École Polytechnique Fédérale de Lausanne

The Effect of Popular Technical Indicators in Price Movement Prediction

FIN-525

Mihai David, Ana-Arina Raileanu

January 24, 2023

Abstract

Automated trading is becoming more and more popular. The rise of new data processing techniques and computing capabilities ensures that trading strategies are continuously evolving. In this study, we assess the importance of typical trading indicators of manual traders, to see whether they can still give a competitive edge in the market of today. We use these trading indicators as input data into various classification models to predict whether the following day will generate positive or negative returns for a stock. The models are assessed on data from the 1960s up to the present time. We limited our scope to the top 100 most traded stocks, and drew conclusions using modern explainability techniques to evaluate the models.

1 Introduction

With creation of mobile apps that have made trading accessible to new types of customers, more and more people have started participating in the market during the recent years. In particular, popular trading apps such as Robinhood[1] have seen a significant increase in daily active users during the pandemic, when most people were looking for new activities to spend their time at home during the lock-downs. Therefore, understanding what sort of indicators to use, and how they affect price movement predictions, could provide a competitive advantage to a trader looking to join a continuously growing market. We focus our study on daily trading data starting in January 1962 up to the present time, with the goal of understanding what are the most useful techniques when predicting whether the price of an asset will increase or decrease.

As a source for our data, we use Yahoo Finance, which contains historical data of financial instruments, aggregated from multiple sources. The data is accessible through an API, and the user can choose the type of data that will be downloaded through the parameters of the downloading function.

Since high trading volumes are often associated with high volatility, particularly for large-cap instruments [1], we focus our research on the top 100 most traded stocks from the United States during the day the data was fetched. Higher volatility is associated with higher risk, as the price of an instrument may change significantly [2]. However, traders can also benefit from volatility if their predictions are in the right direction.

Putting it all together, we extract the tickers of the 100 most active stocks[2], by creating a list of them and querying the Yahoo Finance API for each individual stock, for the maximum period of time available.

2 Exploratory Data Analysis

The data of the 100 stocks with the highest traded volume, extracted from Yahoo Finance, contains the following columns for each instrument:

- Open - the price at the beginning of a trading day.
- High - the highest price achieved during a trading day
- Low - the lowest price achieved during a trading day
- Close - the adjusted price at the end of a trading day
- Volume - the total volume that was traded during a trading day
- Dividends - a value that is different than 0 when a company paid dividends, and represents the amount of dividends per share

[1] https://www.cbinsights.com/research/report/how-robinhood-makes-money/

[2] https://www.tradingview.com/markets/stocks-usa/market-movers-active/

The dataset contains around 600K entries (depending on the most current data that is fetched), and takes around 50 MB of memory. Figure 1 shows two important plots that help us understand the dataset. In the first one, we can see when the top 100 market movers of today first appeared in the Yahoo Finance dataset. A significant part of them first appeared during the last ten years. The second plot helps us understand the behavior of traders throughout different socio-economic periods. Several spikes correspond to rare market events, such as the financial crisis of 2008 (the largest mean yearly volume traded) and the COVID-19 pandemic in 2020, two periods that were marked by large volatility.

