Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Master in Data Science

Master Thesis

Classifying Short-Term Forex Trading Opportunities with Machine Learning

Author: Dakota Mellish

Madrid, June - 2025

This Master Thesis has been deposited in ETSI Informáticos de la Universidad Politécnica de Madrid.

Master Thesis
Master in Data Science

Título: Classifying Short-Term Forex Trading Opportunities with Machine Learning
Junio - 2025

Author: Dakota Mellish
Supervisor: Arminda Moreno-Díaz
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Acknowledgement

A special thank you to Dr. Arminda Moreno-Díaz, whose generous supervision of this project proved to be very helpful and especially reassuring in the most difficult moments of the work. Without her, this project would be nowhere near the state that it is now. Much gratitude is also owed to Dr. Juan Pedro Caraça-Valente Hernández, whose vast knowledge of time series data and clustering methods proved to be quite helpful in the early stages of this project. To all the friends who have I had the privilege of meeting in my time at UPM, you have all helped keep me engaged and stay on top my work. To my family and friends in the United States, your kind words and encouragement have been a tremendous motivation over the past several months.

Summary

In this study, short term price movements in the forex market were predicted using machine learning classifiers and a trading system was developed and tested based off of the model output. Of utmost importance was the availability of a high quality and high volume dataset. Though not free, the Polygon.io forex dataset proved to be a tremendous help to this work and likely the study would have been far more challenging without it. Fourteen years of historical data were selected for the seven major currency pairs related to the U.S. dollar. Virtually no tedious data cleaning and preparation was needed thanks to the high quality nature of the dataset.

The research question of classifying if the market would move upwards, downwards or remain relatively stationary is fundamental and useful to an everyday trader. Though some may consider it a step down from having a model which can accurately predict the precise value of price, the reality is that exact price movements are quite difficult to forecast, as the market may oscillate and behave almost randomly at times. Nevertheless, the researcher desired to extract order from seemingly disordered sequences of events. To produce such an order, it was necessary to derive labels from unlabeled data. This was one of the most important steps of the study, as there wasn't a clear roadmap or methodology to prescribe to the problem. Settling on a future window approach seemed to be the most intuitive and sensible approach, and given the way price data conveniently provides the maximum and minimum values of each hour in the form of high and close, it made it easy to compare how much the market moved up versus down. Furthermore, the fact that the market moved slightly more up than down doesnt́ imply that the market truly moved upward. In this sense, it also made sense to require a minimum ratio between the upwards and downwards. A minimum ratio of 2.5 between the maximum and minimum of upward and downward movement was selected, as it aligns well with commonly used risk reward ratios in trading. To account for situations with very large ratios but very small actual movements, only changes in price above the long term 50th percentile were labeled a buy or sell. If these conditions were not met, a hold label was assigned,. With regards to the future value window, twenty-five hours was selected as the amount of time to consider. Twenty-five hours allows for a trend to form and develop, and it allows for a quick turnaround period from a trading standpoint, as there are fees associated with holding a position for a long time such as days or weeks. With the label created, it is now important to consider how the dependent variable might be classified.

In this work, it was desired to explore a wide variety of independent variables from a technical analysis perspective, meaning other variables such as macroeconomic variables as well as sentiment analysis would be excluded. There are many ways to

study and capture behavior and phenomena that occur in the movement of currency prices and the researcher wanted to take a naive approach. This meant making as few of the prior assumptions as possible and testing which variables may be the most useful in predicting short-term future price movement. Therefore, a wide variety of technical indicators and chart patterns were developed and explored. Furthermore, the idea of time series clustering has also been a key interest to the researcher, and it seemed like a plausible and original addition to forex price classification to incorporate a variable derived from time series clustering, as many models in this space lack a clustering component to them. The main idea is to use the general shape of the price time series as a mechanism for understanding future price movements. Using the k-means dynamic time warping algorithm proved to be a challenge, as many steps were required to transform the data into usable series for the algorithm to function. Many assumptions and small design decisions had to be undertaken, such as the strategy to normalize the data, as well as determine the sequence. Furthermore, different custom variables such as a difference in moving averages were able to be feature engineered using the base dataset.

Beyond just the variables themselves, there is the question of being able to classify each value as a buy, sell, or hold recommendation. The exact method of classifying these three categories was of lesser importance to the study, though three very reputable classification models have been selected in Logistic Regression, Random Forest and Extreme Gradient Boosting. The models went through a standard process of variable subset selection, as well as a hyperparameter grid search to ensure a well-generalized model fit could be obtained. The results obtained were above the expectations of the researcher, with 82.6% class weighted accuracy achieved.

With a reasonably accurate classifier in hand, it was then desired to consider how this classifier could be applied to unseen data but also in a way that was practical for a forex trader. In this sense, it was desired to construct a simple yet rational framework for making trades independent of human judgement. A simple trading system was devised and designed to operate for a single currency at a time. It requires two consecutive buy or sell signals so as to avoid false breakouts or erratic movements in the market, and if this condition is met, a buy or sell signal is then placed, provided that the system is not actively engaged in another trade for the same currency. Risk limits were explored to find the ideal amount of percentage in points (pips) to target in a consistent manner. In the end, the trading system proved to be profitable. The study is considered to be a success, though there are certainly areas of improvement. The clustering work proved to add value (as the mutual information score for the variable was close to .05), though more work and testing could be done to make the clusters more similar, as they all had a low silhouette score. With such a large dataset, it was difficult to do much experimentation with the clusters, as each fit required several hours to train. With respect to the classifier, a leave one out fold cross validation strategy was also desired, as it would be interesting to simulate the classifier and trading system performance across each year and not just 2024, though the computing resources needed to achieve this exceeded that of the researcher's own personal laptop. However, much was learned along the way and the study was a success.

Abstract

In this study, it was desired to investigate if short term movements in the forex market can be classified with success. Using hourly price data from 2010-2024 (with 2024 as the validation set), data labels were constructed for each hour, and technical indicators along with chart patterns and custom variables were constructed and used as independent variables for the classification problem. Time series clustering using k-means and dynamic time warping was also utilized as a feature to characterize the shape of the data to understand its future short-term price movement behavior. Several classifiers were tested, with XGBoost yielding the best results. It obtained a class-weighted accuracy of 82.6% and F1 score of .826. Using this classifier as a way to make simulated trading decisions, a system was constructed and generated 11% annualized returns or higher using the validation dataset, with a maximum portfolio drawdown of 10%.

Keywords: Forex price classification, financial time series, K-means time series, dynamic time warping distance, trading system, supervised multi classification, logistic regression, random forest, XGBoost classifier, variable subset selection, hourly price data, U.S. major currency pairs, technical analysis, chart patterns, moving averages.

Contents

1 Introduction and Objectives **1**
1.1 Context and Motivation 1
1.2 Objectives 2
1.3 Methodology 2
1.4 Structure of the project 3

2 Brief Literature Review **5**

3 Problem Description and Preprocessing **9**
3.1 Dataset Overview 9
3.2 Creation of Data Labels 10
3.3 Variables to be Used 11
3.4 Series Clustering 13
3.5 Scaling and Additional Considerations 17
3.5.1 Scaling for Non-Binary Features 17
3.5.2 Vectorization of Target Output Class 18
3.5.3 Class Balancing Strategy 18
3.6 Classifiers to be Used 19
3.7 Trading System 22

4 Results **27**

5 Discussion and Future Work **33**

References **37**

Appendix **41**

Chapter 1

Introduction and Objectives

1.1 Context and Motivation

The foreign exchange (FOREX) currency market is the largest financial market in the world. It comprises trillions of U.S. dollars worth of daily transactions [3] conducted on the over the counter market. It is a market which is open 24 hours, 5 days a week, beginning with the opening of the Australian stock exchange and ending the week with the closing of the New York stock exchange. Who are the major players in the Forex market? The lion's share of who controls the market belongs to institutional investors, such as investment firms, hedge funds, and banks. Central banks also make up a large share of the volume, near 20%. Lastly, retail investors who speculate on currencies make up the smallest share at around 5% [9]. Annual returns on trading currencies can reach 10-20% for large hedge funds, with some approaching 40-50% in exceptionally good years [11]. With enough starting capital, these returns can fund a living without the need for a typical job.

