S&P 500 Prediction Model Project Report

Brian Witarsa

Abstract

The main objective of this project is to develop a prediction model for the S&P 500 index, aiming to forecast whether the index will increase or decrease on the following trading day. This paper outlines the steps taken to develop the solution, including data collection, the methodology used, and the insights obtained. Furthermore, additional predictors and methods were introduced to enhance the accuracy of the predictions. Comparisons between the model before and after the introduction of new predictors and methods are made. Moreover, a random baseline model was also created to draw comparisons from the Random Forest model, evaluating the effectiveness of the developed model. Additional metrics that measure performance are also introduced to gain additional information on how well the model performs.

Introduction

The S&P 500 is one of the most widely followed stock market indices, comprising 500 leading companies from various sectors. Accurate predictions of the index's movement can provide valuable insights for investors and help inform their stock purchasing decisions. This report presents the development of a prediction model for the S&P 500 index using historical data and machine learning techniques.

Data Collection

The data source for this analysis is historical data for the S&P 500 index. The data is obtained using the yfinance module, specifically downloading data with the symbol GSPC,

which represents the S&P 500. The historical data includes information such as open, close, volume, high, and low prices.

Data Preprocessing

Before training the prediction model, the historical data undergoes data cleaning to ensure reliability and eliminate unnecessary information. Columns such as "Dividend" and "Stock Splits" are removed from the dataset as they are not relevant to the prediction task. These columns are more relevant when measuring individual stocks, and because the S&P 500 is an index fund, its use is not required in this project. Additionally, to reflect the current market conditions accurately, old market data before the 1990s is excluded from the dataset. Market data and conditions prior to 1990 vary drastically to the data and conditions we observe in the market today. This step helps in avoiding false predictions and ensures the model is based on recent and relevant data.

Setting Target and Predictors

The target variable for our model is set to whether the S&P 500 price will increase or decrease on the following trading day. To do this, we must first create a “Tomorrow” column, where we shift the “Close” price column one step backwards. The “Tomorrow” column would then hold the close prices of the following day. The target is generated by comparing the Tomorrow column and the Close column. If the price increases, the “Target” column will hold the boolean value 1. If the price decreases, the column will hold 0.

The predictors used to train the model are the features extracted from the historical data, including Close, Open, Volume, High, and Low prices. These predictors provide valuable information for making predictions based on historical trends. These are the predictors used to

form the initial results. New predictors that incorporate rolling averages and an increased prediction threshold will be introduced, from which we can calculate new metrics of performance to compare with our initial baseline results. This allows us to measure the impact of the new predictors and threshold. A deeper dive into the new predictors and increased threshold is included later in this report.

Machine Learning Model

To develop the prediction model, the Random Forest algorithm is employed. Random Forest is a type of decision tree algorithm that combines multiple individual decision trees from which an average is taken to make accurate predictions. The RandomForestClassifier requires three parameters: n_estimators, min_samples_split, and random_state. n_estimators refers to the number of individual decision trees to be trained, min_samples_split refers to the minimum number of samples required to split an internal node when constructing a decision tree, and the random_state sets the random seed that ensures the random process Random Forest undertakes. A greater value for n_estimators produces more accurate results, but slows the program. A higher value of min_samples_split produces a less accurate model, but one that is also less prone to overfitting.

Random Forest was chosen for this task due to its ability to capture non-linear relationships, which is crucial when dealing with stock prices that often exhibit complex patterns. Random Forest is also relatively fast, which allows the user to perform testing operations quickly. Furthermore, Random Forest is known for its resistance to overfitting, which ensures more reliable predictions.

Model Training and Backtesting

The model is trained using the historical data, with the target variable being the S&P 500 price movement for the following trading day. To verify the model's performance and ensure its generalization ability, backtesting is conducted. Backtesting involves training the model with a portion of the data and then testing it on subsequent data to evaluate its performance. For this project, a rolling backtesting approach is adopted.

In the backtesting process, the model is initially trained with data from the first 10 years, equivalent to 2500 trading days. Subsequently, the model is trained at regular intervals, with each interval increasing by 250 days (approximately one year). By adopting this rolling backtesting approach, we can train the model on an incrementally increasing amount of data for each successive year. Therefore, each iteration becomes better equipped to capture changes in the market, and adapt to evolving patterns and trends.

Improving the model

As mentioned earlier, the initial set of predictors employed were surface level predictors. To enhance the model’s predictive capabilities and produce more accurate results, we must introduce new predictors and adjustments. Thus, we incorporated rolling averages and adjusted the prediction threshold to increase performance.

1. Incorporating Rolling Averages

The first approach to enhance prediction accuracy is by incorporating rolling averages. These averages calculate the mean close price over various specific timeframes, such as the past 2 days, past trading week, past 3 trading months, past trading year, and past 4 trading years. These values are then used to calculate a new “ratio_column” that calculates the ratio between

the S&P 500 close price and the rolling averages calculated across the timeframes set previously. Another column called “trend_column” is used to store the sum of the “Target” column over the time periods outlined earlier. By utilizing this information to add to the existing predictors, the model can capture trend patterns and other ratios, potentially leading to more accurate predictions.

2. Adjusting Prediction Threshold

Another method to improve accuracy is to adjust the threshold for predicting an increase or decrease in the S&P 500 price. The default threshold of 50 percent implies that the model predicts an increase if it estimates a greater than 50 percent chance of it occurring. By increasing the threshold to 60 percent, the model's predictions become more conservative and confident. This adjustment might result in more reliable and actionable predictions for investors.

