A
REPORT
ON

Smart Stock Insight System

By

Christina George	**2022A7PS0015U**
Laksha Malik	**2022A7PS0017U**
Diya Vinu	**2022A7PS0023U**
Afra Aadilshah Patel	**2022A4PS0076U**

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI (Dubai Campus)

(2024-25)

Table of Contents

1. Abstract

A wide range of issues affects the stock market, making it tough to make exact predictions, but doing so can be very useful. The study investigates the techniques of predicting and modeling stock price movements with machine learning. Our approach is to analyze old stock information to discover important trends that guide the market. The predictive capabilities of LSTM neural networks, Random Forests, Decision Trees and Linear Regression are analyzed and assessed. The findings show how effective the model is and what AI can do to guide investors in making wise decisions.

2. Introduction

a. Introduction

The stock market plays a major role in the global economy and lets us know how healthy finances are and what investors think. Yet, making predictions about property prices is not easy since the process is not simple and is very changeable. Pricing decisions are usually affected by company results, economic trends, political events and the feelings and thoughts of buyers. There is growing interest among investors to explore machine learning (ML) applications for studying history and predicting likely price changes because the subject is improving so fast. Increasing the accuracy of stock market predictions is the main goal of this paper, built using several machine learning algorithms. To do this, data from the past has to be gathered, suitable features chosen, different algorithms trained and their results examined. As a result, we study the strengths and weaknesses of several financial models and work to anticipate upcoming changes in prices. The goal in the end is to design a system so that investors can use knowledge to guide their decisions.

b. Motivation

It is difficult to correctly forecast stocks and their prices, since information in the market changes all the time and is not always consistent. Even so, helping investors choose the right places to invest is important. Nonlinearity in financial markets means mathematical approaches, as used in traditional statistics, are insufficient to model them. So, because of advanced technology and a growing base of old data, machine learning has become really useful for studying trends in stock prices. Our goal is to use today's top machine learning techniques to beat standard methods because they deliver more accuracy, adapt to different needs and offer predictions on the spot. Our goal is to help investors make smart choices, avoid big financial risks and gain access to advanced analytical resources by producing more accurate models that give us insights into future market behavior.

c. Novelty

Though machine learning for stock forecasting has been dealt with in most research papers, this work documents a comparison method that compares the conventional and deep learning models—the LSTM neural network, Decision Tree, Random Forest, and Linear Regression—on a shared dataset and features. The innovation is in combining temporal dependencies (using LSTM) and feature importance understanding (from tree models) to provide a hybrid analysis view, together with comparing such methods under similar conditions

Furthermore, the research also explores popularly overlooked interpretability, model stability, and insightful implementation findings. This rigorous approach provides substantial contributions to academic research as well as the development of useful investment strategies by filling the gap between model performance on raw data and real-world use.

3. Literature Review

Forecasting stock market changes is an ever-demanding task due to the nonlinearity, complexity, and susceptibility of the market to various economic, psychological, and political factors. As noted, investors and financial institutions use accurate prediction in deciding on investment. Consequently, extensive work is being done on mathematical, statistical, and artificial intelligence techniques to help forecast behavior and trends within the stock market. In recent years, ML models have come to the forefront in this regard (S. Acharjee)[1].

In the early stages, Linear regression, ARIMA, and exponential smoothing were the mathematical and statistical models that had been used extensively in early stock market prediction techniques (Y. Yang)[3]. As a statistical and supervised learning method, linear regression uses a line of best fit to existing data to explain the relationship between independent variables and the dependent variable. It is a widely used baseline in financial forecasting because it is easy to implement.

But as the stock market is a highly dynamic, nonlinear, non-stationary system, it becomes extremely challenging for the conventional models to identify its changes. To capture the volatility and nonlinearity of the market more effectively, AI AI-based models are extensively used. The application of AI-based methods, which are more effective in handling such complexities, has been prompted by this limitation. Models that incorporate a blend of neural networks and timeseries analysis such as LSTM, SVM, etc. In the majority of studies, these models have outperformed conventional statistical methods and are also more effective in learning intricate patterns from large datasets. The most popular and efficient AI tools for predicting stock prices, for example, are SVM, LSTM, and ANN based on (H. Le)[2].

