A

Project Report

On

Stock Prediction Web App

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Divyarth Sah	**2261639**
Priyanshu Bhatt	**2261441**
Paras Punetha	**2261411**
Lalit Tiwari	**2261333**

Under the Guidance of

Mr. Prince Kumar

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI DISTRICT- NAINITAL-263132

2024-2025

STUDENT'S DECLARATION

We, **Divyarth Sah, Priyanshu Bhatt, Paras Punetha**, **Lalit Tiwari** hereby declare the work, which is being presented in the project, entitled '**Stock Prediction Web App**' in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (B.Tech.)** in the session **2024-2025**, is an authentic record of my work carried out under the supervision of **Mr. Prince Kumar, Assistant Professor**.

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date:

Divyarth Sah

Priyanshu Bhatt

Paras Punetha

Lalit Tiwari

Graphic Era Hill University
BHIMTAL CAMPUS

CERTIFICATE

The term work of Project Based Learning, being submitted by Divyarth Sah (2261639), Priyanshu Bhatt (2261441), Paras Punetha (2261411) and Lalit Tiwari (2261333) to Graphic Era Hill University Bhimtal Campus for the award of Bonafide work carried out by us. They had worked under my guidance and supervision and fulfilled the requirements for the submission of this work report.

(Mr. Prince Kumar) **(Dr. Ankur Singh Bist)**

Faculty-in-Charge **HOD, CSE Dept.**

ACKNOWLEDGEMENT

We take immense pleasure in thanking the Honorable Director **'Prof. (Col.) Anil Nair (Retd.)',** GEHU Bhimtal Campus to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance, and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me with everything that we need. We again want to extend thanks to our president **'Prof. (Dr.) Kamal Ghanshala'** for providing us with all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to **'Dr. Ankur Singh Bisht'** (Head, Department of Computer Science and Engineering, GEHU Bhimtal Campus), our project guide **'Mr. Prince Kumar'** (Assistant Professor, Department of Computer Science and Engineering, GEHU Bhimtal Campus) and other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this report.

Finally, yet importantly, We would like to express my heartiest thanks to our beloved parents, for their moral support, affection, and blessings. We would also like to pay our sincere thanks to all my friends and well-wishers for their help and wishes for the successful completion of this project.

Divyarth Sah , 2261182

Priyanshu Bhatt, 2261441

Paras Punetha, 2261411

Lalit Tiwari, 2261333

Abstract

In recent years, financial markets have witnessed a significant increase in the application of machine learning and artificial intelligence techniques to improve stock market forecasting and investment strategies. This project, titled " Stock Prediction Web App ", presents a robust web-based application that enables users to predict future stock prices through a combination of traditional and deep learning models.

The application integrates four major prediction algorithms: Linear Regression, Random Forest, XGBoost, and Long Short-Term Memory (LSTM) networks. These models are selected to cover a wide range of forecasting approaches — from simple linear trends to complex non-linear temporal patterns. The user interface is developed using Python Flask, an open-source Python framework that provides interactive visualization and model comparison tools in a web environment.

Users can input a stock ticker, select a prediction model, and view visual plots for historical and predicted prices. The application uses Yahoo Finance as a data source and implements preprocessing techniques such as normalization, feature engineering, and time series splitting to enhance model performance. The LSTM model is used for capturing long-term dependencies in stock price sequences, making it particularly useful for modeling complex financial patterns.

The app emphasizes user-friendly design, real-time interactivity, and multi-model comparison, catering to both novice and advanced users. The primary motivation behind this project is to democratize access to predictive financial tools and promote data-driven investment decisions.

This project can be expanded further by integrating more advanced techniques such as sentiment analysis from financial news, technical indicators, and deploying it as a full-scale enterprise solution. The proposed system demonstrates the practical application of machine learning in financial domains and serves as a learning tool for developers and investors alike.

