Algorithmic Momentum Trading Strategy

Big Data Technology ICA Project

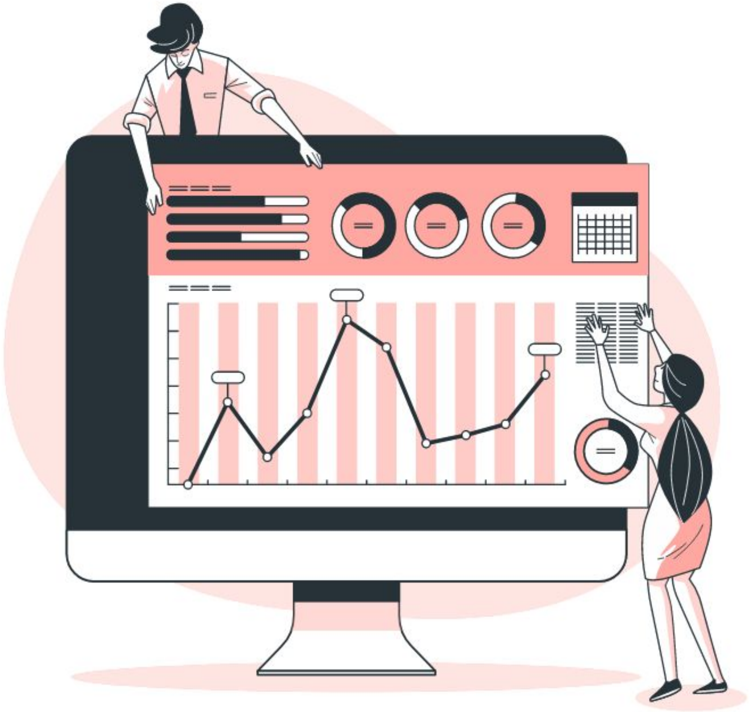
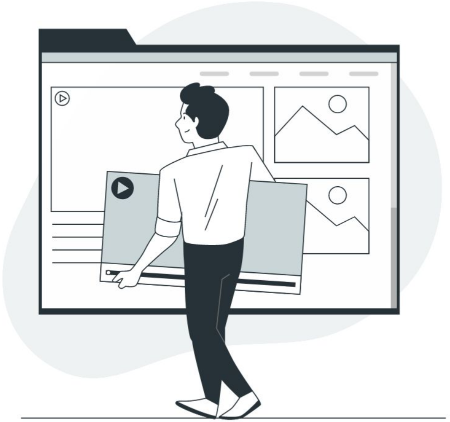


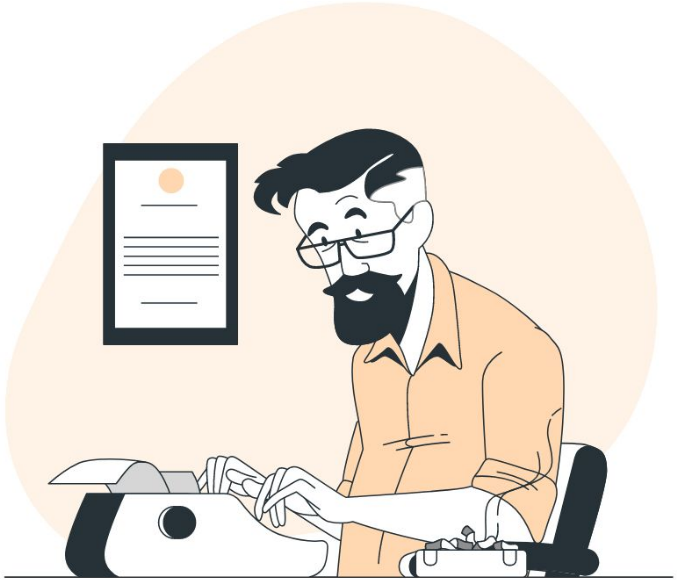
Table of contents

Introduction

Technology has become an asset in finance: financial institutions are now evolving to technology companies rather than only staying occupied with just the financial aspect: besides the fact that technology brings about innovation the speeds and can help to gain a competitive advantage, the rate



and frequency of financial transactions, together with the large data volumes, makes that financial institutions' attention for technology has increased over the years and that technology has indeed become the main enabler in finance.



