.

In summary, the training dataset is a crucial component in the machine learning pipeline, providing the raw material from which the model learns to make predictions or classifications. Its quality, size, and diversity play a significant role in determining the performance and robustness of the trained model.

3.6 TESTING

The usage of the word testing in relation to data science projects is primarily used for testing the model performance in terms of accuracy of the model. It can be noted that the word, “Testing” means different for software development and data science projects developments.

3.6.1. TESTING DATASET

In COVID-19 data analysis projects, several datasets are commonly used for various analyses and modelling tasks. Some of the frequently tested datasets include:

- 1. Case Data:** This dataset includes information about confirmed cases, deaths, and recoveries due to COVID-19. It usually contains attributes such as date, location (country, region), case counts, and demographic information.
- 2. Testing Data:** Information about COVID-19 testing, including the number of tests conducted, test positivity rates, and testing methodologies. This dataset helps in understanding testing trends and assessing the spread of the virus.
- 3. Hospitalization Data:** Data related to COVID-19 hospitalizations, including hospital admissions, ICU occupancy rates, ventilator usage, and hospital capacity.

This dataset assists in evaluating healthcare system readiness and capacity planning.

4. Vaccination Data: Information about COVID-19 vaccination campaigns, including the number of doses administered, vaccination rates, vaccine types, and demographic distribution. This dataset helps in assessing vaccination progress and effectiveness.

5. Genomic Data: Genomic sequences of the SARS-CoV-2 virus, including variants of concern (VOCs) and their prevalence over time and geographic regions.

6. Mobility Data: Data on human mobility patterns, including travel, commuting, and social interactions. This dataset helps in studying the impact of mobility on virus transmission and predicting outbreaks.

7. Policy Data: Information about government interventions and public health measures implemented to control the spread of COVID-19, such as lockdowns, mask mandates, and social distancing regulations. This dataset aids in assessing the effectiveness of different policy interventions.

Testing these datasets involves various steps, including data cleaning, Preprocessing, validation, and verification to ensure data quality, consistency, and reliability for accurate analysis and decision-making in COVID-19 research and public health response efforts.

3.7 SPLITTING, IMPUTATION AND INTERPOLATION

- ▶ Splitting – pandas sample () is used to generate sample random row or column from the data frame.
- ▶ Imputation – The process of replacing the missing data with substituted values.
- ▶ Interpolation – A method of constructing new data points within the range of a discrete set of know points.

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Load your dataset into a DataFrame
data = pd.read_csv('your_dataset.csv')