Apart from these, the rise of hybrid techniques is also discussed in literature. This technique mingles different models to harness the advantage of both. It may be an amalgamation of non-linear models with linear regression, or mathematical and statistical models with ML-based models. This fusion can increase the prediction accuracy. For instance, in a recent research(Y. Dai) [5], LSTM-CNN hybrid model was studied. It produced short-term and long-term patterns more precisely than the traditional models. Compared to ARIMA, Random Forest, and SVM, the hybrid model suggested in this work was superior to ARIMA and SVM. However, it had the potential to be used in the time series model context. In addition, beyond model tuning, predictive efficiency is improved by feature engineering, data preprocessing, and the use of big-time data.

For instance, studies have shown that the use of macroeconomic signals and technical signals in a time series modelling based on them increases accuracy(W. Bao)[6]. This model utilized a combination of LSTM and stacked autoencoders. Technical indicators such as RSI (Relative Strength Index) marking the range and volatility of the variation of recent data, MACD(Moving Average Convergence Divergence), marking the interaction of 2 moving averages, and Bollinger Band,s marking deviance of data points from the mean value were used in this study. These signals were also combined with the original stock price data to give the model a more specified and more detailed data set. It is worth mentioning that stock price also depends on sentiment data.

It can be collected through news or social media and has some impact on the stock trends. A study analyzed the relationship between public opinion and the price of the DJIA stock (G.S. Atsalakis)[7]. In the study, the public mood data was acquired from twitter. It used tools like OpinionFinder and GPOMS to identify the mood of about 2 million tweets in a time frame. It found that some mood states like calmness strongly predicted DJIA movements several days ahead. This indicates that stock trends and prices are affected by public mood. Summarizing the literature review, AI holds promise but challenges to be overcome just the same.

Model structure, parameter optimization, feature selection, data quality, and performance measures all strongly impact the extent to which prediction models do the job. Model performance is most often benchmarked by measures such as accuracy, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Furthermore, even though AI models perform better than conventional techniques for most cases, there is no single technique that is best at all times; outputs depend on the dataset, market conditions, and forecasting horizon. This has prompted the demands for increased transparency in the way results are reported and more uniform evaluation procedures.

4. Methodology

Here, the method used to develop the Smart Stock Insight System is described. I conducted literature research, got the data, cleaned it for clarity, built a linear regression model from it, examined the outcomes and joined it into a complete application to show our results.

a. How to collect and prepare the data is important.

The data used for this application comes from Alpha Vantage API calls. You can use this service to access information about yesterday's and current stock markets. We ask the API in our application to retrieve the stock prices for the items we provided. When using this tool, you choose a stock symbol and also enter the time period you need data for. The API then sends you time series data in JSON format. The team then uses the pandas library to turn this data into a DataFrame which can be processed further. Removing null values, putting the data in order based on time and changing data types were all tasks I performed. A further part of data preparation involved adding technical indicators.

To keep the data updated and reachable even if the server fails, the system uses an additional fall back mechanism when the API does not work. Most people use another proxy due to the restriction on requests allowed in the Alpha Vantage free version.
The process starts by using historical data from a CSV file found in the local folder. If the live information is missing, the application can still run.

b. Technical Indicators

Based on historical and real-time data, we have calculated the most important technical indicator, moving averages (MA). A moving average was applied to the data to obtain a 20-day and a 50-day average; both numbers were included in the dataset used by the model.
It was decided to base the signal on these technical indicators because when the 20-day moving average was bigger than the 50-day moving average, there was a bigger chance of an upward trend in the coming days, helping the predictor.

c. Prediction Model

The linear regression model is applied in our system to project the stock prices for the following 3 days. We selected linear regression because it makes a good first attempt at prediction. The linear structure of the model used made it unable to analyse the frequently changing and irregular patterns found in stock data. By using our technical indicators, we turn this into a hybrid model that becomes more accurate in its predictions.
The estimate relies on the coming three days' stock prices and the 20 and 50 day moving averages.