How are such profits achieved? Since the arrival of the 21st century, algorithmic trading has quickly emerged as an important tool for large and small time investors alike. In fact, as of 2013, as much as 70% of trading in the U.S. stock market was handled by algorithms, and for the Forex market, 92% traded algorithmically [14]. It is quite a sobering fact that most of the decisions are not truly being executed by humans but by a machine. This is due to the impartial reasoning that computers can achieve. They can simultaneously consider the past data occurring recently and long ago without the slightest trace of human emotion. They can also operate without human supervision, as the arrival of the internet can allow for an end to end pipeline. Algorithms can be quite simple in nature. In fact, a successful trading system could be achieved simply by considering a single indicator and executing buy/sell orders if a certain threshold is reached. Of course, the fundamental question in forex trading is always a matter of, "What is the best decision to make?" There are only three options, to buy, sell or do nothing (hold). While many researchers and industry professionals alike have attempted to build robust forecasts for future price movements in Forex, a simpler question presents itself, "Which direction will the market move and what is the best course of action?" This can be simplified into a 3 class classification problem of "Buy", "Hold" and "Sell" and will form the basis of this study.

1.2 Objectives

This paper seeks to gain insight on several important research questions:

1. Can machine learning classification techniques accurately classify the future short-term movements of the market in terms of buy, hold and sell, and if so, how well?
2. Can technical indicators, chart pattern techniques, as well as time series clustering methods be useful variables for classification, and if so, which are the most useful?
3. With objectives 1 and 2 provided, can a profitable trading system be implemented and generate profits over a longer time frame?

Each objective will be explored and tested by the researcher. It is worth mentioning that the nature of this study will more closely resemble that of a retail trader (single individual), as opposed to an institutional investor.

1.3 Methodology

The methodology for the project can be principally thought of in four distinct stages (occurring in chronological order by necessity):

- Data Labeling
- Feature Engineering
- Supervised Learning Multi classification
- Custom Model Evaluation with Trading Simulation

For each step, all code was written using Python and the dataset was managed with the Pandas and Numpy packages. The hardware used was an HP Spectre possessing the following CPU configuration: 12th Gen Intel(R) Core(TM) i7-1255U, 1700 MHz, 10 Core(s), 12 Logical Processors

For data labeling, the methods used were devised entirely through the researcher's experimentation and intuition. A three category classification system was desired, with each row corresponding exactly to a "Buy", "Hold" or "Sell" recommendation. Custom code was created in Python to help provide these labels. The exact process will be further discussed in Section 3.2.

For feature engineering, the base dataset was significantly augmented through the use of an external library called TA-lib, as well as custom Python code which was easy to produce, alongside a time series clustering addition which was made possible through the open source library Tslearn. The indicator and chart pattern functions of Ta-lib were called and joined to the base dataset in Python. The custom variables such as moving averages were derived through standard Pandas dataframe manipulation. The time series clustering approach was realized through Time Series k-means in conjunction with the dynamic time warping distance metric. The short term future behavior occurring for the clusters were studied and it was possible to create a single variable indicating the general direction of the future. This methodology was

made possible through an open source Python library Tslearn. The details and more explanation are provided in Section 3.4

For multi classification, three well-known models were tested in Logistic Regression (with a multiclass setting enabled), Random Forest, and XGBoost classifier. The Sklearn and XGBoost libraries helped make the implementation of the models quite easy. Standard classification metrics were also sourced from Sklearn. Variable subset selection was also made possible through Sklearn, and a univariate filter, as well as custom univariate filter and wrapper method combination were tested against all variables. A hyperparameter grid search using custom built code was also constructed in Python to better tune each of the models. Visualization of certain results were then shown using the python library Matplotlib.

Finally, custom backtesting was implemented in python using the researcher's own code that resembled a simulated trading environment. It involves standard iteration through the validation dataset using all of the above steps and requires a set of rules to carry out simulated trades effectively. Visualization of the results were also realized through Matplotlib. More detail is provided in Section 3.7.

1.4 Structure of the project

In Chapter 2, recent studies involving forex price movement classification will be analyzed and discussed. Attention will also be given to forex time series clustering. Chapter 3 will focus on the process and steps involved in the feature engineering, supervised multi classification as well as the trading system design for this project. Finally, a practical interpretation of the results along with the results of the trading system will be considered in Chapter 4. Future enhancements that could better improve the results of the study will also be considered in Chapter 5. Certain results and details surrounding the variables used in the study which were too long to place in the main part of the document can be found in the Appendix.

Chapter 2

Brief Literature Review

One of the challenges associated with this problem is that many of the best examples in existence are hidden from the research and scientific community, as most models for forex price movements are built by those working in industry. Were the exact algorithms and techniques used by the most successful to be exposed, it could hurt the profitability of the firm.

The focus of the literature review for this paper will center around common modeling approaches towards forex price movement classification and not regression. In addition, forex time series clustering approaches used previously will also be consulted. The sources for the literature review are threefold:

1. Google Scholar
2. Consensus AI
3. Papers discovered as references for examined work

A strong preference was given to research which has been published within the past ten years. In addition, the review was tailored towards studies using forex datasets as opposed to the general stock market or other financial products such as cryptocurrency.

As is the case with machine learning, the models used in forex price movement classification are quite varied. Pande, Divayana and Indrawan [23] demonstrated in 2020 that the k nearest neighbors as well as the Naïve Bayes algorithm can classify the British Pound and Japanese Yen daily price movements with accuracy at or above 50%. They relied on only price data (absent of indicators) and compared recent movements of the previous day to estimate what would occur in the following. Other papers have used support vector machines as a tool for robust classification. Oetama, Heryadi, Lukas and Suparta [21] were able to obtain classification accuracy of nearly 95% for next day price movements of the Euro Dollar daily pair by comparing movement of price data such as open, high, close and volume for previous periods. They relied on both the linear and radial basis function kernel. Using a multi-class Support Vector Machine, Sadeghi, Daneshvar and Zaj [31] were able to achieve accuracy of 77.3%, precision of 81.2% and recall of 72% for daily Euro Dollar price data spanning from 2014-2019. They relied on moving averages, as well as technical indicators such as relative strength index, moving average divergence/converge and stochastic oscillators. An important distinction to their work is their use of “fuzzy labels”, which

produce a sliding scale of membership for price movements beyond the traditional three-class designation. This underscores the importance of data labeling in forex price classification, as it is often quite difficult to produce reliable labels. An example of their approach is shown below in Figure 2.1

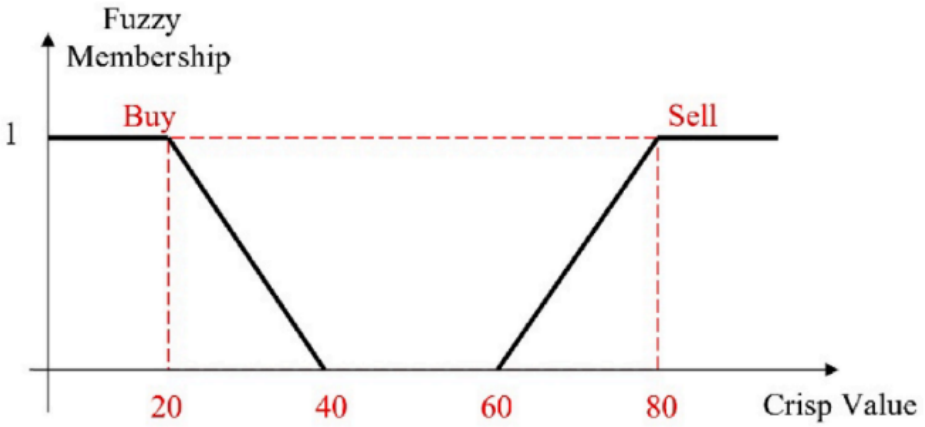

Fig. 4. Fuzzy membership functions of RSI for Buy and Sell rules.

7

Figure 2.1: Fuzzy Membership

Ensemble methods such as Random Forest can also be employed. Santuci, Sbruzzi, Araújo-Filho and Lele [32] used Random Forest along with SVM to support a buy and sell strategy and compared them to strategies used with traditional technical indicators such as relative strength index. They used approximately two years worth of data across four different major currency pairs, and found annualized returns ranging from 5-15% for SVM and Random Forest. This tended to outperform strategies which only employed technical indicators.

Another promising recent work came through the use of a Hidden Markov Model and stacked generalization models. Fereydooni and Mahootchi [19] used 2006-2020 daily forex price data using four Euro currency pairs and constructed buy/sell signals. The researchers selected a combination of Linear Regression, XGBoost, Random Forest, Light Gradient Boosting, Ada Boost, Support Vector Machine and a Multilayer Perceptron as the building blocks of their stacked models. The researchers used the Hidden Markov Model to separate the stacking classifiers into bullish and bearish signals. They used technical indicators, macroeconomic data such as US stock market indices, as well as lagged pricing data. They used principal component analysis to help select the most important features from this collection of data. Though the researchers did not disclose their models' accuracy, they claim to generate between 10-30% annualized returns using their trading system, depending on which voting system was used to reconcile the different recommendations.