TABLE OF CONTENTS

Chapter1: INTRODUCTION

1.1 Prologue

The stock market, known for its dynamic and volatile nature, has always been a subject of interest for investors, economists, and researchers alike. With the rise of data-driven technologies and the proliferation of accessible financial data, the domain of stock price prediction has evolved significantly. Traditional forecasting methods have given way to more intelligent systems powered by machine learning and artificial intelligence, which have shown promising results in uncovering hidden patterns in historical stock data.

This project, titled " Stock Prediction Web App ", embodies the intersection of finance and data science. It has been developed to provide users—whether investors, researchers, or students—with a convenient, web-based platform for predicting future stock prices using state-of-the-art machine learning algorithms. The project combines data acquisition, preprocessing, model training, and result visualization in a streamlined workflow.

The application not only serves as a practical implementation of theoretical machine learning concepts but also reflects a real-world use case with tangible benefits. It highlights the value of integrating multiple models like Linear Regression, Random Forest, XGBoost, and LSTM in a comparative interface to empower users to analyze and interpret market movements effectively.

Through this endeavor, the project aims to contribute to the ongoing innovation in financial technology by offering an accessible tool that bridges the gap between machine learning capabilities and practical investment decision-making.

1.2 Background and Motivations

The financial sector is one of the most data-intensive domains in the world. With trillions of transactions and rapidly changing market dynamics, accurate stock price forecasting has become increasingly important for individual investors, analysts, and institutions. Traditionally, stock prediction relied on statistical models and expert judgment, which often failed to capture complex patterns and trends within vast and volatile datasets.

With the emergence of machine learning (ML) and deep learning (DL) technologies, there is now a powerful alternative to traditional forecasting methods. These technologies offer the ability to learn from historical data, identify hidden patterns, and make predictions with greater accuracy and adaptability. Especially in the field of time series forecasting, machine learning models such as Random Forest, XGBoost, and Long Short-Term Memory (LSTM) networks have demonstrated considerable success.

1.3 Problem Statement

The stock market is inherently volatile and influenced by numerous unpredictable factors such as economic indicators, political events, investor sentiment, and global crises. Traditional methods of stock price prediction, such as statistical regression or technical analysis, often fall short in modeling the complex, nonlinear relationships present in historical stock data. Moreover, many forecasting tools are either too simplistic for meaningful insights or too complex for non-expert users to operate efficiently.

The absence of a unified, user-friendly application that leverages multiple machine learning algorithms for stock prediction presents a significant gap in the current financial technology landscape. Existing platforms may offer some degree of predictive capability but lack transparency in model comparison, accessibility, or interactive visualization. Furthermore, deep learning techniques such as Long Short-Term Memory (LSTM), which are known to capture temporal dependencies in time-series data, are rarely integrated into lightweight forecasting tools usable by non-programmers.

This project addresses the problem by developing a Stock Prediction App using Machine Learning, which provides a comparative and visual environment to forecast stock trends using Linear Regression, Random Forest, XGBoost, and LSTM. The application simplifies the process of obtaining predictions, making it suitable for both technical and non-technical users. By doing so, it bridges the gap between advanced predictive analytics and real-world usability.

1.4 Objectives and Research Methodology

Objectives

The primary objective of this project is to design and develop a robust, interactive web application that predicts stock prices using various machine learning algorithms. The specific goals of the project are as follows:

- To collect and preprocess historical stock data from reliable sources such as Yahoo Finance.
- To implement multiple machine learning models including Linear Regression, Random Forest, XGBoost, and LSTM for forecasting stock prices.
- To provide an interactive, easy-to-use web interface using Streamlit that allows users to select models, input stock tickers, and visualize predictions.
- To compare the performance of different models to help users understand their accuracy and predictive capabilities.
- To facilitate decision-making by providing informative plots, metrics, and trend visualizations for better financial insights.
- To lay the foundation for future enhancements, including real-time data feeds, sentiment analysis, and technical indicator integration.

Research Methodology

The methodology adopted for this project includes the following stages:

1. Literature Review:
   A comprehensive study of stock market behavior, machine learning models, and existing forecasting applications was conducted to identify gaps and opportunities.
2. Data Collection:
   Historical stock price data was sourced using the yfinance API. Data includes daily open, close, high, low, and volume metrics.
3. Data Preprocessing:
   Data cleaning, normalization, feature selection, and formatting were performed to prepare datasets suitable for training different machine learning models.