# Split data into training and testing sets
train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)

print(covid_data_imputed.describe())
```

3.8 DATA FRAMES

In a COVID-19 data science project, data frames are commonly used to organize and analyse data. You can use libraries like pandas in Python to create and manipulate data frames.

These data frames can contain various information such as the number of cases, deaths, recoveries, and other relevant metrics, organized by different attributes like date, location, demographics, etc. They serve as a structured way to handle and process the large amounts of data typically involved in COVID-19 analysis. The main key as,

- ▶ It is the crucial components in Covid-19 data analysis project.
- ▶ They help organize and manipulate data efficiency.
- ▶ Python library packages like NumPy, pandas are used for this purpose.
- ▶ Data Frames accepts many different kinds of inputs.

CHAPTER 4

INTRODUCTION OF MACHINE LEARNING

4.1 GENERAL

AI is a mechanism which features algorithms and calculations based on a normal human intelligence to address a problem. The AI behaves and approaches a problem in a similar way that a normal human brain would. Its working mechanism is influenced by human thinking. A collection of expectation and result is achieved by AI by portraying information in a form termed as 'test information' without making use of any predetermined models or being trained in that particular domain. Problems catering to non-related dimensions such as email sifting, PC vision, location of system gate crashers is addressed. Thus, it is assertive that it is not possible to train an AI to address a particular domain, instead an AI trained with general problem-solving abilities, builds up its own algorithms for a set of problems.

An AI engine is allocated with responsibility of prediction or analysis using a PC framework and set of data. For this an AI engine is allocated with packages of scientific methods, logistic calculations, data sets and knowledge about the field of the problems for performing. Moreover, the entire operation of AI is carried based on unsupervised learning model which leaves a very less room for training a robust AI for only a problem specific solution. However, for business purposes modifications are performed before its application.

4.2 OVERVIEW OF MACHINE LEARNING

The name was authored in 1959 by Arthur Samuel Tom M. Mitchell gave a generally cited, increasingly formal meaning of the calculations contemplated in the AI field. This meaning of the assignments in which AI is concerned offers an in a general sense operational definition as opposed to characterizing the field in psychological terms. This pursues Alan Turing's proposition in his paper "Registering Machinery and Intelligence", in which the inquiry "Can machines believe?" is supplanted with the inquiry "Can machines do what we (as speculation elements) can do?" In Turing's proposition the different attributes that could be controlled by a reasoning machine and the different ramifications in building one is uncovered.

Before the introduction of machine learning a general assumption was that a robot needs to learn everything from a human brain to function appropriately. But as efforts were made to do so, it was realized that it is very difficult to make a robot to learn everything from a human brain as the human brain is very much sophisticated. An idea was then proposed that rather than teaching a robot everything we know, it is easier to make the robot learn on its own. The type of dataset we are working upon largely determines how we approach while training the model. Based on the dataset we will feed to the algorithm; the training model would vary. The size, type and dynamism of the dataset will decide what type of training model we would build. Finally, on deciding upon the training model, modifications need to be made to achieve the proper objective function to generate proper set of output that we wish to achieve. The stages of machine learning process are rather termed as ingredients than steps,

because the machine learning is an iterative process. The iterative process is repeated each time to achieve maximum optimization and efficiency.

4.3 MACHINE LEARNING-BASED APPROCHES

The following is a concise outline of mainstream AI based systems for inconsistency identification.

4.3.1 DENSITY BASED DETECTION OF ANOMALY

It derives its working mechanism from KNN algorithm

Assumption - Relevant data locates themselves around a common point in close proximity whereas irregular data are placed at a distance. The data points are clustered at a closed proximity based on a density score, which may be derived using Euclidian distance or appropriate methods based on the data. Classification is made on two bases:

K closest neighbour: In this method the basic clustering mechanism is dependent on separation measurements of each data points which determines the clustering or similarities of each information considered.

Relative thickness of the information - Also known as Least Outlier Fraction (LOF).

Calculation is performed on the basis of separation metric.

4.3.2 CLUSTERING BASED DETECTION OF ANOMALY

Clustering is an exceptional algorithm known for its optimization and robust nature. For this reason, it is widely used in unsupervised learning

Assumption - Data points that are similar tends to get gather around specific points. The relative distance of each cluster is achieved by its shortest distance from the centroid of the space.

K means is widely used in data classification. It makes use of k means algorithm to cluster closely related data in close proximity forming clusters.

4.3.3 SVM BASED DETECTION OF ANOMALY

- A support vector machine is one of the most important algorithm used for classification purposes
- The SVM uses methods to determine a soft boundary to distinguish data clusters. Data closely related falls within the parameter of a closed boundary. This results in formation of multiple clusters. SVM is widely used for binary classifications also. Most of the SVM algorithms works based on unsupervised learning.
- The yield of an abnormality locator are mostly numeric scalar qualities for distinguishing areas of explicit edges.

In this Jupiter journal we are going to assume the acknowledgment card misrepresentation recognition as the contextual investigation for understanding this idea in detail utilizing the accompanying Anomaly Detection Techniques in particular

4.4 DATASET

A dataset corresponds to a collection of data which may or may not be related to each other. A dataset can consist of data related to a particular

domain. It may consist information for a single member or a group of members. For ex personal and other relevant details of an employee can be termed as a dataset, whereas collection of the information of all the employees working for that company is also a dataset. Thus, the purpose of the problem defines the size of the dataset. A dataset consists of multiple columns often termed as parameters and multiple rows known as tuples. Individual data pieces are also termed as datum. For example, in a data set consisting of employee details of a company.

CHAPTER 5

SYSTEM SPECIFICATION

5.1 GENERAL

The necessity for the most part dependent on two classes: they is practical portray every single required usefulness for framework administrations which are given by the customers. Non-useful necessities characterize the framework properties and compels. The equipment prerequisites indicate the equipment functionalities and required speed and limit of the fringe.

The product prerequisites incorporate programming expected to create and run the framework.

5.2 HARDWARE SPECIFICATION

- System - Core i5
- Mobile - Android
- Monitor - RGB colour
- Hard Disk - 2 TB
- Mouse - Microsoft
- Ram - 8GB

5.3 SPECIFICATION OF THE SOFTWARE

- Operating system - Win 10
- Dataset - csv
- Language - Python

5.4 SOFTWARES USED

- Python 3.5
- NumPy 1.11.3
- Matplotlib 1.5.3
- Pandas 0.19.1
- Seaborn 0.7.1
- SciPy
- Scikit-learn 0.18.1

5.5 PYTHON PACKAGES

Python is one of the most popular programming languages used across various tech disciplines, especially in data science and machine learning. Python offers an easy-to-code, object-oriented, high-level language with a broad collection of libraries for a multitude of use cases. It has over 137,000 libraries.

One of the reasons Python is so valuable to data science is its vast collection of data manipulation, data visualization, machine learning, and deep learning libraries.

5.5.1 NUMPY

NumPy, is one of the most broadly-used open-source Python libraries and is mainly used for scientific computation. Its built-in mathematical functions enable lightning-speed computation and can support multidimensional data and large matrices.