Of course, it should come as no shock that many researchers have channeled their efforts in using deep learning for forex price classification, given the unprecedented expansion of deep learning techniques in industry and research. Gong, Ventre, and O'Hara [12] used minute data to predict the next 5, 10 and 15 minute windows with the well known Inception convolutional neural network and obtained between 44-47% classification accuracy. Fisichella and Garolla [10] used a ResNet50 and Vision

Transformer and provided candlestick chart images to use as training data for the networks. From 2010-2021, they used 9 years of training data examples and three years of data as their validation set. They tested their model using 1 hour, 4 hour, 12 hour and 1 day price data. The 4 hour window proved to be the only consistently profitable time period in which to base their dataset. Moghaddam and Momtazi [20] explored a similar minute data approach through the creation of a hybrid model with a candlestick image classification and time series LSTM neural network . Using just the Euro Dollar pair, they were able to obtain between 62-63% classification accuracy using a three-tiered classification system. Kozák, and Juszczuk [18] experimented with a five class system (differentiating between a weak and strong buy/sell) for seven currency pairs and experimented with the daily and four hour period for their price data. They relied heavily on the commodity channel index for the model decision making process. A decision tree was used as the classifier. However, they obtained relatively poor accuracy, ranging from 25-40% depending on which time period was being utilized.

With respect to time series clustering in the Forex market, it was difficult to locate relevant examples that align with the proposed work of this study. Though more symbolic in nature than pure time series clustering, Ow, Ngo and Lee [22] used 30 minute price data for the AUDUSD and EURUSD currencies for the year of 2012. They constructed different series using a set of custom features that were derived from the price data, and used the k means dynamic time warping algorithm to identify similar series. This was then used for predicting an uptrend or downtrend and achieved 100% accuracy using the first three months of 2013. Without knowing which exact features were used to construct the symbolic series, it is difficult to draw comparisons. Though not in the same domain, Kim, Lee, Ko, Jeong, Byun and Oh [17] were able to devise a trading system in the future commodities market using the k-means dynamic time series warping algorithm. In addition, Tsinaslanidis [36] also demonstrated the ability to use dynamic time warping to predict buy and sell signals in the U.S. stock market. Similar applications have been achieved in the financial time series space as well by D'Urso, De Giovanni and Massari [7] and Tsinaslanidis, Alexandridis, Zapranis, and Livanis [37].

In short, it is clear from the literature review that there are no consensus results on what the expected accuracy of a classifier should be for forex price movements, and it is difficult to compare prior research to the current work due to the difference in variables and timeframes utilized. It was also difficult to find a study which provided an exhaustive set of technical indicators and/or chart patterns, which is also a unique part of this study. It is clear that there hasn't been a study published in the forex space which tries to combine elements of clustering techniques alongside classification. This serves as a good motivation for attempting this technique, as it can be a novel addition.

Chapter 3

Problem Description and Preprocessing

3.1 Dataset Overview

Due to the complexity and vast nature of forex data, it is worth establishing a set of parameters for this study that will be referenced to later. First of all, which currency pairs shall be considered? In the Forex market, some of the most liquid and least volatile currency pairs can be found in the U.S. dollar major pairs [6]. These currencies will form the pairs to be considered shown in Table 3.1

Pair	**Name**
EURUSD	Euro U.S. Dollar
USDCHF	U.S. Dollar Swiss Franc
USDJPY	U.S. Dollar Japanese Yen
AUDUSD	Australian Dollar U.S. Dollar
CADUSD	Canadian Dollar U.S. Dollar
NZDUSD	New Zealan Dollar U.S. Dollar
GBPUSD	British Pound U.S. Dollar

Table 3.1: Major currency pairs

Another important question to consider is, "For what time frame will the data be measured?" Forex data comes in a wide variety of time frames. Some traders utilize daily or even weekly aggregations of price data, while traders who operate within a very short time frame will use minute aggregations of price data. With a desire to explore the possibilities of short term price movements yet maintain broader knowledge of what is going on, the hourly window of price data will be used.

The dataset is sourced from Polygon.io [30]. It was purchased and obtained by the researcher for $50 using a monthly subscription. It spans from January 1, 2010 until December 31, 2024. As mentioned, the data is aggregated at the hourly level. It contains the following variables in Table 3.2.

Variable	Description	Type
Time	Observation time of the price in Unix milliseconds	Integer
Volume	Total transaction volume for the given currency pair	Float
Opening	Opening price of the given currency	Float
Closing	Closing price of the given currency	Float
High	Maximum price of the given currency	Float
Low	Minimum price of the given currency	Float
Pair	Currency pair (e.g., EURUSD)	Categorical

Table 3.2: Polygon.io Dataset Description

3.2 Creation of Data Labels

One of the unique challenges with classifying price data (of any asset class) is that there are no inherent labels included in the dataset. However, with reasonable effort, each row can be assigned one. Future values of price can be considered using a sliding window approach, and the maximum value (price high) as well as minimum value (price low) can be evaluated and compared against one another. What time period will be considered for the future value window? The short answer is that multiple windows can be considered and evaluated, however it is helpful to just pick one. Twenty-five future trading periods will be used to assign labels to the price data. The columns **up_pct_change_25** and **down_pct_change_25** will be constructed and added to the original dataset. They will compare the maximum price high and minimum price low of **t+1** until **t+25** against the opening price of the price at time **t**.

Of course, there will be times in the market where there is relatively little change. It may be useful to require a minimum percent change threshold for buy and sell labels. An easy method to determine the proper thresholds for establishing labels can be achieved by analyzing the distribution of the dataset's future window columns **up_pct_change_25** and **down_pct_change_25**. The validation portion of the dataset will not be included. A simple set of percentiles have been selected to consider in Figure 3.1

Metric	25th Percentile	Median	75th Percentile
up_pct_change_25	0.19%	0.34%	0.60%
down_pct_change_25	0.20%	0.37%	0.64%

Figure 3.1: Percentile Distribution of Future Value Window

Of course, it should come as no surprise that the larger the future window, the larger the up and down percentage change values become. A sensible approach can be to rely on the median as an initial filter for the creation of the data labels. This will ensure that on a percentage basis, a buy/sell opportunity is at least or greater than the 50th percentile of its distribution.

Beyond just the total value, the difference between **up_pct_change** and **down_pct_change** should also be considered. A ratio can be calculated between the two values

and a filter can be applied for a certain ratio. A similar though not exact comparison can be found in the risk ratio, a common measure used by forex traders. It measures the amount willing to be lost versus the amount desired to win. In order to properly manage risk, a return is often desired to be greater than the underlying amount to be risked. As such, many traders often pursue a risk reward ratio of 1:2 or even 1:3 [13]. An implicit risk reward ratio can be calculated with **up_pct_change** against **down_pct_change**. If the ratio between the two exceeds 2.5, the row can be considered a buy/sell opportunity. If this condition is not met, a hold label will be designated. The underlying logic is shown below in Figure 3.2.

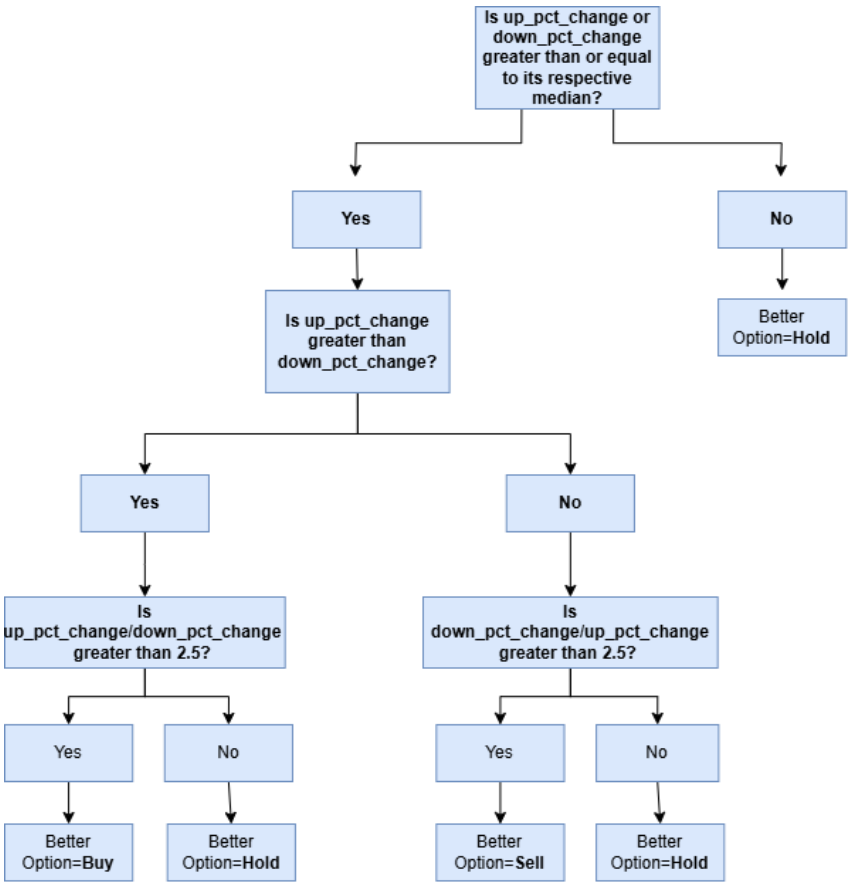

Figure 3.2: Decision Tree for Data Labeling

The strategy is intended to provide labels which are *well-separated* from one another so the model can differentiate between the three market states. Furthermore, the buy and sell labels have been constructed such that the move in the *opposite* direction of the intended label will be at least 2.5 times less than the desired movement. The final column will be named **better_option.**

