Stock Price Prediction

Machine Learning Engineer Assignment

1. Introduction

The objective of this project is to develop a predictive model for forecasting stock price movements using historical financial data. The model is expected to predict the next day's closing price for a given stock.

2. Dataset Analysis

The dataset used in this project is sourced from Kaggle, containing stock market data for different stocks in CSV format. Each CSV file represents historical price data for a specific stock, including the following features:

- Date
- Open
- High
- Low
- Close
- Adjusted Close
- Volume

3. Data Preprocessing

- The structure of each CSV file was checked to ensure the presence of expected columns.
- Data was sorted by date to maintain chronological order.
- Missing values were identified and handled appropriately to ensure data consistency.
- Any missing values in technical indicators were handled by dropping NaN rows after feature computation.

4. Feature Engineering

Technical Indicators Used

The following technical indicators were added to enhance the predictive capability of the model:

1. **Simple Moving Average (SMA)** : 10-day moving average of closing prices.
2. **Exponential Moving Average (EMA)** : 10-day exponentially weighted moving average.
3. **Volatility** : Standard deviation of closing prices over a 10-day window.
4. **Relative Strength Index (RSI)** : Momentum indicator measuring the speed and change of price movements.
5. **Moving Average Convergence Divergence (MACD)** : Difference between a fast EMA and a slow EMA.

6. **MACD Signal Line** : Signal line used for MACD crossover strategies.
7. **Bollinger Bands (High & Low)** : Measures price volatility and relative levels.
8. **Rolling Median & Rolling Standard Deviation** : Helps detect trends and volatility.
9. **Quartiles & Z-Score** : Used for statistical anomaly detection.
10. **Momentum & Rate of Change (ROC)** : Indicates price momentum over a specific period.

These features were selected based on their widespread use in technical analysis for stock price forecasting.

5. Model Development

Selected Models

- LSTM (Long Short-Term Memory Networks)
- XGBoost (Extreme Gradient Boosting)

LSTM Model Architecture:

The LSTM model was implemented using TensorFlow/Keras and includes:

- **Bidirectional LSTM Layers**: Two bidirectional LSTM layers for capturing both forward and backward temporal dependencies.
- **Batch Normalization**: Applied after the first LSTM layer to stabilize learning.
- **Dropout Layer**: Used to prevent overfitting.
- **Dense Layers**: Fully connected layers with ReLU activation for feature extraction.
- **Adam Optimizer**: Chosen for efficient training.

The model was trained using an 80%-20% train-test split.

XGBoost Model Architecture

The XGBoost model was implemented with the following configurations:

- **Tree-based boosting model** to capture non-linear patterns in stock prices.
- **Feature scaling using MinMaxScaler** for better model convergence.
- **Key Features Used**:
  - Price-based features (SMA, EMA, Bollinger Bands, MACD, etc.)
  - Statistical features (Rolling Median, Rolling Std, Z-Score, Quartiles)
  - Momentum-based indicators (ROC, Momentum, Previous Close values)
- **Hyperparameter tuning** included optimization of tree depth, learning rate, and subsampling.

6. Model Training and Evaluation

Performance Metrics Used

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- R-squared (R^2)

Example results:

LSTM Results:

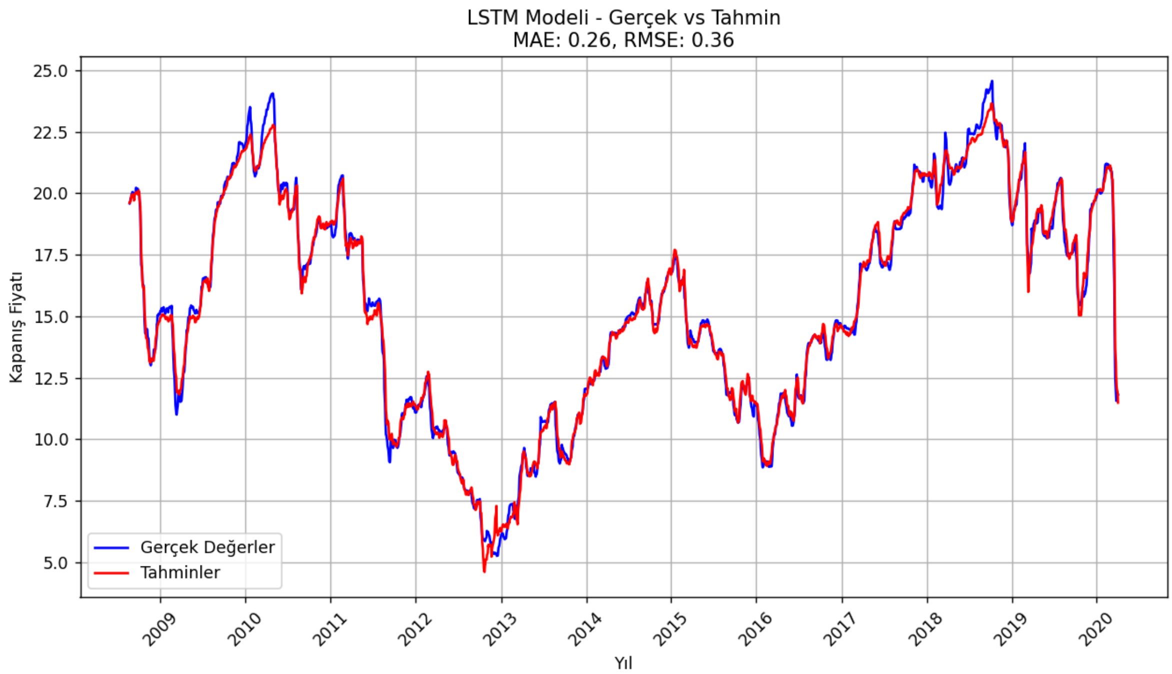

Figure 1 – LSTM Stock Prediction Graph

- **MAE**: 0.26
- **RMSE**: 0.36
- **R^2**: 0.98

XGBoost Results:

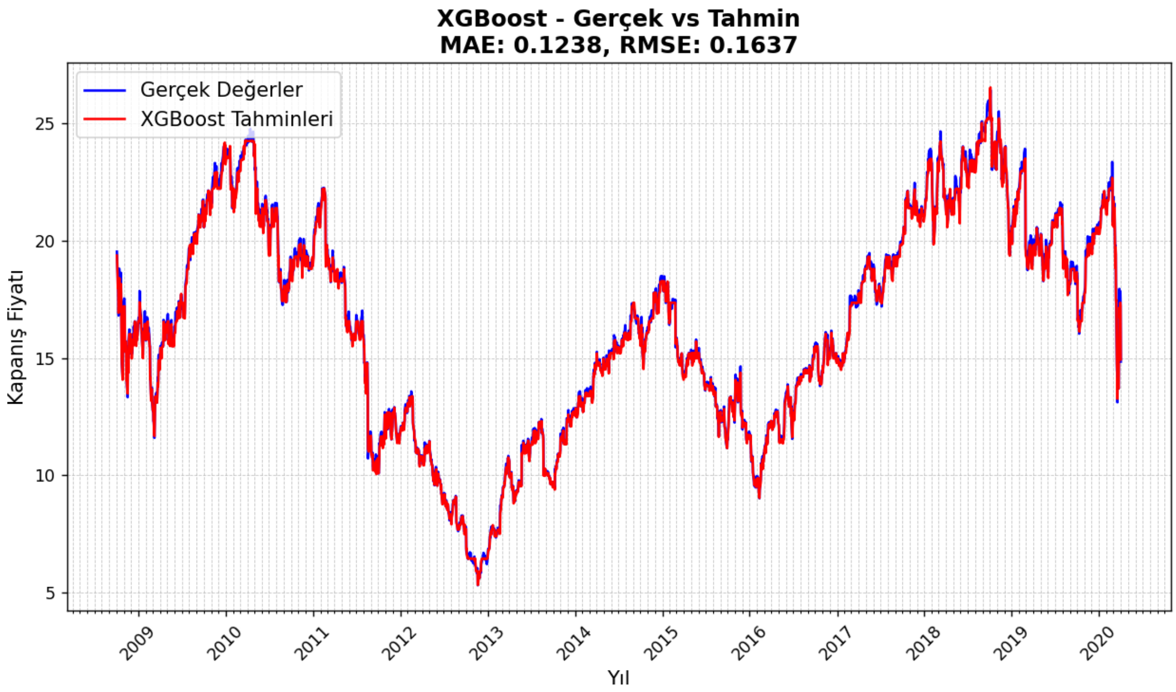

Figure 2 – XGBoost Stock Prediction Graph

- **MAE**: 0.1238
- **RMSE**: 0.1637
- **R^2**: 0.99

7. Conclusion and Future Improvements

- LSTM showed promising results but struggled with highly volatile stocks.
- XGBoost performed well on structured statistical features.
- Some stocks achieved high accuracy, while others did not generalize well.