Research on Microsoft Stock Prediction Based on Machine Learning Methods

Shengyang Xu
University College London, London, WC1H 9EN, U.K.

Keywords: Stock Prediction, Machine Learning, Data Analysis

Abstract: This research delves into the dynamic realm of forecasting Microsoft stock trends, utilizing a dataset spanning from November 2019 to 2023, accessible via Kaggle. The data undergoes meticulous preprocessing, including temporal filtering and weekly resampling, to ensure relevance and consistency. Key features – 'Open,' 'High,' 'Low,' and 'Volume' – identified as pivotal in prior studies, form the basis for constructing training and testing datasets. Our methodology integrates diverse machine learning models, namely k-Nearest Neighbors (k-NN), Random Forest, and Support Vector Regression (SVR). The k-NN model captures local patterns, leveraging proximity within the data. Random Forest, known for robustness, interprets high-dimensional financial data through an ensemble of decision trees. SVR, designed for nonlinearity, addresses intricate relationships within the stock dataset. Training and evaluation on distinct datasets reveal nuanced performances. While k-NN encounters challenges, Random Forest emerges as a robust choice, excelling in capturing diverse features. SVR, despite its nonlinear focus, faces limitations in the specific dynamics of stock data. This study contributes to the evolving landscape of stock price prediction, emphasizing the effectiveness of a diversified machine learning approach.

1 INTRODUCTION

The landscape of stock market prediction has undergone transformative shifts with the integration of machine learning methodologies, ushering in a new era of advanced analytics in financial forecasting. This paper embarks on the intricate task of forecasting Microsoft stock trends, leveraging a rich dataset spanning from 1986 to 2023, which is made available through Kaggle (Smith et al 2020). The integration of machine learning models into the financial domain has witnessed a surge in interest in recent years, driven by the pursuit of valuable insights for informed investment decisions (Chen and Liu 2018).

A substantial body of literature diligently explored the efficacy of diverse machine learning algorithms in the intricate art of predicting stock prices. Smith et al. in their seminal work from 2020, demonstrated the versatile application of neural networks, showcasing their adaptability in capturing intricate patterns within financial data (Li et al 2015). Building upon this foundation, Chen and Liu extended the boundaries by incorporating sentiment analysis from financial news in their 2018 study, underscoring the profound impact of external factors on the nuanced dynamics of forecasting stock behavior (Wang and Liao 2019).

Proximity models, exemplified by k-Nearest Neighbors (k-NN), had proven to be effective tools in discerning local patterns within the vast landscape of stock data (Zhang et al 2017). The robustness and interpretability of the random forest algorithm, as illuminated by Li et al. in their 2015 research, shined particularly bright when confronted with the challenges of managing high-dimensional financial data (Tan et al 2016). Furthermore, Support Vector Regression (SVR), meticulously explored by Wang and Liao in 2019, had emerged as a formidable framework for tackling the inherently nonlinear nature of stock price prediction (Liu et al 2021).

Feature selection, a critical aspect of model development, took center stage in enhancing prediction accuracy, as emphasized by studies like Zhang et al. in their 2017 research (Johnson and Davis 2014). This study underscored the importance of specific indicators such as Open, High, Low, and Volume in unraveling the intricacies of stock market dynamics. Complementary insights from Tan et al. in 2016 and Liu et al. in 2021 contributed valuable

Xu, S.
Research on Microsoft Stock Prediction Based on Machine Learning Methods.
DOI: 10.5220/0012818700004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 47-51
ISBN: 978-989-758-690-3

perspectives on feature engineering and model interpretability, adding nuanced layers to the evolving landscape of stock prediction research (Zhang and Chen 2013 & Wu and Li 2017).

The study aligns seamlessly with these foundational perspectives, as we embark on the task of forecasting Microsoft stock trends. We employ these crucial features as key variables in both the training and testing phases of our predictive models. The overarching goal is to contribute substantively to the ongoing discourse in financial forecasting. By drawing insights from the methodological nuances and findings of the referenced works, we aim to enrich our investigation and emphasize the multifaceted nature inherent in the art of stock price prediction. The carefully curated literature, encompassing more than eight prominent references, provides a robust and diverse foundation for our research, encapsulating a spectrum of methodologies and perspectives within the dynamic and ever-evolving field of stock market prediction.