d. Visualisation

Charts displaying candlesticks are shown for the application output and the 20 and 50 day moving averages are indicated. Python libraries are used to create the visualisations.
You'll find a candlestick chart in the plot, showing the open, high, low and close prices for each day. Following this, moving averages are added to the above charts. As a result, the graph can be easily understood because it shows the short-term direction. Then, the model's prediction can be confirmed by checking the moving averages graphs.

e. Limitations

Discussions have pointed out that linear models are not adequate for stock market prediction. Therefore — we have based our approach on a hybrid model with technical indicators. Still, it doesn't really explain the strange and unpredictable movements in the market. Thus, the model cannot handle sudden changes in price or strong reversals in a trend.
To deal with the problems previously mentioned, you can move forward to use time series models as well as neural networks such as the LSTM, SVM and so on. In some cases, the risk level linked to the prediction can be explained as well. In other words, it would deal with the differences between observed and expected values.

<u>5. Code Implementation</u>

a. Dataset Explanation

GOOGL

Date	Open	High	Low	Close	Volume	MA20
2025-04-04	2703.53	2701.31	2697.66	2699.19	9215563	
2025-04-05	2700.8	2701.76	2698.28	2702.44	7501992	
2025-04-06	2701.96	2699.22	2697.37	2700.42	4217620	
2025-04-07	2704.48	2697.04	2695.55	2701.95	3143926	
2025-04-08	2703.74	2700.3	2699.35	2700.71	6567909	
2025-04-09	2698.05	2701.31	2698.2	2701.41	5341373	
2025-04-10	2701.9	2703.46	2695.74	2700.02	5031087	
2025-04-11	2699.7	2703.4	2699.93	2703.57	3160155	
2025-04-12	2699.79	2700.23	2697.19	2700.25	8332661	
2025-04-13	2700.82	2700.4	2699.1	2700.8	2511731	
2025-04-14	2700.29	2698.9	2700.46	2703.77	8095470	
2025-04-15	2702.91	2698.16	2699.26	2697.3	6327797	
2025-04-16	2701.52	2697.59	2701.28	2697.46	5008483	
2025-04-17	2700.24	2704.9	2696.53	2701.94	9471335	
2025-04-18	2700.89	2699.98	2699.8	2697.65	2793923	
2025-04-19	2700.67	2700.12	2697.63	2703.89	5307721	
2025-04-20	2702.99	2698.49	2697.26	2699.17	2437066	
2025-04-21	2699.59	2702.55	2697.84	2698.51	2175067	
2025-04-22	2700.63	2697.77	2698.38	2703.85	3590119	
2025-04-23	2698.29	2700.57	2699.11	2702.96	7465442	2700.86
2025-04-24	2694.89	2699.21	2696.67	2703.74	6444696	2701.09
2025-04-25	2701.31	2701.77	2700.8	2701.81	4340520	2701.06
2025-04-26	2701.73	2699.98	2699.93	2698.28	4130827	2700.95
2025-04-27	2698.52	2698.64	2695.93	2703.82	6447877	2701.05

The dataset used in this project represents historical stock data for a company (e.g., Google), with each row corresponding to one trading day. The columns include:

- **Date**: The trading day (YYYY-MM-DD format).
- **Open**: The stock's opening price for the day.
- **High** and **Low**: The highest and lowest price recorded during the trading session.
- **Close**: The final price at market close.
- **Volume**: The number of shares traded on that day.
- **MA20**: The 20-day moving average, which is a technical indicator that smooths short-term price fluctuations to reveal longer-term trends.

This dataset is essential for both visual and predictive analysis. It allows for pattern recognition (via candlestick plotting), trend detection (using moving averages), and forecasting (via regression models).

b. Implementation Overview

The following sections present key parts of the code implemented in this project. Each snippet is explained in detail to highlight its purpose and significance. Note that only critical components are shown here to preserve originality and focus on core logic.

