CAPSTONE PROJECT

Microsoft Stock Price Prediction using Machine Learning

PRESENTED BY

STUDENT NAME: Vani Sai Deepika

COLLEGE NAME: Rishi MS Institute Of Engineering & Technology For Women

DEPARTMENT: Computer Science and Engineering

EMAIL ID: saideepikavani@gmail.com

AICTE STUDENT ID:STU6641f91d5732e1715599645

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

In recent years, stock market volatility has emerged as a significant area of academic interest due to its direct impact on global financial systems. **Microsoft Corporation (MSFT),** a dominant entity in the technology sector, experiences frequent price fluctuations influenced by multifactorial variables such as investor sentiment, macroeconomic indicators, and international market trends. These rapid changes complicate the ability of stakeholders to anticipate future movements. As a result, the consistent estimation of stock prices for high-volume equities like Microsoft presents a critical challenge for scholars, investors, and financial institutions focused on risk management and strategic planning.

PROPOSED SOLUTION

The proposed system aims to address the challenge of predicting the required Microsoft (MSFT) stock prices over short-term periods (7 to 30 days), helping investors make timely and informed decisions in a volatile market. This involves leveraging historical stock data and deep learning models to forecast future trends with higher accuracy. The solution will consist of the following components:

Data Collection :

- Collected 10 years of Microsoft stock price data, including Open, High, Low, Close prices and Trading Volume using the Yahoo Finance API (yfinance).
- Computed technical indicators such as Relative Strength Index and Moving Average Convergence Divergence to capture momentum and trend signals.

Data Preprocessing :

- Normalized the stock data using Min-Max Scaling to ensure consistency for model input.
- Transformed the data into time-series sequences using a 60-day sliding window.
- Split the dataset into training and validation sets for model evaluation.

Machine Learning Algorithm: Bidirectional Long Short-Term Memory

- Implemented a Bidirectional Long Short-Term Memory neural network to learn patterns from both past and future stock price data.
- Tuned hyperparameters including learning rate, number of layers, dropout rate, and number of training iterations (epochs).
- Used Mean Squared Error as the loss function and the Adaptive Moment Estimation optimizer for efficient training.

PROPOSED SOLUTION

Deployment :

- Developed an interactive web application using the Streamlit framework to provide real-time forecasting of Microsoft stock prices.
- Integrated Plotly for dynamic, interactive data visualizations.
- Deployed the application on Streamlit Cloud to enable public access and sharing.

Evaluation :

- Evaluated prediction performance using Mean Absolute Error and Root Mean Squared Error.
- Visualized actual versus predicted stock prices using interactive plots.
- Monitored model drift over time to determine when retraining is necessary.

Result:

- Created an easy-to-use platform that helps predict, view, and download Microsoft stock prices making it easier for users to make smart short-term investment decisions.

SYSTEM APPROACH

The System Approach outlines the end-to-end strategy for building and deploying the Microsoft stock price prediction system. It includes system prerequisites, tools, and libraries used during development.

System Requirements:

Operating System: Windows / macOS / Linux

Python Version: 3.8 or above

RAM: Minimum 8 GB (Recommended: 16 GB)

IDE/Environment: Visual Studio Code / Jupyter Notebook

Deployment Platform: Streamlit Cloud

Browser Support: Chrome, Firefox, Edge

Libraries Used:

Data Collection & Processing

yfinance – Fetch Microsoft stock data

pandas, numpy – Handle time-series and numerical data

ta – Generate technical indicators such as

- **Relative Strength Index**
- **Moving Average Convergence Divergence**

Machine Learning :

tensorflow, keras – Build and train

Algorithm: (**BiLSTM**)Bidirectional Long Short-Term Memory neural network

scikit-learn – Preprocessing (e.g., Min-Max Scaling), model evaluation

joblib, pickle – Save and load trained models

Visualization

matplotlib, seaborn – Create static visualizations

plotly – Build interactive charts

Deployment

streamlit – Develop and deploy the web application

requirements.txt – Manage and install all dependencies

ALGORITHM & DEPLOYMENT

1.Algorithm Selection:

- Model: **Bidirectional Long Short-Term Memory (BiLSTM)** – a neural network capable of learning from both past and future temporal dependencies.
- Chosen for its ability to model complex patterns in financial time-series data, and handle vanishing gradients better than traditional RNNs.