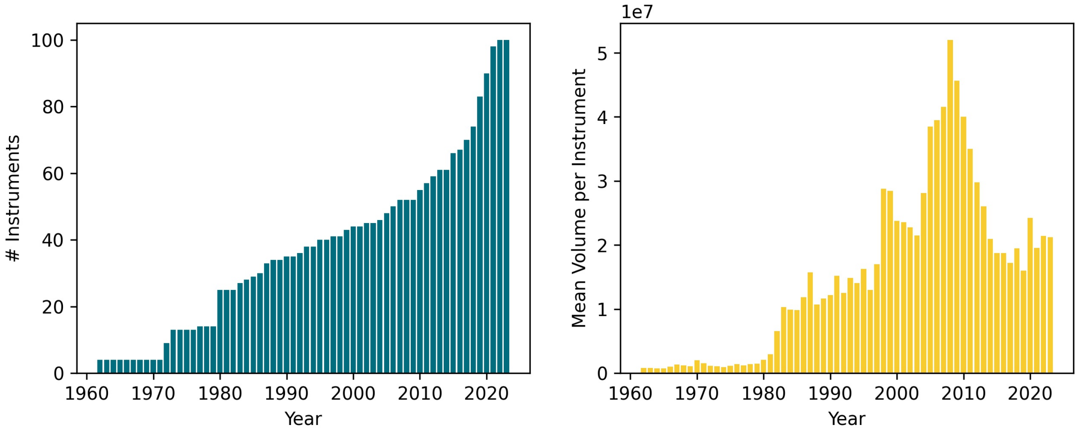

Figure 1: In the first plot, the number of instruments of the top 100 movers of the current times is related to the year of their first appearance in the Yahoo Finance Dataset. In the second plot, the average yearly volume traded per instrument is related to the year.

3 Feature Engineering

Starting from this dataset, we analyze the current literature and enhance it using features that are often used in both manual and automated trading strategies. We group our technical indicators into four categories [3], which are described below.

3.1 Trend Indicators

- **Exponential Moving Average (EMA)** - Represents a type of moving average (MA) that gives the most recent data points more weight and relevance. In comparison to a simple moving average (SMA), which gives equal weight to all observations across the time, an EMA responds more strongly to recent price movements. In the equation below, *Days* is the number of observations (26 in our case) and *Smoothing* is the smoothing factor:

$$EMA_t = Value_t * (\frac{Smoothing}{1 + Days}) + EMA_{t-1} * (1 - \frac{Smoothing}{1 + Days}) \tag{1}$$

3.2 Volume Indicators

- **On-Balance Volume (OBV)** - Based on the idea that a higher traded volume drives the price up, the indicator increases in value by the amount of traded volume when the closing price is higher than the previous day, and decreases by the same amount otherwise. It is a cumulative indicator. Since it is cumulative, a trading day with an unusually high trading volume might have an effect on it over a long time [4]. Its formula is given below:

$$OBV_t = \begin{cases} OBV_{t-1} + Volume_t, & \text{if } Close_t > Close_{t-1} \\ OBV_{t-1}, & \text{if } Close_t = Close_{t-1} \\ OBV_{t-1} - Volume_t, & \text{if } Close_t < Close_{t-1} \end{cases} \tag{2}$$

- **Accumulation/Distribution (AD)** - Represents the difference between the people who buy the stock and those who sell it. Similarly to the OBV, it is based on the assumption that traded volume dictates future price. To compute it, the Money Flow Multiplier (MFM), is summed up over time, after being multiplied with the volume to obtain the Money Flow Volume (MFV). A positive MFM represents that at the end of the day, the closing price was closer to the highest price of the day, than to the lowest one [5]. When the MFM is multiplied by a large volume, there is a stronger tendency to buy the stock. The equations below describe how to compute the indicators.

$$MFM_t = \frac{(Close_t - Low_t) - (High_t - Close_t)}{High_t = Low_t} \tag{3}$$

$$MFV_t = MFM_t \cdot Volume_t \tag{4}$$

$$AD_t = AD_{t-1} + MFV_t \tag{5}$$

- **Volume** & **Previous Day Volume** - The volume that was traded on the current and previous day. The Volume is already present in the dataset, while the Previous Day Volume is the volume shifted by one position.

3.3 Volatility Indicators

- **Average True Range (ATR)** - Shows how much an asset's price moves, on average, during a given time frame [6]. The values of this indicator move up and down as price moves in an asset become larger or smaller. In the formula for ATR given below, TR is the True Range, H is today's high, L is today's low, C_p is yesterday's closing price and n is the number of periods (14 in our case):

$$TR = Max[(H - L), |H - C_p|, |L - C_p|] \tag{6}$$

$$ATR = (\frac{1}{n})\sum_{i}^{n} TR_i \tag{7}$$

3.4 Momentum Indicators

- **Moving Average Convergence/Divergence (MACD)** - The difference in the mean price over a short period of time and a longer period of time. In our case, we chose the short period to be 12 days, a common combination. The mean is computes using an exponentially moving average, which puts more importance on recent price changes than older ones [7]. The formula for the MACD is given below:

$$MACD_t = EWM_{12}([Close_{t-11}, ..., Close_t]) - EWM_{26}([Close_{t-25}, ..., Close_t]) \tag{8}$$

