Application of Machine Learning in High Frequency Trading of Stocks

Obi Bertrand Obi

Worldquant University
201 St. Charles Avenue, Suite 2500
New Orleans, LA 70170, USA
obibertrand@gmail.com

Abstract

Algorithmic trading strategies have traditionally been centered on follwing the market trends and the use of technical indicators. Over the years High Frequency algorithmic Trading has been left only in the hands of institutional players with deep pockets and lots of assets under management, despite huge returns involved. In this project webuilt trading strategies by applying Machine Learning models to technical indicators based on High Frequency Stock data. The result is an automated trading system which when applied to any stock could generate returns which are ten times higher than the market returns without significant increase in volatility. With advancement in technology High Frequency Algorithmic trading can be undertaken even by individuals or retail traders with moderate initial investment and technical skills.

Keywords:*Machine Lerning; Prediction of stock prices movements; Classification reports; Algorithmic trading; High frequency trading; Key performace indicators*

1. Introduction

Not too long ago, Algorithmic Trading was only available for institutional players with deep pockets and lots of assets under management. Recent developments in the areas of open source, open data, cloud computing and storage as well as online trading platforms have leveled the playing field for smaller institutions and individual traders, making it possible to venture in this fascinating discipline with only a modern notebook and an Internet connection. Nowadays, Python and its eco-system of powerful packages is the technology platform of choice for algorithmic trading. Among others, Python allows you to do efficient data analytics (with e.g. numpy, pandas), to apply machine learning to stock market prediction (with e.g. scikit-learn) or even make use of Google's deep learning technology (with tensorflow) and Microsoft's CNTK.

Algorithmic trading basically refers to the trading of financial instruments based on some formal algorithm. An algorithm is a set of operations (mathematical, technical) to be conducted in a certain sequence to achieve a certain goal. For example, there are mathematical algorithms to solve a Rubik's cube (The Mathematics of the Rubik's Cube or Algorithms for Solving Rubik's Cube). Such an algorithm can perfectly solve the problem at hand via a step-by-step procedure. Another example is algorithms for finding the root(s) of an equation (if it (they) exist(s) at all). In that sense, the objective of a mathematical algorithm is often well specified and an optimal solution is often expected

High-frequency trading(HFT)is a type of algorithmic trading characterized by complex computer algorithms that trade in and out of positions in fractions of seconds, leveraging arbitrage strategies in order to profit from the public markets. Commonly, traders take advantage of the penny spread between the bids-ask on equities. For the typical retail trader, this would seem redundant and the pay-off would be minuscule. For HFTs, the profit from the spread accumulates and as thousands of trades are executed, there are millions of dollars to be made [1].

Traditionally, financial markets operated on a quote-driven process where a few market makers provided the sole liquidity and prices for Financial Assets. Recently, major developments have beenmade to automate the Financial Markets which have led to many trading firms using computer algorithms to trade the Assets. High Frequency Trading (HFT), in particular, has been a major topic due to the features that distinguishes it from electronic and manual trading. This includes the extremely high speed of execution (microseconds), multiple executions per session, and very short holding periods (usually less than a day).

1.1. Problem statement

Time series data in financial markets are highly nonlinear, nonstationary and noisy in nature. Traditional models based on statistical methods, such as the Autoregressive Moving Average (ARMA) model, Autoregressive Integrated Moving Average (ARIMA) model, and General Autoregressive Conditional Heteroskedasticity (GARCH) model, suffer from limitations due to their linearity assumption. Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction such as; physical factors, psychological, rational and irrational behaviour, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy. Waren Buffet states that: “Forecasts may tell you a great deal about the forecaster; they tell you nothing about the future.” Hence finding the right algorithm to automatically and successfully predict and trade in financial markets is the Holy Grail in finance.