"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 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 objective of our project is to implement a momentum strategy using machine learning.

Stocks & Trading

Stock trading is then the process of the cash that is paid for the stocks is converted into a share in the ownership of a company, which can be converted back to cash by selling, and this all hopefully with a profit.

Now, to achieve a profitable return, you either go long or short in markets: you either buy shares thinking that the stock price will go up to sell at a higher price in the future, or you sell your stock, expecting that you can buy it back at a lower price and realize a profit. When you follow a fixed plan to go long or short in markets, you have a trading strategy.

Methodology

We aim to formulate the problem of stock trading decision as a classification problem with two different classes: 'Buy' or 'Sell'. We are aiming to identify the most efficient classifier based on some metrics.

We will be including some momentum and volatility technical indicators as predictors and will also use random forest variable importance technique to figure out the insignificant predictors.

Then, machine learning will be divided into two stages. First stage when the model is trained, and a second one, in which the system classifies the data accordingly to the technical indicators trained during the stage one. The result of the analysis will predict the trend of the market index.

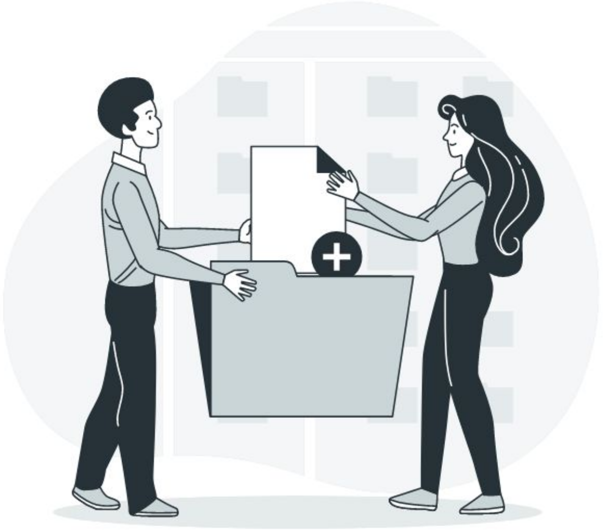
- If the next day trend is **Uptrend**, then the decision is **BUY**
- If BUY decision already exists, then HOLD
- If the next day trend is **Downtrend**, then the decision is **SELL**
- If SELL decision already exists, then HOLD

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, opening price of the day, highest price of the day, lowest price of the day, closing price of the day, traded volume. We used 80% of this data as our training set and 20% as test set.

Since we know that we are using time series data set for stocks so



Feature Construction

We begin by the constructing a dataset that contains the predictors which will be used to make the predictions, and the output variable.

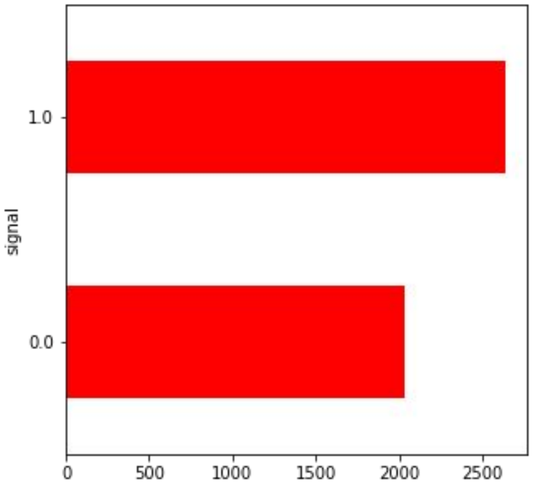
Our dataset is built using raw data comprising of a 5-year price series for four different stocks. The individual stocks and index data consists of Date, Open, High, Low, Close and Volume.

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).

