A

Mini Project Report

On

Stock Price Fluctuations using Machine Learning

Submitted In partial fulfillment of the Requirements for the award of degree of

Bachelor of Technology

In

COMPUTER SCIENCE & ENGINEERING

By

Name	Roll no
K. PRIYAVARSHITHA	22681A0552
K. INRAJU	22681A0545
E. MADHU	22681A0527
B. SHIVA	22681A0512

Under the Esteemed Guidance of

Dr. P. U. ANITHA
(Associate Professor)

Department of Computer Science & Engineering

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana

2024-2025

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon -506167, Telangana

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2024-2025

CERTIFICATE

*This is to certify that is a Bonafide record of project work Entitled **"Stock Price Fluctuations using Machine Learning"** Carried out by **K. PRIYAVARSHITHA (22681A0552), K. INRAJU (22681A0545), E. MADHU (22681A0527), B. SHIVA (22681A0512)** during academic year 2024-2025, in partial fulfillment of the requirement of the award of degree of Bachelor of Technology in Computer Science & Engineering offered by Christu Jyothi Institute of Technology & Science, Yeshwanthapur, Jangaon.*

Co-Guide	**Project Guide**	**HOD-CSE**
Mr. P. Rajashekar	**Dr. P. U. ANITHA**	**Mr. M. RAMARAJU**
Assistant Professor	**Associate Professor**	**Assistant Professor**

External Examiner

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

2024-2025

DECLARATION

*We hereby declare that the document entitled **"Stock Price Fluctuations using Machine Learning"** submitted to the **Christu Jyothi Institute of Technology & Science** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering is a record of an original work done by us under the guidance of **DR. P. U. ANITHA** and **Mr. P. Rajashekar** this document has not been submitted to any other university for the award of any other degree.*

K. PRIYAVARSHITHA	**22681A0552**
K. INRAJU	**22681A0545**
E. MADHU	**22681A0527**
B. SHIVA	**22681A0512**

ACKNOWLEDGEMENT

On the submission of our project entitled ***"Stock Price Fluctuations using Machine Learning*"**, we would like to extend our gratitude and sincere thanks to our guide ***Dr. P. U. ANITHA*** *and co-guide* ***Mr. P. Rajashekar,*** Department of **Computer Science and Engineering** for their valuable and timely suggestions.

We would also like to extend our sincere thanks to **Mr. M. RAMARAJU**, Assistant Professor, Head of the Department, COMPUTER SCIENCE & ENGINEERING for his valuable suggestions and motivating guidance during our project work.

We would like to thank our director, **Rev.Fr. D. VIJAYA PAUL REDDY** for giving us the opportunity to carry out our project work.

We endow our sincere thanks to Principal **Dr. S. CHANDRASHEKAR REDDY** for his consistent cooperation and encouragement.

We are very thankful to our teachers for providing the required background during the project work. We would also like to extend our gratitude to our friends and those who are directly or indirectly helped us in completing our project work.

K. PRIYAVARSHITHA	**22681A0552**
K. INRAJU	**22681A0545**
E. MADHU	**22681A0527**
B. SHIVA	**22681A0512**

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision and Mission of the Institute

Vision

To admit and groom students from rural background and be a truly technical institution, benefiting society and nation as a whole institute.

Mission

The mission of the institute is to create, deliver and refine knowledge. Being a rural technical institute, we aim to:

1. Enhance our position to one of the best technical institutions and to measure our performance against the highest defined standards.
2. Provide highest quality learning environment to our students for their greater well-being so as to equip them with highest technical and professional ethics.
3. Produce engineering graduates fully equipped to meet the ever-growing needs of industry and society.

[illegible]

PRINCIPAL

Principal
Christu Jyothi Institute of Technology & Science
Colombo Nagar, Yeshwanthapuram (VM)
Jangaon(Mdl), Jangaon (Dist)-506167.

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY& SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Department Vision and Mission

Vision

To be a center of eminence to mold young, fresh minds into challenging computer science professionals with ethical values.

Mission

1. Enrich the knowledge and wisdom with repository of books and modernized laboratory aided by dedicated resources.
2. Organize training and activities on upcoming techniques, and inter-personal Skills
3. Develop the ability to provide sustainable solutions to real world situations with collaboration

[illegible]

Head of Department

Head of the Department
Dept. of Computer Science & Engineering
Christu Jyothi Institute of Technology & Science
Colombo Nagar, Yeshwanthapuram (Vill),
Jangaon(Mdl), Warangal (Dist)-506167

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program Educational Objectives

PEO1	Graduates of B. Tech (CSE) are able to Formulate, Analyze and solve hardware and software problems within the constraints and pursue research.
PEO2	Demonstrate knowledge in core areas of computer science and relate engineering to comprehend engineering trade-offs to create novel products.
PEO3	Show the awareness of life-long learning needed for a successful professional career and exhibit ethical values, excellence, leadership and social responsibility.

[illegible]

Head of Department

Head of the Department
Dept. of Computer Science & Engineering
Christu Jyothi Institute of Technology & Science
Colombo Nagar, Yeshwanthapuram (Vill),
Jangaon(Mdl), Warangal (Dist)-506167

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program outcomes

PO No	Program Outcomes
PO1	Engineering knowledge apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
PO3	Design/development of solutions design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tools usage Create, Select, and apply appropriate techniques resources, and modern engineering and IT tools including predictions and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society apply reasoning informed by the contextual knowledge to assess societal and environment contexts, and demonstrate the knowledge of, and need for sustainable development.
PO7	Environment and sustainability understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
P09	Individual and team work function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
PO10	Communication communicates effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations.
PO11	Project management and finance demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning recognizes the need for, and have the preparation and ability to engage and lifelong learning in the broadest context of technology change.

[illegible]

Head of Department

Head of the Department
Dept. of Computer Science & Engineering
Christu Jyothi Institute of Technology & Science
Colombo Nagar, Yeshwanthapuram (Vill),
Jangaon(Mdl), Warangal (Dist)-506167

CHRISTU JYOTHI INSTITUTE OF TECHNOLOGY & SCIENCE

Colombo Nagar, Yeshwanthapur, Jangaon-506167, Telangana.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program Specific Outcomes

PSO1 Proficiency skill:

Understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics and networking for efficient design of computer-Based systems of varying complexity.

PSO2 Problem-Solving Skills:

Apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO3 Successful Career and Entrepreneurship:

Employ modern computer language, environments and platforms in creating innovative career paths to be entrepreneur and a zest for higher studies.

[illegible]

Head of Department

Head of the Department
Dept. of Computer Science & Engineering
Christu Jyothi Institute of Technology & Science
Colombo Nagar, Yeshwanthapuram (Vill),
Jangaon(Mdl), Warangal (Dist)-506167

ABSTRACT

The relationship between financial indicators and stock price fluctuations has long been a focal point in financial research, with implications for investment strategies and risk management. This project leverages advanced artificial intelligence (AI) systems to conduct a comprehensive correlation analysis between key financial indicators—such as earnings per share (EPS), price-to-earnings (P/E) ratio, debt-to-equity (D/E) ratio, and revenue growth—and stock price movements. By employing machine learning algorithms, including regression models, neural networks, and natural language processing (NLP) techniques, the study aims to identify patterns, dependencies, and predictive capabilities of these indicators in forecasting stock price changes. The AI system is trained on historical financial data and real-time market information, enablingss it to adapt to dynamic market conditions. The results of this analysis provide valuable insights into the strength and direction of correlations, offering investors and financial analysts a data-driven approach to decision-making. This research not only enhances the understanding of financial market dynamics but also demonstrates the potential of AI in transforming traditional financial analysis methodologies.

Keywords

Machine Learning, Correlations, Financial data, Real-time market information, Dynamic market conditions

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 PROBLEM STATEMENT

The stock market exhibits complex, non-linear behavior driven by numerous macroeconomic, technical, and psychological factors. Traditional statistical models often fall short in capturing these intricate dynamics, making accurate prediction of stock price movements a longstanding challenge. The lack of reliable forecasting tools limits investors' ability to make informed decisions, emphasizing the need for advanced approaches that can effectively analyze vast, multidimensional financial data.

1.2 BACKGROUND AND MOTIVATION

With the rise of big data and advancements in artificial intelligence, machine learning has emerged as a powerful technique for analyzing financial markets. Its ability to learn from historical patterns, adapt to changing conditions, and handle complex relationships among variables presents new opportunities for market prediction. The motivation behind this project is to harness the predictive capabilities of machine learning to analyze correlations between financial indicators and stock price fluctuations, thereby improving the precision and reliability of forecasting models.

1.3 OBJECTIVES

1. To analyze historical stock data and relevant financial indicators to understand their influence on stock price movements.
2. To identify and select key features that significantly correlate with price changes.
3. To implement and compare various machine learning models for forecasting stock trends.
4. To evaluate model performance using statistical accuracy metrics.
5. To visualize data insights and model outputs for better interpretation.
6. To set the foundation for incorporating real-time data and sentiment analysis in future work.

1.4 PROPOSED APPROACH

The project follows a structured methodology beginning with data collection from reputable financial sources. Preprocessing techniques are applied to clean, normalize, and engineer features, including technical indicators. Various supervised machine learning models, such as linear regression and decision trees are trained and tested on the processed data. The models are evaluated using metrics such as RMSE and R^2 to determine their predictive accuracy. Correlation analysis is conducted to identify influential indicators.

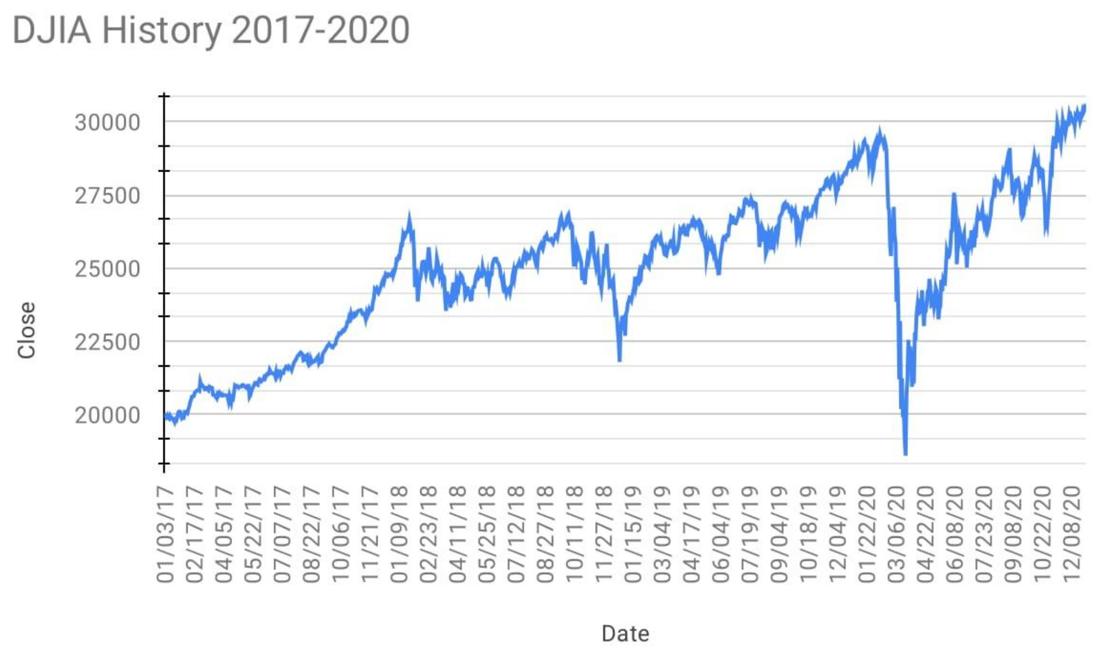


Fig 1.4.1: Stock Price Fluctuations

1.5 HIGHLIGHTS OF WHAT WE HAVE ACHIEVED

1. Successfully implemented multiple ML models and evaluated their performance.
2. Identified key financial indicators with strong correlation to price changes.
3. Provided visual insights into model outputs and data correlations.
4. Established a scalable framework for future enhancements, including real-time prediction and sentiment integration.