It is also used in linear algebra. NumPy Array is often used preferentially over lists as it uses less memory and is more convenient and efficient.

5.5.2 PANDAS

Pandas is an open-source library commonly used in data science. It is primarily used for data analysis, data manipulation, and data cleaning. Pandas allow for simple data modeling and data analysis operations without needing to write a lot of code.

As stated on their website, pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool. Some key features of this library include:

- Data Frames, which allow for quick, efficient data manipulation and include integrated indexing;
- Several tools which enable users to write and read data between in-memory data structures and diverse formats, including Excel files, text and CSV files, Microsoft, HDF5 formats, and SQL databases;
- Intelligent label-based slicing, fancy indexing, and sub setting of large data sets;
- High-performance merging and joining of data sets;

- A powerful group by engine which enables data aggregation or transformation, allowing users to perform split-apply-combine operations on data sets;
- Time series-functionality which enables date range generation and frequency conversion, moving window statistics, date shifting, and lagging. You'll even be able to join time series and create domain-specific time offsets without worrying you'll lose data;
- Ideal when working with critical code paths written in C or Python.

5.5.3 MATPLOTLIB

Matplotlib is an extensive library for creating fixed, interactive, and animated Python visualizations. A large number of third-party packages extend and build on Matplotlib's functionality, including several higher-level plotting interfaces (Seaborn, HoloViews, ggplot, etc.)

Matplotlib is designed to be as functional as MATLAB, with the additional benefit of being able to use Python. It also has the advantage of being free and open source. It allows the user to visualize data using a variety of different types of plots, including but not limited to scatterplots, histograms, bar charts, error charts, and boxplots. What's more, all visualizations can be implemented with just a few lines of code.

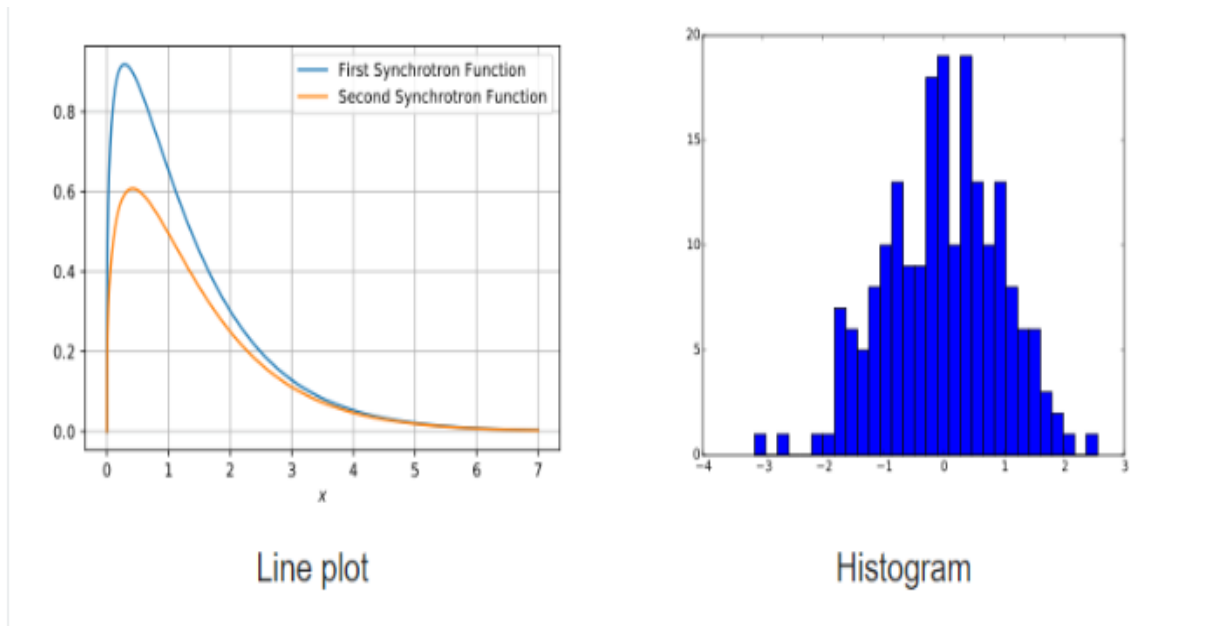


FIG 4.3.3.1 MATPLOTLIB

5.5.4 SEABORN

Another popular Matplotlib-based Python data visualization framework, seaborn is a high-level interface for creating aesthetically appealing and valuable statistical visuals which are crucial for studying and comprehending data.

This Python library is closely connected with both NumPy and pandas' data structures. The driving principle behind Seaborn is to make visualization an essential component of data analysis and exploration; thus, its plotting algorithms use data frames that encompass entire datasets.

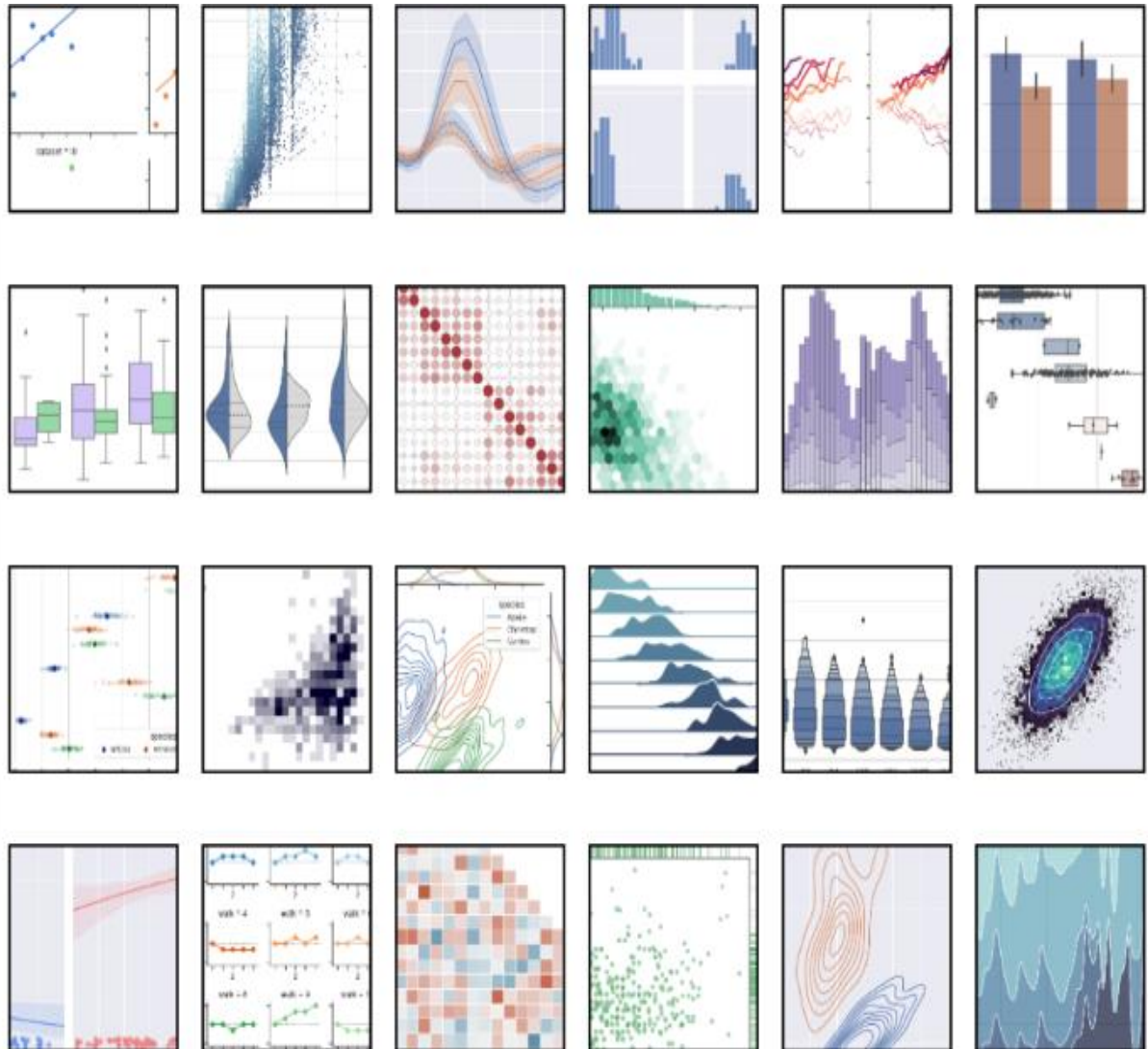


FIG 4.3.4.1 SEABORN

5.5.5 PLOTLY

The hugely popular open-source graphing library Plotly can be used to create interactive data visualizations. Plotly is built on top of the Plotly JavaScript library (plotly.js) and can be used to create web-based data visualizations that can be saved as HTML files or displayed in Jupyter notebooks and web applications using Dash.

It provides more than 40 unique chart types, such as scatter plots, histograms, line charts, bar charts, pie charts, error bars, box plots, multiple axes, sparklines, dendrograms, and 3-D charts. Plotly also offers contour plots, which are not that common in other data visualization libraries.

If you want interactive visualizations or dashboard-like graphics, Plotly is a good alternative to Matplotlib and Seaborn. It is currently available for use under the MIT license.

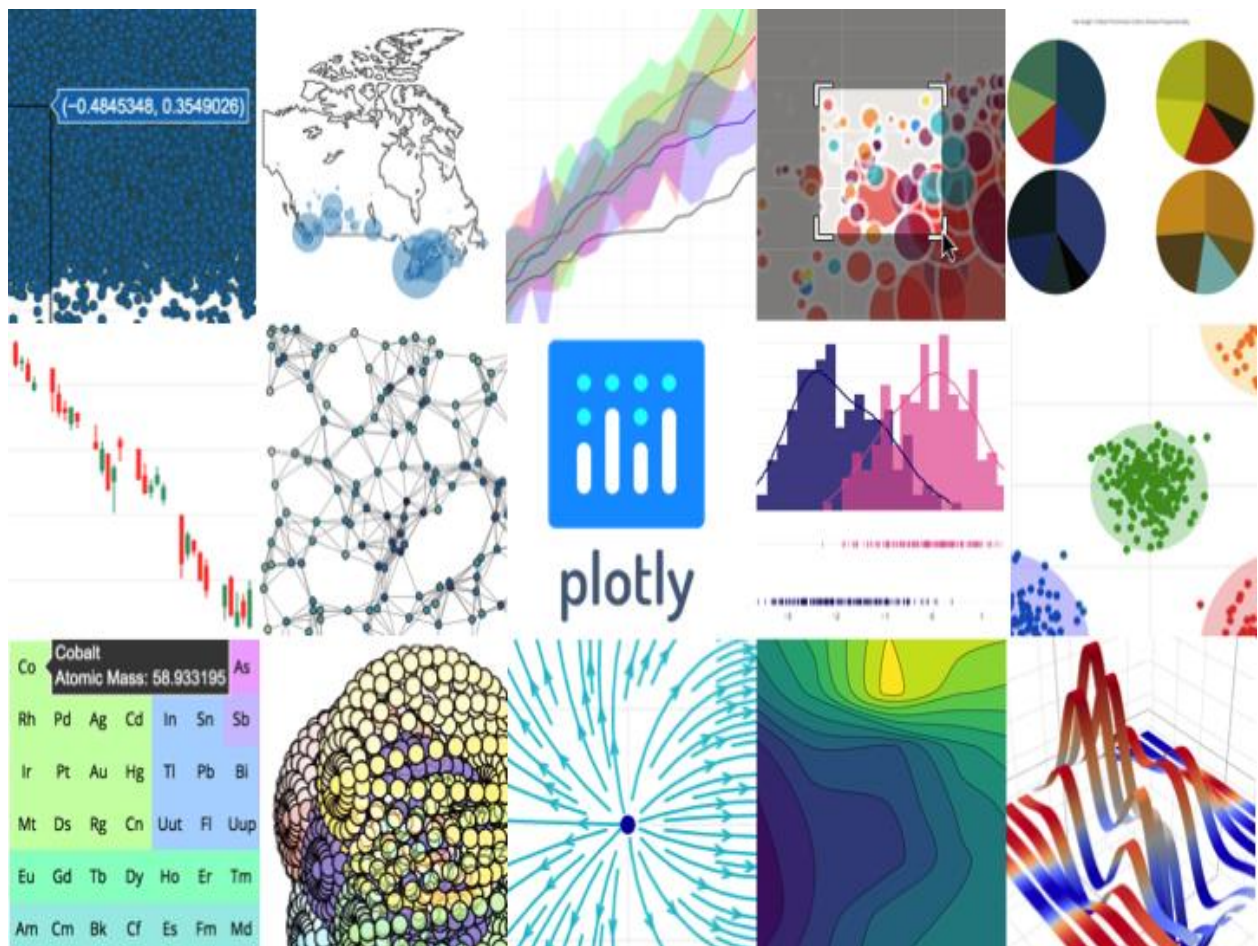


FIG 4.3.5.1 PLOTLY

5.6 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports various programming languages, including Python, R, Julia, and others, making it a versatile tool for data science projects.

1. Interactive Computing: Jupyter Notebooks provide an interactive computing environment where you can write and execute code in individual cells. This allows you to experiment with code, test hypotheses, and explore data interactively.

2. Integration of Code and Documentation: One of the key features of Jupyter Notebooks is the ability to include narrative text, equations, and visualizations alongside code cells. This integration of code and documentation makes it easy to create rich, self-explanatory documents that document your data analysis process step by step.

3. Data Exploration and Visualization: Jupyter Notebooks are well-suited for data exploration and visualization tasks. You can use libraries like Pandas, NumPy, Matplotlib, Seaborn, and Plotly to analyze and visualize data directly within the notebook environment. Interactive visualizations can be created using tools like Plotly or Bokeh, allowing for exploration of complex datasets.

4. Reproducibility: Jupyter Notebooks promote reproducibility in data science projects by capturing the entire data analysis workflow in a single document. By including code, data, visualizations, and explanations in one place, you make it easier for others to understand and reproduce your analysis.

5. Collaboration and Sharing: Jupyter Notebooks can be easily shared with colleagues or collaborators, either as static documents or interactive notebooks hosted on platforms like GitHub or Jupyter Hub. This facilitates collaboration and allows team members to review, comment, and contribute to the analysis.

CHAPTER 6

DESIGN ENGINEERING

6.1 ARCHITECTURE DIAGRAM

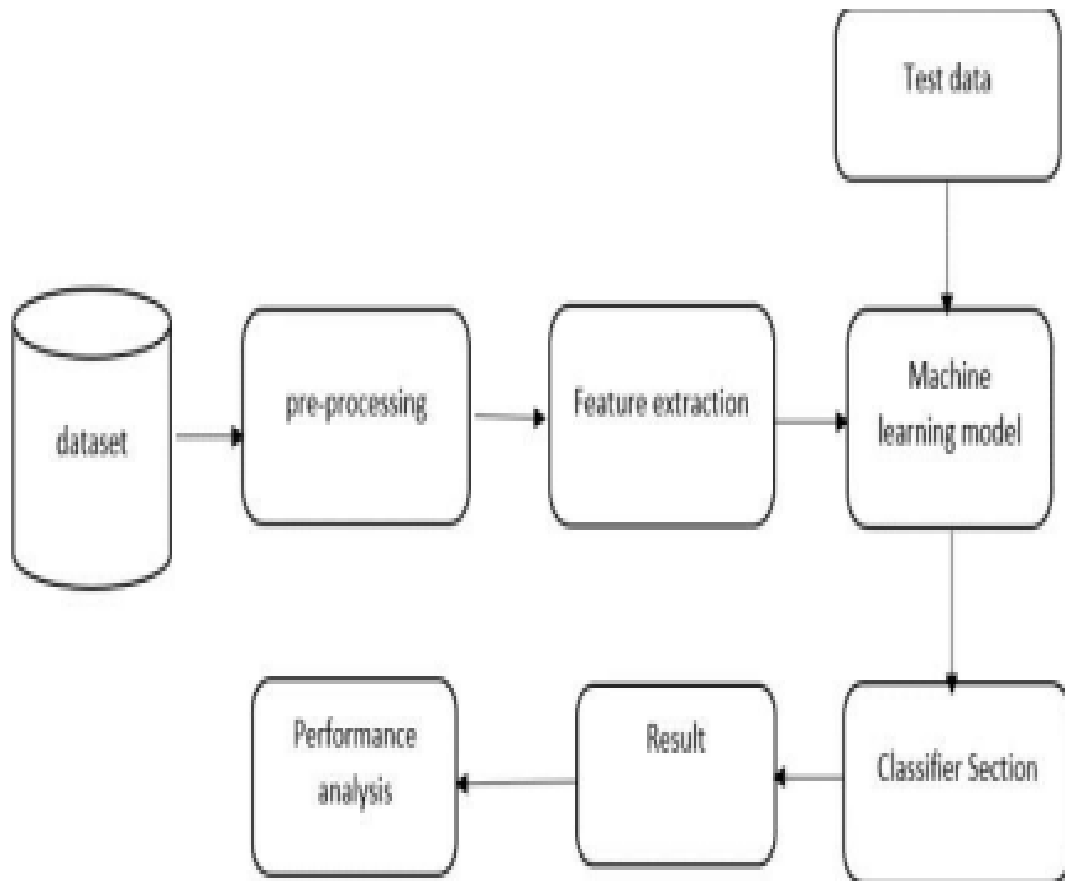


FIG 6.1.1 ARCHITECTURE DIAGRAM

A typical architecture diagram for stock market prediction using machine learning involves several components that work together to analyze historical stock data, extract features, train predictive models, and deploy them for making future predictions. Here's a high-level overview of such an architecture:

Data Collection and Preprocessing:

This component gathers historical stock market data from various sources such as financial APIs, databases, or web scraping tools.

Data Preprocessing involves cleaning the data, handling missing values, and transforming it into a suitable format for analysis.

Feature Extraction and Selection:

Extracting relevant features from the raw data is crucial for building accurate predictive models. These features could include price movements, trading volumes, technical indicators, and fundamental factors.

Feature selection techniques may be employed to identify the most informative features that contribute to the predictive power of the model.

Model Training:

Machine learning models such as regression, decision trees, random forests, or neural networks are trained using historical stock market data.

The training process involves feeding the selected features and corresponding target variables (e.g., future stock prices or price movements) to the model to learn the underlying patterns and relationships.

Model Evaluation:

After training, the model's performance is evaluated using a separate validation dataset or through cross-validation techniques to assess its accuracy, precision, recall, or other relevant metrics.

This step helps in identifying the best-performing model and fine-tuning its parameters if necessary.

Deployment:

Once a satisfactory model is trained and evaluated, it is deployed into a production environment where it can make real-time predictions.

Deployment may involve integrating the model into a web application, API, or trading platform, depending on the specific use case.

Monitoring and Maintenance:

Continuous monitoring of the deployed model's performance is essential to ensure its reliability and effectiveness over time.

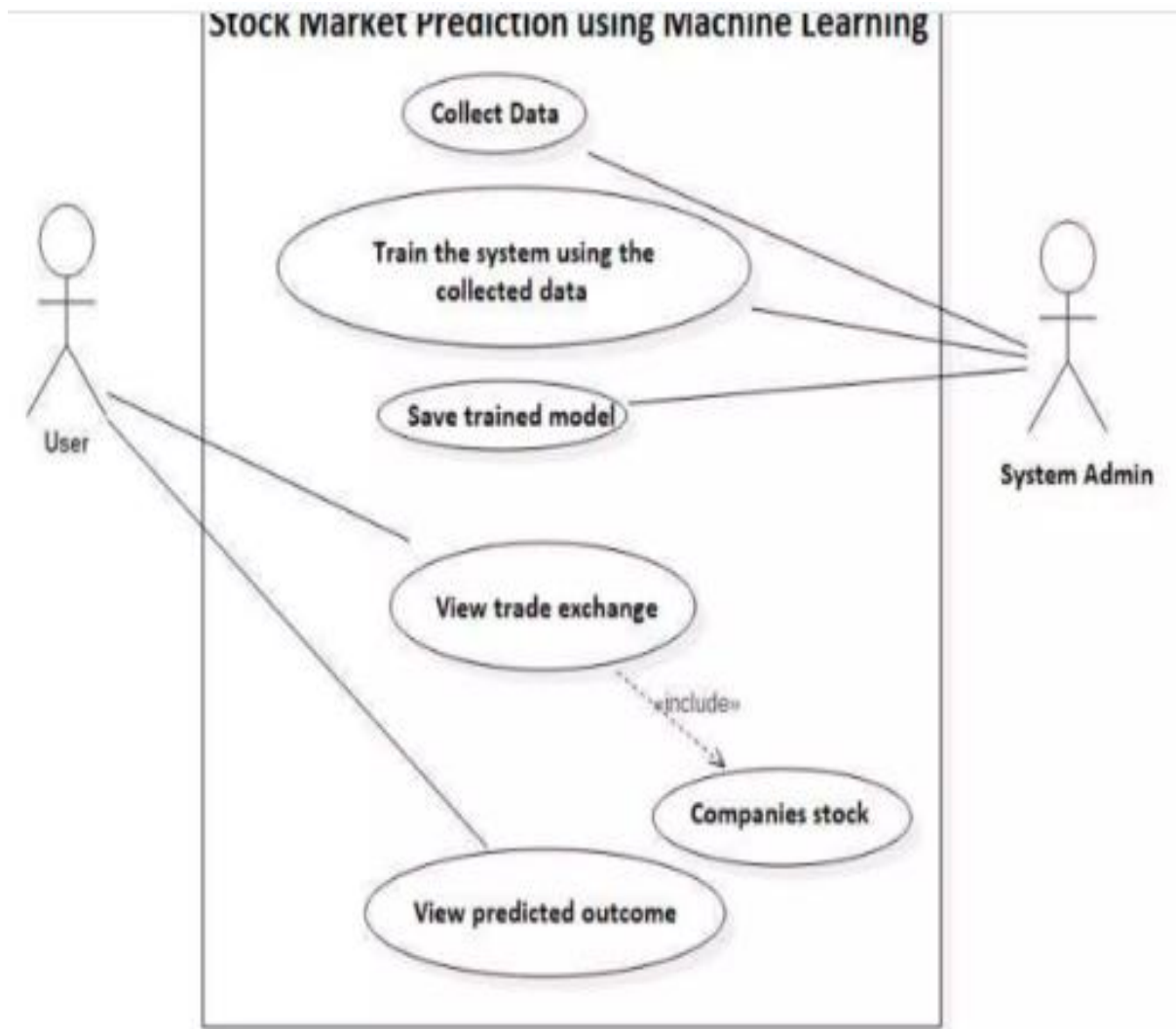
Periodic retraining of the model with updated data may be necessary to adapt to changing market conditions and prevent model degradation.

Feedback Loop:

Feedback from the model's predictions and their actual outcomes can be used to further refine the model or update its features, improving its predictive accuracy and robustness.

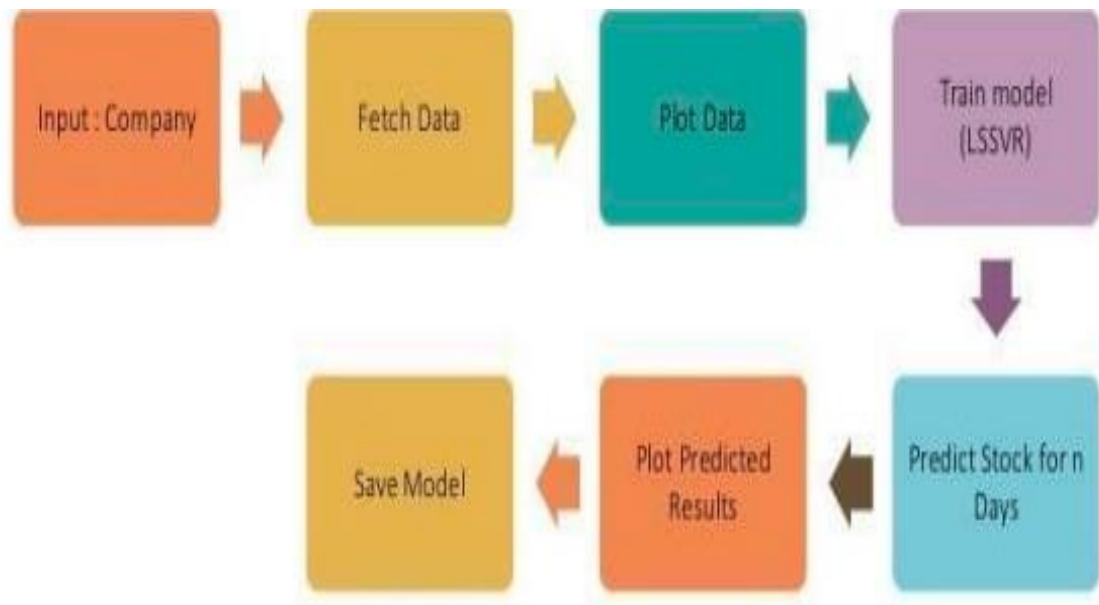
The architecture diagram may visually represent these components and their interactions, illustrating the flow of data and processes from data collection to prediction generation. It helps stakeholders understand the system's design and implementation, facilitating communication and collaboration among team members involved in developing and maintaining the stock market prediction system.

6.2 USE CASE DIAGRAM



6.2.1 USE CASE DIAGRAM

6.3 DATA FLOW DIAGRAM



6.3.1 DFD DIAGRAM

CHAPTER 7

IMPLEMENTAION

7.1 GENERAL

