AI-based Financial Advisor for Stock Market Predictions with Cloud-Integration

MINOR PROJECT REPORT

Submitted by

Rachit Sood [RA2111028010110]
Papai Mondal [RA2111028010116]
Asmi Shrivastava [RA2111028010119]

Under the Guidance of

Mrs. Vijayalakshmi V

(Assistant Professor, Department of Networking and Communications)

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

with specialization in CLOUD COMPUTING

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

NOVEMBER 2024

Department of Networking and Communications
SRM Institute of Science & Technology
Own Work Declaration Form

Degree/ Course	**: B.Tech in CSE specializing in Cloud Computing**
Student Name	**: Rachit Sood, Asmi Shrivastava, Papai Mondal**
Registration Number	**: RA2111028010110, RA2111028010119, RA2111028010116**
Title of Work	**: AI-based Financial Advisor for Stock Market Predictions with Cloud Integration**

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g.fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

We understand that any false claim for this work will be penalized in accordance with the university policies and regulations.

DECLARATION:		
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.		
RACHIT SOOD RA2111028010110	PAPAI MONDAL RA2111028010116	ASMI SHRIVASTAVA RA2111028010119
Date: 18/11/24		
If you are working in a group, please write your registration numbers and sign the date for every student in your group.		

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. T. V. Gopal ,** Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work. We encompass our sincere thanks to **Dr. M. Pushpalatha**, Professor and Associate Chairperson, School of Computing, for her invaluable support.

We are incredibly grateful to our Head of Department, **Dr. M. Lakshmi**, Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinator, **Dr. G. Suseela,** Panel Head, **Dr. P Balamurugan** and Panel Member, **Mr. Karthikeyan H**, Department of Networking and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Mr. Prabakaran J**, Department of Networking and Communications, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Mrs. Vijayalakshmi V**, Department of Networking and Communications, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

Rachit Sood
[RA2111028010110]
Papai Mondal
[RA2111028010116]
Asmi Shrivastava
[RA2111028010119]

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP107L project report titled "**AI-based Financial Advisor for Stock Market Predictions with Cloud-Integration"** is the bonafide work of " **Rachit Sood [RA2111028010110], Papai Mondal [RA2111028010116], Asmi Shrivastava [RA2111028010119]"** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

Mrs. VIJAYALAKSHMI V
Assistant Professor
Department of Networking and Communications

Dr. LAKSHMI M
Professor and Head
Department of Networking and Communications

Examiner I

Examiner II

TABLE OF CONTENTS

ABSTRACT

AI-based financial advisors in the stock market are being enhanced by using the challenges outlined in the current systems to build solutions. The predominant objectives lie in enhancing the flexibility of the algorithms to enable them to adjust themselves to evolving market conditions, assurance of high data security for protecting sensitive financial data, and elimination of algorithmic bias for delivering fair and just recommendations. The report also highlights the need for establishing transparency and explainability in AI models, making sure they comply with regulatory standards to gain the trust of investors. These improvements aim to create a more reliable, secure, and user-centric AI-driven financial advisory platform for stock market investments.This report seeks to improve existing AI-based financial advisory systems by addressing critical issues such as inflexible algorithms, security vulnerabilities, and biased recommendations. Development in algorithmic flexibility, data security, and privacy will further enhance the platform providing safe, balanced, and innovative investment recommendations. The awareness of AI models would create more open architecture and more efforts towards complying with regulations, increasing investor confidence.This report aims to outline the development of a more reliable and scalable AI-powered financial advisory services platform deployed over cloud infrastructure. Cloud technology naturally ensures scalability, allowing market surveillance and stable performance 24/7. The final system will be user-to-user and consumer friendly, thus open to the broader audience, the gated solution guaranteeing safety and transparency providing accurate and personalized investment advisory on stocks.

LIST OF FIGURES

ABBREVIATIONS

AI - Artificial Intelligence

API - Application Programming Interface

AR - AutoRegressive

BAGGING - Bootstrap Aggregating (the technique used in Random Forest)

CSV - Comma-Separated Values

EDA - Exploratory Data Analysis

I - Integrated (referring to differencing)

MA - Moving Average

Max_depth - The maximum depth of each tree

RF - Random Forest

RMSE - Root Mean Squared Error

ROI - Return on Investment

SMAPE - Symmetric Mean Absolute Percentage Error

TN - True Negative

TP - True Positive

CHAPTER 1

INTRODUCTION

Managing personal finances in today's complex financial landscape has become increasingly challenging due to the abundance of information on income, expenses, taxes, loans, and investments. Developing structured financial plans is essential for achieving both short-term and long-term financial goals, yet the inherent unpredictability of financial markets makes planning difficult. Old Financial Advisors have traditionally assisted their clients over market volatility and have offered market strategies that have come at a tremendous cost, yet required further amendments to meet continuous market volatility.

While the rise of technology, particularly in the financial sector, has seen a reduced reliance on human consultants-a trend that has seen artificial intelligence and machine learning tools assist in personal finance management and investment advice-some major challenges still remain.The AI systems currently face a concurrence of challenges with regard to data protection and privacy, appropriately reduced algorithmic bias, and transparency in artificial intelligence. This, therefore, leads one to the caveat: to address these issues must bear a condition sine qua non - to build trust and thereby position AI-based advisory systems amid respectability.

1.1. Background and Motivation

Because of increasing cloud computing, several paths for developing and implementing AI-enabled financial advisory systems, particularly cloud-hosted "robo-brokerage" services, are opened up. Cloud infrastructure provides the capability to scale up, which is critical for this type of advice with system reliability to allow real-time market data observation on the execution of computations via AI algorithms. The ability to scale is extremely crucial in this case, as it allows continuous analysis of ever-growing databanks, ensuring the utmost reliability of financial advisory services on behalf of an investor. Consequently, the cloud-based AI solutions have been able to provide more consistent support than many other conventional models.

Nonetheless, hanging in the wings, however, there remain advantages that solutions based on AI still work under severe handicap constraints. Cloud resources might improve scalability; however, careful consideration arises when treating vast volumes of rapidly shifting data in markets that accurately forecast time for the development of market action.

Besides, predictive analysis concerning market trends has remained a serious problem for all automated systems. The uncertainty of the financial markets, especially in the stocks subsection, makes it difficult to predict future trends which the market follows due to the fact that market dynamics get influenced by various external, often unexpected, factors. This limitation repeatedly points to the degree of trust in AI-oriented financial systems, wherein the investors expect correctness and reliability with respect to these recommendations.

One big concern is the risk that exists for these advisory systems stemming from algorithmic bias. AI model biases, commonly arising from bias-laden training data or model structures, can skew financial recommendations toward certain disadvantaged user groups. Reduction of algorithmic bias rests on model updating as well as improving existing datasets using sufficiently different and representative input variables.

Finally, transparency in algorithmic processes is needed to cultivate trust, because explainability of recommendations presented by AI is crucial for building user confidence. Without transparency or the ability to verify the logic behind their recommendations, investors may be fearful of a full embrace of automated advisory services.

In an effort to provide insight into these difficulties, this report describes an AI-based cloud financial advisory system primarily focused on overcoming various limitations hindering AI-driven financial advisory services. This particular system engages core AI techniques of ARIMA model and Random Forest Regression, analyzing historical stock data and thus laying out reliable predictions regarding market evolution. . Implementation of these models in a cloud framework will afford the system high scalability and upgraded performance while operating on a cross-range of financial data - stocks, cryptocurrencies, and forex markets. Finally, the aim is to provide reliable stock forecasts and recommendations matching the needs of today's investor, therefore increasing comfort and adoption of AI in automated financial advisory systems.

1.2. Scope and Objective of the Project

The most prominent Sustainable Development Goal (SDG) for your AI-based personal financial advisor project is SDG 8: Decent Work and Economic Growth. This project promotes financial literacy and empowers individuals to make informed financial decisions, which can lead to improved economic stability and growth. By providing personalized financial advice, the project helps users manage their finances better, potentially increasing savings, investments, and overall financial well-being, thereby contributing to sustained economic growth.

1.3. SDG of the Project

To develop cloud-based dashboards and to leverage the scalability of cloud computing to efficiently handle large-scale data processing and complex computations required for real-time stock market analysis and to provide investment recommendations.

Investing in the stock market is complex, with constantly changing prices and global events affecting market conditions. Traditionally, financial consultants have provided advice, but their services are expensive and inaccessible for many:

- According to our research, the stock market presents significant challenges due to the unpredictable and volatile nature of stock prices.
- Although AI-based financial advisory systems have emerged to provide automated investment recommendations, they often face limitations in scalability, real-time data processing, and prediction accuracy.
- Moreover, issues such as data security, algorithmic bias, and transparency remain unresolved in many AI models, preventing widespread adoption.
- Our project focuses on using cloud computing to build a reliable, secure, and scalable AI financial advisor that addresses these challenges for stock market investments.

1.4. Software Requirements Specification

Google Collab:

- **Platform:** Cloud-based platform that provides a free Python development environment.
- **Key Features:**
  - Supports Python programming for data analysis, machine learning, and bioinformatics tasks.
  - Pre-installed libraries and easy integration with cloud storage (Google Drive).

- **Dependencies:**
    - Pre-installed libraries such as Pandas, Numpy, Scikit-learn, Matplotlib.
    - Ability to install additional Python packages using the !pip install command.

Jupyter Notebook:

- **Platform:** Local or cloud-based interactive environment for data analysis and visualization.
- **Key Features:**
    - Enables step-by-step code execution and visualization within the same document.
    - Facilitates easy sharing and documentation of code, results, and plots.
- **Dependencies:**
    - Requires Python (3.7 or above).
    - Libraries: Numpy, Pandas, Scikit-learn, Matplotlib, Seaborn, PADEL-Descriptor, LazyPredict.
    - Can be launched through Anaconda or installed using pip install notebook.

Python Libraries:

- Pandas: For data manipulation and preprocessing.
- Numpy: For numerical computations.
- Scikit-learn: For machine learning algorithms.
- Matplotlib and Seaborn: For data visualization.
- LazyPredict: For quick model comparison.

Streamlit Cloud:

Platform: Cloud-hosted environment that allows easy deployment and sharing of Streamlit applications for data science and machine learning.

Key Features:

- Provides a simple way to turn Python scripts into interactive web applications.
- Supports real-time data visualization and interaction, ideal for showcasing ML models and analytics.
- Allows collaborative and real-time feedback on data-driven applications.

Dependencies:

Requires Python (3.7 or above).

Essential libraries include:

- Streamlit: For creating and deploying applications (pip install streamlit).
- Pandas and Numpy: For data manipulation and preprocessing.
- Matplotlib and Seaborn: For data visualization.
- Scikit-learn: For implementing machine learning models.

Additional packages can be installed with pip install as needed.

Hosting and Deployment:

- Direct integration with StreamLit Cloud for public sharing.
- GitHub integration allows for automatic deployment and updates with every code push.

CHAPTER 2

LITERATURE SURVEY

Hui Zhu et al. [1] wrote a literature review and a stimulating perspective on how AI-enabled financial advisory services should be developed, stressing the disruptive potential of these services on human advisors in the retail investment market. Priyanka R. Rao et al. [2] addressed the role of fintech innovations such as robo-advisors in the sphere of personal finance, emphasizing their low cost and ease of use and evaluating their effects and some disadvantages on the behavior of investors. [3] summarized the use of Generative AI in financial decision-making, focusing on risk assessment, investment choices, and financial analysis, while considering the challenges of adopting GAI in finance. [4] discussed the role of Generative AI in virtual financial robo-advisors, exploring how AI can provide personalized asset management solutions and the importance of transparency and trust-building in such systems Zengyi Huang et al.