2. LITERATURE SURVEY

The literature survey reveals several approaches to enhance stock price fluctuations using Machine learning algorithms.

1) **Proceedings of the 2023 International Conference on Management Research and Economic Development DOI: 10.54254/2754-1169/22/20230307 Research on the Stock Price Prediction Using Machine Learning Yang Shi1,a.** Stock price prediction is a complex and challenging problem that has attracted the attention investors and researchers for decades. In recent years, machine learning algorithms have become powerful tools for predicting stock prices. This paper first introduces four popular machine learning algorithms used for stock price prediction which are linear regression, support vector machines, artificial neural networks and long short-term memory.
2) **Stock Price Prediction Using Machine Learning Strategies, Xinran Chen* Dept. of Mathematics, University of Washington, Seattle, United States.** The fluctuation of stock prices, to some extent, reflects the environment of the global economy – whether it's prosperous or in a recession. Some traditional stock traders discredit the authenticity and accuracy of stock price prediction with modern machine learning techniques, deeming that the complicated nature of stock market deny the possibility of making reliable prediction.

3) **Applying machine learning algorithms to predict the stock price trend in the stock market– The case of Vietnam Tran Phuoc1,2, Pham Thi Kim Anh1,2, Phan Huy Tam3,4 & Chien V. Nguyen.** The aims of this study are to predict the stock price trend in the stock market in an emerging economy. Using the Long Short-Term Memory (LSTM) algorithm, and the corresponding technical analysis indicators for each stock code include: simple moving average (SMA), convergence divergence moving average (MACD), and relative strength index (RSI); and the secondary data from VN-Index and VN-30 stocks, the research results showed that the forecasting model has a high accuracy of 93% for most of the stock data .

4) **Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implication Gaurang Sonkavde, Deepak Sudhakar Dharrao.** The financial sector has greatly impacted the monetary well-being of consumers , and financial institutions. In the current era, artificial intelligence is redefining the limits of the financial markets based on state-of-the-art machine learning and deep learning algorithms.

5) **Stock Market Prediction with High Accuracy using Machine Learning Techniques Malti Bansal*, Apoorva Goyal, Apoorva Choudhary.** Stock market trading is a major and predominant activity when one tells about the financial markets. With the inviable uncertainty and volatility in the prices of the stocks, an investor keeps looking for ways to predict the future trends in order to dodge the losses and make the maximum possible profits. However, it cannot be denied that as of yet there is no such technique to predict the upcoming trends in the markets with complete accuracy, while multiple methods are being explored to improve the predictive performance of models to an extent as large as possible.

3. SYSTEM ANALYSIS

3.1 Existing system

The stock market is the fuzzy environment full of many uncertain factors. The existence of these uncertain factors makes people face various risks when investing in securities. The stock market is a very complex system, which is affected by economic, policy and market factors. In existing system, we are used linear regression algorithm for stock price prediction.

In the existing system, Linear Regression is used as the machine learning algorithm for predicting stock prices.

Linear Regression is a supervised learning algorithm that models the relationship between the independent variables (features) and the dependent variable (stock price) by fitting a straight line (called the regression line).

Key Points

1. It assumes a linear relationship between the input features and the target.
2. Suitable for problems where data shows a trend.
3. Easy to interpret and implement.
4. However, it fails to capture complex patterns or nonlinear dependencies in stock market data.

DISADVANTAGES OF EXISTING SYSTEM:

1. Using linear regression algorithm we cannot predict exact stock price values.
2. It involves very lengthy and complicated procedure of calculations and analysis.

Limitation

Higher Mean Squared Error (MSE) and Mean Absolute Error (MAE) compared to other advanced algorithms.

In our case

MSE = 0.12, which is relatively higher.

Accuracy decreases with an increase in error.

3.2 PROPOSED SYSTEM

This paper uses random forest to establish a mathematical model of stock prices, and uses this model to predict the fluctuation of stock prices. Through comparison with actual data, it is found that the prediction results are basically consistent with it. The stock market is a very complex system, which is affected by economic, policy and market factors. For such a complex dynamic nonlinear system, many researchers use the random forest algorithm to predict stock prices.

The proposed system replaces Linear Regression with the Random Forest algorithm for improved prediction performance.

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the average of predictions for regression tasks.

Key Features

1. Handles nonlinear relationships effectively.
2. Reduces overfitting by averaging results from many trees.
3. Robust and accurate for complex datasets like stock price history.

Advantages

Lower MSE = 0.11, indicating better performance.

Improved accuracy due to reduced error rate.

More adaptable to real-world stock price trends, which are often non-linear and affected by multiple factors.

Comparative Error Rates

Algorithm	MSE	MAE	Accuracy Trend
Linear Regression	0.12	Higher	Lower accuracy
Random Forest	0.11	Lower	Higher accuracy
Support Vector Machine (SVM)	0.13	Higher	Lowest accuracy among the three.

Table 1: Comparative Error Rates

3.3 ALGORITHMS

In this project, we have implemented and compared three different machine learning algorithms to predict stock price fluctuations. The algorithms are evaluated based on performance metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). The model with the lowest error is selected for the proposed system.

3.3.1 Linear Regression

Linear Regression (Used in Existing System)

Linear Regression is a supervised learning algorithm used for predictive analysis. It establishes a linear relationship between the dependent variable (stock price) and one or more independent variables (features such as date, volume, etc.). The algorithm fits a straight line (regression line) that minimizes the difference between the actual and predicted values.

Characteristics

1. Simple and easy to interpret.
2. Assumes a linear relationship between input and output.
3. Sensitive to outliers and noise.

Performance

MSE: 0.12

MAE: Relatively high

Accuracy is moderate, but not suitable for highly volatile or non-linear data like stock prices.

3.3.2 Random Forest

Random Forest (Used in Proposed System)

Random Forest is an ensemble learning algorithm that operates by constructing multiple decision trees and taking the average of their predictions in case of regression. It improves the prediction accuracy and reduces overfitting by combining the outcomes of many weak learners (trees).

Characteristics

1. Handles nonlinear relationships efficiently.
2. Robust to noise and overfitting.
3. Provides better generalization on unseen data.

Performance

MSE: 0.11 (Lowest among all)

MAE: Low

Produces the most accurate predictions in our case, hence used in the proposed system.

3.3.3 Support Vector machine

Support Vector Machine (SVM)

Support Vector Machine is a powerful supervised learning model that can be used for both classification and regression tasks. In Support Vector Regression (SVR), the algorithm attempts to fit the best line within a margin, optimizing for the most accurate and stable predictions.

Characteristics

1. Good at handling high-dimensional data.
2. Effective even when the number of samples is less than the number of features.
3. Sensitive to feature scaling.

Performance

MSE: 0.13 (Highest among the three)

MAE: Higher

Accuracy is lower compared to Random Forest and Linear Regression, so not chosen for final implementation.

Summary of Model Performance

Algorithm	**MSE**	**MAE**	**Final Verdict**
Linear Regression	**0.12**	**Higher**	**Used in Existing System**
Random Forest	**0.11**	**Lower**	**Used in Proposed System**
Support Vector Machine	**0.13**	**Highest**	**Not selected due to higher error**

Table 2 : Summary Of Model Performance

3.4 MODULES DESCRIPTION

1. Data Collection Module

1. **Purpose:** To collect and load historical stock market data for model training and testing.
2. **Description:** This module allows users or the system to import stock price data (e.g., from CSV files, APIs, or databases). It includes:
   a. Historical price information (Open, High, Low, Close, Volume)
   b. Fundamental features (EPS, P/E ratio, etc. if available)
   c. Data from multiple time periods for stronger forecasting
3. **Technologies Used:**
   a. Python (pandas, NumPy)
4. **Data sources:** CSV, Excel, or APIs like Alpha Vantage/Yahoo Finance

2. Data Preprocessing Module

1. **Purpose:** To clean, transform, and prepare the data for machine learning algorithms.
2. **Description:**
   a. Handles missing values
   b. Normalizes/standardizes features
   c. Converts time-series into supervised learning format (e.g., using windowing)
   d. Splits the data into training and testing sets
3. **Technologies Used:**
   a. Python (pandas, scikit-learn)

3. Feature Selection Module

1. **Purpose:** To select the most relevant input features for prediction.
2. **Description:**
    a. Identifies and selects key attributes that influence stock prices
    b. Removes irrelevant/noisy data
    c. May include correlation analysis or feature importance from Random Forest
3. **Technologies Used:**
    a. Python (scikit-learn, seaborn for heatmaps)

4. Model Training Module

1. **Purpose:** To train the machine learning model using the preprocessed data.
2. **Description:**
    a. Trains selected ML algorithms (e.g., Random Forest, Neural Network)
    b. Hyperparameter tuning to optimize performance
    c. Stores the trained model for future predictions
3. **Algorithms:**
    a. Random Forest (main)
    b. Compared with Linear Regression for benchmarking
4. **Technologies Used:**
    a. Python (scikit-learn, keras / tensorflow)

5. Prediction Module

1. **Purpose:** To forecast future stock prices based on the trained model.
2. **Description:**
    a. Accepts new input data
    b. Uses the trained model to generate price predictions
    c. Outputs predicted price vs. actual price
3. **Technologies Used:**
    a. Python (scikit-learn, matplotlib for graph)

6. Visualization Module

1. **Purpose:** To display prediction results in an easy-to-understand format.
2. **Description:**
    a. Graphical representation of predicted vs actual values
    b. Line graphs, bar charts, and performance metrics
    c. Supports both historical trends and future forecasts
3. **Technologies Used:**
    a. Python (matplotlib, seaborn)