3.3 Variables to be Used

Technical indicators are a common tool used by algorithmic trading platforms and human traders alike for all asset classes. They attempt to describe various aspects of asset price and price action. They are entirely endogenous to price and do not rely

on news outlets nor economic indicators such as gross domestic product, inflation and unemployment rate. Though sometimes regarded as biased and unuseful due to indifference towards the "bigger picture", they can often be useful in detection of new trends or reversals [16]. Calculation of the most widely used indicators, though simple in nature, can be a tedious process. Fortunately, an open source python library TALib [34] exists and can be used to create a wide variety of technical indicators as well as detect chart patterns. The default parameter values will be utilized. In addition, some new variables can be derived with relative ease. The full list has been included in Table 2 for technical indicators and Table 3 for chart patterns in the Appendix section. Examples of technical indicators include Relative Strength Index, Commodity Channel Index, Rate of Change, Money Flow Index, Moving Average Convergence Divergence and much more. For all the chart patterns, the columns **open**, **close**, **high** and **low** are used. In total, 59 patterns are available for easy use by the classifier model. For the sake of brevity, the full list of patterns will be included in the appendix. Some relevant examples of chart patterns that will be included are Two Crows, Three Black Crows, and Three Inside Up/Down patterns. Examples have been provided below and their sources have been included [38],[2],[1].

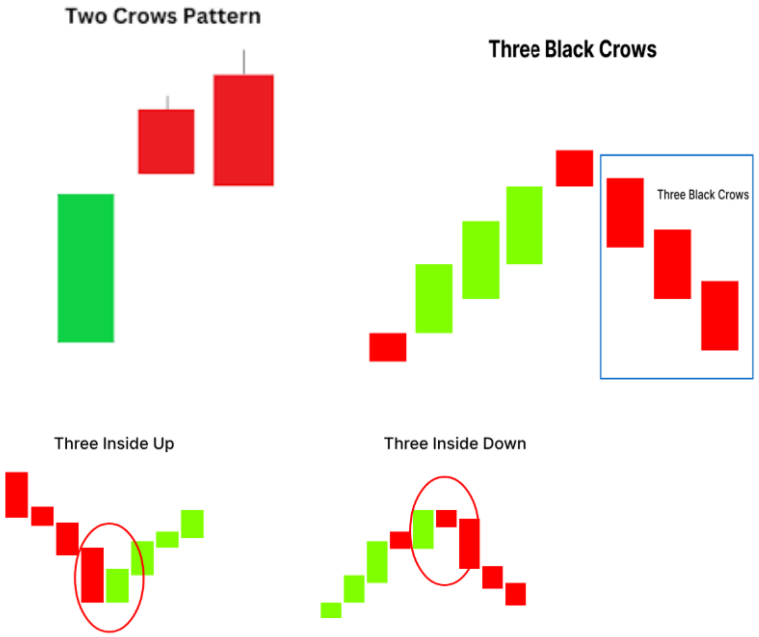

Figure 3.3: Chart Pattern Examples

For each pattern shown above, they signal a potential reversal of price. Though there is certainly no guarantee of a reversal, the chart patterns could prove to be of use for the classifier to interpret a potential price shift.

For the chart patterns, a bullish and bearish signal is provided numerically after the pattern has been detected. However, the signal is not persistent and does not provide any sort of short term memory. As such, as opposed to just using the pure signal, a moving average of twenty-five periods will be used for the chart patterns so that recent occurrences will be considered. Furthermore, some custom variables can be designed pertaining to moving averages of price, time of day, and the given currency pair. They are enumerated below in Table 3.3.

Variable Name	Description
dayofweek	Derived from **t** timestamp and contains value from 0-6
hour (UTC)	Derived from **t** timestamp and contains values from 0-23.
is_usd_jpy	Binary variable flag if pair is USDJPY
is_eur_usd	Binary variable flag if pair is EURUSD
is_gbp_usd	Binary variable flag if pair is GBPUSD
is_chf_usd	Binary variable flag if pair is CHFUSD
is_aud_usd	Binary variable flag if pair is AUDUSD
is_nzd_usd	Binary variable flag if pair is NZDUSD
Price - EMA(20)	Close price against its 20 period EMA
EMA(200) - EMA(50)	Designed for capturing longer term behavior.

Table 3.3: Custom Variables to be Used

3.4 Series Clustering

As mentioned previously, there appeared to be a dearth of research devoted to using time series clustering for classification purposes in the forex space. An idea that had come to mind was to leverage a clustering technique to identify similar series forex currency price data using Time Series K Means and the dynamic time warping metric from the Tslearn python library [35]. The idea would be to group similar series based on close, high low and volume over a certain amount of observations. With this information, the long-term results of similar series can be understood and this may provide a useful feature for the classification model. While implementation isn't particularly difficult, there are a number of steps that must taken in order to undertake clustering.

The price data must be normalized for each currency, as the price scale can vary substantially (particularly for the USDJPY pair). A simple approach is to scale each pair independently between [0,1]. However, it is worth considering that this would imply the loss of local minima and maxima, as each series would just have a single point at 1 and and a single point at 0. Instead, a sliding window approach can be utilized. A sensible amount might be 1000 trading periods, this corresponds to roughly 8.5 weeks of price data (with 24 hours x 5 trading days=120 trading periods), allowing for plenty of distinct windows while not giving too much significance to a single day or week of data. Observations can be grouped into distinct buckets based on 1000 trading periods, and [0-1] min max scaling can be calculated for each distinct period.

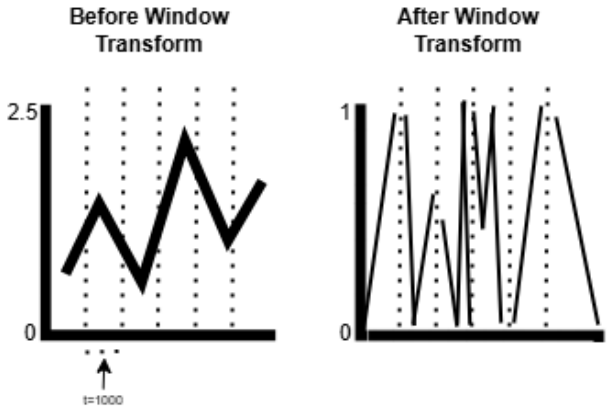

Figure 3.4: Concept of Window Transformation using 1000 trading periods

As shown above, the minimum and maximum are now 0 and 1 respectively, though each distinct window contains a min and max as opposed to a global transform. For each currency pair, the data is sorted and then split into distinct windows of 1000 observations. The minimum and maximum are calculated and the scaling is applied to the close price and volume, respectively..

How can the low and high prices be constructed off of the transformed close price so that they track close price? Well, if the ratio of low/close and high/close are saved at each point **low_close_ratio** and **high_close_ratio**, they can later be used as multipliers for the normalized close price to derive **low_norm** and **high_norm**. This way they are guaranteed to follow the close price as opposed to following their own distinct trajectories and making the clustering fit process more challenging. Example calculations are shown below to better illustrate the idea in Table 3.4.

