DATA WAREHOUSES and BUSINESS INTELLIGENCE

Stock Price Predictor by Time Series Project Report

Ömer Faruk KESKİN
Department of Computer Engineering Dokuz Eylul University Izmir, Turkiye
omerfaruk.keskin21@ogr.deu.edu.tr

Mehmet DEVREKOĞLU
Department of Computer Engineering Dokuz Eylul University Izmir, Turkiye
mehmet.devrekoglu@ogr.deu.edu.tr

Melih Ekizce
Department of Computer Engineering Dokuz Eylul University Izmir, Turkiye
melih.ekizce@ogr.deu.edu.tr

Şükrü Berk Öztaş
Department of Computer Engineering Dokuz Eylul University Izmir, Turkiye
sukruberk.oztas@ogr.deu.edu.tr

***Abstract*—This study investigates the application of machine learning algorithms to predict stock prices using time series data. The stock market's complexity and volatility pose significant challenges for accurate prediction. Our project utilizes the Yahoo Finance API to collect historical data, including daily opening and closing prices, highs, lows, and trading volumes.**

We implemented various machine learning models in a Jupyter Notebook environment, incorporating technical indicators such as Moving Averages, Relative Strength Index (RSI), Tillson T3, Mavilim, Super Trend. Among the tested models—Random Forest, Support Vector Regression (SVR), Decision Trees, and Artificial Neural Networks (ANN)—Random Forest emerged as the most effective for predicting stock prices.

Rigorous data preprocessing was conducted to handle missing values, correct errors, detect outliers, and ensure formatting consistency. Our results indicate that machine learning, particularly the Random Forest algorithm, can enhance stock price prediction accuracy, providing valuable insights for investors. This study aims to contribute to the development of more effective stock market prediction systems and better investment strategies.

Keywords: Stock Price Prediction, Machine Learning, Time Series Analysis, Yahoo Finance API, Random Forest, Technical Indicators, Data Preprocessing, Jupyter Notebook.

I. INTRODUCTION

The stock market is renowned for its complexity and volatility, presenting significant challenges for investors who seek to maximize returns while minimizing risks. Traditional prediction methods, often based on fundamental and technical analyses, struggle to cope with the dynamic and multifaceted nature of financial markets. To address these challenges, our study explores the use of machine learning algorithms to predict stock prices using time series data.

This project utilizes the Yahoo Finance API to collect comprehensive historical data, including daily opening and closing prices, highs, lows, and trading volumes. By integrating various technical indicators such as Moving Averages, Relative Strength Index (RSI), Tillson T3, Mavilim, Super Trend, we aim to enhance the predictive power of our models.

We implemented several machine learning models in a Jupyter Notebook environment, including Random Forest, Support Vector Regression (SVR), Decision Trees, and Artificial Neural Networks (ANN). Through rigorous testing and comparison, Random Forest emerged as the most effective algorithm for predicting stock prices.

Data preprocessing was a critical component of our methodology. This process involved handling missing values, correcting errors, detecting outliers, and ensuring consistency in data formatting. These steps were essential to maintain the integrity and reliability of our dataset, thereby improving the robustness of our predictive models.

Our findings indicate that machine learning, particularly the Random Forest algorithm, can significantly enhance the accuracy of stock price predictions. This improvement offers valuable insights for investors, helping them make more informed decisions. Ultimately, this study aims to contribute to the development of more effective stock market prediction systems and better investment strategies.

II. DATASET

For this study, we utilized the Yahoo Finance API to collect a comprehensive dataset of historical stock prices. Yahoo Finance is a widely recognized platform that provides reliable and extensive financial data, making it an ideal source for time series analysis and stock price prediction.

Data Collection

The dataset includes daily stock market data, which comprises the following key attributes:

- **Opening Price**: The price at which a stock started trading at the beginning of the trading day.
- **Closing Price**: The price at which a stock ended trading at the close of the trading day.