1.2. Project Objectives

The main objective of this project is to develop a High Frequency Trading System which uses Machine Learning to predict the movements of stock market prices with reasonable level of accuracyand to trade the stock with simple trading strategy to generate adequate performance. Other objectives include the following:

1. Comparative analysis of Machine learning Algorithims on High Frequency Stock data to determine algorithms with high predictive power for stock price movements
2. Perform technical analyses as features to the Machine Learning models in the High frequency Trading System
3. Generate and track adequate performance from the High frequency Trading System.
4. Add to the elaborate body of literature on application of Machine learning to Finance and High Frequency Trading

1.3. Hypothesis

Machine Learning Algorithms cannot predict stock price movement with reasonable amount of certainty in High Frequency Trading

2. Literature Review

Several authors have employed Machine learning technologies in predicting and trading stock markets. The following Algorithms have been used in various situations:

Because of their ability to model nonlinear relationships without pre-specification during the modeling process, neural networks (NNs) have become a popular method in financial time-series forecasting. NNs also offer huge flexibility in the type of architecture of the model, in terms of number of hidden nodes and layers. Indeed, Pekkaya and Hamzacebi compare the results from using a linear regression versus a NN model to forecast macro variables and show that the NN gives much better results [3].Many studies have used NNs and shown promising results in the financial markets. Grudnitski and Osburn implemented NNs to forecast S&P500 and Gold futures price directions and found they were able to correctly predict the direction of monthly price changes 75% and 61% respectively [4]. Another study showed that a NN-based model leads to higher arbitrage profits compared to cost of carry models [5]. Phua, Ming and Lin implement a NN using Singapore's stock market index and show a forecasting accuracy of 81% [6].

Another popular machine learning classification technique that does not require any domain knowledge or parameter setting is the decision tree. It also often offers a better visually interpretable model compared to NN, as the nodes in the tree can be easily understood. The simplest type of decision tree model is the classification and regression tree (CART). Sorensen et al. show that CART decision trees perform better than single-factor models based on the same variables in picking stock portfolios [7]. Another study found that a boosted alternating decision tree with expert weighing generated abnormal returns for the S&P500 index during the test period [8]. To improve accuracy, some studies used the random forest algorithm for classification, Booth et al. show that a regency-weighted ensemble of random forests produce superior results when analyzed on a large sample of stocks from the DAX in terms of both profitability and prediction accuracy compared with other ensemble techniques [9]. Similarly, a gradient boosted random forest model applied to Singapore's stock market was able to generate excess returns compared with a buy-and-hold strategy [10]. Some recent researches combine decision tree analysis with evolutionary algorithms to allow the model to adapt to changing market conditions. Hsu et al. present constraintbased evolutionary classification trees (CECT) and show strong predictability of a company's financial performance [11].

Support Vector Machines (SVM) is also often used in predicting market behaviors. Huang et al. compare SVM with other classification methods (random Walk, linear discriminant analysis, quadratic discriminant analysis and elman backpropagation neural networks) and finds that SVM performs the best in forecasting weekly movements of the Nikkei 225 index [12]. Nair et al. propose a system that is a genetic algorithm optimized decision tree support vector machine hybrid and validate its performance on the BSE-Sensex and found that its predictive accuracy is better than that of both a NN and Naive bayes based model [13]

While some studies have tried to compare various machine learning algorithms against each other, the results have been inconsistent. Patel et al. compares four prediction models, NN, SVM, random forest and naive-Bayes and find that over a ten year period of various indices, the random forest model performed the best [14]. However, Ou and Wang examine the performance of ten Machine learning classification techniques on the Hang Seng Index and found that the SVM outperformed the other models [15].

3. Methodology

3.1. Background to study area

This project is centered on stocks in the Dow Jones Industrial Average (DJIA). The Dow Jones Industrial Average [16], or simply the Dow, is a stock market index that indicates the value of 30 large, publicly owned companies based in the United States, and how they have traded in the stock market during various periods of time. The value of the Dow is not a weighted arithmetic mean and does not represent its component companies' market capitalization, but rather the sum of the price of one share of stock for each component company. The sum is corrected by a factor which changes whenever one of the component stocks has a stock split or stock dividend, so as to generate a consistent value for the index. As at the 31st of December 2018; the Market capitalisation of the Dow Jones Industrial Average is $6.56 trillion. The components are traded in the New York Stock Exchange (NYSE) and NASDAQ. The choice of this index is due to the availability of high-frequency financial data with high order-to-trade ratios. Alternative Indices that could be used are: S&P 500, NIFTY, HANSENG, CAC 40, etc.