7. Performance Evaluation Module

1. **Purpose:** To evaluate how well the model is performing.
2. **Description:**

   Calculates metrics like:

    a. Mean Absolute Error (MAE)
    b. Root Mean Square Error (RMSE)
    c. R^2 Score
    d. Compares different algorithms (Random Forest vs Linear Regression, etc.)
3. **Technologies Used:**
    a. Python (scikit-learn metrics)

4. SOFTWARE ARCHITECTURE

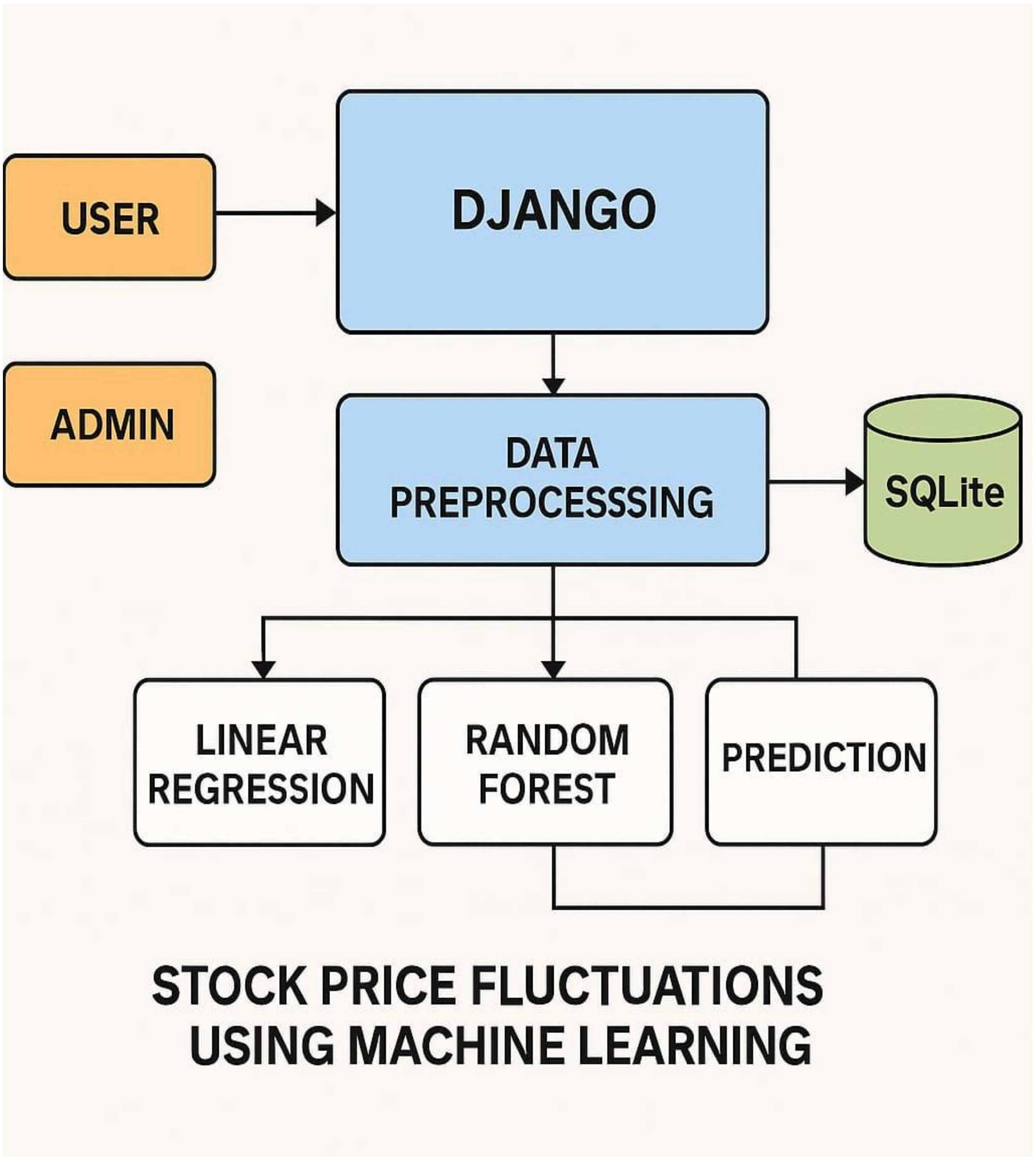


Fig 4.0: Stock Price Fluctuations

Explanation of Components:

1. **USER / ADMIN**: Interfaces for different types of users.
2. **DJANGO**: The web framework handling the application's logic and user interaction.
3. **DATA PREPROCESSING**: Cleans and structures raw stock data.
4. **SQLite**: A lightweight relational database to store the cleaned and processed data.
5. **LINEAR REGRESSION & RANDOM FOREST**: Machine learning models used to analyze and learn patterns in stock price data.
6. **PREDICTION MODULE**: Uses the output from ML models to make future stock price predictions.

4.1 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data moves through a system. It shows the flow of information between processes, data stores, external entities, and data inputs/outputs. DFDs help in understanding how the system operates, what transformations are performed on the data, and how the system components interact**.**

DFDs are commonly developed in levels:

Level 0 DFD – Context Diagram:

Entities:

- User/Analyst

- Stock Market Data Source (e.g., Yahoo Finance, NSE, etc.)

Process:

- Stock Price Prediction System

Data Flows:

- User sends request for stock prediction

- System fetches historical data from external sources

- System returns predicted price/movement to the user

Level 1 DFD – Detailed Flow:

Processes:

1. Data Collection

2. Data Preprocessing

3. Feature Extraction

4. Model Training & Prediction

5. Result Evaluation & Visualization

External Entities:

- User

- Financial Data Source (APIs, CSV files, databases)

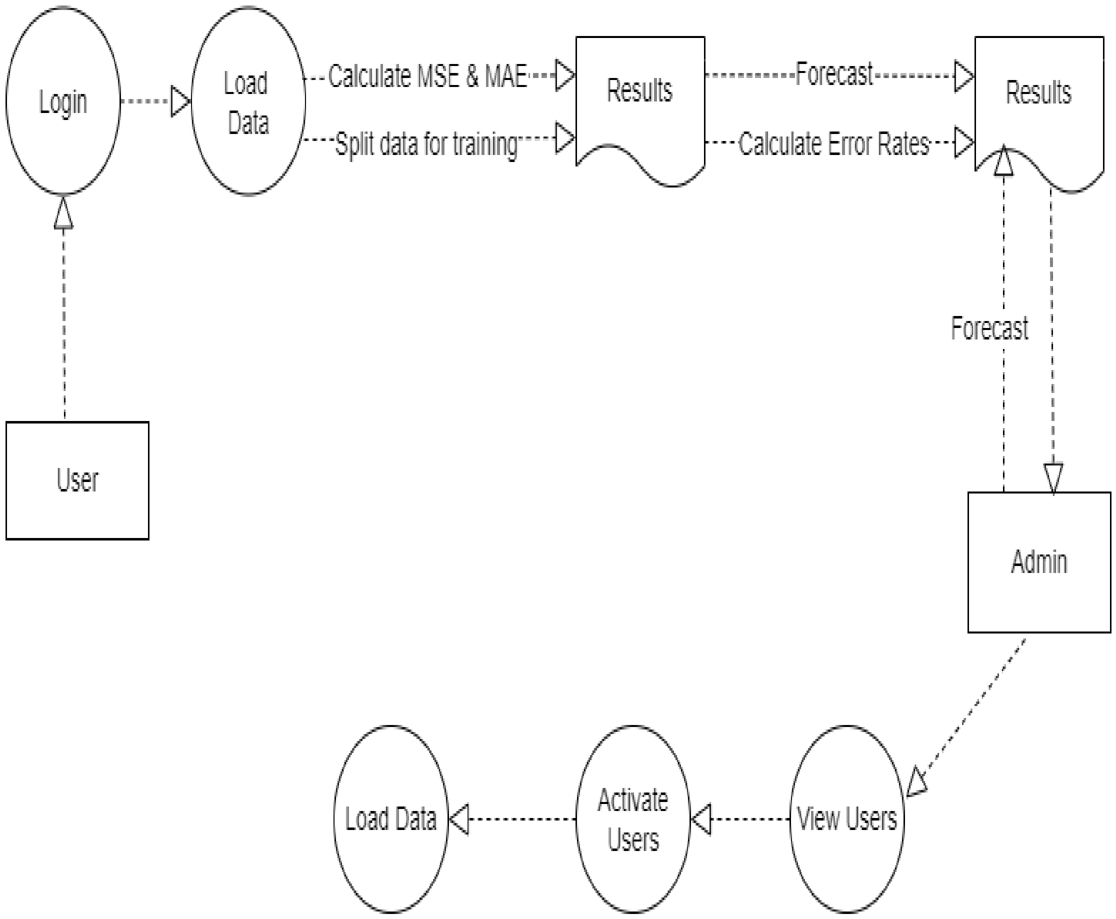


Fig 4.1 Data flow diagram

4.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The goal is for UML to become a common language for creating models of object-oriented computer software. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.1.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