- **Stochastic Oscillator (SO)** - This indicator represents the difference between the closing price of a particular day and the minimum closing price of a 14-day period, divided by the difference between the highest price of that period and the lowest price. Two indicators are created, the second one (%D) being and average over 3 days of the first one (%K) [8]. The signal generated represents overbought/oversold [9].

$$\%K_t = \frac{Close_t - Min([Low_{t-13}, ..., Low_t])}{Max([High_{t-13}, ..., High_t]) - Min([Low_{t-13}, ..., Low_t])} \cdot 100 \tag{9}$$

$$\%D_t = \frac{K_t + K_{t-1} + K_{t-2}}{3} \tag{10}$$

- **Money Flow Index (MFI)** - Represents a technical indicator that generates overbought or oversold signals using both prices and volume data [10]. It is calculated by accumulating the positive and negative Money Flow values and then it creates the money ratio:

$$TypicalPrice = \frac{High + Low + Close}{3} \tag{11}$$

$$RawMoneyFlow = TypicalPrice * Volume \tag{12}$$

$$MoneyFlowRatio = \frac{14PeriodPositiveMoneyFlow}{14PeriodNegativeMoneyFlow} \tag{13}$$

$$MoneyFlowIndex = 100 - \frac{100}{1 + MoneyFlowRatio} \tag{14}$$

- **Relative Strength Index (RSI)** - A momentum oscillator that can point to overbought and oversold securities by measuring the speed and magnitude of recent price evolutions [11]. We use an exponential moving average to calculate the Average Gain and the Average Loss. When computing the Average Gain, the periods with price losses are counted as zero. When computing the Average Loss, the periods with price increases are counted as zero. The formula of RSI is given below:

$$RSI = 100 - \left[\frac{100}{1 + \frac{14DaysAverageGain}{14DaysAverageLoss}}\right] \tag{15}$$

3.5 Clustering

We first compute a log-returns matrix from the existing data. We sort our data according to the *Date* feature and choose a range of records towards the end of our matrix. We than remove all the NaN values, yielding a matrix of 288 records containing 85 different stocks. After using Louvain Correlation Clustering method [12], we end up with 4 clusters containing 39, 34, 10 and 2 stocks, respectively. Finally, we add the clusters found for the specific stocks, as a new feature, to the original data matrix.

3.6 Feature Selection

After removing all of the rows containing NaN values, we split the dataset into a training set and a test set. To do so, we first sort all of their entries by their date. Afterwards, we use the data up to the end of the year 2021 for training and the rest of the data for testing. We split the data chronologically instead of randomly to prevent the models from gaining knowledge about future data.

As a next step, we analyze the correlation between all our of technical indicators. The Pearson correlation between any pair of indicators is shown in Figure 2. We remove the highly-correlated features (coefficient > 0.85), as they can harm training and add biases to the models. Namely, we remove the slow stochastic indicator (%D), which is correlated with the fast indicator, and the Volume traded during the previous day, which is correlated with the Volume traded on that day.

We run one more selection step on the remaining features. Out of the 11 numerical features, we select the 8 best features (2/3), according to the Chi-squared test. As such, the features on which the sign of the next-day returns depends the most are selected. In Figure 3 are plotted the scores of the features that were selected.

4 Models and Methods

The task that we choose to analyze our models on is classification. As such, a model should predict a label of 0 when the closing price of a given day is lower than that of the current day, and 1 otherwise.

$$label_t = \begin{cases} 0, & \text{if } Close_t < Close_{t+1} \\ 1, & \text{otherwise} \end{cases} \tag{16}$$