3.2. Data collection

One of the 30 Stocks of the Dow Jones Industrial Average (DJIA) based on their historical Sharp Ratios is selected. High Frequency Historical (Minute by minute) Stock Datais downloaded from Yahoo Finance [2] using a Data Mining Functiondesigned in Python. Stock prices dataset downloaded include the following features: Date/Time, Open, High, Low, Close, Volume, and Adj. Close, for the last 2700 trading periods (Minute) consisting of 7 Trading Days.

3.3. Data analysis

Three stages of Data analysis are conducted: Feature engineering through Technical Analysis, Machine Learning and choice of high performant learning algorithm, forcasts of market trends and application of simple trading strategy.

3.3.1. Feature engineering: Several features are calculated and added to the features listed above (in data collection). These features will be computed using the following Technical Analysis on the stock data downloaded (Open, High, Low, Close, Volume, and Adj. Close). The features are as follows:

- **Trend Indicators:** Average directional index (A.D.X.), Commodity channel index (CCI), Detrended price oscillator (DPO), Know sure thing oscillator (KST), Ichimoku Kinkō Hyō, Moving average convergence/divergence (MACD), Mass index, Moving average (MA), Parabolic SAR (SAR), Smart money index (SMI), Trend line, Trix, Vortex indicator (VI)
- **Momentum Indicators:** Money flow index (MFI), Relative strength index (RSI), Stochastic oscillator, True strength index (TSI), Ultimate oscillator, Williams %R (%R)
- **Volume Indicators:** Accumulation/distribution line, Ease of movement (EMV), Force index (FI) , Negative volume index (NVI), On-balance volume (OBV), Put/call ratio (PCR), Volume–price trend (VPT)
- **Volatility Indicators :** Average true range (ATR), Bollinger Bands (BB), Donchian channel, Keltner channel, CBOE Market Volatility Index (VIX), Standard deviation (σ)

These indicators (features) are computed and included on the data set based on the degree of relaationship (coorelation) or the effects of these features with the movement in stock prices.

3.3.2. Machine learning models: The following Supervised learning classificationalgorithms (As discussed in quantinsti) [17] will be employed in the forecasting of stock markets

1. Decision Trees (CART)

2. Logistic regression(LR)
3. Naïve Bayes (NB)
4. Support Vector Machines (SVM)
5. K. neighbours(KNN)
6. Random Forest(RF)
7. Linear Decriminant Analyses(LDA)
8. Boosting with Extreme Gradient Boosting(XGBOOST)

The dataset represents; 27 features and one target(y). The target presents an increase in stock price (1) and a decrease in stock price (-1) per trading minute. This data is scaled using the standard scaler algorithm in scikit learn. The data is then partition into training set (80%) and a test set (20%) using the model_selection (train_test_split) algorithm in scikit learn. The data is then fed to the Machine learning algorithms for modelling.

3.3.3. Trading strategy (backtesting)

1. Buy and hold – the stock is purchased at the opening price on the first minute of the test period and then sold at the closing price of the last minute of the test period.

2. The model itself is evaluated as follows: if the model predicts the price will close higher, then the stock is bought at the open and sold at the close. If the model predicts the price will close lower, then the stock is sold at the open and bought at the close

3.4. Project implementation tools

The High Frequency Trading system is implemented in Python 2.7, Anaconda and Jupiter Notebook using the Following Liabraries:

- Numpy for Data analysis
- Pandas for Data Analysis
- Scipy for statistical analysis
- Scikit learn for implementation of Machine learning Algorithms
- Matplotlib and seaborn for graphical representation of results.

