Optimizing Stock Price Prediction through Machine Learning

Nicholas Chludzinski
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ, United States of America
nchludzi@stevens.edu

Optimizing Stock Price Prediction through Machine Learning

Abstract

I aim to develop a predictive model for stock market trends to assist in investment decision-making. I employed a Gradient Boosting Classifier (ensemble technique) to predict the continuous flow of stock price data. The model achieves an overall accuracy of 83%, precision of 71%, recall of 100% and an f1-score of 83%, demonstrating that the model effectively predicts binary stock price movements exactly one week into the future. A major contribution of this work was from the paper, 'Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model ,' in which the LightGBM model utilized GridSearchCV for hyperparameter tuning. An advantage I found over this existing solution is the GBC is able to successfully predict beyond historical data, unlike LightGBM.

Introduction

Problem Statement:

My project aims to develop a predictive model for stock market trends to assist in investment decision-making. The objective is to provide investors with accurate predictions that support proactive investment strategies, enabling investors to capitalize on emerging opportunities and mitigate risks efficiently. This initiative seeks to address the challenges presented by the complexity and uncertainty of the stock market.

Dataset Description:

The dataset contains historical stock market data (1/1/2022 - 8/2/2024) sourced from Yahoo Finance/ historical data and includes standard Open, High, Low, Close, Adjusted Close, and Volume features. I intentionally avoided including data prior to 2022 to avoid the impact that Covid-19 had on the global market, thereby maintaining the integrity of the dataset. Additional features include moving average which smoothes price fluctuations and provides a clearer trend indication. The dataset also includes a crucial engineered feature called the 'Result' column. This is a binary column that I derived to aid specifically in classification, and it indicates whether the

stock price is projected to increase (1) or decrease/ stay the same (0) compared to the previous day's closing price. This Result column serves as the target variable, allowing my model to accurately forecast future price movements based on historical data. Data preprocessing included data type conversion (date column converted to datetime format allowing for easier manipulation and indexing based on dates), filling in missing values with a mean value, and feature selection.

Experimental Results:

The Gradient Boosting Classifier (GBC) achieved an accuracy of 83% (overall correctness), precision of 71% (proportion of true positive predictions), recall of 100% (true positive predictions identified out of all actual positive instances) and an F1-score of 83% (balance between precision and recall) across 12 randomly selected publicly traded corporations. Intermediary results included the optimal combination of hyperparameters (learning_rate, max_depth, and n_estimators) according to each specific dataset.

My Solution Compared to Existing Solutions:

My model is a classification model and achieves an overall accuracy of 83%, precision of 71%, recall of 100% and an f1-score of 83%. The metrics of the paper are based on regression, and its resulting RMSE of 0.0215 indicates a very low prediction error. However, unlike the LightGBM model, which did not predict stock prices beyond the available historical data, my model successfully predicted the price movement one week into the future, making it capable of future forecasting. I believe this forward-looking capability combined with my strong performance metrics, enhances the effectiveness of my model's solutions.

Related Work

In the paper, LightGBM is evaluated alongside multiple other ensemble methods like CatBoost, XGBoost, and AdaBoost. The results indicated that LightGBM outperforms these alternative algorithms in terms of RMSE and MAE. This is most likely due to LightGBM's ability to handle outliers effectively and its vigorous hyperparameter tuning. The high level of accuracy (low prediction error) can be categorized as a pro. A con of this LightGBM may be its computational cost due to extensive hyperparameter tuning.

[1]H. Anggit, K. Yanuar, and P. Yoga, "Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model," *Research Gate*, Dec. 31, 2023. https://www.researchgate.net/publication/377404962_Stock_Price_Time_Series_Data_Forecasting_Using_the_Light_Gradient_Boosting_Machine_LightGBM_Model (accessed Jul. 20, 2024).

Solution

To achieve accurate future stock price prediction, I employed a Gradient Boosting Classifier (GBC), an ensemble method that sequentially builds decision trees in order to minimize prediction error and capture non-linear relationships. This algorithm combines multiple weak learners sequentially, where each decision tree corrects errors of the preceding tree, which enhances the predictive power of the model. Data preprocessing included data type conversion (date column converted to datetime format allowing for easier manipulation and indexing based

on dates), filling in missing values with a mean value, and Standard Scaler for standardizing the features of the dataset. Feature engineering including moving average, and a 'Result' column. Hyperparameter tuning using GridSearchCV successfully enhances the model's reliability.