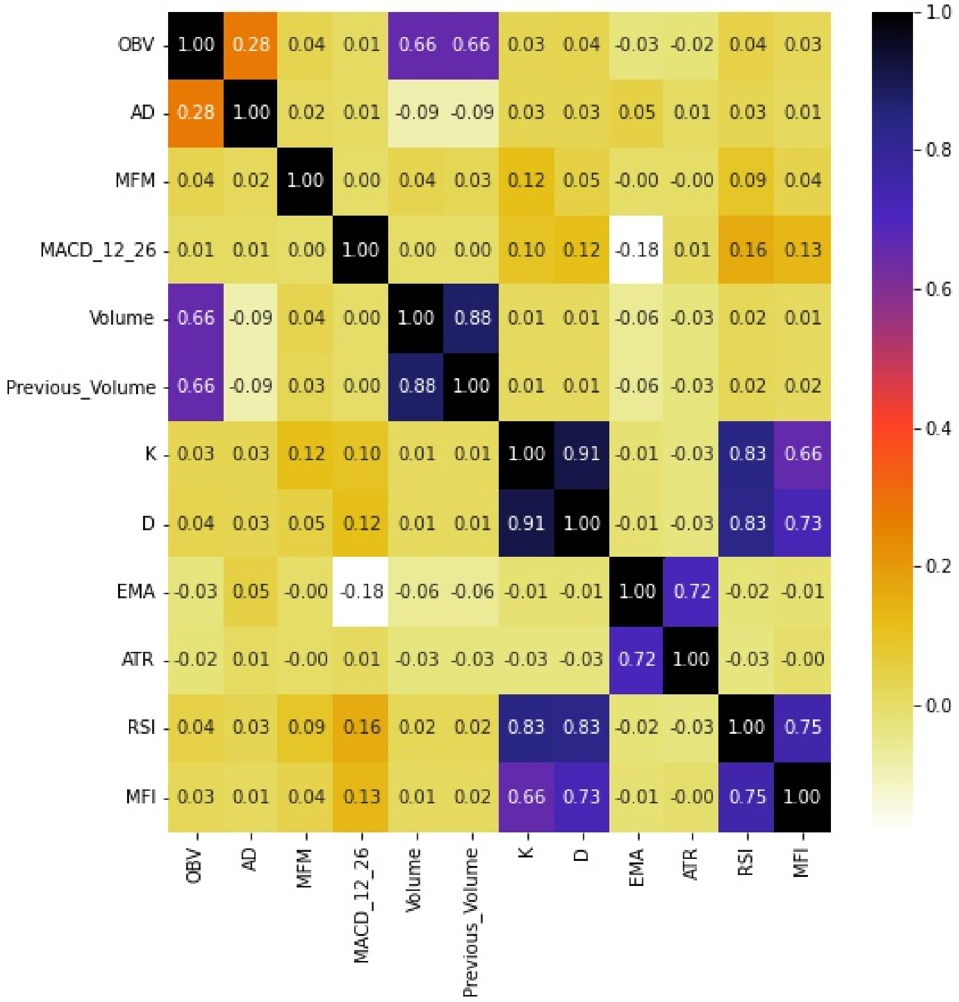

Figure 2: The Pearson Correlation between all technical indicators described.

We tested various classification models and measured their performance on the test data. Below, we list the models with the base parameters that we used:

- **MLPClassifier** - A neural network with one hidden layer and 10 hidden units.
- **Logistic Regression** - Estimates the probability of a class by minimizing the negative log-likelihood.
- **Naive Bayes** - Selects the most probable class label based on how the input features relate to their estimated distributions by class.
- **Decision Tree** - A tree where each node represents a split of a feature. The tree is traversed starting from the root to its leaves, which represent the class labels.
- **CatBoost** - A new approach at using decision trees. CatBoost uses gradient-boosted decision trees, an ensemble of decision trees that build on top of each other, fixing the prediction error of the previous trees [13]. Furthermore, CatBoost was created to work with categorical features, without any additional encoding step.

Additionally, since CatBoost accepts categorical features, we create another model. This model is trained on the same numerical features as all the rest, and a categorical feature representing the cluster to which the asset was assigned. To distinguish between the two models, we refer to the second one as ”CatBoost with Cluster Data”.