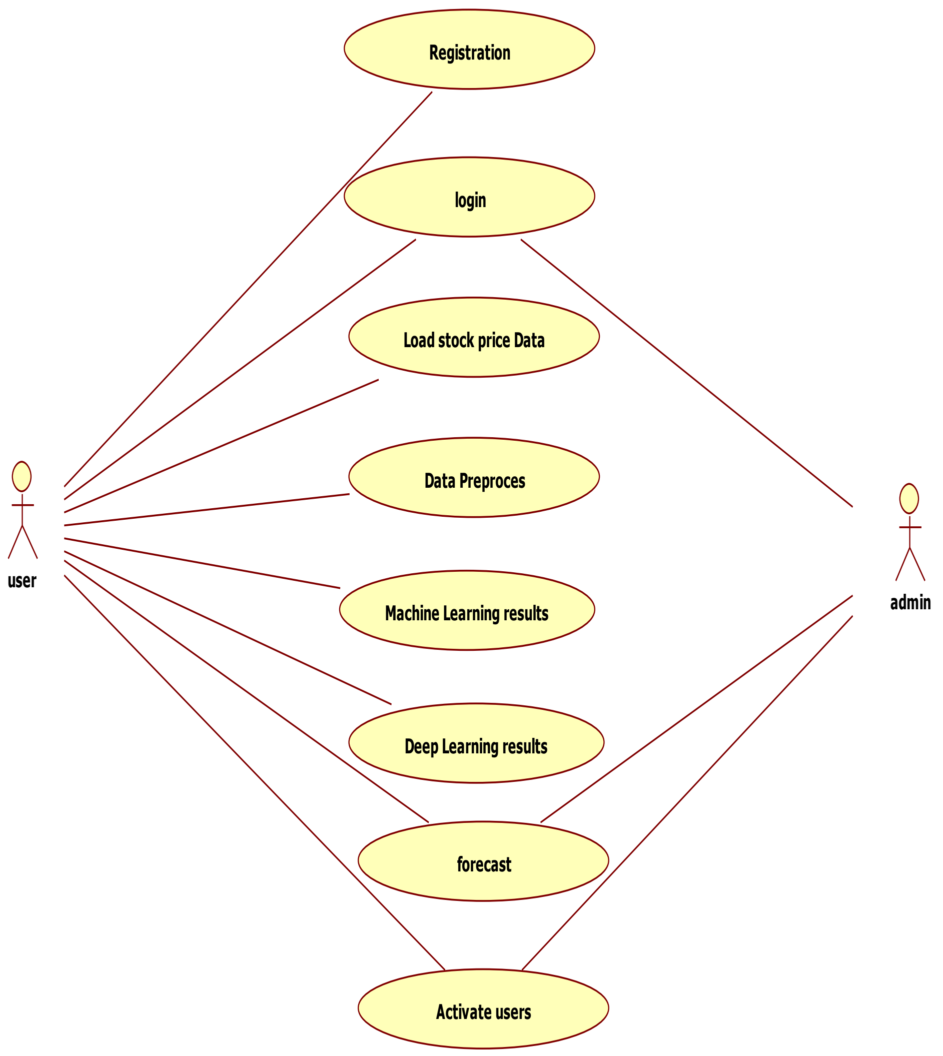


Fig 4.2.1 Use case Diagram

4.2.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

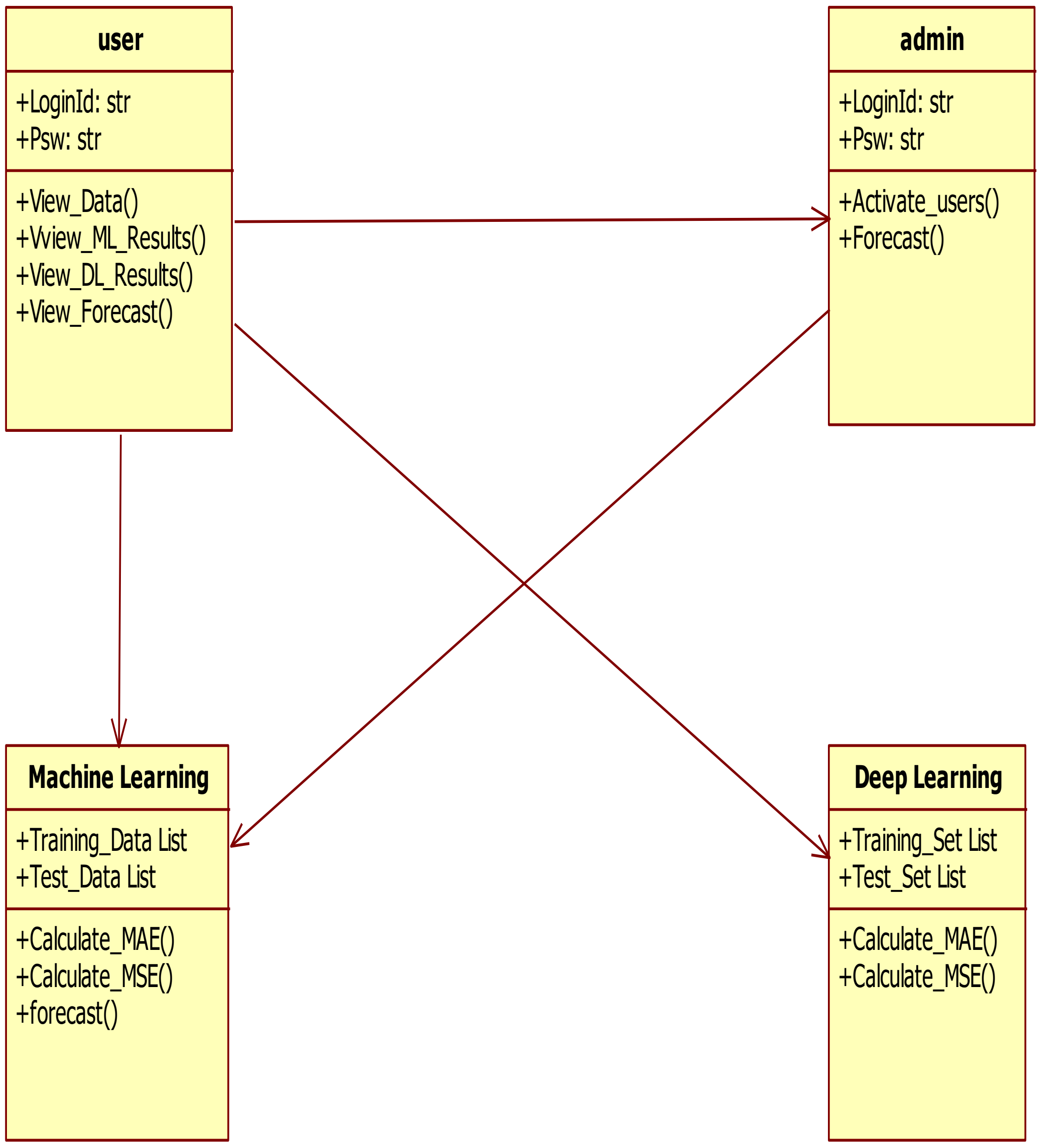


Fig 4.2.2 Class Diagram

4.1.2 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

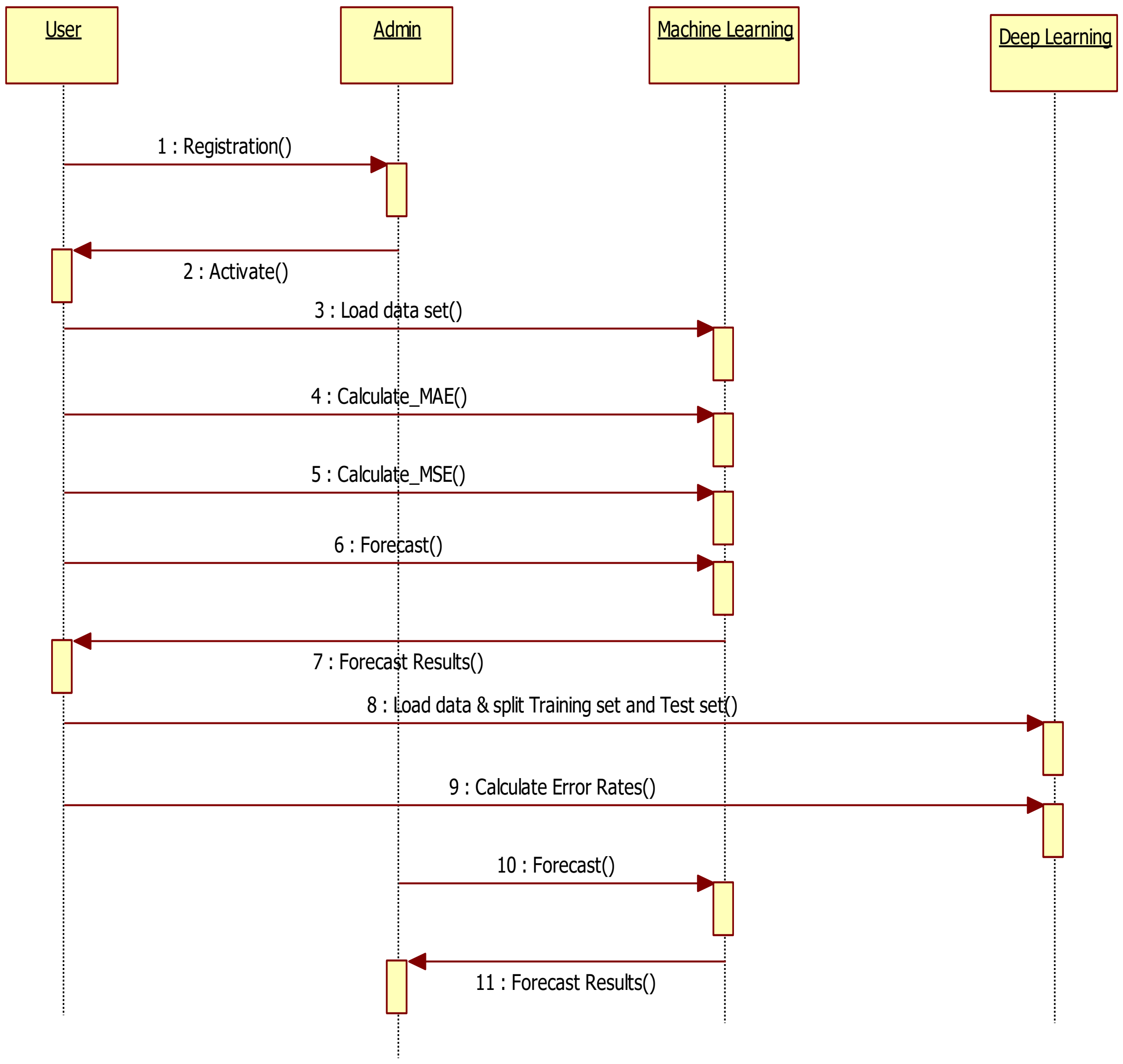


Fig 4.2.3 Sequence Diagram

4.1.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

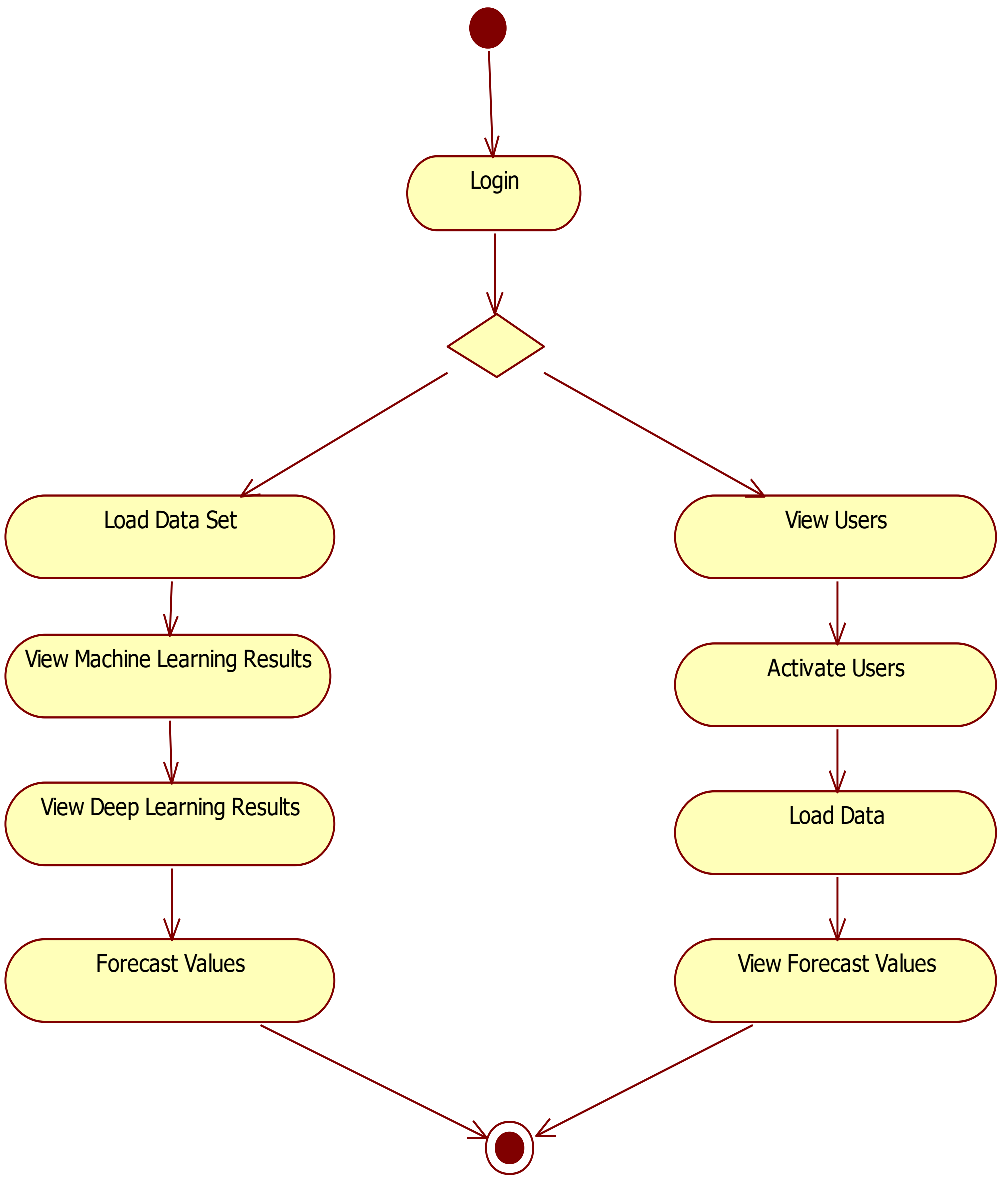


Fig 4.3.4 Activity Diagram

5. SOFTWARE AND HARDWARE REQUIREMENTS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION

5.1 Software Requirements

For developing the application, the following are the Software Requirements:

1. Python
2. Django

Tools Used:

1. PyCharm
2. Visual Studio Code

Operating Systems supported

1. Windows 10 64-bit OS

Debugger and Emulator

1. Any Browser (Particularly Chrome)

5.2 Hardware Requirements

1. Processor: Intel Core i3
2. RAM: 8 GB
3. Space on Hard Disk: minimum 1 TB

6.CODING

6.1 SAMPLE CODE

FORMS.PY

```
from django import forms
from .models import UserRegistrationModel

class UserRegistrationForm(forms.ModelForm):
name = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-zA-Z]+'}),
required=True, max_length=100)
loginid = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-zA-Z]+'}),
required=True, max_length=100)
password = forms.CharField(widget=forms.PasswordInput(attrs={'pattern':
'(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}',
'title': 'Must contain at least one number and one uppercase and lowercase letter, and at
least 8 or more characters'}),
required=True, max_length=100)
mobile = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[56789][0-9]{9}'}),
required=True,
max_length=100)
email = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-z0-9._%+-]+@[a-
z0-9.-]+\.[a-z]{2,}$'}),
required=True, max_length=100)
locality = forms.CharField(widget=forms.TextInput(), required=True, max_length=100
address = forms.CharField(widget=forms.Textarea(attrs={'rows': 4, 'cols': 22}),
required=True, max_length=250)
city = forms.CharField(widget=forms.TextInput(
attrs={'autocomplete': 'off', 'pattern': '[A-Za-z ]+', 'title': 'Enter Characters Only '}),
required=True,
max_length=100)
state = forms.CharField(widget=forms.TextInput(
attrs={'autocomplete': 'off', 'pattern': '[A-Za-z ]+', 'title': 'Enter Characters Only '}),
required=True,
max_length=100)
status = forms.CharField(widget=forms.HiddenInput(), initial='waiting',
max_length=100)

class Meta():
model = UserRegistrationModel
fields = '__all__'
```

MODELS.PY

```
from django.db import models

# Create your models here.
class UserRegistrationModel(models.Model):
name = models.CharField(max_length=100)
loginid = models.CharField(unique=True, max_length=100)
password = models.CharField(max_length=100)
mobile = models.CharField(unique=True, max_length=100)
email = models.CharField(unique=True, max_length=100)
locality = models.CharField(max_length=100)
address = models.CharField(max_length=1000)
city = models.CharField(max_length=100)
state = models.CharField(max_length=10
status = models.CharField(max_length=100)
def __str__(self):
return self.loginid

class Meta:
db_table = 'UserRegistrations'
from django.db import models
```

VIEWS.PY

```
from django.shortcuts import render

# Create your views here.
from django.shortcuts import render,HttpResponse
from django.contrib import messages
from .forms import UserRegistrationForm
from .models import UserRegistrationModel

# Create your views here.
def UserRegisterActions(request):
if request.method == 'POST':
form = UserRegistrationForm(request.POST
if form.is_valid()
print('Data is Valid')
form.save()