Machine Learning Algorithm

Gradient Boosting Classification (GBC) is an appropriate algorithm for stock price prediction because of its ability to handle complex stock trends and make accurate, binary classification predictions (stock price increase or stock price decrease/ neutral) based on the dataset features. Key hyperparameters included n_estimators (50, 100, 150), learning_rate (0.01, 0.1, 0.2) and max_depth or the maximum depth of each tree (3, 5, 7). These hyperparameters were tuned using GridSearchCV through use of cross-validation to evaluate each combination. The overall framework/ main design is first data preprocessing (date to datetime), and filling in any possible missing values with the mean of each column. Next is feature engineering, where I introduce a moving average and a predictive 'Result' column. The data is then split into 80% training and 20% testing. After this, the features are standardized using Standard Scaler and GridSearchCV is used to tune hyperparameters. Data is fitted to the training set, optimal hyperparameters are output as intermediary results, and the model is evaluated using the testing set. The best model is selected based off of the earlier hyperparameter tuning, and predictions are made. These predictions are then compared to closing prices exactly one week in advance from the end of the historical data, giving a binary indication of correctness (yes the prediction is correct or no the prediction is incorrect) as well as if our prediction was a True Positive (TP), True Negative (TN), False Positive (FP) or False Negative (FN). These indicators are then tabulated using a counter and used to calculate the accuracy, precision, recall, and f1-score of the model. Distribution of correct and incorrect predictions is visualized on a pie chart.

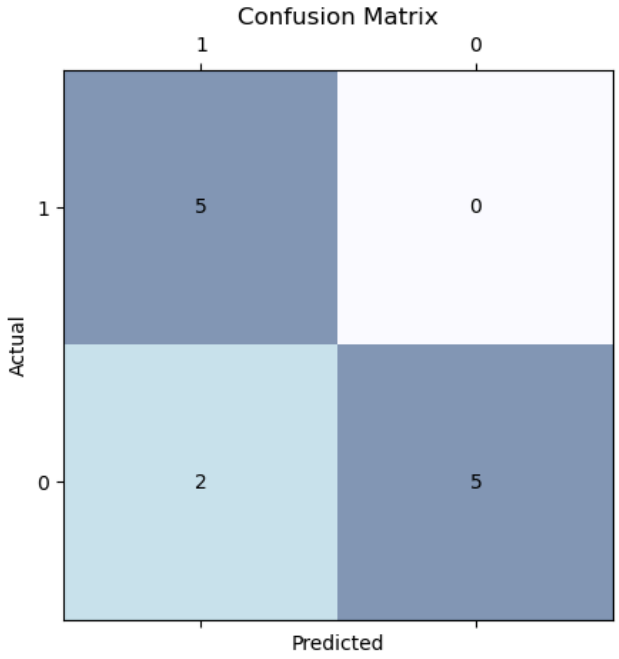
Implementation Details

I tested and validated the performance of my model using a confusion matrix where True Positives (TP) are defined as when the model predicts closing price will increase, and it actually increases, False Positives (FP) are defined as when the model predicts closing price will increase, and it actually decreases or stays the same, True Negatives (TN) are defined as when the model predicts closing price will decrease or stay the same, and it actually decreases or stays the same, and False Negatives (FN) are defined as when the model predicts the price will decrease or stay the same, and it actually increases. These indications are then tabulated using a counter and programmatically fed into functions that determine accuracy, precision, recall and F1-score. These results are output and model accuracy is visualized using a pie chart.
I did not attempt to guess the optimal values of my hyperparameters, rather, I implemented GridSearchCV to tune my hyperparameters through use of cross-validation to evaluate each combination.
The best performing model is selected using GridSearchCV.fit based on the metric of accuracy meaning that the model configuration that achieved the highest cross-validated accuracy during the grid search process is selected to move forward and predict.