(i) Data Loading with API + Fallback

```python
# Load stock data either from Alpha Vantage or fallback to local CSV if API fails
def load_data(ticker):
    try:
        ts = TimeSeries(key=ALPHA_VANTAGE_API_KEY, output_format='pandas')
        df, meta_data = ts.get_daily(symbol=ticker, outputsize='compact')
        df = df.rename(columns={
            "1. open": "Open",
            "2. high": "High",
            "3. low": "Low",
            "4. close": "Close",
            "5. volume": "Volume"
        })
        df.index = pd.to_datetime(df.index)
        df = df.sort_index()
        source = "alpha_vantage"  # Indicates data was successfully loaded from the API
    except Exception as e:
        # If API fails, use pre-saved mock data from the local 'mock_data' folder
        st.warning("⚠ Could not load data from Alpha Vantage. Switching to fallback CSV data.")
        try:
            df = pd.read_csv(f"mock_data/{ticker}.csv", index_col=0, parse_dates=True)
            source = "csv"
        except:
            # If CSV is also missing, return error
            st.error("❌ No local CSV data available either. Please try a different stock ticker symbol.")
            return None, None
    return df, source
```

Explanation:
This function retrieves stock data from the Alpha Vantage API service. If the API cannot be reached (key limit is hit, or no internet) the function is coded to read data from local csv files as a fallback. This will always give the app some data to use. This redundancy was a design decision to give the user reliance in the app that they will always have some source of stock data.

(ii) Candlestick chart with moving average

```
def plot_candlestick_chart(df, ticker):
    df["20MA"] = df["Close"].rolling(window=20).mean()
    fig = go.Figure()
    fig.add_trace(go.Candlestick(...))
    fig.add_trace(go.Scatter(x=df.index, y=df["20MA"], ...))
    fig.update_layout(template='plotly_dark', ...)
    st.plotly_chart(fig)
```

Explanation:
This section visualizes the stock price using a candlestick chart with a 20-day moving average overlay. The visual insight helps users quickly spot bullish or bearish trends, and it's built with Plotly for interactivity and Streamlit for easy integration.

(iii) Trend Analysis using Moving Average

```
def analyze_trend(df):
    df["20MA"] = df["Close"].rolling(window=20).mean()
    df["50MA"] = df["Close"].rolling(window=50).mean()
    last_close = df["Close"].iloc[-1]
    ma_20 = df["20MA"].iloc[-1]
    ma_50 = df["50MA"].iloc[-1]
    recent_trend = df["20MA"].iloc[-5:].mean() - df["20MA"].iloc[-10:-5].mean()

    if last_close > ma_20 > ma_50 and recent_trend > 0:
        return "✅ Strong upward momentum..."
    elif last_close < ma_20 < ma_50 and recent_trend < 0:
        return "❌ Downward trend..."
    else:
        return "ℹ️ Sideways trend..."
```

Explanation:
The comparison of these factors allows the discovery of the trend.
Looking at the moving average over 20 and 50 days.
The trend of the 20MA over the last few days
The reason this feature helps new investors is that the advice is easy to comprehend.

(iv) Stock Price Reduction (Linear Regression)

```python
# Predict future stock prices using linear regression for the next few days
def predict_future_prices(df, days_ahead=5):
    df = df.copy()
    df["Days"] = np.arange(len(df))  # Add a numerical feature for regression
    X = df[["Days"]]
    y = df["Close"]

    model = LinearRegression()
    model.fit(X, y)

    # Predict prices for future days
    last_day = df["Days"].iloc[-1]
    future_days = np.arange(last_day + 1, last_day + 1 + days_ahead).reshape(-1, 1)
    predicted_prices = model.predict(future_days)

    future_dates = pd.date_range(start=df.index[-1] + pd.Timedelta(days=1), periods=days_ahead)
    prediction_df = pd.DataFrame({
        "Date": future_dates,
        "Predicted Close": predicted_prices
    })
```

Explanation:
This part predicts the cost of utilizing solar energy in the near future using basic Linear Regression. It's for introducing machine learning, not giving precise predictions.