Comparison Before and After new predictors

To evaluate the performance of the model with and without new predictors and the increased threshold, we will compare each model’s precision score. The precision score measures the proportion of true positive predictions out of all positive predictions made by the model. It measures the accuracy of the model making a correct positive prediction. Without new predictors like the ratio and trend columns and prior to the threshold being increased to 60 percent, the Random Forest model had a precision score of 53.06 percent. With the additional predictors, however, the model returned 56.93 percent. This means that the new predictors greatly increased the ability for the model to produce correct positive predictions, which demonstrates their impact and importance.

Comparison with Random Baseline

To further evaluate the effectiveness of the developed prediction model, a random baseline model is implemented and compared. The random baseline model randomly predicts whether the S&P 500 price will increase or decrease by randomly generating binary predictions for whether the price will increase or decrease, which is then back tested using our newly created backtest_random method. This will provide us a baseline to compare our Random Forest results against.

Adding metrics to measure performance

Initially, only a precision score was used to evaluate the accuracy of the program. This metric provided a useful means of comparing the results of the model before and after adding new predictors like the ratio and trend columns, and also incorporating an increased threshold. While the precision score is an adequate measure of performance, additional metrics were included to give the user a deeper understanding on the performance of the model.

Metrics like Confusion Matrix and ROC AUC Scores were added, which would provide users with more metrics to derive their conclusions on. The Confusion Matrix is a tabular way to visualize the performance of a model. It consists of the True Positives, True Negatives, False Positives, and False Negatives. The ROC AUC scores represent the degree of separability. The higher the AUC, the better the model is at predicting decreases as decreases and increases as increases. The ROC curve is plotted with True Positive Rate and against the False Positive Rate.

Each of these metrics give the user an insight into how the model performs, as compared to the random baseline results we created earlier.

Interpreting Results

```
Evaluation for RandomForest model
Precision: 0.5693251533742332
ROC AUC Score: 0.508300378914291

Confusion Matrix:
[[1908  351]
 [2234  464]]

Evaluation for Random baseline
Precision: 0.5333333333333333
ROC AUC Score: 0.48884800145435886

Confusion Matrix:
[[1090 1169]
 [1362 1336]]
```

When both the Random Forest approach and the Random Baseline approach were evaluated, the results in the image above were produced. The Random Forest outperformed the Random Baseline approach on most metrics. First, Random Forest produced a higher precision score of 56.93 percent as compared to our Random Baseline's results of 53.33 percent. Comparing the ROC AUC Score for both approaches tells us that Random Forest is marginally better at distinguishing between the true positives and false positives. However, as the ROC scores are very close, it means that Random Forest did not substantially outperform our Random Baseline approach. Finally, we can compare the Confusion Matrix. Random Forest shows better performance in terms of true negatives and false positives, but the random baseline approach performs better in correctly classifying true positive samples.

Interpreting these results, we can see that Random Forest outperformed Random Baseline when it came to precision score and ROC AUC Score. However, when comparing the Confusion Matrix, Random baseline was able to produce more true positive samples compared to our Random Forest model. From these results we can conclude that the Random Forest approach outperformed our Random Baseline approach, but only slightly.

Conclusion

This paper presents the development of a prediction model for the S&P 500 index, aiming to forecast its price movement for the following trading day. The model utilizes the Random Forest algorithm and is trained using historical data from the S&P 500 post 1990. The analysis explores ways to enhance prediction accuracy, such as incorporating rolling averages and adjusting the prediction threshold. The model's performance is evaluated through backtesting and compared to a random baseline model.

When comparing the results, we find that the new predictors and threshold set greatly improved the model's performance, increasing the precision score from 53.06 percent to 56.93 percent. This indicates the importance of adding new predictors into our model. When comparing the Random Forest method to our Random Baseline approach, we find that the Random Forest approach is an improvement upon the Random Baseline approach, denoted by its higher precision and ROC AUC score. However, the confusion matrix indicates that each approach had their positives and negatives, with the Random Baseline approach even outperforming Random Forest by some metrics.

This means that while the Random Forest approach did not show a significant improvement over the Random Baseline method implemented, it still performs better and is therefore more useful for investors. Random Forest is able to analyze non-linear relationships, is resistant to overfitting, and robust to outliers. These strengths allow it to learn meaningful patterns within the historical market data. However, our results indicate that the Random Forest approach only marginally outperformed our Random Baseline approach on several metrics, which indicate a weakness in Random Forest's predictive capabilities. There could be a few reasons for this, most importantly that predicting prices in the stock market is incredibly difficult,

given the volatility of the markets. This task is made especially difficult given limited information. Since we only used historical data of the S&P 500 to formulate our predictions, it would be extremely difficult for any model to substantially outperform the market with time series data alone. I believe the marginal difference in results between Random Forest and our Random Baseline indicates a limitation in the data used, more so than the model. With that said, our model had a precision score of 56.93 percent, which would at least turn a profit if an investor decided to follow its every prediction. The project has also shown the importance of constantly adding new predictors within the model to enhance its predictive accuracy. The incorporation of these predictors greatly improved our model' success. More predictors could certainly be added to further enhance this model's accuracy.

Overall, this prediction model serves as a valuable starting tool for investors looking to gain insights into the future movement of the S&P 500 index. By continuously updating and refining the model with new data and additional predictors, investors can train this model to make more informed decisions, leading to improved stock trading strategies.