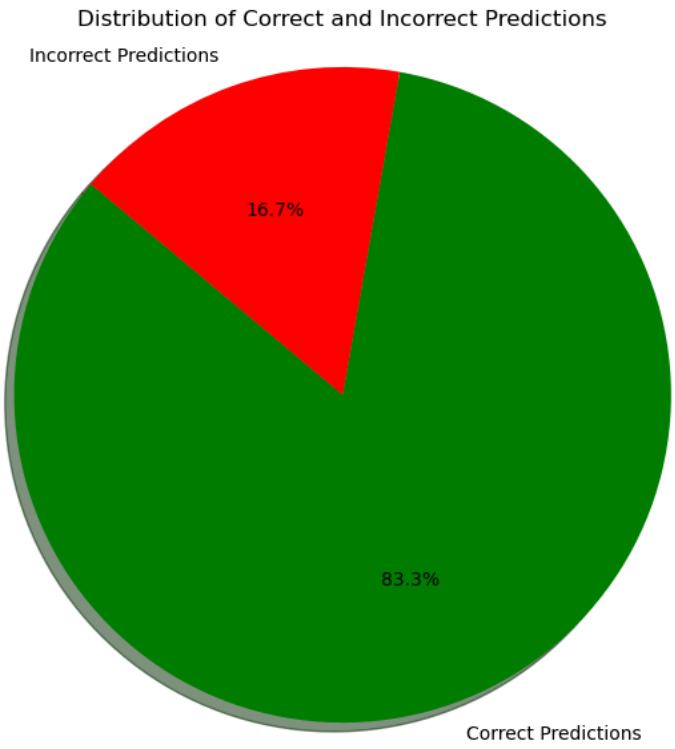
Additional engineered features include a moving average to assist in trend behavior as well as a binary ‘Result’ column derived to aid specifically in classification, allowing my model to forecast future price movements based on historical data features.

```
Predicting for Citigroup...

Best parameters:
{'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 50}
Citigroup - Prediction: The closing price will decrease or stay the same one week from now.
Citigroup - Actual closing price on 8/2: $58.76
Citigroup - Actual closing price on 8/9: $57.84
Citigroup - Was the prediction correct? Yes (TN)

Predicting for HUTCHMED...

Best parameters:
{'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 100}
HUTCHMED - Prediction: The closing price will increase one week from now.
HUTCHMED - Actual closing price on 8/2: $18.13
HUTCHMED - Actual closing price on 8/9: $19.87
HUTCHMED - Was the prediction correct? Yes (TP)

Predicting for Microsoft...

Best parameters:
{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
Microsoft - Prediction: The closing price will decrease or stay the same one week from now.
Microsoft - Actual closing price on 8/2: $408.49
Microsoft - Actual closing price on 8/9: $406.02
Microsoft - Was the prediction correct? Yes (TN)

Predicting for Tesla...

Best parameters:
{'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 50}
Tesla - Prediction: The closing price will increase one week from now.
Tesla - Actual closing price on 8/2: $207.67
Tesla - Actual closing price on 8/9: $200.00
Tesla - Was the prediction correct? No (FP)

Predicting for Morgan Stanley...

Best parameters:
{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
Morgan Stanley - Prediction: The closing price will decrease or stay the same one week from now.
Morgan Stanley - Actual closing price on 8/2: $95.85
Morgan Stanley - Actual closing price on 8/9: $94.72
Morgan Stanley - Was the prediction correct? Yes (TN)

Predicting for META...

Best parameters:
{'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 50}
META - Prediction: The closing price will increase one week from now.
META - Actual closing price on 8/2: $488.14
META - Actual closing price on 8/9: $517.77
META - Was the prediction correct? Yes (TP)
```

```
Predicting for Apple...

Best parameters:
{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 50}
Apple - Prediction: The closing price will decrease or stay the same one week from now.
Apple - Actual closing price on 8/2: $219.86
Apple - Actual closing price on 8/9: $216.24
Apple - Was the prediction correct? Yes (TN)

Predicting for NVIDIA...

Best parameters:
{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 50}
NVIDIA - Prediction: The closing price will increase one week from now.
NVIDIA - Actual closing price on 8/2: $107.27
NVIDIA - Actual closing price on 8/9: $104.75
NVIDIA - Was the prediction correct? No (FP)

Predicting for Rocket Lab...

Best parameters:
{'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 50}
Rocket Lab - Prediction: The closing price will increase one week from now.
Rocket Lab - Actual closing price on 8/2: $4.81
Rocket Lab - Actual closing price on 8/9: $5.37
Rocket Lab - Was the prediction correct? Yes (TP)

Predicting for H&R Block...

Best parameters:
{'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 100}
H&R Block - Prediction: The closing price will increase one week from now.
H&R Block - Actual closing price on 8/2: $56.79
H&R Block - Actual closing price on 8/9: $57.18
H&R Block - Was the prediction correct? Yes (TP)

Predicting for Bank of America...

Best parameters:
{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}
Bank of America - Prediction: The closing price will increase one week from now.
Bank of America - Actual closing price on 8/2: $37.58
Bank of America - Actual closing price on 8/9: $38.28
Bank of America - Was the prediction correct? Yes (TP)

Predicting for Sanofi...

Best parameters:
{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
Sanofi - Prediction: The closing price will decrease or stay the same one week from now.
Sanofi - Actual closing price on 8/2: $53.05
Sanofi - Actual closing price on 8/9: $52.37
Sanofi - Was the prediction correct? Yes (TN)
```



```
Overall Accuracy: 0.83
Overall Precision: 0.71
Overall Recall: 1.00
Overall F1-Score: 0.83
```



Comparison

My model is a classification model and achieves an overall accuracy of 83%, precision of 71%, recall of 100% and an f1-score of 83%. The metrics of the paper are based on regression, and its resulting RMSE of 0.0215 (paper's best model) indicates a very low prediction error. However, unlike the LightGBM model, which did not predict stock prices beyond the available historical data, my model successfully predicted the price movement one week into the future, making it capable of future forecasting. I believe this forward-looking capability combined with my strong performance metrics, enhances the effectiveness of my model's solutions.

Future Directions

Given an extra 3-6 months, I would like to have explored different ways to enhance the model's performance in terms of using text classification, specifically Natural Language Processing.

Conclusion

I believe my project has made significant progress in predicting stock prices. My data and results indicate strong performance with high accuracy. Further techniques, such as incorporating additional financial indicators could enhance the model's predictive accuracy.

References

[1]H. Anggit, K. Yanuar, and P. Yoga, "Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model," *Research Gate*, Dec. 31, 2023. https://www.researchgate.net/publication/377404962_Stock_Price_Time_Series_Data_Forecasting_Using_the_Light_Gradient_Boosting_Machine_LightGBM_Model (accessed Jul. 20, 2024).