Variable	**Value**	**Calculation**
close	1.29	
low	1.287	
high	1.293	
volume	272	
low_close_ratio	0.997	1.287/1.29
high_close_ratio	1.002	1.293/1.29
close_norm	.652	Derived from min-max scaling
low_norm	.650	.997*.652
high_norm	.653	1.002*.652
volume_norm	.25	Derived from min-max scaling

Table 3.4: Example Calculations

What else must be done to prepare the data for k-means clustering? The goal is to locate series that behave similarly. The data has been normalized which is a good start, however it may be necessary to cut the data into smaller pieces. It is also important to remember the objective of the study, which is to identify *short-term* opportunities in the forex market. Therefore, a fifty period window of t_{-49}until t_0 shall be constructed for all observations except for the first 49 observations. This fifty period window can include the **close_norm**, **low_norm**, **high_norm** prices. Each

column can be suffixed according to its relative position. This leads to a set of columns **close_norm_0**, **low_norm_0**, **high_norm_0, volume_norm_0** all the way until **close_norm_49**, **low_norm_49**, **high_norm_49, volume_49**. However, in this case, the 0 index corresponds to t_{-49} , and the 49 index corresponds to $t_{0.}$ In the interest of keeping the validation untouched from the clustering process, the cluster will only be trained on data from 2010-2023 (the training data).

Beyond simply constructing a past series for each observation, it is worth recording *future activity* to later gain understanding of how each cluster behaves. Therefore, the minimum and maximum values of the next twenty-five periods will also be recorded, as well as the relative position that each is recorded. Twenty five is utilized in order to maintain consistency with the target output variable **better_option**. These variables will be called **max_future_value**, **min_future_value**, **idx_max_future_value**, and **idx_min_future_value**. In order to provide a proper comparison of what happens with regard to the latest **close** price, an adjusted price will also be saved called **cur_price**. The K-means time series model can now be utilized for the variables consisting of **close_norm_0**, **low_norm_0**, **high_norm_0, volume_norm_0 . . . close_norm_49**, **low_norm_49**, **high_norm_49, volume_norm_49**. The dynamic time warping distance will be selected as the distance metric of choice. Values of 20, 30, 40, 50 and 60 are to be tested for the primary hyperparameter k. It is worth mentioning that by default, all values from the dataset can be used. However, it is more useful to try a sliding window approach of 25 units in order to minimize overlap. This will mean that the dataset to be used in training the clustering algorithm will be 96% more compact (24 out of every 25 observations will be discarded). The clustering model will then predict the entire dataset, including the validation set.

With the clusters provided, the long term **max_future_value** and **min_future_value** are compared with **cur_price** to derive the columns **cluster_up_pct_change** and **cluster_down_pct_change.** The columns can be averaged using **pair** and **cluster** as the columns to aggregate by. The difference can be calculated between the two columns to measure the overall long term direction the clusters favor, if any. This creates the column **cluster_up_minus_down_pct_change**. However, due to multiple cluster amounts being tested, the variable will be included for each value of k. This will translate to five variables **cluster_up_minus_down_pct_change_20**, **cluster_up_minus_down_pct_change_30**, **cluster_up_minus_down_pct_change_40**, **cluster_up_minus_down_pct_change_50** and **cluster_up_minus_down_pct_change _60**. Each cluster model will be evaluated using the silhouette score method, and the variable belonging to the cluster model with the best fit will be used.

The metric to evaluate the quality of the cluster fits will be the silhouette score in conjunction with the dynamic time warping distance metric. The silhouette score function from Sklearn [33] will be utilized. For each cluster of each model, the average silhouette score will be tabulated, and then the average of all average silhouette scores for the model will be compared. The results for each value of k are shown below in Table 3.5.

K	Average Silhouette Score
20	.103
30	.112
40	.109
50	.101
60	.092

Table 3.5: Average Silhouette Score per Cluster Fit

The highest value is obtained at k=30, however, the silhouette scores appear to indicate there is weak cohesion within the clusters across all levels of k. Nevertheless, this alone does not make the clusters useless for classification purposes. It remains to be seen if the cluster column **up_minus_down_pct_change_30** can be a strong variable in classifying forex price behavior. This variable will be selected and tested later on. With this step complete, it is now time to remark on the long term behavior of the 30 clusters. The average of the maximum high price and the minimum low price obtained in the future twenty five trading periods is calculated for each cluster. This provides a long term expected value of sorts for each cluster. They are provided in Table 1 of the appendix. As shown in Table 1, many of the clusters do not appear to exhibit strong differences in **cluster_up_pct_change** versus **cluster_down_pct_change**. Just five clusters show differences that exceed .2% in absolute value. However, when analyzed at an individual cluster-pair level, the clusters may prove to be of more use. Figure 3.5 shows the values of **cluster_up_minus_down_pct_change_30** per cluster-pair combination.

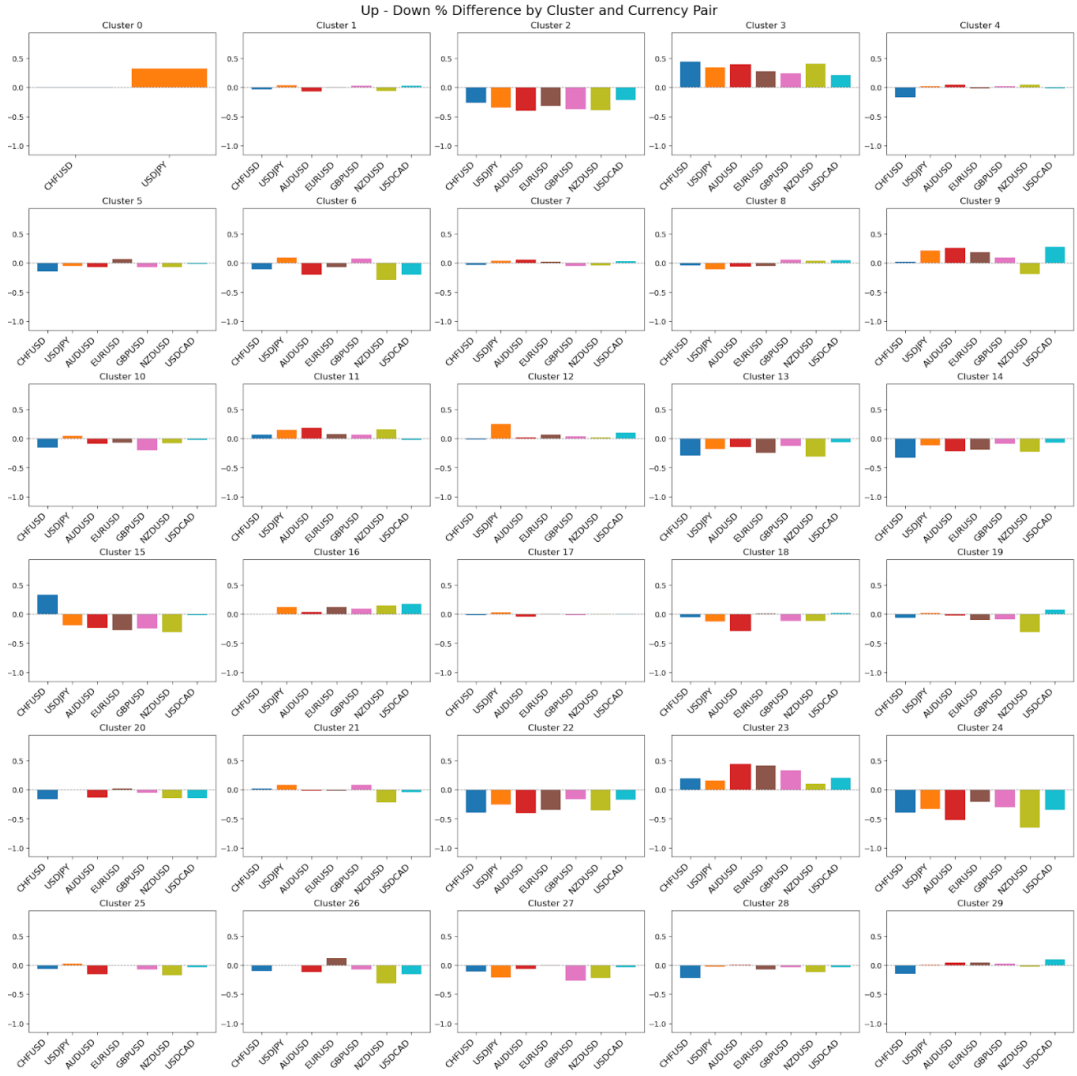

Figure 3.5: Cluster Behavior by Currency Pair

As shown, per cluster the pairs tend to behave similarly, however in certain cases such as a Clusters 9 and 15, at least one of the pairs moves in a different direction than its peers. This may be useful context for the classifier to understand beyond just being given the aggregate long term percentage change per pair. The classifier will be supplied the variable **cluster_up_minus_down_pct_change_30**.

