PREDICTING 1-DAY STOCK PRICE MOVEMENT USING MACHINE LEARNING MODELS

BY CHARLES LUSWATA,

MS, BSTAT

Presented to the Faculty of The University of California, Berkeley Haas in Partial Fulfillment of the Requirements for the Professional Certificate in Machine Learning and Artificial Intelligence

PREDICTING 1-DAY STOCK PRICE MOVEMENT USING MACHINE LEARNING MODELS

BY CHARLES LUSWATA, MS

Introduction

Accurately predicting stock price movements is a challenging yet essential task for investors, financial analysts, and policymakers. Effective short-term predictions can greatly enhance investment strategies and risk management. This capstone project aimed to develop and evaluate machine learning models to predict whether a stock's price will rise (Bullish) or fall (Bearish) the next trading day based on historical data and relevant features.

Problem Statement

Stock price movements are influenced by various factors including historical prices, trading volumes, market sentiment, company fundamentals, and external economic indicators. Traditional financial models often struggle with the non-linearity and volatility inherent in stock prices. This project sought to harness modern machine learning techniques to forecast whether a stock's price would increase or decrease the following day, providing valuable insights for short-term trading and investment decisions.

The project involved a comparative analysis of classifiers: K-Nearest Neighbor (KNN), Logistic Regression (LR), Gradient Boosting (GB) and Random Forest Decision Trees (RF) to classify the next day's stock price as Bullish or Bearish.

K-Nearest Neighbor (KNN) is a non-parametric algorithm that classifies data based on the similarity of feature vectors, with classification determined by the majority class among the k-nearest neighbors. It is particularly useful for tasks where decision boundaries are irregular or complex (Cover and Hart, 1967; Han et al., 2011).

Random Forest (RF) is an ensemble technique that builds multiple decision trees during training and outputs the mode of the classes (for classification tasks) or mean prediction (for regression tasks) from individual trees. It is known for its robustness, ability to handle large datasets, and resistance to overfitting (Breiman, 2001).

Logistic Regression (LR) is a widely-used statistical method for binary classification. It applies the logistic function to estimate probabilities, which are then used to classify observations into one of two classes. Logistic Regression is favored for its simplicity, interpretability, and efficiency, particularly in scenarios where the relationship between the dependent and independent variables can be approximated by a linear boundary (Hosmer et al., 2013).

Gradient Boosting Classifier (GB) build models sequentially, with each model correcting the errors of its predecessors. GBs combine predictions from multiple weak learners, typically decision trees, to form a strong overall model. They are praised for their predictive accuracy and ability to manage various types of data (Friedman, 2001; Chen and Guestrin, 2016). GBs require careful hyperparameter tuning to avoid overfitting.

Objectives

1. Compare KNN, LR, GB and RF models based on accuracy, training time, and ROC-AUC score to identify the most effective model for predicting the next day's stock price action.
2. Analyze the interpretability of results from the selected model to understand the factors influencing stock price movements.
3. Explore complex interaction patterns among key features using the selected model.

Methodology

Data Sources: Historical stock price data was sourced from http://www.barchart.com.

Feature and Target Variables: The dataset was split into features (X) and target labels (y), and then into training and test sets using Python's *train_test_split* function.

Variable Description:

symbol (Symbol)
extractdte (Extract date)
px_open (Open price)
px_low (low price)
px_high (high price)
domth (Day of month)
doyr (Day of year)
pchg_sup_min0 (% price above support for the day)
pchg_floor_min0 (% price above support for current month)
ma_sup_pchg20 (% price above 20-Day average of support)
days_since_sup_lt1 (Days since price was less than or equal to 1% above support)
pchg_ceil_max0 (% price below resistance for current month)
pchg_res_max0 (% price below resistance for the day)
pchg_res_max1 (% price below max resistance for previous 1 month)
days_since_res_lt1 (Days since price was less than or equal to 1% below resistance)
pt_a_sup1 (Points above min support for previous 1 month)
pchg_sup_min1 (% price above min support for previous 1 month)
ma_res_pchg20 (% price below 20-Day average of resistance)
Res_sup_ratio (Resistance/Support ratio)
pt_below_ceil (Points below resistance)
pchg_px_max0 (% price below maximum price for current month)
pchg_px_min0 (% price above minimum price for current month)
pchg_px_avg0 (% price above average price for current month)
pcnt_high_open (% high above open for the day)
pt_open_close (Points gained/dropped from open to close for the day)
pt_last_prevhigh (Points last above/below previous day's high)
pt_last_prevclose (Points last above/below previous day's close)

pt_open_prevopen5 (Points open above/below previous 5th day's open)
Chg10D (Points close above/below previous 10th day's close price)
pcnt_high_low (% high above low for the day)
pcnt_open_low (% open above low for the day)
Pcnt_last_below_1high (% close below 1 month's high)
pcnt_open_prevclose (% open above/below previous day's close)
close_open_ratio (Close/Open prices ratio)
days_since_hvolbuy (Days since maximum of 1 month's volume associated with green days)
days_since_hvolsell (Days since maximum of 1 month's volume associated with red days)
days_since_mvavg_smashed (Days since moving average lines are all smashed togeher)
atr_9d_n (9-Days Average True Range)
rsi_9d_n (9-Days Relative Strength Indicator)
rsi_9d_pchg1 (1-Day % change in 9-Days Relative Strength Indicator)
vix (Volatility Index value)
vix_pchg1 (1-Day % change in Volatility Index value)
vix_pchg2 (2-Days % change in Volatility Index value)
vix_pchg5 (5-Days % change in Volatility Index value)
SPY_chg5d (5-Days Points Change in the S&P500 ETF value)
SPY_ds_bull_advance (Days since the S&P500 ETF entered an advance stage)
SPY_rsi_9d (9-Days Relative Strength Indicator for the S&P500 ETF)
SPY_rsi_9d_p1 (1-Day % change in 9-Days RSI for the S&P500 ETF)
SPY_rsi_9d_p2 (2-Days % change in 9-Days RSI for the S&P500 ETF)
SPY_5dmvavg_pchg1 (1-Day % change in 5-day moving average for the S&P500 ETF)
SPY_ds_bull_jump (Day since the S&P500 ETF jumped over a moving average line)
SPY_ds_bull_Xv (Day since 2 S&P500 ETF moving average lins crossed)
SPY_p_resist (% S&P500 ETF below resistance for the day)
SPY_p_support (% S&P500 ETF above support for the day)
SPY_dcr (Daily closing range for the S&P500 ETF)
SPY_p_50dmvavg (% S&P500 ETF above 50-day moving average)
SPY_5dmvavg_pchg2 (2-Days % change in 5-day moving average for the S&P500 ETF)
SPY_ds_bbl (Days since the S&P500 ETF below lower Bollinger Band)
SPY_ds_bull_3L2H (Days since 3 lows and 2 highs pattern in the S&P500 ETF)
SPY_pcnt_hvol_buysell (Days since highest buying volume in the S&P500 ETF)
SPY_bbwidth (Bollinger Bands width for the S&P500 ETF)
days_to_fomc (Days to FOMC meeting)
days_to_jobs (Days to jobs report)
days_to_cpi (Days to CPI report)
days_to_pce (Days to PCE report)
pcnt_chg_n (Day % change in price)
GR (Green-Red pattern)
Pcnt_stoch3 (3-Day % Stochastic)
mth (Month)
dcr_pchg2 (2-Days % change in Daily Closing Range)
mvavg_10d_pchg2 (2-Days % change in 10-days moving average)
Fib_382pcnt_n (Fibonacci 38.2%)
macd_9d_n (9-Days MACD)
MFM (Accumulation/Distribution Indicator)
mvavg_5d_pchg5 (5-Days % change in 5-days moving average)
mvavg_10d_pchg1 (1-Day % change in 10-days moving average)
Pcnt_Above_mvavg_5d (% above 5-days moving average)
Pcnt_Above_mvavg_10d (% above 10-days moving average)
Pcnt_5d_mvavg_10d (% 5-days moving average above 10-days moving average)
signal_len_ratio (Ratio of number of bullish to bearish patterns)

Target Variable:

anal00_daytrade (Next day class of price action: 1 - Bullish; 0 - Bearish)

Inclusion Criteria: The analysis targeted stocks from companies with a market capitalization of at least $1 billion. Additional requirements were:

1. Excluded data from 10 days before to 30 days after a stock split.
2. The stock must have traded at or above $25 at least once in the past 2 years.
3. A minimum 40% increase in trading volume over the last 30 days.