messages.success(request, 'You have been successfully registered')
form = UserRegistrationForm()
return render(request, 'UserRegistrations.html', {'form': form})
else:
messages.success(request, 'Email or Mobile Already Existed')
print("Invalid form")
elseform = UserRegistrationForm()return render(request, 'UserRegistrations.html',
{'form': form})
def UserLoginCheck(request):
if request.method == "POST":
loginid = request.POST.get('loginid')
pswd = request.POST.get('pswd')
print("Login ID = ", loginid, ' Password = ', pswd)
try:
check = UserRegistrationModel.objects.get(loginid=loginid, password=pswd)
status = check.status
```

```
print('Status is = ', status)
if status == "activated":
request.session['id'] = check.id
request.session['logged user'] = check.name
request.session['loginid'] = loginid
request.session['email'] = check.email
print("User id At", check.id, status)
return render(request, 'users/UserHomePage.html', {})
else:
messages.success(request, 'Your Account Not at activated')
return render(request, 'UserLogin.html')
except Exception as e:
print('Exception is ', str(e))
pass
messages.success(request, 'Invalid Login id and password')
return render(request, 'UserLogin.html', {})

def UserHome(request):
return render(request, 'users/UserHomePage.html', {})

def user_view_data(request):
import pandas as pd
from django.conf import settings
import os
path =os.path.join(settings.MEDIA_ROOT, 'ada.csv')
df = pd.read_csv(path)
df = df[['Date','Open', 'High', 'Low', 'Close']]
# df = df.head(10)
# df = df.head(100).to_html
df = df.to_html

return render(request, 'users/view_data.html',{'data': df})
```

```
def user_ml(request):
from .utility import processml
mae, mse, r2 = processml.calc_linear_regression()
rf_mae, rf_mse, rf_r2 = processml.calc_random_forest_regressor()
svm_mae, svm_mse, svm_r2 = processml.calc_svm()
lr = {'mae': mae,'mse': mse,"r2": r2}
rf = {'rf_mae': rf_mae, 'rf_mse': rf_mse, "rf_r2": rf_r2}
svm = {'svm_mae': svm_mae, 'svm_mse': svm_mse, "svm_r2": svm_r2}
return render(request,'users/ml_result.html',{"lr": lr, "rf": rf, "svm":svm} )

def user_deep_learning(request):
from .utility import deep_learnig_model
train_score,train_rmse,test_score,test_rmse =
deep_learnig_model.calculate_rnn_results()
return render(request, "users/rnn_results.html", {
"train_score": train_score, "train_rmse": train_rmse, "test_score": test_score,
"test_rmse": test_rmse
})
def user_forecast(request):
from .utility.ForeCastModel import StockPriceForeCast
obj = StockPriceForeCast()
rslt = obj.start_future_prediction()
rslt = rslt.to_html
return render(request,"users/forecast_result.html", {"rslt": rslt}
```

PROCESS ML.PY

```
import pandas as pd
from django.conf import settings
import os
from sklearn.model_selection import train_test_split
path =os.path.join(settings.MEDIA_ROOT, 'ada.csv')
df = pd.read_csv(path)
df = df[['Open', 'High', 'Low', 'Close']]
X = df.iloc[:,:-1].values
y = df.iloc[:,-1].values
print(f'X Len{len(X)} y len {len(y)}')
X_train,X_test,y_train,y_test = train_test_split(X,y, train_size=0.80,random_state=0)

def calc_linear_regression():
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y_pred,y_test)
from sklearn.metrics import mean_squared_error
mse= mean_squared_error(y_pred,y_test)
from sklearn.metrics import r2_score
r2 = r2_score(y_pred,y_test)
return mae,mse,r2

def calc_random_forest_regressor():
# from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
```

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y_pred,y_test)
from sklearn.metrics import mean_squared_error
mse= mean_squared_error(y_pred,y_test)
from sklearn.metrics import r2_score
r2 = r2_score(y_pred,y_test)
return mae,mse,r2