Sergiu-Alexandru et al.[5] reviewed the impact of advanced data management technologies in transforming financial decision-making, with a focus on the evolution of data storage and processing in the finance sector. [6] analyzed AI's positive impact on finance, including risk management, trading, and fraud detection, and discussed the challenges and ethical considerations Marko Ranković et al.M. F. Anshari et al. [7] have suggested a model where robo-advisors are integrated with digital twins for better personal financial management. According to J.K. Hentzen et al [8], with regards to AI applications in customer service activities, have conducted a systematic review, pinpointing research gaps and stressing the dichotomy between data-centric and theory-centric approaches. Implying the integration of AI in Islamic finance, S. Khan, [9] proposed an AI and NLP based chatbot which provides real-time Islamic finance advice This was the first of its kind applicable in Islamic Banking and finance. Balaji Dhashanamoorthi [10] gave an in-depth presentation on the application of AI within the banking and finance industry addressing the issues of cyber security and cybercrime but related to non-usual personal finance advisory services. A. Bhatia et al. [11] defined the understanding and recognition the Indian investors have about robo-advisors regarding their factors such as cost effectiveness while trust and data security appeals to adoption.

Manchuna Shanmuganathan [12] incorporated the effects of behavioral finance on the investment decision process through the use of artificial intelligence based robo-advisors. Alison Lui et al. [13] outlined an ‘augmented intelligence collaborator’, which can be used as a model of AI regulation that would help to regain trust towards the financial services sector. J.A. Al-Gasawneh et al. [14] studied the adoption of AI in the financial services context focusing on security perception, endorsements and perceived risk intensity and relationship. L. Fan et al. [15] have placed the diffusions of innovations and information search models as frameworks in the context of robo investors. [16] P. Gomber et al. interpreted the forces of innovation disruption and transformation in the financial services industry which was emanated by the fintech advancements. [17] systematically reviewed literature on AI in customer interaction in financial services and suggested future research opportunities J.K. Hentzen et al. [18] T. Kraiwani et al focused on the social acceptance of financial robo-advisors in the context of developing countries and explored some of the determinants. D. Kwon et al.[19] performed empirical investigation regarding the factors affecting the intention to utilize the service of robo-advisors. D. Belanche et al. [20] addressed the adoption of AI in fintech: users’ attitudes towards robot advisors.

Pokhrel et al. [21] analysed and compared the performance of three deep learning models, LSTM, GRU, and CNN, in predicting the next day’s closing price of the Nepal Stock Exchange (NEPSE) index. The study uses fundamental market data, macroeconomic data, technical indicators and financial text data of the stock market of Nepal.Flesch et al. [22] uncovered that the “rich” learning approach, which structures the hidden units to prioritise relevant features over irrelevant ones, results in neural coding patterns consistent with how the human brain processes information. Additionally, they found that these patterns evolve as the task progresses. Baek and Kim [23] proposed ModAugNet, a framework integrating a novel data augmentation technique for stock market index forecasting. The model comprises a prediction LSTM module and an overfitting prevention LSTM module. The performance evaluation using S&P500 and KOSPI200 datasets demonstrated ModAugNet-c’s superiority over a monolithic deep neural network.

In order to predict challenging financial markets’ fluctuations and accurately forecast them, researchers have proposed several machines and deep learning methods such as the CNNs, the variants of RNNs, namely the GRU and LSTM, and their hybrid and single architectures. For example, Galeshchuk and Mukherjee [24] suggested a CNN for predicting the price change direction in the Forex market. They utilized the daily closing rates of EUR/USD, GBP/USD, and USD/JPY currency pairs. Moreover, they compared the results of CNN with baseline models. Althelaya et al. [25] investigated LSTM architectures to forecast the closing prices of the S&P 500 for eight years. Their findings showed that the Bidirectional LSTM (BLSTM) was the most appropriate model; outperforming the MLP-ANN, the LSTM and the stacked LSTM (SLSTM) models; achieving the lowest error in the short- and long-term predictions. Lu et al. [26] proposed a predicting technique for stock prices employing a combination of CNN and LSTM, which utilizes the memory function of LSTM to analyze relationships among time series data and the feature extraction capabilities of CNN. Their CNN-LSTM model uses opening, highest, lowest and closing prices, volume, turnover, ups and downs, and change as input and extracts features from the previous ten days of data. Their method is compared to other forecasting models.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 Components Architecture

This architecture Fig. 3.1.1 integrates multiple predictive models and data sources to offer comprehensive financial insights within the Financial Advisor application. The Stock Prediction System uses both ARIMA and Random Forest Regression models to forecast trends, sourcing data from the YFinance DB for stock.

Additionally, the Forex Research Analysis component uses LLM + GPT-4 for foreign exchange market research and language-based analysis, enhancing the consultancy's advisory offerings. A Finance ChatBot powered by LLM enables users to interact with the system, allowing for real-time, conversational insights into stocks, cryptocurrencies, mutual funds, and forex trends. This cohesive setup provides a robust, data-driven financial advisory platform, making it easier for users to gain insights into multiple financial domains.

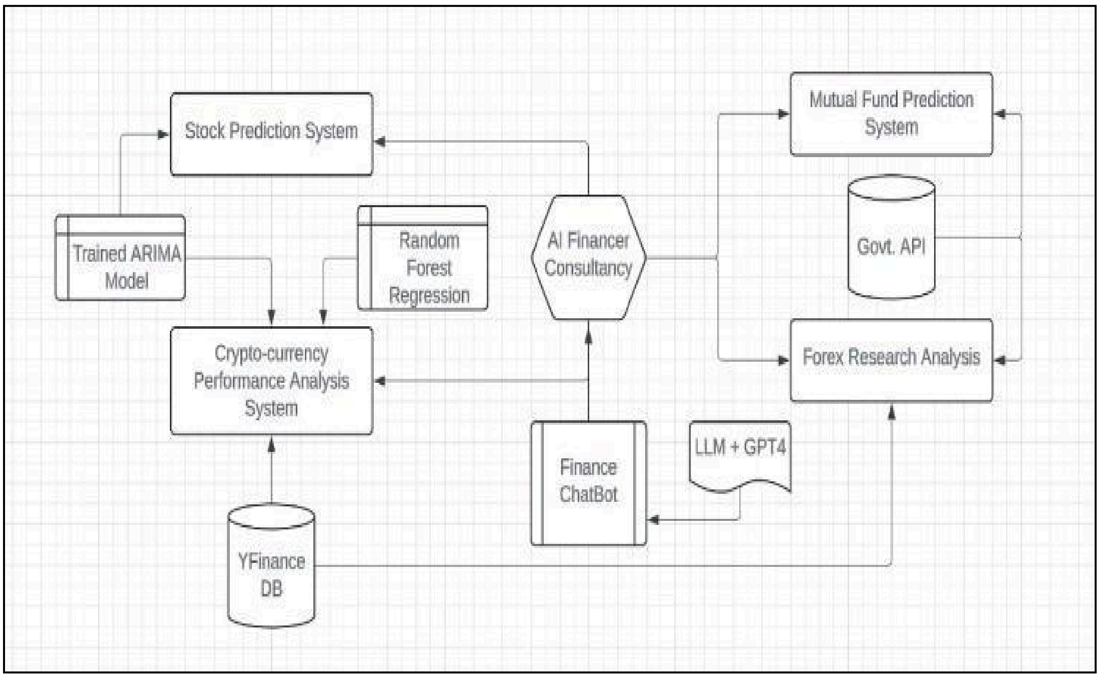

Fig. 3.1.1 Components architecture diagram

3.2 Design of Modules

Module 3.2.1: Data Collection:

Objective: First, facilitate the collection of high-quality and extensive stock market information, automatic updates on data at the end-of-the-day session, and storage in such a way that would allow its easy access and utilization for training modeling.

Steps:

1. **Getting the Historical Stock Market Data:**:
   - To focus on obtaining core stock data like daily stock prices, stock trading volume, dividends, and stock splits as fundamental features for developing the predictive model.
   - Selected yahoo finance as a credible source for its wide-range historical data and easy accessibility, along with viability for other financial indicators.
2. **Utilized the yfinance Library**:
   - The yfinance library is used to ease the process of accessing and retrieving data from Yahoo Finance. It is an easy-to-use Python library that fetches a plethora of financial data directly from Yahoo Finance using API.
   - Used yfinance built-in functions to retrieve various financial indicators, including:
     - **Stock prices (open, high, low, close)** for each trading day.
     - **Trading volume** facilitates an understanding of the number of shares traded.
     - **Dividends and stock splits** that help correct for corporate actions that have affected the stock value over time.
3. **The Data will need to be stored in a structured form (CSV, database):**
   - Had to choose a storage format that enables easy access and processing during modeling for training and testing. We chose a simple CSV because it is compatible with most of the data processing tools.
   - Otherwise, a database can be implemented, which has the potential to offer faster querying and filtering of the specific records for very large volumes of data, such as filtering based on a date range, stock symbol, or other criteria.

4. **Documented the Data Collection Process**:

   - Produced documentation describing the data-retrieval method, including source descriptions, variables collected, and the details of the automated scripts.

Module 3.2.2: Data Preprocessing:

Objective: To clean the stock market data and put it in a form that can be used to check missing values, deal with outliers, and map it to a certain frequency. This ensures that the data is uniform and hence is suitable for training a model capable of providing consistent results. From a practical perspective, calculating SMAPE provides an early measurement of accuracy in forecasting.

Steps:

1. **Data Cleaning and Missing Value Treatment**:
   - During the analysis of data, there were missing values and inconsistencies, which are often seen in financial data owing to incomplete records.
   - Filled missing values, where necessary, with forward-filling for continuous data or eliminated rows when the gaps were too wide for imputation.
2. **Frequency-Based Data Consistency**:
   - Removed incomplete records that were contrary to the chosen frequency of time, be it daily, weekly, monthly, or quarterly. For example, a week with missing days would be removed for a week-wise analysis.
   - Aligned all entries in a particular time period, resulting in a uniform set of data that can facilitate sound analysis and model training.
3. **Data Normalization**:
   - Applied normalization techniques to scale numerical features within a similar range, making the model more robust to varying scales of data, especially when combining technical indicators with stock prices.
   - Standardized features by scaling them to a fixed range (e.g., between 0 and 1), which helps reduce model bias toward larger values and speeds up the training process.
4. **Ensuring Correct Alignment of Data with Frequency**:
   - Confirmed that data points aligned precisely with the selected frequency. For instance, if data is set to daily, each day should have a corresponding entry; if set to monthly, each month should have one aggregated entry.
   - Gaps are filled where necessary to avoid discrepancies while training and testing under the frequency setting.

5. **Calculating Symmetric Mean Absolute Percentage Error (SMAPE)**:

- While SMAPE itself is computed, an appropriate measure of accuracy of prediction entails the summation of errors with respect to size and direction. SMAPE is advantageous in stock forecasting as it measures percentage errors symmetrically, preventing bias towards large values.
- Used SMAPE in the preprocessing phase to assess how well the model's predictions align with real values, helping guide feature selection and model adjustments.

Module 3.2.3: Model Development:

Objective: Building and fine-tuning a series of time series and machine learning models including ARIMA, Holt-Winters, SES, and Random Forest-have been designed together to capture the linear and nonlinear periodic stock price patterns through this module to optimize each respective model's parameters, and assessing them through SMAPE measures.

Steps:

1. **Modeling the Training Time Series Prediction Models**:

- The various time series modeling were constructed to capture broad trends occurring within the historical stock price data.
- **The ARIMA (AutoRegressive Integrated Moving Average)** model was selected for its growing ability to discover linear relationships in time series data in this modeling phase. It configured ARIMA to have specific values for p (AR order), d (differencing order), and q (MA order) in order to capture trends and seasonal patterns.
- **Holt-Winters** is an ideal candidate for modeling time series data with repeating patterns, which utilizes level, trend, and seasonal components for forecasts.
- **Simple Exponential Smoothing (SES)**: Used for fewer term forecasts, SES generalized enabled the smoothing of data while predicting values when immediate, short-term trends were pertinent and foretold.

2. **Fine-Tuning Model Parameters**:

- The models became determined to tune parameter optimization in an attempt to increase forecasting availability:
- ARIMA involves adjusting values of p, d, and q through the use of autocorrelation and partial autocorrelation plots which aided in identifying the best-fitting

parameters based on historical data.

- Holt-Winters parameter optimization in regards to the trend and seasonality to accurately capture cyclical behaviors.