4. Model Implementation:

   - Linear Regression: Applied as a baseline predictive model.

   - Random Forest & XGBoost: Implemented for non-linear regression and ensemble learning.

   - LSTM: Designed using Keras/TensorFlow for modeling sequential dependencies in time-series data.

5. Model Evaluation:
   Each model was evaluated using standard metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). Predictions were visualized using Matplotlib and Seaborn.

6. Application Development:
   A responsive web-based interface was created using Streamlit, allowing real-time interaction, model selection, and dynamic plotting.

7. Testing and Validation:
   The application was tested with different stocks, time ranges, and scenarios to ensure reliability and usability.

1.5 Project Organization

The project is divided into the following modules:

Module	Description
backend/server.py	Starts the web server and manages connections.
backend/handler.py	Handles routing of requests and serving static files.
backend/auth.py	Manages user login and session logic.
data/user.txt	Stores username-password pairs.
data/data. Json	Contains student profile data.
static/index.html	Login page.
static/dashboard.html	Dashboard with user info.
static/profile.html	Profile view/edit page.
static/CSS/ and Js/	Contains all frontend styling and interactivity scripts.

CHAPTER 2 PHASES OF SOFTWARE DEVELOPMENT CYCLE

Hardware and Software Requirements

2.1 Hardware Requirement

Component	Minimum Requirements	Recommended Requirements
CPU	2 cores	4+ cores
RAM	4 GB	8+ GB
Disk Space	50 GB	100+ GB
Network Adapter	1 Gbps	10 Gbps

2.2 Software Requirement

Software	Minimum Requirements	Recommended Requirements
Operating System	Linux(Ubuntu,CentOS), Windows Server	Linux(Ubuntu,CentOS), Windows Server
Web Server Software	Chrome	Chrome
Programming Language	Html, CSS ,Python	Html, CSS, Python
Database	MySQL, PostgreSQL	MySQL, PostgreSQL
Development Tools	GCC, JDK, Python IDE	GCC, JDK, Python IDE

CHAPTER 3: CODING OF FUNCTIONS

This chapter outlines the functional implementation of the stock prediction system. Developed using Python, the system leverages various machine learning and data processing libraries to predict stock prices based on historical data. The application is modular, consisting of well-defined components for data handling, model training, and result visualization.

1. Data Collection and Preprocessing

The system begins by collecting historical stock data from online financial sources. A specific stock ticker (such as "AAPL" for Apple) is input by the user, and the corresponding data is fetched from a reliable API. The data mainly includes the stock's closing prices over a long historical period.

Once collected, the data is cleaned by removing any missing values and is then normalized. Normalization ensures that all values are within a specific range (usually between 0 and 1), which helps improve the accuracy and efficiency of the machine learning model.

Each thread handles the parsing of the HTTP request, routes it to the appropriate handler (login, file upload, profile fetch, etc.), and returns the appropriate response to the client browser. The use of multithreading ensures that long-running operations (such as file uploads or database access) do not block the entire server, preserving responsiveness**.**

2.Data Preparation for LSTM

To prepare the data for the prediction model, it is structured into sequences. This process involves creating input-output pairs from the time-series data, where the model uses a specific number of previous days' prices to predict the next day's price. This sequential formatting is essential for training the LSTM (Long Short-Term Memory) neural network, which is designed to learn from time-dependent data.

The dataset is then split into two parts: training data (used to train the model) and testing data (used to evaluate the model's performance).

3. Model Building

The core predictive model used in this project is an LSTM neural network. LSTM is well-suited for time-series forecasting because of its ability to remember long-term patterns in sequential data.

The model includes layers that are responsible for learning features from the input sequences and producing a single output, which is the predicted closing price. It is compiled with an optimization algorithm and a loss function that measures prediction accuracy during training.

4. Model Training and Evaluation

Once the model is defined, it is trained using the training dataset. This involves passing the data through the model multiple times (epochs), allowing it to learn the patterns and trends in the stock price history.
After training, the model is evaluated using the test data. Its performance is assessed using metrics such as Root Mean Square Error (RMSE), which indicates how closely the predicted values match the actual values.