3.5. Presentation of results

- Heat maps for feature engineering, showing relationship of features and technical indicators
- Table showing performance matrix of different Machine Learning Algorithms
- Table showing the classification report of the Machine Learning Algorithm retained for the project
- Line graphs showing evolution in performance of the machine learning trading strategy against the market(Buy and Hold)
- Key performance indicators Matrix showing annualised performance ratios of the Machine Learning Trading system and the market
- SWOT (Streghth, Weakness, Opportunities and Threats) Analyses of the trading system will also be conducted

4. Results

Interesting results were obtained from the application of the Machine learning project as follws:

4.1. Heatmap of relationship of features or indicators used for modelling

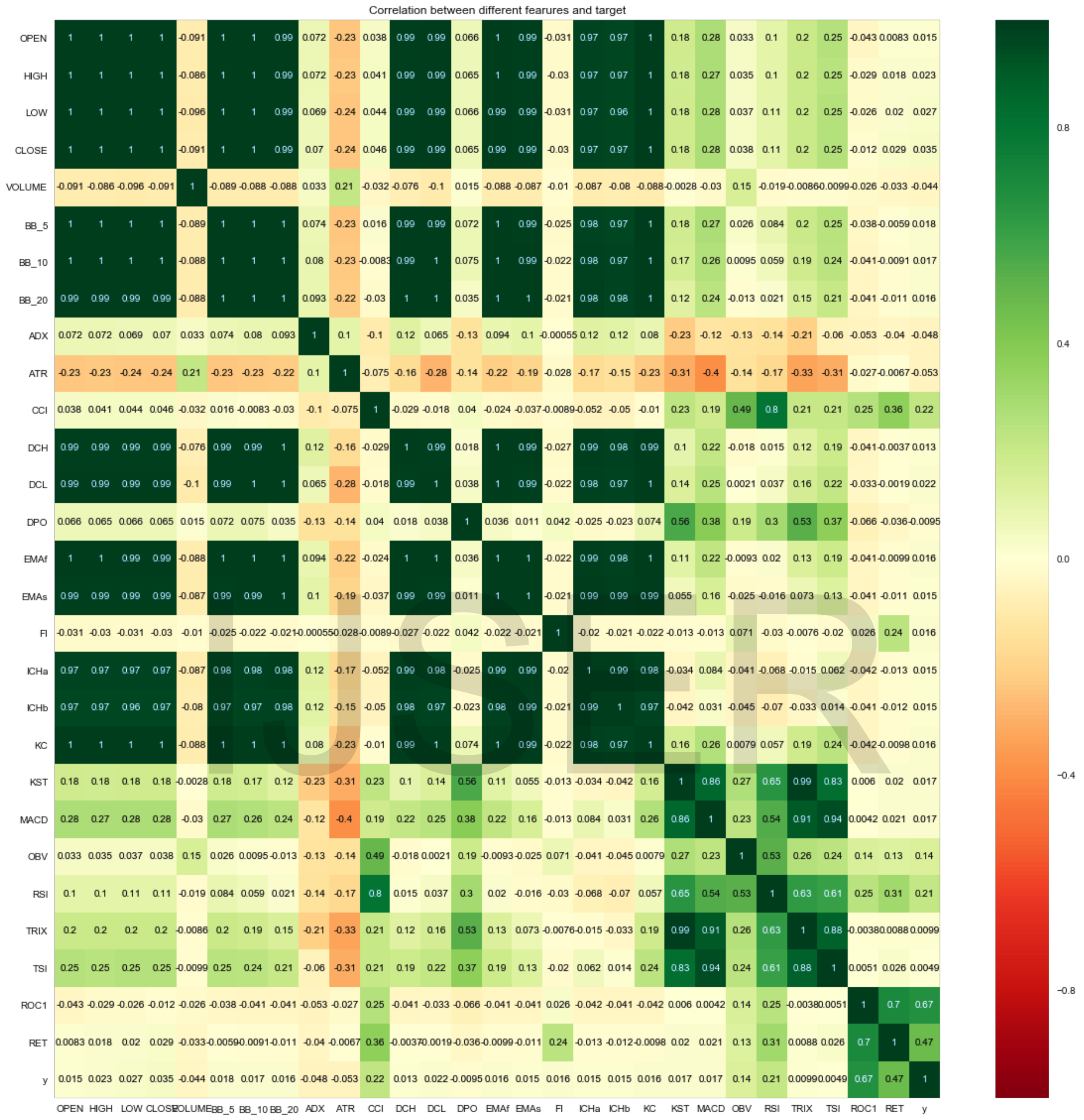

Figure1.6:HeatmapshowingtherelationshipofvariousFeaturesortechnicalindicatorsforthe model.