3. **Hyperparameter Tuning for Random Forest**:

- Tuning increasingly fine hyperparameter Random Forest to improve its model performance:
    - **Number of Estimators**: The problem of balancing the number of decision trees built versus the accuracy of the model and the time taken to compute became fully relieved.
    - **Tree Depth**: This highly optimized adjustment of the maximum tree depth allowed the model to capture intricate patterns without uncontrollably overfitting into the training data.
- A variety of hyperparameter combinations were examined through grid search or random search regarding a range of hyperparameter values to minimize the error.

4. **Evaluating Model Performance with SMAPE**:

- The primary evaluation metric was chosen to be the Symmetric Mean Absolute Percentage Error (SMAPE), which renders a balanced measurement of forecast errors at either extremes (the highest or lowest stock price levels).
- SMAPE was selected for the practical percentage error it provides; this means the error can be easily interpreted in terms of assessing how accurate the forecast is and comparing it between models.

Module 3.2.4: Model Evaluation and Selection:

Objective: The aim of this module was to assess six forecasting methods on the accuracy of the stock price prediction (this includes ARIMA, SES-Moving Averages, Simple Exponential Smoothing, Random forests, Holt s Winter, and Multilayer perceptron) using two indicators-SMAPE and RMSE-to identify one model that can be proposed for stock prediction. By comparing both approaches, traditional and machine learning, this module also will lead to the best outcomes of stock price prediction.

Steps:

1. **Evaluating Model Accuracy with SMAPE and RMSE**:

- The useful metrics were chosen to directly measure model performance in terms of accuracy in these experiments, namely, SMAPE and RMSE. SMAPE, which is expressed as a percentage, allows for a convenient interpretation of prediction errors across differing ranges of stock prices, while RMSE measures absolute error and weighs errors larger than some cutoff level from reality more heavily than acceptably small deviations.

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|F_t - A_t\right|}{\left(\left|A_t\right| + \left|F_t\right|\right)/2} \times 100$$

Where:

F_t = Forecasted value at time *t*

A_t = Actual value at time *t*

n = Total number of data points

- Calculations of SMAPE and RMSE for each model's prediction consisted of high and low stock prices.

2. **Comparison Between Traditional and Machine Learning Models**:

- Compared were the traditional time series models with machine learning models, such as Random Forest.
- Generally, traditional models, such as ARIMA, tend to perform well in capturing linear and seasonal patterns, while Random Forest can deal better with complex nonlinear relationships.

3. **SMAPE Analysis for Each Model**:

- **Simple Exponential Smoothing (SES)**:
  - SMAPE for High: 0.028
  - SMAPE for Low: 0.03
  - At those error percentages, it can be seen that SES was moderately accurate, although, overall, it lagged in its ability to capture fluctuations across stock prices relative to other methods.
- **ARIMA Model**:
  - SMAPE for High: 0.018
  - SMAPE for Low: 0.021
  - SMAPE values of the smaller order indicate that ARIMA performed well for predicting both high and low price values, therefore, enabling its usage as a solid choice for forecasting linear trends.

- **Random Forest**:
  - SMAPE for High: 0.016
  - SMAPE for Low: 0.019
  - Among all the models employed, Random Forest exhibits a SMAPE as low as those of the other models considered; this suggests that this model has most closely related the predicted price values to the actual values in this regard, more or less with the complex and nonlinear nature of the structures.

4. **Selecting the Most Accurate Model**:
   - Taking into consideration SMAPE and RMSE random forest comparison, it takes the first position. Be sure not to have low and high p-values gets it to be the most accurate model, hence giving very low error.
   - ARIMA works correctly under linear trend but random forest's ability to capture both linear and nonlinear patterns is more efficacious to consistent forecasting across a big range of stock intervals.

Module 3.2.5: Cloud-Based Deployment:

Objective: To host the selected stock prediction model on Streamlit Cloud, giving users an interactive interface for real-time predictions.

Steps:

1. **Deploying the Model on Streamlit Cloud**:
   - The model, selected for stock prediction, will be hosted on Streamlit Cloud, hence open to users for a friendly online web interface.
   - Streamlit Cloud was picked because of its ease of developing interactive dashboards and applications, which suits the requirements of an AI-based stock advisor.
2. **Configuring the Deployment Environment**:
   - Ensured the activating of the Streamlit Cloud, making sure that all model execution-inclined libraries and frameworks, such as pandas, scikit-learn, and yfinance-fit-for-acceptance-were available.
   - Conducted tests to ascertain the cloud infrastructure's compatibility with Streamlit, thereby ensuring that the model placed itself well into action according to customer requests without a hitch.
3. **Managing Dependencies with requirements.txt**:
   - Designed a requirements.txt file listing all necessary Python-based libraries to

facilitate seamless installation and configuration on the Streamlit Cloud.

- This enabled the cloud to automatically install all dependencies, thereby easing setup for deployment.

4. **Designing User-Friendly Error Handling**:

- Custom added messages that would guide the users on errors, for instance when wrong inputs or unwanted data could not be provided. The aim of improving the user experience.

CHAPTER 4

METHODOLOGY

The AI-based financial advisor leverages the Auto Regression Integrated Moving Average (ARIMA) model to analyze trends in stock predictions by effectively modeling time-series data. ARIMA's strengths lie in its handling of trend, seasonality, and statistical events within stock price data. First off, multiple historical stock data sources are aggregated and uploaded to a cloud place to store them, and then they are loaded on demand in the necessary format. This is necessary for ARIMA - ARIMA the series of data entering it should be stationary or have some transformers like differencing, applied to make the series stationary. Such a requirement has nothing to do with LSTM.VALUES! . Cloud computing resources are utilized for model training and parameter tuning, specifically to optimize ARIMA's parameters—p (autoregressive order), d (degree of differencing), and q (moving average order).

Once the data is stationary, ARIMA uses grid search techniques within a cross-validation framework to fine-tune these parameters, which enhances model accuracy and reliability. After completing the training, the ARIMA model was deployed via Streaming Cloud to allow users to input stock symbols and be presented with predictions for future high and low stock prices. Since the model can now reside in Cloud space, it claims its scalability to process new incoming data in a real-time manner. This is crucial, as stock market predictions are often made under a fast, time-sensitive environment-almost instantaneous. By this means, the AI system is able to keep evolving and adapt to the most recent changes in the market, thus giving investors responsive and accurate financial advice.

The Random Forest Regression is also applied with the additional advantage of being able to model complex nonlinear relationships in stock market data that are not identified by simple linear ARIMA model. As a supervised learning method, Random Forest lowers the very high variance of the individual decision trees by creating them into a "forest," thereby making predictions more accurate by aggregating the outputs of several trees. This ensemble-based approach does increase robustness and accuracy in stock price prediction, especially in volatile markets where linear models might not be able to deliver much.

4.1 YFinance Data Retrieval

Data collection forms a backbone of the financial advisory system. The primary source of stock market data is YFinance DB (Yahoo Finance Database), which provides historical stock prices, market indices and other key financial data for model training and forecasting. YFinance DB’s extensive database supplies essential inputs for time-series analysis and predictive modeling, ensuring that the models have access to comprehensive data for accurate predictions. Historical data are uploaded to a secure cloud platform, facilitating the storage of large datasets needed for ongoing model training and deployment.

The ARIMA model requires tweaking through pre-processing before it can be used for training. Such techniques as differencing to eliminate trends and stabilize the series should be done for this step. This process is critical because the forecasting capabilities of the ARIMA modeling approach are critically dependent on the data being stationary. On the other hand, shuffled forest regression processes inputs of raw historical data and other financial indicators, such as market volatility, historical highs, and lows, into it. Random Forest does not apply stationary data, which further means it could capture the non-linearities and dynamics of the market.

By integrating other data sources, this real-time and historically rich information from government agencies lends a veil of trust to ensure that the model has accommodated a broad array of economic factors. The AI engine adopts many data pipelines to uphold this diversity and augment the engines' prospects of forecasting in stock, forex, and cryptocurrency markets to deliver relevant and cogent financial advice.

4.2 Model Design

The AI-based advisory system’s model design includes ARIMA for time-series analysis and Random Forest Regression for predictive modeling. ARIMA is used for analyzing trends over constant time. It requires data preprocessing to create stationary time series, using differencing techniques to remove trends and seasonality. This whole process enables ARIMA to devise its models based on past price movements and accurately forecast prices; however, it may be hampered in dealing with sudden and non-linear changes. Lagged price patterns are captured by AR components of ARIMA, while MA components deal with noise that enhances the prediction's stability. . The ARIMA model is thus ideal for predicting steady trends, even if it is less responsive to abrupt market fluctuations.

In contrast, Random Forest Regression is used to handle the non-linear and dynamic nature of stock market data. By processing multiple trees in parallel, Random Forest incorporates a multitude of decision trees, where each tree learns from a subset of the data and contributes to an aggregated prediction output. Such an approach helps avoid complex dependencies, persisting in their forms with the data. For regression, it gives an ensemble average on the outputs, so there is lesser chance of overfitting and better accuracy rate is given.Random Forest is complemented with ARIMA thus accepting unpredictability of market movements making it more robust to volatile conditions.

The system implementation occurs in the cloud-based setting via cloud-hosted Streamlit which is an interactive prediction model. Here users select stocks and access real-time prediction output.A personalized AI chatbot based on LLM + GPT-4 is offered to answer users about stocks and get investment recommendations based on model insights. Additional features include the opening page where users create investment accounts, get real-time tracking of market trends, and a structured tabbed setting to navigate stock information. All these features provide investors with the cloud-hosted AI-based financial advisor which is set to simplify the investment decision-making process.

4.3 Output Generation and Interpretation

We are instructing the pip package manager to install the yfinance library. Fig. 4.3.1 shows pip as a tool used to manage Python packages, and yfinance is a library specifically designed for downloading historical market data from Yahoo Finance. Output: The subsequent lines in the image are the output from pip as it checks for the necessary dependencies and their installation status.

```
!pip install yfinance

Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-packages (0.2.46)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.26.4)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.10/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.9.4)
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.3.6)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2024.2)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-packages (from yfinance) (3.17.7)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.12.3)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.6)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3.0->yfinance) (2024.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2024.8.30)
```

Fig. 4.3.1 Setting up libraries

1. **Data Manipulation and Analysis:**
    - pandas as pd: Imports the pandas library, a powerful tool for data analysis and manipulation. It's often used for working with structured data in tabular form (like CSV or Excel files).
    - numpy as np: Imports the NumPy library, which provides efficient numerical computations and array operations. It's frequently used for mathematical calculations and scientific computing.
2. **Data Visualization:**
    - seaborn as sns: Imports the Seaborn library, built on top of Matplotlib, for creating aesthetically pleasing statistical visualizations. It's often used for exploratory data analysis and generating insightful plots.
    - matplotlib.pyplot as plt: Imports the Matplotlib library, a comprehensive plotting library that provides a wide range of customization options for creating various types of plots.
3. **Inline Plotting:**
    - %matplotlib inline: This Jupyter Notebook magic command sets up the environment to display plots directly within the notebook.
4. **Additional Imports for Stock Data:**
    - yfinance as yf: Imports the yfinance library, which allows you to fetch historical market data from Yahoo Finance. This is useful for analyzing stock prices, dividends, and other financial metrics.
    - datetime: Imports the datetime module, which provides tools for working with dates and times in Python.
    - pandas_datareader.data as pdr: Imports the pandas-datareader library, which provides functions for fetching data from various online sources, including Yahoo Finance.
    - yf.pdr_override(): This line overrides the default data reader to use yfinance, ensuring that you can fetch data using the yfinance library.

