Report: Stock Price Prediction using LSTM, Random Forest & XGBoost

Name: Vaishali Deshmukh

Roll no: 250240325071

1. **Objective**

To predict stock closing prices using historical data with three models:

I'm testing **three different prediction methods** to see which one gives the most accurate guess of tomorrow's stock-market closing price:

1. **LSTM** – a deep-learning model that looks at the last few weeks of prices in order.
2. **Random Forest** – many decision-tree models averaged together.
3. **XGBoost** – lots of small trees built one after another to correct each other's mistakes.

The goal is to identify which model performs best based on accuracy metrics: RMSE, MAE, and R^2.

1. **MAE (Mean Absolute Error)**

What it means:
On average, how many rupees our prediction is wrong by.

Example:
If we predicted ₹105, but the actual price was ₹100, then the error is ₹5.
If the **MAE is 1.5**, it means we are usually off by about **₹1.50** per prediction.

2. **RMSE (Root Mean Squared Error)**

What it means:
Like MAE, but it **gives more weight to big mistakes**.

Example:
If one day we predict ₹110 but the real price is ₹90, the 20-rupee mistake hurts more in RMSE than in MAE.

3. **R^2 Score (R-squared)**

What it means:
It tells how well the model is doing overall.
If it's **close to 1**, the model is doing a great job.
If it's **close to 0**, the model is almost guessing.

2. Dataset Overview
- File Used: stocks.csv
- Columns: Date, Symbol, Open, High, Low, Close, Volume
- Stocks Included: Multiple, handled per unique Symbol

3. Data Preprocessing & Feature Engineering

1. **Cleaning Steps:**
   Sorted data by Symbol and Date
   Removed rows with missing values after creating lagged features.

2. **Lagged Features:**
   Moving Averages:
   MA5, MA10, and MA20 represent 5-day, 10-day, and 20-day moving averages of the Close price.
   Purpose: Capture trend and momentum over short, medium, and longer time frames.
   Models Used: All (Random Forest, XGBoost, LSTM, Directional LSTM)
3. **Price Range Feature**
   High_Low = High - Low
   Represents the daily volatility or price fluctuation.
   Helps the model understand how much the stock moved during the day.
   Models Used: All
4. **Daily Return**
   Daily_Return = Percentage change in closing price from the previous day.
   Daily_Return = df.groupby('Symbol')['Close'].pct_change()
   Captures the direction and strength of short-term changes.
   Models Used: All
5. **Target Variable Creation**
   Next_Close = Close price of the next day.
   This is the main variable we are trying to predict.
   Next_Close = df.groupby('Symbol')['Close'].shift(-1)
   Models Used: All regression models (RF, XGBoost, LSTM)

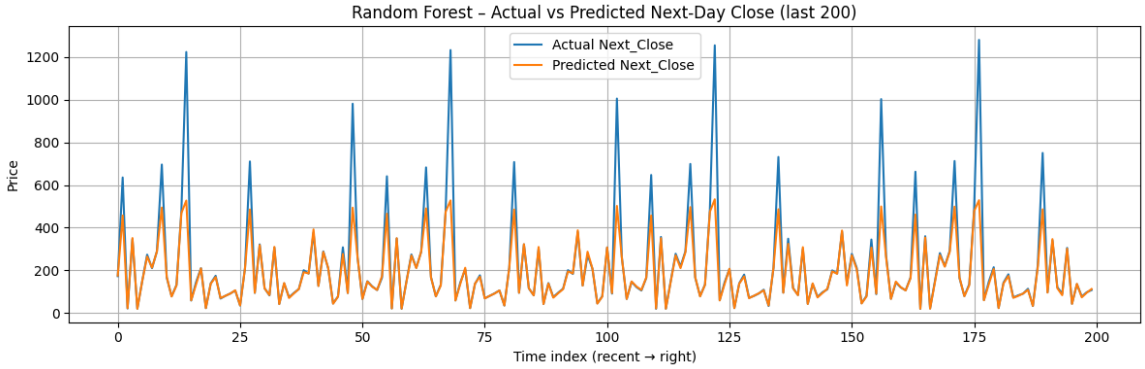
These features were used for Random Forest and XGBoost models. For LSTM, the multivariate input sequence included Open, High, Low, Close, and Volume.

4. **Model Architectures & Training**

Random Forest:

A machine learning model that builds many decision trees and takes the average of their outputs.