From the heatmap above it is clear that all the features contribute to the prediction of the target variable (y). The most outstanding features are: The rate of change (ROC), returns (RET), Relative Strength Index (RSI), Commodity Channel Index (CCI). The target variable (y); represents the increae (1) or the decrease (-1) in stock prices.

4.2. Performance of machine learning algorithms in prediction of stock price movements

Eight classification algorithms were used. The accuracy score of the different Machine Learning Models were computed. The results are shown on table 1. below:

Table 1.Acuracy score of machine learning models:

	Model	Description	Acuracy_Score
0	LR	Logistic Regression Classifier	0.994307
1	LDA	Linear Decriminant Analysis Classifier	0.888046
2	KNN	K Nearest Neighbours Clasifier	0.827324
3	CART	Decision Trees Classifier	1.000000
4	NB	Gausian naïve Bayes Classifier	0.810247
5	SVM	Support Vector Machines Classifier	0.954459
6	RF	Random Forest Classifier	1.000000
7	XGBoost	Extreme Gradient Boosting Classifier	1.000000

From the result above; all the models achieve considerable level of performance in predicting movements in stock prices. Outstanding models include: Decision Trees (CART), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) with an accuracy score of 100%.

4.2.1. Hypothesis testing: The Null hypothesis stipulates that Machine Learning Algorithms cannot predict movements in high frequency stock prices with reasonable level of accuracy:
The Hypothesis was tested in python by applying eight Machine learning models to forecast the direction of movement of high frequency (One minute) stock prices obtained from Yahoo Finance. Theminumum accuracy score of the models is 80% justifying the hypothesis that application of Machine Learning algorthms can predict stock prices movements in High Frequency Trading setting.

Decision Trees Classifier (CART) is thus retained to predict stock prices movements for the purpose of this project.

4.2.2. Detail performance of machine learning algorithm retained

Decision Trees Classifier was used to train 80 Percent of the data set consisting of 2105 data point with 27 features. The model was then use to classify or predict the target(y) consisting of 527 data points. The following classification performance report was produced in table 2.

Table2.Classificationreportofmachinelearningalgorithm

Class(y)	precision	Recall	f1-score	support
-1	1.00	1.00	1.00	245
1	1.00	1.00	1.00	282
avg / total	**1.00**	**1.00**	**1.00**	**527**

From the table, the model predicts two classes of data points "Y"; a predicted decrease in stock price is denoted by "-1" while a predicted increase is denoted by "1". The model predicted 245 decreases and 282 increases for the stock price of IBM tested. This is inline with the target variable (yTest) giving an accuracy score of 100% as denoted by the f1-score. This also justifies the hypothesis that Machine Learning Algorithms can predict the stock market with reasonable accuracy.

4.3. Performance of the trading strategy

The performance of the trading strategy is measured against the market returns (Buy and Hold strategy). The trading strategy is as follows:

- If the Model predicts an increase in price; we buy at the Open
- If the model predicts a decrease in stock price, we sell the stock.
- The asumption for this strategy is that short selling is allowwed, No transaction cost and there is equal investment.

The performance of this strategy and the market ('Buy and Hold') is presented in the following Line graphs and performance table as follows:

4.3.1. Performance evolution

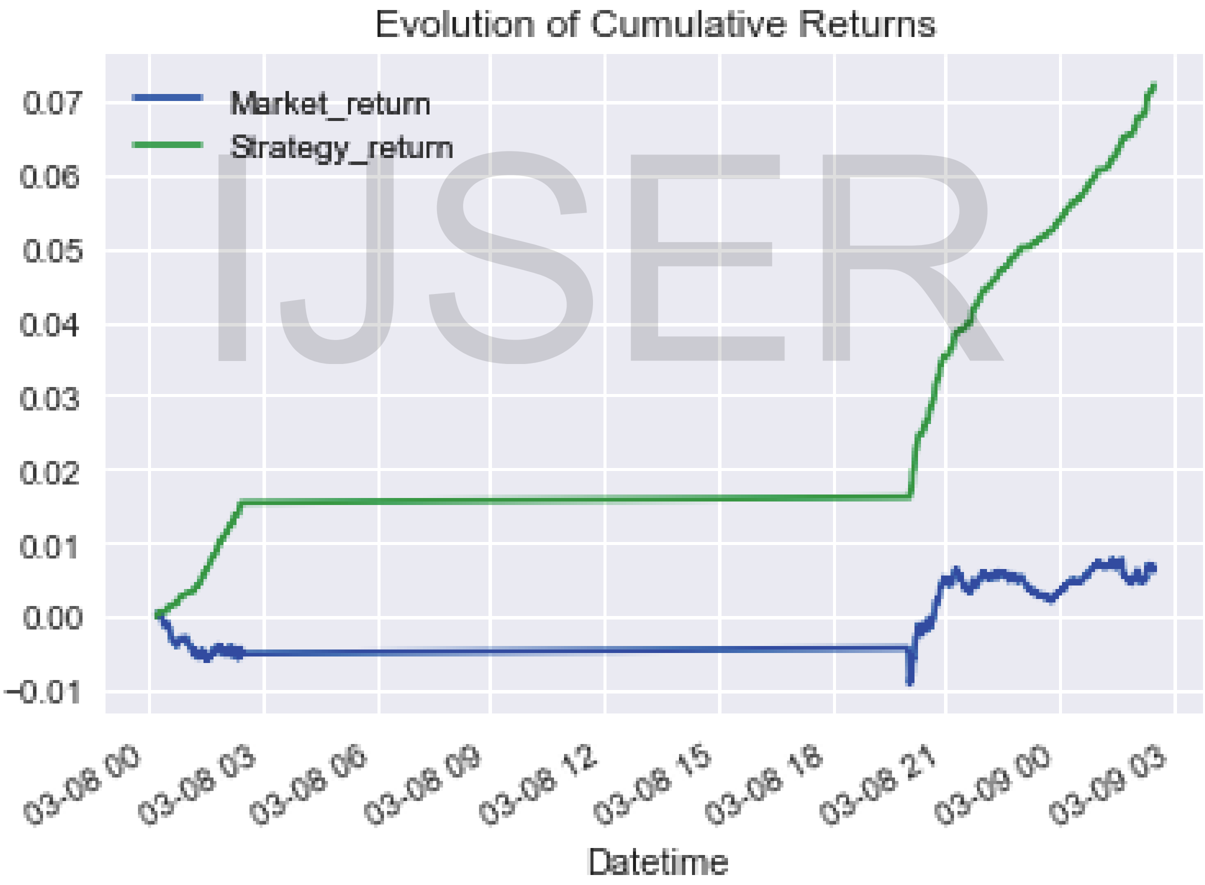

Figure 2: Evolutions of Cumulative Returns:

Evolution of returns shows that at the end of the trading period the cumulative strategy returns is 10 times higher than the market returns. This return is generated from 527 trading operations (282 Purchase orders and 245 sales orders of stocks). These transactions are all carried out within one day (The 8th of March 2019) representing the test set in the prediction model. This is highly different from low frequency trading system which could have conducted only one or two trades per stock in a day.

4.3.2: Key performance indicators of the strategy Vs Market

TableIII.PerformanceofstrategyVsMarket:

Indicator	Market_return	Strategy_return
Annualized Return	0.5121	9.5550
Annualized Std Dev	0.0696	0.3508
Avg Loss Return	-0.0038	NaN
vg Win Return	0.0049	0.0379
Gain to Pain Ratio	2.0517	NaN
Lake Ratio	0.1407	0.0000
Loss Rate	0.3245	0.0000
Max Drawdown	-0.4824	0.0000
Sharpe Ratio	7.3529	27.2359
Trade Expectancy	0.0045	NaN
Win Rate	0.6755	1.0000

From the table above; the performance indicators of the strategy are highly superior to those of the market:

1. **The annualised return** of the strategy is almost 10 times higher than that of the market. Hence an investor using Machine Learning system could perform significantly higher than the market
2. **The annualised standard deviation** shows the risk or volatility of the system. Here the volatility of the strategy is higher than the Market. This is common in HFT, due to the fact that several trade orders of thesame stock are carried out in minutes, coupled with orders received from other investors causes the market to be highly volatile. This justifies the fact that High fequency markets are highly volatile.
3. **The Average Loss Return for** the strategy is 0, compared to -0.0038 for the market.This is due to the fact that the strategy model could comfortably predict the stock movements in the period concerned
4. **The Average Win return** is also 0.0379, greater than 0.0049 for the market. This supports the increase in strategy returns morethan the market
5. **The Lake Ratio, Loss rate and the Maximum Draw Down** for the strategy is 0. This surports the positive evolution of cumulative returns as shown on Figure 2.
6. **The sharp Ratio** is the ratio of excess returns and volatitility (Annualised standard deviation). The strategy Sharp Ratio (27) is highly superior to that of the Market (7). This is due to the minimal risk (volatility) of the strategy of 0.3. This justifies the fact that the increased strategy returns is due to smart trading strategy (using Machine learning), and not an increase in volatility.
7. **The Win Rate** of the strategy is 100% as compared to 65% for the market. This with other indicators shows clearly that Machine Learning Algorithms could effectively predict stock price movements, trade and produce superior return than the market in High Frequency Environment.

4.4. Swot analysis of the system

The strengths, weaknesses, opportunities and threats of the sytem are analysed below:

4.4.1. Strengths

1. Ability to generate several trades(527) in a day using simple machine learning strategy on High Frequency data
2. The system can make use of trading opportunities immidietely as they present themselves in minutes.
3. Ability to generate superior returns about 10 times higher than the market
4. Simple trading strategy based on accurate prediction of market movements using simple Machine Learning Algorithms
5. High Win rate of 100% with a win return per trade of 3%
6. Increased anualised sharp ratio leading to high alpha generation and higher profits.
7. It ensures "best execution" of trades as it minimizes the human element in trading decision making.
8. Improves liquidity with lesser Drawdowns
9. The system also reduces transactions costs significantly due to limited human interferences
10. The system performs significantly well on all the stocks in Dow Jones Industrial Average index and even on stocks out of the index

4.4.2. Weaknesses

1. Increase in volatility (from 6% to 35%) due to large number of trades within a limited time frame.
2. The system is not very interactive to the user. opportunities exist to make the system fully functional and interactive
3. Require huge amount of time in designing the functions and optimising the algorithms.
4. Strict monitoring of the system to avoid system overruns and failures
5. Difficulties in applying the system to several(morethan one) stock at a time due to difficulties in obtaining free High frequency data
6. Market sentiment indicators and Government regulation were not included as part of the features set. These indicators can greatly influence the market returns
7. Transaction cost and other expenses are not factored into the system further development will include the modules

4.4.3. Opportunities

1. Availbility of performant computers, software and internet facilities which facilitates the implementation of High Frequency algorithmic trading
2. Availability of simple programming languages, application development tools and modules like python, pandas, scikit learn, statsmodels, CNTK, matplotlib, Technical analyses library, etc to facilitate the designing of this project
3. Availability of huge trading opportunities in minutes to take advantage of.
4. Availability of financial markets with regulatory mechanisms (Securities Exchange Commision in USA, etc) to curtail the effects and imperfections of high frequency algorithmic trading.

4.4.4. Threats

1. Unavailability of free quality High Frequency data for longer period of times. For this project we could only get one minute data for the last seven trading days.
2. Increase in volatility could lead to frequent stock market breakdowns and imperfections
3. It requires high testing, monitoring and regulation as error in the system could lead to high lost of capital
4. It requires huge investment for the system implementation and trading.

5. Return margins are very tiny. Low investment and volatility will lead to very low profits and low cash flow.
6. High cost of acquisition of data for the trading sytem on longer time frames.