def calc_svm():
# from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
model = SVR()
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y_pred,y_test)
from sklearn.metrics import mean_squared_error
mse= mean_squared_error(y_pred,y_test)
from sklearn.metrics import r2_score
r2 = r2_score(y_pred,y_test)
return mae,mse,r2
```

7.TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.1 TYPES OF TESTS

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual

software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

a. **Valid Input:** identified classes of valid input must be accepted.
b. **Invalid Input:** identified classes of invalid input must be rejected.
c. **Functions:** identified functions must be exercised.
d. **Output:** identified classes of application outputs must be exercised.
e. **Systems/Procedures :** interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.1.7 Validation

Validation means observing the behavior of the system. The verification and validation mean, that will ensure that the output of a phase is consistent with its input and that the output of the phase is consistent with the overall requirements of the system. This is done to ensure that it is consistent with the required output. If not, apply certain mechanisms for repairing and there by achieved the requirements.

7.1.8 Test Cases

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- **a.** All field entries must work properly.
- **b.** Pages must be activated from the identified link.
- **c.** The entry screen, messages and responses must not be delayed.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test Case ID	Module	Test Scenario	Input Data	Expected Output	Actual Output	Status
TC001	Data Upload	Upload a valid stock CSV file	CSV with Date, Open, High, Low, Close, Volume	File accepted, data preview shown	As expected	Pass
TC002	Data Upload	Upload an invalid file format	PDF or incorrect CSV format	Error message: "Invalid file format"	As expected	Pass

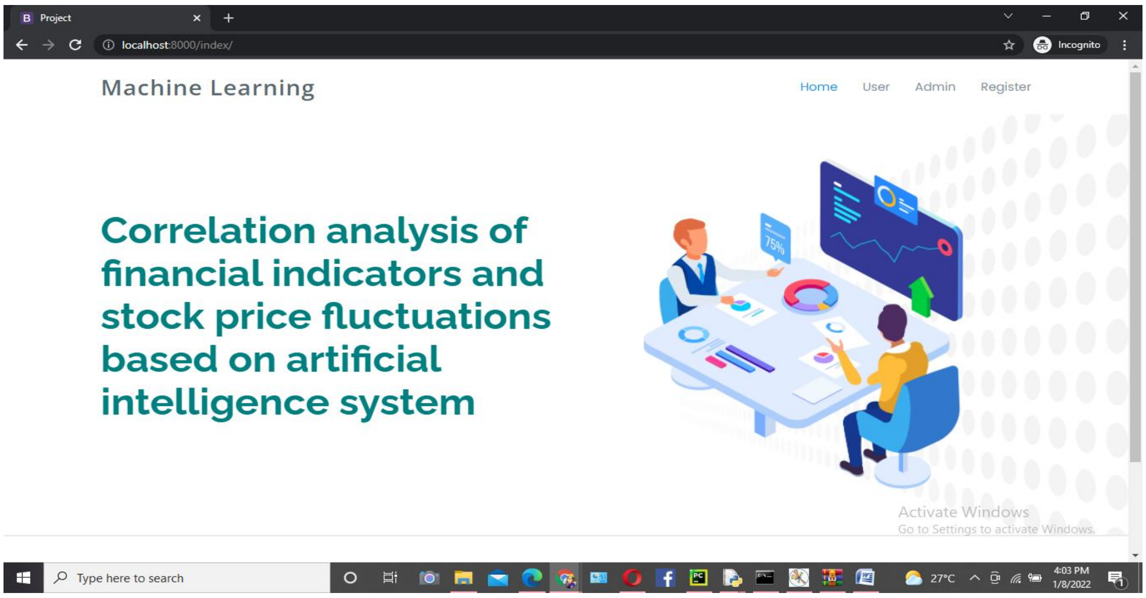
TC003	Data Preprocessing	Handle missing values	CSV with some missing values	Missing values handled (dropped/filled)	As expected	Pass
TC004	Feature Selection	Select key features	Clean dataset	Selected features displayed (e.g., Close, Volume)	As expected	Pass
TC005	Model Training	Train model using Random Forest	Training dataset	Model trained successfully	As expected	Pass
TC006	Model Training	Train model with insufficient data	< 20 rows of data	Warning or error message	As expected	Pass
TC007	Prediction	Generate stock price predictions	New test dataset	Predicted values displayed	As expected	Pass
TC008	Performance Evaluation	Evaluate model accuracy	Predicted vs actual values	MAE, RMSE, R^2 score shown	As expected	Pass
TC009	Visualization	Display predictions on graph	Model output	Line chart showing actual vs predicted values	As expected	Pass
TC010	User Interface (optional)	Complete flow from upload to prediction	Uploaded file through UI	All steps run smoothly and results are shown	As expected	Pass

Table 3: Test case Table

8.OUTPUT SCREENS

SCREEN SHOTS

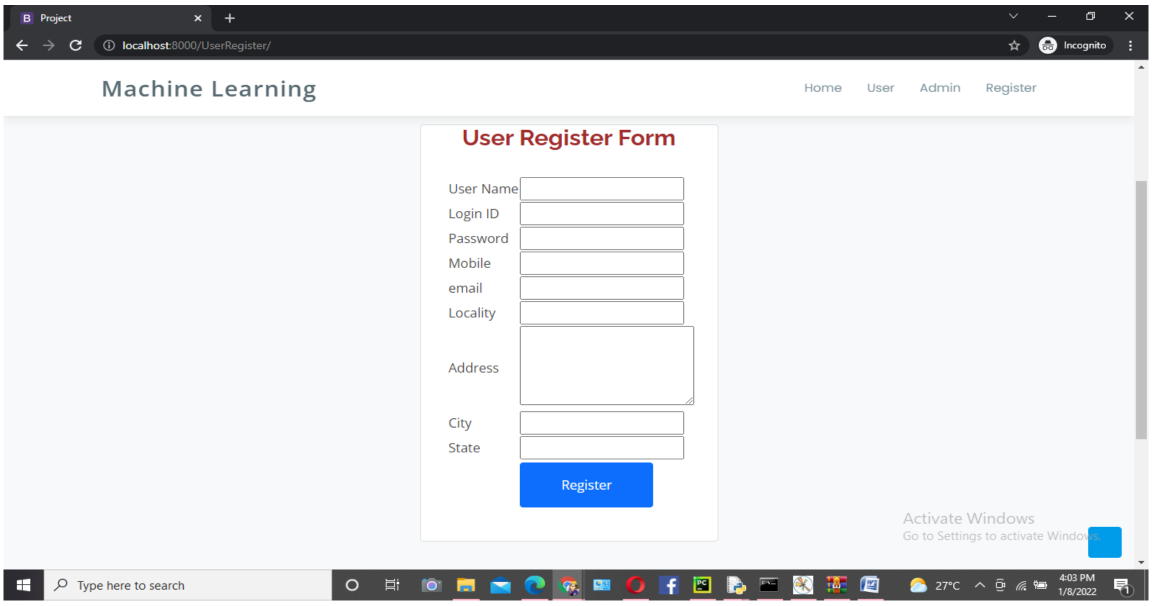
8.1 Home page



Correlation analysis of financial indicators and stock price fluctuations based on artificial intelligence system

Fig 8.1 Home Page

8.2 User Register Form



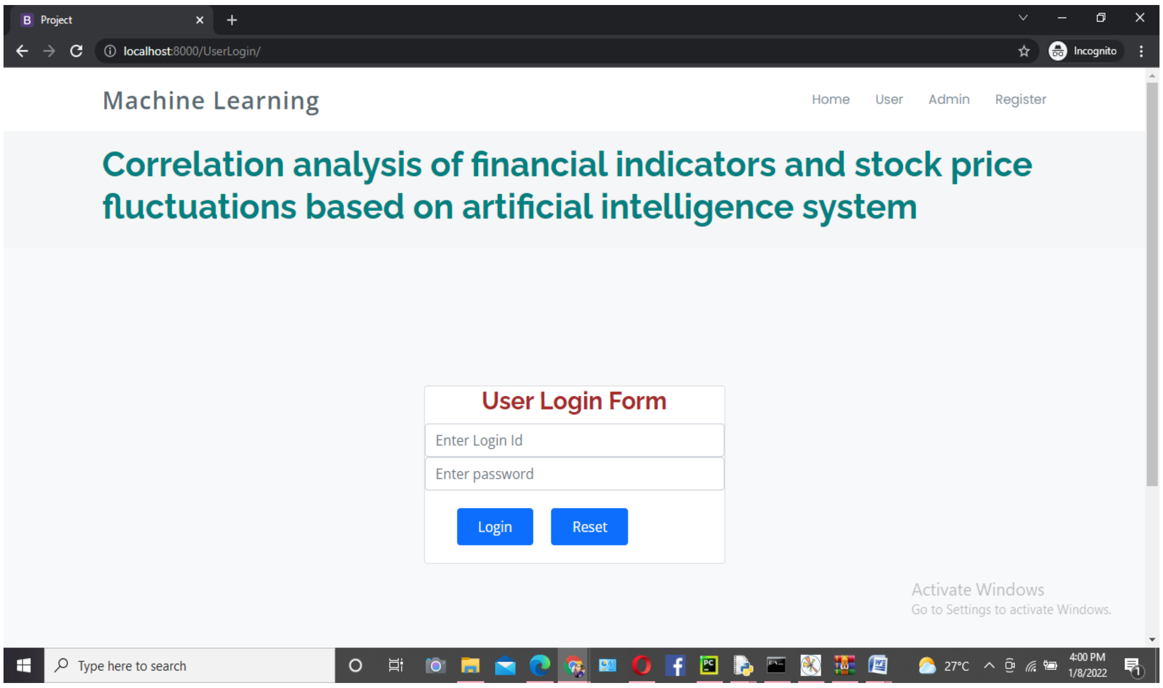
User Register Form

User Name	
Login ID	
Password	
Mobile	
email	
Locality	
Address	
City	
State	

Register

Fig 8.2 User registration From

8.3 User Login Form



Correlation analysis of financial indicators and stock price fluctuations based on artificial intelligence system