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('whitegrid')
plt.style.use("fivethirtyeight")

# For reading stock data from yahoo
import yfinance as yf
from datetime import datetime
from pandas_datareader import data as pdr
from pandas_datareader.data import DataReader
yf.pdr_override()
```

Fig. 4.3.2 Reading dataset

5. **Define Stock and Date Range** (Fig. 4.3.3)**:**
    - stock = 'TSLA': Sets the stock symbol to 'TSLA' for Tesla.
    - end = datetime.now(): Sets the end date to the current date and time.
    - start = datetime(2020, 1, 1): Sets the start date to January 1, 2020.
6. **Fetch Stock Data**:
    - df = yf.download(stock, start, end): Downloads the stock data for TSLA between the specified start and end dates using the yf.download() function. The data is stored in a pandas DataFrame named df.
7. **Display the Latest 10 Days:**
    - df.tail(10): Displays the last 10 rows of the DataFrame, representing the most recent 10 days of stock data.
    - The summary stats and attributes of the dataset are retrieved simultaneously (Fig 4.3.4).

```python
# The tech stocks we'll use for this analysis
stock = 'TSLA'

end = datetime.now()
start = datetime(2000, 1, 1)
start = '2020-01-01'

df = yf.download(stock, start, end)
df.tail(10)
```

```
[*********************100%***********************]  1 of 1 completed
```

Date	Open	High	Low	Close	Adj Close	Volume
2023-03-02	186.740005	193.750000	186.009995	190.899994	190.899994	181500700
2023-03-03	194.800003	200.479996	192.880005	197.789993	197.789993	153800400
2023-03-06	198.539993	198.600006	192.300003	193.809998	193.809998	128100100
2023-03-07	191.380005	194.199997	186.100006	187.710007	187.710007	148125800
2023-03-08	185.039993	186.500000	180.000000	182.000000	182.000000	151897800
2023-03-09	180.250000	185.179993	172.509995	172.919998	172.919998	170023800
2023-03-10	175.130005	178.289993	168.440002	173.440002	173.440002	191007900
2023-03-13	167.460007	177.350006	163.910004	174.479996	174.479996	167790300
2023-03-14	177.309998	183.800003	177.139999	183.259995	183.259995	143717900
2023-03-15	180.800003	182.339996	176.029999	180.449997	180.449997	145632300

Fig. 4.3.3 Set up End and Start times for data grab

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3200 entries, 2010-06-29 to 2023-03-15
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Open       3200 non-null   float64
 1   High       3200 non-null   float64
 2   Low        3200 non-null   float64
 3   Close      3200 non-null   float64
 4   Adj Close  3200 non-null   float64
 5   Volume     3200 non-null   int64
dtypes: float64(5), int64(1)
memory usage: 175.0 KB
```

Fig. 4.3.4 Summary Stats

Fig. 4.3.5, We generated a line plot to visualize the historical closing price of a stock, in this case, Tesla (TSLA). The plot displays how the stock price has changed over time, showing periods of growth, decline, and volatility. The upward trend in the plot indicates that the stock price has generally increased over the years, suggesting positive performance for Tesla.

```python
# historical view of the closing price
plt.figure(figsize=(10, 6))
df['Adj Close'].plot()
plt.ylabel('Adj Close')
plt.xlabel(None)
plt.title(f"Closing Price of {stock}")

plt.tight_layout()
plt.show()
```

Fig. 4.3.5 Historical view of the closing price

```python
# total volume of stock being traded each day
plt.figure(figsize=(10, 6))
df['Volume'].plot()
plt.ylabel('Volume')
plt.xlabel(None)
plt.title(f"Sales Volume of {stock}")

plt.tight_layout()
plt.show()
```

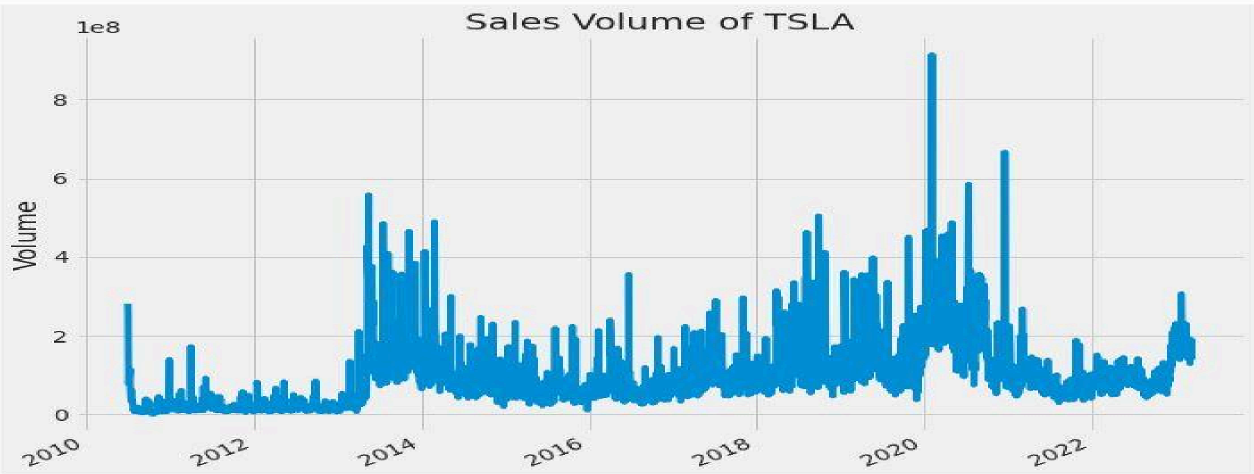

Fig. 4.3.6 Plot the total volume of stock being traded each day

We needed to generate a line plot to visualize the sales volume of a stock, in this case, Tesla (TSLA), over time. The plot (Fig. 4.3.7) shows how the trading volume has fluctuated over the years. The plot reveals several insights:

1. Overall Trend: The plot shows an increasing trend in trading volume over the years, indicating growing interest and activity in the stock.
2. Volatility: The plot also shows periods of high and low volatility, with some spikes representing days with significantly higher trading volumes. These spikes might be associated with specific news events, earnings announcements, or other factors that influence investor interest.
3. Seasonal Patterns: There might be seasonal patterns in the trading volume, with higher volumes during certain periods of the year. This could be due to factors like quarterly earnings reports or holiday seasons.

By analyzing this plot, investors can gain insights into market sentiment, identify potential trends, and make informed decisions.

```
# pct_change to find the percent change for each day
df['Daily Return'] = df['Adj Close'].pct_change()

# Then we'll plot the daily return percentage
plt.figure(figsize=(15, 8))
plt.plot(df['Daily Return'])
plt.title('Daily Returns')
plt.show()
```

Fig. 4.3.7 Plot the daily return percentage

Lastly to create a line plot to visualize the actual and predicted values of a time series dataset. The dataset seems to represent the price of Brent oil over time. The plot in Fig. 4.3.8 shows three lines:

1. Train: This line represents the actual values of the training data.
2. Val: This line represents the actual values of the validation data.

3. Predictions: This line represents the predicted values generated by a model for the validation data.

The goal of the model is to predict future values based on past data. The plot helps assess the model's accuracy by comparing the predicted values to the actual values in the validation set.

Ideally, the "Predictions" line should closely follow the "Val" line, indicating that the model is making accurate predictions. Deviations between the two lines suggest areas where the model could be improved.

```
# Plot the data
train = data.iloc[5300:training_data_len, :]
valid = data.iloc[training_data_len:, :]
valid['Predictions'] = predictions

# Visualize the data
plt.figure(figsize=(20, 6))

plt.title('Model')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price USD ($)', fontsize=18)

plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='best')

plt.show()
```

```
<ipython-input-86-6563f516ec06>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid['Predictions'] = predictions
```

Fig. 4.3.8 Visualize the predictions data

CHAPTER 5

IMPLEMENTATION

5.1 Sprint Planning

5.1.1 Sprint Backlog

Fig. 5.1.1 provided shows the sprint backlog template for the project. This sprint backlog outlines different tasks associated with different epics such as "Authentication" and "Stock Prediction". Each task includes a user story, such as "As a user, I want to be able to securely access the web through the cloud," which helps clarify the end user's needs. The priority of each task is marked as "Must" within the MoSCoW priority model, emphasizing that these tasks are critical to the sprint. The status bar displays the current progress of each task, with stages such as "In Progress", "Testing" and "Completed" offering an overview of the team's progress. Acceptance criteria, functional requirements and non-functional requirements are specified for each task, ensuring that both user expectations and technical standards are met. Functional requirements for this role include secure password hashing, while non-functional requirements specify that the response time should be less than two seconds. The "Original Estimate" and "Actual Effort" columns help track the estimated and actual time (in days) spent on each task, enabling better time management and sprint planning.

Sprint Backlog sample template for AI Based Personal Finance Advisor (Detailed Estimate for current sprint)											
ID		Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
1	Sprint	User Authentication	Authentication	As a user, I want to securely access the website through cloud.	Must	In Progress	1. The user can register with a valid email and password. 2. User can log in with registered credentials. 3. Password reset functionality is secure and straightforward.	Secure password hashing and storage. Account lockout after multiple failed login attempts.	Response time for authentication actions should be less than 2 seconds. Password reset emails should be delivered within 5 minutes.	5 days	
2		Stock Selection	Stock Prediction	As a user, I want to select a country, stock exchange, and stock so that I can view specific stock data	Must	In Testing	1. User can successfully select a country, stock exchange, and stock from dropdown lists.	Implement dropdown menus for country, stock exchange, and stock.	Selection should be quick with no significant lag.	5 days	5 days
3		Forecast Horizon	Performance	As a user, I want to adjust the forecast horizon to see predictions for different time ranges.	Must	In Testing	1. Forecast horizon slider adjusts the number of forecasted days shown in the graph.	Implement an interactive slider for forecast horizon.	The slider should update the forecast graph in real-time without refreshing the page.	3 days	3 days
4		Date Range Selection	Stock Prediction	As a user, I want to select a date range for historical data analysis so that I can control the period displayed.	Must	Completed	1. User can set start and end dates.	Implement a date picker for start and end dates.	The date picker should work smoothly and allow selecting past and future dates.	4 days	4 days
5		Model Parameters	Stock Prediction	As a user, I want to fine-tune the parameters of the forecasting models to optimize the results.	Must	Completed	1. User can adjust parameters (e.g., alpha for smoothing models) and see immediate impact on the forecast.	Allow dynamic adjustment of model parameters and show real-time updates to the prediction graph.	Adjusting parameters should not cause significant performance issues.	3 days	3 days

Fig. 5.1.1 Sprint Backlog

5.1.2 Functional Test Cases

The figure 5.1.2 shows a functional test case template for the "Sprint Final" of an AI-based personal finance project. It lists key features that are tested such as UI responsiveness, stock selection features, ARIMA parameter adjustment, SMAPE value calculation, model selection, error handling for invalid input, and Random Forest prediction. Each test case contains the steps to execute the test, the expected output and the actual output, with all tests marked as "Success". For example, in test case TC001 for UI Responsiveness, the application is opened on different devices and resized, with expected and actual outputs confirming that the UI will scale correctly. Similarly, additional tests confirm the successful performance of loading data from the warehouse, modifying parameters, and handling errors, ensuring that all functions will work as intended in the final sprint.

Sprint Final

Functional Test Case Template

Feature	Test Case	Steps to execute test case	Expected Output	Actual Output	Status	More Information
Validate UI responsiveness	TC001	1. Open the app on different devices 2. Resize browser window	UI elements adjust without overlapping or breaking	UI adjusted perfectly on mobile and desktop views	Pass	
Verify stock selection functionality	TC002	1. Open the stock prediction app 2. Select a stock from the dropdown	Selected stock data should load on the interface	Stock data for the selected stock (For eg. BHEL.BO) loaded successfully	Pass	
Test ARIMA parameter adjustment	TC003	1. Adjust AR, MA, and differencing values 2. Click "Apply"	Predictions graph updates with new ARIMA values	Graph updated with new ARIMA parameters	Pass	
Test SMAPE value calculation	TC004	1. View the SMAPE value for the chosen model	Correct SMAPE value is displayed with % difference compared to ARIMA	SMAPE values displayed correctly with improvement over ARIMA	Pass	
Validate model selection	TC005	1. Select a model from the model dropdown (e.g., ARIMA, Random Forest)	Model parameters should appear for configuration	Parameters for ARIMA and Random Forest appeared successfully	Pass	
Verify error handling for invalid input	TC006	1. Enter invalid stock symbol or leave required fields blank 2. Click "Predict"	Error message displayed, and no prediction is run	Error message "Invalid stock symbol" displayed, no prediction run	Pass	
Validate Random Forest prediction	TC007	1. Set Random Forest parameters 2. Click "Predict"	Predicted stock prices for the given range are displayed	Random Forest predicted stock prices successfully displayed	Pass	

Fig. 5.1.2 Functional Test Cases

5.1.3 MS Sprint Planner

Lastly MS Planner in Fig. 5.1.3, a task management tool within Microsoft 365 designed to help teams organize, assign, and track work visually was used by us to create plans, add tasks, assign team members, and set due dates within a dashboard that uses a Kanban-style board layout, allowing for easy viewing of task progress and team workload.