4.5. Further Research:

Further research will be based on building a fully functional interactive Algorithm trading sytem in the following areas:

- Continue and build a fully functional and interactive trading system with multiple stcoks and portfolios at a time.
- Build a database of historical high frquency stock data for major stock exchanges in the world and populate it with Data for the last five years of trading (Data would be purchased).
- Add other technical indicators to the system and enable the system to trade based on technical indicators and machine learning at a time.
- Add other Machine Learning and Reinforcement Learning algorithms to trade combination of technical indicators and machine learning on huge High Frequency stock data
- Add Options, Commodities, Forex and Cryto data on the system for effective High frequency algorithmic trading
- Develop and add other backtest functions based on technical indicators

5.0 Conclusion

Enormous opportunities exist in High Frequency trading with the possibility of making exceptional returns within the shortest possible time. Technological advancements and application of Machine Learning algorithms facilitates the implementation of High Frequency Trading and development of performant trading strategies by individuals, retail and institutional traders. Machine Learning is thus recommended for modelling High frequency trading strategies.

Acknowledgements

My immense gratitude goes to Worldquant University for a challenging opportunity in developing my capacity in modern technology and applications in Finance. Special appreciation goes to the Faculty and Professors for their regular support through out the program.

Reference

This project and codes are found on my github page: https://github.com/bertrandobi/WQU-CAPSTONE-Project. Other references consulted are as follows:

[1]. High Frequency Trading:https://en.wikipedia.org/wiki/High-frequency_trading.
[2]. Yahoo Finance:https://query1.finance.yahoo.com/v8/finance/chart.
[3]. Mehmet Pekkaya and CoÅ§kun HamzaÃğebi. An application on forecasting exchange rate by using neural network. Congress of YAEM, 27, 2007.
[4]. Gary Grudnitski and Larry Osburn. Forecasting s&p and gold futures prices: An application of neural networks. Journal of Futures Markets, 13:631–643, 1993.
[5]. Shang-Wu Yu. Forecasting and arbitrage of the nikkei stock index futures: An applicationofbackpropagationnetworks. Asia-Pacific Financial Markets, 6:341– 354, 1999.
[6]. Ming D. Phua, P.K.H. and W. Lin. Neural network with genetically evolved algorithms for stocks prediction. Asia-Pacific Journal of Operational Research, 18:103–107, 2001.
[7]. Miller-Keith L. Sorensen, Eric H. and Chee K. Ooi. The decision tree approach to stock selection. Journal of Portfolio Management, 27:42, 2000.
[8]. GermanCreamerandYoavFreund. Automatedtradingwithboostingandexpert weighting. Quantitative Finance, 4:401–420, 2010.

[9]. Gerding Enrico Booth, Ash and Frank McGroarty. Automated trading with performance weighted random forests and seasonality. Expert Systems with Applications, 41:3651–3661, 2014.

[10]. Qing-Guo; Li Jin Qin, Qin; Wang and Shuzhi Sam Ge. Linear and nonlinear trading models with gradient boosted random forests and application to Singapore stock market. Journal of Intelligent Learning Systems and Applications, 5:1–10, 2013.

[11]. Yuan Lin Hsu, Chi-I; Hsu and Pei Lun Hsu. Financial performance prediction using constraint-based evolutionary classification tree (cect) approach. Advances in Natural Computation, 3612:812–821, 2005.

[12]. Yoshiteru Huang Wei, Nakamori and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 32:2513–2522, 2005.

[13]. Modhandas V.P Nair, Binoy B. and N.R. Sakthivel. A genetic algorithm optimized decision tree-svm based stock market trend prediction system. International journal on computer science and engineering, 2:2981–2988, 2010.

[14]. Shah Shail Thakkar Priyank Patel, Jigar and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications, 42:259–268, 2015.

[15]. PhichhaungouandHengshanWang. Predictionofstockmarketindexmovement by ten data mining techniques. Modern Applied Science, 3:28–42, 2009.

[16]. Dow Jones Industrial Average. https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average

[17]. Machine Learning Algorithms in Finance. https://www.quantinsti.com/blog/top-10-machine-learning-algorithms-beginners

[18]. Trading Strategy: Technical Analysis with Python TA-Lib: https://towardsdatascience.com/trading-strategy-technical-analysis-with-python-ta-lib-3ce9d6ce5614