User Login Form

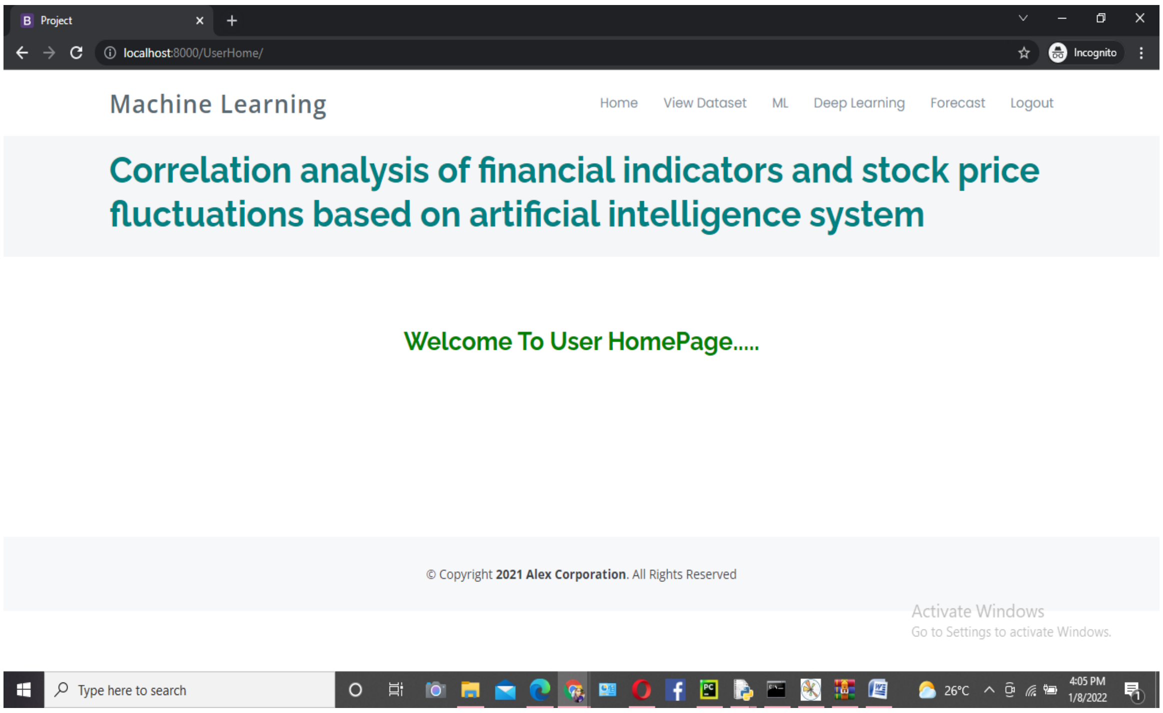
Enter Login Id

Enter password

Fig 8.3 User Login From

This is the login frame of the system where admin have to enter the required credentials to have access for the main dashboard.

8.4 User Home Page



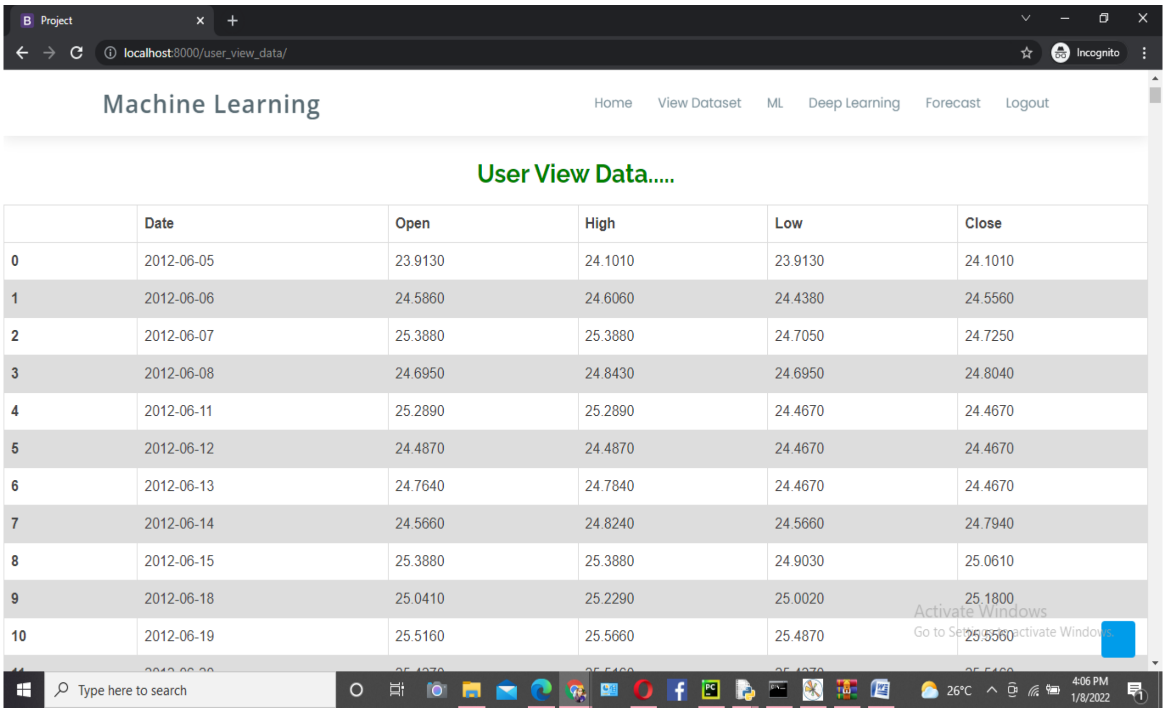
Correlation analysis of financial indicators and stock price fluctuations based on artificial intelligence system

Welcome To User HomePage.....

© Copyright **2021 Alex Corporation**. All Rights Reserved

Fig 8.4 User Home Page

8.5 View Dataset

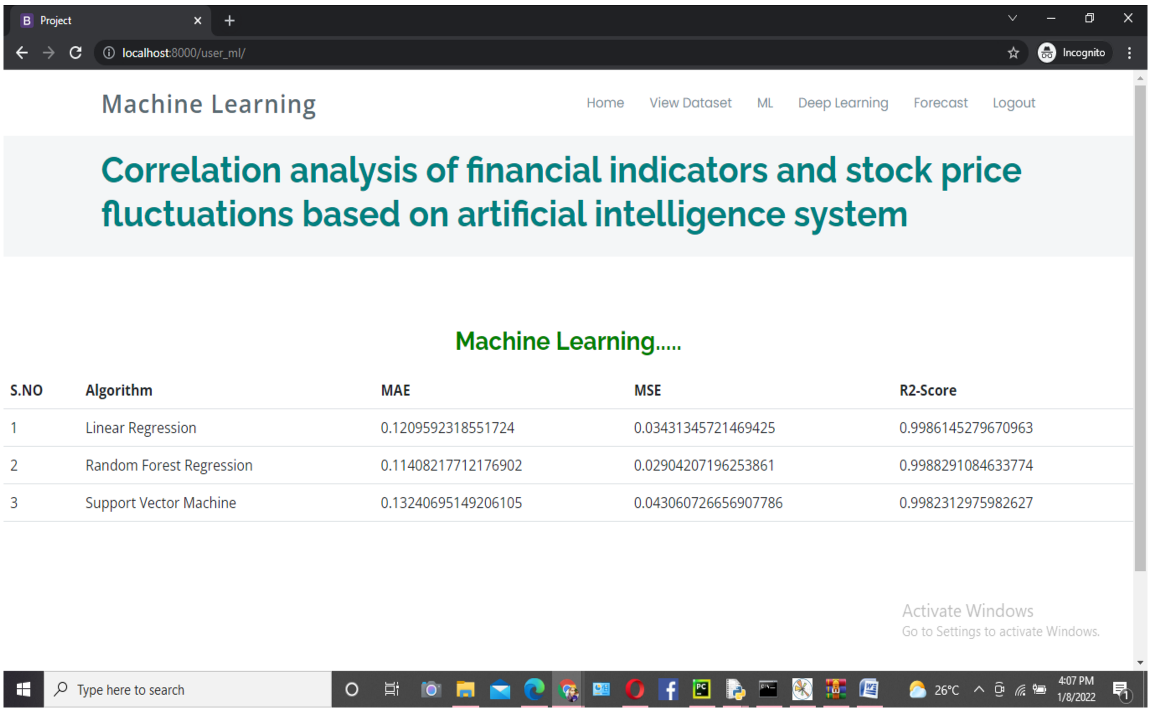


User View Data.....

	Date	Open	High	Low	Close
0	2012-06-05	23.9130	24.1010	23.9130	24.1010
1	2012-06-06	24.5860	24.6060	24.4380	24.5560
2	2012-06-07	25.3880	25.3880	24.7050	24.7250
3	2012-06-08	24.6950	24.8430	24.6950	24.8040
4	2012-06-11	25.2890	25.2890	24.4670	24.4670
5	2012-06-12	24.4870	24.4870	24.4670	24.4670
6	2012-06-13	24.7640	24.7840	24.4670	24.4670
7	2012-06-14	24.5660	24.8240	24.5660	24.7940
8	2012-06-15	25.3880	25.3880	24.9030	25.0610
9	2012-06-18	25.0410	25.2290	25.0020	25.1800
10	2012-06-19	25.5160	25.5660	25.4870	25.5560

Fig 8.5 View Dataset

8.6 Machine Learning Algorithms



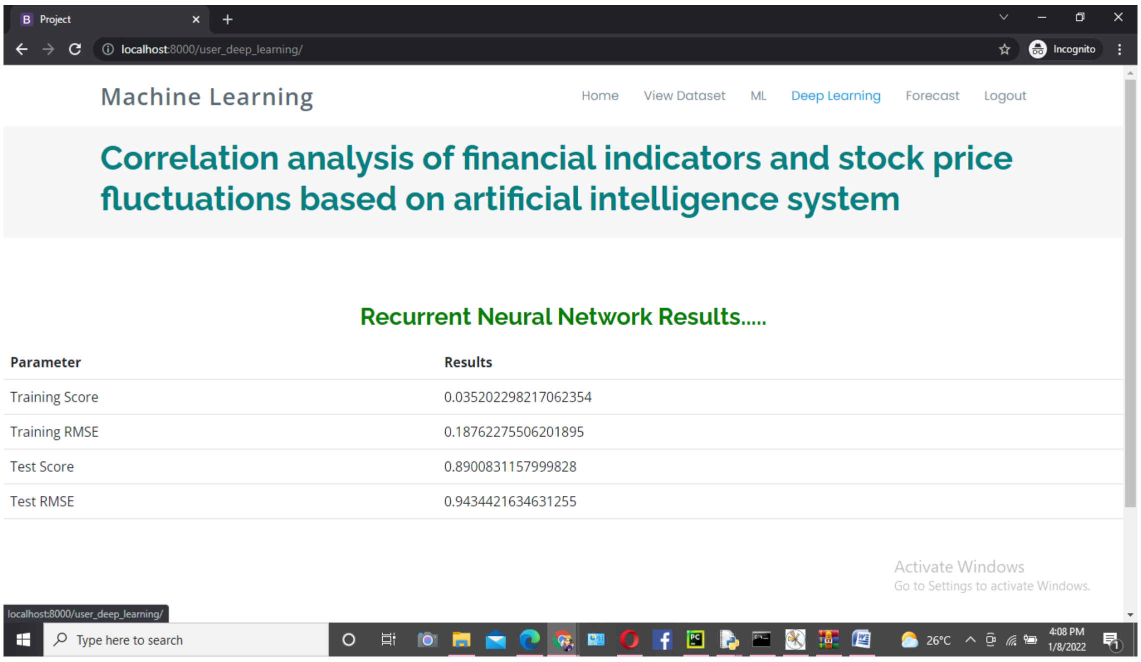
Correlation analysis of financial indicators and stock price fluctuations based on artificial intelligence system

Machine Learning.....

S.NO	Algorithm	MAE	MSE	R2-Score
1	Linear Regression	0.1209592318551724	0.03431345721469425	0.9986145279670963
2	Random Forest Regression	0.11408217712176902	0.02904207196253861	0.9988291084633774
3	Support Vector Machine	0.13240695149206105	0.043060726656907786	0.9982312975982627

Fig 8.6 Machine Learning

8.7 Deep Learning



Correlation analysis of financial indicators and stock price fluctuations based on artificial intelligence system

Recurrent Neural Network Results.....

Parameter	Results
Training Score	0.035202298217062354
Training RMSE	0.18762275506201895
Test Score	0.8900831157999828
Test RMSE	0.9434421634631255

Fig 8.7 Deep learning

8.8 Forecast

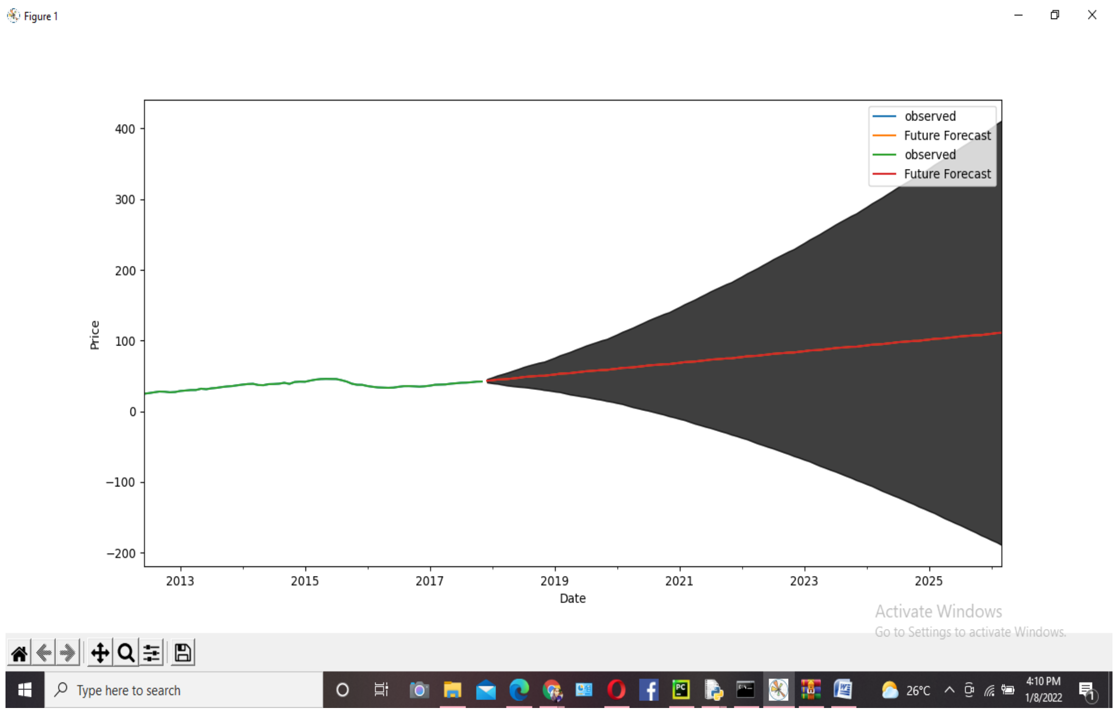
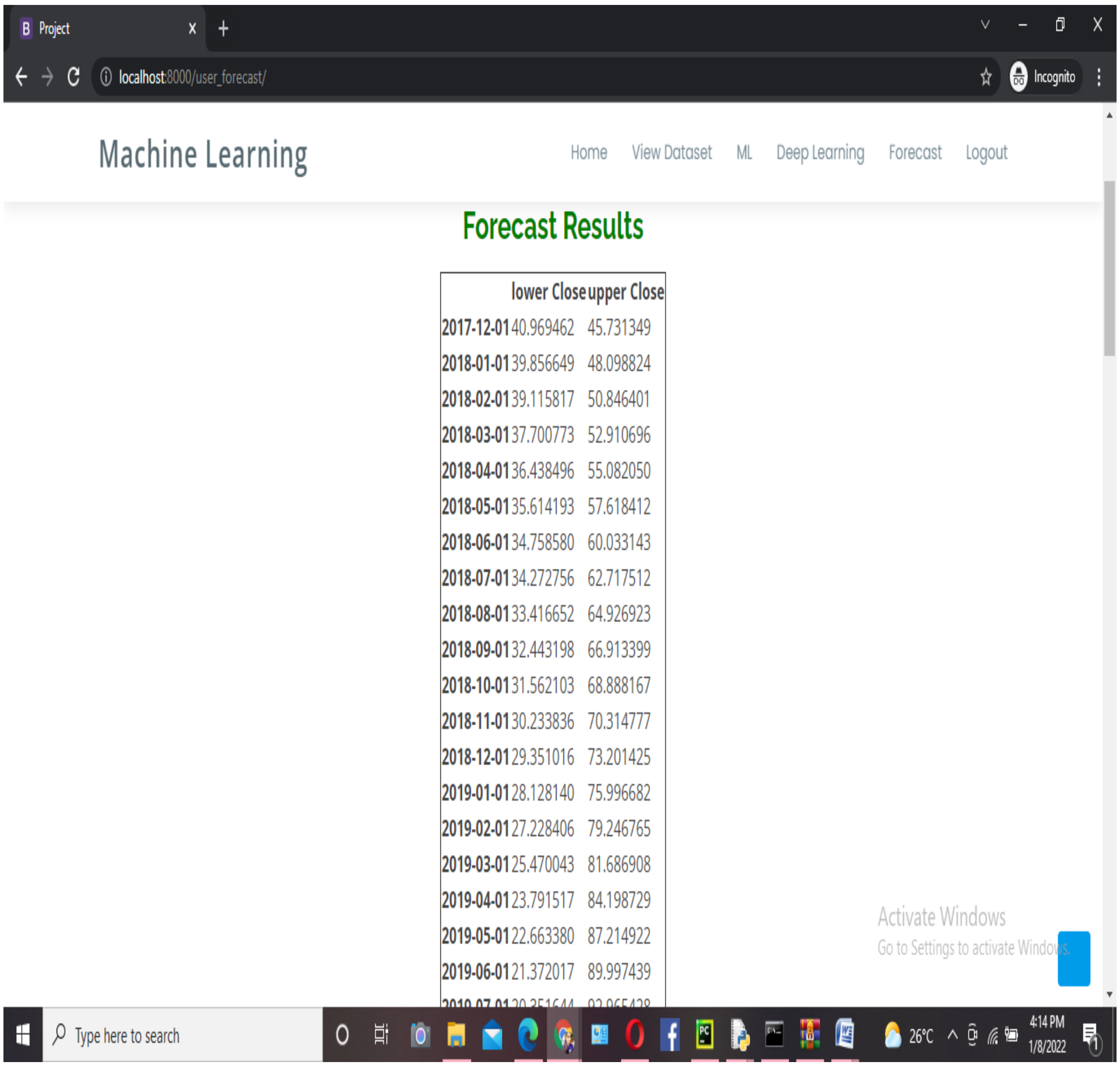


Fig 8.8: Forecast

8.9 Forecast Results



Forecast Results

	lower Close	upper Close
2017-12-01	40.969462	45.731349
2018-01-01	39.856649	48.098824
2018-02-01	39.115817	50.846401
2018-03-01	37.700773	52.910696
2018-04-01	36.438496	55.082050
2018-05-01	35.614193	57.618412
2018-06-01	34.758580	60.033143
2018-07-01	34.272756	62.717512
2018-08-01	33.416652	64.926923
2018-09-01	32.443198	66.913399
2018-10-01	31.562103	68.888167
2018-11-01	30.233836	70.314777
2018-12-01	29.351016	73.201425
2019-01-01	28.128140	75.996682
2019-02-01	27.228406	79.246765
2019-03-01	25.470043	81.686908
2019-04-01	23.791517	84.198729
2019-05-01	22.663380	87.214922
2019-06-01	21.372017	89.997439

Fig 8.9 Forecast Results

9. CONCLUSION

8.

The project on Stock Price Fluctuations Using Machine Learning has demonstrated the significant potential of machine learning in predicting stock market behavior. By utilizing various ML models and financial indicators, this project successfully highlighted the complex relationships between market factors and stock price movements. The results indicate that leveraging historical data, technical indicators, and advanced machine learning algorithms can provide meaningful insights into predicting stock price fluctuations. Furthermore, the integration of sentiment analysis and real-time data streaming holds promise for enhancing the accuracy and timeliness of predictions. While the project has