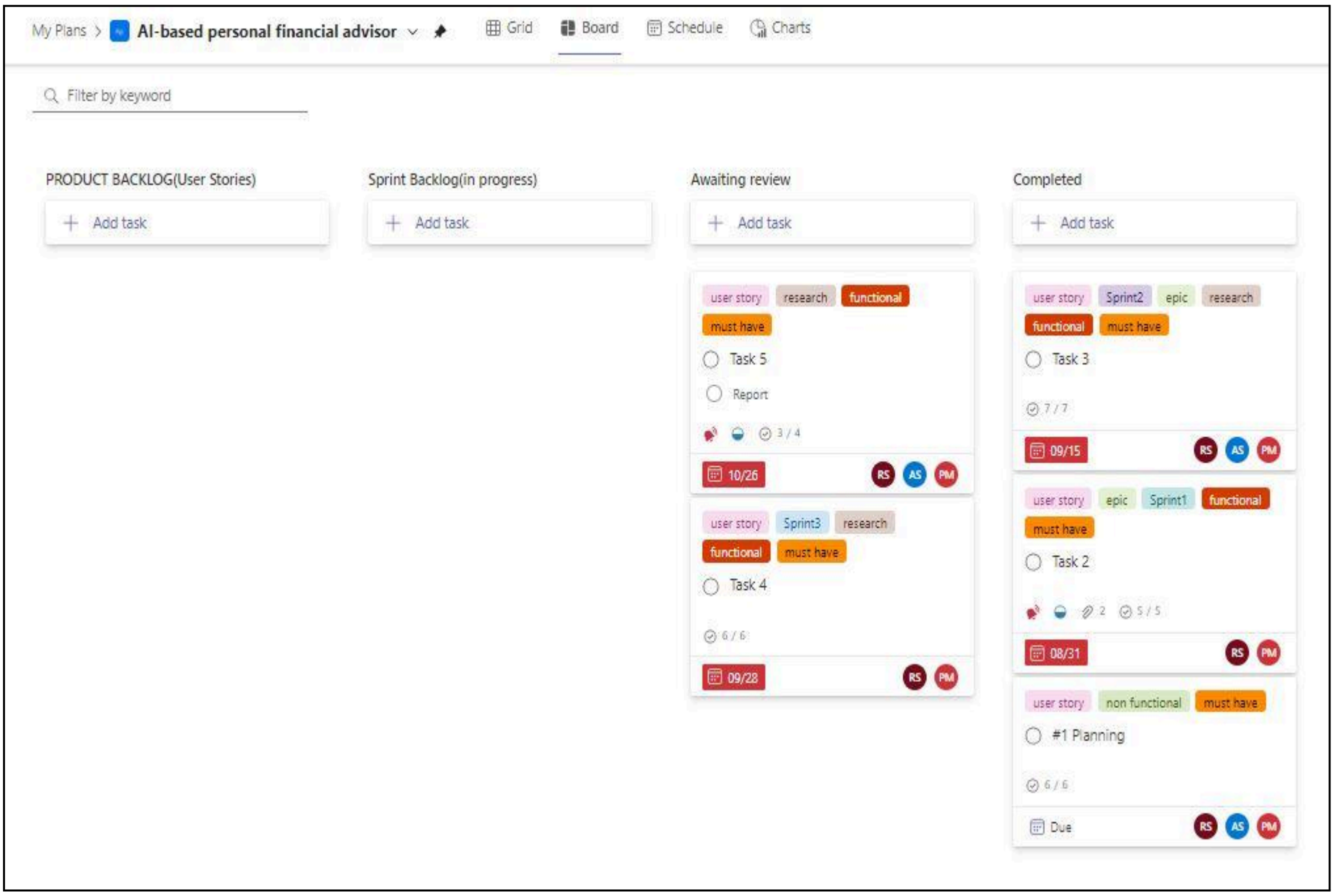

Fig. 5.1.3 MS Planner

5.2 Key Code Components

The Key code components encompass functions for loading datasets, training models, and processing the inputs:

5.2.1 Stream Integration with models

The Streamlit application is aimed at providing an easy-to-use interface for the prediction of the highest/lowest stocks prices based on historical prices. The app works with SQLite to obtain available countries, exchanges, and stocks therefore making it possible to filter the stocks by the country and exchange, stock ticker, dates and intervals as well as create a data period. Stock price data is gathered via the yfinance API. In forecasting the high and low prices, users can opt for any of the numerous forecasting models e.g. Simple Exponential Smoothing, Holt-Winters, ARIMA, Random Forest amongst others.

With each model, there are adjustable parameters such as smoothing levels, trend and seasonal factors. These models are also depicted through line charts with error measurements e.g SMAPE. There is also an extra feature that has been integrated into the software allowing users to hide the GitHub icon through the use of a CSS style. The connection with the SQLite database is also terminated at the end of the working session.

//CODE STARTS

```
import streamlit as st

import pandas as pd

import sqlite3 as sq

import datetime

import yfinance as yf

from preprocess import preprocessing

#import matplotlib as mpl

#import matplotlib.pyplot as plt

import warnings


def hide_github_icon():

    hide_github_style = """

        <style>

        /* Hide the GitHub icon from the Streamlit app menu */

        button[kind="header-button"] {

            display: none;

        }

        </style>

    """
```

```
    st.markdown(hide_github_style, unsafe_allow_html=True)

warnings.filterwarnings("ignore")

db = sq.connect('stocks.db')

# get country

query = "SELECT DISTINCT(Country) FROM tkrinfo;"

country = pd.read_sql_query(query, db)

choice_country = st.sidebar.selectbox("Pick country", country)

# get exchange

query = "SELECT DISTINCT(Exchange) FROM tkrinfo WHERE Country = "' +
choice_country + "'"

exchange = pd.read_sql_query(query, db)

choice_exchange = st.sidebar.selectbox("Pick exchange", exchange, index = 1)


# get stock name

query = "SELECT DISTINCT(Name) FROM tkrinfo WHERE Exchange = "' +
choice_exchange + "'"

name = pd.read_sql_query(query, db)

choice_name = st.sidebar.selectbox("Pick the Stock", name)


# get stock tickr

query = "SELECT DISTINCT(Ticker) FROM tkrinfo WHERE Exchange = "' +
choice_exchange + "'" + "and Name = "' + choice_name + "'"

ticker_name = pd.read_sql_query(query, db)

ticker_name = ticker_name.loc[0][0]


hide_github_icon()
```

```
st.write("This is a nice country  ", choice_country)

st.write("It has exchange:,",choice_exchange)

st.write(choice_name)


#get start date

start_date = st.sidebar.date_input("Start Date", value=datetime.date.today() -
datetime.timedelta(days=30))

st.write(start_date)


# get end date

end_date = st.sidebar.date_input("End Date", value=datetime.date.today())

st.write(end_date)

st.write(str(ticker_name))


# get interval

interval = st.sidebar.selectbox("Interval", ['1d', '1wk', '1mo', '3mo'])


#get period

period = st.sidebar.selectbox("Period",['1mo','3mo','6mo','1y','2y','5y','10y','max'],index = 2)


# get stock data

stock = yf.Ticker(str(ticker_name))

#data = stock.history(interval=interval, start=start_date, end=end_date)

data = stock.history(interval=interval, period=period)

if len(data)==0:

    st.write("Unable to retrieve data.This ticker may no longer be in use. Try some other
```

```
stock")
else:
#preprocessing
    data = preprocessing(data,interval)


    if period == '1mo' or period == '3mo':
        horizon = st.sidebar.slider("Forecast horizon",1,15,5)
    else:
        if interval == '1d' or interval == '1wk':
            horizon = st.sidebar.slider("Forecast horizon", 1, 30, 5)
        else:
            horizon = st.sidebar.slider("Forecast horizon", 1, 15, 5)


    model = st.selectbox('Model',['Simple Exponential Smoothing','Holt-Winter Model',
                              'ARIMA Model','Random Forest' ])


    if model=='Simple Exponential Smoothing':
        col1,col2 = st.columns(2)
        with col1:
            alpha_high = st.slider("Alpha_high",0.0,1.0,0.20)
        with col2:
            alpha_low = st.slider("Alpha_low",0.0,1.0,0.25)
        from SES import SES_model
        data_final, smap_low, smap_high, optim_alpha_high, optim_alpha_low =
SES_model(data,horizon,alpha_high,alpha_low)
```

```
#data_final
        st.line_chart(data_final[['High','Forecast_High','Low','Forecast_Low']])
        col1,col2 = st.columns(2)
        with col1:
            st.write("SMAPE for High: {}".format(smap_high))
            st.write("Optimal Alpha for High : {} ".format(optim_alpha_high))
        with col2:
            st.write("SMAPE for Low: {}".format(smap_low))
            st.write("Optimal Alpha for Low: {} ".format(optim_alpha_low))

    elif model == 'Halt Model':
        col1, col2,col3,col4 = st.columns(4)
        with col1:
            level_high = st.slider("Level High", 0.0, 1.0, 0.20)
        with col2:
            trend_high = st.slider("Trend high", 0.0, 1.0, 0.20)
        with col3:
            level_low = st.slider("Level low", 0.0, 1.0, 0.20)
        with col4:
            trend_low = st.slider("Trend Low", 0.0, 1.0, 0.20)
        from SES import Holt_model

data_final,smap_low,smap_high,optim_level_high,optim_level_low,optim_trend_high,
optim_trend_low = Holt_model(data,horizon,level_high,level_low,trend_high,trend_low)
        st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])
        col1, col2 = st.columns(2)
```

```
        with col1:
            st.write("SMAPE for High: {}".format(smap_high))
            st.write("Optimal Level for High : {} ".format(optim_level_high))
            st.write("Optimal Trend for High : {} ".format(optim_trend_high))
        with col2:
            st.write("SMAPE for Low: {}".format(smap_low))
            st.write("Optimal Level for Low: {} ".format(optim_level_low))
            st.write("Optimal Trend for Low: {} ".format(optim_trend_low))
elif model == 'Holt-Winter Model':
        col1, col2 = st.columns(2)
        with col1:
            level_high = st.slider("Level High", 0.0, 1.0, 0.20)
            trend_high = st.slider("Trend high", 0.0, 1.0, 0.20)
            season_high = st.slider("Seasonal high", 0.0, 1.0, 0.20)
        with col2:
            level_low = st.slider("Level low", 0.0, 1.0, 0.20)
            trend_low = st.slider("Trend Low", 0.0, 1.0, 0.20)
            season_low = st.slider("Seasonal Low", 0.0, 1.0, 0.20)
        from SES import Holt_Winter_Model
        data_final, smap_low, smap_high, optim_level_high, optim_level_low,
optim_trend_high, optim_trend_low, optim_season_high, optim_season_low =
Holt_Winter_Model(data,horizon,level_high,level_low,trend_high,trend_low,season_high,
season_low)

        st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])
        col1, col2 = st.columns(2)
        with col1:
```

```
        st.write("SMAPE for High: {}".format(smap_high))
        st.write("Optimal Level for High : {} ".format(optim_level_high))
        st.write("Optimal Trend for High : {} ".format(optim_trend_high))
        st.write("Optimal Seasonal smoothing for high: {}".format(optim_season_high))
    with col2:
        st.write("SMAPE for Low: {}".format(smap_low))
        st.write("Optimal Level for Low: {} ".format(optim_level_low))
        st.write("Optimal Trend for Low: {} ".format(optim_trend_low))
        st.write("Optimal Seasonal smoothing for Low: {}".format(optim_season_low))

elif model == 'Random Forest':
    col1, col2 = st.columns(2)
    with col1:
        p_high = st.slider("Order of High", 1, 30, 1)
    with col2:
        p_low = st.slider("Order of Low", 1, 30, 1)
    from SES import AR_model


    data_final, smap_high, smap_low = AR_model(data,horizon,p_high,p_low)
    st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])
    col1, col2 = st.columns(2)
    with col1:
        st.write("SMAPE of High: {}".format(smap_high))
    with col2:
st.write("SMAPE of Low : {}".format(smap_low))
elif model == 'Moving Average Model':
```

```
        col1, col2 = st.columns(2)
        with col1:
            q_high = st.slider("Order of High", 1, 30, 1)
        with col2:
            q_low = st.slider("Order of Low", 1, 30, 1)
        from SES import AR_model
        data_final, smap_high, smap_low = AR_model(data, horizon, q_high, q_low)
        st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])
        col1, col2 = st.columns(2)
        with col1:
            st.write("SMAPE of High: {}".format(smap_high))
        with col2:
            st.write("SMAPE of Low : {}".format(smap_low))

    elif model == 'ARMA Model':
        col1, col2 = st.columns(2)
        with col1:
            p_high = st.slider("Order of AR High", 1, 30, 1)
            q_high = st.slider("Order of MA High", 1, 30, 1)
        with col2:
            p_low = st.slider("Order of AR Low", 1, 30, 1)
            q_low = st.slider("Order of MA Low", 1, 30, 1)
        from SES import ARMA_model
data_final, smap_high, smap_low=
ARMA_model(data,horizon,p_high,p_low,q_high,q_low)
        st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])
```

```
        col1, col2 = st.columns(2)
        with col1:
            st.write("SMAPE of High: {}".format(smap_high))
        with col2:
            st.write("SMAPE of Low : {}".format(smap_low))
    elif model == 'ARIMA Model':
        col1, col2 = st.columns(2)
        with col1:
            p_high = st.slider("Order of AR High", 1, 30, 1)
            q_high = st.slider("Order of MA High", 1, 30, 1)
            i_high = st.slider("Order of Differencing High" , 0,10,0)
        with col2:
            p_low = st.slider("Order of AR Low", 1, 30, 1)
            q_low = st.slider("Order of MA Low", 1, 30, 1)
            i_low = st.slider("Order of Differencing Low", 0, 10, 0)
        from SES import ARIMA_model
        data_final, smap_high, smap_low =
ARIMA_model(data,horizon,p_high,p_low,q_high,q_low,i_high,i_low)
        st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])
        col1, col2 = st.columns(2)
        with col1:
            st.write("SMAPE of High: {}".format(smap_high))
        with col2:
            st.write("SMAPE of Low : {}".format(smap_low))
    elif model == 'Linear Regression':
        from SES import Auto_Arima
```

```
        st.write("Note: This model may take some time to fit")
        data_final = Auto_Arima(data,horizon)
        st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])


    else:
        from ML_models import forecast
        #data_final = forecast(data,horizon,model)
        data_final, smape_high, smape_low = forecast(data,horizon,model)
        st.line_chart(data_final[['High', 'Forecast_High', 'Low', 'Forecast_Low']])


        col1, col2 = st.columns(2)
        with col1:
            st.write("SMAPE of High: {}".format(smape_high))
        with col2:
            st.write("SMAPE of Low : {}".format(smape_low)),
db.close()
```