3.5 Scaling and Additional Considerations

3.5.1 Scaling for Non-Binary Features

Given that the dataset and its variables are temporal, special consideration must be given to it beyond a typical dataset. An important problem to be solved is the scaling of the dataset so the classifier does not mistakenly place greater importance to larger values. A common approach is to use a standard scaling technique where the mean is set to 0 with a standard deviation of 1. A similar recent Forex modelling research paper in New Zealand also made use of this technique [8] for quantitative variables.

All columns which are not binary (0 or 1) will be scaled using the Standard Scaler method in the python Sklearn library [29].

3.5.2 Vectorization of Target Output Class

The column better_option contains three values, “Buy”, Sell” and “Hold”. However, it must be transformed for use by classification models. As such, the LabelEncoder method from Sklearn [28] can be used to transform the column into a 3x1 vector (Example: [1 0 0]) corresponding to the appropriate target output class.

3.5.3 Class Balancing Strategy

Due to the problem being multiclass in nature, it is important to check for class balance between buy, hold, and sell. Upon initial inspection of the training dataset, it is clear that hold is much more common than buy or sell and accounts for almost 50% of the dataset. Given that there is much data to be utilized, an undersampling strategy makes more sense, as it doesn't introduce artificial rows that do not exist in reality. As such, the Imblearn library function RandomUnderSampler [15] can be utilized to ensure the data is balanced between the three labels. After implementation, the dataset is equally balanced between buy, sell and hold. It is important to mention that the validation dataset will not be adjusted.

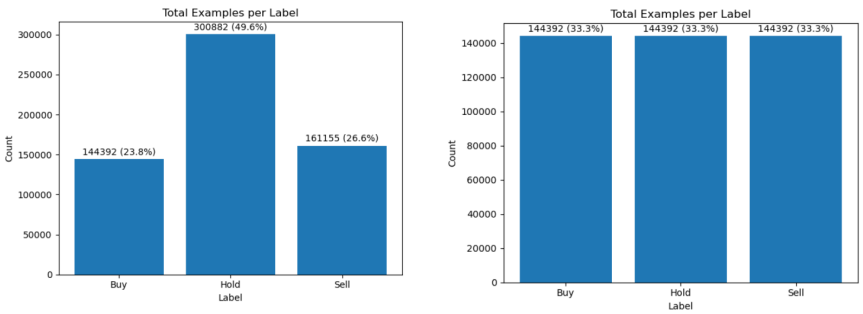

Figure 3.6: Class Balance Before (Left) and After (Right) Undersampling

To recap the steps taken for preprocessing the data, as well as feature engineering, a diagram of the data pipeline has been provided below. The following pipeline will be used in Figure 3.7

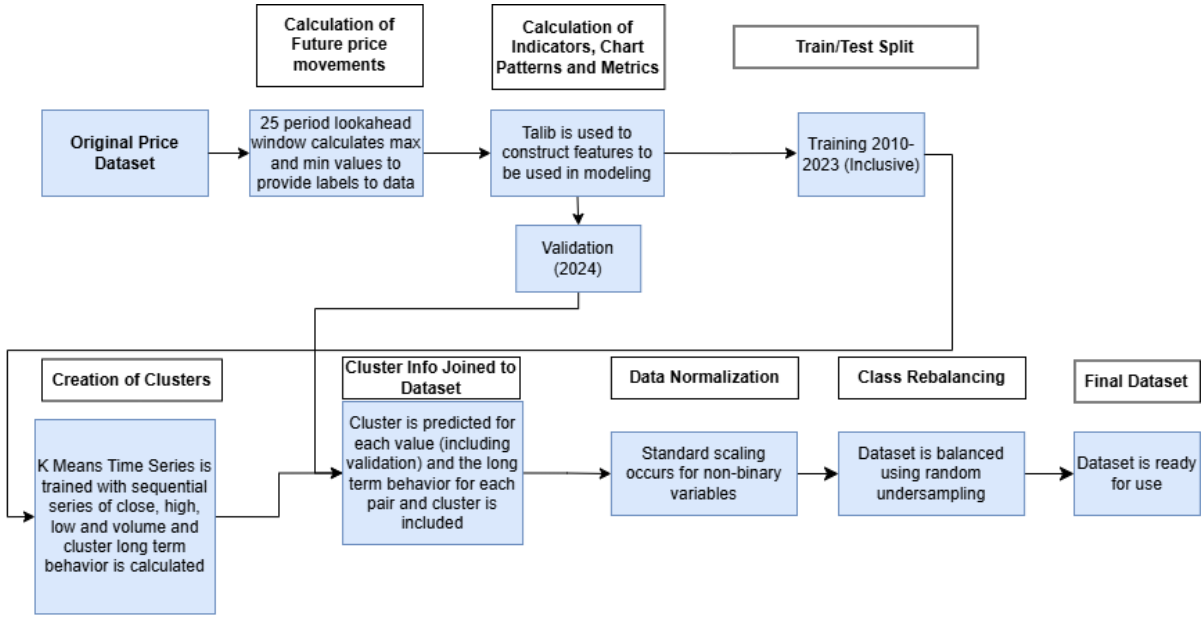

Figure 3.7: Pipeline of Data Preprocessing

3.6 Classifiers to be Used

While many classifiers can and could be utilized for classifying short term forex price movements, it was determined to focus more squarely on just three well-known and utilized classifiers which have been demonstrated to have been used previously in the literature. These classifiers can be used in a multi-classification setting, have won various awards in performance and even today can outperform the most sophisticated deep learning models. The models to be used are given below. Each model has a library available for open source use in Python. Although it was desired to also also utilize the Support Vector Classifier as part of this study, it proved to be quite slow in early model fitting and was excluded. They are Logistic Regression [27], Random Forest [24], and Extreme Gradient Boosting Classifier [39].

Model	**Library**
Logistic Regression	Sklearn
Random Forest	Sklearn
XGBClassifier	XGBoost

Table 3.6: Classifiers to be Used

It is important to call out which model hyperparameters will be used by default for each classifier.

Once each subset selection method has been tested, the overall classification performance can be measured using F1 score, class-weighted precision, recall, log loss and accuracy and the best subset selection method can be determined. From there, each model can undergo a hyperparameter grid search for the most important com-

Model	Grid
Logistic Regression	**C**: 1 **Penalty:** L2
Random Forest	**n_estimators**: 100 **criterion:** 'gini' **max_depth**: None **min_samples_split**: 2 **min_samples_leaf**: 1 **max_features**: 'sqrt' **class_weight**: None
XGBClassifier	**n_estimators**: 100 **learning_rate**: 0.01 **max_depth**: 3 **subsample**: 0.6 **colsample_bytree**: 0.6 **gamma**: 0 **scale_pos_weight**: 1

Table 3.7: Default Hyperparameters for each Table

ponents of each classifier using the best set of variables. A custom iterative loop has been created for the grid search.

Model	Grid
Logistic Regression	**C**: [.1,.5,1,2,5,10] **Penalty:** [None, 'L1', 'L2', 'ElasticNet']
Random Forest	**max_depth**: [None, 10, 20, 30], **min_samples_split**: [2, 5, 10], **min_samples_leaf**: [1, 2, 4],
XGBClassifier	**max_depth**: [3, 6, 10], **min_child_weight:** [3,5,10], **subsample:** [0.6, 0.8, 1], **colsample_bytree**: [0.6, 0.8, 1.0]

Table 3.8: Hyperparameter Grid Search for each Classifier

With such a large dataset (over 400 thousand rows and over 100 variables), a modest grid size was desired, therefore, relatively few hyperparameters are selected as well as few values per parameter. Logistic Regression on its own does not contain many hyperparameters and the only meaningful set of parameters to test are the inclusion of a penalty such as L1, L2 or elastic net regularization. The C hyperparameter can also be used to control the amount of regularization used. With respect to Random Forest, van Rijn and Hutter [21] in 2018 examined the various random forest parameters and their overall contribution to the variance in model performance. They concluded that minimum samples leaf (controls if an internal node is split), minimum samples split (threshold to become a leaf node) and bootstrap (adding variability to the dataset used to train) were the three most impactful features in model performance variability. By default, bootstrap is included as part of the random forest model. In turn,

the maximum depth can also be considered as it helps control the size of the tree. When left unchecked, at times the model can overfit. In a recent 2024 article related to the most important XGBoost hyperparameters by the company XGBoosting [39], the authors cite in order that max depth (how deep the tree is), followed by minimum child weight (minimum sum required in nodes), subsample (amount of training data to randomly sample) as well as column sample by tree (amount of columns to be randomly sampled) as the most important hyperparameters to test. Although the number of estimators is also an important hyperparameter, early testing showed a much longer run time with an increase in estimators for the XGBoost and Random Forest classifiers. With a default value of 100 estimators for both classifiers, there should be enough decision trees and weak learners for each respective method.