5. Prediction Interface

The user interacts with the system through a graphical interface built using Python Flask. This interface allows users to input a stock symbol, view the historical data, and see a graph of the predicted future trends.

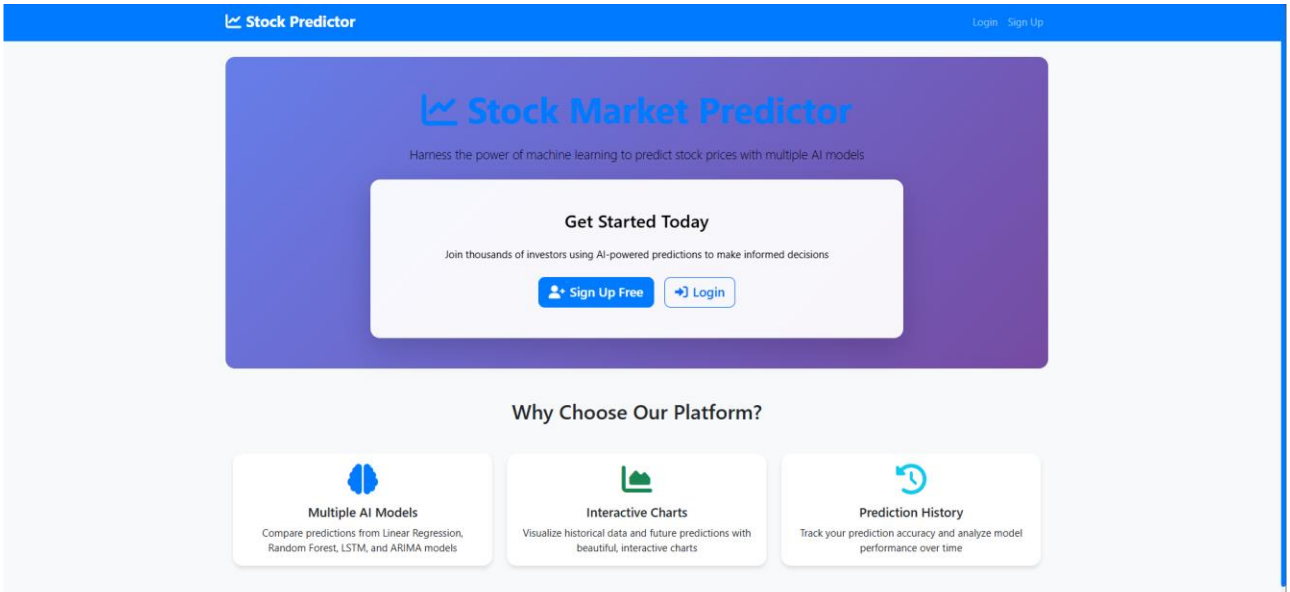
The prediction results are displayed alongside the actual historical stock prices, allowing users to visually compare the model's predictions with real market behavior.