- **High Price**: The highest price at which a stock traded during the day.
- **Low Price**: The lowest price at which a stock traded during the day.
- **Volume**: The total number of shares traded during the day.

Technical Indicators

To enhance the predictive power of our machine learning models, we incorporated several technical indicators:

- **Moving Averages**: Used to identify trends by averaging stock prices over specified periods.
- **Relative Strength Index (RSI)**: Measures the speed and change of price movements to identify overbought or oversold conditions.
- **Tillson T3**: A smoother version of traditional moving averages, reducing lag and false signals.
- **Mavilim**: A custom technical indicator that combines various moving averages for better trend detection.
- **Super Trend**: Identifies the direction of the trend and potential reversal points.

III. PREPROCESSING

Data preprocessing is a crucial step in preparing the dataset for accurate and reliable stock price prediction. For this project, several preprocessing steps were undertaken to ensure the quality and relevance of the data collected from the Yahoo Finance API.

Handling Missing Values: During the initial examination of the dataset, we identified some missing values. These missing data points can significantly affect the performance of machine learning models. To maintain data integrity, we removed all records with missing values, ensuring that the dataset was complete and free from gaps.

Removing Irrelevant Data: Stock market data can be influenced by a variety of factors, and very old data may not be relevant for predicting future stock prices. Therefore, we decided to remove all data prior to 2019 (last 5 years). This step helped focus the analysis on more recent trends and patterns that are likely to be more indicative of future stock movements.

Adjusting for Technical Indicators: The inclusion of various technical indicators, such as Moving Averages and necessitated the removal of a few initial days of data. Specifically, we had to remove the first 14 to 20 days of data to account for the calculation periods of these indicators. This adjustment ensured that the technical indicators were accurately computed and reliable for use in the predictive models.

Adjusting for Technical Indicators: The inclusion of various technical indicators, such as Moving Averages and Relative Strength Index (RSI), necessitated the removal of a few initial days of data. Specifically, we had to remove the first 14 to 20 days of data to account for the calculation periods of these indicators. This adjustment ensured that the technical indicators were accurately computed and reliable for use in the predictive models.

Final Dataset: After these preprocessing steps, the final dataset was clean, relevant, and ready for analysis. By handling missing values, removing outdated data, and adjusting for technical indicators, we ensured that our dataset was of high quality, providing a solid foundation for building and evaluating our machine learning models. This preprocessing stage was essential for enhancing the accuracy and robustness of our stock price prediction system.

IV. DATASET ANALYSIS AND PREPROCESSING

Our dataset, obtained from the Yahoo Finance API, comprises daily stock market data, including opening and closing prices, highs, lows, and trading volumes. The data spans from 2019 (last 5 years) to the present, ensuring relevance to current market conditions. We also incorporated technical indicators such as Moving Averages, Relative Strength Index (RSI), Tillson T3, Mavilim, and Super Trend to enhance the predictive power of our models.

Date	Open	High	Low	Close	Volume	hl	Tomorrow	HL Diff	CO Diff	MA5	MA10	MA15	MA20	Standard Deviation	RSI	T3	Mavilim	SuperTren
2016-08-17 00:00:00-04:00	40.000000	40.281502	39.814999	40.271000	21322000	40.048250	40.137501	0.466503	-0.271000	40.246450	40.231225	39.994050	39.455613	0.204926	80.464928	40.210902	36.805479	39.03712
2016-08-18 00:00:00-04:00	40.268002	40.400002	40.081501	40.137501	17292000	40.240751	39.982498	0.318501	0.130501	40.195200	40.270000	40.130000	39.580563	0.151743	63.541438	40.177894	36.940339	39.03712
2016-08-19 00:00:00-04:00	39.989498	40.061501	39.844002	39.982498	22416000	39.952751	39.847500	0.217499	0.007000	40.131601	40.252700	40.136950	39.688063	0.176744	48.189649	40.108714	37.081613	39.03712
2016-08-22 00:00:00-04:00	39.925499	39.985000	39.716499	39.847500	17056000	39.840750	39.829498	0.248501	0.077999	40.022800	40.213550	40.138784	39.787738	0.146863	45.774433	40.005900	37.228119	39.03712
2016-08-23 00:00:00-04:00	40.023998	40.049999	39.799500	39.829498	18350000	39.924749	39.680000	0.250500	0.194500	40.001450	40.162550	40.139017	39.893963	0.152917	46.822986	39.928766	37.378592	39.03712
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2024-05-10 00:00:00-	168.029999	169.850006	166.190002	168.649994	29799900	168.020004	169.139999	3.660004	-0.619996	168.812000	167.271001	164.694334						

