

# **Multiple Model-based Binary Classification for Forex Trading**

**By**

LEUNG Cheuk Hei, Victor

**Advised By**

David Paul, ROSSITER

**COMP 4971C - Independent Work**

[ 2022-2023 Summer ]

**The Hong Kong University of Science and Technology**

[ Department of Computer Science and Engineering ]

**Date of submission**

[ August 12, 2023 ]

# Table of Contents

<b>Abstract</b>	<b>3</b>
<b>Objective</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
Asset Classes	4
Trading Terminologies	6
Types of Machine Learning	9
<b>Methodology</b>	<b>13</b>
Data Collection	14
Software Framework	14
Initial Account Setting	15
Ideation	15
Strategies	17
Analytical Metrics	21
<b>Performance Evaluation</b>	<b>30</b>
Visualization	37
Evaluation	42
Comparison	42
Abnormality	44
<b>Improvement</b>	<b>44</b>
Hardware	44
Data Source	44
Technical Analysis	44
<b>Conclusion</b>	<b>45</b>
<b>References</b>	<b>46</b>
<b>Appendix</b>	<b>48</b>
A - Swing Trading Summary	48
B - Day Trading Summary	50

## **1. Abstract**

This document aims to juxtapose the efficacies of forex trading between fixed deposits and Multi-model-based Binary Classification. The project employs a rigorous trading strategy that leverages most machine-learning models to analyze daily trends. Swing and day trading strategies are implemented upon recognizing classical trends. The classification process intends to employ the entire spectrum of regressors and classifiers from sci-kit-learn. Both strategies intend to buy at a lower price and sell at a higher price. The outcomes are scrutinized to assess the viability of a buy-and-hold strategy. The study employs comprehensive 10-year data (2000/01/01 - 2010/01/01) for training. Trading will be conducted for 10 years (2010/01/02 - 2022/01/01). However, the average training and trading frames are reduced by half after dropping the NA data frame. The resulting trading behaviors, [Buy], [Sell], [Retain], [No Action], and [Buy & Sell] are constructed from the classification results. Assets, models, and metrics are integrated into arrays to assess their autonomous influences. The parameters of models remain untouched without additional tuning. The assessment criterion encompasses both classification metrics (accuracy, etc.) and trading metrics (Sharpe ratio, etc.), providing a holistic evaluation of the effectiveness of the trading strategy.

## **2. Objective**

In the fast-paced and high-stakes world of trading, the ability to quickly and accurately process vast amounts of data is essential. Machine learning comprises unsupervised and supervised learning, data preprocessing, feature engineering, model selection, evaluation, and optimization techniques [1]. It has emerged as a critical tool for traders, providing advanced algorithms and models capable of analyzing data with incredible speed and precision. The integration of natural language processing (NLP) and chatbots powered by GPT-4 has transformed the way traders approach the market, enabling them to quickly parse and analyze news articles, social media feeds, and other sources of information to identify trends and patterns that can impact the market. For example, a chatbot powered by GPT-4 can be used to monitor social media feeds for mentions of a particular stock or company,

analyzing sentiment and identifying potential risks or opportunities. By leveraging the latest advances in machine learning and NLP, traders can stay ahead of the curve and make informed investment decisions that drive success.

## **3. Introduction**

### **3.1. Asset Classes**

#### **3.1.1. Stock**

It is commonly referred to as equities or shares, epitomize a form of ownership in a publicly traded company [2]. By purchasing a stock, an individual becomes a shareholder and can lay claim to a portion of the company's assets and earnings. Trading of stocks occurs on stock exchanges, where a convergence of buyers and sellers takes place. Market forces of supply and demand determine the price of a stock, and it is subject to fluctuation based on various factors such as the company's performance, prevailing economic conditions, and investor sentiment.

#### Top 10 Volume Stocks in 2010

1. Bank of America (BAC)
2. Citigroup (C)
3. Ford Motor Company (F)
4. General Electric (GE)
5. Cisco Systems (CSCO)
6. Pfizer Inc. (PFE)
7. Microsoft Corporation (MSFT)
8. Alcoa Inc. (AA)
9. AT&T Inc. (T)
10. Intel Corporation (INTC)

#### **3.1.2. Commodity**

It represents fundamental raw materials or primary products that are exchanged on commodity markets [3]. These products span a broad spectrum and include metals, such as gold and silver, energy sources, such as oil and natural gas, and agricultural products, such as wheat and corn. The pricing of commodities is established by the interplay between market forces of supply and demand, and they remain susceptible

to fluctuation based on a wide array of factors such as meteorological conditions, political events, and global economic trends.

#### Top 10 Volume Commodities in 2010:

1. Crude Oil (CL)
2. Gold (GC)
3. Natural Gas (NG)
4. Corn (C)
5. Heating Oil (HO)
6. Brent Oil (CO)
7. Silver (SI)
8. Soybeans (S)
9. Copper (HG)
10. Wheat (W)

#### **3.1.3. Forex**

It is commonly referred to as foreign exchange or currency trading necessitates the astute buying and selling of diverse currencies to capitalize on profit derived from fluctuations in their exchange rates [4]. The Forex market functions as a platform for the union of buyers and sellers in the trading of currencies. The pricing of these currencies is subject to the multifarious interplay of market forces of supply and demand, which can be influenced by various factors including economic indicators, geopolitical events, and trends in the global market.

#### Top 10 Volume Forexes in 2010

1. US Dollar/Japanese Yen (USD/JPY)
2. Euro/US Dollar (EUR/USD)
3. British Pound/US Dollar (GBP/USD)
4. Australian Dollar/US Dollar (AUD/USD)
5. US Dollar/Swiss Franc (USD/CHF)
6. US Dollar/Canadian Dollar (USD/CAD)
7. Euro/Japanese Yen (EUR/JPY)
8. British Pound/Japanese Yen (GBP/JPY)
9. Euro/Swiss Franc (EUR/CHF)

## 10. New Zealand Dollar/US Dollar (NZD/USD)

### 3.2. Trading Terminologies

#### 3.2.1. Bid-Ask Spread



Figure 1: Bid and Ask Price from MetaTrader 5

It is the difference between the highest price that a buyer is willing to pay for a security (the bid price) and the lowest price that a seller is willing to accept (the asking price) [5]. The bid-ask spread represents the transaction cost of buying and selling a security, and it can vary depending on factors such as market liquidity, trading volume, and volatility.

#### 3.2.2. Limit Order

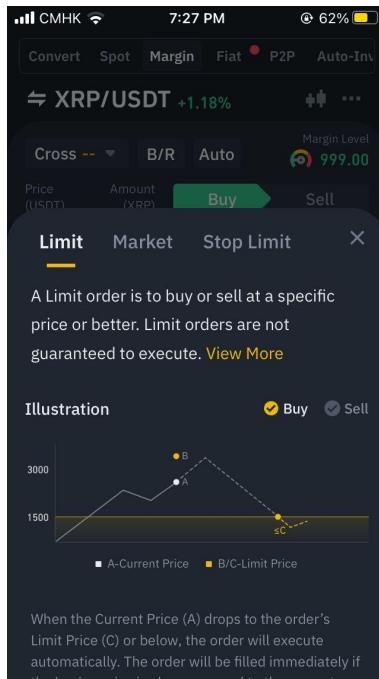


Figure 2: Principle of Limit Order from Binance

It is an order to buy or sell a security at a specific price or better [6]. A buy limit order is placed below the current market price, while a sell limit order is placed above the current market price. Limit orders are used to control the price at which a trade is executed and to avoid unfavorable price movements.

### 3.2.3. Stop Loss Order

It is an order to sell a security at a specific price or worse. It is used to limit potential losses on trade and to manage risk [7]. A sell-stop order is placed below the current market price, while a buy-stop order is placed above the current market price.

### 3.2.4. Margin

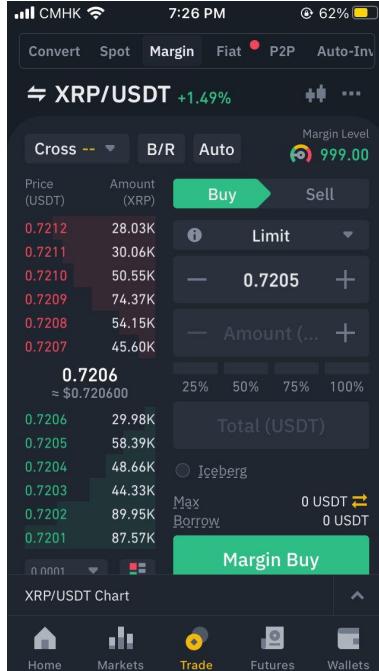


Figure 3: Margin on Binance

It denotes the guarantee that an investor must furnish to their broker or exchange as a safeguard against the credit risk that they may pose [8]. It allows traders to leverage their positions and potentially increase their profits, but it also increases their risk. Margin requirements can vary depending on the broker, the type of security being traded, and other factors.

### 3.2.5. Short Selling

It is the practice of selling a security that the trader does not own, intending to buy it back at a lower price in the future [9]. Short selling is used to profit from a decline in the price of a security. It can be risky, as losses can be unlimited if the price of the security rises instead of falling.

### 3.2.6. Liquidity

It refers to the ease with which a security can be bought or sold without significantly affecting its price [10]. Highly liquid securities can be traded quickly and with minimal impact on their price, while illiquid securities may be difficult to trade and can experience significant price movements as a result.

### 3.2.7. Volatility

It is a statistical metric that quantifies the extent of dispersion of returns for a particular security or market index [11]. High volatility can provide opportunities for traders to profit, but it also increases risk. Volatility can be influenced by a variety of factors, such as economic data releases, political events, and market sentiment.

### 3.2.8. Technical Analysis

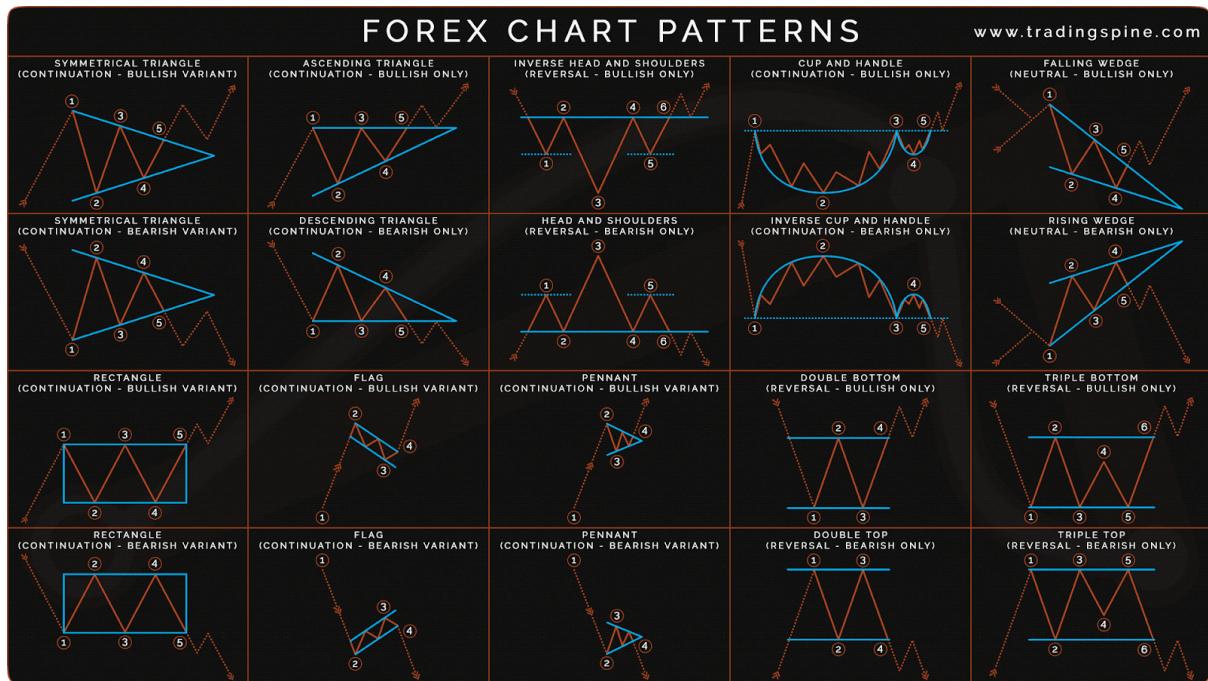


Figure 4: Chart Patterns from Trading Spine

It is a method of analyzing securities based on their price and volume movements [12]. It involves using charts and other tools to identify trends, patterns, and other signals that can help traders make trading decisions. Technical analysis can be used in conjunction with fundamental analysis, which involves analyzing a security's underlying financial and economic data.

## 3.3. Types of Machine Learning

### 3.3.1. Supervised Learning

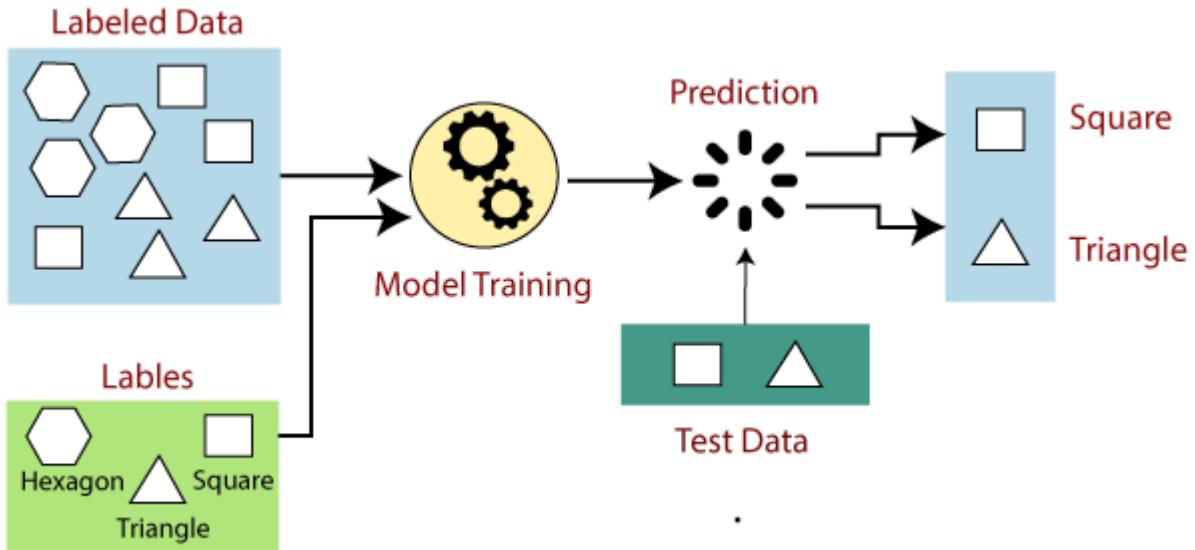


Figure 5: Supervised Learning from JavaTpoint

It is a type of machine learning where the algorithm is trained on a labeled dataset [13]. The input data is labeled with the correct output, and the algorithm learns to map the input to the correct output based on this labeled data.

Classification	Regression
The algorithm learns to classify input data into different categories based on the labeled data. For example, an algorithm might be trained to classify email messages as spam or not spam based on a labeled dataset of emails.	The algorithm learns to predict a continuous output value based on the labeled data. For example, an algorithm might be trained to predict the price of a house based on a labeled dataset of house prices and features.

Table 1: Types of Supervised Learning

### 3.3.2. Unsupervised Learning

It is a type of machine learning where the algorithm is trained on an unlabeled dataset [14]. The algorithm learns to find patterns and structure in the data on its own, without any explicit guidance.

<b>Clustering</b>	<b>Anomaly Detection</b>
The algorithm learns to group similar data points based on the underlying structure of the data. For example, an algorithm might be trained to cluster customers based on their purchasing habits.	The algorithm learns to identify data points that are significantly different from the rest of the data. For example, an algorithm might be trained to detect credit card fraud based on a dataset of credit card transactions.

Table 2: Types of Unsupervised Learning

### **3.3.3. Semi-Supervised Learning**

It is a machine-learning paradigm that leverages a sparse set of labeled data and an abundance of unlabeled data to train a predictive model [15]. It involves training an algorithm with a combination of labeled and unlabeled data, to improve the accuracy of the algorithm's predictions.

<b>Semi-Supervised Classification</b>
The algorithm is trained on a small amount of labeled data and a large amount of unlabeled data. The labeled data provides some guidance on the classification task, while the unlabeled data helps the algorithm to learn the underlying structure in the data.

Table 3: Example of Semi-supervised Learning

### **3.3.4. Reinforcement Learning**

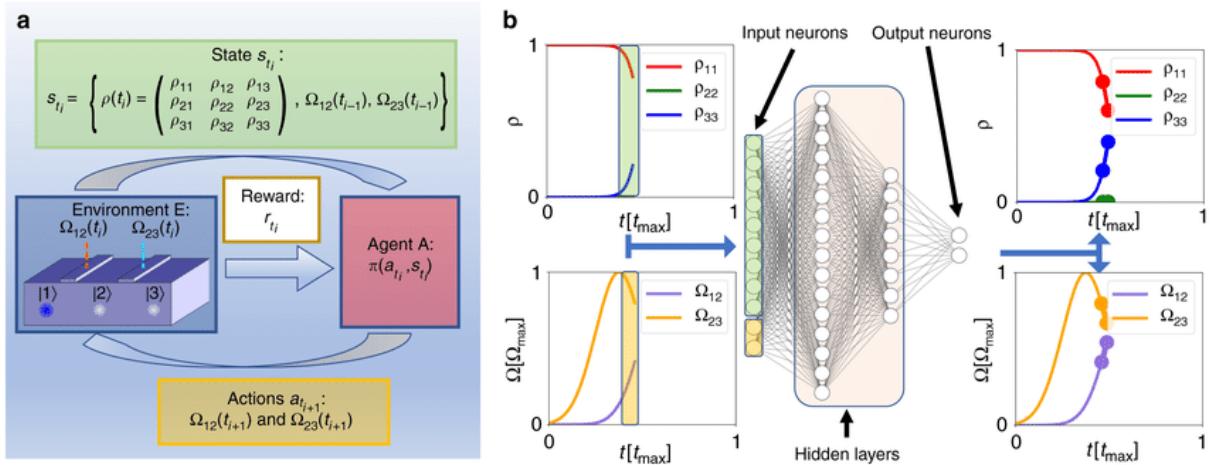


Figure 6: Deep Reinforcement Learning Architecture from Research Gate

It is a type of machine learning that involves training an agent to make decisions in a dynamic environment, based on a system of rewards and punishments [16].

Game Playing	Robotics
The algorithm learns to play a game by taking actions and receiving rewards or punishments based on the outcome of each action. The goal is to maximize the total reward throughout the game.	The algorithm learns to control a robot by taking actions and receiving feedback from the environment in the form of rewards or punishments. The goal is to perform a specific task, such as navigating a maze or manipulating objects.

Table 4: Types of Reinforcement Learning

### 3.3.5. Deep Learning

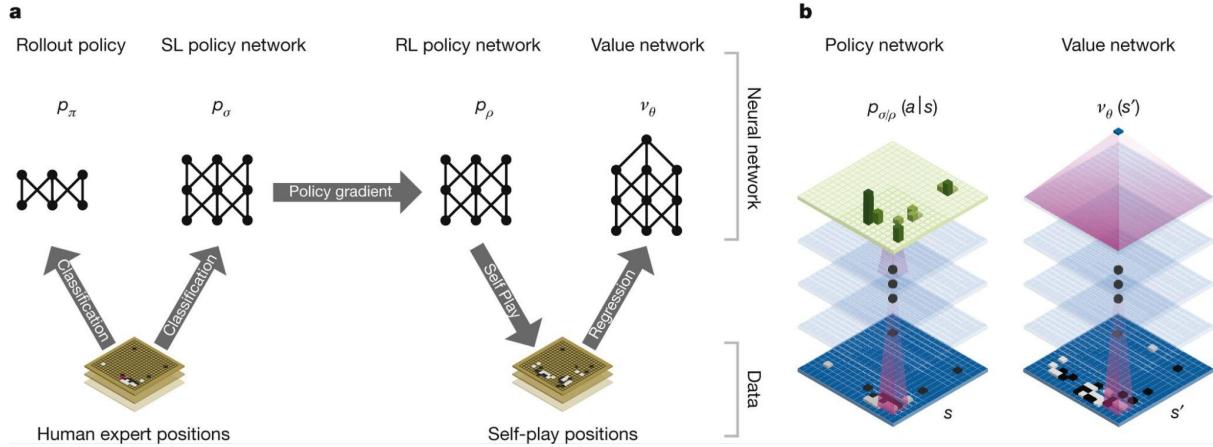


Figure 7: Deep Neural Networks and Tree Search of AlphaGo on Nature

It is a subset of machine learning that involves the use of neural networks with multiple layers [17]. These networks are designed to learn hierarchical representations of data, where each layer learns increasingly abstract features of the input data.

Image Recognition	Natural Language Processing
The network learns to identify objects in images based on a labeled dataset of images.	The network learns to process and understand natural language based on a labeled dataset of text data.

Table 5: Types of Deep Learning

## 4. Methodology

To facilitate machine trading for trading, the following library was employed.

Python Library
<ol style="list-style-type: none"> <li>1. math</li> <li>2. numpy</li> <li>3. pandas</li> <li>4. yfinance</li> <li>5. warnings</li> </ol>

- |  |
|--|
| <ol style="list-style-type: none"> <li>6. statistics</li> <li>7. matplotlib</li> <li>8. IPython</li> <li>9. scripy</li> <li>10. google</li> <li>11. sklearn</li> </ol> |
|--|

Table 6: Python Libraries Utilization

#### 4.1. Data Collection

Historical prices of the top 10 assets in forexes were gathered from Yahoo Finance API with Python. The data were retrieved on a daily time frame basis, which compromises open, high, low, close, adjusted close, and volume. The data was gathered from January 1, 2000, to January 1, 2020.

Provider	Period	Dataset
Yahoo Finance	2000-01-01 - 2020-01-01	<ol style="list-style-type: none"> <li>1. Open</li> <li>2. High</li> <li>3. Low</li> <li>4. Close</li> <li>5. Adjusted Close</li> <li>6. Volume</li> </ol>

Table 7: Context of Collected Data

#### 4.2. Software Framework

Google Colab would be used to employ trading strategies. Python libraries - math, numpy, pandas, statistics, matplotlib, and IPython would be used for visualization and data analysis. Lastly, Classification will be conducted with the Scikit-learn library, as general machine learning models - calibration, cluster, ensemble, gaussian process, isotonic, and more are available.

Colab	Runtime	Hardware	GPU Type	Runtime
-------	---------	----------	----------	---------

<b>Version</b>	<b>Type</b>	<b>Accelerator</b>		<b>Shape</b>
Pro	Python 3	TPU/GPU	v2/A100	High-RAM

Table 8: Setting of Google Colab Notebooks

### 4.3. Initial Account Setting

The initial capital in this study is HKD 10,000.

### 4.4. Ideation

Harnessing the collective power of all sci-kit-learn models for binary classification, this approach entails employing models that strive to fit the data through continuous regression, subsequently truncating the continuous predictions to yield discrete classifications. Intriguingly, linear regression, which constitutes the cornerstone of logistic regression. Thus, the logistic function inherently encapsulates a linear function enveloped within a sigmoid function. Despite possessing a narrower decision boundary, regressors possess the capacity to attain comparable levels of accuracy as classifiers. Therefore, our models embark on the classification endeavor by embracing regression as their modus operandi.

#### 4.4.1. Most Scikit-learn Models

This model utilized most models from Probability Calibration, Dummy estimators, Ensemble Methods, Gaussian Processes, Linear classifiers, Classical linear regressors, Regressors with variable selection, Bayesian regressors, Outlier-robust regressors, Generalized linear models (GLM) for regression, Miscellaneous, Naive Bayes, Nearest Neighbors, Neural network models, Semi-Supervised Learning, Support Vector Machines, and Decision Trees to compare their effectiveness.

#### Regressors As Classifiers

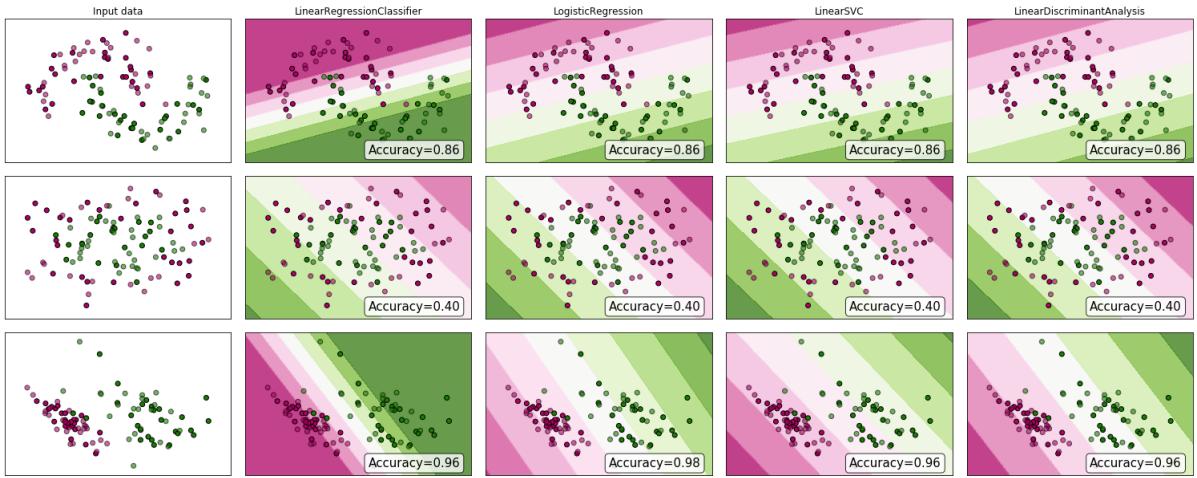


Figure 8: Accuracy of Classifiers and Regressors from Scikit-learn

Within the aforementioned feature space, it is noteworthy that a regression model can exhibit analogous behavior to classifiers, furnishing a hyperplane capable of partitioning the distinct classes. Consequently, regressors have the potential to achieve commensurate levels of accuracy as classifiers.

#### 4.4.2. Feature Discretization

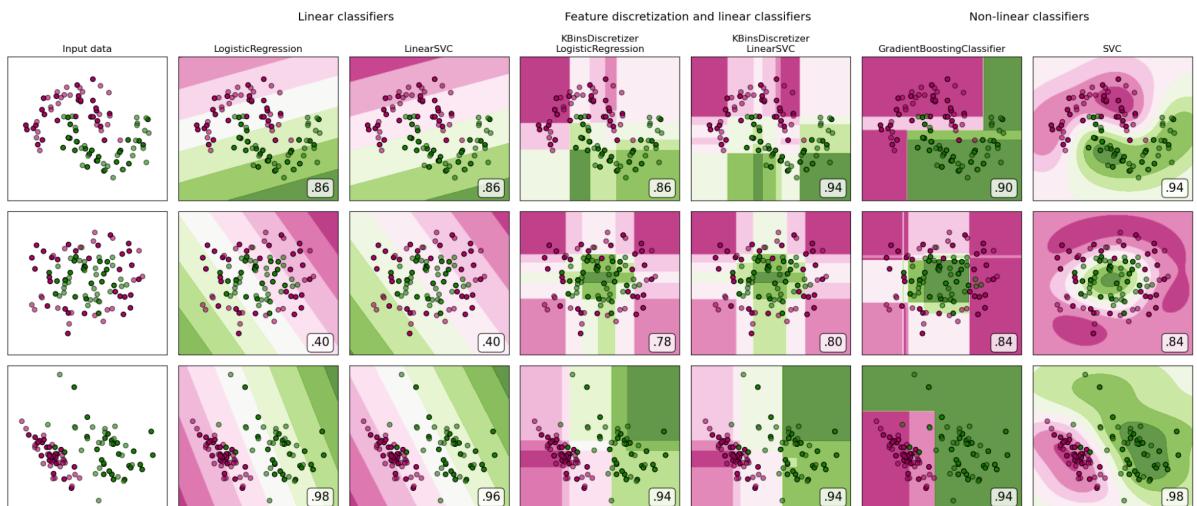


Figure 9: Feature Discretization from Scikit-learn

#### Dimension Reduction

The preliminary rows exhibit datasets characterized by non-linear separability, specifically exemplified by the intricate arrangements of the moons and concentric circles. In contrast, the third row portrays a dataset that manifests a semblance of

linearity in its separability. The judicious utilization of feature discretization yields a profound amplification in the effectiveness of linear classifiers when confronted with datasets that defy linear separability. Alas, when confronted with the linearly separable dataset, the efficacy of linear classifiers is woefully undermined by the application of feature discretization.

In this instance, price data is more analyzable when continuous price action is interpreted in daily batches with the same duration.

## 4.5. Strategies

After obtaining the data frame from yahoo finance, the data is processed to yield two new data frames (Volatility column, and Decision column). If the open price exceeds the close price, a decision of 1 will be generated; vice versa, a 0 will be generated. To associate trading with classification, trading behaviors, and profit, the Action column, and Profit column are created from the classification results. The trading rules for swing trading and day trading are derived from the same classification results (Decision column). Their only difference would be their final interpretation. No scaling would be used as the future prices of assets are unpredictable. To reduce complexity, no technical indicators will be employed in this study. All positions will be closed on the last day.

Action Type	
Swing Trading	Day Trading
<ul style="list-style-type: none"> <li>1. Buy</li> <li>2. Retain</li> <li>3. Sell</li> <li>4. No Action</li> </ul>	<ul style="list-style-type: none"> <li>1. Buy &amp; Sell</li> <li>2. No Action</li> </ul>

Table 9: Actions Derived from Machine Learning Strategies

As shown in the table above, two distinctive action sets for swing trading and day trading are yielded from the same classification results.

Step - 0							
	Date	Open	High	Low	Close	Adj Close	Volum e
0	2012-01-02 00:00:00	0.778089	0.780396	0.777122	0.778695	0.778695	0
1	2012-01-03 00:00:00	0.781006	0.790514	0.780518	0.781006	0.781006	0
2	2012-01-04 00:00:00	0.790326	0.790514	0.785731	0.790389	0.790389	0
3	2012-01-05 00:00:00	0.786596	0.787774	0.779727	0.786596	0.786596	0

Table 10: Original Data of [NZDUSD=X]

As shown in the above table, the original data is obtained from yahoo finance. No preprocessing is conducted yet.

Step - 1			
	...	Volatility	Decision
0	...	0.000000	0.000000
1	...	0.002968	1.000000

2	...	0.012014	0.000000
3	...	-0.004798	0.000000

Table 11: Generation of Volatility and Decision Columns

In the aforementioned table, two new columns - Volatility and Decision columns are created and appended to the original data sets. To infer today's action from the previous day, Decision columns are shifted one row up.

#### 4.5.1. Strategy 1 - Swing Trading

Positions remain open from one to multiple days. A Decision of 1 from the previous day triggers a Buy action tomorrow. A Sell action will be triggered with a Decision of 0. Retain will be triggered if a Decision of 1 is detected with opened positions. The position will only be entered and exited at an open price.

Step - 2			
	...	Swing Action	Swing Profit
0	...	No Action	10000.000000
1	...	No Action	10000.000000
2	...	Buy	5000.395466
3	...	Sell	9976.404403

Table 12: Attachment of Action and Profit Columns for Swing Trading

As revealed in the above table, the profit of the swing trading strategy is calculated based on the corresponding actions. Both columns are appended for further analysis.

#### 4.5.2. Strategy 2 - Day Trading

Positions remain open for only one day. A Decision of 1 from the previous day triggers a Buy & Sell action tomorrow. The position will only be entered at the open price and exited at the close price on the same day.

Step - 3			
	...	Day Action	Day Profit
0	...	No Action	10000.000000
1	...	No Action	10000.000000
2	...	Buy & Sell	10000.395535
3	...	No Action	10000.395535

Table 13: Attachment of Action and Profit Columns for Day Trading

As displayed in the above table, the profit of the day trading strategy is computed based on the actions taken, with both columns included for additional examination.

	Date	Open	High	Low	Close	Adj Close	Volume	Volatility	Decision	Swing Action	Swing Profit	Day Action	Day Profit
0	2000-01-03 00:00:00	102.070000	103.330002	101.309998	101.690002	101.690002	0	0.000000	0.000000	No Action	10000.000000	No Action	10000.000000
1	2000-01-04 00:00:00	101.639999	103.320000	101.470001	103.139999	103.139999	0	0.014259	1.000000	No Action	10000.000000	No Action	10000.000000
2	2000-01-05 00:00:00	103.129997	104.480003	102.750000	104.089996	104.089996	0	0.009211	1.000000	Buy	5049.760132	Buy & Sell	10185.279823
3	2000-01-06 00:00:00	104.089996	105.559998	103.900002	105.230003	105.230003	0	0.010952	1.000000	Retain	5049.760132	Buy & Sell	10407.581192
4	2000-01-07 00:00:00	105.260002	105.879997	104.830002	105.330002	105.330002	0	0.000950	1.000000	Retain	5049.760132	Buy & Sell	10421.371132
5	2000-01-10 00:00:00	105.099998	105.849998	104.279999	105.160004	105.160004	0	-0.001614	1.000000	Retain	5049.760132	Buy & Sell	10433.252159
6	2000-01-11 00:00:00	105.150002	106.309998	105.050003	106.019997	106.019997	0	0.008178	0.000000	Retain	5049.760132	Buy & Sell	10605.511192
7	2000-01-12 00:00:00	106.010002	107.309998	105.480003	105.709999	105.709999	0	-0.002924	1.000000	Sell	10138.240234	No Action	10605.511192
8	2000-01-13 00:00:00	105.720001	106.230003	105.389999	106.019997	106.019997	0	0.002933	0.000000	Buy	5169.400177	Buy & Sell	10665.510277
9	2000-01-14 00:00:00	106.080002	106.580002	105.410004	105.879997	105.879997	0	-0.001320	0.000000	Sell	10155.160263	No Action	10665.510277
10	2000-01-17 00:00:00	105.839996	105.860001	104.250000	104.889999	104.889999	0	-0.009350	1.000000	No Action	10155.160263	No Action	10665.510277

Figure 10: Data Frame after Preprocessing

Figure 11: Evaluation Array for all Regressors and Classifiers

## 4.6. Analytical Metrics

To determine the profitability of the strategies, a collection of classification, and trading metrics have been utilized. Below listed the contexts of each metric.

### 4.6.1. Classification Metrics

Most classification metrics have been embedded in this report. However, not all of them are analyzed because of the limitation of computing resources. Only the result of accuracy is further processed.

<b>metrics.accuracy_score</b>  (The higher the better)	This metric measures the proportion of correctly classified samples out of the total number of samples and therefore higher values indicate better performance.
<b>metrics.auc</b>  (The higher the better)	The area under the curve (AUC) measures the ability of a binary classifier to distinguish between positive and negative samples and values close to 1 indicate better performance.
<b>metrics.average_precision_score</b>  (The higher the better)	This metric measures the weighted average of precision achieved at different threshold values and therefore higher values indicate better performance.
<b>metrics.balanced_accuracy_score</b>  (The higher the better)	This metric is similar to accuracy but takes into account imbalanced datasets by computing the average of recall obtained for each class and therefore higher values indicate better

	performance.
<b>metrics.brier_score_loss</b>  (The lower the better)	This metric measures the mean squared difference between the predicted probability and the real outcome and therefore lower values indicate better performance.
<b>metrics.classification_report</b>	This is not a metric but a function that generates a text report with various classification metrics including precision, recall, and F1 score for each class. The interpretation of these metrics is similar to what is described for <code>metrics.precision_recall_fscore_support</code> .
<b>metrics.cohen_kappa_score</b>  (The higher the better)	This metric measures the agreement between two raters and values close to 1 indicate better agreement.
<b>metrics.confusion_matrix</b>	This is not a metric but a function that generates a matrix with the number of true positives, false positives, true negatives, and false negatives. The interpretation of this matrix depends on the context and the application.
<b>metrics.dcg_score</b>  (The higher the better)	This metric measures the quality of a ranking of documents and therefore higher values indicate better performance.
<b>metrics.det_curve</b>	This is not a metric but a function that generates a curve of detection rate (i.e.,

	true positive rate) as a function of false positive rate for different threshold values. The interpretation of this curve is similar to what is described for metrics.roc_curve.
<b>metrics.f1_score</b>  (The higher the better)	This metric is the harmonic mean of precision and recall and therefore higher values indicate better performance.
<b>metrics.fbeta_score</b>  (The higher the better)	This metric is a weighted harmonic mean of precision and recall with a weight parameter beta and therefore higher values indicate better performance.
<b>metrics.hamming_loss</b>  (The lower the better)	This metric measures the fraction of misclassified labels and therefore lower values indicate better performance.
<b>metrics.hinge_loss</b>  (The lower the better)	This metric measures the margin between the predicted score and the true score and therefore lower values indicate better performance.
<b>metrics.jaccard_score</b>  (The higher the better)	This metric measures the intersection over the union between two sets and therefore higher values indicate better similarity.
<b>metrics.log_loss</b>  (The lower the better)	This metric measures the negative log-likelihood of the predicted probability and therefore lower values indicate better performance.

<b>metrics.matthews_corrcoef</b>  (The lower the better)	This metric measures the correlation between the predicted and true binary labels and values close to 1 indicate better performance.
<b>metrics.multilabel_confusion_matrix</b>	This is not a metric but a function that generates a matrix of confusion matrices, one for each label. The interpretation of these matrices is similar to what is described for metrics.confusion_matrix.
<b>metrics.ndcg_score</b>  (The higher the better)	This metric is a normalized version of DCG and therefore higher values indicate better performance.
<b>metrics.precision_recall_curve</b>	This is not a metric but a function that generates a curve of precision as a function of recall for different threshold values. The interpretation of this curve is similar to what is described for metrics.roc_curve.
<b>metrics.precision_recall_fscore_support</b>	This is not a metric but a function that generates arrays of precision, recall, and F1 scores for each class. The interpretation of these metrics is similar to what is described for metrics.f1_score.
<b>metrics.precision_score</b>  (The higher the better)	This metric measures the fraction of true positives among the predicted positives and therefore higher values indicate better performance.
<b>metrics.recall_score</b>	This metric measures the fraction of

(The higher the better)	true positives among the actual positives and therefore higher values indicate better performance.
<b>metrics.roc_auc_score</b>  (The higher the better)	This metric measures the ability of a binary classifier to rank positive samples higher than negative samples and values close to 1 indicate better performance.
<b>metrics.roc_curve</b>	This is not a metric but a function that generates a curve of true positive rate as a function of false positive rate for different threshold values. The interpretation of this curve is similar to what is described for metrics.det_curve.
<b>metrics.top_k_accuracy_score</b>  (The higher the better)	This metric measures the proportion of samples for which the true label is among the top k predicted labels and therefore higher values indicate better performance.
<b>metrics.zero_one_loss</b>  (The higher the better)	This metric measures the fraction of misclassified samples and therefore lower values indicate better performance.

Table 14: Description of Classification Metrics

Illustrated in the table, the classification report helps us to evaluate the effectiveness of trading strategies. A great strategy should show a directly proportional relationship between classification and profit.

#### 4.6.2. Trading Metrics

To examine the profitability of models, major trading metrics have been added.

#### 4.6.2.1. Net Profit

It is the total profit or loss made by a trader over a specific period after all expenses and commissions have been accounted for. It is a measure of the trader's profitability.

$$\text{Net Profit} = \text{Total Revenue} - \text{Total Expenses} - \text{Total Losses}$$

where

1. *Total Revenue* is the total amount earned from the investment or portfolio,
2. *Total Expenses* is the total amount spent on expenses related to the investment or portfolio, and
3. *Total Losses* is the total amount lost from the investment or portfolio.

#### 4.6.2.2. Compounded Annual Growth Rate

It is the rate of return on an investment over a specific period, taking into account the effect of compounding on the investment returns. It is a measure of the trader's performance over time.

$$\text{CAGR} = \left( \frac{V_f}{V_i} \right)^{\frac{1}{n}} - 1$$

where

1.  $V_i$  is the initial value of the investment or portfolio,
2.  $V_f$  is the final value of the investment or portfolio after  $n$  years, and
3.  $\frac{1}{n}$  is the annualization factor.

```
def cagr(account_money, years):
    initial_value = account_money[0]
    final_value = account_money[-1]
    cagr = (final_value / initial_value) ** (1 / years) - 1
    return cagr
```

Code snippet 1: Compounded Annual Growth Rate

#### 4.6.2.3. Sharpe Ratio

It is a measure of the risk-adjusted return of an investment, taking into account the volatility of the investment returns and comparing them to the risk-free rate of return. It is a measure of the trader's ability to generate returns relative to the amount of risk taken.

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

where

1.  $R_p$  is the portfolio return,
2.  $R_f$  is the risk-free rate of return, and
3.  $\sigma_p$  is the standard deviation of the portfolio return.

```
def sharpe_ratio(money_change, num_years, risk_free_rate):
    trade_returns = []
    for i in range(len(money_change) - 1):
        trade_returns.append((money_change[i+1] - money_change[i]) / money_change[i])
    total_return = np.prod([1 + r for r in trade_returns]) - 1
    num_trades = len(trade_returns)
    annualized_return = (1 + total_return) ** (1 / num_years) - 1
    annualized_std_dev = np.std(trade_returns) * np.sqrt(num_trades / num_years)
    sharpe = (annualized_return - risk_free_rate) / annualized_std_dev
    if sharpe == np.inf or sharpe == -np.inf:
        sharpe = np.nan
    return sharpe
```

Code snippet 2: Sharpe Ratio

#### 4.6.2.4. Maximum Drawdown

It is the maximum loss a trader has experienced during a specific period, from the highest peak to the lowest trough. It is a measure of the trader's risk tolerance and ability to manage drawdowns.

$$\text{Maximum Drawdown} = \max_{i,j: i < j} \left\{ \frac{V_i - V_j}{V_i} \right\}$$

where

1.  $V_i$  is the portfolio value at time  $i$ ,
2.  $V_j$  is the portfolio value at time  $j$ , and
3. the maximum is taken over all pairs of indices  $i$  and  $j$  such that  $i < j$ .

```
def max_drawdown(prices):
    max_so_far = prices[0]
    max_drawdown = 0
    for price in prices:
        if price > max_so_far:
            max_so_far = price
        else:
            drawdown = 1 - price / max_so_far
            if drawdown > max_drawdown:
                max_drawdown = drawdown
    return max_drawdown
```

Code snippet 3: Maximum Drawdown

#### 4.6.2.5. Recovery Rate

It is the percentage of the maximum drawdown that a trader has been able to recover. It is a measure of the trader's ability to recover from losses.

$$\text{Recovery Rate} = \frac{\text{Amount Recovered}}{\text{Amount Owed}} \times 100\%$$

where

1.  $\text{Amount Recovered}$  is the total amount that is recovered after a default or other loss event, and
2.  $\text{Amount Owed}$  is the total amount that was originally lent or invested.

```
def recovery_rate(prices, max_drawdown):
    max_so_far = prices[0]
    for price in prices:
        if price > max_so_far:
            max_so_far = price
```

```

else:
    drawdown = 1 - price / max_so_far
    if drawdown > max_drawdown:
        max_drawdown = drawdown
recovery_rate = 1 - max_drawdown
return recovery_rate

```

Code snippet 4: Recovery Rate

#### 4.6.2.6. Maximum Profit

It is the highest profit a trader has made on a single trade. It is a measure of the trader's ability to identify and capitalize on profitable opportunities.

$$\text{Maximum Profit} = \max_{i,j:i < j} V_j - V_i$$

where

1.  $V_i$  is the portfolio value at time  $i$ , and  $V_i$  is the portfolio value at time  $i$ , and
2.  $V_j$  is the portfolio value at time  $j$ , and  $V_j$  is the portfolio value at time  $j$ , and
3. the maximum is taken over all pairs of indices  $i$  and  $j$  such that  $i < j$ .

#### 4.6.2.7. Maximum Loss

It is the highest loss a trader has experienced on a single trade. It is a measure of the trader's ability to manage risk and cut losses.

$$\text{Maximum Loss} = \max_{i,j:i < j} V_i - V_j$$

where

1.  $V_i$  is the portfolio value at time  $i$ , and  $V_i$  is the portfolio value at time  $i$ , and
2.  $V_j$  is the portfolio value at time  $j$ , and  $V_j$  is the portfolio value at time  $j$ , and
3. the maximum is taken over all pairs of indices  $i$  and  $j$  such that  $i < j$ .

#### 4.6.2.8. Total Trades

It is the total number of trades executed by a trader over a specific period. It is a measure of the trader's activity and engagement in the market.

$$\text{Total Trades} = \sum_{i=1}^n I_i$$

where

1.  $n$  is the total number of time periods, and  $n$  is the total number of time periods, and
2.  $I_i$  is an indicator variable that takes a value of 1 if a trade was executed at time  $i$ , and 0 otherwise.

## 5. Performance Evaluation

### Notice

1. The analysis is pruned due to the limitation of computing power and cost.
2. Only accuracy is analyzed.
3. Only forex with the top 10 market capitalization is tested.
1. Please notice regressors are used with normalization in the range of [0,1].
2. Use asset\_report(, True) to see the detailed classification report.
3. No Scaling is conducted as the future range of assets is unpredictable.

### Dataframes

[yFinance]

Training	Trading
1. 2000-01-01 - 2010-01-01 (Daily)	1. 2010-01-01 - 2020-01-01 (Daily)

### Scalers

[Sklearn]

[Deprecated]

1. MinMaxScaler(feature\_range=(0, 100)),
2. StandardScaler()

<b>Metrics</b> [Sklearn]	
<b>Trading</b>	<b>Classification</b>
1. Net Profit 2. Compound Annual Growth Rate 3. Sharpe ratio 4. Maximum Dropdown 5. Recovery Rate 6. Maximum Profit 7. Maximum Loss 8. Maximum Holding Time	1. metrics.accuracy_score, 
	[Deprecated] 2. metrics.auc, 3. metrics.average_precision_score, 4. metrics.balanced_accuracy_score, 5. metrics.brier_score_loss, 6. metrics.classification_report, 7. metrics.cohen_kappa_score, 8. metrics.confusion_matrix, 9. metrics.dcg_score, 10. metrics.det_curve, 11. metrics.f1_score, 12. metrics.fbeta_score, 13. metrics.hamming_loss, 14. metrics.hinge_loss, 15. metrics.jaccard_score, 16. metrics.log_loss,

	17. metrics.matthews_corrcoef, 18. metrics.multilabel_confusion_matrix, 19. metrics.ndcg_score, 20. metrics.precision_recall_curve, 21. metrics.precision_recall_fscore_support, 22. metrics.precision_score, 23. metrics.recall_score, 24. metrics.roc_auc_score, 25. metrics.roc_curve, 26. metrics.top_k_accuracy_score, 27. metrics.zero_one_loss,
--	--

## Assets

[GPT-3]

Stock	Commodity	Forex
[Deprecated] <ul style="list-style-type: none"> <li>1. 'BAC', # Bank of America</li> <li>2. 'C', # Citigroup</li> <li>3. 'F', # Ford Motor Company</li> <li>4. 'GE', # General Electric</li> <li>5. 'CSCO', # Cisco Systems</li> <li>6. 'PFE', # Pfizer Inc.</li> <li>7. 'MSFT', # Microsoft Corporation</li> </ul>	[Deprecated] <ul style="list-style-type: none"> <li>1. 'CL=F', # Crude oil</li> <li>2. 'GC=F', # Gold</li> <li>3. 'NG=F', # Natural gas</li> <li>4. 'ZC=F', # Corn</li> <li>5. 'HO=F', # Heating Oil</li> <li>6. 'BZ=F', # Brent Oil</li> <li>7. 'SI=F', # Silver</li> <li>8. 'ZS=F', # Soybeans</li> </ul>	<ul style="list-style-type: none"> <li>1. 'USDJPY=X', # US Dollar/Japanese Yen</li> <li>2. 'EURUSD=X', # Euro/US Dollar</li> <li>3. 'GBPUSD=X', # British Pound/US Dollar</li> <li>4. 'AUDUSD=X', # Australian Dollar/US Dollar</li> <li>5. 'USDCHF=X', # US Dollar/Swiss Franc</li> </ul>

8. 'AA', # Alcoa Inc. 9. 'T', # AT&T Inc. 10. 'INTC' # Intel Corporation	9. 'HG=F', # Copper 10. 'ZW=F', # Wheat	6. 'USDCAD=X', # US Dollar/Canadian Dollar 7. 'EURJPY=X', # Euro/Japanese Yen 8. 'GBPJPY=X', # British Pound/Japanese Yen 9. 'EURCHF=X', # Euro/Swiss Franc 10. 'NZDUSD=X', # New Zealand Dollar/US Dollar
--	--	--

### Models

[Sklearn]

Probability Calibration	Dummy estimators	Ensemble Methods
1. calibration.CalibratedClassifierCV(),	2. dummy.DummyClassifier(), 3. dummy.DummyRegressor(),	4. ensemble.AdaBoostClassifier(), 5. ensemble.AdaBoostRegressor(), 6. ensemble.BaggingClassifier(), 7. ensemble.BaggingRegressor(), 8. ensemble.ExtraTreesClassifier(),

		9. ensemble.ExtraTreesRegressor(), 10. ensemble.GradientBoostingClassifier(), 11. ensemble.GradientBoostingRegressor(), 12. ensemble.IsolationForest(), 13. ensemble.RandomForestClassifier(), 14. ensemble.RandomForestRegressor(), 15. ensemble.HistGradientBoostingRegressor(), 16. ensemble.HistGradientBoostingClassifier(),
<b>Gaussian Processes</b>	<b>Linear classifiers</b>	<b>Classical linear regressors</b>
17. gaussian_process.GaussianProcessClassifier(), 18. gaussian_process.GaussianProcessRegressor(),	19. linear_model.LogisticRegression(), 20. linear_model.LogisticRegressionCV(), 21. linear_model.PassiveAggressiveClassifier(),	27. linear_model.LinearRegression(), 28. linear_model.Ridge(), 29. linear_model.RidgeCV(), 30. linear_model.SGDRegressor(),

	22. linear_model.Perceptron(), 23. linear_model.RidgeClassifier(), 24. linear_model.RidgeClassifierCV(), 25. linear_model.SGDClassifier(), 26. linear_model.SGDOneClassSVM(),	
<b>Regressors with variable selection</b>	<b>Bayesian regressors</b>	<b>Outlier-robust regressors</b>
31. linear_model.ElasticNet(), 32. linear_model.ElasticNetCV(), 33. linear_model.Lars(), 34. linear_model.LarsCV(), 35. linear_model.Lasso(), 36. linear_model.LassoCV(), 37. linear_model.LassoLars(), 38. linear_model.LassoLarsCV(), 39. linear_model.LassoLarsIC(),	42. linear_model.ARDRegression(), 43. linear_model.BayesianRidge(),	44. linear_model.HuberRegressor(), 45. linear_model.TheilSenRegressor(),

40.linear_model.Ort hogonalMatching Pursuit(), 41. linear_model.Ort hogonalMatching PursuitCV(),		
<b>Generalized linear models (GLM) for regression</b>	<b>Miscellaneous</b>	<b>Naive Bayes</b>
46.linear_model.Pois sonRegressor(), 47. linear_model.Twe edieRegressor(),	48.linear_model.Pass iveAggressiveRegr essor(),	49. naive_bayes.Bern oulliNB(), 50. naive_bayes.Gaus sianNB(),
<b>Nearest Neighbors</b>	<b>Neural network models</b>	<b>Semi-Supervised Learning</b>
51. neighbors.KNeigh borsClassifier(), 52. neighbors.KNeigh borsRegressor(), 53. neighbors.Radius NeighborsRegress or(), 54. neighbors.Nearest Centroid(),	55. neural_network. MLPClassifier(), 56. neural_network. MLPRegressor(),	57. semi_supervised. LabelPropagation() , 58. semi_supervised. LabelSpreading(),
<b>Support Vector Machines</b>	<b>Decision Trees</b>	
59. svm.LinearSVC(), 60. svm.LinearSVR(), 61. svm.NuSVR(),	65. tree.DecisionTree Classifier(),	

62.svm.OneClassSV M(), 63.svm.SVC(), 64.svm.SVR(),	66.tree.DecisionTree Regressor(), 67.tree.ExtraTreeClas sifier(), 68.tree.ExtraTreeReg ressor(),	
---	---	--

Table 15: Context of Multiple Model-based Trading Strategy

## 5.1. Visualization

### 5.1.1. Strategy 1 - Swing Trading

The detailed report of the swing trading strategy could be found in Appendix A.

#### 5.1.1.2. Accuracy

	Asset	Model	Accuracy	Net Profit
1	USDCHF=X	CalibratedClas sifierCV	64.5725 %	-749.515360
2	USDCHF=X	RidgeClassifie rCV	64.5725 %	85.232508
3	USDCHF=X	LogisticRegres sionCV	64.5253 %	101.367663
4	USDCHF=X	MLPRegressor	64.4308 %	91.833141
5	USDCHF=X	RadiusNeigh orsRegressor	64.1946 %	272.701421

Table 16: Result with Top 5 Accuracy

Among the available forex markets, USDCHF=X demonstrates the highest level of predictability. Nevertheless, accurately predicting binary outcomes of 0 or 1 does not lead to a significant increase in net profit. To enhance the alignment between accuracy and profit, it may be necessary to adjust the accuracy metric to focus solely on correctly predicting either 0 or 1 independently. Moreover, accuracy tends to rise as the number of trades decreases. This heightened accuracy is a result of reduced

trading activity.

#### 5.1.1.2. Compound Annual Growth Rate

	<b>Asset</b>	<b>Model</b>	<b>Compound Annual Growth Rate</b>	<b>Net Profit</b>
1	GBPJPY=X	RidgeClassifier	37.8626 %	52985.301506
2	GBPJPY=X	RidgeClassifierCV	37.6160 %	52342.094276
3	GBPJPY=X	LogisticRegression	33.5976 %	42603.612122
4	EURJPY=X	LinearSVR	30.5949 %	39863.330818
5	AUDUSD=X	LinearRegression	30.0322 %	25784.408967

Table 17: Result with Top 5 Compound Annual Growth Rate

Among the models, RidgeClassifier stands out as the most profitable on GBPJPY=X. The outcomes of RidgeClassifier and RidgeClassifierCV are comparable, given their classification within the Linear classifiers category. A swing trading strategy, tailored to capture market trends, has been employed, resulting in robust performance. Furthermore, it is noteworthy that the majority of top-performing models have generated remarkable profits.

#### 5.1.1.3. Sharpe Ratio

	<b>Asset</b>	<b>Model</b>	<b>Sharpe Ratio</b>	<b>Net Profit</b>
1	USDCHF=X	TheilSenRegressor	3.000000	1377.845525
2	EURJPY=X	LinearSVR	2.191932	39863.330818
3	AUDUSD=X	LinearRegression	2.150565	25784.408967
4	AUDUSD=X	HuberRegressor	2.150565	25784.408967

5	USDJPY=X	TheilSenRegressor	2.047036	50270.607712
---	----------	-------------------	----------	--------------

Table 18: Result with Top 5 Sharpe Ratio

Unlike other top-performing models, TheilSenRegressor does not exhibit excessive returns on USDCHF=X. Despite its reduced risk, this model infrequently initiates trading. Conversely, TheilSenRegressor has achieved significant profitability on USDJPY=X, despite a lower Sharpe ratio. This suggests that volatility plays a crucial role in maximizing profits, highlighting the limited potential gains of a low-risk strategy. Nonetheless, it is important to note that models with a high Sharpe ratio typically demonstrate outstanding profitability.

#### 5.1.1.4. Maximum Drawdown

	Asset	Model	Maximum Drawdown	Net Profit
1	GBPJPY=X	MLPClassifier	85.2956 %	-8529.563568
2	EURJPY=X	MLPClassifier	60.2867 %	-6017.995468
3	USDCHF=X	RandomForest Classifier	25.4931 %	-2437.002280
4	AUDUSD=X	MLPRegressor	24.3309 %	-2433.085092
5	USDCHF=X	ExtraTreesRegressor	23.7798 %	-2277.876966

Table 19: Result with Top 5 Maximum Drawdown

MLPClassifier exhibits the poorest performance among the models when considering Maximum Drawdown on GBPJPY=X. Both MLPClassifier and MLPRegressor consistently incur losses on GBPJPY=X, EURJPY=X, and AUDUSD=X. These results suggest that MLP requires parameter tuning to improve its performance. Additionally, the strategy limits further losses by ceasing trading when the original capital is reduced by half or more.

#### 5.1.2. Strategy 2 - Day Trading

The detailed report of the swing trading strategy could be found in Appendix B.

### 5.1.2.1. Accuracy

	<b>Asset</b>	<b>Model</b>	<b>Accuracy</b>	<b>Net Profit</b>
1	USDCHF=X	CalibratedClassifierCV	64.5725 %	-2.809674
2	USDCHF=X	RidgeClassifierCV	64.5725 %	-29.829384
3	USDCHF=X	NuSVR	64.5253 %	3.305626
4	USDCHF=X	LogisticRegressionCV	64.5253 %	-28.825868
5	USDCHF=X	MLPRegressor	64.4308 %	3.112147

Table 20: Result with Top 5 Accuracy

Despite the superior accuracy of CalibratedClassifierCV and RidgeClassifierCV, their profitability remains inadequate due to limited trading activity. A decrease in the number of trades corresponds to a reduction in prediction error. Consequently, a minimal trading threshold may be necessary for a comprehensive evaluation of the models' performance.

### 5.1.2.2. Compounded Annual Growth Rate

	<b>Asset</b>	<b>Model</b>	<b>Compound Annual Growth Rate</b>	<b>Net Profit</b>
1	NZDUSD=X	DummyClassifier	2.6956 %	1640.929907
2	NZDUSD=X	SGDClassifier	2.6956 %	1640.929907
3	NZDUSD=X	BernoulliNB	2.6956 %	1640.929907
4	NZDUSD=X	Perceptron	2.6956 %	1640.929907
5	NZDUSD=X	RadiusNeighborsRegressor	2.6956 %	1640.929907

Table 21: Result with Top 5 Compounded Annual Growth Rate

The prevalence of the DummerClassifier highlights the ineffectiveness of the day trading strategy. Given its inability to extract meaningful insights from the dataset, the consistent outcomes observed in the top-performing models suggest that they

merely produce a repetitive sequence of 1s. Consequently, these findings reveal the failure of most models in identifying patterns within the price action under the day trading framework.

#### 5.1.2.3. Sharpe Ratio

	<b>Asset</b>	<b>Model</b>	<b>Sharpe Ratio</b>	<b>Net Profit</b>
1	NZDUSD=X	Perceptron	3.000000	1640.929907
2	NZDUSD=X	DummyClassifier	3.000000	1640.929907
3	NZDUSD=X	RadiusNeighborsRegressor	3.000000	1640.929907
4	NZDUSD=X	PassiveAggressiveClassifier	3.000000	1640.929907
5	NZDUSD=X	Lasso	3.000000	1640.929907

Table 22: Result with Top 5 Sharpe Ratio

The initial Sharpe ratio raises concerns due to its susceptibility to data snooping and calculation errors. Consequently, this report utilizes a modified version of the Sharpe ratio. The results indicate that the majority of models are unable to generate intelligent decisions, with only those significantly exceeding the threshold being considered satisfactory.

#### 5.1.2.4. Maximum Drawdown

	<b>Asset</b>	<b>Model</b>	<b>Maximum Drawdown</b>	<b>Net Profit</b>
1	USDJPY=X	BernoulliNB	54.5315 %	-5452.733780
2	USDJPY=X	ARDRegression	49.8830 %	-4985.915359
3	USDJPY=X	TheilSenRegressor	49.4338 %	-4939.594887
4	USDJPY=X	LogisticRegression	47.9751 %	-4791.089417
5	USDJPY=X	MLPRegressor	46.7477 %	-4668.786728

Table 23: Result with Top 5 Maximum Drawdown

Within the day trading schema, BernoulliNB has experienced the greatest capital loss. To identify the optimal models, considering a recovery rate as a filtering criterion may be necessary. Additionally, USDJPY=X proves to be the most challenging forex for implementing this strategy. A stop-loss mechanism has been implemented, temporarily halting trading activities when the capital is reduced by half.

## 5.2. Evaluation

Accuracy	Compound Annual Growth Rate	Sharpe Ratio	Maximum Dropdown
Accuracy is not directly proportional to net profit.	The higher the compound annual growth rate, the higher the net profit.	The higher the Sharpe ratio, the lower the risk per unit of returns.	The higher the compound annual growth rate, the higher the net profit.

Table 24: Metrics relationship with Net Profit

$$\text{Total Results} = 10 \text{ forexes} * 2 \text{ scalers} * 68 \text{ models} = 1360 \text{ results}$$

Although this approach has included the scaler in the array. After trial and error, It is proven to be ineffective and deprecated for trading. To obtain the number of unique results, the total results should be divided in half.

$$\text{Unique Results} = 1360 * 0.5 = 680 \text{ results}$$

The presented metric-based optimization is hardly perfect regarding the different trading situations. For instance, the accuracy, and maximum drawdown could be maximized and minimized due to the lack of trading for the strategy. The fewer the trades, the higher the accuracy. Therefore, an additional filter has to be employed to filter out trading approach that seldom trades.

## 5.3. Comparison

### 5.3.1. Filtered Models VS All Models

For numerical comparison, it is recommended to view the Colab notebooks. It can even run multiple assets for analysis. With the exhaustion of computation resources, it only analyses forex data with accuracy metrics.

```
Gain of diversify investment in swing trading:  
net profit ---> -8220001.630469799 ( -0.8456791800894855 % )
```

Figure 12: Dummy Investment with All Models in Swing Trading Strategy

```
Gain of diversify investment in day trading:  
net profit ---> -13209162.181232274 ( -1.0075638582175648 % )
```

Figure 13: Dummy Investment with All Models in Day Trading Strategy

Despite the superiority of the performance of an individual model, the overall return of using all models to trade is not great. Therefore, we should always select the best few models for the sake of profit maximization.

### 5.3.1. Filtered models VS Buy-and-Hold Approach

	asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
680	USDCAD=X	CalibratedClassifierCV	5.802740	0.497167	2847.411386	-556.082358	-0.009811	2.999649	0.057270	0.942730	21.056642	-13.167994	1292
723	USDCAD=X	HuberRegressor	5.802740	0.513220	2847.411386	-414.852934	-0.007275	2.999648	0.043251	0.956749	21.160768	-13.202031	1040
729	USDCAD=X	GaussianNB	5.802740	0.507082	2847.411386	-459.527938	-0.008074	2.999655	0.049582	0.950418	21.156657	-13.192832	1167
728	USDCAD=X	BernoulliNB	5.802740	0.380076	2847.411386	-673.228026	-0.011939	2.999738	0.078747	0.921253	61.641415	-38.334748	2116
727	USDCAD=X	PassiveAggressiveRegressor	5.802740	0.498584	2847.411386	-164.739622	-0.002859	2.999746	0.028256	0.971744	61.641415	-38.334748	859
726	USDCAD=X	TweedieRegressor	5.802740	0.500472	2847.411386	-539.454992	-0.009511	2.999650	0.054946	0.945054	21.090894	-13.189152	1271
725	USDCAD=X	PoissonRegressor	5.802740	0.500472	2847.411386	-539.454992	-0.009511	2.999650	0.054946	0.945054	21.090894	-13.189152	1271
724	USDCAD=X	TheilSenRegressor	5.802740	0.486308	2847.411386	-464.927482	-0.008171	2.999648	0.049593	0.950407	24.256355	-13.167074	1113
722	USDCAD=X	BayesianRidge	5.802740	0.500472	2847.411386	-539.454992	-0.009511	2.999650	0.054946	0.945054	21.090894	-13.189152	1271
748	USDCAD=X	CalibratedClassifierCV	5.802740	0.497167	2847.411386	-556.082358	-0.009811	2.999649	0.057270	0.942730	21.056642	-13.167994	1292

Figure 14: Maximum Profit for the Buy-and-Hold Approach [USDCAD=X]

The profit threshold represents the profit of a Buy-and-Hold approach. Illustrated in the above figure, the profit of the buy-and-hold approach is surprisingly low compared to ML approaches. Simple supervised learning models - logistic regression or linear regression can already achieve great results. Also, regressors could have a

similar effect with classifiers for classification.

## **5.4. Abnormality**

Dummy Regressor/Classifier is performing great in certain strategies. It has implied a great model should significantly outperform not only buy-and-hold approaches but also the dummy models. Its occupancy is an implication of successful bad trade. As dummy models fail to deliver constancy, the result is valid.

# **6. Improvement**

## **6.1. Hardware**

The temporal constraints imposed by Google Colab, which may be as restrictive as 12 hours per session, can pose a significant challenge for complex, long-running machine-learning tasks that necessitate days or even weeks of uninterrupted processing. Furthermore, the notebook's limited functionality and inability to generate comprehensive reports for all classification metrics underscore the necessity of leveraging a more potent TPU/GPU to unlock the full potential of machine learning models.

## **6.2. Data Source**

Despite the vast quantity of data provided by Yahoo Finance's Python API, its quality is subject to variability, with potential discrepancies arising from the absence of adjustments for corporate actions such as dividends, stock splits, and other related events. This may impede the accuracy and reliability of any ensuing analysis. Moreover, the API's coverage may be incomplete, with less prominent or smaller stocks omitted from the dataset. Such limitations may restrict the scope of analysis, rendering Yahoo Finance's stock data an unsuitable option for professional trading analysis, particularly given its lack of inclusion of spreads.

## **6.3. Technical Analysis**

The machine learning models solely deduce trading action based on rudimentary upward or downward trends in the market, which may prove inadequate for professional traders who typically rely on a range of technical indicators to inform their decisions. As such, the incorporation of classical trading patterns, such as

double tops, higher highs, and cup-and-handle formations, is imperative for a more nuanced and sophisticated approach to trading.

## 7. Conclusion

To conclude, machine learning models have yielded auspicious outcomes, with supervised learning paradigms proving effective in the classification of market trends and patterns. Such models have empowered traders to cultivate intricate strategies for the anticipation of forthcoming market movements, furnishing them with a marked competitive advantage over their counterparts.

Machine learning models have evinced noteworthy potential for the recognition of market trends, the prognostication of future movements, and the establishment of remunerative trading strategies. Despite the substantial demands of resources and proficiency requisite for their implementation, the potential advantages of these models render them a compelling toolset for traders striving to acquire a competitive edge in the mercurial and constantly evolving realm of finance.

## **8. References**

- [1] V. K. Pandey et al., “Machine learning algorithms and fundamentals as emerging safety tools in preservation of fruits and vegetables: A review,” MDPI, <https://www.mdpi.com/2227-9717/11/6/1720>.
- [2] A. Hayes, “Stocks: What they are, main types, how they differ from bonds,” Investopedia, <https://www.investopedia.com/terms/s/stock.asp>.
- [3] J. Fernando, “What is a commodity and understanding its role in the stock market,” Investopedia, <https://www.investopedia.com/terms/c/commodity.asp>.
- [4] C. Mitchell, “Forex (FX): Definition, how to trade currencies, and examples,” Investopedia, <https://www.investopedia.com/terms/f/forex.asp>.
- [5] A. Ganti, “What is a bid-ask spread, and how does it work in trading?,” Investopedia, <https://www.investopedia.com/terms/b/bid-askspread.asp>.
- [6] M. J. Kramer, “What is a limit order in trading, and how does it work?,” Investopedia, <https://www.investopedia.com/terms/l/limitorder.asp>.
- [7] M. J. Kramer, “Stop-loss orders: One way to limit losses and reduce risk,” Investopedia, <https://www.investopedia.com/terms/s/stop-lossorder.asp>.
- [8] J. Fernando, “Margin and margin trading explained plus advantages and disadvantages,” Investopedia, <https://www.investopedia.com/terms/m/margin.asp>.
- [9] A. Hayes, “Short selling: Definition, Pros, Cons, and examples,” Investopedia, <https://www.investopedia.com/terms/s/shortselling.asp>.
- [10] A. Hayes, “Understanding liquidity and how to measure it,” Investopedia, <https://www.investopedia.com/terms/l/liquidity.asp>.
- [11] A. Hayes, “Volatility: Meaning in finance and how it works with stocks,” Investopedia, <https://www.investopedia.com/terms/v/volatility.asp>.
- [12] S. Seth, “Technical analysis for stocks: Beginners overview,” Investopedia, <https://www.investopedia.com/articles/active-trading/102914/technical-analysis-strategies-beginners.asp>.
- [13] A. S. Gillis and D. Petersson, “What is supervised learning?: Definition from TechTarget,” TechTarget, <https://www.techtarget.com/searchenterpriseai/definition/supervised-learning>.
- [14] A. Ross, “What is unsupervised learning?,” Unsupervised, <https://unsupervised.com/resources/blogs/what-is-unsupervised-learning/>.

- [15] P. Potrimba, “What is semi-supervised learning? A guide for beginners.,” Roboflow, <https://blog.roboflow.com/what-is-semi-supervised-learning/>.
- [16] P. Bajaj, “Reinforcement learning,” GeeksforGeeks, <https://www.geeksforgeeks.org/what-is-reinforcement-learning/>.
- [17] K. Reyes, “What is Deep Learning and How Does It Works [Explained],” Simplilearn, <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning>.

## 9. Appendix

### A - Swing Trading Summary

asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
612	USDCHF=X	CalibratedClassifierCV	5.800000	0.645725	636.181269	-749.515360	-0.013343	0.734512	0.078790	0.921210	38.388565	-787.903925
635	USDCHF=X	RidgeClassifierCV	5.800000	0.645725	636.181269	85.232508	0.001464	-3.000000	0.000000	1.000000	41.848689	0.000000
544	USDCHF=X	CalibratedClassifierCV	5.800000	0.645725	636.181269	-749.515360	-0.013343	0.734512	0.078790	0.921210	38.388565	-787.903925
567	USDCHF=X	RidgeClassifierCV	5.800000	0.645725	636.181269	85.232508	0.001464	-3.000000	0.000000	1.000000	41.848689	0.000000
563	USDCHF=X	LogisticRegressionCV	5.800000	0.645253	636.181269	101.367663	0.001740	-2.337815	0.000000	1.000000	41.848689	0.000000
631	USDCHF=X	LogisticRegressionCV	5.800000	0.645253	636.181269	101.367663	0.001740	-2.337815	0.000000	1.000000	41.848689	0.000000
667	USDCHF=X	MLPRegressor	5.800000	0.644308	636.181269	91.833141	0.001577	-1.780331	0.000000	1.000000	41.775069	0.000000
596	USDCHF=X	RadiusNeighborsRegressor	5.800000	0.641946	636.181269	272.701421	0.004650	0.504944	0.000906	0.999094	149.345701	-9.112154
664	USDCHF=X	RadiusNeighborsRegressor	5.800000	0.641946	636.181269	272.701421	0.004650	0.504944	0.000906	0.999094	149.345701	-9.112154
599	USDCHF=X	MLPRegressor	5.800000	0.639584	636.181269	-925.448974	-0.016604	0.773764	0.093875	0.906125	45.409957	-789.028125
												23

Figure 15: Accuracy Report [high to low]

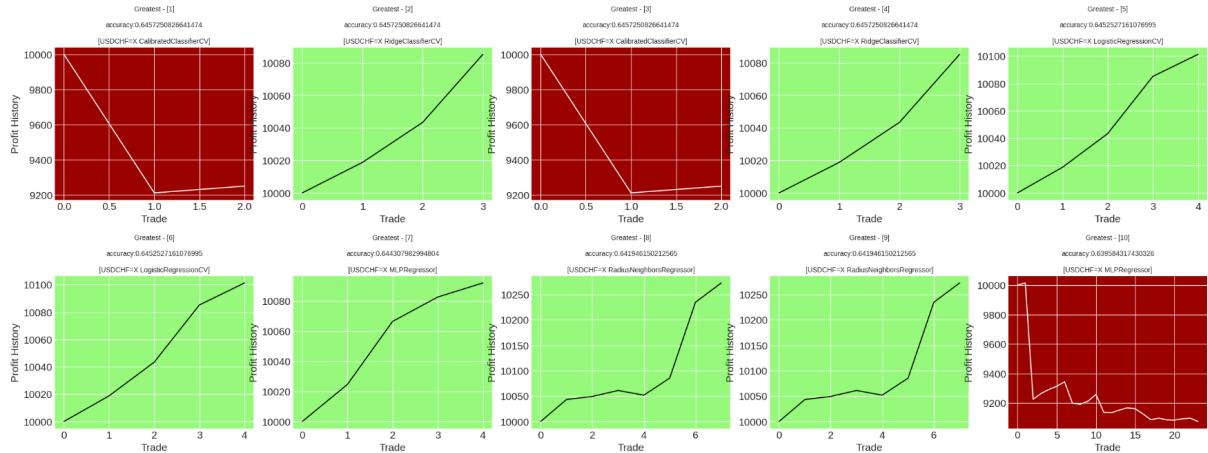


Figure 16: Accuracy Graph [high to low]

asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
1042	GBPJPY=X	RidgeClassifier	5.731507	0.507170	1683.827545	52985.301506	0.378626	1.843910	0.013219	0.986781	1516.726883	-242.171417
974	GBPJPY=X	RidgeClassifier	5.731507	0.507170	1683.827545	52985.301506	0.378626	1.843910	0.013219	0.986781	1516.726883	-242.171417
1043	GBPJPY=X	RidgeClassifierCV	5.731507	0.504302	1683.827545	52342.094276	0.376160	1.835712	0.013448	0.986652	1505.655884	-241.200256
975	GBPJPY=X	RidgeClassifierCV	5.731507	0.504302	1683.827545	52342.094276	0.376160	1.835712	0.013448	0.986652	1505.655884	-241.200256
1038	GBPJPY=X	LogisticRegression	5.731507	0.502868	1683.827545	42603.612122	0.335976	1.695421	0.013475	0.986625	1460.707809	-306.787582
970	GBPJPY=X	LogisticRegression	5.731507	0.502868	1683.827545	42603.612122	0.335976	1.695421	0.013475	0.986625	1460.707809	-306.787582
943	EURJPY=X	LinearSVR	6.019178	0.525262	884.481636	39863.330818	0.305949	2.191932	0.010018	0.989982	677.700233	-157.680176
434	AUDUSD=X	LinearRegression	4.854795	0.516366	-3186.365659	25784.408967	0.300322	2.150565	0.014615	0.985385	465.473476	-358.894947
451	AUDUSD=X	HuberRegressor	4.854795	0.516366	-3186.365659	25784.408967	0.300322	2.150565	0.014615	0.985385	465.473476	-358.894947
502	AUDUSD=X	LinearRegression	4.854795	0.516366	-3186.365659	25784.408967	0.300322	2.150565	0.014615	0.985385	465.473476	-358.894947

Figure 17: Compounded Annual Growth Rate Report [high to low]

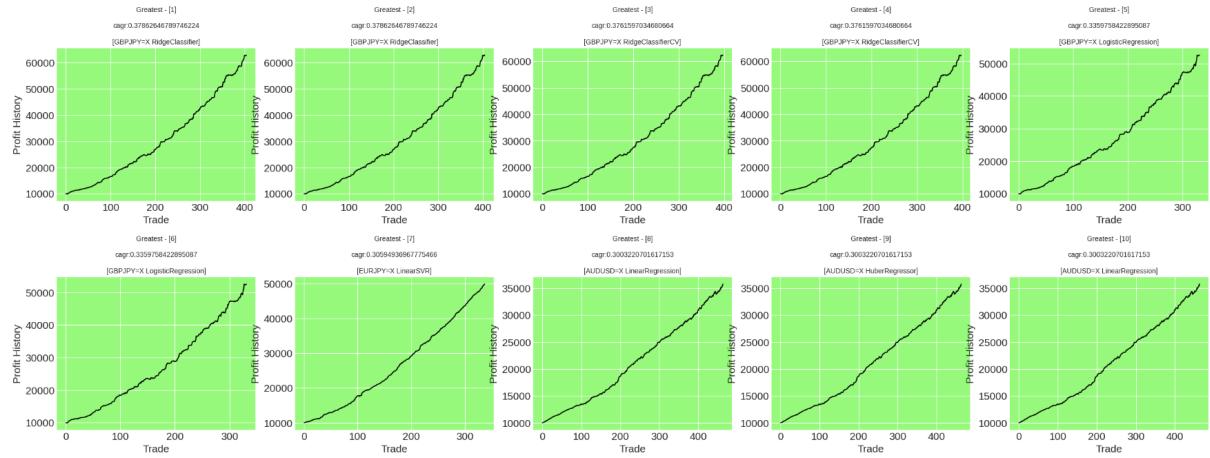


Figure 18: Compounded Annual Growth Rate Graph [high to low]

	asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
656	USDCHF=X	TheilSenRegressor	5.800000	0.354275	636.181269	1377.845525	0.022505	3.000000	0.000000	1.000000	714.220189	0.000000	2
943	EURJPY=X	LinearSVR	6.019178	0.525262	884.481636	39863.330818	0.305949	2.191932	0.010018	0.989982	677.700233	-157.680176	337
434	AUDUSD=X	LinearRegression	4.854795	0.516366	-3186.365659	25784.408967	0.300322	2.150565	0.014615	0.985385	465.473476	-358.894947	463
451	AUDUSD=X	HuberRegressor	4.854795	0.516366	-3186.365659	25784.408967	0.300322	2.150565	0.014615	0.985385	465.473476	-358.894947	463
502	AUDUSD=X	LinearRegression	4.854795	0.516366	-3186.365659	25784.408967	0.300322	2.150565	0.014615	0.985385	465.473476	-358.894947	463
519	AUDUSD=X	HuberRegressor	4.854795	0.516366	-3186.365659	25784.408967	0.300322	2.150565	0.014615	0.985385	465.473476	-358.894947	463
44	USDJPY=X	TheilSenRegressor	7.106849	0.528142	1907.281288	50270.607712	0.287562	2.047036	0.139756	0.860244	632.735962	-214.262718	635
86	USDJPY=X	LogisticRegression	7.106849	0.517733	1907.281288	53664.750221	0.297526	1.989273	0.133610	0.866390	723.748337	-145.139946	624
18	USDJPY=X	LogisticRegression	7.106849	0.517733	1907.281288	53664.750221	0.297526	1.989273	0.133610	0.866390	723.748337	-145.139946	624
939	EURJPY=X	MLPRegressor	6.019178	0.519800	884.481636	18130.391533	0.187474	1.913141	0.002429	0.997571	460.836159	-39.187195	205

Figure 19: Sharpe Ratio Report [high to low]

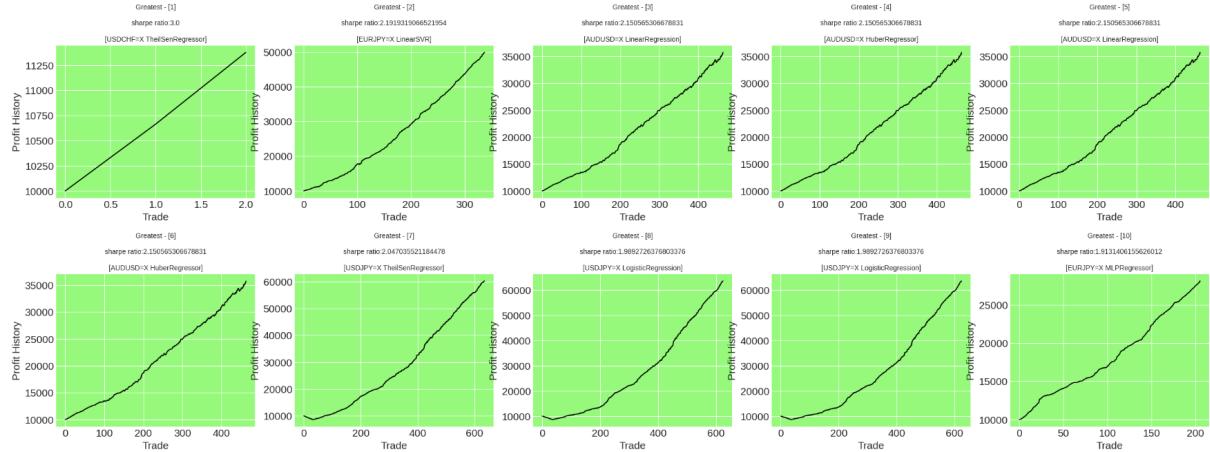


Figure 20: Sharpe Ratio Graph [high to low]

	asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
1006	GBJPY=X	MLPClassifier	5.731507	0.483748	1683.827545	-8529.563568	-0.284284	-0.077276	0.852956	0.147044	30.680008	-199.045013	491
938	EURJPY=X	MLPClassifier	6.019178	0.506145	884.481636	-6017.995468	-0.141851	0.285278	0.602867	0.397133	262.044090	-206.888153	240
556	USDCHEF=X	RandomForestClassifier	5.800000	0.558810	636.181269	-2437.002280	-0.047017	0.695994	0.254931	0.745069	120.420825	-772.486318	323
531	AUDUSD=X	MLPRegressor	4.854795	0.531038	-3186.365659	-2433.085092	-0.055810	0.686158	0.243309	0.756691	92.676243	-201.864238	274
620	USDCHEF=X	ExtraTreesRegressor	5.800000	0.545111	636.181269	-2277.876966	-0.043590	0.714130	0.237798	0.762202	127.414016	-791.276526	380
1018	GBJPY=X	ExtraTreeClassifier	5.731507	0.498566	1683.827545	-1233.692848	-0.022711	0.813681	0.234935	0.765065	284.160294	-597.135040	405
619	USDCHEF=X	ExtraTreesClassifier	5.800000	0.548418	636.181269	-2104.578111	-0.039923	0.726564	0.226194	0.773806	143.046928	-774.252919	365
624	USDCHEF=X	RandomForestClassifier	5.800000	0.554558	636.181269	-2113.766724	-0.040116	0.719226	0.222172	0.777828	121.803528	-781.479922	344
1026	GBJPY=X	BaggingRegressor	5.731507	0.498088	1683.827545	-1563.593002	-0.029230	0.777303	0.210899	0.789101	139.346626	-560.945038	373
552	USDCHEF=X	ExtraTreesRegressor	5.800000	0.545583	636.181269	-1852.734204	-0.034711	0.750652	0.198810	0.801190	142.569928	-816.749037	371

Figure 21: Maximum Drawdown Report [high to low]

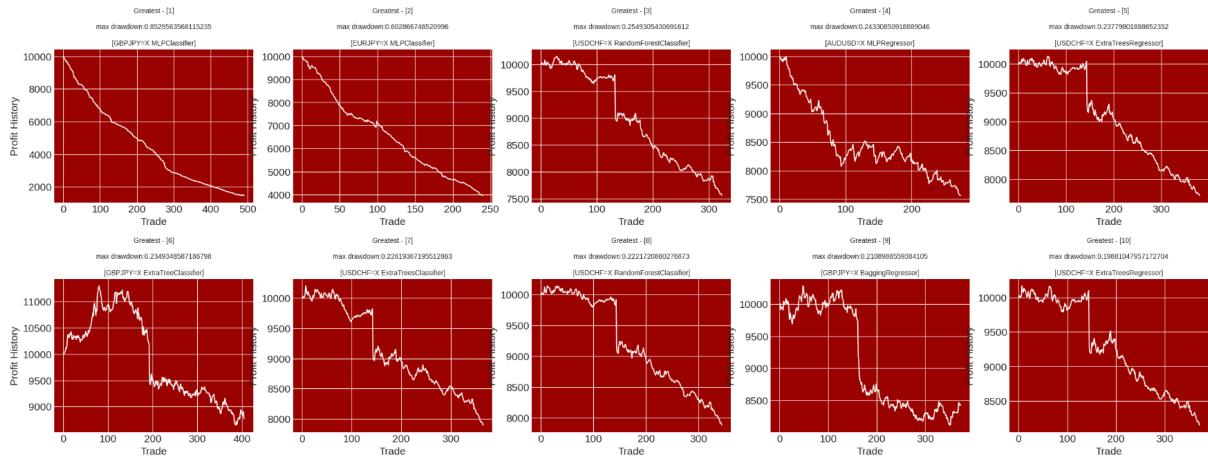


Figure 22: Maximum Drawdown Graph [high to low]

## B - Day Trading Summary

	asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
612	USDCHEF=X	CalibratedClassifierCV	5.800000	0.645725	636.181269	-2.809674	-0.000048	2.996980	0.000611	0.999389	4.681382	-6.115300	3
635	USDCHEF=X	RidgeClassifierCV	5.800000	0.645725	636.181269	-29.829384	-0.000515	2.998994	0.002983	0.997017	4.668783	-28.399367	3
544	USDCHEF=X	CalibratedClassifierCV	5.800000	0.645725	636.181269	-2.809674	-0.000048	2.996980	0.000611	0.999389	4.681382	-6.115300	3
567	USDCHEF=X	RidgeClassifierCV	5.800000	0.645725	636.181269	-29.829384	-0.000515	2.998994	0.002983	0.997017	4.668783	-28.399367	3
604	USDCHEF=X	NuSVR	5.800000	0.645253	636.181269	3.305626	0.000057	2.994650	0.000138	0.999862	4.681382	-1.375756	2
563	USDCHEF=X	LogisticRegressionCV	5.800000	0.645253	636.181269	-28.825868	-0.000498	2.999064	0.002983	0.997017	4.668783	-28.399367	4
631	USDCHEF=X	LogisticRegressionCV	5.800000	0.645253	636.181269	-28.825868	-0.000498	2.999064	0.002983	0.997017	4.668783	-28.399367	4
672	USDCHEF=X	NuSVR	5.800000	0.645253	636.181269	3.305626	0.000057	2.994650	0.000138	0.999862	4.681382	-1.375756	2
667	USDCHEF=X	MLPRegressor	5.800000	0.644308	636.181269	3.112147	0.000054	2.997258	0.000611	0.999389	4.681982	-6.116200	4
596	USDCHEF=X	RadiusNeighborsRegressor	5.800000	0.641946	636.181269	-4.589290	-0.000079	2.998039	0.001106	0.998894	4.682781	-6.404397	15

Figure 23: Accuracy Report [high to low]

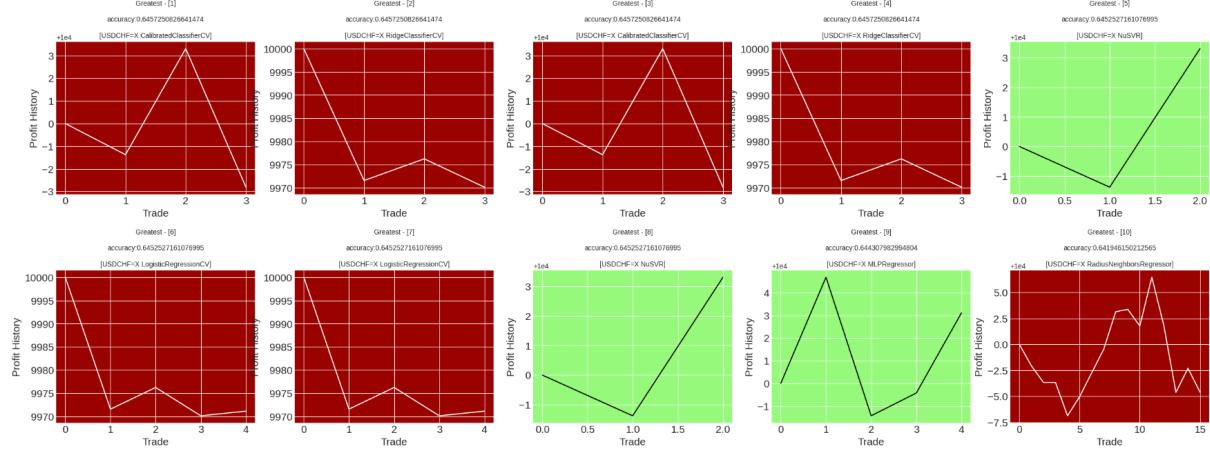


Figure 24: Accuracy Graph [high to low]

asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
1293	NZDUSD=X	DummyClassifier	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1316	NZDUSD=X	SGDClassifier	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1272	NZDUSD=X	BernoulliNB	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1313	NZDUSD=X	Perceptron	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1344	NZDUSD=X	RadiusNeighborsRegressor	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1278	NZDUSD=X	MLPClassifier	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1340	NZDUSD=X	BernoulliNB	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1312	NZDUSD=X	PassiveAggressiveClassifier	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1260	NZDUSD=X	LassoLars	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580
1258	NZDUSD=X	Lasso	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580

Figure 25: Compounded Annual Growth Rate Report [high to low]

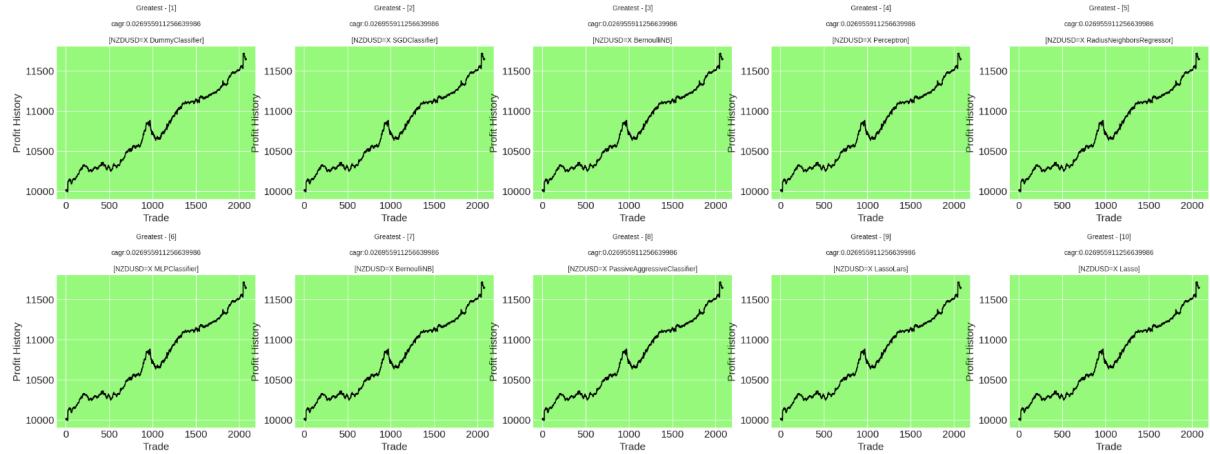


Figure 26: Compounded Annual Growth Rate Report [high to low]

	asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
1245	NZDUSD=X	Perceptron	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1225	NZDUSD=X	DummyClassifier	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1344	NZDUSD=X	RadiusNeighborsRegressor	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1244	NZDUSD=X	PassiveAggressiveClassifier	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1326	NZDUSD=X	Lasso	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1248	NZDUSD=X	SGDClassifier	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1254	NZDUSD=X	ElasticNet	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1258	NZDUSD=X	Lasso	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1260	NZDUSD=X	LassoLars	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083
1340	NZDUSD=X	BernoulliNB	5.712329	0.477218	-1224.100618	1640.929907	0.026956	3.000000	0.022668	0.977332	181.023530	-60.807580	2083

Figure 27: Sharpe Ratio Report [high to low]

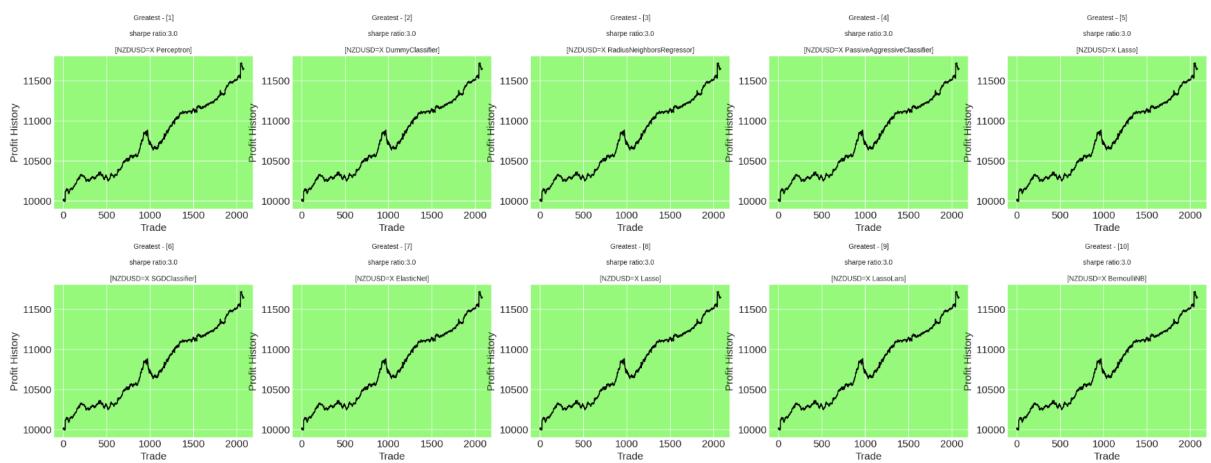


Figure 28: Sharpe Ratio Graph [high to low]

	asset	model	duration	accuracy	profit threshold	net profit	cagr	sharpe ratio	max drawdown	recovery rate	max profit	max loss	total trade
48	USDCAD=X	BernoulliNB	7.106849	0.500771	1907.281288	-5452.733780	-0.104960	2.999848	0.545315	0.454685	51.060028	-458.483276	1278
116	USDCAD=X	BernoulliNB	7.106849	0.500771	1907.281288	-5452.733780	-0.104960	2.999848	0.545315	0.454685	51.060028	-458.483276	1278
109	USDCAD=X	ARDRegression	7.106849	0.482267	1907.281288	-4985.915359	-0.092568	2.999865	0.498830	0.501170	91.159790	-479.056244	1988
41	USDCAD=X	ARDRegression	7.106849	0.482267	1907.281288	-4985.915359	-0.092568	2.999865	0.498830	0.501170	91.159790	-479.056244	1988
44	USDCAD=X	TheilSenRegressor	7.106849	0.528142	1907.281288	-4939.594887	-0.091393	2.999862	0.494338	0.505662	115.752686	-473.178253	927
86	USDCAD=X	LogisticRegression	7.106849	0.517733	1907.281288	-4791.089417	-0.087687	2.999869	0.479751	0.520249	115.752686	-473.178253	1040
18	USDCAD=X	LogisticRegression	7.106849	0.517733	1907.281288	-4791.089417	-0.087687	2.999869	0.479751	0.520249	115.752686	-473.178253	1040
123	USDCAD=X	MLPRegressor	7.106849	0.499614	1907.281288	-4668.786728	-0.084703	2.999853	0.467477	0.532523	92.231323	-341.003540	1891
43	USDCAD=X	HuberRegressor	7.106849	0.535852	1907.281288	-4558.073235	-0.082052	2.999868	0.455807	0.544193	117.236694	-481.995239	397
111	USDCAD=X	HuberRegressor	7.106849	0.535852	1907.281288	-4558.073235	-0.082052	2.999868	0.455807	0.544193	117.236694	-481.995239	397

Figure 29: Maximum Drawdown Report [high to low]

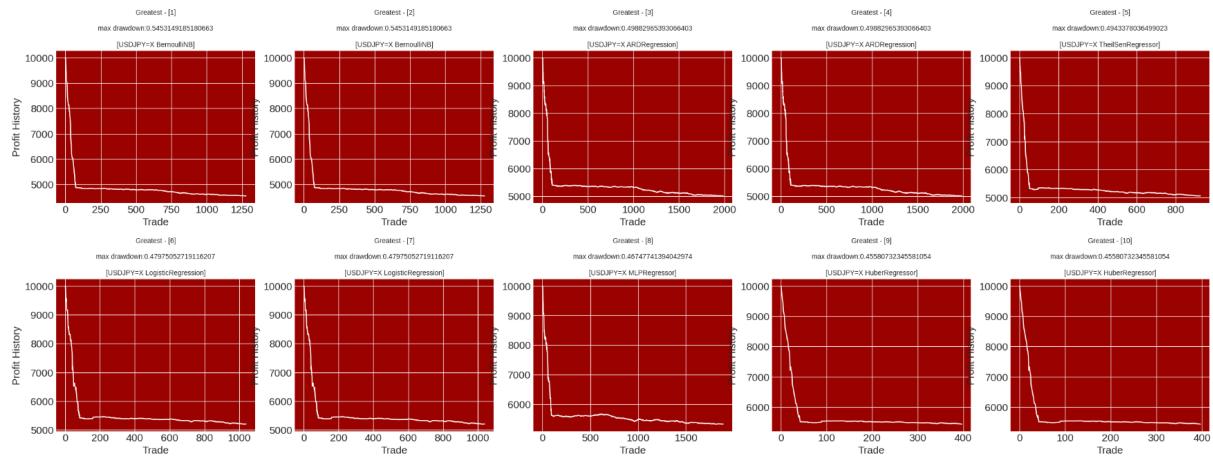


Figure 30: Maximum Drawdown Graph [high to low]