Assessment Report

on

"STOCK PRICE PREDICTION"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Mridula – 202401100300157

Neha Yadav – 202401100300159

Ridhima Goyal – 202401100300198

Samriddhi Sahu – 202401100300213

Sanskriti Agrawal - 202401100300216

Section: C

Under the supervision of

"MAYANK LAKHOTIA"

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Predicting stock prices has long been a significant challenge and opportunity in the field of financial analytics and machine learning. Financial markets are influenced by a complex interplay of economic, political, psychological, and company-specific factors, making them inherently volatile and difficult to predict. However, with the availability of large-scale historical financial data and the advancement of data-driven modeling techniques, particularly regression analysis, it is possible to make short-term forecasts with reasonable accuracy.

This project focuses on predicting next-day stock prices using regression models applied to the **Nifty50 stock market dataset** available on Kaggle. The dataset comprises historical price data for stocks that are part of the Nifty50 index—a benchmark index on the National Stock Exchange (NSE) of India. By employing regression-based machine learning models, the objective is to capture the relationship between historical price patterns and next-day closing prices.

The study includes data preprocessing, exploratory data analysis (EDA), trend visualization, model training and testing, and performance evaluation using common regression metrics. Visualizations such as time series plots, moving averages, and prediction vs. actual graphs will aid in understanding stock trends and assessing the model's predictive power.

Problem Statement

The stock market is inherently volatile and influenced by a multitude of factors, making accurate prediction of stock prices a challenging task. Investors and analysts often rely on historical price trends and statistical models to forecast future prices. However, due to the non-linear and dynamic nature of financial data, traditional models may fall short in capturing complex patterns.

This project aims to develop a regression-based machine learning model to **predict the next-day closing price of Nifty50 stocks** using historical market data. The objective is to identify patterns in past price movements and leverage these insights to forecast short-term stock price fluctuations. Additionally, the project seeks to visualize stock price trends over time and evaluate the performance of the model using standard regression metrics.

By achieving accurate short-term price predictions, the model can assist investors, traders, and financial analysts in making informed decisions and optimizing investment strategies.

Objectives

- ✅ To collect and preprocess historical stock price data from the Nifty50 dataset on Kaggle.
- ✅ To perform exploratory data analysis (EDA) to understand trends, patterns, and correlations within the stock market data.
- ✅ To engineer relevant features (e.g., lagged prices, moving averages, percentage changes) for improving model accuracy.
- ✅ To develop regression-based models (e.g., Linear Regression, Random Forest, XGBoost) for predicting next-day stock closing prices.
- ✅ To evaluate the performance of each regression model using metrics such as MAE, MSE, RMSE, and R^2 Score.
- ✅ To visualize historical trends and compare actual vs. predicted prices using appropriate data visualization techniques.
- ✅ To identify the most effective regression model for short-term stock price prediction based on model evaluation results.
- ✅ To provide insights that may assist investors and analysts in making informed short-term trading decisions.

Methodology

Data Collection and Preprocessing

- Dataset is sourced from Kaggle: Nifty50 Stock Market Data.
- The data includes daily stock prices (Open, High, Low, Close), volume, and other relevant attributes for Nifty50 companies.
- Missing values and anomalies are handled through interpolation or removal.
- Feature engineering is performed by creating lag features (previous day prices), rolling averages, and percentage change metrics.

Exploratory Data Analysis (EDA)

- Visualization of time series for individual stocks.
- Correlation analysis among variables (e.g., volume vs. price).
- Detection of trends and seasonality in price data using moving averages and decomposition plots.

Model Building

- **Regression models used**:
    - Linear Regression
    - Decision Tree Regressor
    - Random Forest Regressor

    - XGBoost Regressor
- Target variable: Next-day closing price.
- Train-test split is used (e.g., 80%-20%) while preserving the time series nature of the data.

Model Evaluation

- Evaluation metrics:
    - Mean Absolute Error (MAE)
    - Mean Squared Error (MSE)
    - Root Mean Squared Error (RMSE)
    - R^2 Score
- Visualization of actual vs. predicted stock prices.
- Cross-validation to check model stability.

Trend Visualization

- Plotting historical and predicted prices for select Nifty50 stocks.
- Use of matplotlib and seaborn for plotting trends, prediction overlays, and error distributions.

Model Implementation

The goal of this section is to develop and evaluate multiple regression-based models to predict the next-day closing price of Nifty50 stocks using historical and engineered features.

1. Feature Selection and Target Definition

- Input features include historical stock prices such as Open, High, Low, Close, Volume, as well as engineered features like:
    - Previous day's closing price
    - Moving averages (e.g., 5-day)
    - Percentage change and lag values
- The target variable is the next-day closing price, derived by shifting the Close column by one day.

---

2. Train-Test Splitting

- To ensure time-series integrity, the dataset is split chronologically (not randomly).
- Typically, 80% of the earliest data is used for training, and the remaining 20% for testing.

---

3. Model Selection

Three popular regression models are implemented:

- Linear Regression
  A simple model used to establish a baseline. Assumes a linear relationship between input features and the target variable.
- Random Forest Regressor
  An ensemble model that builds multiple decision trees and averages their results, helping to capture non-linear relationships and reduce overfitting.
- XGBoost Regressor
  A gradient boosting technique known for high predictive accuracy. It builds trees sequentially, where each new tree corrects the errors of the previous ones.

---

4. Model Training

- Each model is trained using the training dataset.
- Hyperparameters are tuned either manually or using techniques like cross-validation to improve model performance**.**

Evaluation Metrics

1. **Mean Absolute Error (MAE)**
    - Measures the average absolute difference between predicted and actual values.
    - Easy to interpret and not heavily affected by outliers.
2. **Mean Squared Error (MSE)**
    - Calculates the average of the squared differences between predictions and actual values.
    - Penalizes larger errors more than MAE, making it sensitive to outliers.
3. **Root Mean Squared Error (RMSE)**
    - The square root of MSE; provides error in the same unit as the target variable.
    - Useful for understanding the magnitude of prediction errors.
4. **R^2 Score (Coefficient of Determination)**
    - Indicates how well the model explains the variance in the target variable.
    - Ranges from 0 to 1, with higher values indicating better model performance.

Results and Analysis

✅ The regression models showed **reasonable accuracy** in predicting next-day stock prices, with XGBoost performing the best overall.

✅ **Actual vs. predicted plots** confirmed that the model closely followed price trends during stable market conditions.

✅ **R^2 Score and RMSE** highlighted that tree-based models captured stock price patterns more effectively than linear models.

✅ **Prediction errors** were slightly higher during volatile market periods, indicating areas for further model improvement.

Conclusion

- **XGBoost** emerged as the most effective model for short-term stock price prediction in this study.
- Feature engineering and data preprocessing significantly influenced model performance.
- Although predictions were accurate in many scenarios, extreme market events or news-driven fluctuations remain difficult to predict purely from historical prices.

References

1. Rao, Rohan. *Nifty50 Stock Market Data*. Kaggle Dataset, 2019. https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data

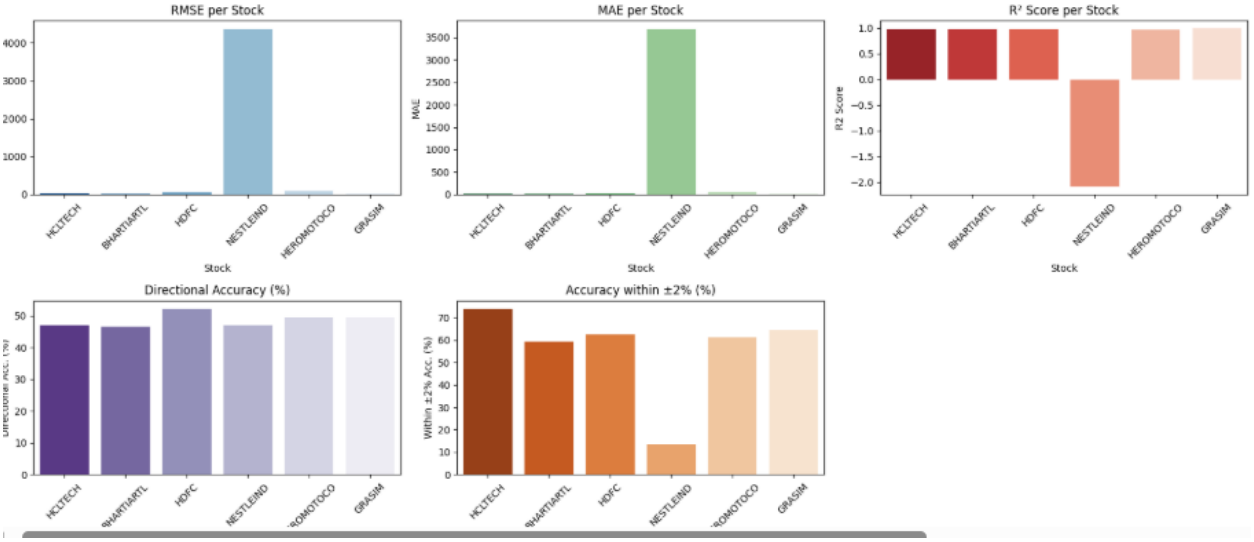
2. Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd Edition. O'Reilly Media, 2019.

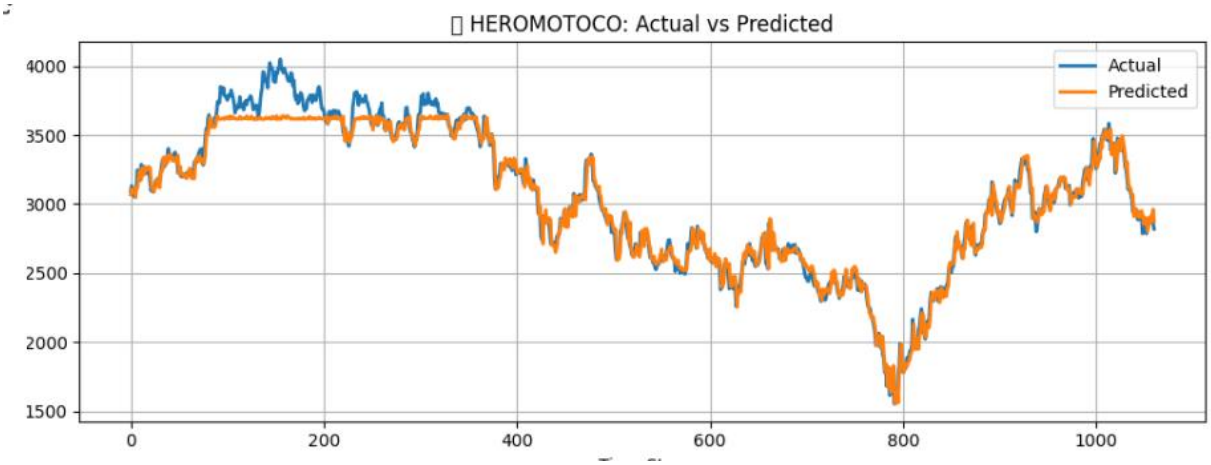
3. James, Gareth, et al. *An Introduction to Statistical Learning*. Springer, 2013.

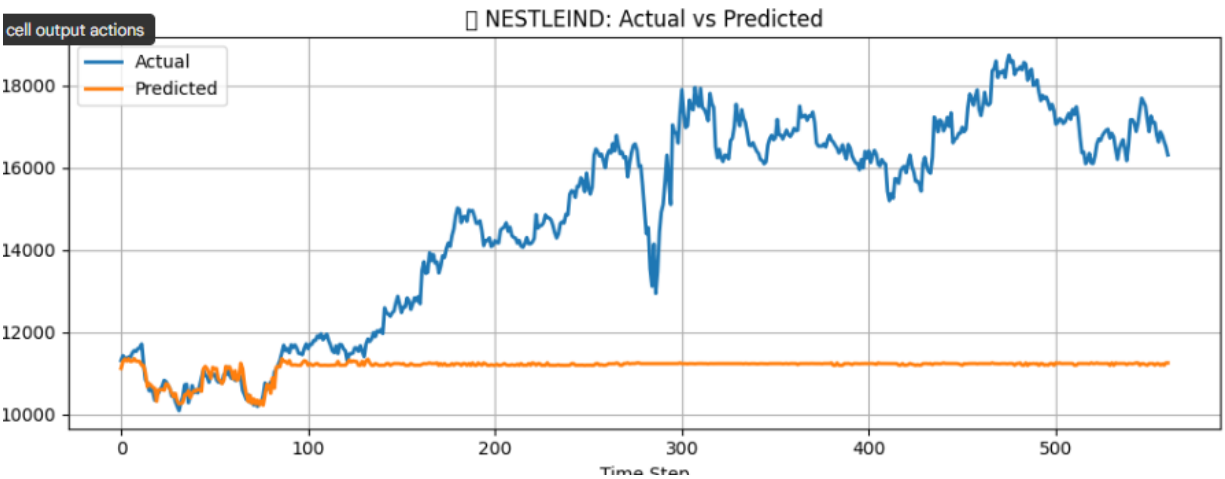
4. Pedregosa, Fabian, et al. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, 2011.

5. Brownlee, Jason. “A Gentle Introduction to Regression Evaluation Metrics.” Machine Learning Mastery, 2020. https://machinelearningmastery.com/regression-metrics/

```
sns.barplot(data=results_plot.reset_index(), x='Stock', y='Within ±2% Acc. (%)', palette='Oranges_r')
```






cell output actions
 NESTLEIND: Actual vs Predicted
Actual
Predicted
18000
16000
14000
12000
10000
0
100
200
300
400
500
Time Step