5.2.2 SMAPE CALCULATION

The below code contains several functions designed to preprocess stock data and evaluate the accuracy of forecasts for high and low stock prices. The smape function calculates the symmetric mean absolute percentage error (SMAPE) between the actual and predicted values to evaluate the accuracy of the forecast. The preprocessing function adjusts the frequency of the inventory data based on the selected interval (1day, 1 week, 1 month, 3 months) by setting the frequency to weekdays, Mondays, beginning of the month, or beginning of the quarter, and handles missing data accordingly. The process_high and process_low functions forecast high and low stock prices, format the forecast data into a DataFrame, calculate the SMAPE for error assessment, and return the forecast, forecasts, and SMAPE values.

//CODE STARTS

```
import pandas as pd

import datetime

import numpy as np

def smape(y_true, y_pred):

  numerator = np.abs(y_true - y_pred)

  denominator = (np.abs(y_true) + np.abs(y_pred)) / 2

  ratio = numerator / denominator

  return (ratio.mean())


def preprocessing(data,interval):

    if interval=='1d':

        data = data.asfreq('B') #set frequency as Business Day

        data.ffill(inplace=True)


    elif interval == '1wk':

        data.dropna(inplace = True) # weekly/monthly data has days on which dividends are
paid, while all other values as NA

        if int(str((data.index[-1]-data.index[-2]))[0])<7: #if the last data point is of today and
today is not monday

            data = data.iloc[:-1,]

        data = data.asfreq('W-MON') #week  monday

    elif interval == '1mo':

        data.dropna(inplace=True)

        if data.index[-1].day!=1: #if the last data point is of today not the first of month

            data = data.iloc[:-1,]

data = data.asfreq('MS') #month start
```

```
    elif interval == '3mo':
        data.dropna(inplace=True)
        if data.index[-2].month-data.index[-1].month <3:  # if the last data point is of today
not the first of month
            data = data.iloc[:-1,]
        freq = 'QS-'+data.index[-1].month_name()[0:3].upper()
        data = data.asfreq(freq)  # quarter start
    return data
#def seasonal(interval,period):
 #   if interval== '1d':
  #      if period=='1mo' or period=='3mo' or period=='6mo':
    #      season = 7
def process_high(data,res_high ,fore_high):
    fore_high = fore_high.to_frame()
    fore_high.columns = ['Forecast_High']
    pred_high = res_high.predict(start=data.index[0], end=data.index[-1])
    smap_high = round(smape(data['High'], pred_high), 3)
    return [fore_high,pred_high,smap_high]
def process_low(data,res_low ,fore_low):
    fore_low = fore_low.to_frame()
    fore_low.columns = ['Forecast_Low']
    pred_low = res_low.predict(start=data.index[0], end=data.index[-1])
    smap_low = round(smape(data['Low'], pred_low), 3)
    return [fore_low,pred_low,smap_low]
```

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Performance Evaluation:

In this case, the ARMA model used earns its acclaim from the high and low stock prices of Bharat Heavy Electricals Ltd. (BHEL.BO) predicted over the specified period. Fig. 6.1.1 specifies a period of 6 months with an interval of 1 day.

1. Model and Parameters:
   - Designated as the ARMA model for forecasting high and low prices of BHEL.
   - The interface allows adjusting the Order of AR and Order of MA parameters, separately for high and low price predictions. The sliders in Fig. 6.1.1 indicates that all orders are set to 1, which is the minimal complexity setting for the AR and MA components. One can turn the settings from one to 30, to tune the model's sensitivity to past values and moving average smoothing.
2. Forecast Visualization in Fig. 6.1.2 and Fig. 6.1.3:
   - A line chart shows both actual and forecasted values of high-low prices:
   - The red line depicts actual high and low stock prices over the periods.
   - The blue line shows the forecasted price for high, while the light blue line shows the forecasted price for low.

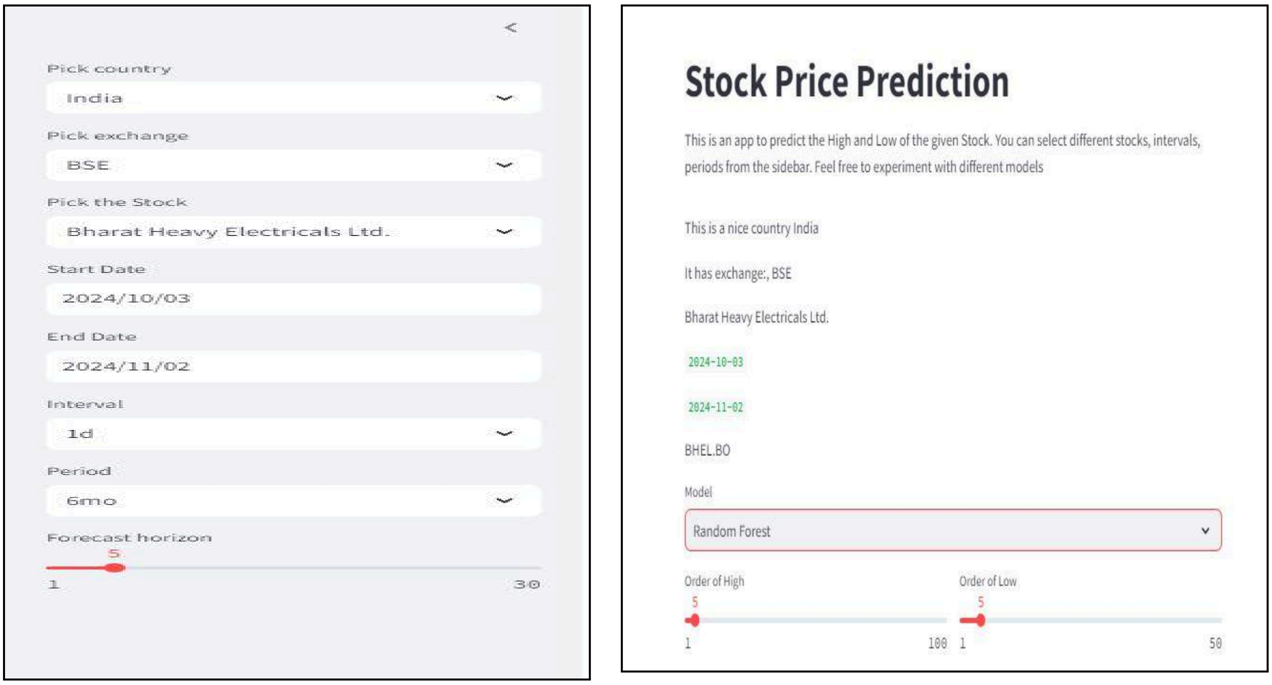

Fig. 6.1.1 Model Parameters

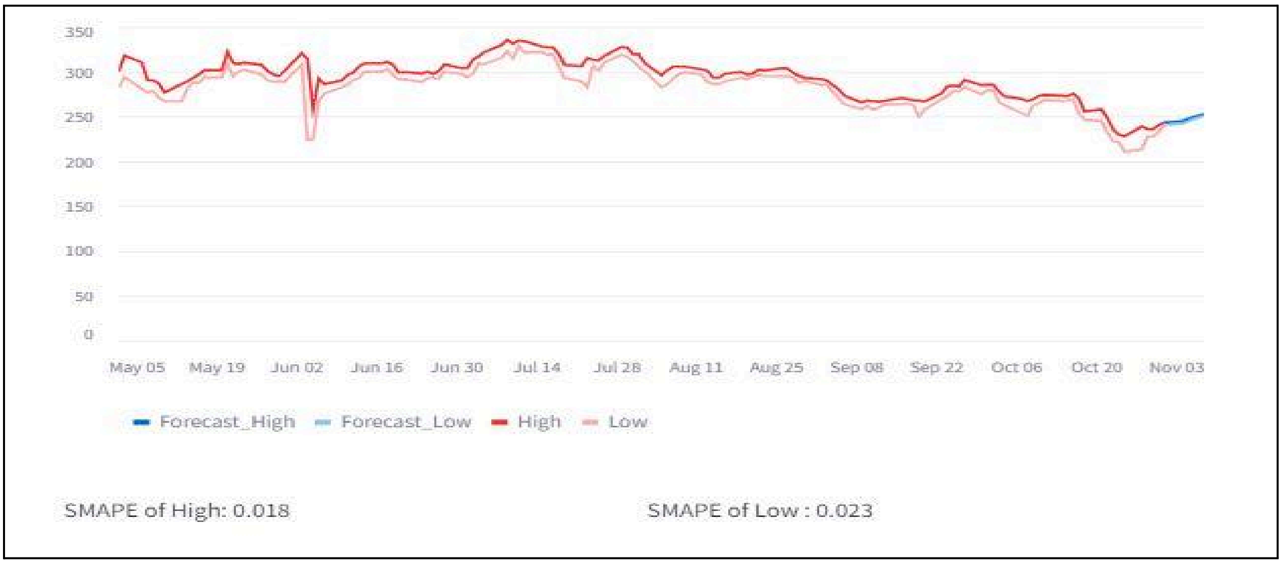

Fig. 6.1.2 Forecast Visualization

Fig. 6.1.3 Forecast SMAPE

6.2 Comparison of the models (eg. ARIMA, Random Forest):

```
In [2]: import matplotlib.pyplot as plt


        models = ["Simple Exp Smoothing", "ARIMA", "Random Forest"]
        smape_high = [0.028, 0.018, 0.016]  # SMAPE for High values
        smape_low = [0.03, 0.021, 0.019]    # SMAPE for Low values

        x = range(len(models))  # Positions for the bars
        width = 0.35  # Width of the bars

        plt.figure(figsize=(8, 5))


        bars1 = plt.bar([pos - width/2 for pos in x], smape_high, width, label='High', color='skyblue')
        bars2 = plt.bar([pos + width/2 for pos in x], smape_low, width, label='Low', color='lightgreen')


        for bars in [bars1, bars2]:
            for bar in bars:
                plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height() + 0.001,
                         f"{bar.get_height():.3f}", ha='center', fontsize=10)


        plt.ylabel("SMAPE Value", fontsize=12)
        plt.title("SMAPE Comparison: High vs Low Predictions", fontsize=14)
        plt.xticks(x, models, fontsize=12)
        plt.legend(title="Prediction Type", fontsize=10)

        # Display the plot
        plt.show()
```