Data Preprocessing: Included cleaning, handling duplicates and missing data, visualization, and feature engineering including one-hot encoding of categorical features.

Model Training and Evaluation

Hyperparameter Tuning: Explored the following parameter grids:

1. **KNN:** n_neighbors values of 3, 5, 7, 10, 20, 30.
2. **Logistic Regression:** C values of 0.001, 0.01, 0.1, 1, 10, 100; Penalty: l1, l2, none and Solver: liblinear.
3. **Random Forest:** n_estimators values of 5, 7, 10, 50, 100; criterion: gini and entropy; random_state values of 20, 30; max_depth values of None,2, 3, 4, 5, 6, 10; min_samples_split values of 2, 5; min_samples_leaf values of 2,4,5; and bootstrap; True, False.
4. **Gradient Boosting:** n_estimators values of 5, 7, 10, 50, 100; learning rate values of 0.001, 0.01, 0.1, 0.2; max_depth values of 3, 4, 5, 6, 10; min_samples_split values of 2, 5; min_samples_leaf values of 1, 2, 4.

Used GridSearchCV with 5-fold cross-validation to optimize hyperparameters and measured training duration with Python’s time.time() function.

Predictions and Performance Metrics:

1. **Prediction Scores:** Computed for training and test sets.
2. **Accuracy Scores:** Evaluated for training and test datasets.
3. **Classification Report:** Detailed precision, recall, and F1-score for test sets.
4. **Confusion Matrix:** Visualized model performance.
5. **ROC-AUC Score:** Assessed performance on imbalanced classes.

The final model was optimized with the best parameters to predict the stock price movement for the following day.

Results

Model Comparison

```
Model: KNN
Train Accuracy: 0.8635
Test Accuracy: 0.7297
ROC-AUC Score: 0.7874
Fit Time: 7.4480 seconds
Best CV score: 0.7327
Best Parameters: {'n_neighbors': 3}
Classification Report (Test):
              precision    recall  f1-score   support

         0.0       0.75      0.76      0.75      6827
         1.0       0.71      0.70      0.70      5724

    accuracy                           0.73     12551
   macro avg       0.73      0.73      0.73     12551
weighted avg       0.73      0.73      0.73     12551

Confusion Matrix (Test):
[[5169 1658]
 [1735 3989]]


Model: Logistic Regression
Train Accuracy: 0.7720
Test Accuracy: 0.7681
ROC-AUC Score: 0.8408
Fit Time: 13.5572 seconds
Best CV score: 0.7708
Best Parameters: {'C': 0.01, 'penalty': 'l1', 'solver': 'liblinear'}
Classification Report (Test):
              precision    recall  f1-score   support

         0.0       0.78      0.80      0.79      6827
         1.0       0.75      0.73      0.74      5724

    accuracy                           0.77     12551
   macro avg       0.77      0.77      0.77     12551
weighted avg       0.77      0.77      0.77     12551

Confusion Matrix (Test):
[[5446 1381]
 [1530 4194]]
```

Model: Random Forest

```
Train Accuracy: 1.0000
Test Accuracy: 0.8775
ROC-AUC Score: 0.9466
Fit Time: 4162.7868 seconds
Best CV score: 0.8732
Best Parameters: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 100, 'random_state': 20}
Classification Report (Test):
              precision    recall  f1-score   support

         0.0       0.88      0.90      0.89      6827
         1.0       0.87      0.86      0.86      5724

    accuracy                           0.88     12551
   macro avg       0.88      0.88      0.88     12551
weighted avg       0.88      0.88      0.88     12551

Confusion Matrix (Test):
[[6113  714]
 [ 824 4900]]
```

Model: Gradient Boosting

```
Train Accuracy: 1.0000
Test Accuracy: 0.8837
ROC-AUC Score: 0.9509
Fit Time: 10016.6132 seconds
Best CV score: 0.8830
Best Parameters: {'learning_rate': 0.2, 'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 100}
Classification Report (Test):
              precision    recall  f1-score   support

         0.0       0.89      0.90      0.89      6827
         1.0       0.88      0.87      0.87      5724

    accuracy                           0.88     12551
   macro avg       0.88      0.88      0.88     12551
weighted avg       0.88      0.88      0.88     12551

Confusion Matrix (Test):
[[6126  701]
 [ 759 4965]]
```

Summary of Results:

Model	Train Time (s)	Train Accuracy	Test Accuracy	ROC_AUC
KNN	7.45	0.86	0.73	0.79
Logistic Regression	13.56	0.77	0.77	0.84
Random Forest	4162.79	1.00	0.88	0.95
Gradient Boosting	10016.61	1.00	0.88	0.95

Interpretation:

- **Random Forest (RF)** and **Gradient Boosting (GB)** had the highest test accuracy and ROC-AUC scores, indicating strong performance and the ability to distinguish between classes (Bulls vs Bears) effectively.
- **KNN** showed a significant drop from training to test accuracy, suggesting overfitting.
- **Logistic Regression** performed well with reasonable accuracy and ROC-AUC scores but did not achieve the high performance of the ensemble methods.

Overall, Random Forest and Gradient Boosting were the top-performing models in terms of test accuracy and ROC-AUC scores, while Logistic Regression and KNN had lower performance metrics and were less effective at distinguishing between the classes. Gradient Boosting required the most computing resources.
Random Forest was selected for its outstanding performance, achieving a test accuracy of 88% and a top ROC-AUC score of 95%. It also had the advantage of requiring less than half the training time compared to Gradient Boosting. These metrics highlight its strong generalization to unseen data and its effectiveness in distinguishing between classes. The algorithm excels in capturing intricate relationships and interactions between features. Furthermore, it provides valuable insights into feature importance, making it a robust choice for practical machine learning applications, particularly when computational resources are available.

