Algorithmic Momentum Trading Strategy

Kartikay Laddha Follow

Apr 10 · 8 min read

Infusing Big Data + Machine Learning & Technical Indicators for a Robust Algorithmic Momentum Trading Strategy

Introduction

Big data is completely revolutionizing how the stock markets across the world are functioning and how investors are making their investment decisions. Machine learning — the practice of using computer algorithms to find patterns in massive amounts of data — is enabling computers to make accurate predictions and human-like decisions when fed data, executing trades at rapid speeds and frequencies.

The business archetype monitors stock trends in real-time and incorporates the best possible prices, allowing analysts to make smart decisions and reducing manual errors that arise due to behavioural influences and biases. In conjunction with big data, algorithmic trading is thus resulting in highly optimized insights for traders to maximize their portfolio returns.

> **Better stock prices direction prediction is a key reference for better trading strategy and decision making by ordinary investors and financial experts** — *Jim Simons.*

Objective

Better stock prices direction prediction is a key reference for better trading strategy and decision making by ordinary investors and financial experts. Due to the increasingly large volume of data, manually analyzing data for some tasks like predicting the stock market movement has become impractical if not impossible for humans hence the need for automation. By providing large amounts of data, machine learning algorithms explore the data and search for a model that will achieve this goal.

The main objective is to implement an algorithmic momentum trading strategy using machine learning.

Methodology

We formulated the problem of stock trading decision as a classification problem with two different classes: buy and sell. We aim to identify the most efficient classifier based on some metrics. We used some momentum and volatility technical indicators with time periods of 7, 14 and 28 days as predictors. Since some of these indicators may be irrelevant for our data. We used the random forest variable importance technique to figure out the insignificant predictors. As a result, we obtained these relevant indicators: Relative Strength Index, Commodity Channel Index, Momentum(for time period=7), William's %R, Ultimate Oscillator, Rate of Change. These indicators are then standardized to be fed as input in different models.

As usual machine learning can be divided into two stages. The first stage is when the model is trained, and a second one, in which the system classifies the data accordingly to the technical indicators trained during stage one. The result of the analysis is the predicted trend of the market index, which can be used to set out some trading rules:

• *If the next day trend is **Uptrend**, then the decision is **BUY***

• *If **BUY decision** already **exists**, then **HOLD***

• *If the **next day trend** is **Downtrend**, then the decision is **SELL***

- *If **SELL decision** already **exists**, then **HOLD***

According to the result obtained with these rules, the return of strategy has been calculated.

Dataset

We pulled the daily historical data from Yahoo Finance. We chose 4 stocks (JINDALSTEL.NS, JSWSTEEL.NS, HINDALCO.NS and TATASTEEL.NS) in the Metallurgy sector — Nifty Metal Index (^CNXMETAL) of the National Stock Exchange Nifty India. The time period is from 01/01/2015 to 01/01/2020. The dataset is composed of 6 variables: date, the opening price of the day, the highest price of the day, the lowest price of the day, the closing price of the day, traded volume. We used 80% of this data as our training set and 20% as a test set.

Feature Construction

We begin by constructing a dataset that contains the predictors which will be used to make the predictions, and the output variable. Our dataset is built using raw data comprising of a 5-year price series for four different stocks. The individual stocks and index data consists of Date, Open, High, Low, Close and Volume. Using this data we calculated our indicators based on various technical indicators i.e. Exponential Moving Average, Stochastic Oscillator %K and %D, Relative Strength Index(RSI), Rate Of Change(ROC), Momentum (MOM).

Link to the Dataset Used:

kartikayladdha/BDT

Contribute to kartikayladdha/BDT development by creating an account on GitHub.

github.com

Technical Indicators

Various Technical indicators have been included in this strategy for better results and robust feature creation.

1. *Moving Averages*

Moving averages are widely used in technical analysis, a branch of investing that seeks to understand and profit from the price movement patterns of securities and indices. Generally, technical analysts will use moving averages to detect whether a change in momentum is occurring for a security, such as if there is a sudden downward move in a security's price. Other times, they will use moving averages to confirm their suspicions that a change might be underway. For example, if a company's share price rises above its 200-day moving average, that might be taken as a bullish signal.

2. Relative Strength Index

The relative strength index (RSI) is a momentum indicator used in technical analysis that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator (a line graph that moves between two extremes) and can have a reading from 0 to 100. Traditional interpretation and usage of the RSI are that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates an oversold or undervalued condition.

3. Stochastic Oscillator

A stochastic oscillator is a momentum indicator comparing a particular closing price of a security to a range of its prices over a certain period of time. The sensitivity of the oscillator to market movements is reducible by adjusting that time period or by taking a moving average of the result. It is used to generate overbought and oversold trading signals, utilizing a 0–100 bounded range of values.

The stochastic oscillator is range-bound, meaning it is always between 0 and 100. This makes it a useful indicator of overbought and oversold conditions. Traditionally, readings over 80 are considered in the overbought range, and readings under 20 are considered oversold.

Stochastic oscillator charting generally consists of two lines: one reflecting the actual value of the oscillator for each session, and one reflecting its three-day simple moving average. Because the price is thought to follow momentum, the intersection of these two lines is considered to be a signal that a reversal may be in the works, as it indicates a large shift in momentum from day today.

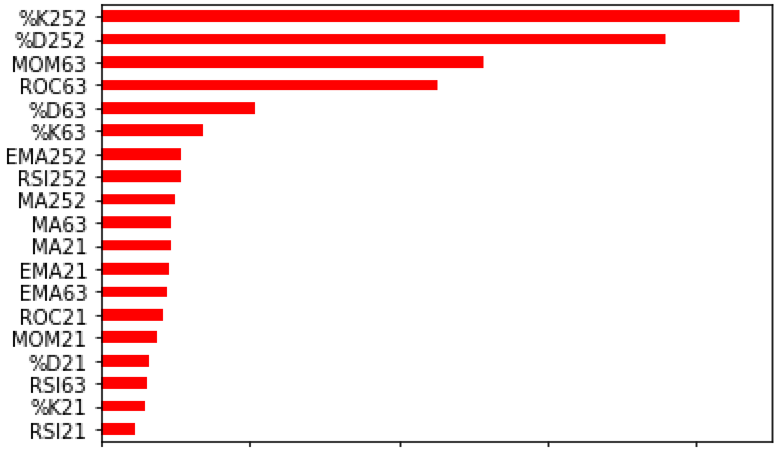
4. Momentum Price Strength

Momentum is the speed or velocity of price changes in stock, security, or tradable instrument. Momentum shows the rate of change in price movement over a period of time to help investors determine the strength of a trend. Stocks that tend to move with the strength of momentum are called momentum stocks. Momentum is used by investors to trade stocks in an uptrend by going long (or buying shares) and going short (or selling shares) in a downtrend. In other words, stock can be exhibit bullish momentum, meaning the price is rising, or bearish momentum where the price is steadily falling.

Feature Selection

Feature selection is the process of selecting a subset of features that are most relevant for model construction which aids in creating an accurate predictive model. There is a wide range of feature selection algorithms, and these mainly fall in one of the three categories: Filter method– selects features by assigning a score to them using some statistical measure. Wrapper method– evaluates a different subset of features, and determines the best subset. Embedded method — This method figures out which of the features give the best accuracy while the model is being trained.

In our model, we will use the filter method utilising the random.forest.importance function. The random.forest.importance function rates the importance of each feature in the classification of the outcome, i.e. class variable. The function returns a data frame containing the name of each attribute and the importance value based on the mean decrease in accuracy.



Machine Learning Algorithms

We used the following models and an ensemble of these models.

1. *K Nearest Neighbour*

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on the Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

2. Decision Tree

Decision tree — It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

3. Gaussian Naïve Bayes

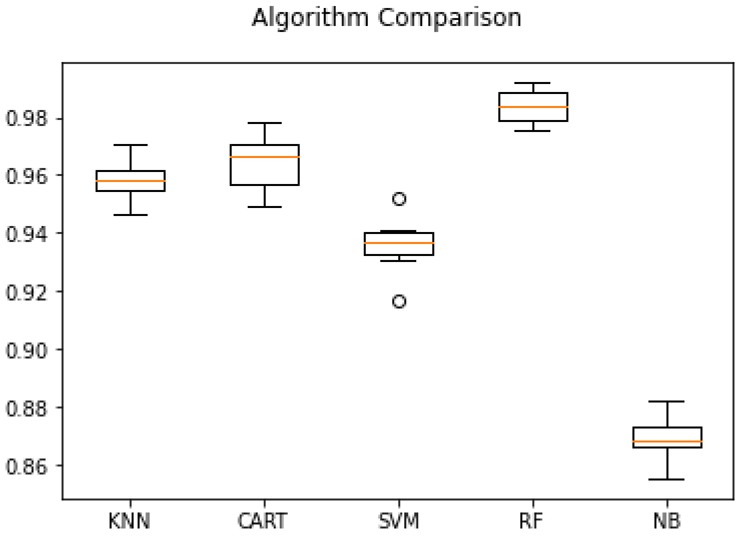
Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution.

4. Support Vector Machine

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

5. Random Forest

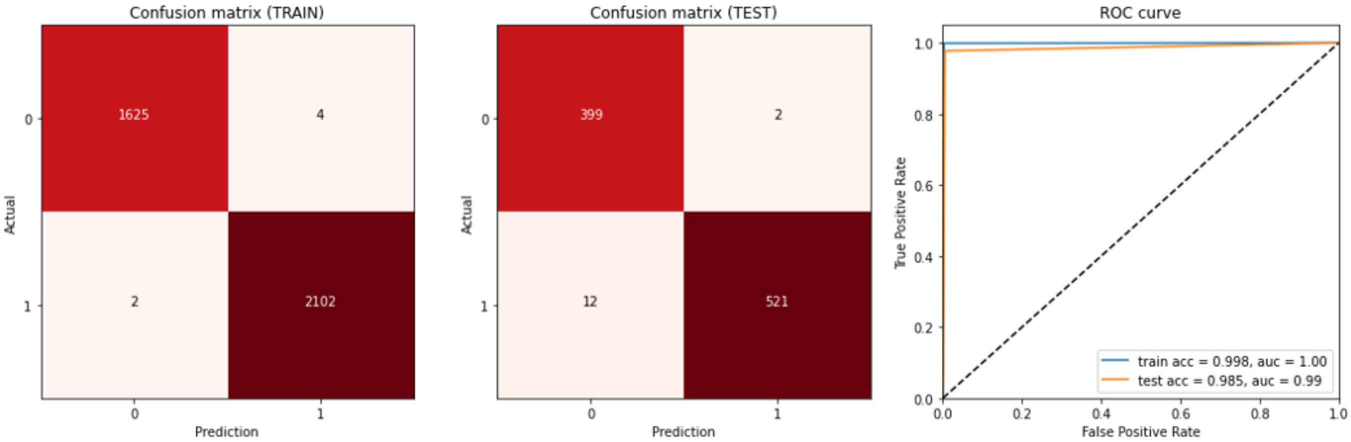
Random forest is a supervised learning algorithm. The “forest” it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. Put simply: random forest builds multiple decision trees and merges them to get a more accurate and stable prediction. Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction.



Model Accuracy

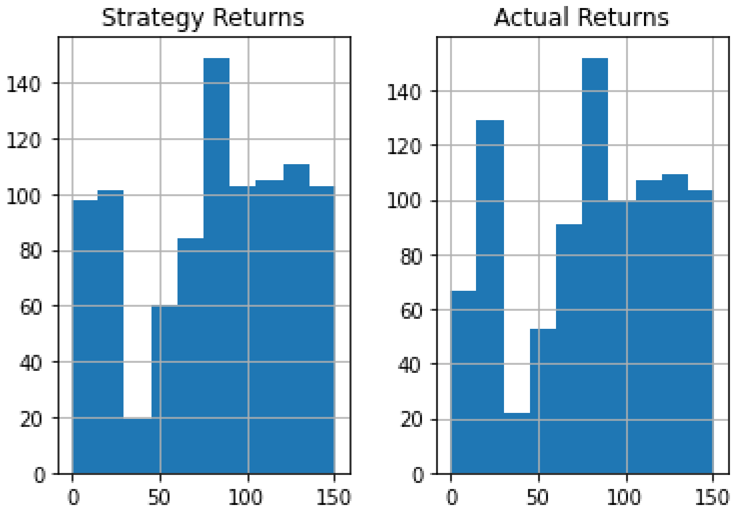
- K Nearest Neighbour — 95%
- Decision Tree — 96%
- Random Forest — 98%
- Support Vector Machine — 93%
- Gaussian Naïve Bayes — 86%

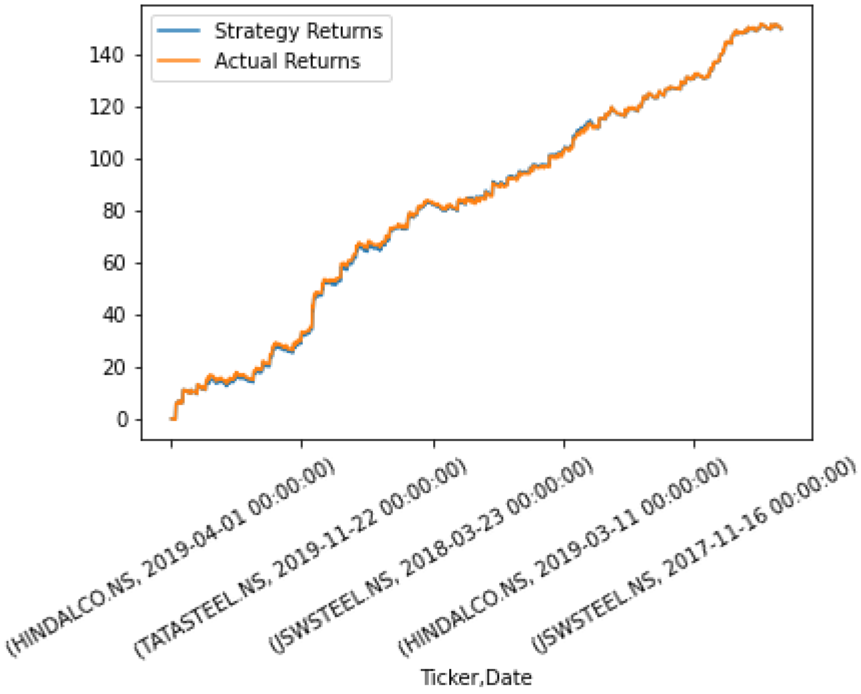
Prediction



Results

We see that returns are indeed positive and in line with actual market returns. Moreover, the strategy returns are slightly higher in some periods which is an added benefit.





Conclusion

Based on the return from our strategy, we do not deviate that much from the actual market return. Indeed, the achieved momentum trading strategy made us well predict the stock prices direction to invest/disinvest to make profits. However, as our accuracy is not 100% (but more than 98%) therefore, we made relatively few losses compared to the actual returns.

You can find more information about the same strategy on the link below:

kartikayladdha/BDT

Contribute to kartikayladdha/BDT development by creating an account on GitHub.

github.com