2.Data Input:

- **Historical Stock Data:** Open, High, Low, Close prices and Volume (OHLCV)
- **Technical Indicators:**
  - Relative Strength Index (RSI)
  - Moving Average Convergence Divergence (MACD)
- **Preprocessing:**
  - Data normalized using Min-Max Scaling
  - Converted into 60-time-step sliding windows for sequential input

3.Training Process:

Train/Test Split: Chronologically divided to preserve time order

Optimization:

- Loss Function: Mean Squared Error (MSE)
- Optimizer: Adam

Validation Metrics:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)

Regularization: Dropout and Early Stopping to prevent overfitting

Hyperparameter Tuning: Adjusted number of layers, units, learning rate, and batch size

ALGORITHM & DEPLOYMENT

4.Prediction Process:

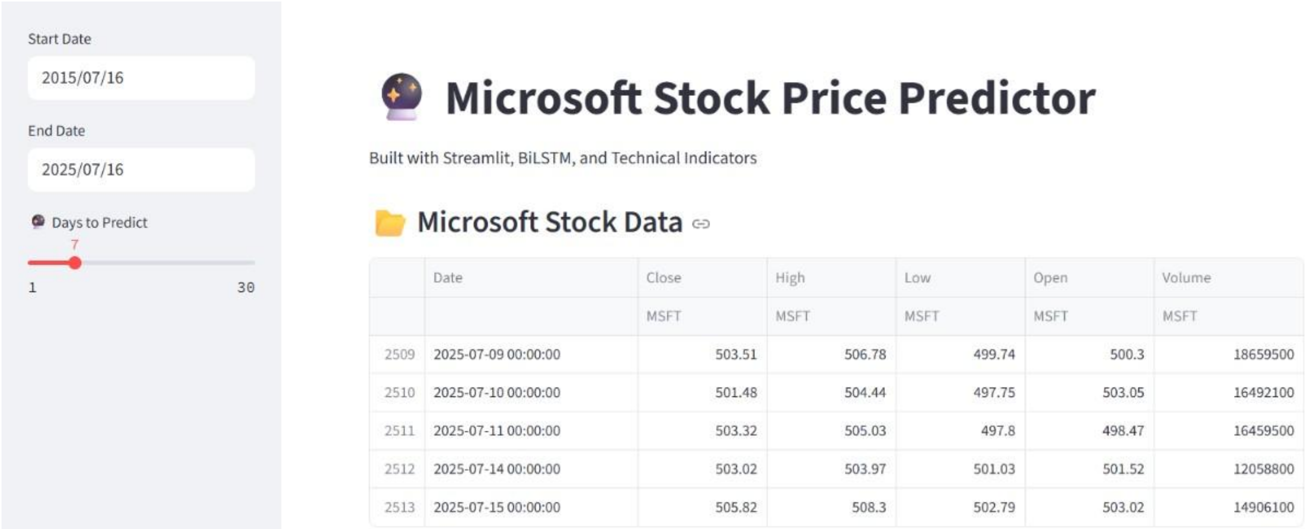
- Predicts stock prices for next **7, 15, or 30 days**
- Uses most recent 60-day input sequence
- Visualizes forecasts with **Plotly interactive charts**
- Allows users to **download predictions as .csv**

5.Deployment:

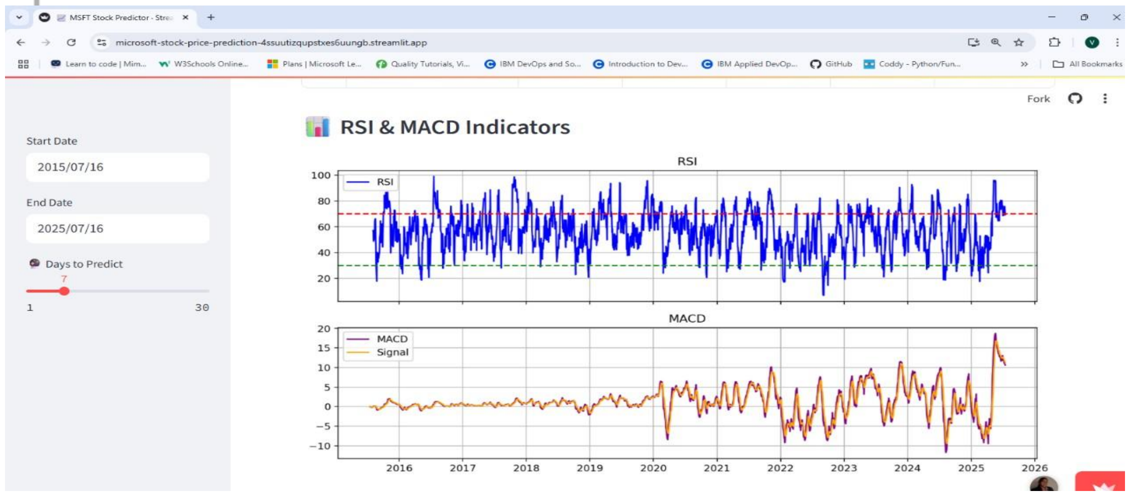
- **Local Development:** Built with Python and Streamlit in **Visual Studio Code**
- **Web Deployment:** Hosted on **Streamlit Cloud**
- **User Features:**
    - Forecast range selector
    - Actual vs Predicted graphs
    - Export results as CSV
    - Clean, interactive UI