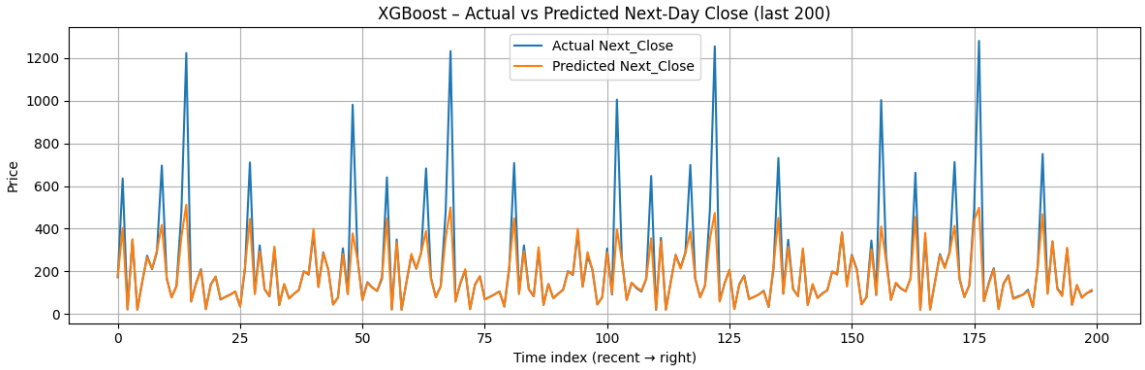
- **No loss function or optimizer needed**, as it's not a neural network.
- Trained using features like Open, High, Low, Close, Volume, and moving averages.



XGBoost:

An advanced tree-based model that builds trees one after another, fixing mistakes step by step.

- **Loss Function**: Mean Squared Error (by default).
- Optimized internally using gradient boosting techniques.

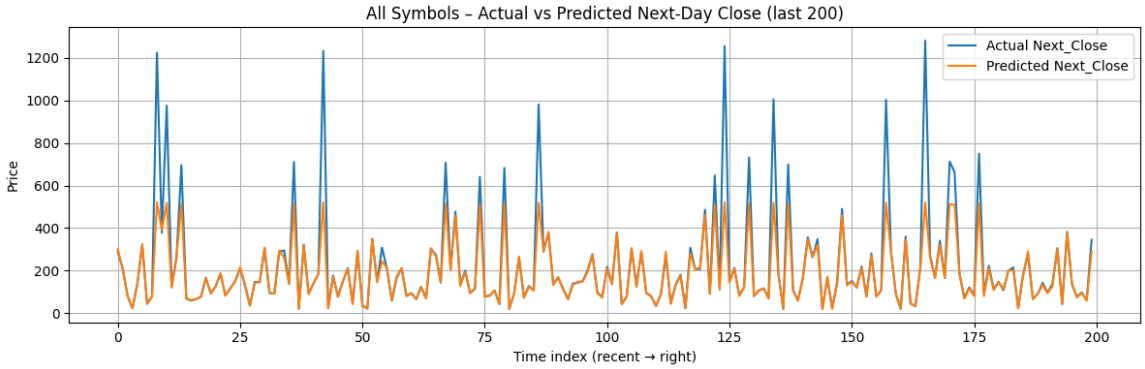


LSTM (Long Short-Term Memory):

A deep learning model that learns from sequences (past 20 days of stock prices). It is good at capturing time-based patterns.

- **Loss Function**: Mean Squared Error (MSE).

- **Optimizer**: Adam (a popular and fast optimizer for deep learning).



5. Evaluation Metrics

Used the following

RMSE: Root Mean Squared Error
MAE: Mean Absolute Error
R^2 Score: Proportion of variance explained by the model

6. Results Summary

Model Name	MAE	RMSE	R-Squared
Random Forest	9.023	42.856	0.9170
LSTM	9.678	41.989	0.9204
XGBoost	14.914	57.304	0.8517

7. Insights & Conclusions

- In this case study, I built and compared three powerful models — **Random Forest**, **XGBoost**, and **LSTM** — to **predict the next day's closing stock price** using historical data.

- Among all models, **LSTM performed the best**. It had the **lowest big error (RMSE)** and explained the **highest portion of stock price movement (R^2 = 0.920)**.
- **Random Forest** also performed well, giving similar average accuracy with only slightly higher errors.
- **XGBoost** lagged behind both in terms of accuracy and consistency.

7. Challenges Faced

Large Dataset Processing
The dataset had over **3 lakh records**, which increased training time.
Overfitting Risk
Tree-based models like Random Forest gave **very high R^2**, which looked good but were **too close to the naïve baseline**.
Time Formatting Errors
Faced issues with inconsistent date formats (e.g., 13-01-1998) while parsing.

8. **Future Enhancements**
   Adopt Transformer-Based Models
   Explore advanced deep learning architectures like the **TimeSeriesTransformer**, which can better capture long-term dependencies and patterns in sequential data.

   Enhance prediction accuracy by incorporating **news sentiment analysis**, **macro-economic indicators**, or **sector-specific trends**, which can provide deeper insights into stock price movements.

Conclusion:

In this project, I built models to predict the next day's stock closing price using past data. I tried **Random Forest**, **XGBoost**, and **LSTM** models.

- **LSTM** gave the best results because it understands time patterns well.
- **Random Forest** was fast and easy to use, giving decent accuracy.
- **XGBoost** worked well but didn't beat LSTM.