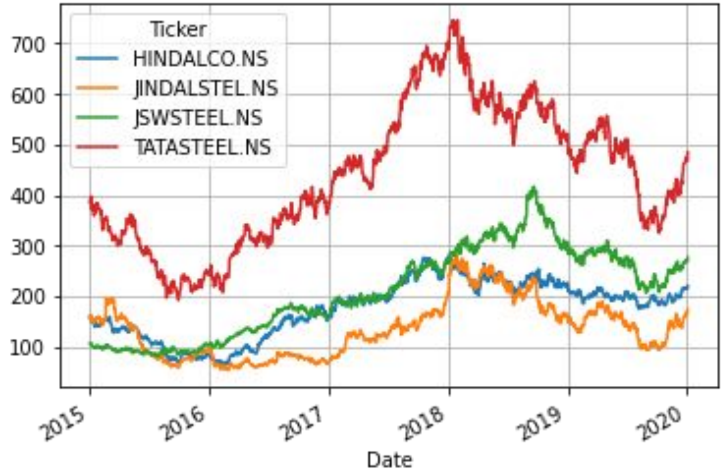
Initial Signal Generation

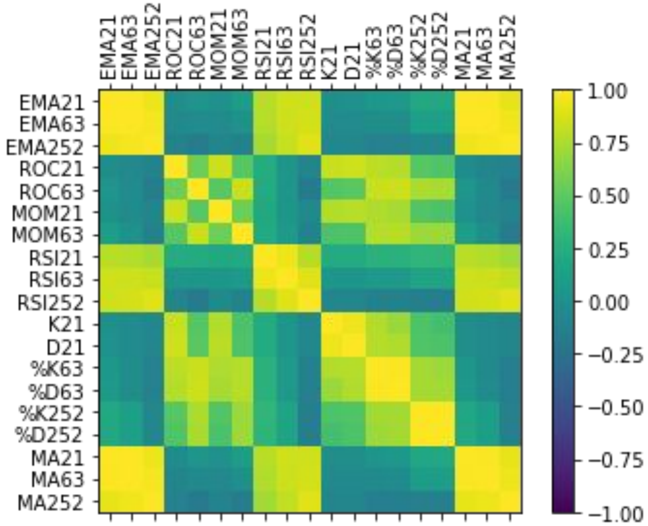
Initial Signals were generated using the "*Moving Average Convergence Divergence*" Strategy. This becomes our dependent variable now.

		High	Low	Open	Close	Volume	Adj Close	signal
Ticker	**Date**							
JINDALSTEL.NS	**2016-01-12**	85.449997	79.199997	83.699997	80.000000	10828818.0	80.000000	1.0
	2016-01-13	82.800003	74.599998	82.199997	77.800003	12945160.0	77.800003	1.0
	2016-01-14	76.750000	71.500000	76.699997	72.199997	11858071.0	72.199997	1.0
	2016-01-15	73.199997	62.000000	72.699997	64.000000	15515217.0	64.000000	1.0
	2016-01-18	64.900002	59.000000	62.349998	60.000000	15693048.0	60.000000	1.0