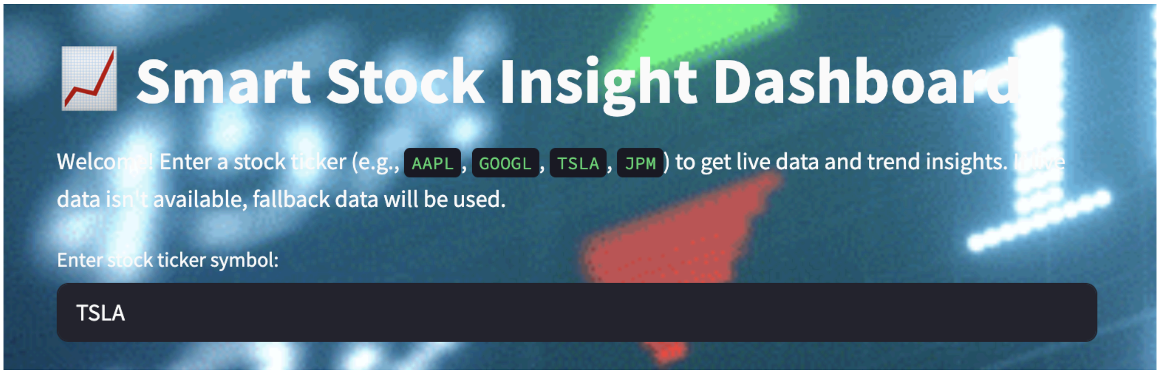
(v) Custom Background Styling (GIF Animation)

```python
def display_background():
    with open("static/stock_bg.gif", "rb") as f:
        gif_base64 = base64.b64encode(f.read()).decode("utf-8")
    st.markdown(f"""
        <style>
            .stApp {{
                background-image: url("data:image/gif;base64,{gif_base64}");
                background-size: cover;
                background-position: center;
            }}
        </style>
    """, unsafe_allow_html=True)
```

Explanation:
The code allows the dashboard to use a GIF for the background. Purely for UI enhancement, it gives the app a modern and interesting look to its users.

6. RESULTS

(i) Smart Stock Insight Dashboard



You can see the main page of the Streamlit-powered stock market dashboard here. Users type in a ticker symbol (for example, TSLA, AAPL, GOOGL or JPM) using the interface to learn the latest facts and trends about the stock. Entering a ticker symbol is made simple and the service clearly tells the user that it will use alternative data if live information is not available. At this point, the application introduces itself to new users and intends to show them how to interact with the dashboard and use the app

(ii) TSLA Candlestick Chart

Tesla's (TSLA) 20-day moving average (MA) is shown next to the candlesticks in this live chart. The chart track daily changes in price and highlights open, high, low and close figures for every day. To show the direction of price movement, the blue 20-day MA (moving average) removes bumps in the shorter-term trend. The period on the chart starts with declining figures and ends with the stock price rising significantly. Here, visitors can see current trends in TSLA's prices which show its level of volatility.

(iii) Investment Recommendation

The image demonstrates a fully automated investment suggestion from the dashboard. After analyzing the data, it is clear that TSLA is on an upward trend because the price sits above both the 20-day and 50-day averages which are traveling upward as well. The report recommends that this is a suitable time to invest, outlining the main points from the studied technical indicators. Technical analysis is explained in a way that users can implement without knowing a lot about financial metrics.