FITTING THE SELECTED MODEL: RANDOM FOREST

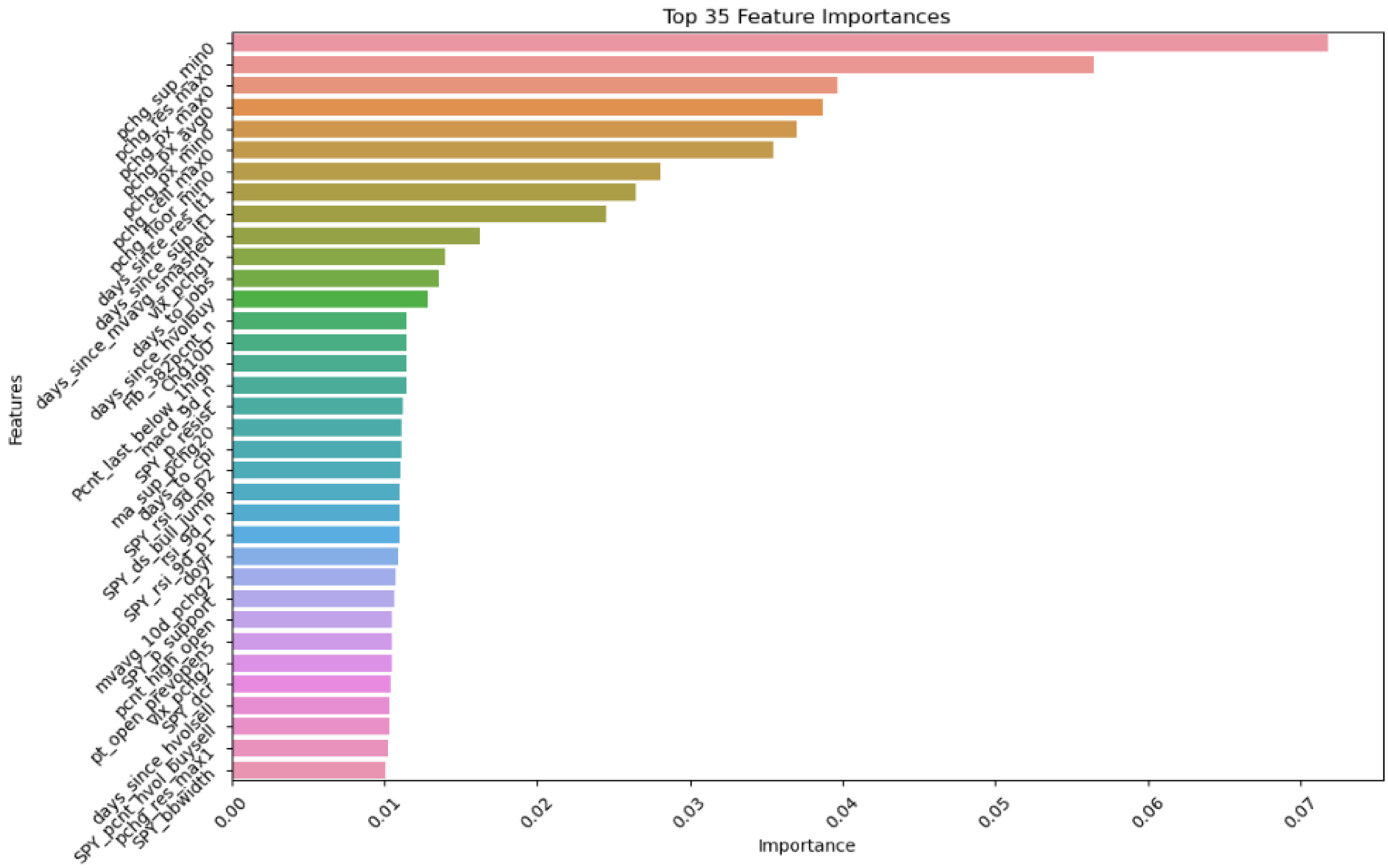
A random forest is essentially an ensemble of decision trees, each with slight variations. The concept behind random forests is that while individual trees might perform well in predicting outcomes, they are prone to overfitting the training data. By combining the predictions from multiple trees (100 in our case), the model mitigates overfitting through averaging, thereby enhancing overall performance (Müller and Guido, 2017).

Random Forest's ability to handle complex, non-linear relationships, its robustness to overfitting and noise, and its provision of feature importance make it a powerful tool for stock market prediction. Its ensemble learning approach ensures stability and accuracy, which are crucial for making reliable predictions in the unpredictable and often volatile financial markets.

Feature Importance

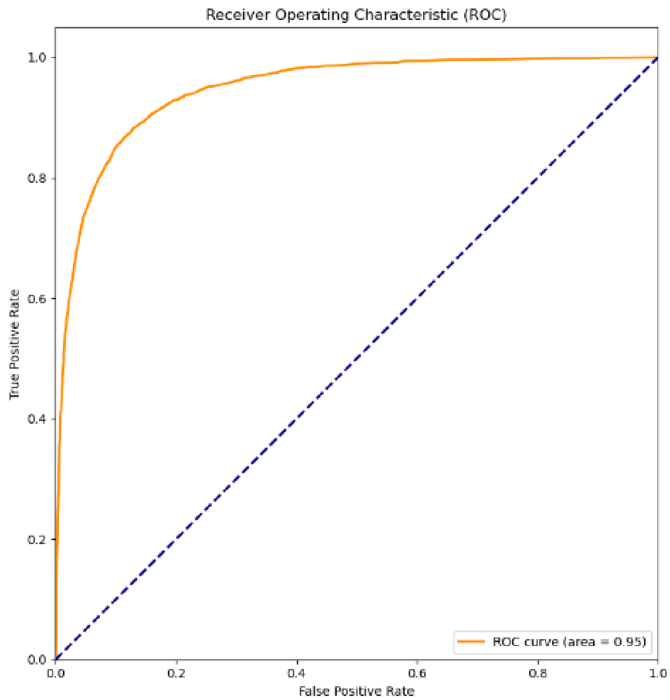
The random forest provided feature importances, which are computed by aggregating the feature importances over the trees in the forest.

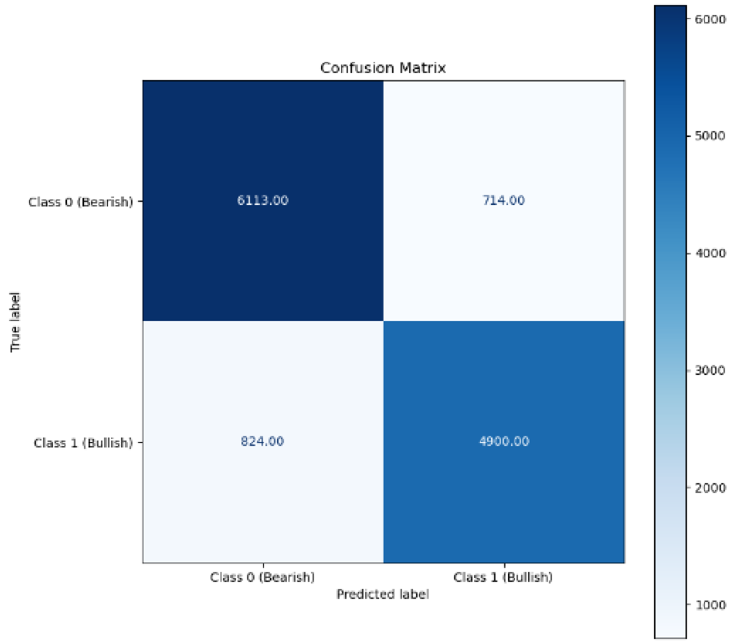
The most significant features identified by the Random Forest classifier include stock price support and resistance levels, price action metrics (such as minimum, maximum, and average prices over the past 30 days), the volatility index, moving averages, the number of days until key economic reports (CPI, PCE, Jobs, and FOMC), the behavior of the S&P 500, and trading volume.