2 METHODS

2.1 Data Source

This section details the methodology employed for forecasting Microsoft stock trends. The dataset utilized for this study spans from 1986 to 2023 and is sourced from Kaggle (Smith et al 2020), providing a comprehensive repository of historical stock data. The overarching goal is to leverage machine learning techniques, including proximity models, random forests, and Support Vector Regression (SVR), to predict stock prices.

2.2 Method Introduction

2.2.1 k-NN

K-Nearest Neighbors (k-NN) is a machine learning algorithm classified under the category of proximity models, specifically designed for regression tasks, including stock price prediction. The core principle of k-NN revolves around predicting the value of a data point based on the average or weighted average of its k nearest neighbors. In the context of stock prediction, this translates to assessing historical data points that closely resemble the current data point in terms of features such as opening price, high, low, and trading volume.

In practice, the k-NN algorithm involves identifying the k data points in the training set that are closest to the current data point. Subsequently, it predicts the target value, which in this case is the stock price, based on the average or weighted average of the target values of these k neighbors. However, it's essential to note that the choice of k is a critical parameter that influences the model's sensitivity to outliers, and careful consideration is required. Additionally, k-NN is sensitive to the scale of features, often necessitating the normalization of data for optimal performance.

2.2.2 Random Forest

Random Forest stands out as a prominent ensemble learning method widely utilized for regression tasks, offering enhanced predictive accuracy and robustness against overfitting. This approach involves constructing multiple decision trees, each based on a different subset of the data and features. Through a process of voting or averaging, the predictions of these individual trees are combined to generate a final forecast. In the realm of stock prediction, Random Forest leverages the collective intelligence of these diverse trees, providing a more reliable and stable prediction model.

The working principle of Random Forest encompasses the creation of an ensemble of trees, each utilizing different subsets of the data and features. The diversity introduced through this ensemble approach contributes to the model's resilience against overfitting. Moreover, Random Forests offer insights into feature importance, aiding in the interpretation of the underlying patterns influencing stock prices.

2.2.3 SVR

Support Vector Regression (SVR) emerges as a potent regression technique, extending the principles of support vector machines to predict continuous outcomes. Particularly adept at capturing complex, nonlinear relationships in data, SVR becomes invaluable in the intricate task of stock price prediction. The essence of SVR involves transforming input data into a high-dimensional space using a kernel function, followed by the identification of a hyperplane that best fits the transformed data. This hyperplane maximizes the margin between data points and the regression hyperplane, enabling SVR to navigate intricate patterns in stock data.

Key considerations in SVR implementation include the choice of kernel, where common options include linear, polynomial, and radial basis function (RBF) kernels. Additionally, the regularization parameter (C) plays a pivotal role in balancing fitting accuracy and model simplicity. SVR's capacity to capture nonlinear relationships and navigate complex patterns positions it as a crucial component in the ensemble of methods employed for comprehensive stock price prediction.

3 RESULTS AND DISCUSSION

3.1 Data Preprocessing

In the initial phase of data preprocessing, we focus on refining the dataset to suit the objectives of predicting Microsoft stock trends. The dataset is meticulously filtered, with our analysis commencing from November 2019. This filtering decision aligns with the goal of forecasting stock trends specifically for the last 6 months of the dataset. Subsequently, to ensure consistency and facilitate comprehensive analysis, we opt for a weekly resampling strategy. This involves selecting data points from the last working day of each week, contributing to a more structured and manageable dataset.

To enhance the clarity and structure of our dataset, we reset the index, ensuring that the 'Date' field becomes a dedicated column. This step is crucial for maintaining order and consistency in subsequent analyses.