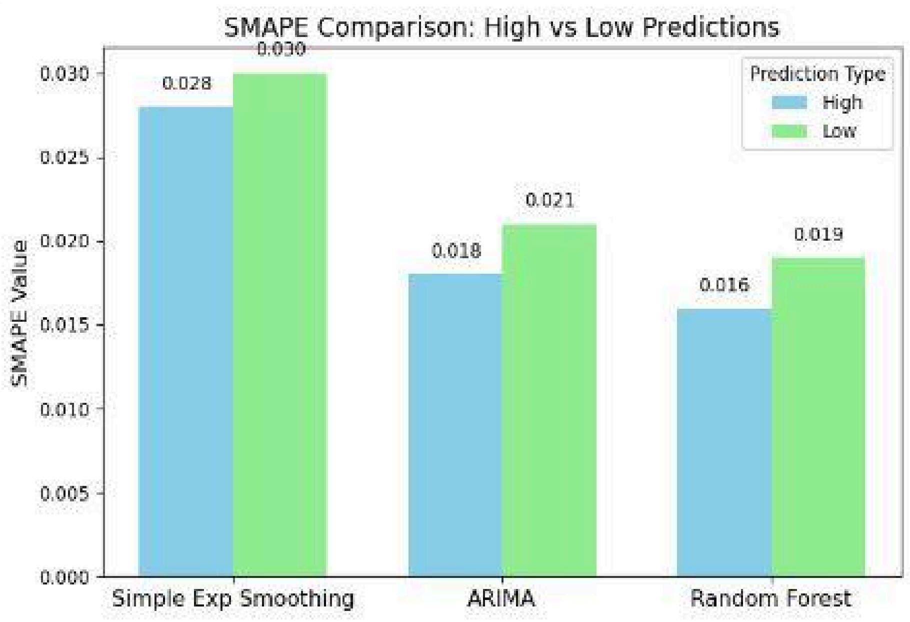

Fig. 6.2.1 Comparison graph

6.3 Discussion over results:

Comparison of forecasting accuracy among the ARIMA, Simple Exponential Smoothing (SES), Holt-Winters models, and the Random Forest model are done using Fig 6.2.1:

1. **ARIMA vs. Simple Exponential Smoothing (SES)**:

- The performance of the ARIMA model was compared to that of the SES model, a simple approach which is predominantly applied to time-series data without trends and seasonality.

- **Accuracy Improvements**:
  - In respect of the final analysis, high predictions evidenced that **ARIMA** exhibited a **35.71% gain** against SES. This indicates that the fact that ARIMA possesses the ability to ascertain a couple of complex structures and trends in historical data enhances its predicting productivity in comparison to SES, which might be perceived as a more simplistic model of forecasting.
  - ARIMA gave **30% better accuracy** than SES under low-prediction circumstances, emphasizing how much efficient ARIMA, substantively, is when handling low-to-high stock prices

2. **ARIMA vs. Holt-Winters Model**:

- The Holt-Winters model that accounts for the seasonality and trend behaviors of time series data was then compared with ARIMA.
- **Accuracy Improvements**:
  - High prediction indicated that **ARIMA** had performed better than Holt-Winters by about **40%**. This height difference indicates that Holt-Winters might be a little weaker than the usage of ARIMA in stock price dynamic modeling.
  - FLow prediction demonstrated that **ARIMA** gave a **43.24% lead on Holt-Winters**, which further showed that the modes of ARIMA value grouping provide solutions that show further values of past data, stabilization of variance by integrating difference modeling, and generate higher accuracy with observations exhibiting lower values particularly in stock prices.

3. **Comparison between ARIMA and Random Forest**:

- Once ARIMA had proven to be a good candidate for stock price projection, a final analysis was undertaken against another forecasting estimator, Random Forest, which utilized an ensemble learning framework to decipher the non-linear relationships in the data.
  - The analyses suggested that **Random Forest was the best-performing model**, since it was found to be **11.11% more accurate than ARIMA**. Random Forest, thus, is better at detecting complex patterns that might be missed by traditional linear models such as ARIMA.

- Random Forest is fed on assembly shapes through embedding numerous choice trees, each modeled on various data parcels, which boosts its predictive performance through robustness in different market scenarios. Therefore, the generalization ability of Direct for a wide variation of datasets is reaffirmed as the most effective model chosen in the analysis.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Summary of Finding

This project develops an artificial intelligence (AI)-based personal finance advisor to assist investors in making decisions regarding stock markets made possible through cloud computing. The integration of AI models such as ARIMA and Random Forest, with cloud services leads to efficient performance, scalability and real-time decisions. The system affords ease of access to users in identifying personalized financial advice, while addressing some of the major concerns related to data security, algorithmic bias, and transparency. In this sense, this study highlights the ability of AI to change the landscape of stock market investment years down the road, providing reliable and affordable investment advice.

The examination illustrates how different forecasting systems-such as SES, ARIMA and Random Forest-have their respective advantages as well as disadvantages during stock trend demands. This is because Random Forest outperformed ARIMA in terms of predicting stock prices, mostly on highly volatile stocks. However, with the superior and simple-to-interpret form of the ARIMA model, it made itself quite useful when facing movement of more stable markets. Further, cloud computing allows the application to be scalable and thus enables it to address huge datasets while providing predictions perpetually

This financial advisor represents a giant leap forward towards democratizing the financial decision-making tool; thus, providing a greater say in the informed stock markets to a wider audience. The new system does combine manicured advanced AI models with the cloud, overcoming some challenges facing small investors, such as limited access to solid financial advice and the complexity of sifting through enormous amounts of market data to find value.

6.2 Impact of the project

The impact of our project is significant in improving users' financial decision-making. Using machine learning models and AI-driven insights, the project provides personalized financial advice that enables users to make informed investment decisions across stocks foreign exchange markets. Integration with real-time data sources such as YFinance ensures that forecasts remain relevant, while Finance ChatBot offers immediate support and makes financial insights accessible to users. Our project gives individuals a comprehensive approach to financial planning with artificial intelligence, promoting greater financial literacy and investment confidence.

6.3 Future Work

This research data has future intentions of providing the AI-based prediction models with working improvements to enhance trading decisions for stronger accuracy minus biases in financial forecasts. That initial objective is likely to be improvement to robustness of the algorithmic model, aiding better adaptivity to an extended variety of situations or market conditions and still giving stock predictions with good accuracy. A wider scope of datasets encompassing the world financial markets will need to be expanded, complemented by macroeconomic indicators and alternative investment paths also, allowing increased predictions by the system and thus of providing deeper investment advice.

An important aspect would entail research towards assurance of customer data privacy and data security with the second set of changes. As financial data is highly sensitive, maintaining user trust through secure cloud services and safeguarding personal information will be vital for the continued success of the AI-based advisor. Exploring more sophisticated encryption methods and ensuring compliance with evolving data protection regulations will be critical steps in advancing the system’s reliability and user confidence.

Future work will be concentrated on continuous optimization of the model to improve fairness and reduce algorithmic bias, which is critical for developing a system that will serve a diverse range of users. Addressing these challenges and integrating new technologies as they emerge, the AI-based financial advisor will be better equipped to deal with the complexities of the changing fintech landscape and empower users with better investment tools and guidance for long-term financial success.

REFERENCES

[1] Hui Zhu,Olli Vigren, Inga-Lill Söderberg, Journal of Business Research, 4 citations, "Implementing artificial intelligence empowered financial advisory services: A literature review and critical research agenda 2024", DOI: https://doi.org/10.1016/j.jbusres.2023.114494.

[2] Priyanka R Rao, K.S. Lakshmi, "A Review on the role of Robo-advisory service in transforming Personal Finance in the Digital- Era 2024", DOI: https://doi.org/10.52783/jier.v4i2.1077.

[3] Yuning Liu, Junliang Wang, Academic Journal of Science and Technology, "Analysis of Financial Market Using Generative Artificial Intelligence, 2024", DOI: https://doi.org/10.54097/y17mrj84.

[4] Zengyi Huang, Chang Che, Haotian Zheng, Chen Li, "Applications of generative AI-based financial robot advisors as investment consultants", Academic Journal of Science and Technology, 10 citations May 2024, Applied and Computational Engineering 67(1):28-33, DOI: 10.54254/2755-2721/67/2024MA0057.

[5] Sergiu-Alexandru Ionescu, Vlad Diaconita , "Transforming Financial Decision-Making: The Interplay of AI, Cloud Computing and Advanced Data Management Technologies", International Journal of Computers Communications & Control, 8 citations, DOI: https://doi.org/10.15837/ijccc.2023.6.5735.

[6] Marko Ranković, Elena Gurgu, Oliva M. D. Martins, Milan Vukasović, "Artificial Intelligence and the Evolution of Finance: Opportunities, Challenges and Ethical Considerations", January 2023, EdTech Journal 3(1):20-23, DOI:10.18485/edtech.2023.3.1.2.

[7] Muhammad Anshari, Mohammad Nabil, Masairol Masri, "Digital Twin: Financial Technology's Next Frontier of Robo-Advisor", April 2022, Journal of Risk and Financial Management 15(4) DOI:10.3390/jrfm15040163.

[8] Janin Karoli Hentzen, Arvid Hoffmann, Rebecca Dolan, Erol Pala International Journal of Bank Marketing ISSN: 0265-2323, "Artificial intelligence in customer-facing financial services: a systematic literature review and agenda for future research", 2021.

[9] Shahnawaz Khan & Mustafa Raza Rabbani, 2021. "Artificial Intelligence and NLP

-Based Chatbot for Islamic Banking and Finance," International Journal of Information Retrieval Research (IJIRR), IGI Global, vol. 11(3), pages 65-77, July.

[10] Ankita Bhatia, Arti Chandani, Rizwana Atiq, Dr Mita Mehta, "Artificial intelligence in financial services: a qualitative research to discover robo-advisory services", September 2021, Qualitative Research in Financial Markets ahead-of-print(ahead-of-print), DOI:10.1108/QRFM-10-2020-0199.

[11] Manchuna Shanmuganathan, "Behavioral finance in an era of artificial intelligence: Longitudinal case study of robo-advisors in investment decisions", 2021, DOI:https://doi.org/10.1016/j.jbef.2020.100297.

[12] Alison Lui, George William Lamb, "Artificial intelligence and augmented intelligence collaboration: Regaining trust and confidence in the financial sector", June 2018, Information & Communications Law, 27(3):1-17, DOI:10.1080/13600834.2018.1488659.