Exploratory Data Analysis

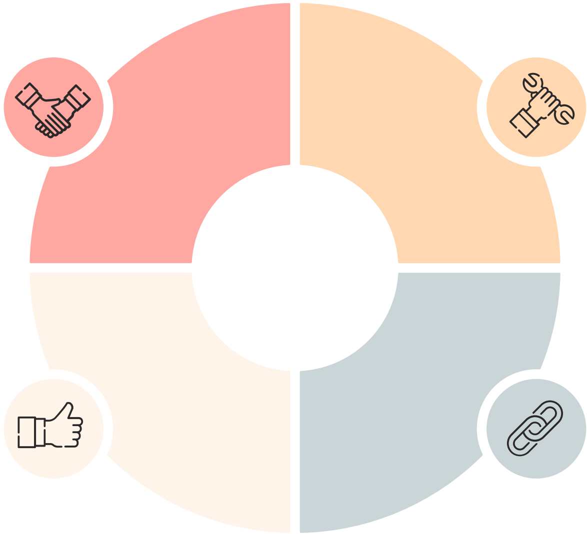




Technical Indicators

Moving Averages

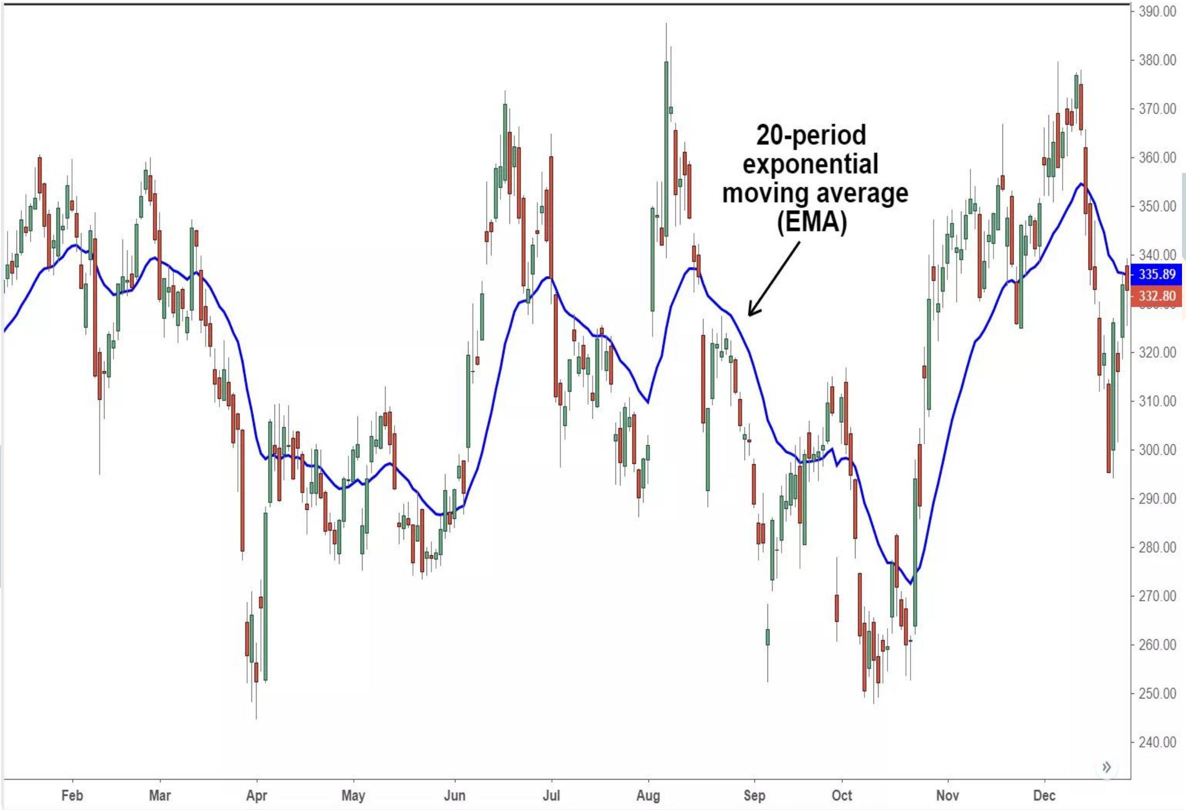
Relative Strength Index



Stochastic Oscillator

Momentum Price Strength

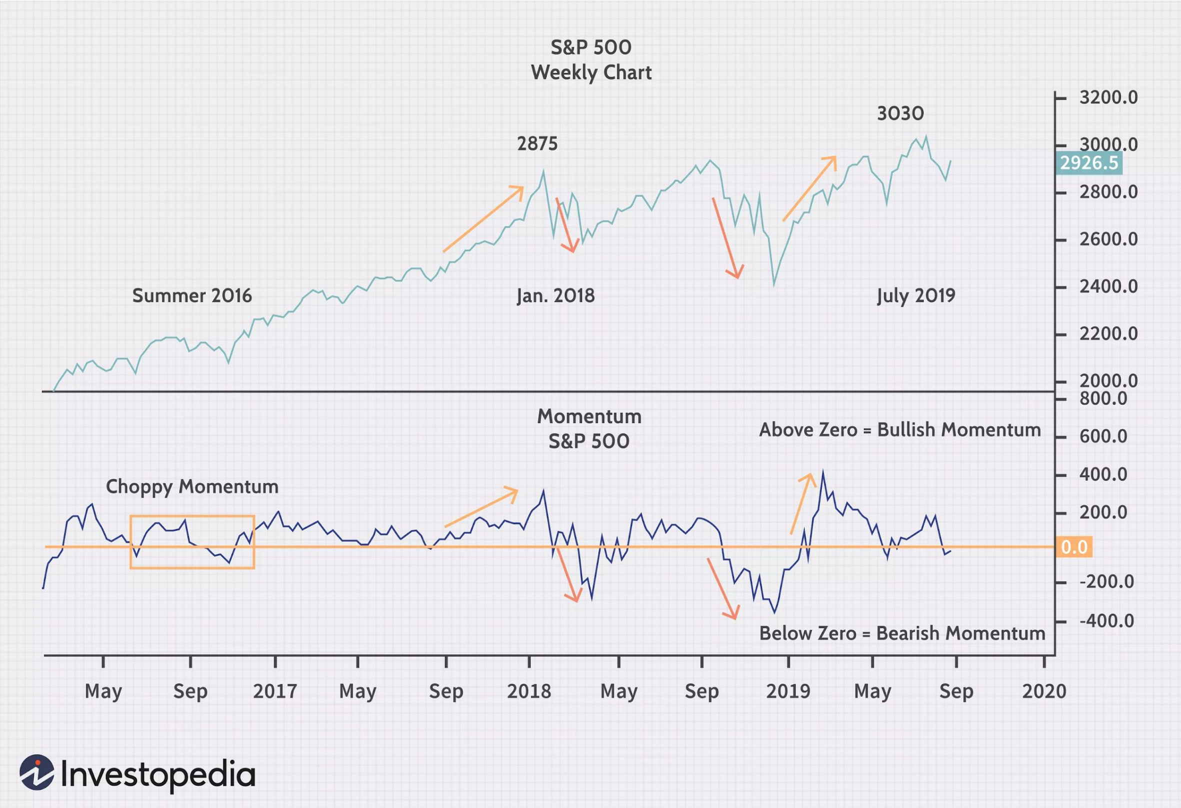
MOVING AVERAGES



EXPONENTIAL MOVING AVERAGE