Finally, we create four specialized models, one per cluster, to see whether training models only on data from the same cluster might improve their performance. The dataset is split by cluster, and each of the models only

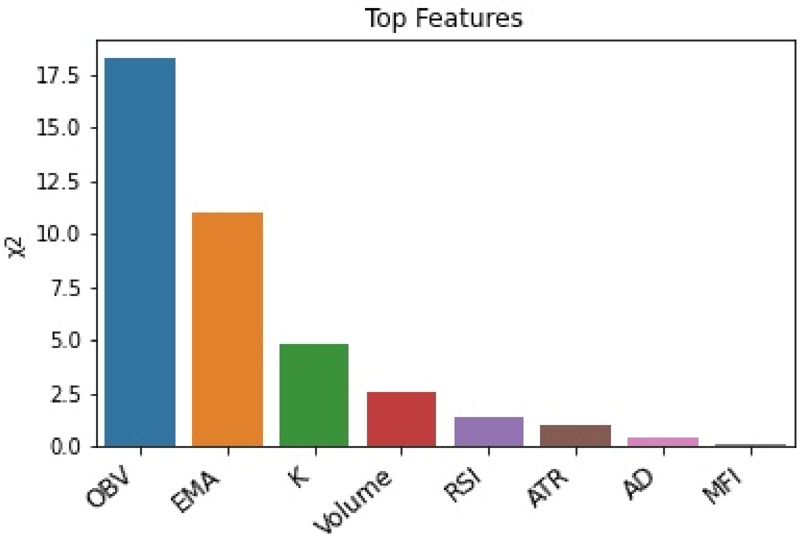

Figure 3: The Chi-squared score of the top features that have the highest correlation to the sign of the next-day returns.

has access to data that belongs to the cluster it is specialized in. For prediction, we first check the cluster of the asset, then use its corresponding model to predict the change in price. The two additional CatBoost models are summarized below:

- **CatBoost with Cluster Data** - Uses the same data as the CatBoost model, with one additional column representing the cluster of the asset.
- **Specialized CatBoost per Cluster** - An ensemble of four CatBoost models, each one of them trained only on data from a single cluster.

5 Results

5.1 Backtesting

We measure the performance of the classification models on historical data. In order to do so, we assume a simple trading strategy, where a prediction of "1" (an expected increase in the closing price during the next day) is associated with buying an asset, whereas a prediction of "0" (the price is expected to decrease) is associated with shorting an asset. The equation below shows how we compute the returns that the model makes (the captured returns) on a given day for a given asset:

$$captured\_returns_t(asset) = \begin{cases} returns_{t+1}(asset), & \text{if the closing price is predicted to increase (label "1")} \\ -returns_{t+1}(asset), & \text{if the closing price is predicted to decrease (label "0")} \end{cases} \tag{17}$$

Afterwards, we group all of the *captured_returns* by day, and we sum them up, to find the returns of an investor that trades all of the available assets. Given a starting date (i.e. the beginning of the train set, or the beginning of the test set), we compute the balance of the investor over time, by cumulatively summing the *captured_returns* per day.

As our baseline, we consider a strategy where the investor always chooses to buy one unit of each stock every day - this strategy models the overall performance of the market.

A comparison between the accuracy and F1-score (the harmonic mean between the precision and recall) is

	F1 Score	Accuracy
MLPClassifier	0.443	0.501
Decision Tree	0.502	0.501
Logistic Regression	0.298	0.513
Naive Bayes	0.083	**0.514**
Standard CatBoost	0.512	0.503
CatBoost with Cluster Data	**0.521**	0.501
Specialized CatBoost per Cluster	0.514	0.499

Table 1: The F1-score and accuracy of the models on the test set.

shown in Table 1. The CatBoost model with information about the clusters has the highest F1-score, which supports out belief that clustering instruments could be beneficial for predictions.

In Figure 4 we can observe the performance of all the models on the train set. The CatBoost models significantly outperforms all the other models. Furthermore, the third best model is a decision tree, suggesting the utility of tree-based models in the current experiment setup.