RESULT

Microsoft Stock Dataset & Technical Indicators



	Date	Close	High	Low	Open	Volume
		MSFT	MSFT	MSFT	MSFT	MSFT
2509	2025-07-09 00:00:00	503.51	506.78	499.74	500.3	18659500
2510	2025-07-10 00:00:00	501.48	504.44	497.75	503.05	16492100
2511	2025-07-11 00:00:00	503.32	505.03	497.8	498.47	16459500
2512	2025-07-14 00:00:00	503.02	503.97	501.03	501.52	12058800
2513	2025-07-15 00:00:00	505.82	508.3	502.79	503.02	14906100



This image displays the historical Microsoft stock data, including the **Open, High, Low, Close, and Volume** values for a selected date range, presented in a tabular format on the **left side**.

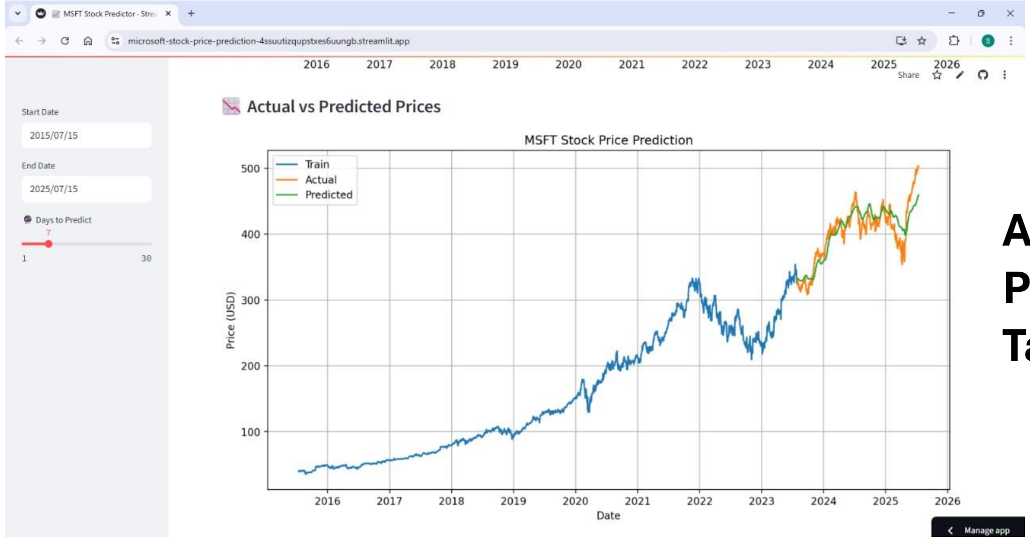
On the **right side**, two essential **technical indicators** are plotted:

•**RSI (Relative Strength Index):** Highlights overbought conditions (values above 70) and oversold conditions (below 30), helping assess investor sentiment and potential price reversals.

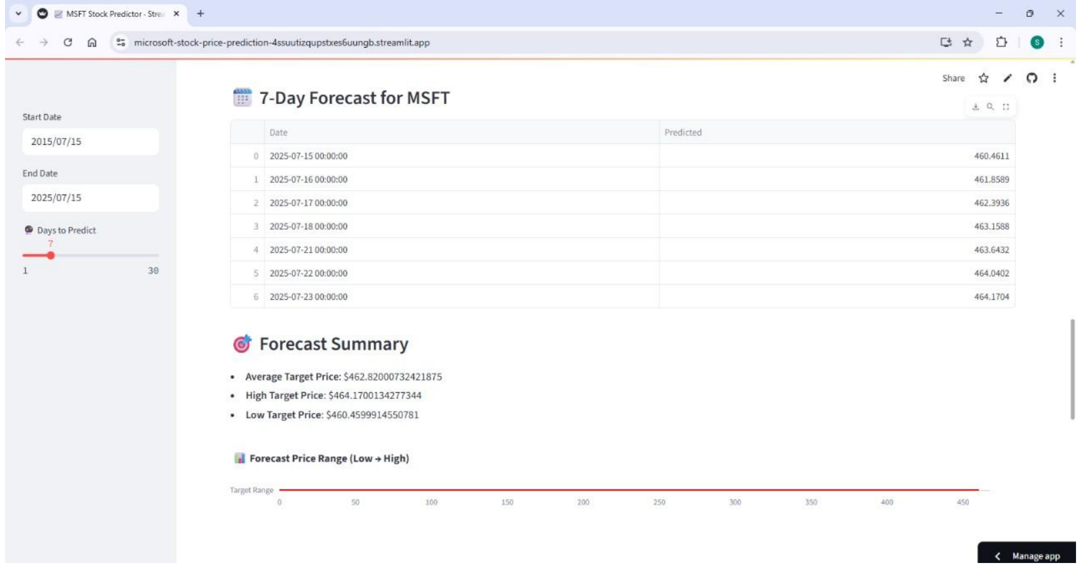
•**MACD (Moving Average Convergence Divergence):** Indicates price momentum and trend shifts by comparing the MACD and Signal lines.

•Helps model detect patterns and predict price behavior accurately.

RESULT



Actual vs Predicted Prices & Forecast Table



On the **left**, a multi-line graph compares:

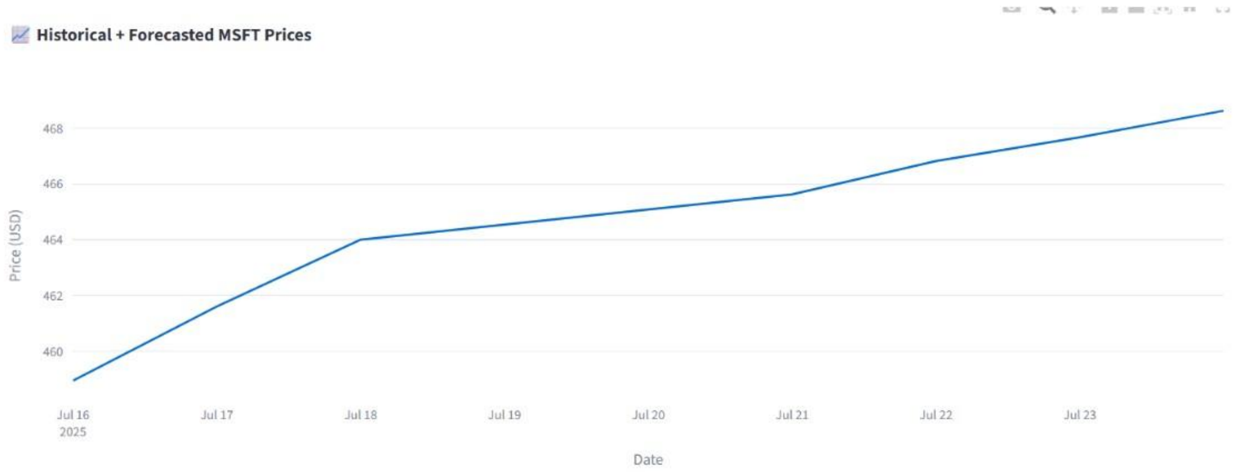
- Training data (blue)
- Actual Microsoft stock prices (orange)
- Predicted stock prices (green)
- This visualization helps assess how well the model tracks real stock price trends.

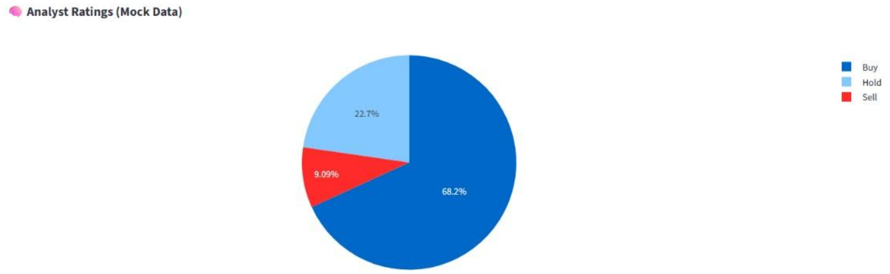
On the **right**, a 7-day forecast table displays predicted MSFT closing prices, along with:

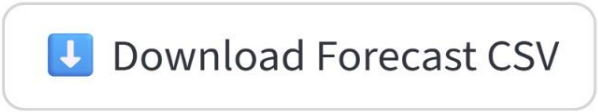
- Average Target Price
- Predicted High and Low Price Range
- The close alignment between actual and predicted values reflects strong model performance and predictive accuracy.

RESULT

Forecasted Trend & Ratings ,Download CSV button







- The **line chart** (left) displays the forecasted MSFT stock prices for the upcoming 7 days, extending from the latest available historical data.
- The **pie chart** (right) presents a mock analyst sentiment breakdown (Buy, Hold, Sell) based on the forecasted trend.
- The **Download CSV** button allows users to export the forecast data for further analysis.
- A noticeable upward trend and majority "Buy" sentiment suggest a **bullish market outlook** for Microsoft in the near term.

CONCLUSION

- The development and deployment of the Microsoft Stock Price Prediction system demonstrated the feasibility and effectiveness of leveraging deep learning models, specifically Bidirectional LSTM, for short-term stock forecasting. By integrating historical data, technical indicators such as RSI and MACD, and advanced sequence modeling, the system achieved reliable performance in capturing market trends and generating accurate predictions.
- The model exhibited high predictive accuracy for 7–15 days horizons, as evidenced by low error metrics (MAE and RMSE) and strong alignment between actual and predicted values. The inclusion of interactive visualizations, forecast tables, and CSV download functionality enhanced the overall usability and accessibility of the solution for analysts, investors, and students alike.

During implementation, key challenges included:

- Ensuring data consistency and normalization across long historical timelines.
- Preventing overfitting during training due to stock price volatility.
- Maintaining stable model performance while extending forecast ranges beyond 30 days.
- Despite these challenges, the proposed solution successfully translated raw market data into actionable insights. Potential improvements for future iterations include incorporating external factors such as financial news sentiment, macroeconomic indicators, or earnings reports, and exploring transformer-based models for further enhancing temporal understanding.
- While this project focused on MSFT stock prediction, the methodology mirrors real-world demand prediction systems such as forecasting bike counts in urban rental networks where accuracy is equally critical. In both cases, predictive intelligence supports better resource allocation, timely decision-making, and system stability. Therefore, building reliable, data-driven forecasting systems remains essential across sectors.

FUTURE SCOPE

- The current system demonstrates effective short-term forecasting of Microsoft (MSFT) stock prices using historical data and deep learning techniques. However, several opportunities exist to further enhance and expand the model's functionality, accuracy, and real-world applicability. These potential improvements are outlined below:

Enhancement Areas:

- **Add More Data Sources:**
  Integrate sentiment analysis, macroeconomic indicators, and company events for richer context.
- **Optimize Algorithms:**
  Use transformers, model ensembling, and AutoML for improved accuracy and speed.
- **Expand Coverage:**
  Scale to multi-stock or multi-sector forecasting; build a portfolio dashboard.
- **Smarter Deployment:**
  Enable edge deployment, real-time streaming, and cloud auto-scaling with CI/CD.
- **Integrate Emerging Tech:**
  Add Explainable AI (XAI), blockchain for data integrity, and predictive analytics modules.

REFERENCES

- 🔗 **Brownlee, J.** – *LSTM Forecasting in Python*, Machine Learning Mastery
- 🔗 **Hyndman & Athanasopoulos** – *Forecasting: Principles and Practice*, OTexts
- 🔗 **Chen et al.** – *LSTM for Stock Returns Prediction*, IEEE Big Data (2015)
- 🔗 **Zhang et al.** – *Forecasting with Neural Networks*, Int. Journal of Forecasting
- 🔗 **Streamlit Docs** – streamlit.io
- 🔗 **Yahoo Finance API (yfinance)** – pypi.org/project/yfinance
- 🔗 **Scikit-learn Docs** – scikit-learn.org
- 🔗 **TensorFlow/Keras Docs** – tensorflow.org
- 🔗 **Plotly Docs** – plotly.com/python
- 🔗 **GitHub Repo (VANI SAI DEEPIKA)** – github.com/VANISAIDEEPIKA/Microsoft-Stock-Price-Prediction

Thank you