Stock Trend Prediction and Analysis Based on Machine Learning

Ailin Sun *

Department of Mathematics, Imperial College London, London, UK

* Corresponding author: ailin.sun19@imperial.ac.uk

Abstract. Stock trend prediction is a critically important activity in the financial sector. Investors aim to gain a better understanding of market dynamics, make wiser investment decisions, and manage investment risks more effectively by forecasting stock price trends. This is the focus of our upcoming research. This paper aims to utilize machine learning methods such as Support Vector Machines (SVM), based on spatial interval maximization strategies, Random Forest, based on multi-decision tree parallel learning strategies, and eXtreme Gradient Boosting (XGBoost), based on classification functions, to devise accurate classification function strategies for predicting and analyzing fluctuations in Dow Jones stock prices. Principal Component Analysis (PCA) will then be used to select some suitable features, identifying key factors with the greatest impact and minimal inter-correlation related to stock prices. Through a combination of various classification models, the research aims to forecast the future rise and fall of Dow Jones stocks. It has been observed that performing feature curation through PCA in advance enhances the predictive accuracy of different machine learning methods. Notably, the combination of PCA and XGBoost achieves the highest predictive rate. This approach provides a robust and scientifically grounded method for future stock price predictions and the formulation of trading strategies.

Keywords: Stock trend prediction, Support Vector Machines (SVM), Random Forest, eXtreme Gradient Boosting (XGBoost), Principal Component Analysis (PCA).

1. Introduction

Stock price prediction has consistently been a popular issue, attracting researchers from various fields such as finance, economics, mathematics, and computer science. Successfully forecasting changes in stock prices contributes to a thorough market analysis, enhancing our understanding of operational patterns. This, in turn, provides a more comprehensive perspective for long-term investment and trading, aiding in making informed decisions when buying or selling stocks. Financial time series data is full of noise which means that the training data will not contain enough information for the network to learn the function, but is also susceptible to various factors, including news events, rumors, natural elements, etc [1]. Therefore, accurately predicting stock prices has always been an exceedingly challenging task.

Therefore, it is very challenging to predict the direction of stock prices through data mining. This paper aims to identify more effective machine learning methods for forecasting stock price movements and enhance the performance of these methods through feature engineering. In this paper, the author compares several machine learning methods on the Dow Jones dataset and employs Principal Component Analysis (PCA) as a preprocessing step.

We believe that performing certain feature engineering will enhance the model's generalization for more accurate predictions. This paper will compare several popular machine learning methods and include machine learning methods with PCA as a preprocessing step in the comparison. The model with the highest accuracy on the selected dataset will be identified. This will help us place greater emphasis on the importance of feature engineering for the model in future work.

2. Literature Review

Nowadays, there are many methods for predicting stock market prices. Initial research favored statistical methods, but their performance was subpar [2]. The emergence of machine learning

methods and deep learning methods has greatly improved the accuracy and prediction ability of traditional statistical methods.

In recent years, various kinds of deep learning methods and mixed models have been proposed to achieve the transcendence of the traditional method. Erkam Guresen and colleagues assessed the performance and advantages of multiple neural network models and hybrid networks in stock forecasting. They found that MLP was still the most effective artificial neural network structure [3].

Yuling Lin and colleagues utilized Support Vector Machines (SVM) to predict stock trends [4]. They first selected a good subset of indicators through a method and obtained features through evaluation. Then, employing quasi-linear SVM with the chosen indicators as weighted inputs, they predict stock market movements in historical data.

Ash Booth and colleagues proposed a prediction system based on a performance-weighted ensemble of random forests for forecasting the price impact of order book events [5]. Due to the use of different subsets for training decision trees at each node, random forests avoid over-reliance on the entire dataset. Moreover, as the construction of each decision tree is independent of others, random forests can be easily parallelized, accelerating the model training speed.

XGBoost was initially proposed by Tianqi Chen in 2016, and the literature has demonstrated its characteristics, including low computational complexity, fast execution speed, and high accuracy [6].