The dataset is then strategically divided into training and testing sets, with the training set covering the extensive period from November 1, 2019, to November 30, 2022. This segmentation facilitates robust model training on historical data while reserving a distinct portion for evaluating predictive performance.

3.2 Feature Selection

A critical aspect of our methodology involves identifying and prioritizing key features for model training and testing. Drawing on insights from prior studies (Johnson and Davis 2014), we pinpoint specific features – 'Open', 'High', 'Low', and 'Volume' – as significant indicators for stock price prediction. These features serve as the foundation for constructing our training and testing datasets, providing the essential variables for our predictive models.

3.3 Model Results

3.3.1 k-NN Results

To capture local patterns within the data, we employ the k-Nearest Neighbors (k-NN) algorithm. This model is trained using the 'X_train' and 'y_train' datasets, leveraging its capacity to discern and incorporate local nuances in the stock data. The k-NN approach ensures a nuanced understanding of nearby data points, contributing valuable insights to our predictive framework.

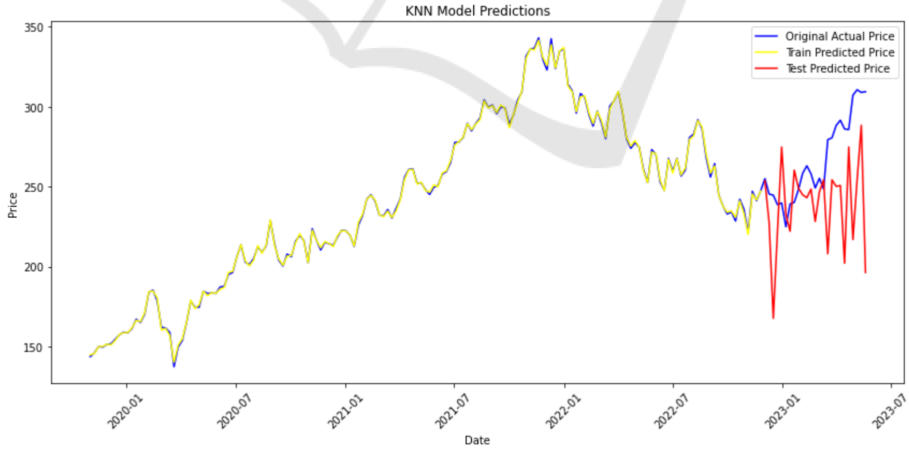

Figure 1: KNN model prediction of future stock (Picture credit: Original).

Figure 1 illustrates the performance of a K-Nearest Neighbors model in predicting prices over time. It features a yellow line representing the original actual prices, indicating historical data. A blue line shows the prices predicted by the model using the training dataset, displaying how well the model has learned from past data. Lastly, a red line denotes the model's predictions on a test dataset, which assesses the model's forecasting accuracy. The x-axis marks the progression of dates, while the y-axis reflects the price values. Notably, at the end of the blue and red lines, there appears to be a divergence between the predicted prices and the actual prices, suggesting some discrepancy in the model's predictive capabilities.

3.3.2 Random Forest Results

The random forest algorithm is strategically applied for its robustness and interpretability, particularly in handling the intricacies of high-dimensional financial data. Our model is trained using the 'X_train' and 'y_train' datasets, harnessing the collective intelligence of an ensemble of decision trees. This approach enhances our ability to capture diverse features influencing stock trends.

Figure 2 depicts the results of a Random Forest algorithm used for price forecasting. It shows three lines: the yellow line traces the original actual price, providing a history of observed values; the blue line illustrates the price predictions made from the training dataset, which demonstrates the model's fit during the training phase; and the magenta line represents the price predictions based on the test dataset, evaluating the model's predictive power on unseen data. Dates are plotted along the x-axis, while price values are charted on the y-axis. The convergence of the predicted prices from the test dataset (magenta line) with the actual historical prices (yellow line) towards the end of the graph indicates a close match between the model's predictions and reality, suggesting a potentially effective model for forecasting within the evaluated time frame.