Implementation phase brings out the design tweaked out into a operational system. Hence this can be deliberated to be most precarious juncture in accomplishing the efficacious system and in convincing the user faith that system will operate and be effective. This phase encompasses vigilant planning & design, examination of prevailing system and constraints on execution, design & scheming of methods to change over.

7.2 PROCEDURE FOLLOWED DURING IMPLEMENTATION

The application – Credit Card Fraud Detection which is in itself the complete & full-fledged GUI enabled application to envisage/foresee the authenticity & legitimacy of a transaction has been implemented, as per the following steps:

- Install Anaconda from a reliable source.
- Import packages: pandas, Scipy, Matplotlib, Seaborn
- Load the dataset, a dataset is the pool of data for analytical/critical purpose, a (.CSV) file.
- Reconnoiter and get through the dataset through data. Shape, data. Describe.

- Determine the count of fraud cases by checking if class is 0 or 1.
- In the similar procedure, get the correlation matrix.
- Next, there is a need to determine the local outlier factor.
- The GUI is developed using PyQt library.
- The PyQt library, provides tools to achieve a complete GUI enabled application, similar to swings in java environment.
- Define the constructor in the file.

7.2.1 DATASET DESIGN

The screenshot shows an Excel spreadsheet titled 'tesla.csv - Excel (Product Activation Failed)'. The spreadsheet contains a dataset of Tesla stock prices. The columns are labeled: Date, Open, High, Low, Close, Adj Close, and Volume. The data is organized into rows, with the first row (row 1) containing the column headers. The subsequent rows (rows 2 to 23) contain numerical data for each column. The 'Date' column contains dates in YYYY-MM-DD format. The 'Open', 'High', 'Low', and 'Close' columns contain decimal values. The 'Adj Close' column contains decimal values. The 'Volume' column contains integer values. The spreadsheet is displayed in a window with a green title bar and a standard Excel ribbon interface.

Date	Open	High	Low	Close	Adj Close	Volume
2019-12-31	19	25	17.54	23.89	23.89	18766300
2020-01-02	25.79	30.42	23.3	23.83	23.83	17187100
2020-01-03	25	25.92	20.27	21.96	21.96	8218800
2020-01-06	23	23.1	18.71	19.2	19.2	5139800
2020-01-07	20	20	15.83	16.11	16.11	6866900
2020-01-08	16.4	16.63	14.98	15.8	15.8	6921700
2020-01-09	16.14	17.52	15.57	17.46	17.46	7711400
2020-01-13	17.58	17.9	16.55	17.4	17.4	4050600
2020-01-14	17.95	18.07	17	17.05	17.05	2202500
2020-01-15	17.39	18.64	16.9	18.14	18.14	2680100
2020-01-16	17.94	20.15	17.76	19.84	19.84	4195200
2020-01-17	19.94	21.5	19	19.89	19.89	3739800
2020-01-21	20.7	21.3	20.05	20.64	20.64	2621300
2020-01-22	21.37	22.25	20.92	21.91	21.91	2486500
2020-01-23	21.85	21.85	20.05	20.3	20.3	1825300
2020-01-27	20.66	20.9	19.5	20.22	20.22	1252500
2020-01-28	20.5	21.25	20.37	21	21	957800
2020-01-29	21.19	21.56	21.06	21.29	21.29	653600
2020-01-30	21.5	21.5	20.3	20.95	20.95	922200
2020-02-02	20.91	21.18	20.26	20.55	20.55	619700
2020-02-03	20.55	20.9	20.51	20.72	20.72	467200
2020-02-04	20.77	20.88	20	20.35	20.35	616000

FIG 7.2.1.1 DATASET

Datasets for stock market prediction using machine learning can vary in size, granularity, and the type of data they include. Here are some common types of datasets used in stock market prediction:

Historical Price Data:

This dataset includes historical stock prices over a period of time, typically recorded at regular intervals such as daily, hourly, or minute-by-minute.

Each data point may contain attributes like opening price, closing price, highest price, lowest price, and trading volume.

Fundamental Data:

Fundamental datasets provide information about a company's financial health and performance, including balance sheets, income statements, cash flow statements, and key financial ratios.

Fundamental data can be used to derive features that capture the fundamental characteristics of companies, such as earnings per share (EPS), price-to-earnings (P/E) ratio, debt-to-equity ratio, etc.

Technical Indicators:

Technical indicators are derived from historical price and volume data using mathematical formulas. They provide insights into market trends, momentum, volatility, and other aspects of price movements.

Common technical indicators include moving averages, Relative Strength Index (RSI), MACD (Moving Average Convergence Divergence), Bollinger Bands, etc.

Market Sentiment Data:

Market sentiment datasets capture public sentiment and perception about stocks or the overall market. They may include data from news articles, social media posts, analyst reports, and online forums.

Natural language processing (NLP) techniques can be applied to analyze textual data and extract sentiment scores or relevant keywords.

Alternative Data:

Alternative datasets encompass non-traditional sources of data that can provide unique insights into market trends and dynamics. Examples include satellite imagery, credit card transactions, foot traffic data, and weather data.

Integrating alternative data sources with traditional financial data can enhance the predictive power of machine learning models.

Macro-Economic Indicators:

Macro-economic datasets contain information about broader economic factors such as GDP growth, inflation rates, interest rates, unemployment rates, and geopolitical events.

Changes in macro-economic indicators can impact the performance of financial markets and influence stock prices.

When building machine learning models for stock market prediction, researchers and practitioners often combine multiple datasets, leveraging the complementary information provided by different sources. However, it's essential to preprocess and normalize the data appropriately to address issues like missing values, outliers, and data inconsistencies, ensuring the quality and reliability of the predictive models.

7.2.2 PREPROCESSING

The data values have been plotted using histogram describing the numerical distribution of the data values.

After selecting the dataset, the first step is to pre-process the data to make it suitable for model training and testing. In this step, the data were processed in the following ways.

- Finding and filling/removing any null values.
- Standardizing the ‘Amount’ column to make it easy for analysis.
- Removing the ‘Time’ Column from the dataset as it was not contributing much during training and evaluation.
- Checking and removing duplicate entries in the dataset.

The dataset used in the process was devoid of any missing or null values. It is important to mention that intentional actions to reduce the influence of outliers were not included. The conclusion was based on the understanding that the selected machine-learning model is naturally resistant to outliers. Moreover, incorporating outliers into the dataset was considered advantageous since it brings the model into closer alignment with the complexities of real-world situations. This study sought to improve the model’s capacity to handle the dynamic and different nature of credit card transactions by not using explicit outlier-handling strategies. This approach made the model more adaptable and applicable to real-world scenarios.

The reason for standardizing the ‘Amount’ column instead of normalizing it is that, as mentioned in the description of the dataset, all features were the result of Principal Component Analysis (PCA) except ‘Time’ and ‘Amount’, and the ‘Amount’ scale differed significantly from all other features (V1–V28). Hence, the ‘Amount’ feature was standardized.

Therefore, the use of feature selection techniques was not possible because it would have required clear visibility of feature information. To avoid any potential confusion caused by algorithmic feature selection, a deliberate choice was made to abstain from this process. Furthermore, the ‘Time’ column was excluded from consideration during manual analysis because it did not contribute any meaningful information. It only reflected a sequential count of entries without any temporal significance. Although the lack of feature selection techniques may result in longer training and testing durations, this strategy was considered the best choice to guarantee the retention of all potentially relevant features without relying on feature-specific knowledge.

7.2.3 PREDICTION

The prediction that has been achieved using the Isolation Forest Algorithm and Local Outlier Factor Algorithm has been shown below

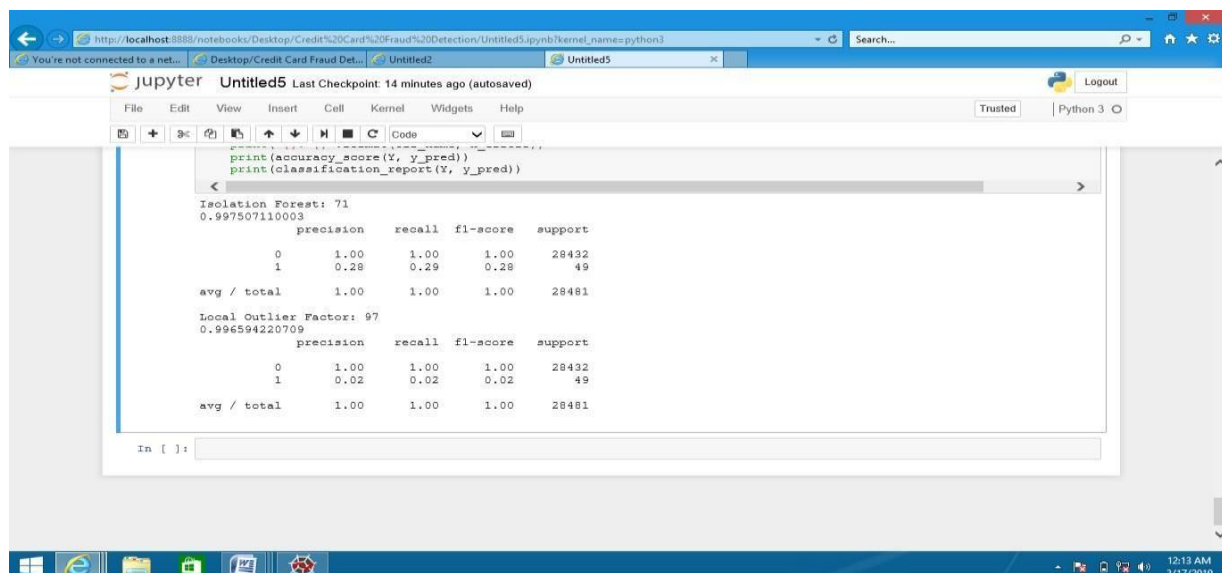


FIG 7.2.3.1 ACCURACY

CHAPTER 8

SOFTWARE TESTING

8.1 GENERAL

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown

8.2 TESTING

Table 8.1: Tabulated Results

Test Case (sample split)	Assumption	Description	Expected Output	Actual Output		Log Message
				Isolation Forest Algorithm - Algorithm I Accuracy(%)	Local Outlier Factor - Algorithm II Accuracy(%)	