RELATIVE STRENGTH INDEX

STOCHASTIC OSCILLATOR



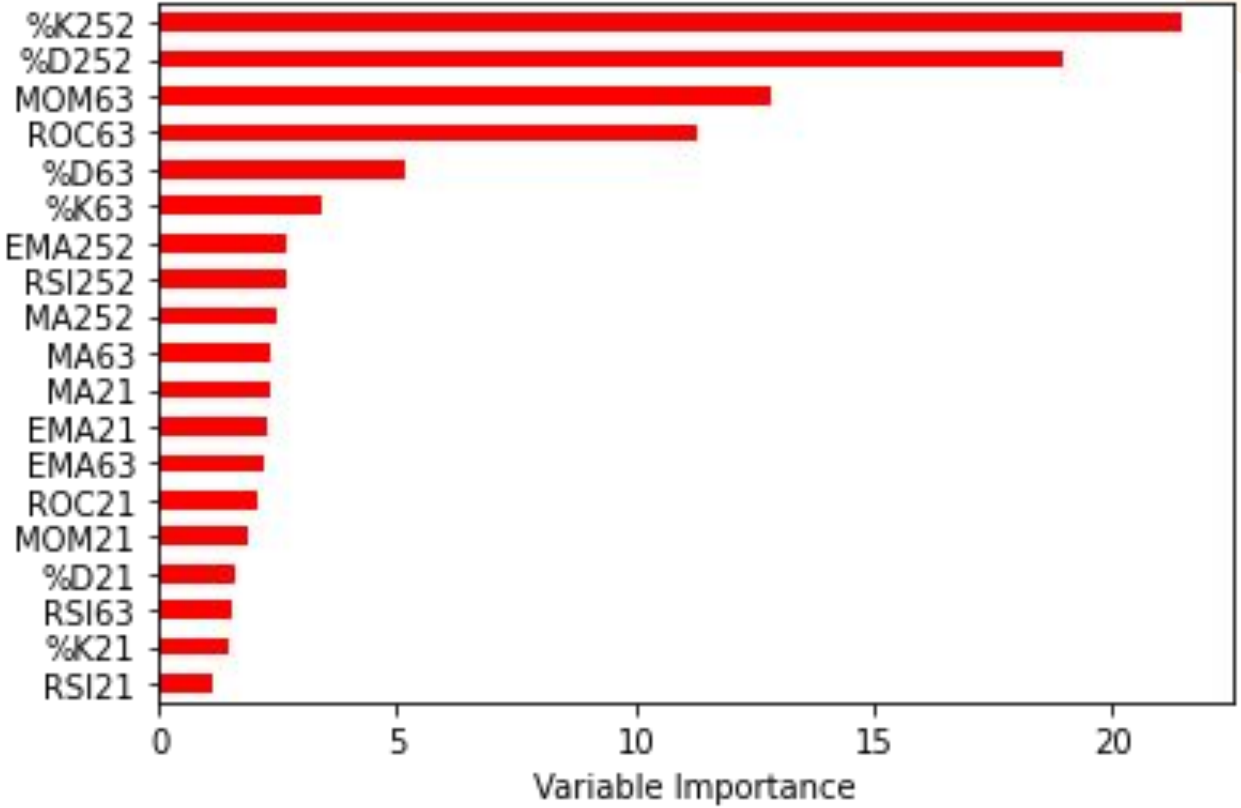
MOMENTUM PRICE STRENGTH

Feature Selection

Feature selection is the process of selecting a subset of features that are most relevant for model construction which aid in creating an accurate predictive model. There are 