Kyung Keun Yun and colleagues propose a model called GA-XGBoost, which selects the best set of feature sets by extending the feature set before the XGBoost algorithm [7]. But they used a considerable number of features. However, the quality and quantity of features included in stock forecasting research need to be further studied. Moreover, as ordinary investors often have limited feature data, we will select datasets with fewer features and modify their methods in feature engineering to better reflect real-world scenarios.

Sadegh Jalalian shared several machine learning code snippets on his Kaggle profile [8]. We appreciate the ideas provided by this individual. We organized and tested these machine learning methods on our dataset. Additionally, we proposed incorporating PCA as a preprocessing step before applying machine learning methods to enhance model accuracy and generalization.

3. EDA (Statistics Summary)

This paper first selected the dataset of the Dow Jones Industrial Average from April 1, 2008, to March 29, 2018, and conducted exploratory data analysis.

3.1. Dataset Overview

First, an overview of the data reveals a total of 2448 days of stock prices and related data. In addition to the date, there are a total of 13 features, LABEL (the next day's closing price change relative to the current day, where 1 indicates an increase and 0 indicates a decrease), InterestRate (interest rate), ExchangeRate (exchange rate), VIX (volatility index), Gold (gold price), Oil (oil price), TEDSpread (the difference between U.S. Treasury and Eurodollar deposit rates), and EFFR (the target interest rate of the Federal Reserve). The 'LABEL' is also the indicator we aim to predict. From Figure 1, it can be observed that 45.8% of the days in the entire dataset show a decrease, while 54.2% show an increase.

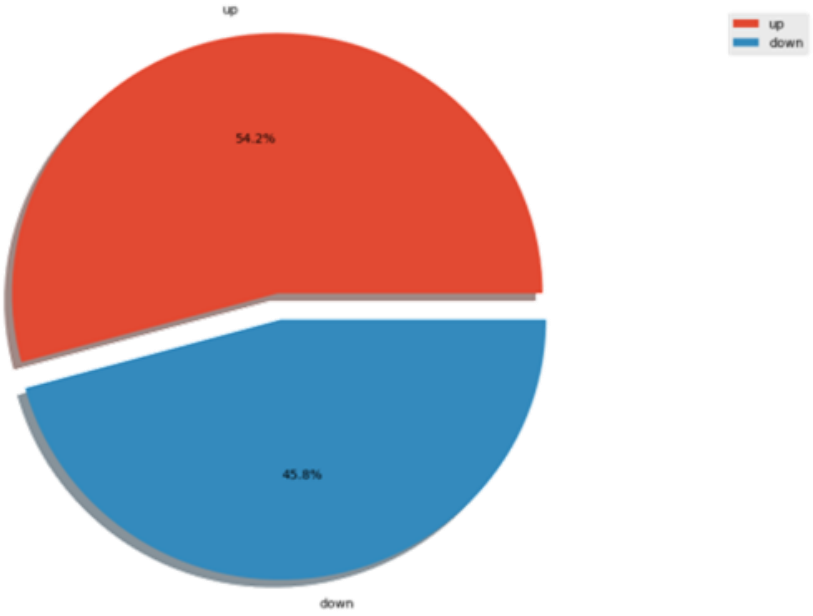

Figure 1. Price Change Ratio Chart

3.2. Missing Data Analysis

Now, we will analyze the missingness of the data. We use the Python package 'missingno' to inspect the data, and Figure 2 visualizes this inspection. In the visualization, black indicates the presence of data, and if there are white horizontal lines, it signifies missing values. From Figure 2, it can be observed that there are no missing values in any column, indicating that the data is complete and available.

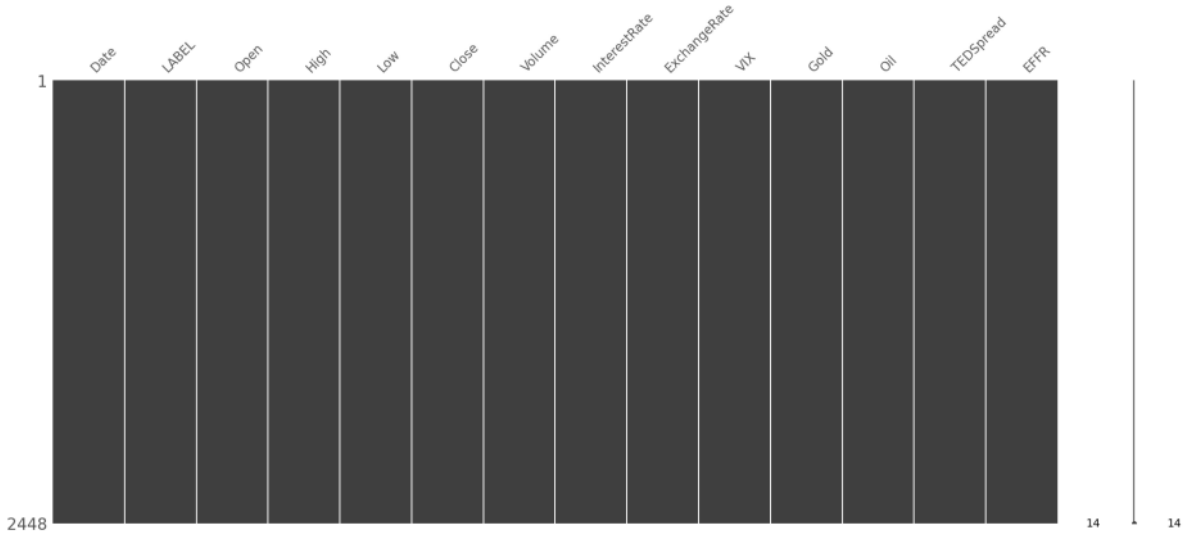

Figure 2. Missing Values Visualization

3.3. Correlation Analysis

This paper plotted a correlation heatmap for the 13 features excluding the date. As shown in Figure 3, darker colors indicate a stronger negative correlation between two features, while lighter colors suggest a stronger positive correlation. It can be observed that the opening price, highest price, lowest price, and closing price are all positively correlated with each other. Additionally, these values show a relatively large negative correlation with the values of interest rate, exchange rate, and volatility.

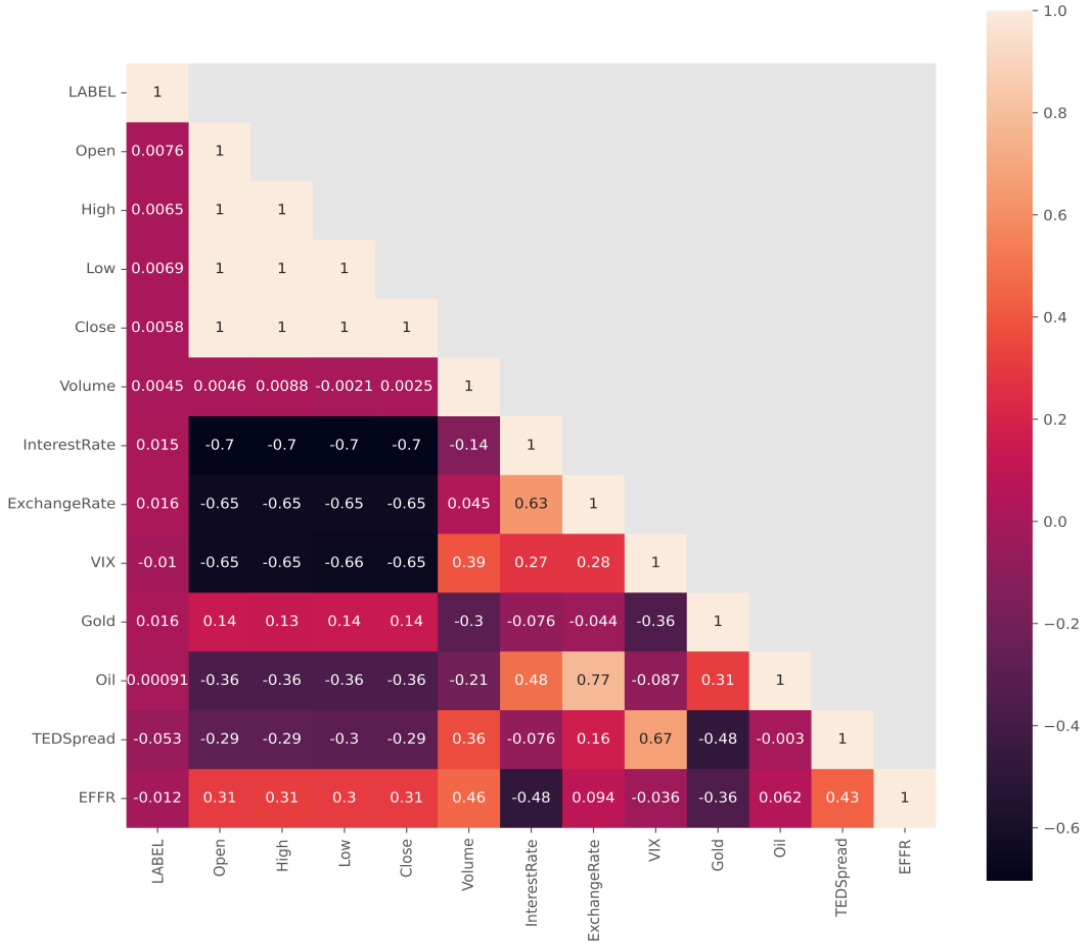

Figure 3. Correlation Heatmap for 13 features

Of particular interest is the correlation with the 'LABEL' feature, as these features have the most significant impact on our predictive results. It can be seen that relatively speaking, interest rate, exchange rate, gold price, and the difference between U.S. Treasury and Eurodollar deposit rates have a slightly larger correlation with the 'LABEL' feature.

4. Methods

Before commencing our experiment, we need to partition the dataset into an 80:20 ratio, with 80% of the days designated as the training set, and then test the remaining 20% set to evaluate the accuracy of these model predictions.

4.1. SVM

This method involves finding a hyperplane that can correctly classify the training samples:

$$\begin{cases} \omega^T \mathrm{X}_i + b \geq +1 & \mathrm{y}_i = +1 \\ \omega^T \mathrm{X}_i + b \leq -1 & \mathrm{y}_i = -1 \end{cases} \tag{1}$$

Where X_i represents the vector composed of 13 features for the i-th data point, and y_i represents the category corresponding to the label. The optimal separating hyperplane refers to the hyperplane with the maximum margin, which is achieved by solving the following optimization problem:

$$\begin{cases} \min_{\omega,b} \frac{1}{2} \| \omega \|^2 \\ s.t. \ \ y_i(\omega^T \mathrm{X_i} + b) \geq -1 \end{cases} \tag{2}$$

Using the first 80% of the data to solve such an optimization problem, obtaining the values for ω and b, and then substituting them into the features of the remaining 20%, we obtain the value of $\omega^T \mathrm{X_i} + b$. If the result is positive, the predicted label is 1; if it is negative, the predicted label is 0.

4.2. Random Forest

The algorithm creates a forest where every decision tree in the forest is randomly generated, and each node of these trees is a random subset of the characteristics that interact with the final output. If it is a classification problem, the random forest in the test data will determine the results of the voting system based on the output categories of the various decision trees.

In Python, we can directly train the model using the Random Forest Classifier function, and for this instance, we choose to use 10 trees.

4.3. XGBoost

XGBoost (eXtreme Gradient Boosting) is a machine learning algorithm based on gradient boosting trees. The main steps of XGBoost are as follows:

1. Initialize the model: Set the initial parameters of the model, including the learning rate, the number of trees, the depth of the trees, the sum of the smallest sample weights in the child node, evaluation indicators for regression or classification problems, etc.
2. Build trees: XGBoost progressively improves the model's performance by iteratively building trees. In each iteration, a new tree is constructed to correct the errors of the previous trees.
3. Compute the gradient of the loss function: At each step, XGBoost calculates the gradient of the loss function concerning the target variable.
4. Build a new tree: Construct a new decision tree using the gradient information and add it to the model.
5. Regularization: During the tree-building process, XGBoost considers regularization terms to prevent overfitting.
6. Update the model: Combine the newly constructed tree with the previous trees to obtain a more powerful model.
7. Repeat: Repeat the above steps until the stopping conditions are met (e.g., reaching the specified number of trees).

We use the `XGBClassifier` function in Python for training and classification prediction.

4.4. PCA-SVM, PCA-Random Forest and PCA-XGBoost

Principal component analysis (PCA) is a method of reducing the high feature dimension to the low feature dimension, removing redundancy and preserving the most critical information of the data. The goal of PCA is to obtain a set by selecting the principal components in the data, which greatly preserves the variance in the original data.

The specific steps are as follows:

1. Data Centering: For each feature, subtract its mean across the entire dataset to ensure that the center of the dataset is at the origin.
2. Compute Covariance Matrix: Calculate the covariance matrix of the dataset, where covariance reflects the relationships between different features.
3. Compute Eigenvalues and Eigenvectors: Use SVD decomposition to find eigenvalues and eigenvectors of covariance matrix.

4. Select Principal Components: The eigenvalues are arranged from the largest to the smallest, and we expect them to be reduced to the k dimension, then the eigenvectors corresponding to the first k eigenvalues are selected as principal components.

5. Projection: The data set is simply projected onto the selected principal component to obtain a low-dimensional data set.

In this problem, we choose 5 principal components and then incorporate these components into the previous model for experimentation.

5. Results

According to the above method, the accuracy for each corresponding method is shown in Table 1.

Table 1. Accuracy of Predictions by Different Methods

Methods	Accuracy	PCA-Method	Accuracy
SVM	51.43%	SVM	51.63%
Random Forest	49.18%	Random Forest	51.63%
XGBoost	51.43%	XGBoost	53.67%

6. Conclusion

According to the experimental results in Table 1, we observe that the accuracy of SVM and XGBoost on the dataset is slightly higher than that of Random Forest. Moreover, each method shows an improvement in classification accuracy when PCA is applied as a preprocessing step. This aligns with our hypothesis, as PCA enables each model to be trained using more effective features, enhancing the model's generalization capability.

We find that PCA has the most significant impact on improving Random Forest, indicating that the PCA operation has the most pronounced effect on this algorithm. Ultimately, the accuracy of PCA-XGBoost is the highest among the tested methods.

References

[1] Abu-Mostafa, Y.S., Atiya, A.F. “Introduction to financial forecasting”. Applied Intelligence. 1996. 205 – 213.

[2] Hyun jung Kim, Kyung shik Shin. “A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets”. Applied Soft Computing Journal. 2007, 569 – 576.

[3] Erkam Guresen, Gulgun Kayakutlu, and Tugrul U. Daim.“Using artificial neural network models in stock market index prediction”. Expert Systems with Applications. 2011, 10389 – 10397.

[4] Yuling Lin, Haixiang Guo, and Jinglu Hu.“An SVM-based approach for stock market trend prediction”. The 2013 International Joint Conference on Neural Networks (IJCNN). 2013, 1 – 7.

[5] Ash Booth, Enrico Gerding, and Frank McGroarty. “Predicting equity market price impact with performance weighted ensembles of random forests”. 2014 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr). 2014, 286 – 293.

[6] Tianqi Chen, Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD ’16. ACM. 2016.

[7] Kyung Keun Yun, Sang Won Yoon, and Daehan Won. “Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process”. Expert Systems with Applications. 2021.

[8] Sadegh Jalalian. https://www.kaggle.com/sadeghjalalian.