[13] Mugdha Shailendra Kulkarni , Kanchan Patil, "Artificial Intelligence in Financial Services: Customer Chatbot Advisor Adoption", November 2019, International Journal of Innovative Technology and Exploring Engineering, 9(1), DOI: 10.35940/ijitee.A4928.119119.

[14] Kraiwanit, T., Jangjarat, K., & Atcharanuwat, J. (2022). "The acceptance of financial robo-advisors among investors: The emerging market study. Journal of Governance and Regulation", 11(2, special issue), 332–339. 10.22495/jgrv11i2siart12.

[15] Belanche, D., Casal´o, L. V., & Flavi´an, C. (2019). "Artificial Intelligence in FinTech: Understanding robo-advisors adoption among customers. Industrial Management & Data Systems", 119(7), 1411–1430. https://doi.org/10.1108/IMDS-08-2018-0368.

[16] Gomber, P., Kauffman, R. J., Parker, C., & Weber, B. W. (2018). "On the Fintech Revolution: Interpreting the Forces of Innovation, Disruption, and Transformation in Financial Services". Journal of Management Information Systems, 35(1), 220–265. https://doi.org/10.1080/07421222.2018.1440766.

[17] Al-Gasawneh, J. A., AL-Hawamleh, A. M., Alorfi, A., & Al-Rawashdeh, G. (2022). "Moderating the role of the perceived security and endorsement on the relationship between per-ceived risk and intention to use the artificial intelligence in financial services." International Journal of Data and Network Science, 6(3), 743–752. 10.5267/j. ijdns.2022.3.007.

[18] Dhashanamoorthi, Balaji. "Artificial Intelligence in combating cyber threats in Banking and Financial services." International Journal of Science and Research Archive 4.1 (2021): 210-216.

[19] Baek, Yujin, and Ha Young Kim. "ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module." Expert Systems with Applications 113 (2018): 457-480.

[20] Galeshchuk, Svitlana, and Sumitra Mukherjee. "Deep learning for predictions in emerging currency markets." International conference on agents and artificial intelligence. Vol. 2. SCITEPRESS, 2017.

[21] Pokhrel, N.R.; Dahal, K.R.; Rimal, R.; Bhandari, H.N.; Khatri, R.K.C.; Rimal, B.; Hahn, W.E. Predicting NEPSE Index Price Using Deep Learning Models. Machine Learning with Applications 2022, 9, 100385. https://doi.org/10.1016/j.mlwa.2022.100385

[22] Flesch, T.; Juechems, K.; Dumbalska, T.; Saxe, A.; Summerfield, C. Orthogonal Representations for Robust Context-Dependent Task Performance in Brains and Neural Networks. Neuron 2022, 110, 1258-1270.e11. https://doi.org/10.1016/j.neuron.2022.01.005.

[23] Baek, Y.; Kim, H.Y. ModAugNet: A New Forecasting Framework for Stock Market Index Value with an Overfitting Prevention LSTM Module and a Prediction LSTM Module. Expert Systems with Applications 2018, 113, 457–480. https://doi.org/10.1016/j.eswa.2018.07.019.

[24] Galeshchuk, S.; Mukherjee, S. Deep Networks for Predicting Direction of Change in Foreign Exchange Rates. Intell Sys Acc Fin Mgmt 2017, 24, 100–110. https://doi.org/10.1002/isaf.1404.

[25] Althelaya, K.A.; El-Alfy, E.-S.M.; Mohammed, S. Evaluation of Bidirectional LSTM for Short-and Long- Term Stock Market Prediction. In Proceedings of the 2018 9th International Conference on Information and Communication Systems (ICICS); IEEE: Irbid, April 2018; pp. 151–156.

[26] Lu, W.; Li, J.; Li, Y.; Sun, A.; Wang, J. A CNN-LSTM-Based Model to Forecast Stock Prices. Complexity 2020, 1–10. https://doi.org/10.1155/2020/6622927.

APPENDIX A

RESEARCH PAPER DETAILS

We submitted our research paper for publication at the **International Conference on Emerging Trend in Intelligent Computing Techniques**. We have selected the **Track Name: ICETICT2024**. Proof of submission is attached in figure B.1.

26/10/2024, 18:59 SRM Institute of Science and Technology Mail - International Conference On Emerging Trend in Intelligent Computing Techniques ...

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY

Vijayalakshmi V <vijayalv@srmist.edu.in>

International Conference On Emerging Trend in Intelligent Computing Techniques : Submission (158) has been created.

1 message

Microsoft CMT <email@msr-cmt.org> Sat, Oct 26, 2024 at 4:17 PM
Reply-To: Microsoft CMT - Do Not Reply <noreply@msr-cmt.org>
To: vijayalv@srmist.edu.in

Hello,

The following submission has been created.

Track Name: ICETICT2024

Paper ID: 158

Paper Title: AI-based Financial Advisor for Stock Market Predictions using Cloud

Abstract:
The research paper intends to improve the existing personal finance advisors leveraging AI techniques for stock market investments by solving a few critical problems noted in the current systems. The main objectives consist of improving
the flexibility and the creativeness offered by the algorithms,
enhancing data security, data coupling privacy concerns, and
fully eliminating algorithmic prejudices to fairer and unbiased
recommendations. Further, the specific orientation of the research is also connected with strengthening the explainability
of AI models and their orientation towards regulatory standards
aimed at increasing investors` confidence, as well as enhancing
the predictive accuracy of the models by incorporating behavioral finance. With these improvements in place, the research is aiming to deliver a more trustworthy, safe, and consumer oriented AI powered financial advisory platform.

Created on: Sat, 26 Oct 2024 10:46:37 GMT

Last Modified: Sat, 26 Oct 2024 10:46:37 GMT

Authors:
- vijayalv@srmist.edu.in (Primary)
- rd7384@srmist.edu.in
- pk0562@srmist.edu.in
- as6252@srmist.edu.in

Secondary Subject Areas: Not Entered

Submission Files:
Stock Market.pdf (346 Kb, Sat, 26 Oct 2024 10:46:30 GMT)

Submission Questions Response: Not Entered

Thanks,
CMT team.

A.1: Submission Notification

The Research Paper cover page has been attached below.

AI-based Financial Advisor for Stock Market Predictions using Cloud

Rachit Sood
Department of Networking and Communication
School of computing
SRM Institute of Science and Technology
Chennai, Tamil Nadu
rd7384@srmist.edu.in

Papai Mondal
Department of Networking and Communication
School of computing
SRM Institute of Science and Technology
Chennai, Tamil Nadu
pk0562@srmist.edu.in

Asmi Shrivastava
Department of Networking and Communication
School of computing
SRM Institute of Science and Technology
Chennai, Tamil Nadu
as6252@srmist.edu.in

Vijayalakshmi V
Department of Networking and Communications
School of Computing
SRM Institute of Science and Technology
Chennai,Tamil Nadu
vijayalv@srmist.edu.in

Abstract— The research paper intends to improve the existing personal finance advisors leveraging AI techniques for stock market investments by solving a few critical problems noted in the current systems. The main objectives consist of improving the flexibility and the creativeness offered by the algorithms, enhancing data security, data coupling privacy concerns, and fully eliminating algorithmic prejudices to fairer and unbiased recommendations. Further, the specific orientation of the research is also connected with strengthening the explainability of AI models and their orientation towards regulatory standards aimed at increasing investors' confidence, as well as enhancing the predictive accuracy of the models by incorporating behavioral finance. With these improvements in place, the research is aiming to deliver a more trustworthy, safe, and consumer oriented AI powered financial advisory platform.

Keywords— *Artificial Intelligence, Security, LSTM, Cloud, Financial Decision.*

I. Introduction

Considering an ever-growing volume of information about income, expenses, taxes, loans and investments one must manage his or her personal finances in today's world, one must realize how crucial it is to come up with a good financial plan. Short-term as well as long-term financial goals are necessary for success, but these markets are also unpredictable. Of course, financial advisors have been an important part of the clients' journey in such cases, advising on financial behavior or trends that needed to be followed. On the other hand, both are often expensive and involve the risk of having to keep up with market rates which might be prone to volatility all the time. The introduction of technology however has changed the financial industry tremendously such that reliance on consultants is not as necessary. Applications and online tools for AI (Artificial Intelligence), which constitute a form of machine learning (ML), have been created for the automatic management of personal finances as well as recommendations on investment.

Nonetheless, there are still issues such as safeguarding data, eliminating algorithmic bias, and making the recommendations clear and easy to explain. Furthermore, in this context, as demand for such solutions continues to grow, cloud computing has emerged as an essential platform for AI-based," robo-brokerage" systems. Cloud computing resources make it possible to pursue round-the-clock monitoring of financial markets and deploy AI algorithms in a reliable and scalable manner. In respect to this research paper seeks to construct AI based personal financial advisors hosted on cloud resources. The main goals towards which the system is directed include provision of accurate financial counsel by addressing a number of challenges such as data security, bias and transparency using the scalable features of cloud infrastructure. Thus, as part of this study, we strive to further develop the theme of AI in financial services and show new opportunities for the market generated by cloud technologies for providing affordable and secure financial consulting.

II. Related Work

Hui Zhu et al. [1] wrote a literature review and a stimulating perspective on how AI-enabled financial advisory services should be developed, stressing the disruptive potential of these services on human advisors in the retail investment market. Priyanka R. Rao et al. [2] addressed the role of

APPENDIX B

PLAGIARISM REPORT

6% Overall Similarity

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

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
mdpi-res.com		2%
2	Internet	
fastercapital.com		1%
3	Student papers	
Manchester Metropolitan University		1%
4	Internet	
journals.fe.up.pt		1%
5	Internet	
www.nepjol.info		1%
6	Student papers	
University of Sydney		0%
7	Internet	
doi.fil.bg.ac.rs		0%
8	Internet	
www.coursehero.com		0%

PLAGIARISM REPORT

Format - I

SRM Institute of Science and Technology, Kattankulathur **(Deemed to be University u/ s 3 of UGC Act, 1956)**		
Office of Controller of Examinations		
REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES **(To be attached in the dissertation/ project report)**		
1	Name of the Candidate **(IN BLOCK LETTERS)**	RACHIT SOOD, PAPAI MONDAL, ASMI SHRIVASTAVA
2	Address of the Candidate	SRM Institute of Science and Technology, Kattankulathur
3	Registration Number	RA2111028010110, RA2111028010116, RA2111028010119
4	Date of Birth	02/10/2003, 28/03/2004, 07/12/2004
5	Department	Networking and Communications
6	Faculty	Engineering and Technology, School of Computing
7	Title of the Dissertation/Project	AI-based Financial Advisor for Stock Market Predictions with Cloud Integration
8	Whether the above project /dissertation is done by	Individual or group : (Strike whichever is not applicable) **a)** Rachit Sood b) Papai Mondal c) Asmi Shrivastava

9	Name and address of the Supervisor / Guide	Mrs. Vijayalakshmi V **Mail ID:** vijayalv@srmist.edu.in **Mobile Number:** 8678921167
10	Name and address of Co-Supervisor / Co-Guide (if any)	**Mail ID: NA** **Mobile Number: NA**

11	Software Used	Turnitin		
12	Date of Verification			
13	**Plagiarism Details: (to attach the final report from the software)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self-citation)**	**% of plagiarismafter excludingQuotes, Bibliography, etc.,**
1	INTRODUCTION			
2	LITERATURE REVIEW			
3	SYSTEM ARCHITECTURE			
4	METHODOLOGY			
5	CODING AND TESTING			
6	RESULT AND DISCUSSION			
7	CONCLUSION AND FUTURE SCOPE			
8	OVERALL			
	Abstract			
	Appendices			

I/ We declare that the above information has been verified and found true to the best of my/ our knowledge.

Signature of the Candidate	**Name & Signature of the Staff (Who uses the plagiarism check software)**
Name & Signature of the Supervisor/ Guide	**Name & Signature of the Co-Supervisor/Co-Guide**

Name & Signature of the HOD