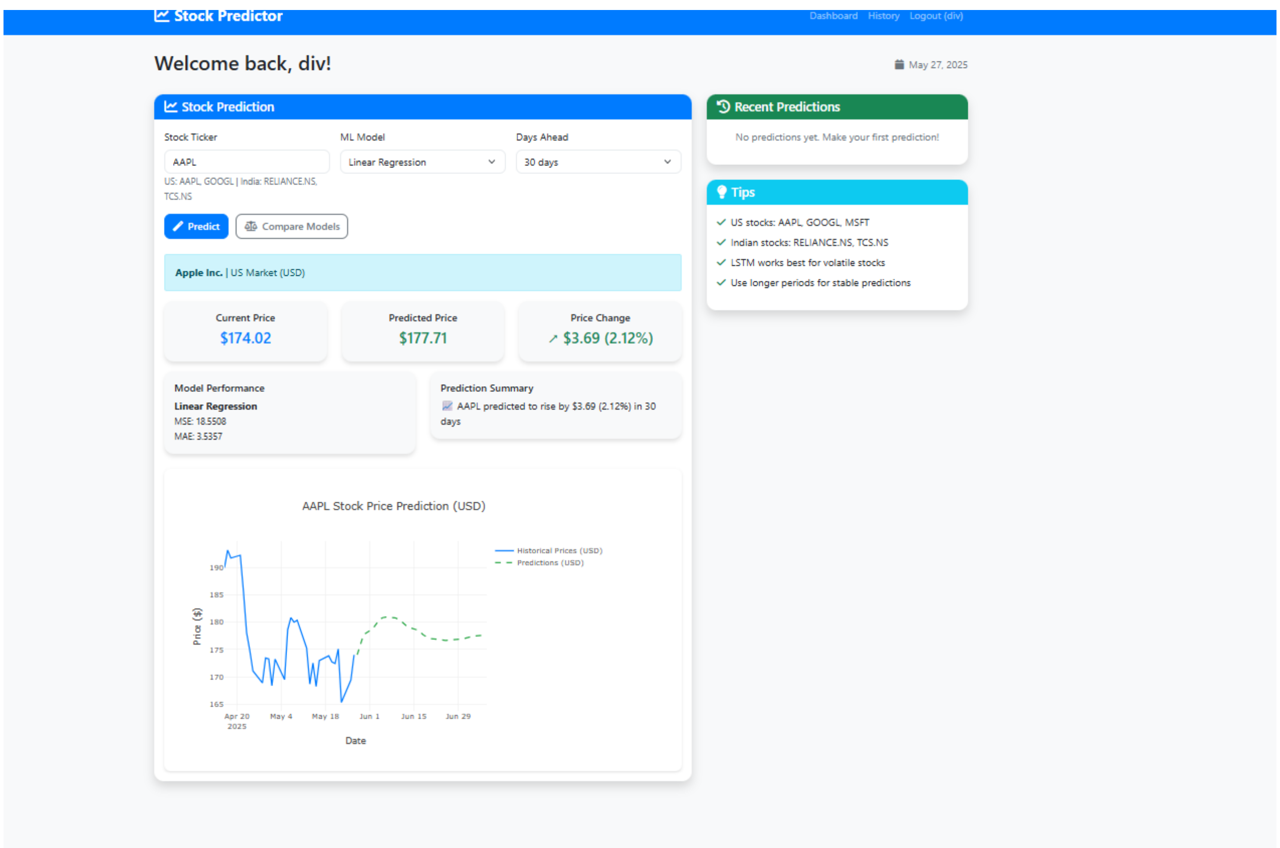
6. Visualization

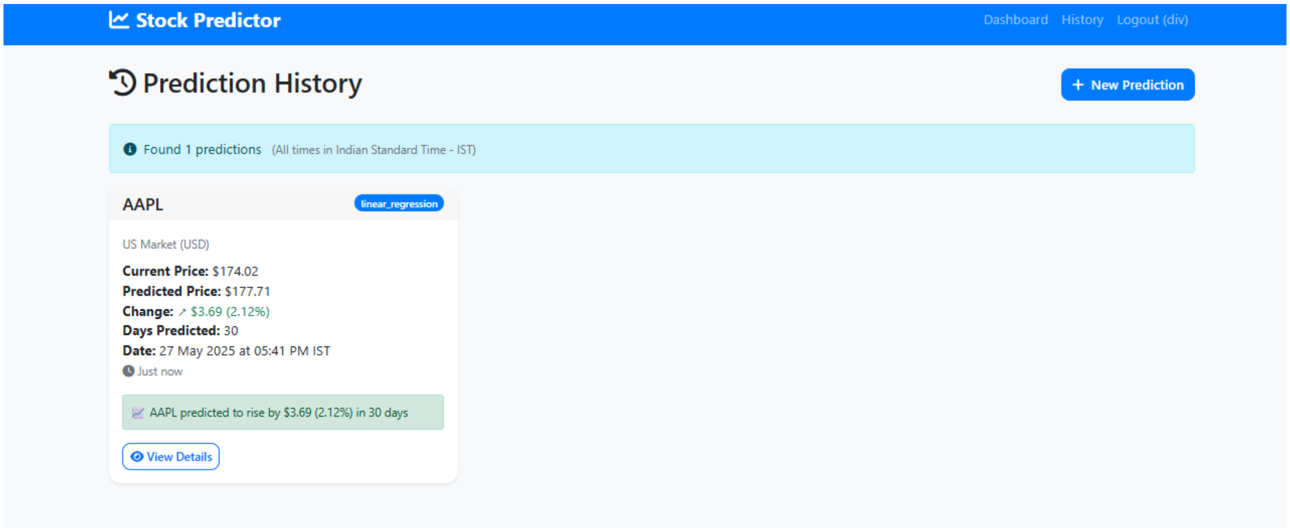
The application includes visual representations of both the raw historical data and the predicted trends. These visualizations help users understand how the stock has performed over time and how it is expected to behave in the near future.