10:90	Algorithm-I will perform better	Check for accuracy at 10% training of data	99.70505	99.75071	99.65942	Success

15:85	Algorithm-II will perform better	Check for accuracy at 15% training of data	99.71675	99.75421	99.67931	Fail
20:80	Algorithm-II will perform better	Check for accuracy at 20% training of data	99.73485	99.69628	99.77352	Success
25:75	Algorithm-I will perform better	Check for accuracy at 25% training of data	99.73311	99.77107	99.69523	Success
30:70	Algorithm-I will perform better	Check for accuracy at 30% training of data	99.73425	99.77645	99.69218	Success

The test cases has been based on the following sample split (train: test) :- (10:90), (15:85), (20:80), (25:75) and (30:70).

Outlier Fraction: Describes the ratio of outlier values to the real values in the dataset

Data Shape: Describes the number of rows and columns in the training sample.

Isolation Forest Algorithm Accuracy: Describes the accuracy achieved on the test dataset using Isolation Forest Algorithm

Local Outlier Factor Accuracy: Describes the accuracy achieved on the test dataset using Local Outlier Factor

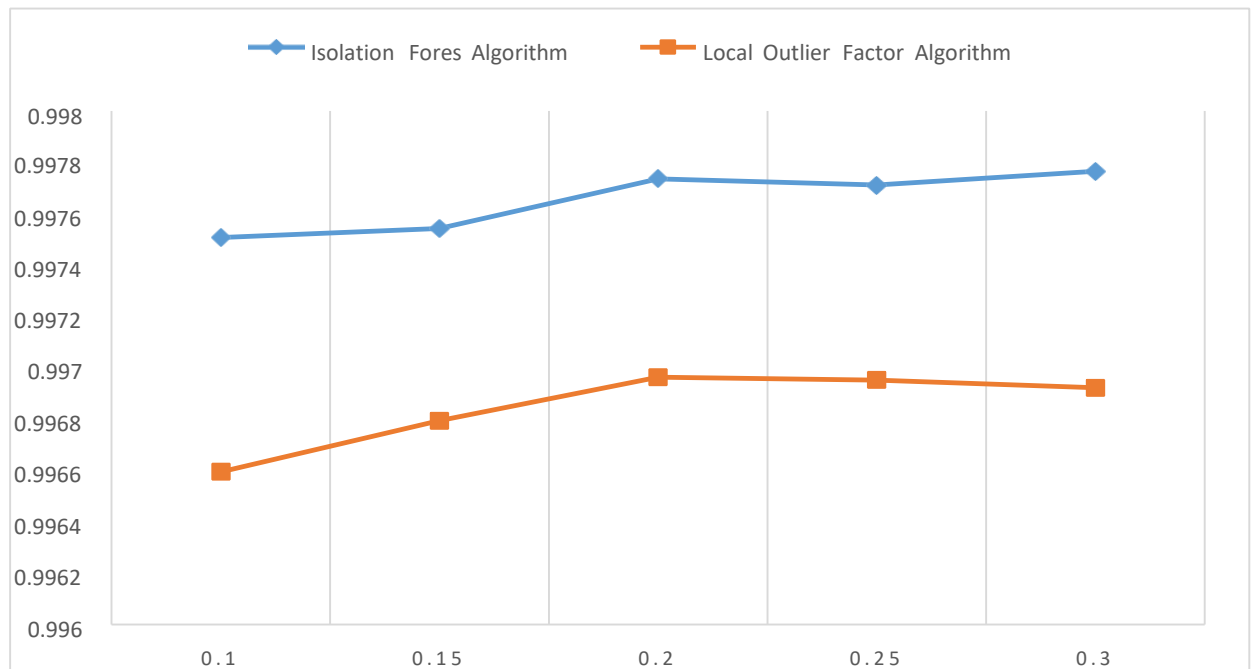


Figure 8.1: Comparison Chart

As we tested the application under different test conditions, the application gave appropriate results. The above chart depicts the accuracy based on two algorithms used, i.e. the Isolation Forest Algorithm and the Local Outlier Factor Algorithm.

CHAPTER 9

ALGORITHM

9.1 Logistic Regression Logistic

Logistic Regression is a Classification model, which tries to classify the data based on the probability of it occurring.

This algorithm is used in multiple places where classification is required, we have used it to classify if the patient is susceptible to be infected by covid or not This is one of the classification methods which we have used. It used Sigmoid function to classify the data.

Logistic regression is a supervised machine learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event, or observation. The model delivers a binary or dichotomous outcome limited to two possible outcomes: yes/no, 0/1, or true/false.

Logical regression analyzes the relationship between one or more independent variables and classifies data into discrete classes. It is extensively used in predictive modeling, where the model estimates the mathematical probability of whether an instance belongs to a specific category or not.

For example, 0 – represents a negative class; 1 – represents a positive class. Logistic regression is commonly used in binary classification problems where the outcome variable reveals either of the two categories (0 and 1).

Some examples of such classifications and instances where the binary response is expected or implied are:

1. Determine the probability of heart attacks: With the help of a logistic model, medical practitioners can determine the relationship between variables such as the weight, exercise, etc., of an individual and use it to predict whether the person will suffer from a heart attack or any other medical complication.

2. Possibility of enrolling into a university: Application aggregators can determine the probability of a student getting accepted to a particular university or a degree course in a college by studying the relationship between the estimator variables, such as GRE, GMAT, or TOEFL scores.

3. Identifying spam emails: Email inboxes are filtered to determine if the email communication is promotional/spam by understanding the predictor variables and applying a logistic regression algorithm to check its authenticity.

9.1.1 ADVANTAGES OF LOGISTICS ALGORITHM

The logistic regression analysis has several advantages in the field of machine learning.

1. Easier to implement machine learning methods: A machine learning model can be effectively set up with the help of training and testing. The training identifies patterns in the input data (image) and associates them with some form of output (label). Training a logistic model with a regression algorithm does not demand higher computational power. As such, logistic regression is easier to implement, interpret, and train than other ML methods.

2. Suitable for linearly separable datasets: A linearly separable dataset refers to a graph where a straight line separates the two data classes. In logistic regression, the y variable takes only two values. Hence, one can effectively classify data into two separate classes if linearly separable data is used.

3. Provides valuable insights: Logistic regression measures how relevant or appropriate an independent/predictor variable is (coefficient size) and also reveals the direction of their relationship or association (positive or negative).

$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

e = Euler's number ~ 2.71828

Sigmoid function converts input into range 0 to 1

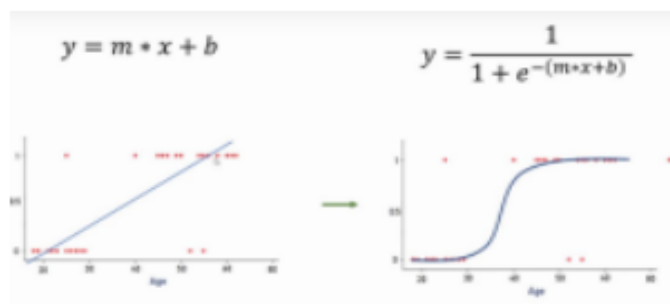


FIG 9.1.1 GRAPH FOR LOGISTIC REGRESSION

9.2 KNN

KNN is a supervised machine learning algorithm. KNN forms groups based on the criteria's and then decides for the incoming data where to put in which category. It can be used for regression and for classification too, but mostly for the classification only it is used.

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs' images and based on the most similar features it will put it in either cat or dog category.

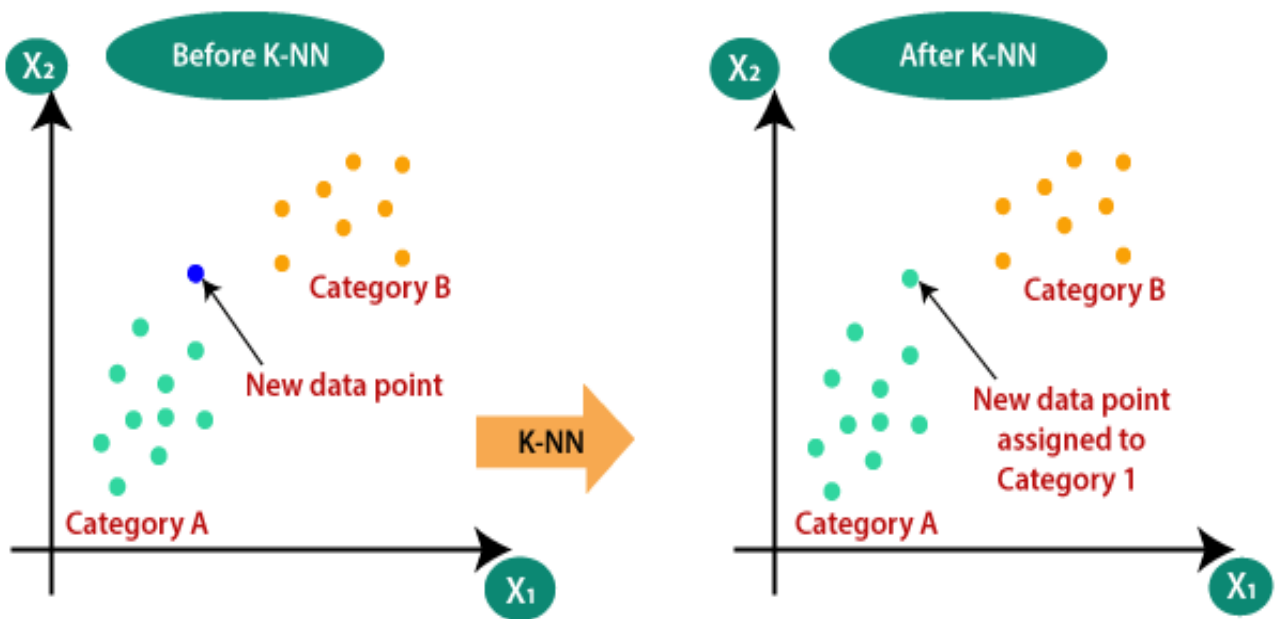


FIG 9.2.1 REPRESENTATION OF KNN ALGORITHM

9.3 RANDOM FOREST CLASSIFIER

Random forest is a supervised learning algorithm. The "forest" it builds is a group of decision trees, usually trained with the “bagging” system.

The general idea of the bagging system is that a combination of learning models increases the overall result.

Put simply: random forest builds multiple decision trees and combines them together to get a more accurate and stable prediction.

One big advantage of random forest is that it can be used for both classification and regression problems, which form the most of current machine learning systems.

A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning.

We know that a forest comprises numerous trees, and the more trees more it will be robust. Similarly, the greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability.

Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

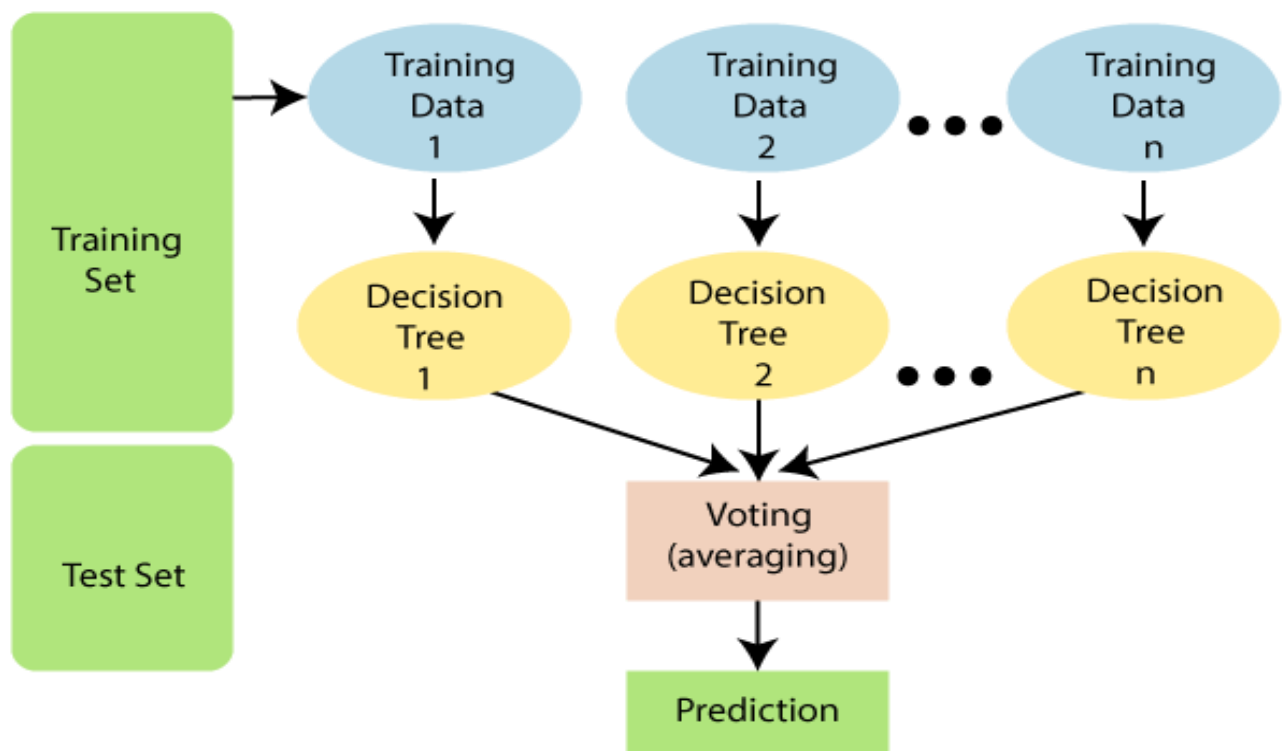


FIG 9.3.1. RANDOM FOREST CLASSIFIER

9.4 DECISION TREE ALGORITHM

- A. Decision Tree is a supervised machine learning algorithm.
- B. Two nodes which are decision node and leaf node are the ones making the decision.
- C. Repeated if clauses are at work when deciding the classification for the algorithm.

A decision tree is a **non-parametric supervised learning algorithm for classification and regression tasks**. It has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes. Decision trees are used for classification and regression tasks, providing easy-to-understand models.

A decision tree is a hierarchical model used in decision support that depicts decisions and their potential outcomes, incorporating chance events, resource expenses, and utility.

This algorithmic model utilizes conditional control statements and is non-parametric, supervised learning, useful for both classification and regression tasks. The tree structure is comprised of a root node, branches, internal nodes, and leaf nodes, forming a hierarchical, tree-like structure.

It is a tool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems.

The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

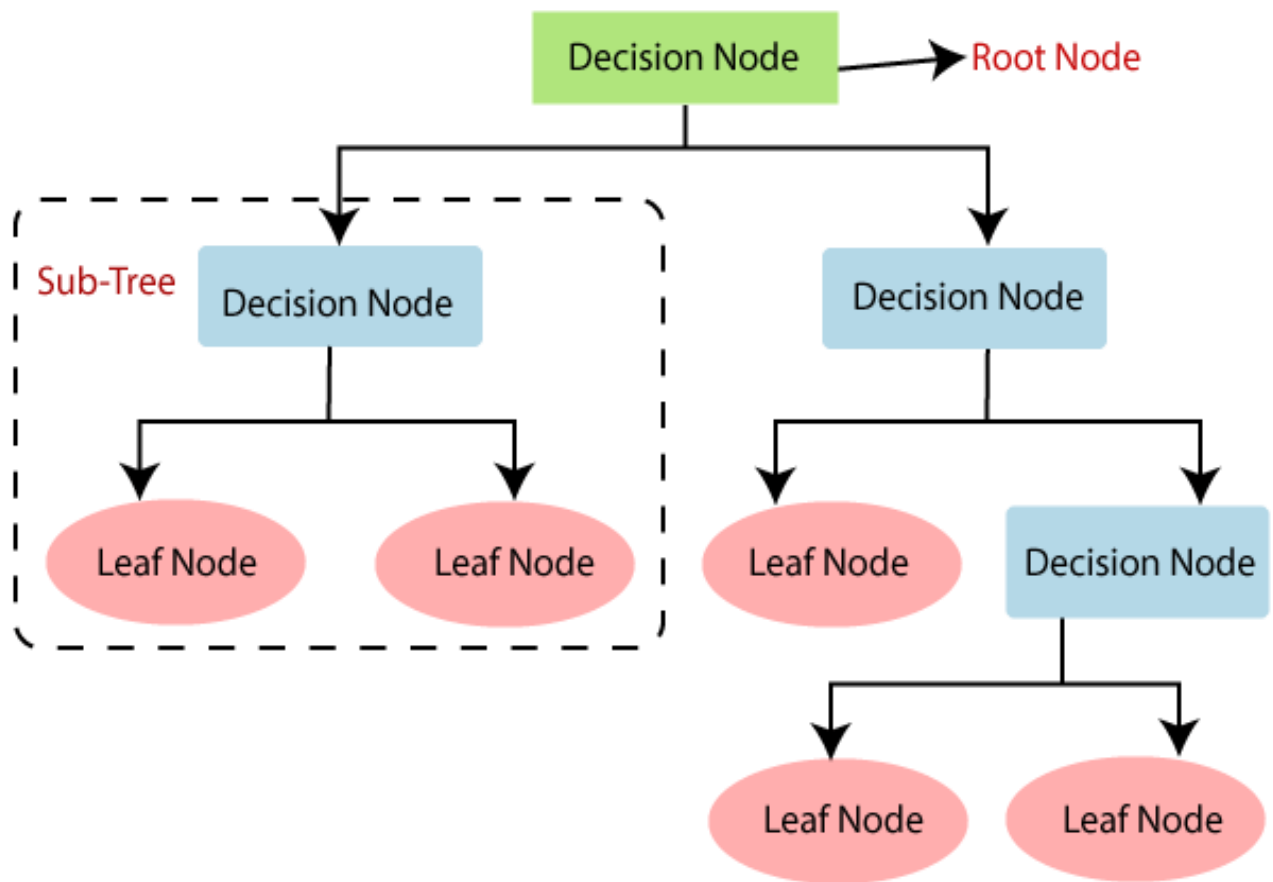


FIG 9.4.1 DECISION TREE ALGORITHM

CHAPTER 10

IMPLEMENTATION

Random forest algorithm has a set of rules is used for characteristic extraction. Random forests or random selection forests are an ensemble gaining knowledge of technique for category, regression and different duties that perform with the aid of using building a mess of selection bushes at training time and outputting the magnificence this is the mode of the instructions for category or imply prediction for regression of the person bushes.

Algorithm 1 Feature Extraction Procedure

Input: Data set as CSV file.

Output: Selected the important attribute Listed.

- 1: Read the dataset.
- 2: Import RandomForestClassifier from sklearn.ensemble.
- 3: Assign the Random Forest Classifier to local variable model.
- 4: Train Rfc =(nestimators=100,randomstate=0,njobs=-1).
- 5: Create clf =sfm(clf,threshold=0.15)
- 6: assign sfm to clf

7: get the important attribute Using the random forest algorithm the data has been split.They split the data for training and testing using the cross validation. The are fit in Random forest algorithm and split using cross validation. The split the train data in 70 percentage and test in 30 percentage.

Algorithm 2 Data Train, Test Split Procedure

- 1: Read the dataset.
- 2: Import RandomForestClassifier from sklearn.ensemble.