Model Evaluation

```
Classification Report (Training Set):
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     20428
         1.0       1.00      1.00      1.00     17225

    accuracy                           1.00     37653
   macro avg       1.00      1.00      1.00     37653
weighted avg       1.00      1.00      1.00     37653


Classification Report (Test Set):
              precision    recall  f1-score   support

         0.0       0.88      0.90      0.89      6827
         1.0       0.87      0.86      0.86      5724

    accuracy                           0.88     12551
   macro avg       0.88      0.88      0.88     12551
weighted avg       0.88      0.88      0.88     12551
```

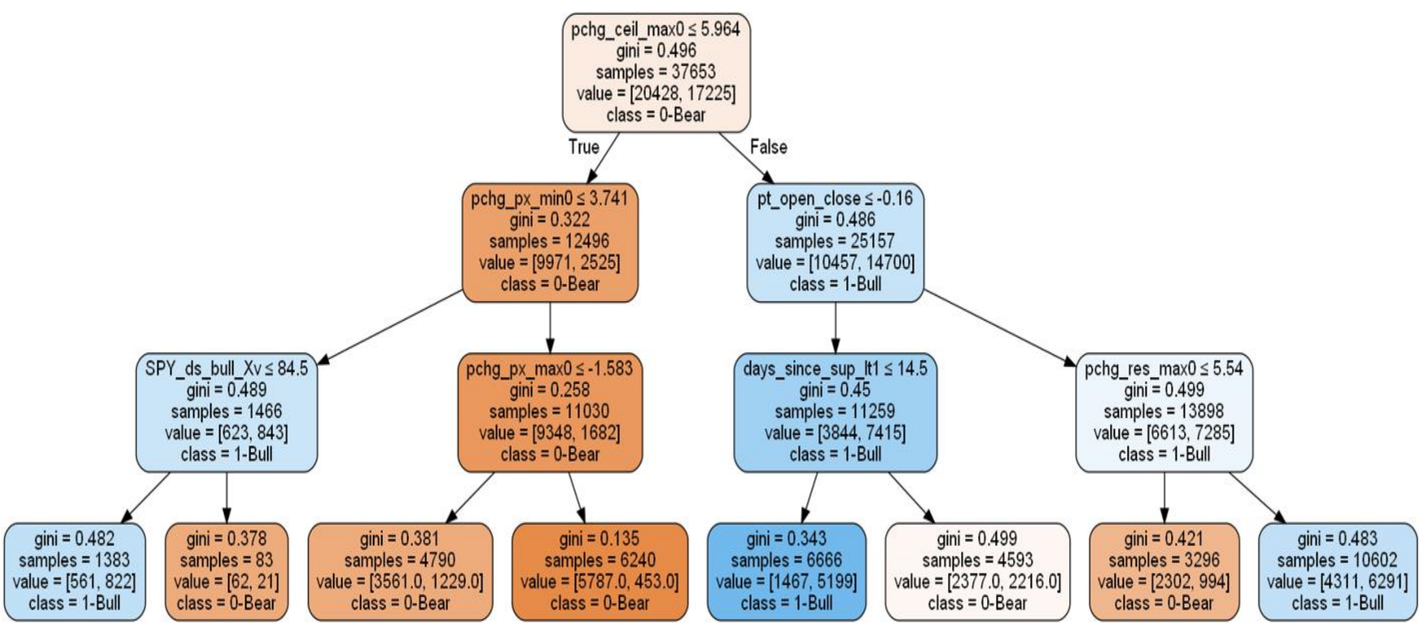




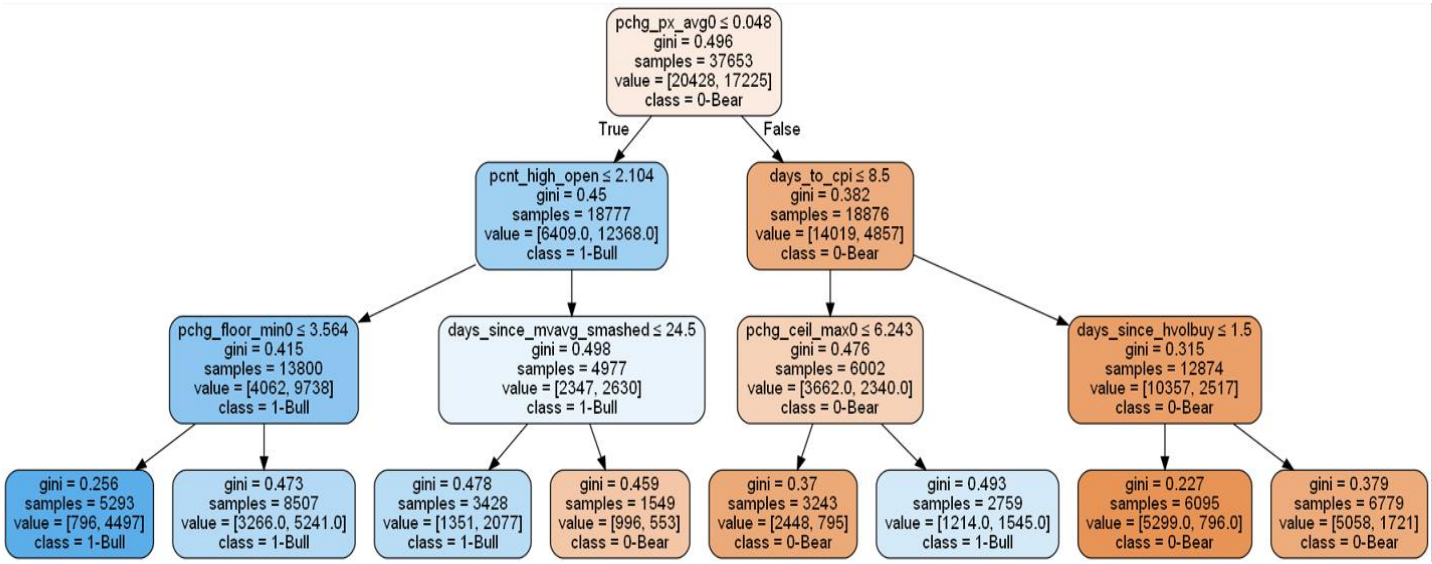
The classification report indicated that the model performed well overall, with high precision, recall, and F1-scores for both classes. The accuracy of 88%, ROC-AUC score of 95% and balanced averages across metrics suggest that the model is effective at distinguishing between the classes and handles class imbalance well, making it a robust model for the given task.

Visualizing a Sample of Individual Decision Trees (out of 100)