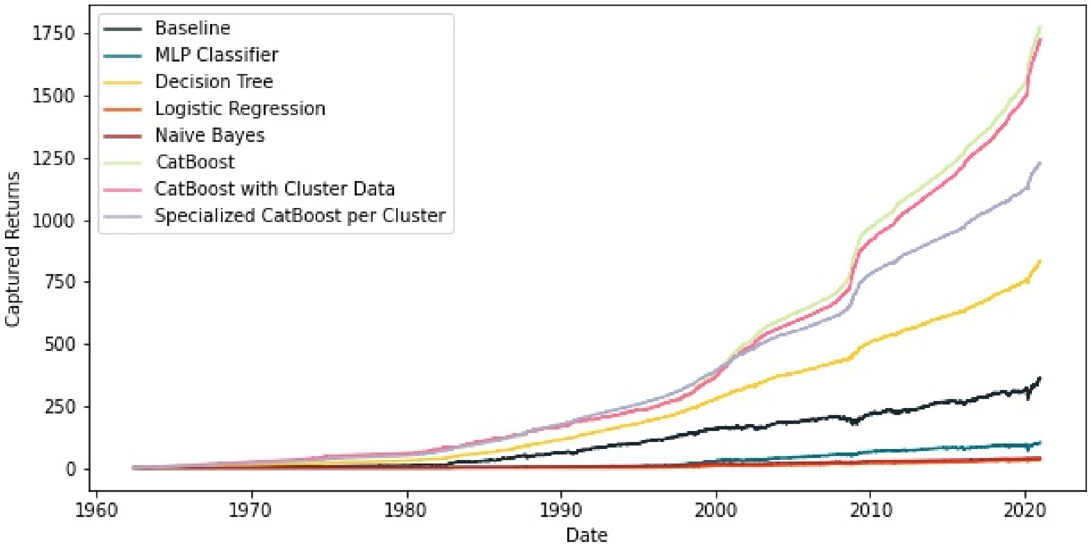

Figure 4: The performance on the train set of all classification models, compared to the Baseline (market performance). The only models that surpassed the baseline were the tree-based architectures.

Figure 5 shows the performance of the models that do not have an architecture based on trees on the test set. By comparison, Figure 6 shows the performance of the models that are based on trees. In 2021, in both plots the Baseline model obtains high profits. This is due to the rise in prices caused by the COVID-19 pandemic. After an initial decrease in March 2020, the stock market has reached record returns. However, this unexpected situation is hard to predict for the models, as no similar event has occurred in the test set. In 2022, the tree-based models manage to not only match, but surpass the performance of the market. The CatBoost model that has information about clusters manages to obtain the highest overall score (8.429) after two years of trading, compared to the Baseline model (0.264). Therefore, the CatBoost model seems to be able to produce profit in normal market conditions.

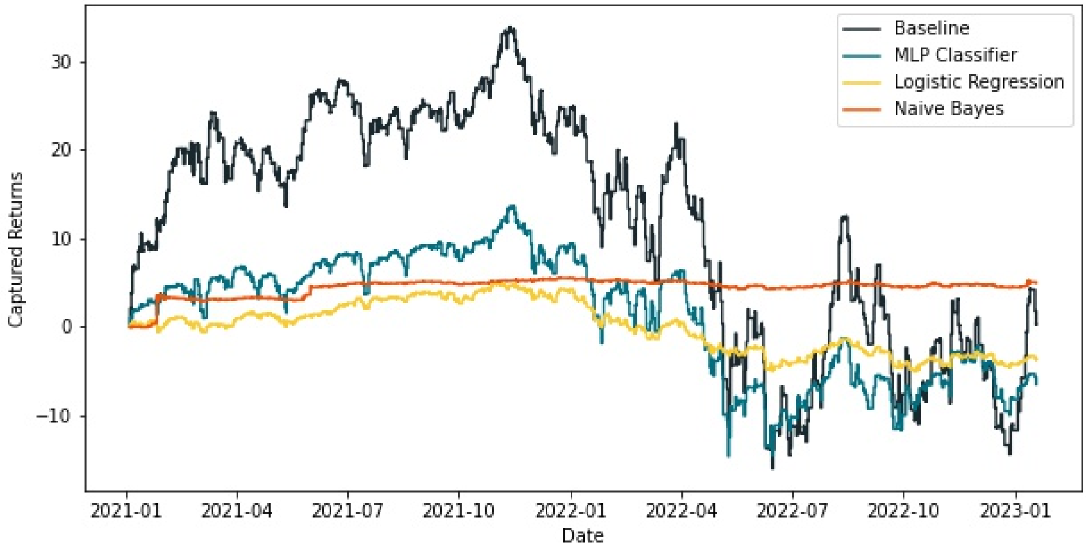

Figure 5: The performance on the test set of the classification models that do not have a tree-based architecture.