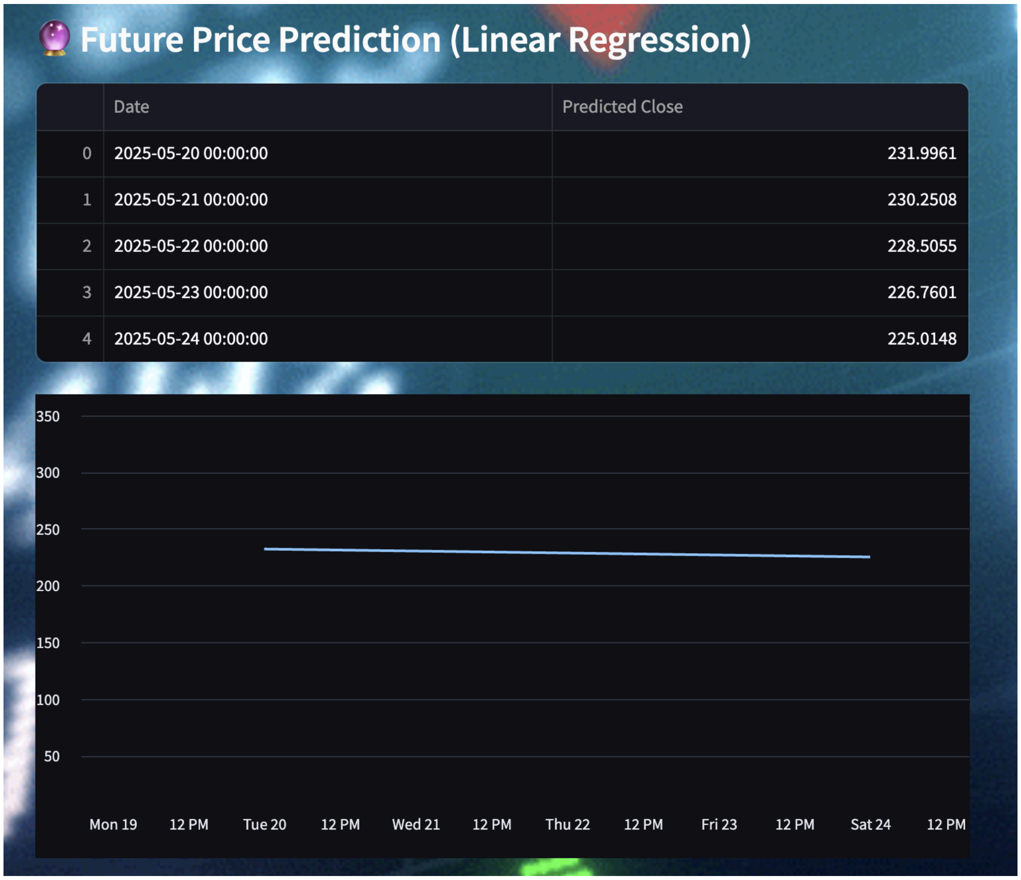
(iii) Historical Data Table (last 30 days)


Historical Data Table (Last 30 Days)

date	Open	High	Low	Close	Volume	20MA	50MA
2025-05-06 00:00:00	273.105	277.73	271.35	275.35	76715792	261.6115	261.279
2025-05-07 00:00:00	276.88	277.92	271	276.22	71882408	264.3295	260.7474
2025-05-08 00:00:00	279.63	289.8	279.41	284.82	97539448	264.9605	260.6278
2025-05-09 00:00:00	290.21	307.04	290	298.26	132387835	267.2535	260.954
2025-05-12 00:00:00	321.99	322.21	311.5	318.38	112826661	270.557	261.462
2025-05-13 00:00:00	320	337.5894	316.8	334.07	136992574	274.643	262.4504
2025-05-14 00:00:00	342.5	350	337	347.68	136997264	279.3215	263.9632
2025-05-15 00:00:00	340.34	346.1393	334.7153	342.82	97882596	284.385	265.2376
2025-05-16 00:00:00	346.24	351.62	342.33	349.98	95895665	289.8155	266.9682
2025-05-19 00:00:00	336.3	343	333.37	342.09	88869853	295.545	268.5566

A table of TSLA's recent stock data for the last 30 days is shown in this image. You will find columns in the table for date, open, high, low, close, volume, 20-day moving average and 50-day moving average. Detailed analysis, showing current and recent trends in prices, volume traded and major charts, is now possible for users. The platform makes it possible for users to examine and validate the trends and suggestions suggested by the charts.

(iii) Future Price Prediction (Linear Regression)



🔮 Future Price Prediction (Linear Regression)

	Date	Predicted Close
0	2025-05-20 00:00:00	231.9961
1	2025-05-21 00:00:00	230.2508
2	2025-05-22 00:00:00	228.5055
3	2025-05-23 00:00:00	226.7601
4	2025-05-24 00:00:00	225.0148

This figure displays what a linear regression model can predict for the future of this company's stock.