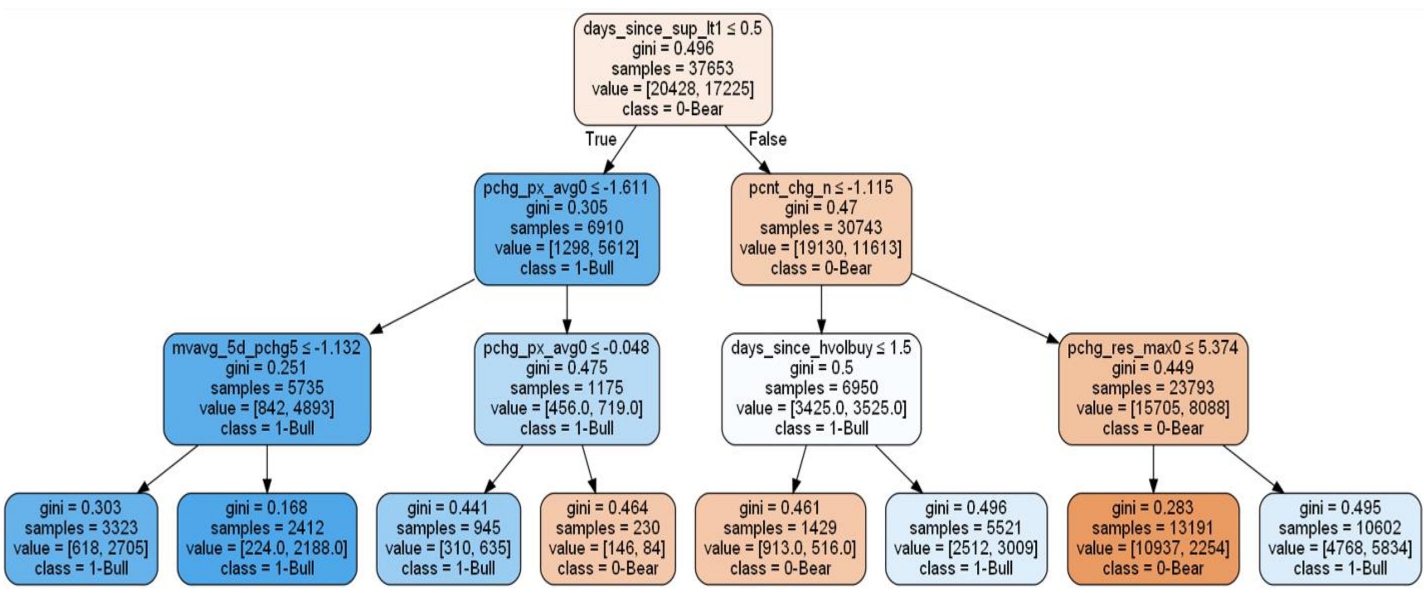
<u>Tree 1</u>



Tree 2



Tree 12



Complex interaction patterns analysis

Decision Trees are adept at uncovering complex interaction patterns among features. These patterns can significantly enhance the performance of traditional linear and logistic regression models by revealing intricate relationships that linear models might overlook. This improvement in understanding can lead to increased predictive accuracy and robustness, ultimately enabling better generalization to new, unseen data.

Here is an example of how to interpret complex interaction patterns from various decision trees within the random forest:

Tree 1

- *pchg_px_min0 interacts with pchg_ceil_max0 to determine the class (1 - Bullish or 0 - Bearish) for the next day. The interaction is further influenced by SPY_ds_bull_Xv.*
- *pchg_px_max0 interacts with pchg_ceil_max0 to determine the class.*
- *pt_open_close directly affects days_since_sup_lt1 to determine the class when pchg_ceil_max0 > 5.96.*
- *pt_open_close interacts with pchg_res_max0 to determine the class outcome.*

Tree 2

- *pchg_px_avg0 and pcnt_high_open: When pchg_px_avg0 ≤ 0.05, pcnt_high_open does not influence the class outcome; the class is always 1.0 regardless of pchg_floor_min0.*
- *pcnt_high_open and days_since_mvavg_smashed: When pchg_px_avg0 ≤ 0.05, days_since_mvavg_smashed impacts the class outcome; the class is 1.0 if days_since_mvavg_smashed ≤ 24.50, otherwise 0.0.*
- *pchg_px_avg0 and days_to_cpi: When pchg_px_avg0 > 0.05, days_to_cpi affects the class outcome based on pchg_ceil_max0; the class is 0.0 if days_to_cpi ≤ 8.50 and pchg_ceil_max0 ≤ 6.24, otherwise 1.0 if pchg_ceil_max0 > 6.24.*
- *days_to_cpi and days_since_hvolbuy: When pchg_px_avg0 > 0.05 and days_to_cpi > 8.50, days_since_hvolbuy determines the class outcome; the class is consistently 0.0 regardless of days_since_hvolbuy.*

Tree 12

- *days_since_sup_lt1 and pcnt_chg_n: When days_since_sup_lt1 is greater than 0.50, the class outcome depends on pcnt_chg_n and days_since_hvolbuy.*
- *pcnt_chg_n and pchg_res_max0: When days_since_sup_lt1 is greater than 0.50, if pcnt_chg_n is greater than -1.12, the class outcome depends on pchg_res_max0.*

The interactions are complex because they involve the interplay of multiple features to determine the final class, rather than the outcome being determined by a single feature in isolation.

Conclusion

In summary, our analysis found that the Random Forest classifier outperformed the traditional Logistic Regression model and other machine learning techniques. It proved to be more efficient, utilizing less than half the computational resources required by Gradient Boosting, and achieved an accuracy rate of 88% in predicting the next day's stock price movement. The study successfully identified key features influencing price action, such as support and resistance levels, price metrics (minimum, maximum, average prices over the past 30 days), the volatility index, moving averages, the timing of key economic reports (CPI, PCE, Jobs, and FOMC), the behavior of the S&P 500, and trading volume.

According to the Efficient Market Hypothesis (EMH), financial markets are considered "informationally efficient," meaning asset prices reflect all available information—both public and private—at any given time. As our analysis is based on historical data, there is inherent risk involved, and financial losses can occur without proper research and risk management. Short-term investors and traders often seek to profit from market anomalies, misinterpreted information, and behavioral biases. Therefore, passively managed index funds are frequently recommended for those seeking to match market returns rather than exceed them.

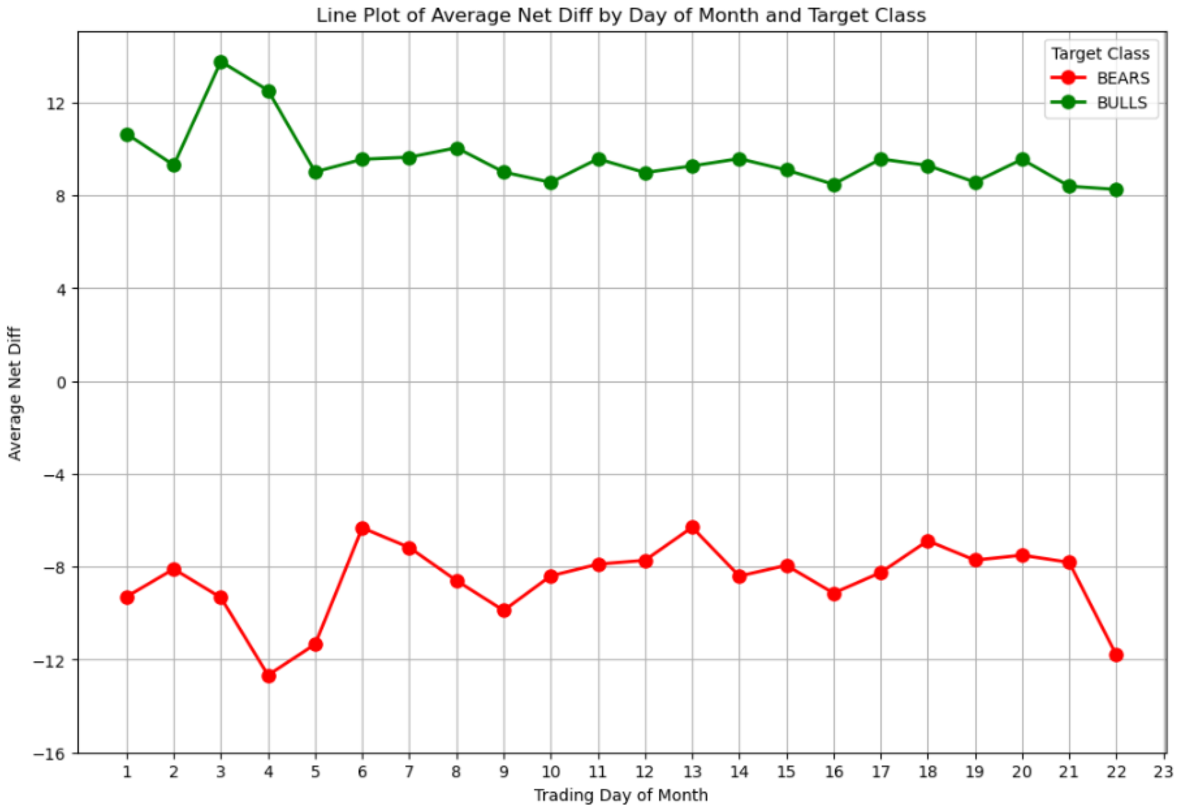
Furthermore, continued research into complex feature interactions is crucial for advancing predictive modeling and machine learning. By exploring how different features interact, researchers and practitioners can develop models that are not only more accurate but also more interpretable and applicable to real-world scenarios.