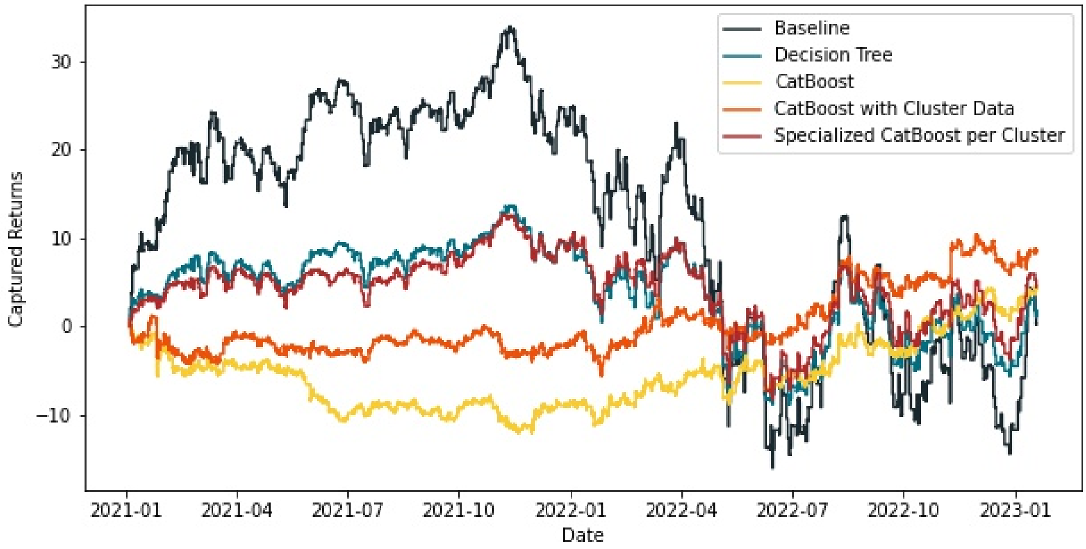

Figure 6: The performance on the test set of the classification models that have a tree-based architecture.

5.2 Explainability

One other benefit of using architectures based on trees in price prediction is that a forecast is easier to analyze. For example, the SHAP package [14] contains tools for visualizing the decision process of a model. Each dot represents one value in the dataset. The color of the dot represents its value. Higher values have colors that are close to pink, whereas lower ones have colors that are close to blue. The horizontal axis represents the impact on the prediction of the features. All of the values on the right side of the vertical axis at 0 increase the prediction (the price of a stock is more likely to increase during the next day), whereas the ones on the left side of the axis decrease it. Therefore, this plot tells us that a model will predict an increase in stock when there is an increase in the following indicators:

- On-Balance Volume

- Accumulation/Distribution
- Exponential Moving Average

At the same time, high values of the fast Stochastic Oscillator (K) and the Relative Strength Index decrease the predictions, as they are predominantly on the left side of the vertical axis at 0. Explainability could provide a starting point for future feature engineering strategies, or allow investors to observe patterns that go against the common knowledge of the market.

All of the other features have high values on either side of the vertical line, which suggests that there might be some interaction effects that are not visible in the plot.