- 3: Create Xtest,Xtrain,Ytest,Ytrain.

- 4: Create featable.
- 5: Assign Date,open,close inside featable.
- 6: Assign traintestsplit(dfx, dfy, testsize=0.2, rs=0).
- 7: X,Y are fit using Randomforestclassifier.
- 8: create clf variable and fit randomforest in that variable
- 9: for feature in featable do
- 10: Print feature

11: get the train data and test data This algorithm is used create the independent data set X and store the data in the variable dates. Create the dependent data set y and store the data in the variable prices. Both can be done by appending the data to each of the lists. The independent data set we want only the day from the date, so use the split function to get just the day and cast it to an integer while appending the data to the dates list. Support Vector Regression is used predict the result, using SVR train the dataset to get the accuracy of the prediction and linear regression is also used to approach to modeling the relationship between a scalar response or dependent variable and one or more explanatory variables or independent variables. Create a function that uses 3 different 336 Reshma R et al. / Stock Market Prediction Using Machine Learning Techniques.

Algorithm 3 Ticker Data Processing Procedure

- 1: Read the dataset
- 2: Create the list dates and prices
- 3: for date in dates do
- 4: dates append to date.split[o].
- 5: for prices in open do
- 6: prices append to open
- 7: Print the dates

8: Print the prices Support Vector Regression SVR models with three different kernels to see which one performs the best. The function will average three parameters, the dates, prices, and the day that we want to do the prediction on to get the price. first, I will create the three SVR models with three different kernels are linear, polynomial, radial basis function. Also add in the linear regression model.

Algorithm 4 Algorithm Evaluation Procedure

Input: The trained dataset

Output: The predicted open price for the day as the result

- 1: Read the data set
- 2: import SVR from sklearn.SVM
- 3: import matplotlib.pyplot
- 4: Create the linear kernel
- 5: Create the polynomial kernel
- 6: Create the rbf kernel
- 7: Train the linear in dates,prices
- 8: Train the polynomial in dates,prices
- 9: Train the rbf in dates ,prices
- 10: Create the linear regression
- 11: Train the linear regression
- 12: plot the days in Xlabel
- 13: plot the price in Ylabel
- 14: plot dates and prices in poly and linear and rbf
- 15: Return rbf predicted result.

CHAPTER 11

APPLICATION AND FUTURE ENHANCEMENT

11.1 APPLICATION

Machine learning has significantly impacted stock market analysis by offering powerful tools to analyze data, identify patterns, and make predictions. Here are some applications of machine learning in stock market analysis:

Predictive Modeling: Machine learning algorithms can be trained on historical stock price data to predict future price movements. Techniques such as regression, time series analysis, and ensemble methods like Random Forest or Gradient Boosting can be employed for accurate predictions

Sentiment Analysis: Natural Language Processing (NLP) techniques are used to analyze news articles, social media posts, and other textual data to gauge market sentiment. Sentiment analysis can provide insights into how news and public opinion influence stock prices.

Algorithmic Trading: Machine learning algorithms can automate trading decisions based on predefined criteria and market signals. These algorithms can execute trades at high speeds and frequencies, taking advantage of small price discrepancies and market inefficiencies.

Risk Management: Machine learning models can assess the risk associated with different investment strategies or portfolios. By analyzing historical data and market trends, these models can estimate the probability of financial loss under various scenarios.

Portfolio Optimization: Machine learning algorithms can optimize investment portfolios by selecting the best combination of assets to achieve specific objectives such as maximizing returns or minimizing risk. Techniques like Markowitz's mean-variance optimization or advanced optimization algorithms can be applied.

Anomaly Detection: Machine learning models can detect anomalies in stock market data, such as sudden price changes or unusual trading volumes. These anomalies may indicate potential opportunities or risks in the market.

Pattern Recognition: Machine learning algorithms can identify recurring patterns in stock price data, such as chart patterns or technical indicators. Traders can use these patterns to make informed decisions about buying or selling stocks.

Market Microstructure Analysis: Machine learning techniques can analyze the microstructure of financial markets, including order book data and trade execution data. This analysis can provide insights into market dynamics, liquidity, and price formation processes.

Reinforcement Learning: Reinforcement learning algorithms can be used to develop trading strategies that adapt to changing market conditions. These algorithms learn from experience by interacting with the market and receiving feedback on their actions.

News Impact Analysis: Machine learning models can quantify the impact of news events on stock prices by analyzing the relationship between news articles and price movements. This analysis can help investors understand the drivers of market volatility and make better-informed decisions.

Overall, machine learning has revolutionized stock market analysis by enabling data-driven decision-making, enhancing predictive accuracy, and uncovering valuable insights from vast amounts of financial data.

11.2 FUTURE ENHACEMENT

In this paper, several deep learning models are used including MLP model, LSTM model, CNN model and UA model to predict the one-day-ahead closing price of three stock indices traded in different financial markets. We select SP500 index traded in the U.S financial market, CSI300 index traded in China mainland financial market and Nikkei225 index traded in Tokyo financial market.

SP500 index represents the most developed financial market with a sophisticated trading system while the other two indices represent the less developed financial market and developing financial market, respectively. In each market, seven variables are selected as the inputs including daily trading data, technical indicators and macroeconomic variables. In the MLP model, we design for four hidden layers and the corresponding neurons are 70, 28, 14, 7 in each hidden layer.

In the LSTM model, the unit of the hidden layer is designed to be 140 and one neuron in the output state. In the CNN model, three convolution layers with the corresponding channels 7, 5, 3 of kernel size 3 are chosen as the encoder. Drop layer is applied after each convolution layer. In the UA model, the hidden units of two RNNs are set to be 70. The embedding size of v is 7.

One fully connected layer is used to output the final predictions. The time step was set to be twenty. MAPE was set to be the main predictive accuracy measurement. From the results, UA model which is attention-based deep learning method has the best performance in stock index prediction.

Among the four alternative models, the UA model has the smallest MAPE in all the three stock indices and it could explain the non-linear relationships and allocate more contribution to more important variables in financial time series

along with long term prediction. Furthermore, all of the models perform differently in the three financial markets. The results show that all the four models have better performance in the most developed financial market with the SP500 index than the developing financial market.

Predicting financial time series is a tough work due to its low signal-noise ratio and there is too much noise in this kind of series. The neural network has many advantages in explaining the non-linear relationships in time series.

In the future, we might consider combining linear and non-linear models to build a new model in stock predicting such as using an exponential smoothing method to fit the linear part in financial time series and using the neural network to fit the non-linear part. In exponential smoothing, some specific neural network could also be used to estimate the coefficients.

In financial time series, there might be lots of indicators which could influence the trend of the stock price. Selecting proper indicators is another problem researcher may face. Recently, unsupervised learning has been popular in deep learning.

An appropriate model could be constructed by using an unsupervised learning method so that the model could extract vital information in many indicators to reduce the dimension in the inputs and reduce the parameters for training.

CHAPTER 12

CODING