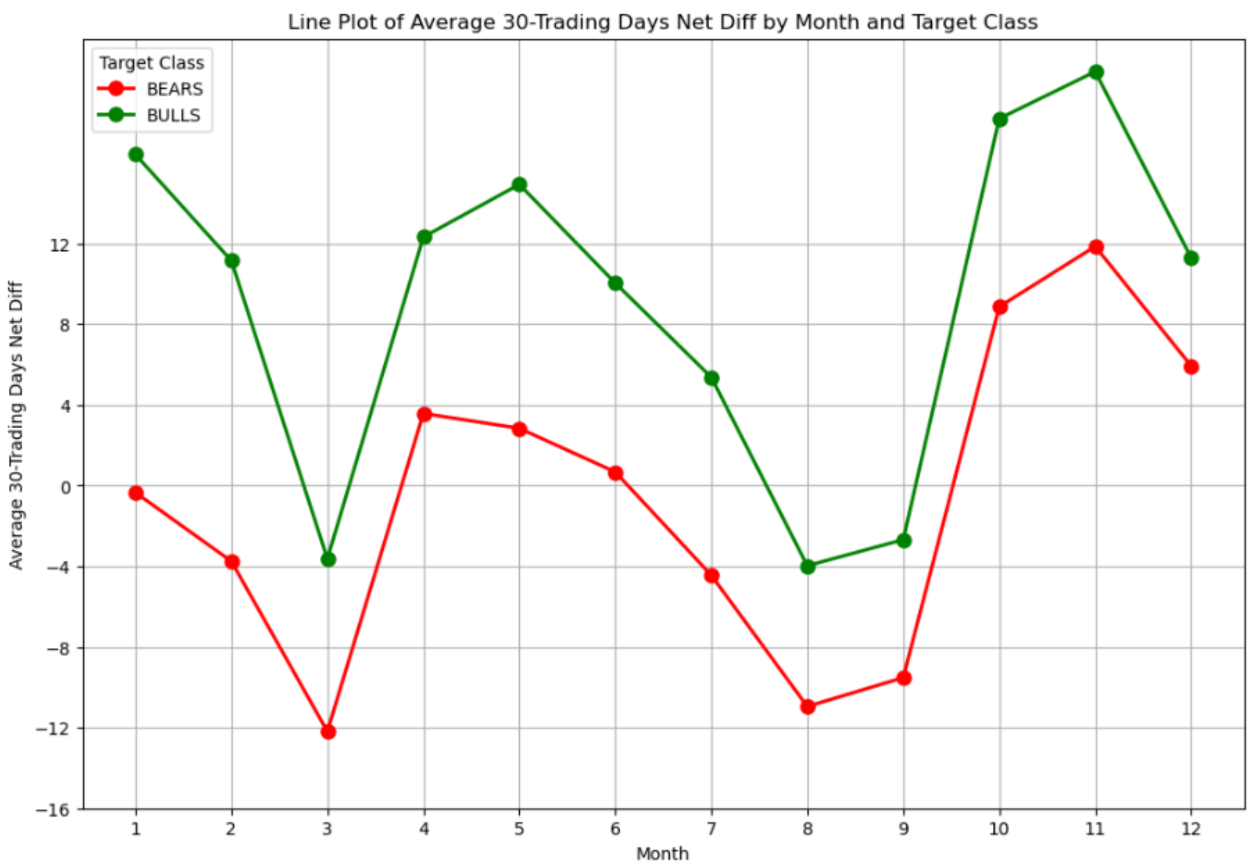
DISCLAIMER: The results and conclusions drawn from this study are based on historical data and machine learning models. Past performance is not indicative of future results, and investing in financial markets carries risks. It is important to conduct thorough research and implement robust risk management strategies before making investment decisions.

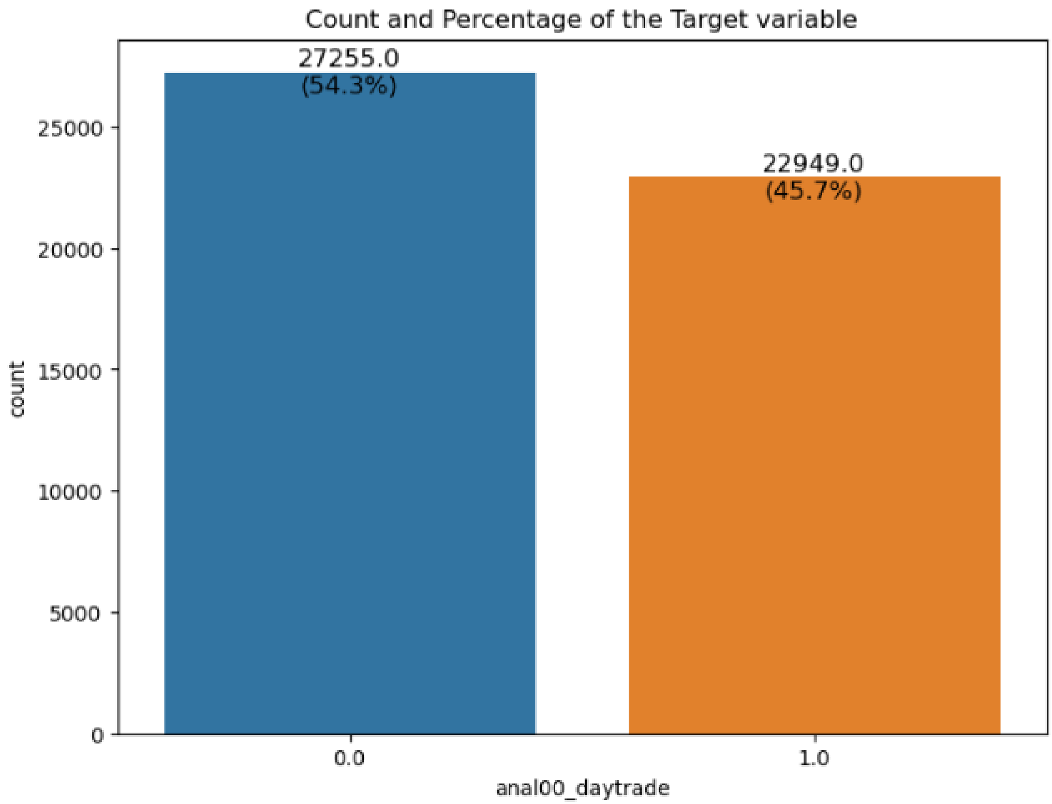
References

- Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.
- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189-1232.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Morgan Kaufmann.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.
- Müller, Andreas C., and Sarah Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 2017.