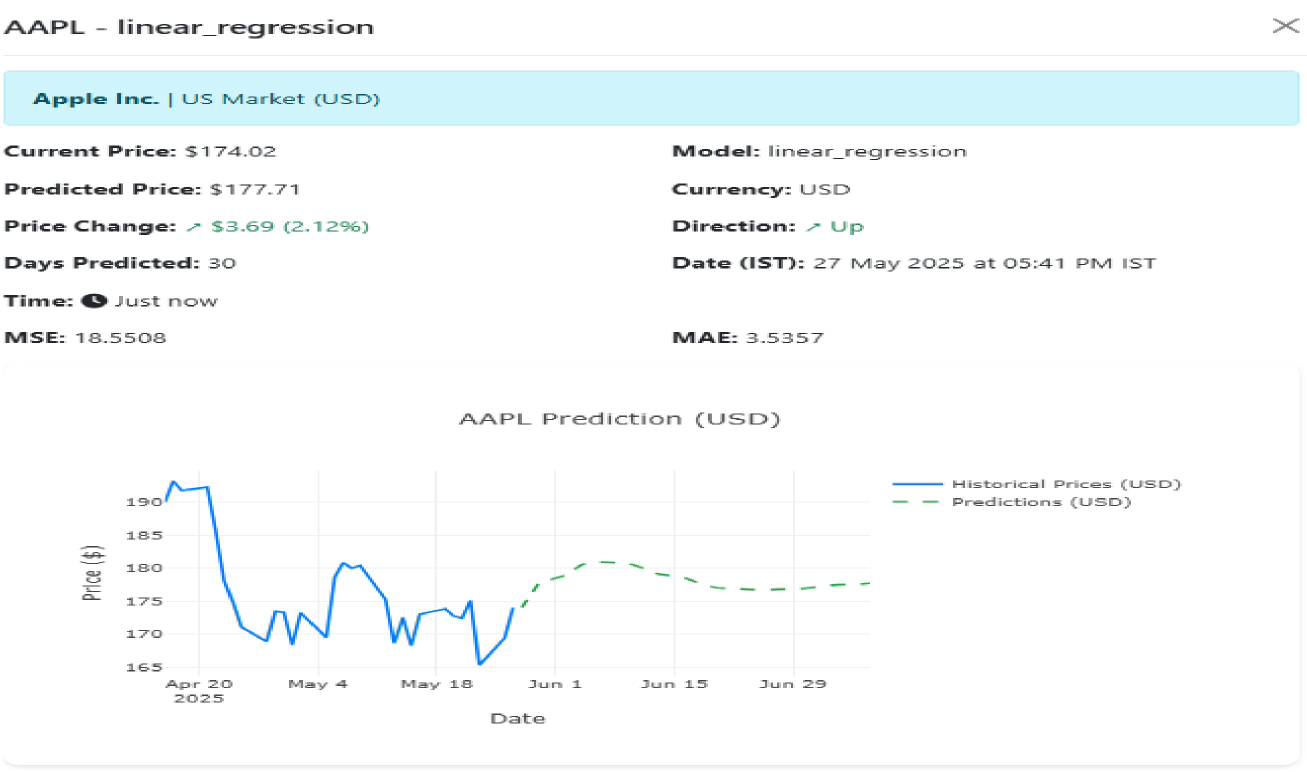
Graphs are used to show the difference between actual and predicted values, making it easier for users to interpret the model's output and judge its reliability.