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 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 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.

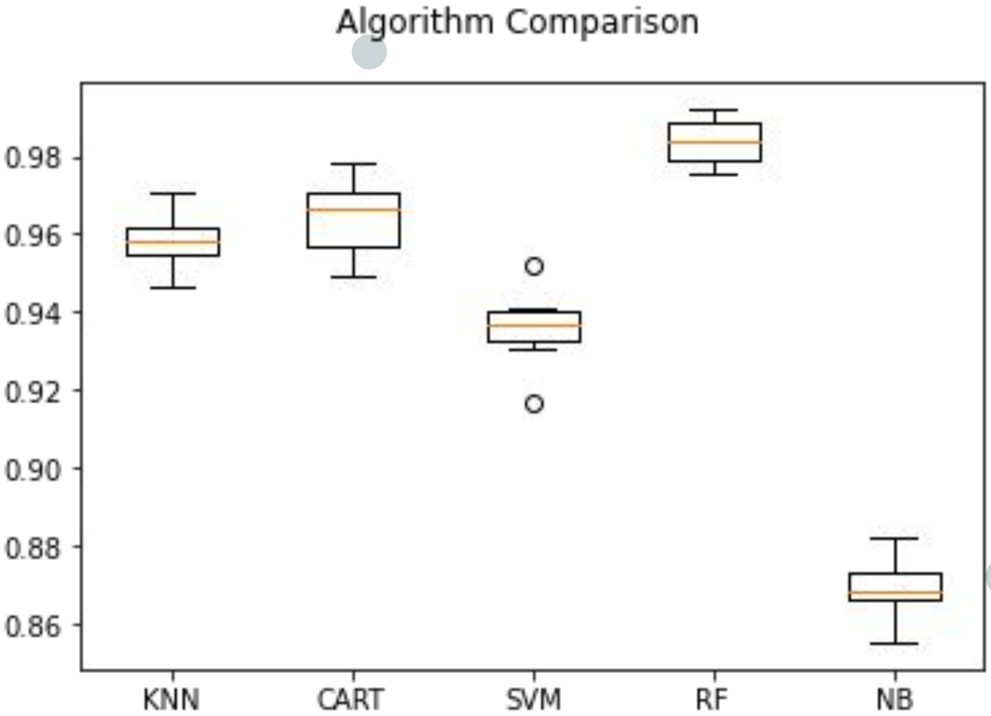

%K252
%D252
MOM63
ROC63
%D63
%K63
EMA252
RSI252
MA252
MA63
MA21
EMA21
EMA63
ROC21
MOM21
%D21
RSI63
%K21
RSI21
0
5
10
15
20
Variable Importance