APPENDICES


Line Plot of Average Net Diff by Day of Month and Target Class
Target Class
BEARS
BULLS
Average Net Diff
Trading Day of Month
12
8
4
0
−4
−8
−12
−16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23


Line Plot of Average 30-Trading Days Net Diff by Month and Target Class
Target Class
BEARS
BULLS
Average 30-Trading Days Net Diff
Month
12
8
4
0
−4
−8
−12
−16
1 2 3 4 5 6 7 8 9 10 11 12


Count and Percentage of the Target variable
27255.0
(54.3%)
22949.0
(45.7%)
count
anal00_daytrade
0.0
1.0


Heatmap of Feature Correlations Above Threshold

Predictions based on Random Forest Classifier

Machine learning and AI are often described as a "black box" because many algorithms function in ways that make it hard to grasp how they reach their decisions or predictions.

symbol	extractdte	domth	px_open	px_low	px_high	last	last_diff1	net_diff1	net_diff2	Target_n	Target	Target_pred
NVDA	12jul2024	9	128.3	127.2	131.9	129.24	1.260002	-2.119995	-4.950005	NaN	NaN	0.0
NVDA	15jul2024	10	130.6	127.2	131.4	128.44	1.779999	-2.080002	-12.319990	NaN	NaN	0.0
NVDA	16jul2024	11	128.4	124.6	129.0	126.36	1.269997	-3.360001	0.389999	NaN	NaN	0.0
NVDA	17jul2024	12	121.4	116.7	121.8	117.99	4.529999	-0.760002	0.010002	NaN	NaN	0.0
NVDA	18jul2024	13	121.8	116.6	122.4	121.09	1.250000	-2.419998	4.959999	NaN	NaN	0.0
NVDA	19jul2024	14	120.4	117.4	121.6	117.93	3.720001	3.230003	2.860001	NaN	NaN	1.0
NVDA	22jul2024	15	120.4	119.9	124.1	123.54	1.910004	-0.190002	-11.250000	-11.2	0.0	0.0
NVDA	23jul2024	16	122.8	122.1	124.7	122.59	0.809998	-4.919998	-10.460000	-10.5	0.0	0.0
NVDA	24jul2024	17	119.2	113.4	120.0	114.25	5.979996	-0.760002	4.849998	NaN	NaN	0.0
NVDA	25jul2024	18	113.0	106.3	116.6	112.28	1.479996	-3.130005	-0.199997	NaN	NaN	0.0
NVDA	26jul2024	19	116.2	111.6	116.2	113.06	2.589996	-2.100006	-13.050000	NaN	NaN	0.0
NVDA	29jul2024	20	113.7	111.3	116.3	111.59	1.190002	-7.789993	14.689990	-7.8	0.0	0.0
NVDA	30jul2024	21	111.5	102.5	112.0	103.73	6.139999	4.119995	-2.249992	NaN	NaN	1.0
NVDA	31jul2024	22	112.9	110.9	118.3	117.02	2.630005	-8.320000	-16.880000	-16.9	0.0	0.0
NVDA	01aug2024	1	117.5	106.8	120.2	109.21	5.899994	3.509995	-15.990000	NaN	NaN	0.0
NVDA	02aug2024	2	103.8	101.4	108.7	107.27	11.350010	9.980011	14.160000	14.2	1.0	1.0
NVDA	05aug2024	3	92.1	90.7	103.4	100.45	3.870003	0.580009	-0.769997	NaN	NaN	0.0
NVDA	06aug2024	4	103.8	100.6	107.7	104.25	0.990005	-8.899994	-4.470009	-8.9	0.0	0.0
NVDA	07aug2024	5	107.8	98.7	108.8	98.91	7.450005	2.970001	7.010002	7.0	1.0	1.0
NVDA	08aug2024	6	102.0	97.5	105.5	104.97	1.320000	-0.889999	7.300003	NaN	NaN	0.0
NVDA	09aug2024	7	105.6	103.4	106.6	104.75	4.750000	4.690002	9.970001	10.0	1.0	1.0
NVDA	12aug2024	8	106.3	106.3	111.1	109.02	4.559998	3.699997	4.859993	NaN	NaN	1.0
NVDA	13aug2024	9	112.4	111.6	116.2	116.14	4.010002	-0.449997	8.040001	NaN	NaN	1.0
NVDA	14aug2024	10	118.5	114.1	118.6	118.08	5.389999	4.099998	5.470001	NaN	NaN	1.0
NVDA	15aug2024	11	118.8	117.5	123.2	122.86	3.400002	2.639999	7.239998	NaN	NaN	1.0
NVDA	16aug2024	12	121.9	121.2	125.0	124.58	6.580002	5.720001	2.350006	5.7	1.0	1.0
NVDA	19aug2024	13	124.3	123.4	130.0	130.00	1.480011	-1.149994	0.240006	NaN	NaN	0.0
NVDA	20aug2024	14	128.4	125.9	129.9	127.25	2.030006	1.370010	-2.160011	NaN	NaN	0.0
NVDA	21aug2024	15	127.3	126.7	129.4	128.50	0.729996	-6.280006	0.970009	-6.3	0.0	0.0
NVDA	22aug2024	16	130.0	123.1	130.8	123.74	4.149994	3.509995	0.809990	NaN	NaN	0.0
NVDA	23aug2024	17	125.9	125.2	129.6	129.37	2.089996	-3.110008	-2.550003	NaN	NaN	0.0
NVDA	26aug2024	18	129.6	124.4	131.3	126.46	4.420006	3.250000	-2.109993	NaN	NaN	0.0
NVDA	27aug2024	19	125.0	123.9	129.2	128.30	2.970001	-2.509995	-9.830002	-9.8	0.0	0.0
NVDA	28aug2024	20	128.1	122.6	128.3	125.61	3.070000	-3.770004	-2.169998	-3.8	0.0	0.0
NVDA	29aug2024	21	121.4	116.7	124.4	117.59	2.220001	-0.159996	-14.460000	-14.5	0.0	0.0
NVDA	30aug2024	22	119.5	117.2	121.8	119.37	0.709999	-8.010002	-6.110001	-8.0	0.0	0.0
NVDA	03sep2024	1	116.0	107.3	116.2	108.00	7.859993	6.569992	-2.979996	NaN	NaN	1.0
NVDA	04sep2024	2	105.4	104.1	113.3	106.21	4.660004	4.430008	-5.310005	NaN	NaN	1.0
NVDA	05sep2024	3	105.0	104.8	109.6	107.21	1.880005	-5.209999	NaN	-5.2	0.0	1.0
NVDA	06sep2024	4	108.0	101.0	108.2	102.83	NaN	NaN	NaN	NaN	NaN	1.0

The predicted Target class is for the following trading day.

symbol	extractdte	domth	px_open	px_low	px_high	last	last_diff1	net_diff1	net_diff2	Target_n	Target	Target_pred
SMCI	26jul2024	19	710.6	697.6	724.3	712.19	9.280029	-23.000000	-59.010010	-59.0	0.0	0.0
SMCI	29jul2024	20	720.7	692.3	730.0	697.73	10.000000	-38.690000	49.060000	49.1	1.0	1.0
SMCI	30jul2024	21	705.0	656.3	707.0	666.31	10.560000	-8.349976	-29.339970	NaN	NaN	1.0
SMCI	31jul2024	22	710.0	691.8	720.6	701.65	19.890010	-32.320010	-141.990000	-142.0	0.0	0.0
SMCI	01aug2024	1	704.6	658.6	724.4	672.24	42.190000	-5.349976	-63.200010	-63.2	0.0	0.0
SMCI	02aug2024	2	630.0	582.5	637.0	624.65	91.149960	85.059940	61.370000	85.1	1.0	1.0
SMCI	05aug2024	3	535.6	529.5	626.7	608.83	28.120000	0.929993	-139.910000	-139.9	0.0	0.0
SMCI	06aug2024	4	616.0	588.8	628.8	616.94	14.539980	-39.760010	-28.700010	-39.8	0.0	0.0
SMCI	07aug2024	5	532.5	488.9	547.0	492.70	31.620000	13.179990	-0.179962	NaN	NaN	1.0
SMCI	08aug2024	6	497.0	478.6	528.6	509.94	16.720000	-1.630005	67.469970	NaN	NaN	0.0
SMCI	09aug2024	7	510.4	492.0	515.0	508.76	52.890010	52.250030	58.030000	58.0	1.0	1.0
SMCI	12aug2024	8	511.1	510.5	564.0	540.98	20.029970	4.819946	-0.850037	NaN	NaN	1.0
SMCI	13aug2024	9	563.8	548.6	583.9	567.43	30.740050	-7.029968	78.290040	78.3	1.0	1.0
SMCI	14aug2024	10	584.1	546.4	585.3	577.09	53.500000	42.210020	41.859990	42.2	1.0	1.0
SMCI	15aug2024	11	584.5	573.2	636.8	626.69	16.729980	4.789978	-26.729980	-26.7	0.0	0.0
SMCI	16aug2024	12	624.0	612.1	639.7	628.80	28.339970	1.619995	3.809998	NaN	NaN	0.0
SMCI	19aug2024	13	622.0	595.3	629.8	623.62	13.760010	-5.830017	1.650024	NaN	NaN	0.0
SMCI	20aug2024	14	616.7	598.4	630.5	610.91	22.000000	10.890010	2.519958	NaN	NaN	0.0
SMCI	21aug2024	15	612.9	601.8	628.8	623.78	2.580017	-24.549990	-2.190002	-24.5	0.0	0.0
SMCI	22aug2024	16	629.4	602.2	630.8	604.82	15.829960	8.559937	-57.440000	NaN	NaN	1.0
SMCI	23aug2024	17	611.2	603.9	627.0	613.24	8.179993	-47.489990	-102.920000	-102.9	0.0	0.0
SMCI	26aug2024	18	610.0	555.2	618.2	562.51	38.099980	32.699950	-161.820000	NaN	NaN	1.0
SMCI	27aug2024	19	518.9	513.5	557.0	547.64	48.310000	-42.450010	28.039980	-42.5	0.0	0.0
SMCI	28aug2024	20	485.9	395.2	487.5	443.49	39.449980	38.929960	-31.840000	38.9	1.0	1.0
SMCI	29aug2024	21	435.9	435.4	475.3	448.82	18.150020	-20.269990	14.260010	-20.3	0.0	0.0
SMCI	30aug2024	22	458.0	419.6	459.3	437.70	34.350010	33.010010	-49.289980	NaN	NaN	0.0
SMCI	03sep2024	1	430.0	428.7	464.4	441.78	15.300020	3.580017	-15.060030	NaN	NaN	0.0
SMCI	04sep2024	2	420.1	408.4	435.4	423.47	10.579990	6.059967	-39.130000	-39.1	0.0	1.0
SMCI	05sep2024	3	411.4	406.8	421.9	414.60	4.690002	-15.540010	NaN	-15.5	0.0	1.0
SMCI	06sep2024	4	402.0	382.8	406.7	386.46	NaN	NaN	NaN	NaN	NaN	1.0

symbol	extractdte	domth	px_open	px_low	px_high	last	last_diff1	net_diff1	net_diff2	Target_n	Target	Target_pred
SPY	26jul2024	19	542.3	541.5	547.2	544.44	2.040039	-1.260010	-3.914917	NaN	NaN	0.0
SPY	29jul2024	20	546.0	542.7	547.0	544.76	3.484985	-4.260010	14.984990	NaN	NaN	0.0
SPY	30jul2024	21	546.3	538.5	547.3	542.00	4.520020	3.120056	-6.780029	NaN	NaN	0.0
SPY	31jul2024	22	549.0	547.6	553.5	550.81	3.580017	-9.559998	-26.270020	-26.3	0.0	0.0
SPY	01aug2024	1	552.6	539.4	554.9	543.01	4.300049	-2.849976	-26.720000	NaN	NaN	0.0
SPY	02aug2024	2	535.8	528.6	537.0	532.90	11.940000	10.569980	13.769990	13.8	1.0	1.0
SPY	05aug2024	3	511.6	510.3	523.6	517.38	10.530030	9.180054	2.020020	9.2	1.0	1.0
SPY	06aug2024	4	519.2	517.9	529.8	522.15	3.120056	-9.809998	3.489990	NaN	NaN	0.0
SPY	07aug2024	5	528.5	518.0	531.6	518.66	8.809998	6.740051	9.940002	9.9	1.0	1.0
SPY	08aug2024	6	523.9	521.8	531.3	530.65	4.700012	3.450012	3.609985	NaN	NaN	0.0
SPY	09aug2024	7	529.8	528.6	534.5	532.99	2.320007	-0.940002	11.330020	NaN	NaN	1.0
SPY	12aug2024	8	534.2	531.0	535.7	533.27	5.759949	5.509949	6.519958	6.5	1.0	1.0
SPY	13aug2024	9	536.5	536.3	542.3	542.04	3.630005	0.900024	13.239990	NaN	NaN	1.0
SPY	14aug2024	10	542.8	540.1	545.0	543.75	4.190002	3.570007	4.040039	NaN	NaN	1.0
SPY	15aug2024	11	549.5	548.9	553.4	553.07	3.600037	3.440063	7.189941	7.2	1.0	1.0
SPY	16aug2024	12	551.4	551.3	555.0	554.31	5.750000	4.880005	4.700073	4.9	1.0	0.0
SPY	19aug2024	13	554.7	553.9	559.6	559.61	1.690002	-0.450012	-1.330078	NaN	NaN	0.0
SPY	20aug2024	14	559.2	557.3	560.8	558.70	5.890015	0.849976	1.320007	NaN	NaN	0.0
SPY	21aug2024	15	559.8	554.7	562.1	560.62	1.239990	-6.340027	2.220032	NaN	NaN	0.0
SPY	22aug2024	16	562.6	555.0	563.2	556.22	4.840027	2.599976	2.579956	NaN	NaN	1.0
SPY	23aug2024	17	559.5	557.3	563.1	562.13	1.739990	-2.390015	-2.579956	NaN	NaN	0.0
SPY	26aug2024	18	563.2	559.0	563.9	560.79	3.239990	2.070007	-3.690002	NaN	NaN	0.0
SPY	27aug2024	19	559.5	558.3	562.1	561.56	3.260010	-2.910034	4.169983	NaN	NaN	0.0
SPY	28aug2024	20	561.2	555.0	561.6	558.30	3.369995	-1.960022	0.480042	NaN	NaN	0.0
SPY	29aug2024	21	560.3	557.2	563.7	558.35	6.539978	2.909973	-11.020020	NaN	NaN	0.0
SPY	30aug2024	22	560.8	557.1	564.2	563.68	2.570007	-8.389954	-6.429993	-8.4	0.0	0.0
SPY	03sep2024	1	560.5	549.5	560.8	552.08	4.229980	3.489990	-2.990051	NaN	NaN	1.0
SPY	04sep2024	2	550.2	549.5	554.4	550.95	2.909973	-1.280029	-9.859985	-9.9	0.0	0.0
SPY	05sep2024	3	550.9	547.1	553.8	549.61	1.659973	-9.580017	NaN	-9.6	0.0	0.0
SPY	06sep2024	4	549.9	539.4	551.6	540.36	NaN	NaN	NaN	NaN	NaN	1.0