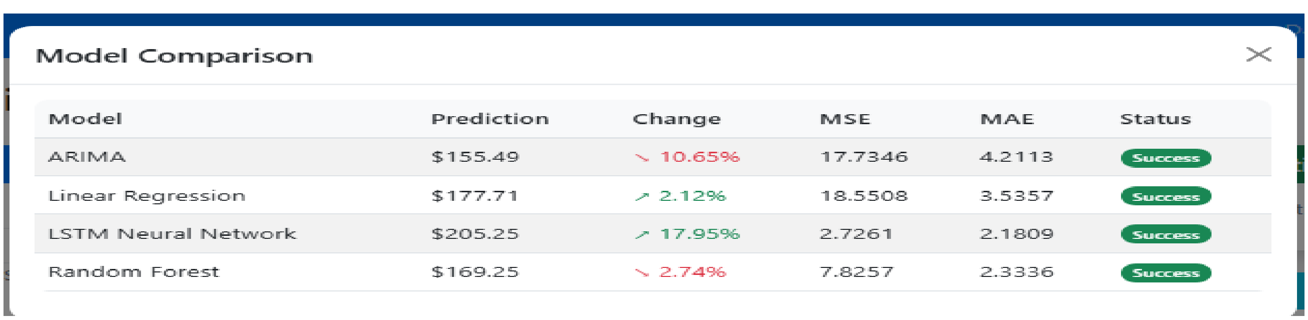
CHAPTER 4: SNAPSHOTS











Model	Prediction	Change	MSE	MAE	Status
ARIMA	$155.49	↘ 10.65%	17.7346	4.2113	Success
Linear Regression	$177.71	↗ 2.12%	18.5508	3.5357	Success
LSTM Neural Network	$205.25	↗ 17.95%	2.7261	2.1809	Success
Random Forest	$169.25	↘ 2.74%	7.8257	2.3336	Success

CHAPTER 5: LIMITATIONS

Despite the effectiveness and functionality of the stock prediction system, several limitations restrict its performance, scalability, and real-world applicability. These are discussed below:

1. **Limited Accuracy of Predictions**: Stock market prices are influenced by a wide range of unpredictable factors such as global events, political instability, company news, and investor sentiment. Since this model only relies on historical price data, it cannot account for such real-time variables, which can significantly affect prediction accuracy.

2. **Data Dependency**: The model's performance heavily depends on the quality and completeness of the historical stock data. Any missing or inaccurate data can lead to flawed predictions. Moreover, the system uses past trends assuming they will repeat, which is not always the case in volatile markets..

3. **Scalability and Performance**: The model is built using LSTM and processes data sequentially, which may not be optimal for handling high-frequency trading data or large-scale multi-stock predictions. The system can also become computationally expensive with long sequences or large datasets.

4. **Absence of External Features**: Currently, the system does not incorporate technical indicators (like MACD, RSI) or external financial data such as earnings reports, macroeconomic indicators, or news sentiment. Including such features would likely improve prediction quality and make the model more robust.

5. **No Real-time Prediction**: The system does not support real-time streaming of data or live market predictions. Users can only run predictions based on the most recently available historical data. This limits its use in active trading scenarios.

6. **Simplified User Interface**: While the application provides a usable interface, it lacks advanced UI features such as stock filtering, historical comparison, multi-stock visualization, or mobile responsiveness. This restricts its appeal to end-users who expect a professional-grade application experience.

7. **No Financial Risk Modeling:** The application does not assess or communicate the financial risks involved in acting upon the predicted results. It also does not provide any investment advice or probabilistic confidence levels, which are critical for informed decision-making.

.