Machine Learning Algorithms

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

- K Nearest Neighbour
- Decision Tree
- Random Forest
- Support Vector Machine
- Gaussian Naïve Bayes

Comparison of Algorithms



Accuracy

95%

K Nearest Neighbour

96%

Decision Tree

98%

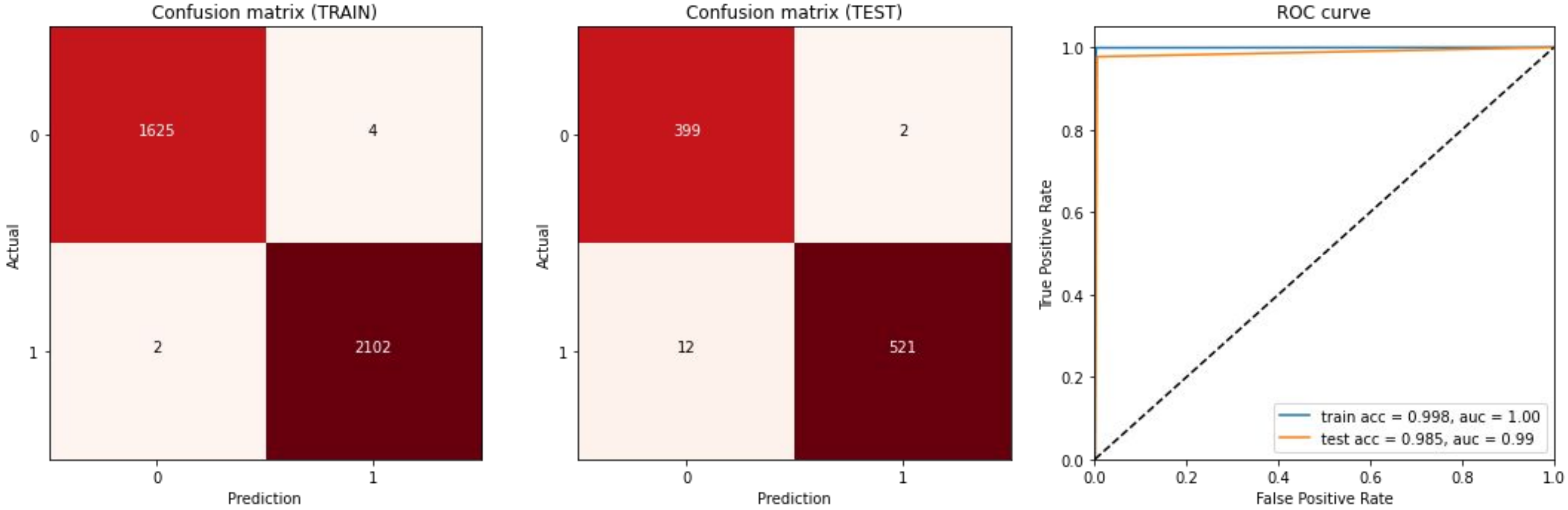
Random Forest

93%

SVM

86%

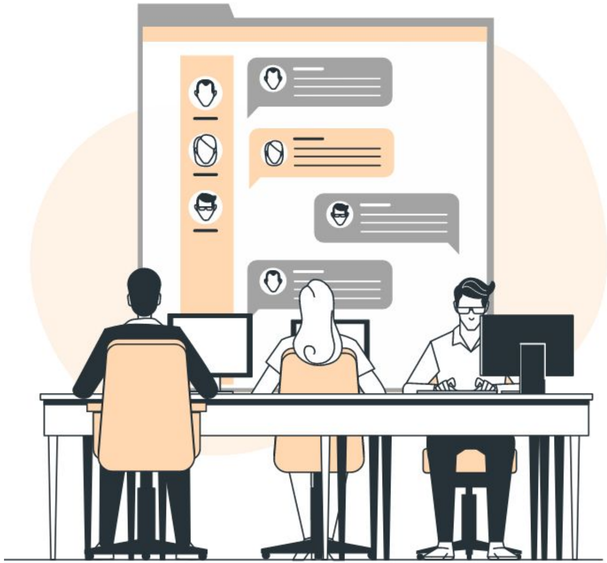
Gaussian Naïve Bayes

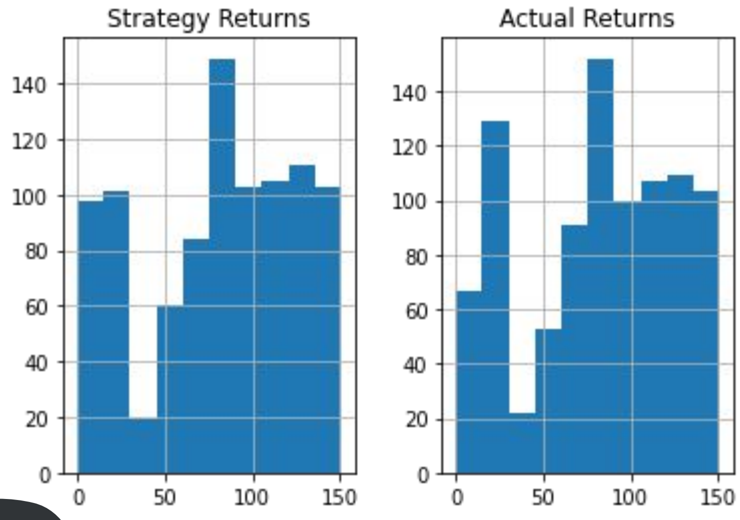


Confusion matrix (TRAIN)
1625
4
2
2102
Actual
Prediction
Confusion matrix (TEST)
399
2
12
521
Actual
Prediction
ROC curve
True Positive Rate
False Positive Rate
train acc = 0.998, auc = 1.00
test acc = 0.985, auc = 0.99