The classifiers will be evaluated and tested thoroughly. The first test will be the identification of the best subset of variables to use for the classifier. The classifiers will be tried using the following methods of feature subset selection. While by no means exhaustive (as this would require the testing of the powerset of variables), it is certainly beneficial to at least experiment with several subsets of variables, as oftentimes all features are not needed and some features are simply not useful in classification. The models will use the default hyperparameters as provided by the library for this initial stage. The Univariate Filter and Wrapper method both come from Sklearn [25] and [26].

Subset Selection Method	**Method**	**Library**
None (All Variables)	N/A	N/A
Univariate Filter	Mutual Information Percentile Filter (Top 30%)	Sklearn
Wrapper Method + Univariate Base Filter	Sequential Forward Feature Selection along with initial Mutual Information Filter of >=.01	Sklearn

Table 3.9: Feature Subset Selection Method

3.7 Trading System

A simple trading system will be devised to be used by the best classifier and evaluate hypothetical long term performance. The model will iterate through the validation dataset in chronological order, and for each currency, provide a classification of buy, hold, or sell. In order to provide a higher degree of assurance, two consecutive buy or sell signals must appear for the trading system to take a potential action. If two consecutive signals appear, an action will be taken, provided that the model is currently not engaged in a trade. Each currency pair will be assigned a starting balance of $1,000 U.S. dollars and operate independently from one another. The trading system will only permit a single position open at a time per currency. Each trade will have clearly defined stop loss and take profit prices so as to simulate real risk management strategies employed by traders. When a trade position is open, the system will evaluate the open, close, high, and low prices. The system will check if either of these prices have been reached. If so, the trade will be closed out and evaluated either as a win, partial win, loss, or partial loss. The system will also maintain an active counter that tracks the number of periods in which the trade has been opened, to avoid leaving trades open indefinitely. For the sake of simplicity, if neither the trade stop loss nor take profit has been met, the trade will automatically close at the end of 25 consecutive trading periods (not necessarily hours). However, this also aligns nicely with the time horizon of which the classifier model has been trained on; twenty-five periods in the future. Beyond this time frame, the model has no reference point for which to make an informed classification. A set of decision trees for how the trading system will operate are given below.

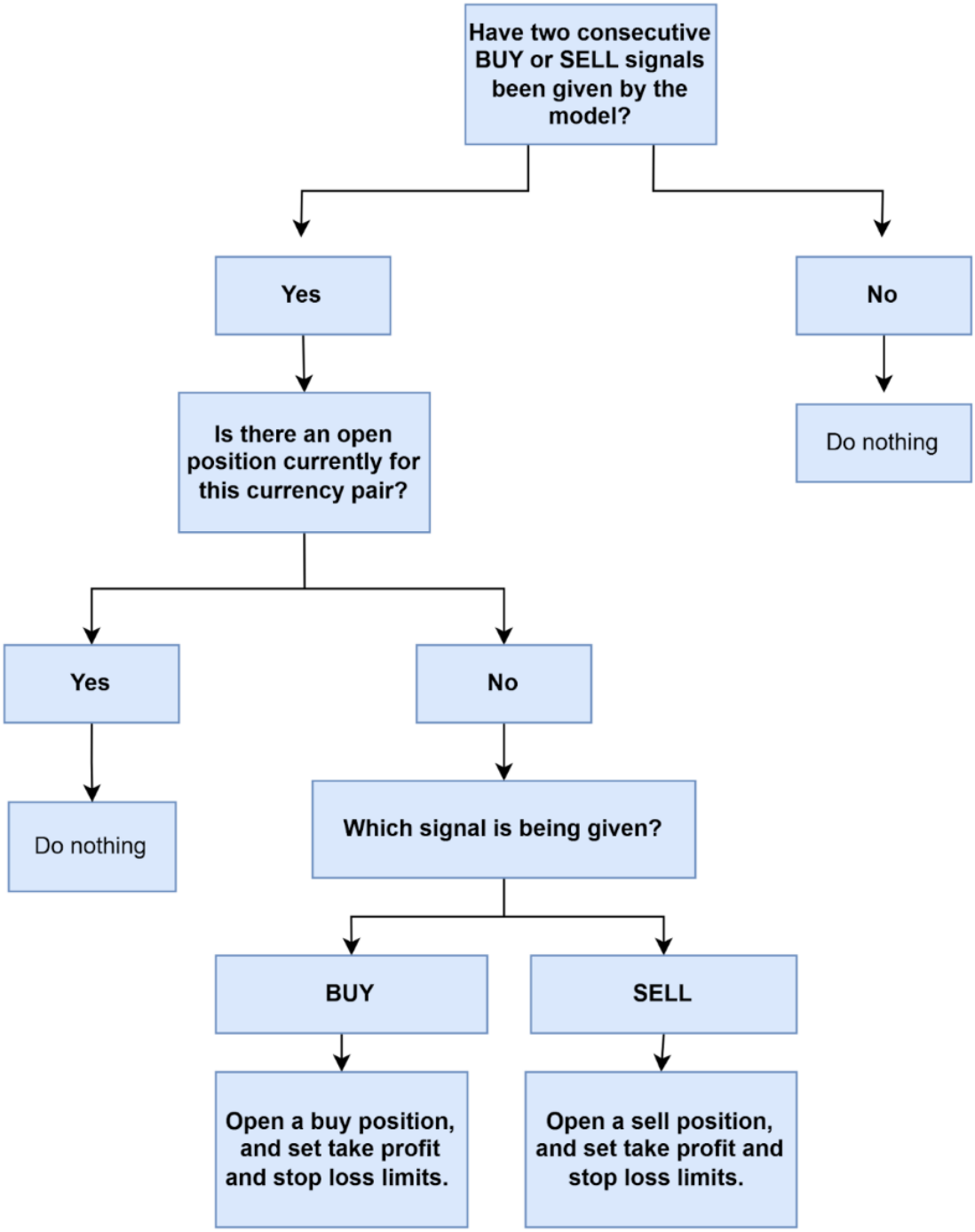

Figure 3.8: Decision Tree for Active Positions

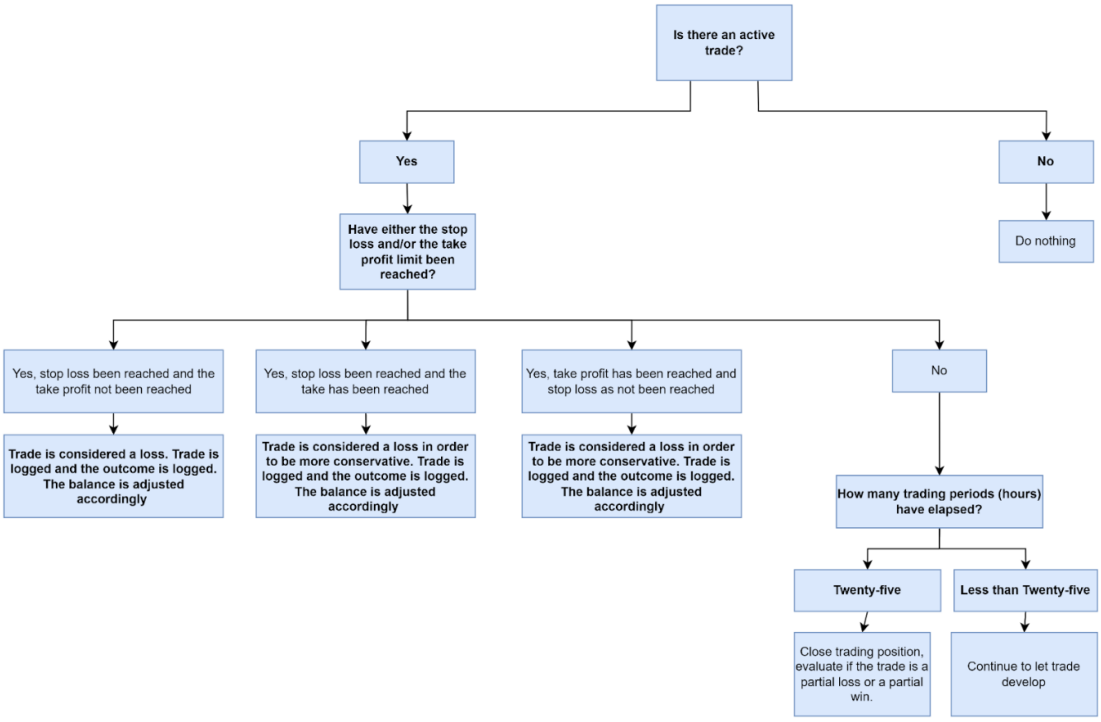

Figure 3.9: Decision Tree for Opening Positions

In addition, to help with the proper position sizes, the classifier probability can be used as a determinant. At an intuitive level, the more confident the classifier is, the more that should be invested in the position. A simple sliding scale has been created that places a larger percentage of the account balance when the probability of the classifier is higher. Although the strategy may be aggressive, the idea of the portfolio is to rely on the model to make decisions and not on human judgement.