CHAPTER 6: ENHANCEMENTS

To address the limitations discussed in the previous chapter and to improve the system's effectiveness, scalability, and user experience, several enhancements can be proposed for future development:

1. **Integration of Real-Time Data**: One of the most important upgrades would be incorporating real-time data feeds from multiple financial APIs. This would enable the system to perform live market analysis and make it suitable for real-time stock monitoring or intraday trading.

2. **Use of Technical and Fundamental Indicators:** Enhancing the model by including technical indicators (e.g., Moving Averages, RSI, MACD) and fundamental data (e.g., P/E ratio, earnings reports) would provide more context to the predictions. These additional features can help improve the accuracy and interpretability of the model.

3. **Advanced Machine Learning Models**: While LSTM performs well on sequential data, exploring more advanced or hybrid models like GRU (Gated Recurrent Units), Transformer architectures, or ensemble learning could further improve prediction performance. AutoML tools could also be used to optimize model selection and tuning.

4. **Transition to Scalable Infrastructure**: Migrating the model and application to a cloud-based platform (e.g., AWS, GCP, or Azure) would support better scalability, continuous deployment, and real-time usage. This would also allow for parallel processing and faster predictions for multiple stocks..

5. **Improved User Interface**: Redesigning the frontend with modern web frameworks like React.js or Vue.js could enhance responsiveness, interactivity, and visual appeal. Features like search filters, comparison tools, and mobile responsiveness would significantly improve usability.

6. **Historical and Portfolio Analysis**: Adding features for portfolio tracking and historical performance comparison would provide users with deeper insights. Users could input their portfolio and analyze how well their investments might perform based on the model's predictions.

7. **Risk Assessment Module**: To make the portal more inclusive, we have to Introduce a risk modeling component which would help users understand the volatility or uncertainty associated with predictions. Visualization of confidence intervals and recommendations based on risk appetite could guide better decision-making.

CHAPTER 7: CONCLUSION

The development of the stock prediction system marks a significant step toward applying machine learning techniques in financial forecasting. This project demonstrated how historical stock data can be leveraged using deep learning models—specifically LSTM networks—to make informed predictions about future stock price movements.

The core objective of building a functional, user-interactive, and data-driven application was successfully achieved. Through the integration of Python-based libraries and tools, the system was able to handle data preprocessing, model training, and visualization in a streamlined manner. The use of a simple and intuitive web interface also enabled users to interact with the system effectively, making it accessible even to those without a technical background.

Moreover, this project highlighted the importance of modular design, data normalization, and sequence modeling when dealing with time-series data such as stock prices. It also provided hands-on experience in implementing neural networks and understanding the challenges associated with financial prediction, including the unpredictability and volatility of stock markets.

However, the system is not without its limitations. It currently lacks real-time data capabilities, robust risk analysis, and integration with multiple data sources. Additionally, the reliance on historical data alone restricts the model's ability to factor in real-world events, news, and sentiment that often have a significant impact on stock prices.

Looking ahead, this project serves as a strong foundation for future enhancements. Features like real-time data streaming, hybrid modeling, risk visualization, and a more advanced frontend can transform this prototype into a powerful financial tool. Incorporating external data such as news articles, technical indicators, and economic factors would also greatly enhance predictive performance and practical utility.

In summary, the stock prediction system successfully bridges the gap between theoretical machine learning concepts and real-world financial applications. It not only provides a valuable platform for understanding time-series forecasting but also opens opportunities for innovation and expansion in the domain of intelligent financial systems.

REFERENCES

1. **Brownlee, J. (2017).**"Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.

2. **Abadi, M., et al. (2016).** ***TensorFlow:*** A System for Large-Scale Machine Learning. OSDI, 16, 265–283..

3. **Goodfellow, I., Bengio, Y., & Courville, A. (2016).** *Deep Learning*. MIT Press.Parker, B. (2020). Concurrency and Multithreading in Web Server Design. O'Reilly Media.

4. **Zheng, A., & Casari, A. (2018).** Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O’Reilly Media.

5. **Zhang, G., Eddy Patuwo, B., & Hu, M. Y. (1998).** Forecasting with Artificial Neural Networks: The State of the Art. International Journal of Forecasting, 14(1), 35–62..