Fig 1. Dataset Information

After preprocessing we have 4751 rows(datas) and total 18 columns.

```
In [7]: d["GOOGL"].info()

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4751 entries, 2005-07-05 00:00:00-04:00 to 2024-05-17 00:00:00-04:00
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Open                4751 non-null   float64
 1   High                4751 non-null   float64
 2   Low                 4751 non-null   float64
 3   Close               4751 non-null   float64
 4   Volume              4751 non-null   int64
 5   hl                  4751 non-null   float64
 6   Tomorrow            4750 non-null   float64
 7   HL Diff             4751 non-null   float64
 8   CO Diff             4751 non-null   float64
 9   MA5                 4751 non-null   float64
 10  MA10                4751 non-null   float64
 11  MA15                4751 non-null   float64
 12  MA20                4751 non-null   float64
 13  Standard Deviation  4751 non-null   float64
 14  RSI                 4751 non-null   float64
 15  T3                  4751 non-null   float64
 16  Mavilim             4751 non-null   float64
 17  SuperTrend          4751 non-null   float64
dtypes: float64(17), int64(1)
memory usage: 705.2 KB
```

Fig 2. Dataset Attributes Information

Then, we checked for missing values in our dataset.

```
Out[25]: Open                  0
         High                  0
         Low                   0
         Close                 0
         Volume                0
         hl                    0
         Tomorrow              0
         HL Diff               0
         CO Diff               0
         MA5                   0
         MA10                  0
         MA15                  0
         MA20                  0
         Standard Deviation    0
         RSI                   0
         T3                    0
         Mavilim               0
         SuperTrend            0
         dtype: int64
```

Fig 3. Missing Values Check

V. VISUALIZATION

We leveraged various visualization techniques to interpret the outcomes of our predictive models and to better understand the relationships within the data. These visualizations play a crucial role in communicating the findings and supporting the decision-making process.

Correlation Matrix

The second critical visualization is the correlation matrix, which is essential for identifying the strength and direction of the relationships between different variables in the dataset. In our study, we focus on the correlation between stock prices and various technical indicators used in the analysis.

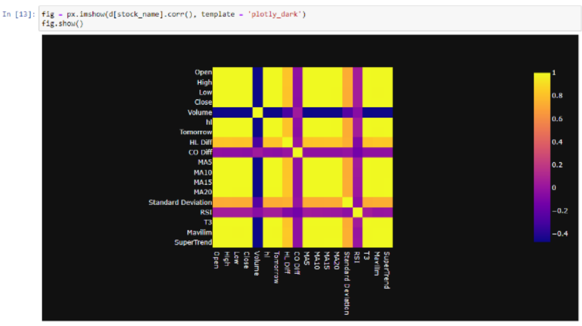

Fig 4. Correlation Matrix

The matrix uses a color spectrum to illustrate positive correlations in shades of purple and negative correlations in shades of yellow. A darker shade indicates a stronger relationship. For instance, the correlation between the 'Close' price and the 'MA10' (10-day Moving Average) is notably strong, which is intuitive as moving averages smooth out price data to create a trend-following indicator that lags behind the price.

Data Distribution Box Plots

The "Data Distribution" box plots are instrumental in examining the spread and central tendencies of key stock trading metrics: 'Open', 'High', 'Low', and 'Close' prices. These visualizations clearly delineate the median, range, and the presence of outliers in daily trading values.