produced valuable results, there are several avenues for further research and enhancement. Integrating volatility modeling, real-time prediction systems, and the exploration of multi-market analysis will provide more comprehensive and actionable insights for investors. Additionally, incorporating explainability features will ensure transparency and trust, crucial for deploying machine learning models in financial decision-making contexts. Ultimately, this project underscores the importance of machine learning in the financial sector, offering a foundation for more sophisticated predictive models and strategies. The future enhancements outlined provide a clear roadmap for advancing the predictive capabilities of the system, ensuring it stays relevant and effective in a rapidly evolving market.

10. FUTURE ENHANCEMENTS

Future enhancements in stock price prediction can include the adoption of advanced machine learning models like XGBoost, LightGBM, and CatBoost for improved accuracy. Deep learning methods such as LSTM, GRU, and Transformers may be used to better capture time series trends. Enhanced feature engineering using technical indicators and dimensionality reduction techniques like PCA or autoencoders can refine input data. Real-time prediction systems using live financial data APIs can offer continuous forecasts. Sentiment analysis from news and social media using NLP can improve market behavior understanding. Forecasting volatility and risk with models like GARCH adds depth to predictions. Explainability tools like SHAP and LIME can ensure model transparency. Back testing tools can assess trading strategies, and portfolio optimization can guide investment decisions. Finally, expanding to multi-asset and cross-market analysis can uncover broader market insights.

REFERENCES

[1] Nti, IK. Adekoya, AF., and Weyori, BA. (2020) A systematic review of fundamental and technical analysis of stock market predictions. Artificial Intelligence Review, 53: 3007-3057.

[2] Obthong, M. Tantisantiwong, N. Jeamwatthanachai, W., and Wills, G. (2020) A survey on machine learning for stock price prediction: algorithms and technique. 2nd International Conference on Finance, Economics, Management and IT Business, 63-71.

[3] Ho, M., Darman, H., & Musa, S. (2021). Stock Price Prediction Using ARIMA, Neural Network and LSTM Models. Journal of Physics. Conference Series, 1988(1), 12041.

[4] Mehtab S, Sen J and Dutta A 2020 Stock Price Prediction Using Machine Learning and LSTM Based Deep Learning Models Second Symposium on Machine Learning and Metaheuristics Algorithms.

[5] Sen J 2018 Stock Price Prediction Using Machine Learning and Deep Learning Frameworks International Conference on Business Analytics and Intelligence.

[6] Gülmez B (2023) Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. Expert Syst Appl 227:120346.

[7] Pahwa N, Khalfay N, Soni V, Vora D (2017) Stock prediction using machine learning a review paper. Int J Computer Appl 163(5):36–43

[8] Ghosh, Achyut & Bose, Soumik & Maji, Giridhar & Debnath, Narayan & Sen, Soumya. (2019). Stock Price Prediction Using LSTM on Indian Share Market. 10.29007/qgcz.

[9] Polamuri, Subba Rao, Kudipudi Srinivas, and A. Krishna Mohan. 2019. Stock Market Prices Prediction using Random Forest and Extra Tree Regression. *International Journal of Recent Technology and Engineering* 8: 1224–28.

[10] Mukherjee, Somenath, Bikash Sadhukhan, Nairita Sarkar, Debajyoti Roy, and Soumil De. 2021. Stock market prediction using deep learning algorithms. *CAAI Transactions on Intelligence Technology* 8: 82–94.