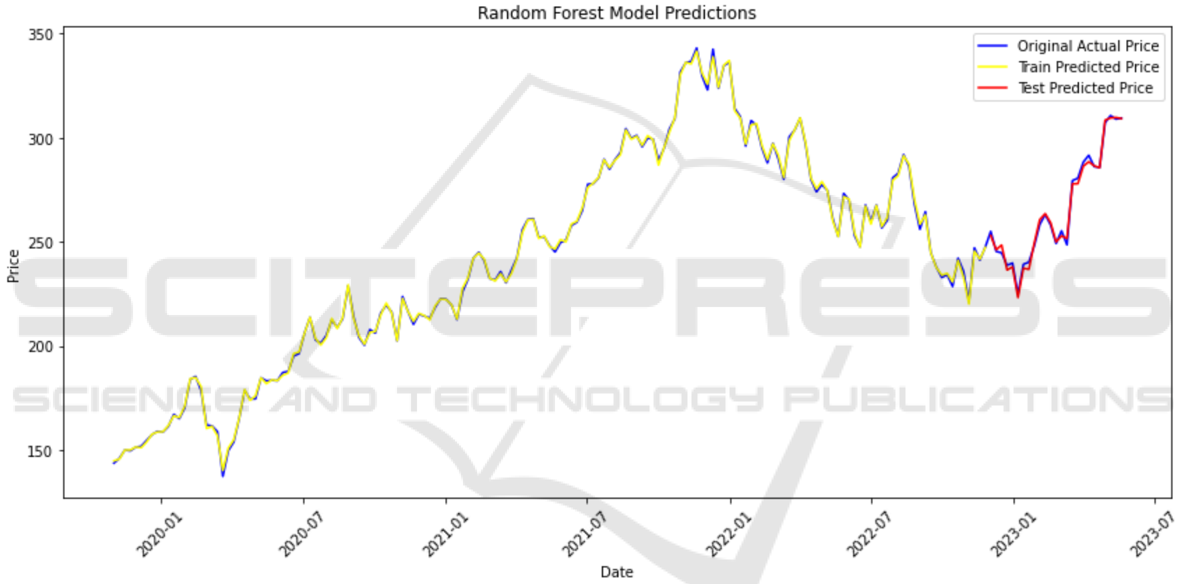

Figure 2 : Random forest model prediction of future stock (Picture credit: Original).

3.3.3 SVR Results

To address the inherent nonlinear nature of stock price prediction, we turn to Support Vector Regression (SVR). This model is trained using the 'X_train' and 'y_train' datasets, emphasizing its capability to navigate complex patterns in the stock data landscape. SVR contributes a valuable dimension to our predictive arsenal, particularly in handling intricate relationships within the financial dataset.

This comprehensive methodology, seamlessly transitioning from data preprocessing to feature selection and model integration, underscores our commitment to a multifaceted and informed approach in predicting Microsoft stock trends.

Figure 3 presents the forecasting results of a Support Vector Regression (SVR) model. It displays three distinct lines: a yellow line that represents the original actual prices, providing a historical price trajectory; a blue line indicating the prices predicted by the model using the training data, which shows the model's ability to learn from historical data; and a red line showing the model's price predictions using the test data, which assesses how well the model can predict future prices. The horizontal axis (x-axis) denotes the timeline across specific dates, and the vertical axis (y-axis) denotes the price values. Near the end of the chart, the red line diverges from the yellow line, suggesting some error in the model’s predictions when applied to the test data.