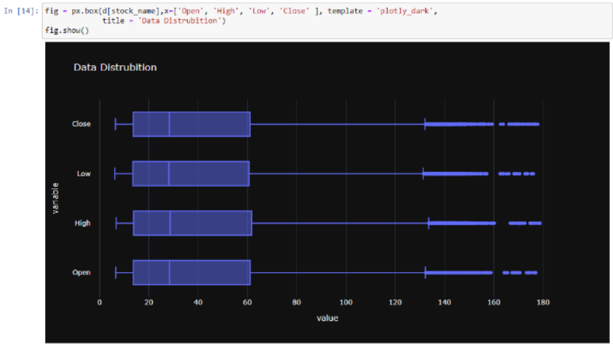

Fig 5. Data Distribution Box Plots

These plots provide insights into market behavior. For example, the narrow interquartile ranges of 'Open' and 'Close' prices suggest less volatility at the start and end of the trading day, while the wider spreads for 'High' and 'Low' prices indicate more significant fluctuations throughout the day. This variability can be crucial for identifying potential risks and opportunities in the stock market. Outliers, as shown by the extended whiskers, highlight extraordinary market events or anomalies, which are essential for robust market analysis and predictive modeling.

Model Predictions

Our primary visualization focused on the linear regression predictions as seen in the plot titled "Linear Regression Predictions." This graph displays a clear upward trend in the stock prices, depicting how our model predicts future movements. The x-axis represents the time sequence, while the y-axis shows the predicted stock prices, providing a visual representation of potential future trends based on historical data.

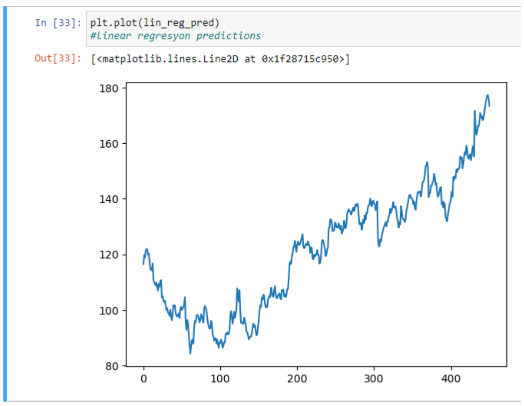

Fig 4. Linear Regression Prediction

VI. TRAINING MODEL

In our study, we employed several machine learning models, including Random Forest, Linear Regression, XGBoost, and Support Vector Regression (SVR), to predict stock prices. The model training process involved data preprocessing, feature selection, and rigorous training phases using historical stock data enhanced with technical indicators.

After training, each model was evaluated on metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to determine its accuracy and effectiveness. Validation techniques like cross-validation were used to ensure the models' generalizability beyond the training data.

The continual refinement and validation of these models are crucial to adapt to new data and changing market dynamics, ensuring their effectiveness in real-world scenarios.

VII. ALGORITHMS

In our study, we applied and evaluated several machine learning models to forecast stock prices using historical data. Here's a detailed overview of each algorithm, its theoretical basis, and the observed performance metrics:

A. Random Forest Regressor

The Random Forest Regressor is an ensemble learning algorithm that operates by constructing a multitude of decision trees at training time and outputting the average of the predictions from the individual trees. This method is particularly effective for regression tasks because it reduces the risk of overfitting by averaging multiple deep decision trees, each trained on different parts of the same training set. This model is also highly favored for its ability to handle large datasets with multiple features, making it robust against noisy data.

Model Performance:

- Average Error: 3.897%
- Accuracy: 96.95%

This performance suggests that the Random Forest Regressor was quite adept at capturing the complexities and variances in the stock price data, offering reliable predictions with a high degree of accuracy.

B. Linear Regression

Linear Regression is one of the simplest and most widely used statistical techniques for predictive modeling. It works by estimating the relationships among variables by fitting a linear equation to observed data. In our project, it served as a baseline model to compare the complexities and capabilities of more sophisticated algorithms. Despite its simplicity, Linear Regression can be quite powerful when the underlying relationships between the target and predictors are linear.