```
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

#for offline plotting
from plotly.offline import download_plotlyjs, init_notebook_mode, plot,
init_notebook_mode(connected=True)
tesla=pd.read_csv('C:/Users/lenovo/Music/datasetsandcodefilesstockmark
etprediction/tesla.csv')

tesla.head()

tesla=pd.read_csv('C:/Users/lenovo/Music/datasetsandcodefilesstockmark
etprediction/tesla.csv')
tesla.head()

tesla['Date'] = pd.to_datetime(tesla['Date'])
print(f'Dataframe contains stock prices between {tesla.Date.min()}
{tesla.Date.max()}')
print(f'Total days = {(tesla.Date.max() - tesla.Date.min()).days} days')
```

Setting the layout for our plot

```
layout = go.Layout(  
    title='Stock Prices of Tesla',  
    xaxis=dict(  
        title='Date',  
        titlefont=dict(  
            family='Courier New, monospace',  
            size=18,  
            color='#7f7f7f'  
        )  
    ),  
    yaxis=dict(  
        title='Price',  
        titlefont=dict(  
            family='Courier New, monospace',  
            size=18,  
            color='#7f7f7f'  
        )  
    )  
)  
  
tesla_data = [{'x':tesla['Date'], 'y':tesla['Close']}]  
plot = go.Figure(data=tesla_data, layout=layout)  
  
#plot(plot)  
#plotting offline  
iplot(plot)
```


Building the regression model

```
from sklearn.model_selection import train_test_split
```

#For preprocessing

```
from sklearn.preprocessing import MinMaxScaler
```

```
from sklearn.preprocessing import StandardScaler
```

#For model evaluation

```
from sklearn.metrics import mean_squared_error as mse
```

```
from sklearn.metrics import r2_score
```

#Split the data into train and test sets

```
X = np.array(tesla.index).reshape(-1,1)
```

```
Y = tesla['Close']
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3,  
random_state=101)
```

#Creating a linear model

```
lm = LinearRegression()
```

```
lm.fit(X_train, Y_train)
```

#Plot actual and predicted values for train dataset

```
trace0 = go.Scatter(
```

```
    x = X_train.T[0],
```

```
    y = Y_train,
```

```
    mode = 'markers',
```

```
    name = 'Actual'
```

```
)
```

```
trace1 = go.Scatter(
```

```

x = X_train.T[0],
y = lm.predict(X_train).T,
mode = 'lines',
name = 'Predicted'
)
tesla_data = [trace0,trace1]
layout.xaxis.title.text = 'Day'
plot2 = go.Figure(data=tesla_data, layout=layout)

```

#Calculate scores for model evaluation

```

scores = f"""
{'Metric'.ljust(10)}{'Train'.center(20)}{'Test'.center(20)}
{'r2_score'.ljust(10)}{r2_score(Y_train,
lm.predict(X_train))}\t{r2_score(Y_test, lm.predict(X_test))}
{'MSE'.ljust(10)}{mse(Y_train, lm.predict(X_train))}\t{mse(Y_test,
lm.predict(X_test))}
"""
print(scores)
X_train = []
y_train = []

for i in range (60,1149): #60 : timestep // 1149 : length of the data
    X_train.append(trainData[i-60:i,0])
    y_train.append(trainData[i,0])

X_train,y_train = np.array(X_train),np.array(y_train)
model = Sequential()

```

```

model.add(LSTM(units=100, return_sequences = True, input_shape
=(X_train.shape[1],1)))
model.add(Dropout(0.2))

model.add(LSTM(units=100, return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units=100, return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units=100, return_sequences = False))
model.add(Dropout(0.2))

model.add(Dense(units =1))
model.compile(optimizer='adam',loss="mean_squared_error")
plt.plot(hist.history['loss'])
plt.title("Training model loss")
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train'], loc='upper left')
plt.show()
testData = pd.read_csv('Google_test_data.csv')
testData["Close"]=pd.to_numeric(testData.Close,errors='coerce')
testData = testData.dropna()
testData = testData.iloc[:,4:5]
y_test = testData.iloc[60:,0:].values
#input array for the model
inputClosing = testData.iloc[:,0:].values
inputClosing_scaled = sc.transform(inputClosing)

```

```

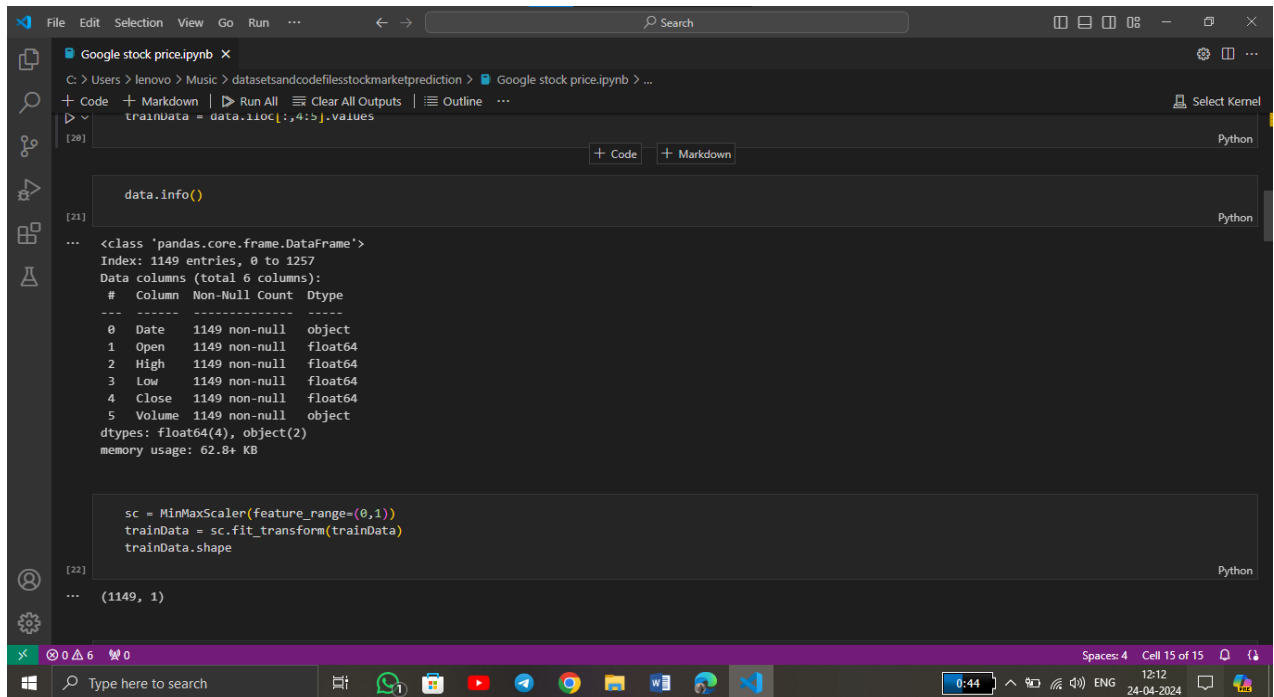
inputClosing_scaled.shape
X_test = []
length = len(testData)
timestep = 60
for i in range(timestep,length):
    X_test.append(inputClosing_scaled[i-timestep:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
X_test.shape
plt.plot(y_test, color = 'red', label = 'Actual Stock Price')
plt.plot(predicted_price, color = 'green', label = 'Predicted Stock Price')
plt.title('Google stock price prediction')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.legend()
plt.show()

```

CHAPTER 13

OUTPUT

13.1 SETTING THE DATA FRAME



The screenshot shows a Jupyter Notebook interface with a dark theme. The file name is 'Google stock price.ipynb'. The current cell is [20] and contains the code `trainData = data.iloc[:,4:5].values`. The output of this cell is not visible. The next cell is [21] and contains the code `data.info()`. The output of this cell is a detailed summary of the data frame, including the index range (0 to 1257), the number of columns (6), and the data types of each column. The output is as follows:

```
<class 'pandas.core.frame.DataFrame'>
Index: 1149 entries, 0 to 1257
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype  
---  -
0    Date        1149 non-null   object  
1    Open        1149 non-null   float64  
2    High        1149 non-null   float64  
3    Low         1149 non-null   float64  
4    Close       1149 non-null   float64  
5    Volume      1149 non-null   object  
dtypes: float64(4), object(2)
memory usage: 62.8+ KB
```

The next cell is [22] and contains the code `sc = MinMaxScaler(feature_range=(0,1))`, `trainData = sc.fit_transform(trainData)`, and `trainData.shape`. The output of this cell is `(1149, 1)`.

FIG 13.1.1 DATA FRAME

13.2 TRAINING THE MODEL

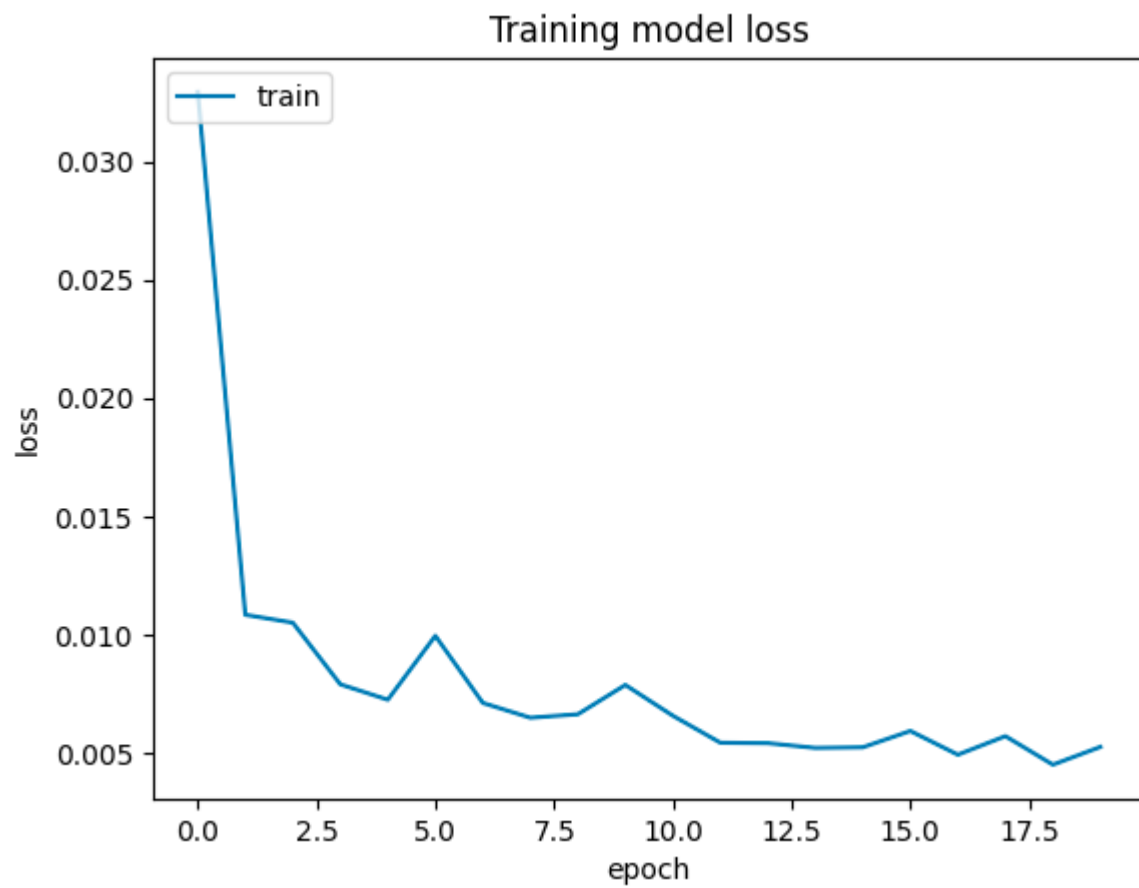


FIG 13.2.1 TRAINING THE MODEL

13.3 GOOGLE STOCK PRICE PREDICTION

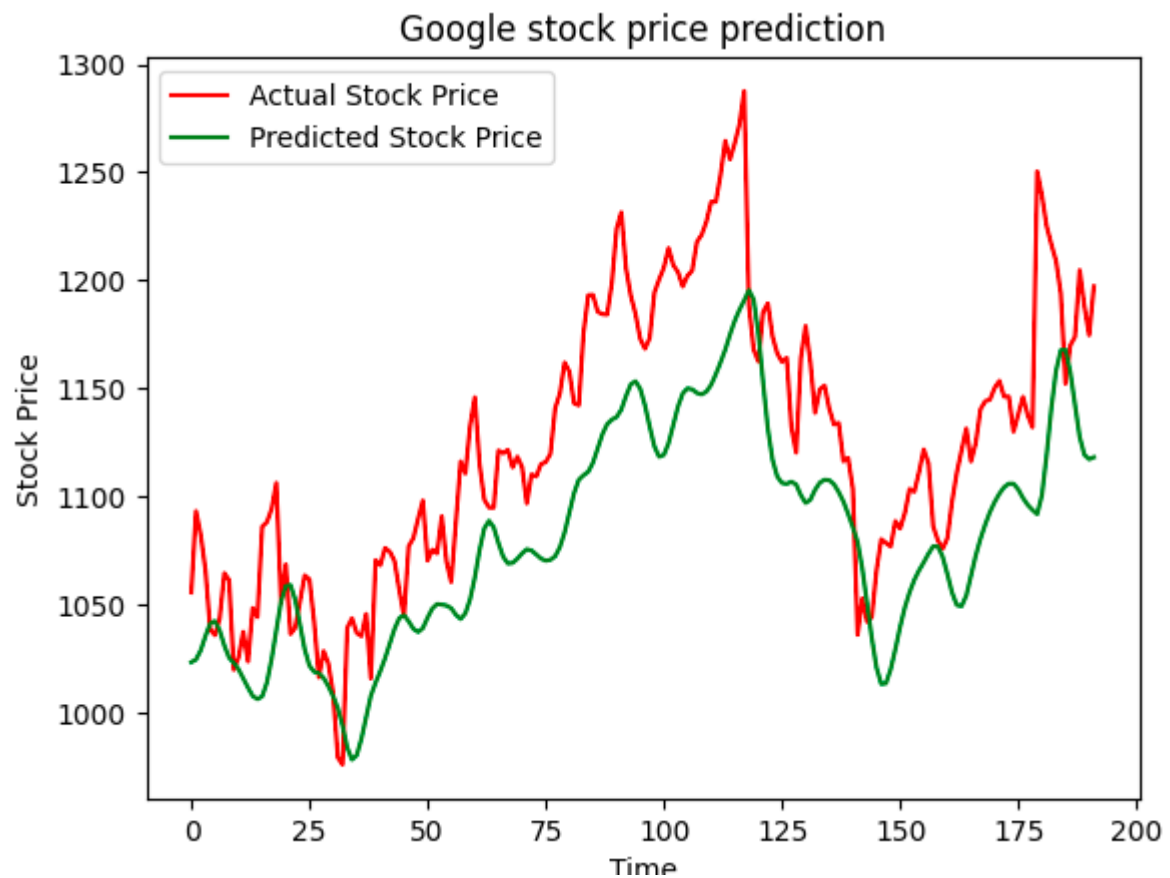


FIG 13.3.1 GOOGLE STOCK PRICE PREDICTION

13.4 STOCK PRICE OF TESLA



FIG 13.4.1 STOCK PRICE OF TESLA

13.5 PREDICTING THE VALUES

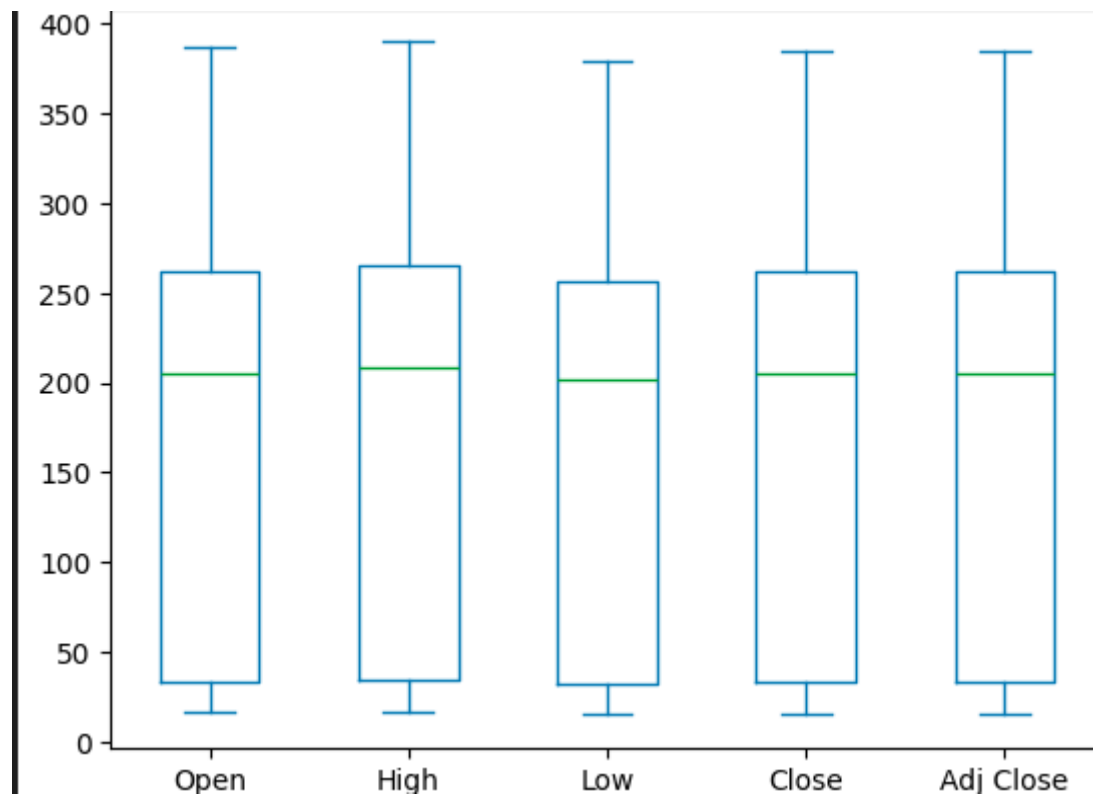


FIG 13.5 PREDICTING THE VALUES

CHAPTER 14

CONCLUSION

This work is concerned with prediction of stock price. Two techniques have been used in this proposed system which is Support vector regression and Linear regression have shown the improvement in accuracy of the prediction by using these two techniques.

There by which leads to the positive result in the prediction. Using the proper algorithm able to predict the stock price with more accuracy. Using machine learning that leads to positive prediction of the stock price. There by which leads to the promising result in the prediction.

Therefore, this project leads to the conclusion that can predict the stock market price with more accuracy using machine learning. In the future the stock market prediction can be further more improved by applying different algorithm to bring more accuracy.

Use real time dataset than the dataset available on public repository that has been used in this work to predict. The more dataset is used our prediction will give more accuracy that can be improved in future work. The stock market plays a remarkable role in our daily lives.

It is a significant factor in a country's GDP growth. In this tutorial, you learned the basics of the stock market and how to perform stock price prediction using machine learning.

Do you have any questions related to this tutorial on stock prediction using machine learning? In case you do, then please put them in the comments section. Our team of experts will help you answer your questions.

If you are interested in learning further about Machine Learning, including the various ML applications across industries, do explore Sampliner's Post Graduate Program in AI and Machine Learning in partnership with Purdue University, and in collaboration with IBM.

This comprehensive 12-month program covers everything from Statistics, Machine Learning, Deep Learning, Reinforcement Learning, to Natural Language Programming and more. You get to learn from global experts and at the end of the program walk away with great endorsements from industry and academic leaders and a skillset that is today the most in-demand in organizations across the world.

CHAPTER 15

REFERENCES

1. Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7), e0180944.
2. Tsantekidis, A., Passalis, N., Tefas, A., & Kannianen, J. (2017). Using deep learning for price prediction by exploiting stationary limit order book features. *Expert Systems with Applications*, 83, 688-697.
3. Zhang, Y., Mao, W., & Yang, Y. (2019). A hybrid model for stock price prediction with deep neural networks. *IEEE Access*, 7, 88081-88089.
4. Chen, Q., Song, H., Nie, L., Li, X., & Zhang, W. (2015). Stock price prediction based on LSTM neural network. In *2015 IEEE international conference on big data (Big Data)* (pp. 2823-2824). IEEE.
5. Lim, C. P., & Yang, H. J. (2014). Forecasting stock prices using fuzzy time series. *Applied Soft Computing*, 21, 178-187.
6. Patel, J., & Shah, S. (2015). Stock market prediction using machine learning algorithm. *International Journal of Computer Applications*, 114(14), 22-29.
7. De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.
8. Chen, M., Zhou, Y., Zhang, L., & Chang, L. (2015). Stock market index prediction using neural networks. In *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive*

intelligence and computing (CIT/IUCC/DASC/PICOM) (Vol. 3, pp. 1089-1092). IEEE.

9. Yu, L., & Yin, J. (2009). A novel hybrid approach to the prediction of stock returns with an application to the Shanghai stock market index. *Expert Systems with Applications*, 36(5), 8849-8861.

10. Sun, Y., Li, Y., & Wu, H. (2020). Stock price prediction model based on attention mechanism. *IEEE Access*, 8, 18523-18530.

11. Wang, J., & Wang, J. (2016). Stock market prediction based on a hybrid model. In 2016 12th World Congress on Intelligent Control and Automation (WCICA) (pp. 7333-7338). IEEE.

12. Zheng, Z., Wang, S., Zhang, L., & Zhao, H. (2014). Stock market index prediction using neural network ensemble. *Neurocomputing*, 139, 28-39.

13. Zhang, Z., Tan, K. C., & Wang, T. (2011). Forecasting stock indices: A comparison of classification and level estimation models in the Chinese stock market. *International Journal of Forecasting*, 27(3), 795-804.

14. Poon, S. H., & Granger, C. W. (2003). Forecasting volatility in financial markets: A review. *Journal of Economic literature*, 41(2), 478-539.

15. Nakkiran, P., Likhitha, K., & Sai, V. V. R. (2015). Stock market price prediction using neural network. *International Journal of Engineering Research & Technology*, 4(12), 259-264.

16. Akita, R., & Geva, T. (2003). Enhancing stock trend prediction using time stamped news articles. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 617-622).

17. Liu, X. Y., Xu, Z. S., & Cheung, Y. M. (2012). Financial time series prediction using ensemble learning. *Neurocomputing*, 80, 64-71.
18. Zhang, L., & Jiang, J. (2004). Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583) (Vol. 2, pp. 1077-1082). IEEE.
19. Shen, J., & Li, X. (2009). Stock market prediction based on neural networks and recurrent fuzzy system. *Expert Systems with Applications*, 36(3), 7312-7319.
20. Lee, Y. W., & Kim, J. H. (2003). Stock price prediction using reinforcement learning. *International Journal of Computational Intelligence and Applications*, 3(02), 209-220.
21. Hsiao, C. C., & Wan, C. C. (2018). Forecasting stock prices using a novel hybrid model based on CNN and GRU. *Expert Systems with Applications*, 114, 532-543.
22. Zhang, Y., Mao, W., & Yang, Y. (2018). Stock price prediction using attention-based multi-input LSTM. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 4328-4334).
23. Patel, N., & Shah, S. (2016). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, 61, 33-43.
24. Yeh, C. C., & Liao, L. T. (2019). A stock price prediction model using a hybrid deep learning strategy. *Expert Systems with Applications*, 126, 155-163.

25. Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In Proceedings of the international joint conference on neural networks (Vol. 1, pp. 1-6).
26. Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, 38(8), 10389-10397.
27. Xie, Y., & Liu, Y. (2018). Stock price prediction using reinforcement learning. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 1296-1301). IEEE.
28. Wang, J., & Wang, J. (2017). Stock market forecasting with big data and machine learning techniques. In 2017 13th IEEE International Conference on Electronic Measurement & Instruments (pp. 1252-1257). IEEE.
29. Tai, K., & Chau, K. W. (2017). A review on the forecasting of exchange rate: A perspective from forecasting using artificial neural networks. *International Journal of Electrical Power & Energy Systems*, 88, 188-197.