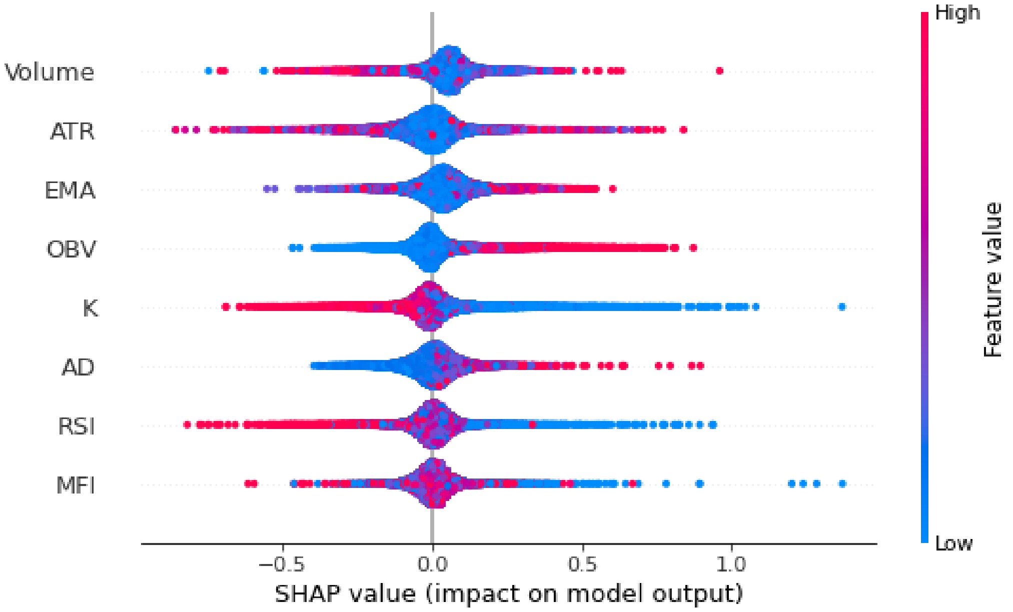

Figure 7: The SHAP summary plot of the CatBoost model.

6 Conclusion

We showed how popular technical indicators could be used to predict whether the price of an asset will increase or decrease. Cleaning the data, taking into consideration the most representative indicators, dividing the assets into clusters and using a decision tree based classifier, enabled us to come up with a strategy that produces profit and surpasses the market performance in 2022. Tree-based models seemed to perform particularly well when compared to other architectures. Moreover, we provided insights into how specific technical indicators affect the model prediction based on SHAP values, giving a good interpretability of the model.

References

[1] D. Ikizlerli, “The relation between trading volume and return volatility: Evidence from borsa istanbul,” *Business and Economics Research Journal*, 2022.

[2] R. K. Narang, *Inside the black box: A simple guide to quantitative and high-frequency trading.* John Wiley amp; Sons, Inc., 2013.

[3] Deriv, “The 4 most common types of technical indicators,” Jun 2022. [Online]. Available: https://eu.deriv.com/academy/blog/posts/the-4-most-common-types-of-technical-indicators

[4] A. Hayes, “On-balance volume (obv): Definition, formula, and uses as indicator,” Sep 2022. [Online]. Available: https://www.investopedia.com/terms/o/onbalancevolume.asp

[5] C. Mitchell, “Accumulation/distribution indicator (a/d): What it tells you,” Jan 2023. [Online]. Available: https://www.investopedia.com/terms/a/accumulationdistribution.asp

[6] H. Adam, “Average true range (atr) formula, what it means, and how to use it,” Dec 2022. [Online]. Available: https://www.investopedia.com/terms/a/atr.asp#citation-1

[7] B. Dolan, “Macd indicator explained, with formula, examples, and limitations,” Dec 2022. [Online]. Available: https://www.investopedia.com/terms/m/macd.asp

[8] W. Wheeler, “Create a stochastic oscillator in python,” Jan 2022. [Online]. Available: https://medium.com/wwblog/create-a-stochastic-oscillator-in-python-a7da42473677

[9] A. Hayes, “Stochastic oscillator: What it is, how it works, how to calculate,” Oct 2022. [Online]. Available: https://www.investopedia.com/terms/s/stochasticoscillator.asp

[10] M. Cory, “Money flow index - mfi definition and uses,” May 2022. [Online]. Available: https://www.investopedia.com/terms/m/mfi.asp

[11] F. Jason, “Relative strength index (rsi) indicator explained with formula,” July 2022. [Online]. Available: https://www.investopedia.com/terms/r/rsi.asp

[12] M. MacMahon and D. Garlaschelli, “Community detection for correlation matrices,” *arXiv preprint arXiv:1311.1924*, 2013.

[13] A. V. Dorogush, V. Ershov, and A. Gulin, “Catboost: gradient boosting with categorical features support,” 2018. [Online]. Available: https://arxiv.org/abs/1810.11363

[14] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf