Predicting the direction of stock market prices using tree-based classifiers

CHAGNON Pierre, MOHAMED Shamir, SARFATI Alban, TAYLOR Thomas, TRIGANO Elie

CentraleSupélec

Abstract. Machine learning techniques have recently become the norm for detecting patterns in financial markets. The problem addressed in this work involves predicting the direction of stock price changes, specifically whether they will increase or decrease compared to the price n days prior. To achieve this, we employ two algorithms, Random Forest and Gradient Boosted decision trees, which rely on decision tree ensembles. We evaluate our approach and report the testing performances for two technological stocks, demonstrating its superiority over existing prediction models.

Keywords: Stock price movement · Random Forest · XGB.

1 Introduction

The financial sector is well suited to the application of machine learning, as large amounts of data are available due to the revolution of electronic trading and more accessible computing power.

In a machine learning context, a basis trading task it to use models to analyze financial data and make prediction about future behavior of financial markets such as the direction of securities movements (upward or downward).

Financial data series have unique characteristics that make prediction tasks difficult, such as their non-linear and non-stationary nature. To try to reveal the entire spectrum of information, powerful AI systems have been engineered with a single goal of maximizing predictive performance.

The goal in this work is to design a framework using tree-based method to learn from the market data and predicts the direction in which a stock price will change at the closing time everyday. The models are train on time intervals of 1, 5, 10, 15, 21 days. These lagged days are intended to provide ML algorithms with information in two different time horizons: the recent and medium past. The aim is to generate models that generalise the forecasts for a short to medium term execution window.

2 Data

We construct the dataset by using daily historical data from yahoo finance, the data has been requested for technological stocks - Apple and META - over the last twenty years. Each instance is indexed by a timestamp and characterized by the open price, close price, highest price, lowest price and volume. The close price has been exponentially smoothed ($\alpha = 0.095$) in order to remove random variation or noise from the historical data, allowing the model to easily identify the long-term price trend in the stock price behavior. Based on these data, for each stock, we create the features and the labels that will allow to form the training and testing dataset.

2.1 Features

The features we implemented are technical indicators, they represent statistical tools and are extensively use to make investment decisions by generating signals. Seven of them were developed to identify trends, regime switches, momentum and potential reversal points in the stock market. They can be can be divided into four categories:

Trend followers identify the recent main movements of stock prices in recent past, using indicators like Exponential Moving Average (EMA) with a period of 7, 14 and 21 days.

Divergence identifiers aim to detect potential regime switches, where the current trend ends and the prices start moving in the opposite direction, using indicators like the Moving Average Convergence Divergence (MACD).

Momentum indicators measure the strength and direction of a stock's price movement, such as the Relative Strength Index (RSI).

Oscillators track the price variation of stocks to identify possible reversal points using indicators like Williams %R and STOCH.

2.2 Label

The label is a binary variable given by the price direction. It is therefore used to predict upward or downward movements in the market.

2.3 Transformation

The features are scaled within the range $(0, 1)$ by applying a MinMaxScaler fit on the training set. The rescaling is a necessary step as it un-biased learning and ensure that each feature contributes equally to the decision-making process. The transformation is learnt on the training set only in order to ensure that no data-leakage happens.

2.4 Environment

The training and testing set are constructed based on 2.1, 2.2, 2.3 and 2.4 with a training split corresponding to the first 90% time series instances, i.e. the testing represents the last 10% of the dataset and goes from 2021 to 2023. The training part is coupled to a cross validation with rolling window to tune the hyper-parameters. The testing part evaluate the generalization error, by looking at the accuracy and the performance of the trading strategy.

3 Model

3.1 Random Forest

In the forecasting process, we have considered Random Forest which is a decision-tree-based model and therefore is intrinsically explainable. Moreover, by being composed of deep de-correlated decision trees, as a consequence of bagging and random feature selection, Random Forest can handle noisy and outliers instances, which can be particularly useful when working with stock price data. Optimizing the hyper-parameters of the RandomForestClassifier is an important step in achieving optimal accuracy and trading strategy's performance. For each stock and each trading window, the number of trees in the forest, the maximum number of samples and the maximum number of features considered at each split have been tuned.

3.2 XGBoost

We also employed XGBoost, a tree-based gradient boosting model that has proven to be highly effective in many applications, including stock market prediction. XGBoost works by sequentially adding decision trees to the model, with each subsequent tree seeking to minimize the residual error of the previous tree. This approach allows XGBoost to create complex models that can capture non-linear relationships between features and target variables. Additionally, XGBoost provides several hyper-parameters that have been optimized to achieve optimal performance, such as the learning rate, maximum tree depth and number of trees in the ensemble.

3.3 Tuning

The hyper-parameters have been tuned using blocked cross-validation with k=5. The data is split into several non-overlapping blocks, with each block containing a contiguous sequence of time points, the first block is used as the training set and the second block is used as the validation set to select the hyper-parameters. The process is then repeated for the subsequent blocks, with each block serving as the testing set once. This approach provides an accurate estimate of a model's performance on time series data as it respects the temporal structure of the data.

We use F1 score as the metric to maximize. Indeed, both false positives and false negatives can have significant financial consequences and the F1 score balances the trade-off between precision and recall to provide a better overall measure of the model's performance.

3.4 Metrics

In the experiments we provide the performance of these models according to different metrics: accuracy, F1-score, AUC, brier-score. However, the good performance of these metrics is not necessarily correlated to a good financial performance. Indeed, when a large number of transactions with a modest impact is correctly matched, while transactions with a very high impact is missed, the outcome would be a good statistical performance but a potentially very bad financial performance. Hence, it is important to correctly predict large market movements and not just the majority of market movements.

Trading Strategy The direction's predictions of the model are transformed into positions, i.e. if at time t the model predict 0 (upward) then $position_t = 1$ else $position_t = -1$. The positions are backtested in the following steps:

*At each time step t, the return of the strategy is $position_{t-1} * stockreturn_t$.

*Deduction of transaction costs when a trade has taken place.

*Calculation of the net asset value of the strategy and the market.

*Calculation of the absolute and the out performance of the strategy.

The performance (absolute or out) provide a complete picture of the model's financial performance.

4 Results

The ability to predict the direction of a stock's price change is influenced by its variability over time. Stocks with relatively stable prices are easier to predict than those with more volatility. In machine learning classifiers, a lower variance in the data is indicative of better predictability. This stability is also important from an economic perspective, as it contributes to the overall stability of the market.

For demonstrating the efficacy of our approach, we present the testing results of Apple and META. It is important to note that META went public on 18 May 2012, which significantly reduced the number of instances in the original data set compared to Apple.

4.1 Random Forest

Company Name	Trading Window	Accuracy	F-Score	Brier Score	AUC	Out-Perf
AAPL	1	0.5	0.48	0.26	0.51	0.29
	5	0.55	0.55	0.25	0.6	0.03
	10	0.49	0.47	0.3	0.4	-0.15
	15	0.49	0.48	0.25	0.53	-0.38
	21	0.51	0.51	0.27	0.52	-0.57
META	1	0.5	0.49	0.27	0.51	-0.17
	5	0.41	0.33	0.33	0.47	-0.21
	10	0.43	0.34	0.29	0.54	-0.26
	15	0.52	0.52	0.3	0.51	0.31
	21	0.61	0.62	0.22	0.71	1.43

Table 1. Results of classification using Random Forest.

Overall, the accuracy range from 40% to 60% depending on the stock and the trading window. One can note that the F-score doesn't increases with the increase in window-width, the trend is non-linear for both stocks. Moreover, the out-performance of the Apple RF model is positive in short-term window (1 and 5 days) while META RF model is positive for medium-term window (15 and 21 days). Overall, we can't conclude to any generalized trend.

4.2 XGB

Company Name	Trading Window	Accuracy	F-Score	Brier Score	AUC	Out-Perf
AAPL	1	0.49	0.39	0.29	0.49	-0.04
	5	0.52	0.51	0.25	0.6	-0.3
	10	0.6	0.56	0.25	0.53	0.53
	15	0.51	0.51	0.28	0.53	-0.4
	21	0.51	0.44	0.33	0.65	-0.54
META	1	0.5	0.48	0.28	0.51	0.63
	5	0.38	0.36	0.33	0.42	-0.37
	10	0.54	0.54	0.25	0.62	0.66
	15	0.4	0.31	0.37	0.46	-0.27
	21	0.5	0.46	0.37	0.49	0.58

Table 2. Results of classification using XGBoost.

We can observe a high variance in F1-score which is still non-linear. We expected higher F1-score as XGBoost often performs better than Random Forest, especially on complex datasets, as it can learn complex non-linear relationships between features and target variables.

4.3 Comparison of performance

The goodness of classification observed for a certain window-width in the case of XGBoost is not comparable to the goodness of classification for the same window-width in the case of Random Forest. Indeed, both model performances varies depending on the stock and trading window. For instance, Random Forest has higher F1-score on 21 days window while XGBoost has higher F1-score on 10 days window. Hence, for high outperformance on a 10 days window XGBoost are to be preferred while the Random Forest' outperformance on META 21 days window is substantial. To improve the performance of these models, one could increase the amount and quality of data, selecting more relevant features by looking for the most important ones, tuning with more labor the hyperparameters or addressing non-stationarity.

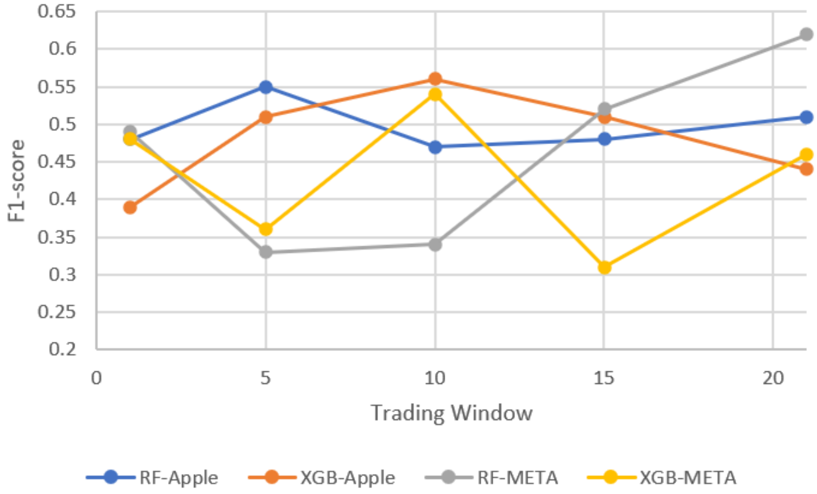

Table 3. Trend of F1-score against the trading width.

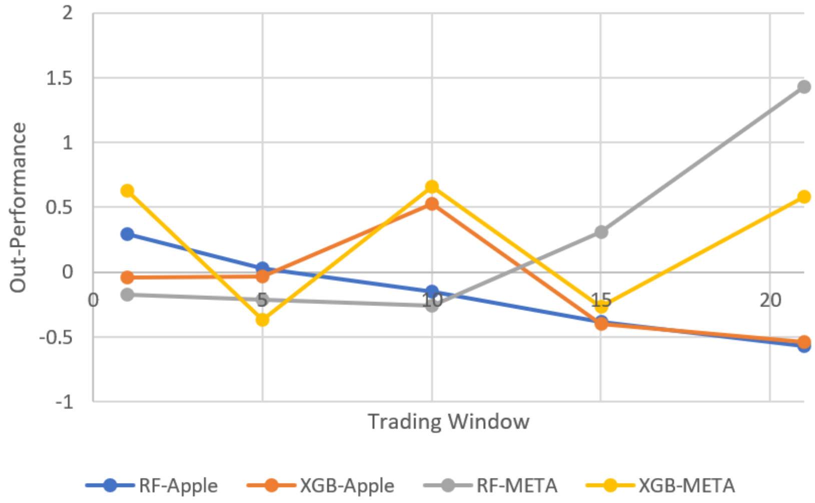

Table 4. Trend of out-performance against the trading width.

4.4 Trading indications

The model suggests trading decisions based on n-day intervals, with red dots indicating a predicted drop in prices and blue dots indicating a predicted rise. This information can guide buying or selling decisions. The graph suggests that buying is recommended when the model predicts a price increase after n days, while selling is recommended when the model predicts a price decrease. While it's interesting to explore whether the payoff function exhibits a trend that could inform a trading model in time series, it's important to note that this tool is not an all-in-one trading strategy. Rather, it's one component of a larger toolkit that can assist with investment decisions.

Moreover, we can observe the out-performance of the model strategy compared to the market, the framework allowed to improve the financial performances by providing significant long-short predictive signals.

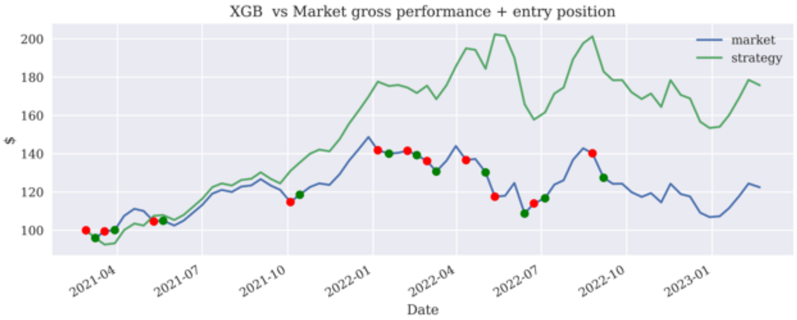

Table 5. Trading suggestions made by the model on AAPL for 10 days trading window.

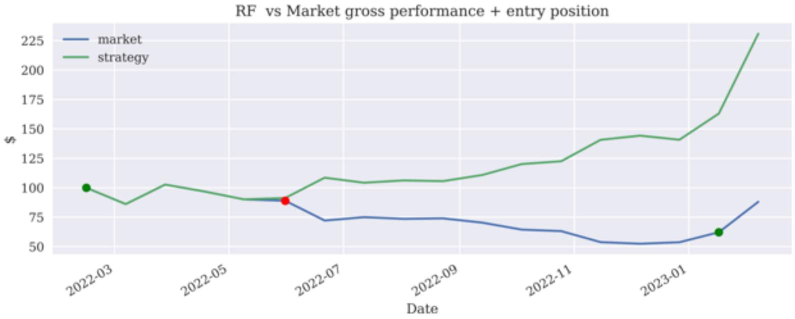

Table 6. Trading suggestions made by the model on META for 21 days trading window.

5 Conclusion

This paper used Random Forest and XGBoost classifiers to build predictive models for the direction of stock movements, which can produced impressive results. The models were evaluated using various parameters and have achieved high performance on some configurations. An innovative element of this work was the selection and application of technical indicators as features, providing flexibility and interpretation for financial analysis. In addition, hyper-parameters tuning with the introduced blocked cross-validation reduces the risk of overfitting and provides more accurate estimates of model performance on unseen data. The proposed models can be used for developing new trading strategies or managing stock portfolios by changing stocks based on trend predictions.