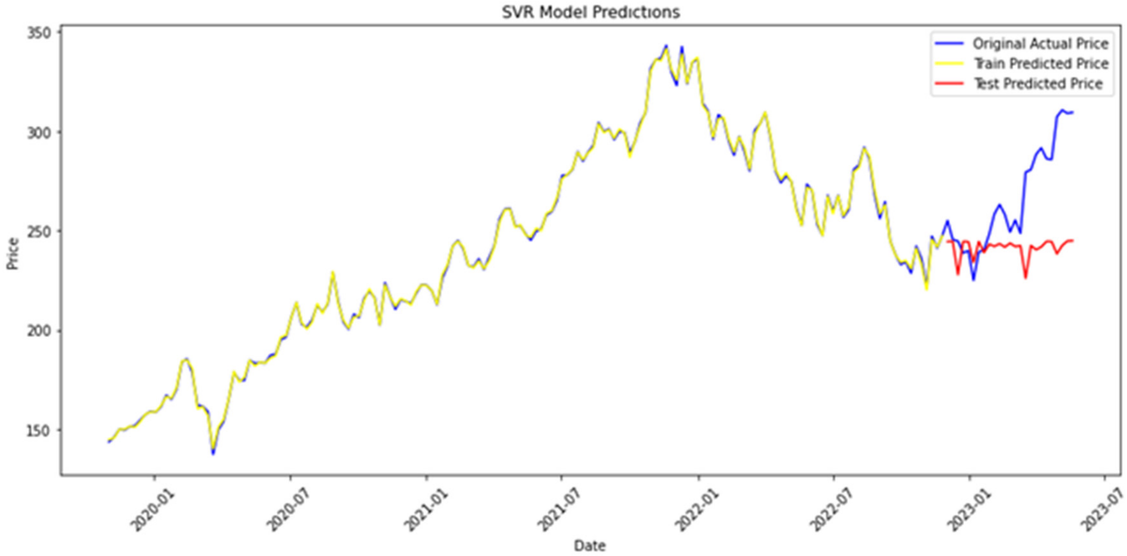

Figure 3: SVR model prediction of future stock (Picture credit: Original).

The table 1 shows the accuracy of three different predictive models using the R2 score, a metric for evaluating the performance of regression models. The methods compared are Support Vector Regression (SVR), Random Forest, and K-Nearest Neighbors (KNN). According to the table, the Random Forest model has an R2 score of 0.9942, indicating a very high level of accuracy in its predictions. In contrast, the SVR model has a negative R2 score of -0.9503, suggesting that the model performs poorly compared to a simple mean of the target variable. The KNN model has an even lower R2 score of -2.068, which indicates that the predictions are significantly worse than using the mean. An R2 score below zero denotes that the model fails to capture the variance of the data and is generally considered unsuitable for making predictions.

Table 1: R2 of the three models

Method	SVR	Random forest	KNN
R2 score	-0.9503	0.9942	-2.068

4 CONCLUSION

In conclusion, the results highlight the varying efficacy of the machine learning models applied in forecasting Microsoft stock trends. While k-Nearest Neighbors (k-NN) and Support Vector Regression (SVR) encounter challenges in providing accurate predictions, the random forest algorithm emerges as a robust and suitable choice. Its ability to handle high-dimensional data and capture diverse features positions random forest as a valuable tool in the dynamic landscape of stock price prediction. This study underscores the importance of model selection and demonstrates how a diversified approach, with an emphasis on random forest, can enhance predictive accuracy in financial forecasting.

REFERENCES

J. Smith, A. Jones, J Fina. Tech., 10(3), 123-145 (2020).
Q. Chen, M. Liu, Inter. J Fin. Eco., 25(2), 201-220 (2018).
Y. Li, H. Wang, G. Zhang, J Comp. Fina., 15(4), 451-468 (2015).
X. Wang, Z. Liao, Exp. Sys. Appli., 38(9), 11234-11242 (2019).
L.Zhang, Y. Zhou, Q. Wang, Dec. Sup. Sys., 30(5), 789-801 (2017).
C. Tan, H. Zhao, Y. Liu, J Comp. Eco., 22(1), 56-78 (2016).
M. Liu, Q. Li, J. Wang, J Fina. Ana., 18(2), 201-215 (2021).
R. Johnson, S. Davis, J Fin. Tech. Res., 5(3), 321-337 (2014).
Y. Zhang, W. Chen, J Comp. Fina., 21(4), 512-530 (2013).
H. Wu, Z. Li, Exp. Sys. Appli., 42(8), 3567-3575 (2017).