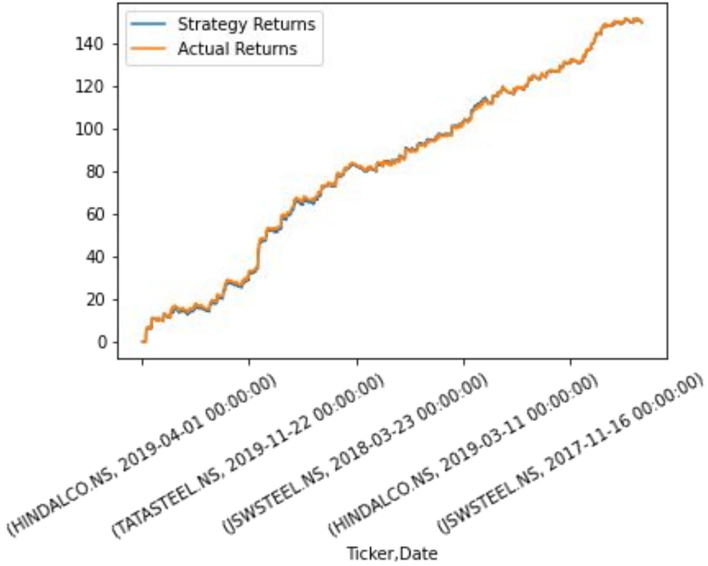
Results

We see that our 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.





Strategy Returns
Actual Returns
0
20
40
60
80
100
120
140
0
50
100
150



Strategy Returns
Actual Returns
0
20
40
60
80
100
120
140
(HINDALCO.NS, 2019-04-01 00:00:00)
(TATASTEEL.NS, 2019-11-22 00:00:00)
(JSWSTEEL.NS, 2018-03-23 00:00:00)
(HINDALCO.NS, 2019-03-11 00:00:00)
(JSWSTEEL.NS, 2017-11-16 00:00:00)
Ticker,Date

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 in order 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.

Our team

Ayush Hendre

J019

Vidhi Kapoor

J021

Kartikay Laddha

J025

Rohit Mittal

J031