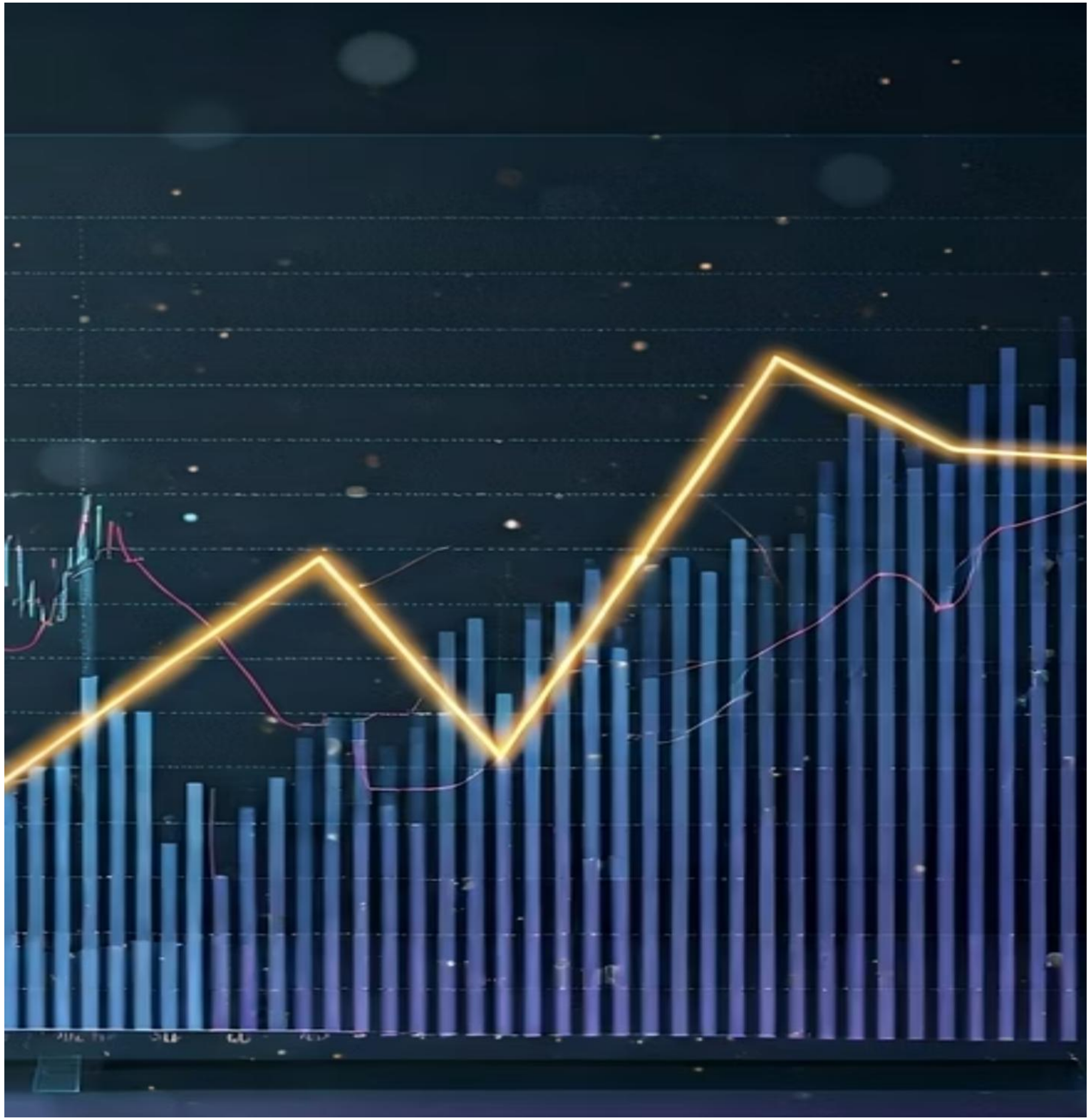
STOCK PRICE PREDICTION USING MACHINE LEARNING



PRESENTED BY

STUDENT NAME: M N RAGUL

COLLEGE NAME: JNN INSTITUTE OF ENGINEERING

DEPARTMENT: B TECH AI & DS

EMAIL ID:rnagarajan493@gmail.com

AICTE STUDENT ID:STU67698d68aeef01734970728

PROBLEM STATEMENT

Problem Statement: Stock Price Prediction Using Machine Learning

- In today’s fast-paced financial markets, accurately predicting stock price movements has become crucial for investors, traders, and financial institutions. With the rapid fluctuation in stock prices influenced by various factors such as market trends, trading volume, and economic indicators, making informed investment decisions is a significant challenge.
- Predicting the future trend of stock prices enables market participants to minimize risk and maximize returns.
- The objective is to develop a machine learning model that can forecast the stock price movement (e.g., upward or downward trend) based on historical data.
- The model should learn patterns from features like opening price, closing price, high, low, volume, and technical indicators to predict the stock’s behavior in the near future.

PROPOSED SOLUTION

1. Data Collection & Preparation

- Fetch historical stock data (Tesla Dataset from 2010–2017) including Open, High, Low, Close, Volume
- Handle missing dates (weekends/holidays): drop NA or fill-forward.

2. Exploratory Data Analysis (EDA)

- Visualize time-series (price trends, seasonality).
- Compute technical indicators: Moving Averages (SMA/EMA), RSI, MACD.
- Analyze volatility, autocorrelation, and periodic trends

3. Feature Engineering

- Create lagged features (e.g., previous 3–5 days' closes).
- Add technical indicators as new columns.
- Include date-related features: day-of-week, month, quarter.

4. Train/Test Split & Scaling

- Split chronologically (e.g., 80% train, 20% test).
- Apply scaling (MinMaxScaler on numeric features to [0,1]).

PROPOSED SOLUTION

5. Model Selection

Model	Use Case
Linear Regression	Baseline forecasting
Random Forest / XGBoost	Captures non-linear relationships
LSTM / CNN-LSTM	Sequence modeling for time-series
CatBoost	Handles categorical/date features

6. Evaluation

- For regression: use **RMSE**, **MAE**, **MAPE**; visualize predicted vs. actual.
- For classification (up/down): use **Accuracy**, **Precision**, **Recall**, **F1-Score**.
- Implement backtesting strategies based on model outputs.

SYSTEM REQUIREMENTS

- **Hardware Requirements:**
- **Processor:** Intel Core i5 or higher (or equivalent AMD/Ryzen)
- **RAM:** Minimum 8 GB (16 GB recommended for deep learning models like LSTM)
- **Storage:** Minimum 5 GB of free disk space (for datasets, models, and logs)
- **GPU (Optional):** NVIDIA GPU with CUDA support for training LSTM models faster

- **Software Requirements :**
- **Operating System:** Windows 10 / Linux / macOS
- **Python Version:** Python 3.8 or above
- **IDE:** Jupyter Notebook , Google Colab
- **Web Browser:** Chrome or Firefox

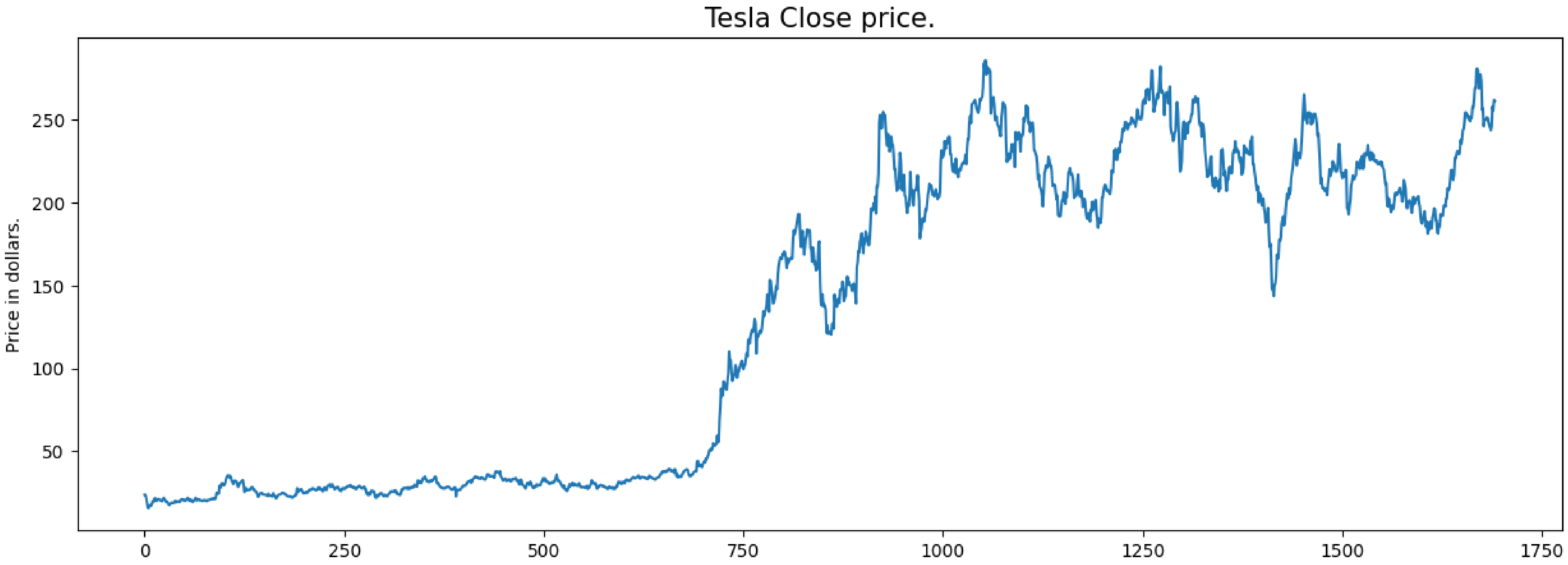
ALGORITHM & DEPLOYMENT

- **In the Algorithm section, describe the machine learning algorithm chosen for predicting stock price movements.**
- Algorithm Selection:
  - LSTM is ideal for capturing time-series dependencies in stock prices, making it suitable for sequential data forecasting.
  - XG Boost is effective for tabular data with non-linear relationships and performs well with engineered features like technical indicators.
- Data Input:
  - **Core features** include historical Open, High, Low, Close, Volume (OHLCV), and technical indicators like RSI, MACD, and moving averages.
  - Date-based inputs such as day of the week, month, or holiday flags help capture cyclical or seasonal patterns in stock movements.
- Training Process:
  - **Time-based data split** ensures chronological order is preserved, avoiding data leakage during training and testing phases.
  - **Hyperparameter tuning** (via GridSearchCV or manual tuning) and normalization (e.g., MinMaxScaler for LSTM) enhance model performance.
- Prediction Process:
  - The trained model predicts the **next-day closing price** or **trend direction** using the most recent sequence of input features.
  - For **real-time use**, the model can be deployed as an API that ingests live market data and returns predictions instantly

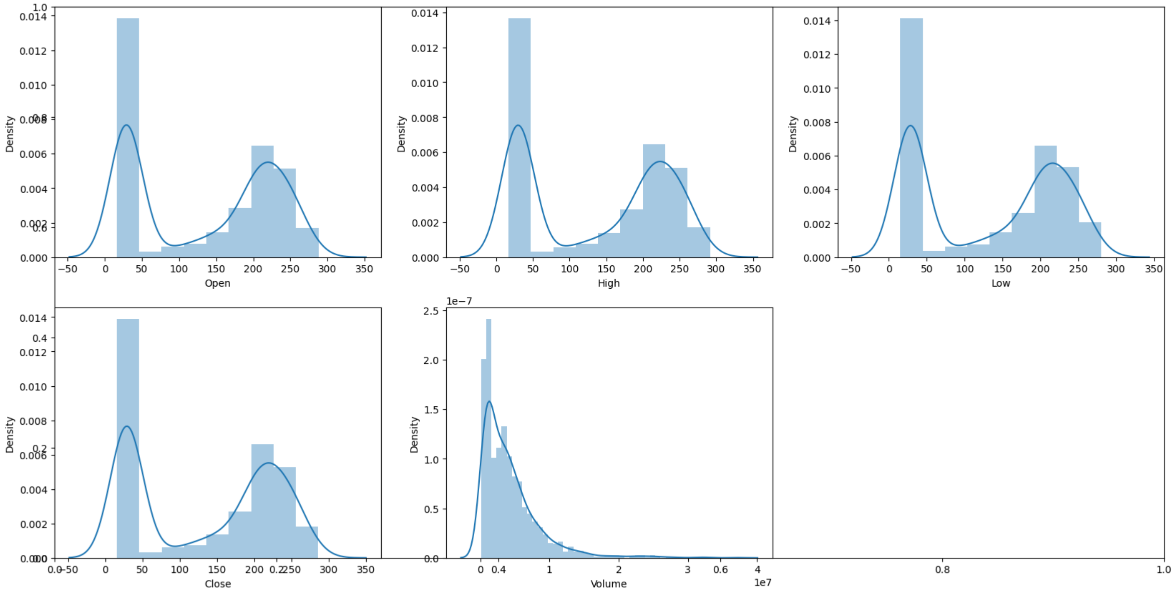
RESULT

- Present the results of the machine learning model in terms of its accuracy and effectiveness in predicting stock prices. Include visualizations and comparisons between predicted and actual stock prices to highlight the model’s performance.
- Use evaluation metrics such as **Mean Absolute Error (MAE)**, **Root Mean Squared Error (RMSE)**, and **R^2 Score** to assess model accuracy.
- Provide **line charts** comparing predicted vs. actual stock prices over time to visually demonstrate how closely the model follows real market trends.

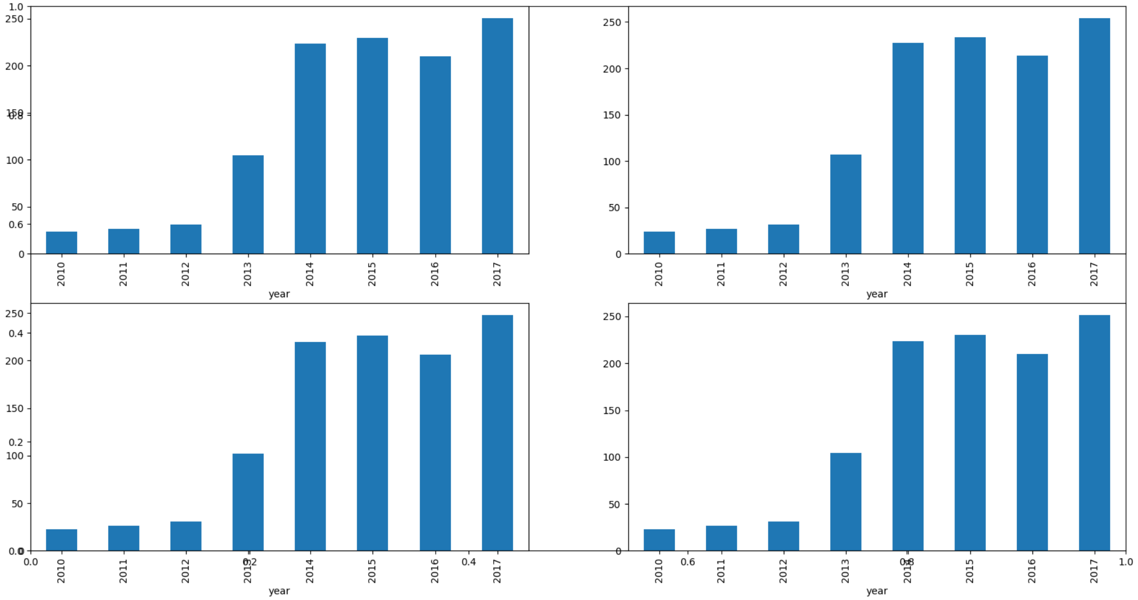
OUTPUT


Tesla Close price.
Price in dollars.
50
100
150
200
250
0
250
500
750
1000
1250
1500
1750

OUTPUT

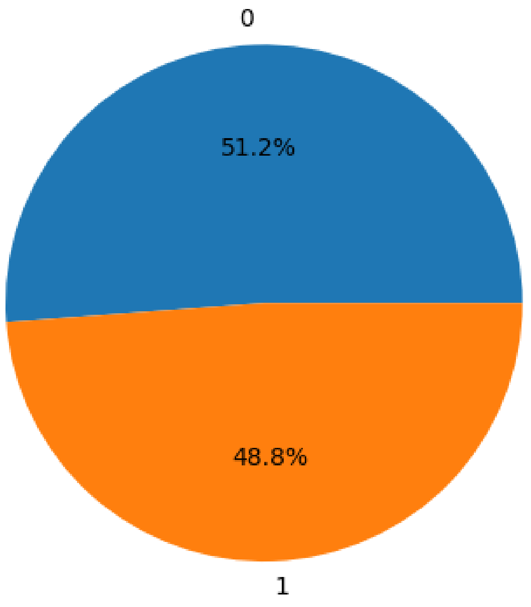


Distribution Plot

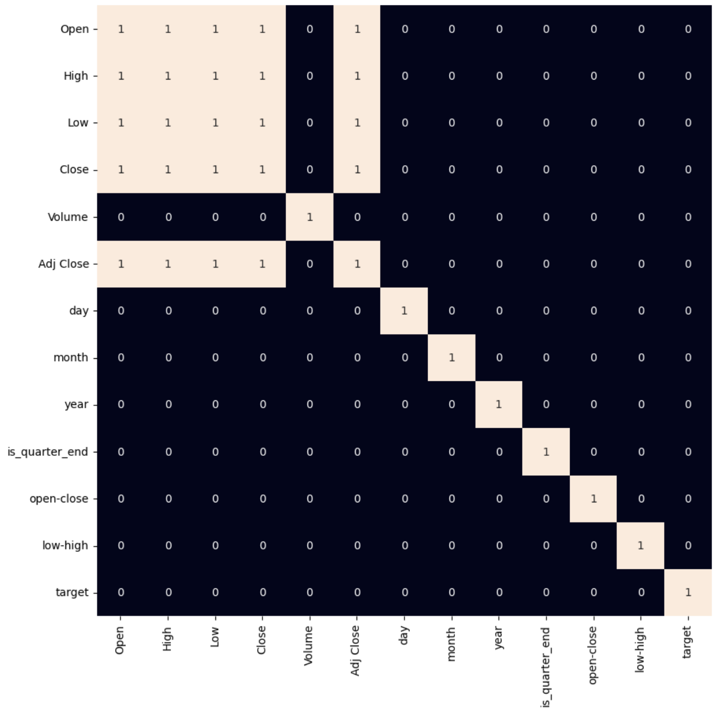


Bar Graph

OUTPUT

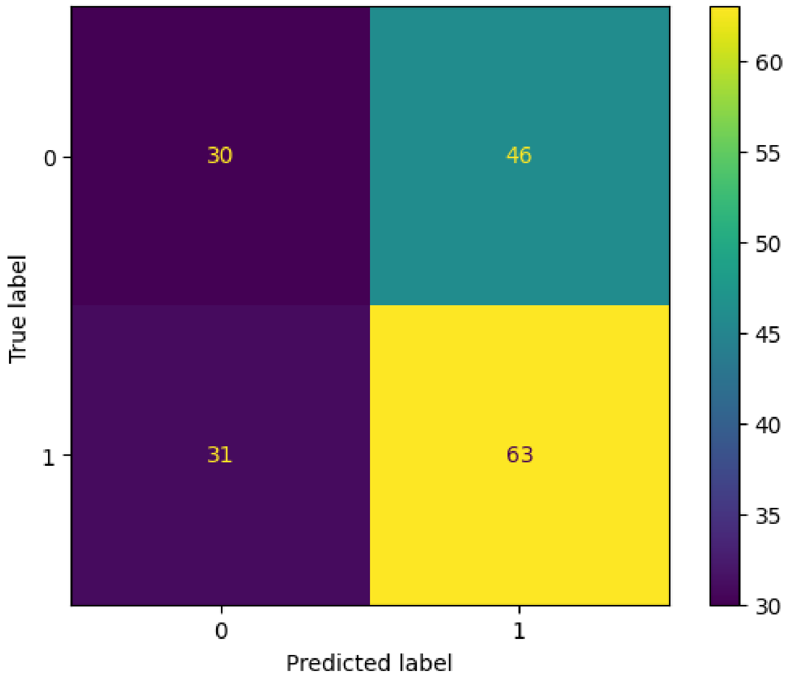


Training Set



	Open	High	Low	Close	Volume	Adj Close	day	month	year	is_quarter_end	open-close	low-high	target
Open	1	1	1	1	0	1	0	0	0	0	0	0	0
High	1	1	1	1	0	1	0	0	0	0	0	0	0
Low	1	1	1	1	0	1	0	0	0	0	0	0	0
Close	1	1	1	1	0	1	0	0	0	0	0	0	0
Volume	0	0	0	0	1	0	0	0	0	0	0	0	0
Adj Close	1	1	1	1	0	1	0	0	0	0	0	0	0
day	0	0	0	0	0	0	1	0	0	0	0	0	0
month	0	0	0	0	0	0	0	1	0	0	0	0	0
year	0	0	0	0	0	0	0	0	1	0	0	0	0
is_quarter_end	0	0	0	0	0	0	0	0	0	1	0	0	0
open-close	0	0	0	0	0	0	0	0	0	0	1	0	0
low-high	0	0	0	0	0	0	0	0	0	0	0	1	0
target	0	0	0	0	0	0	0	0	0	0	0	0	1

Heat Map



Confusion Matrix

CONCLUSION

- Summarize the findings and discuss the effectiveness of the proposed stock price prediction model. Highlight any challenges encountered during the implementation, such as handling high volatility, data noise, or overfitting.
- Mention potential improvements like incorporating sentiment analysis or real-time news feeds.
- Emphasize the importance of accurate stock price forecasting in supporting smarter investment decisions, minimizing financial risk, and enhancing portfolio management strategies.

FUTURE SCOPE

- Discuss potential enhancements and expansions for the stock price prediction system.
- This could include integrating alternative data sources such as financial news sentiment, social media trends, or macroeconomic indicators to improve forecasting accuracy.
- Future improvements may also involve optimizing the model with advanced techniques like transformer-based architectures or ensemble methods.
- Additionally, deploying the system on scalable cloud platforms or incorporating real-time data pipelines can make the model suitable for live trading environments.

REFERENCES

1.Geeks for Geeks. (2021). *Stock Price Prediction using Machine Learning in Python*.
Link: https://www.geeksforgeeks.org/machine-learning-stock-price-prediction-using-machine-learning-in-python

2.Brownlee, J. (2020). *Introduction to Time Series Forecasting with Python*. Machine Learning Mastery.

3.Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.

4.Scikit-learn Developers. (2023). *Scikit-learn: Machine Learning in Python*.
Link: https://scikit-learn.org/stable/

5.TensorFlow Developers. (2023). *Time Series Forecasting*.
Link: https://www.tensorflow.org/tutorials/structured_data/time_series

THANK YOU