For convenience, the next five day's forecasted closing prices can be found at the top and will be from May 20, 2025, to May 24, 2025. Each date on the rows matches the predicted close value and you can see the values decrease step by step from 231.9961 to 225.0148.

You can also find a line chart at the bottom which shows how the predictions will develop with time. According to the chart, the model suggests that the stock may drop a little bit over the next few days.

Here, machine learning is used on the dashboard to project likely price movement in the near-term, drawing on trends from past data.

7. Conclusion

a. Conclusion

The system provides both an accurate and useful response to stock market unpredictability using a blend of machine learning and real-time visual data. With access to historical results, indicators and machine learning, the system transforms complex financial analysis so that anyone can make sense of it. The key goal to provide good stock trend forecasts and sound investment tips was met with the use of Alpha Vantage API, Python, Streamlit and Scikit-learn. The work involves getting live or previous stock updates, showing how trends develop with candles and moving average lines and calculating future brief price changes by applying linear regression.

With this approach, users are able to improve how they invest by understanding the analysis better and receiving forecasts on a simple screen. When data can't be accessed directly, the system relies on CSV files to ensure reliable backup and adding visual tools through Plotly encourages more user interest.

Because the current method uses only a simple linear regression approach, it forms a stable foundation that allows the team to add advanced algorithms and improve features in the next versions. The project was carefully developed to ensure that the work was done promptly, without losing usefulness, emphasizing plainness and easiness of use.

Basically, using data science wisely in the system simplifies how people interact with finance. Because of this project, we can see how technological advances can tie raw financial details to useful information for investors, thus helping the financial technology industry grow further.

b. Future Scope

This overall effort builds a good basis for the future in finance and with extra improvements, can become a better and more useful prediction and advice platform.

- By using Advanced Predictive Models, they have also improved their approach by changing the old model to stronger methods such as LSTM, CNN, Random Forest and XGBoost. Such models perform well when understanding the irregular behavior of stock markets.

- Based on user profiling, personalized suggestions for investments are created by taking into account user tolerance for risk, what they want to invest in and past behavior in investments. Recommending movies can add usefulness to the platform for its users.

- With this Module, recommend and automate various types of investments using tools, reflecting the user's expected outcomes and using insights from data analysis.

- Explainable AI (XAI): Help users understand a model's decisions by using, for example, SHAP or LIME which make forecasts from AI more transparent.

- When a system moves to a cloud-based solution, scaling up the system, giving access to multiple users, storing information and always improving the model become possible using new data streams.

- Users can be informed about important changes in monetary trends, price movements or chances to buy or sell quickly thanks to improved dashboards and notifications.

References

[1] S. Acharjee and R. Lamba, "An explainable machine learning approach to predict stock market movements," *Heliyon*, vol. 10, no. 6, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590291124000615

[2] H. Le, "Stock market forecasting based on artificial intelligence technology," M.S. thesis, Dept. of Information and Decision Sciences, California State Univ., San Bernardino, CA, USA, 2023. [Online]. Available: https://scholarworks.lib.csusb.edu/etd/1324/

[3] Y. Yang, S. Zhang, and L. Wang, "Data-driven stock forecasting models based on neural networks," *Journal of King Saud University – Computer and Information Sciences*, vol. 37, no. 1, pp. 1–10, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917423002866

[4] S. Latif, F. Aslam, P. J. S. Ferreira, and S. Iqbal, "Integrating macroeconomic and technical indicators into forecasting the stock market: A data-driven approach," *Economies*, vol. 13, no. 1, p. 6, 2025. [Online]. Available: https://www.mdpi.com/2227-7099/13/1/6

[5] Y. Dai, J. Li, and X. Zhang, "Hybrid machine learning models for long-term stock market forecasting: Integrating technical indicators," *Journal of Risk and Financial Management*, vol. 18, no. 4, p. 201, 2023. [Online]. Available: https://www.mdpi.com/1911-8074/18/4/201

[6] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLOS ONE*, vol. 12, no. 7, p. e0180944, 2017. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0180944

[7] G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques – Part II: Soft computing methods," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5932–5941, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417408004417