Stock Model Final Report

Name: Anujay Kalbhor  
Roll no: 250240325012

# 1. Objective

To predict stock closing prices using historical data with three models:  
- LSTM (Deep Learning)  
- Random Forest (Ensemble Decision Trees)  
- XGBoost (Boosted Trees)  
  
The goal is to identify which model performs best based on accuracy metrics: RMSE, MAE, and R².

# 2. Dataset Overview

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

# 3. Exploratory Data Analysis (EDA)

## 3.1 Yearly Closing Price Analysis

As part of the exploratory analysis, yearly closing price trends were examined for each stock:  
  
- Date Processing: Converted Date column to datetime format and sorted chronologically  
- Year Extraction: Added Year column for temporal grouping  
- Year-end Analysis: Extracted the last trading day closing price for each stock per year using groupby(['Symbol', 'Year']).tail(1)  
- Visualization: Created individual time series plots for each stock showing year-end closing prices with the following specifications:  
 - Plot size: 8x5 inches  
 - Marker style: Circle markers with teal color  
 - Grid enabled for better readability  
 - Individual plots per stock symbol for clear comparison  
  
This analysis provided insights into long-term price trends and helped identify stocks with consistent growth patterns versus those with high volatility across years.

# 4. Data Preprocessing & Feature Engineering

## 4.1 Cleaning Steps:

- Sorted data by Symbol and Date  
- Removed rows with missing values after creating lagged features

## 4.2 Engineered Features:

- Close\_lag\_1, Close\_lag\_3, Close\_lag\_7  
- Daily\_Return = percent change in Close  
- Rolling\_Mean\_5, Rolling\_Std\_5 = 5-day moving statistics  
- Volatility\_10 = 10-day rolling std dev  
- Volume\_Change = percent change in Volume  
  
These features were used for Random Forest and XGBoost models. For LSTM, the multivariate input sequence included Open, High, Low, Close, and Volume.

# 5. Model Architectures & Training

## 5.1 LSTM (Keras)

- Input: 100 timesteps, 5 features (Open, High, Low, Close, Volume)  
- Architecture:  
 - LSTM (50, return\_sequences=True)  
 - LSTM (50, return\_sequences=True)  
 - LSTM (50)  
 - Dense (1)  
- Loss: Mean Squared Error  
- Optimizer: Adam  
- Epochs: 100

## 5.2 Random Forest

Used RandomizedSearchCV for tuning:  
- n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf  
  
Input: Engineered tabular features

## 5.3 XGBoost

Used RandomizedSearchCV for tuning:  
- n\_estimators, max\_depth, learning\_rate, subsample, colsample\_bytree  
  
Input: Same as RF

# 6. Evaluation Metrics

Used the following:  
- RMSE: Root Mean Squared Error  
- MAE: Mean Absolute Error  
- R² Score: Proportion of variance explained by the model

# 7. Results Summary

|  |  |  |
| --- | --- | --- |
| Model | RMSE | R² Score |
| LSTM (Train) | ~0.005 | 0.99 |
| LSTM (Test) | ~0.005 | 0.98 |
| Random Forest | ~21.0 | 0.22 |
| XGBoost | ~22.0 | 0.20 |

# 8. Insights & Conclusions

- LSTM outperformed both tree-based models significantly with an R² of 0.98 on the test set  
- Random Forest and XGBoost suffered from low variance and underfitting. Even after tuning, their R² remained close to 0.2, indicating poor fit  
- Initial flat-line predictions were resolved by introducing meaningful, non-redundant features  
- The yearly EDA analysis revealed distinct trend patterns across different stocks, which informed the modeling approach

# 9. Challenges Faced

- Redundant lag features leading to flat predictions (fixed via advanced lags & rolling stats)  
- High correlation between original features (Open, High, Low)  
- Need for per-stock model handling due to differing data lengths  
- Varying yearly trends across stocks required careful consideration in feature engineering

# 10. Future Enhancements

- Use transformer-based models (e.g., TimeSeriesTransformer)  
- Add news sentiment, macro-economic indicators  
- Use cross-validation and Shapley feature importance for interpretability  
- Incorporate seasonal decomposition based on yearly trend analysis  
- Implement stock-specific models based on EDA findings

# Conclusion

LSTM is the most reliable model for this sequence-based problem. Tree-based models can serve as benchmarks but need deeper feature engineering or different architectures to compete. The yearly EDA analysis provided valuable insights into long-term stock behavior patterns that can inform future modeling strategies.