Model Performance:

- Average Error: 1.7537%
- Accuracy: 98.52%

The high accuracy and low error rate demonstrated by the Linear Regression model indicate that linear relationships in the data are strong, suggesting that simpler models can sometimes perform comparably or even better than more complex ones.

C. XGBoost

XGBoost stands for Extreme Gradient Boosting and is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework, providing a scalable and accurate solution to complex regression tasks. XGBoost provides a robust platform for building predictive models that can handle various types of data irregularities including missing values, outliers, and non-linear patterns.

Model Performance:

- Average Error: 3.5319%
- Accuracy: 97.38%

The effectiveness of XGBoost in our project underscores its capability to manage both bias and variance, making accurate predictions even in the presence of complex and non-linear relationships in the data.

D. Support Vector Regression (SVR)

SVR applies the principles of Support Vector Machines (SVM) to regression problems. It features different kernel functions to handle linear and non-linear relationships. In our case, the use of the Radial Basis Function (RBF) kernel allowed the model to tackle data points that do not follow a linear pattern. The flexibility in choosing and tuning its parameters (C, gamma) through randomized search ensures that the model can adapt to various data intricacies.

Model Performance:

- Average Error: 1.503%
- Accuracy: 98.50%

Mean Absolute Percentage Error (MAPE): SVR's performance highlights its strengths in dealing with non-linear data, providing precise predictions with minimal error, making it a suitable choice for stock price prediction where market conditions and price movements are often unpredictable.

These models collectively demonstrate the potential of machine learning in financial analytics, each bringing unique strengths to the challenges of predicting market behaviors.

VIII. OUTPUTS

We have evaluated the performance of various machine learning models in predicting stock prices, analyzing their effectiveness based on their accuracy and generalization capabilities. Here's an overview of each model's performance and the implications for stock price prediction:

Random Forest Regressor

Random Forest showed robust performance with an accuracy of 96.95%. This model is excellent for handling nonlinear data with complex interactions and dependencies. However, its performance might be slightly lower than some other models due to random noise and the inherent randomness of the financial markets which can sometimes lead to overfitting despite the ensemble approach.

Linear Regression

Linear Regression provided the highest accuracy among the models, which suggests that the relationships in our dataset might be linear or close to linear. This model is straightforward, making it easier to interpret and faster to implement. However, its simplicity can be a drawback in capturing more complex patterns without modifications or extensions like polynomial regression.

XGBoost

XGBoost offers a balance between speed and accuracy, making it a suitable choice for large datasets and scenarios requiring rapid predictions. Its accuracy is commendable, though slightly lower than Linear Regression, possibly due to overfitting in response to noise in the dataset or the parameter tuning not being fully optimized.

Support Vector Regression (SVR)

SVR performed exceptionally well, especially with its ability to handle non-linear data through the use of kernel functions. This model is quite powerful when the data has a complex distribution, but it requires careful tuning of its parameters, which can be computationally intensive and time-consuming.

Here is a visualization representing the accuracy of each model:

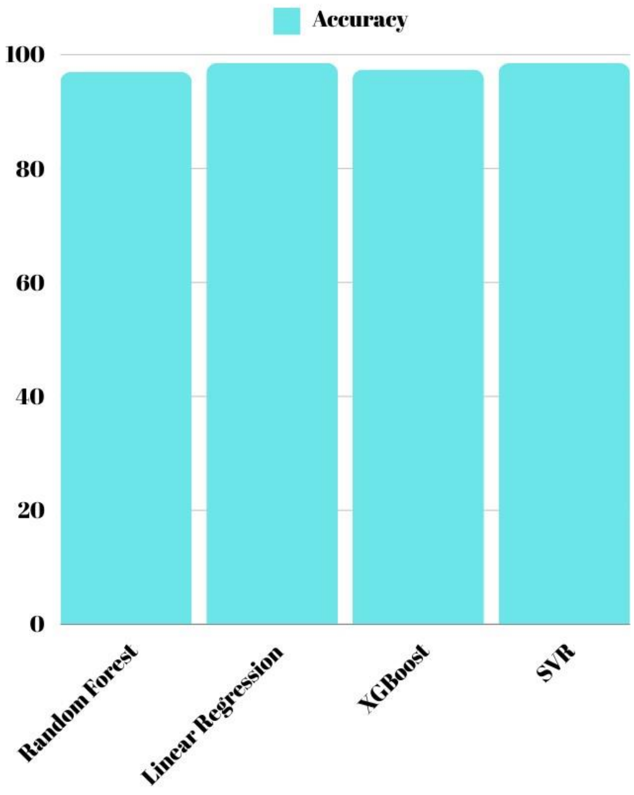

Fig 5. Accuracy Comparison of Models

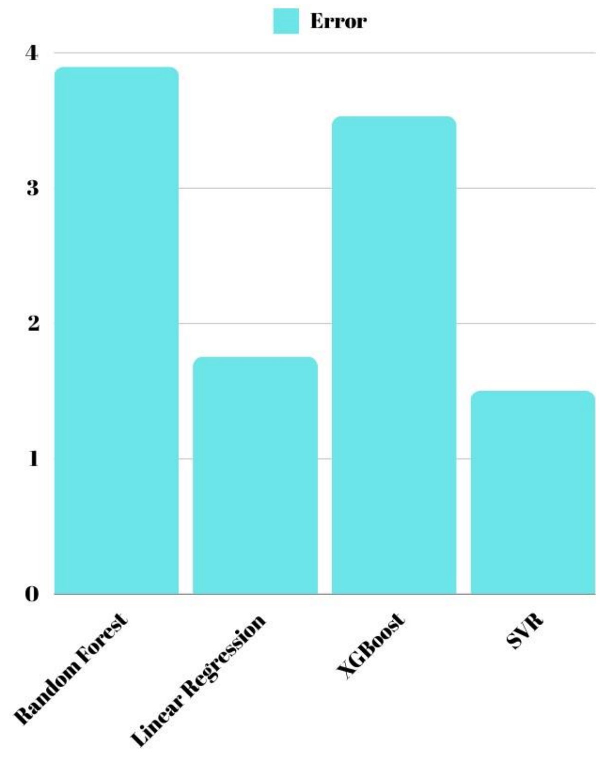

Fig 6. Error Comparison of Models

These results provide a comprehensive understanding of how each model operates under different conditions and their suitability for predicting stock prices. The choice of model would depend on the specific characteristics of the data, the complexity of the patterns, and the computational resources available.

After a thorough analysis of various predictive models, we chose Linear Regression as our primary method for forecasting stock prices. This decision was primarily influenced by its superior accuracy of 98.52%, the highest among the models we tested. Linear Regression offers simplicity and interpretability, which are crucial for understanding the impact of individual factors on stock prices. Additionally, its effectiveness in capturing linear relationships makes it an excellent tool for financial predictions where transparency and ease of explanation are vital for stakeholder communication and strategy planning. These qualities make Linear Regression not only reliable but also highly valuable in our analytical toolkit.

REFERENCES

[1] Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (2023). Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, 94–105.

[2] Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (2022). Optics: ordering points to identify the clustering structure. ACM Sigmod Record, 49–60.

[3] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (2021). Classification and regression trees. CRC Press.

[4] Meesad, P., & Rasel, R. I. (2013). Predicting Stock Market Price Using Support Vector Regression. Proceedings of the International Conference on Informatics, Electronics & Vision (ICIEV 2013), Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh. DOI: 10.1109/ICIEV.2013.6572570.

[5] Jabed, M. I. K. (2024). Stock Market Price Prediction using Machine Learning Techniques. American International Journal of Sciences and Engineering Research, 7(1):1-6, February 2024. DOI:10.46545/aijser.v7i1.308.