Classifier Probability Range	**% of Portfolio to Invest**
>=.5 and <.6	3%
>=.6 and <.7	5%
>=.7 and <.8	7%
>=.8 and <.9	9%
>=.9	11%

Table 3.10: Position Sizing with Model Probability

One of the most difficult problems in conducting backtesting is finding a reasonable set of take-profit and stop-loss amounts to use. A search will also be conducted for the optimal amount of risk (in pips) for the stop loss and take profit amount for each currency pair. The combination with the highest average ending balance across the seven currency pairs will be utilized.

Variable	**Pip values to be Tried**
Take profit	25, 50, 75,100,125,150,175,200
Stop loss	25, 50, 75,100,125,150,175,200,225,250, 275, 300

Table 3.11: Pip Value Grid Search

Beyond this, it is worth mentioning additional assumptions required. The trading system will rely on 20:1 trading leverage, allowing a trader to hold a position size that is twenty times the size of their actual capital. Although there is no standard amount of leverage used in Forex trading, some brokers will allow up to 50:1 or 100:1 leverage, while some traders may trade using 1:1 leverage (100% trader capital). This amount errs on the conservative side.

Chapter 4

Results

After evaluating each base classifier with the three different subsets of variables, results are able to be compared using the validation set.

Metric	Model	Method		
		None	**Univariate Filter**	**Wrapper + Base Filter**
Accuracy	Logistic Regression	75.18%	71.00%	72.30%
	XGBoost	81.86%	76.64%	78.11%
	Random Forest	78.66%	74.59%	75.78%
F1 Score	Logistic Regression	0.751	0.710	0.724
	XGBoost	0.818	0.766	0.781
	Random Forest	0.786	0.746	0.728

Table 4.1: Model Performance by Feature Selection Method

The best class accuracy and F1 score were obtained for Logistic Regression while XGBoost and Random Forest classifiers perform best when using all variables. Accordingly, for the model hyperparameter grid search, all variables will be used. The features selected by the univariate method are enumerated in Appendix Table 6, and the variables selected by the wrapper and univariate filter have been included in Appendix Table 5.

The hyperparameter grid search yielded the following results (results for the best set of hyperparameters for each model are shown).

Model	Accuracy	F1 Score	Precision	Recall	Log Loss
Logistic Regression	75.2%	0.751	76.55%	75.22%	0.571
XGBoost	82.6%	0.826	83.07%	82.62%	0.410
Random Forest	78.76%	0.789	79.64%	78.86%	0.5197

Table 4.2: Tuned Model Results

The best performing hyperparameters from the search are shown below.

Model	Grid
Logistic Regression	**C**: 0.1 **Penalty**: L2
Random Forest	**max_depth**: None **min_samples_split**: 5 **min_samples_leaf**: 1
XGBClassifier	**max_depth**: 10 **min_child_weight**: 10 **subsample**: 1 **colsample_bytree**: 0.6

Table 4.3: Selected Hyperparameters from Grid Search

Unsurprisingly, the best results are only marginally better than the base results using the default hyperparameters. From initial inspection, it is clear that the XGBoost classifier has the best all-around performance, as it leads the three models in accuracy, F1 score, precision, recall, and log loss. The XGBoost classifier with its optimized set of hyperparameters will be selected for testing on the validation set in the context of the trading system.

The confusion matrix can also be displayed below for the leading classifier for the validation set.

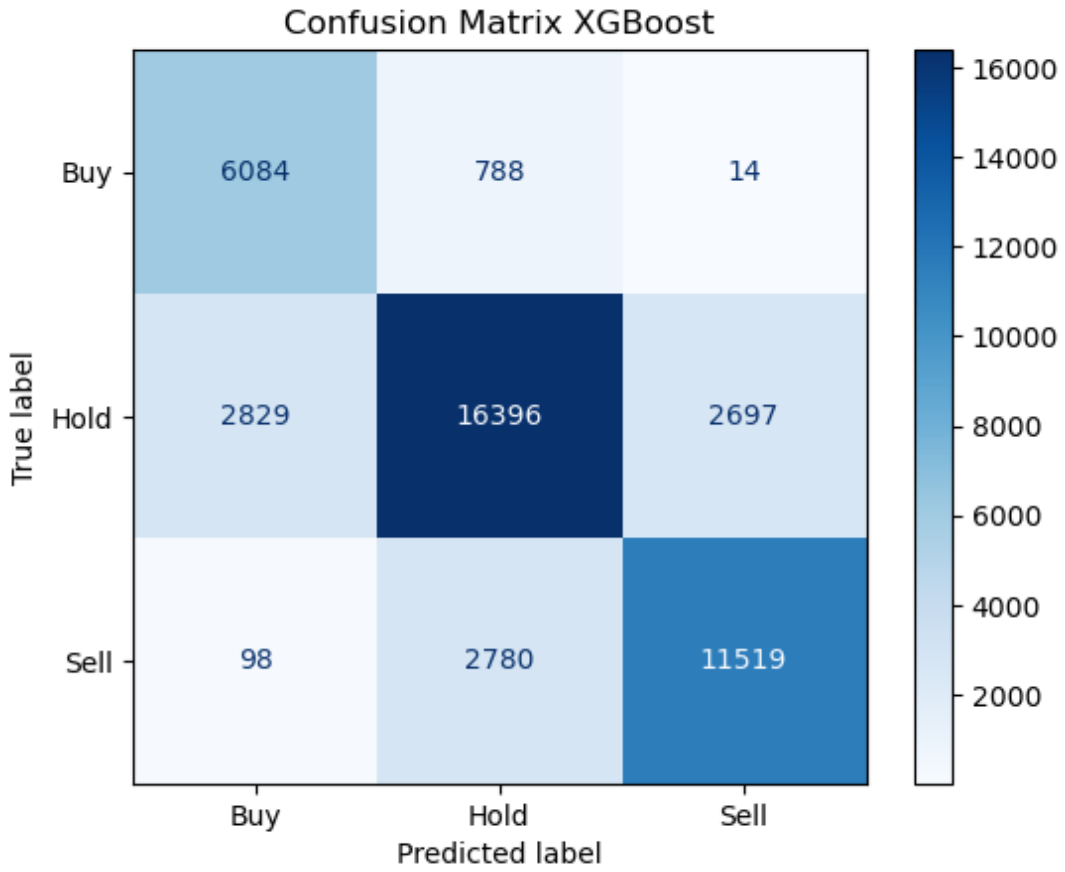

Figure 4.1: Confusion Matrix XGBoost

The results can be further broken down per class and currency using precision and recall.

Better Option	**Buy**		**Hold**		**Sell**	
Pair	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
AUDUSD	69.36%	91.83%	85.01%	74.60%	83.94%	85.06%
CHFUSD	72.52%	87.41%	81.03%	76.90%	83.73%	81.87%
EURUSD	78.98%	89.51%	90.38%	85.15%	80.77%	83.66%
GBPUSD	77.42%	90.97%	90.19%	80.89%	78.72%	86.44%
NZDUSD	66.73%	91.79%	77.76%	73.41%	87.99%	83.03%
USDCAD	61.49%	85.53%	85.57%	78.43%	81.24%	83.75%
USDJPY	80.94%	89.30%	87.68%	80.33%	82.61%	88.56%

Table 4.4: Precision and Recall Per Currency per Class

A summary can also be provided of total trading outcomes as well as the final simulated balance and drawdown statistics. As mentioned before, a search was conducted for the optimal amount of percentage in points (pips) for the take profit and stop loss values. It was determined that a stop loss of 200 pips and a take profit of 50 pips yielded the highest average ending balances across the seven major currency pairs.

Pair	**Total Trades**	**Part. Loss**	**Part. Win**	**Win**	**Loss**	**Max Drawdown (%)**	**Balance After ($)**	**Annual Return (%)**
AUDUSD	260	96	85	78	1	-8.49	2185.31	118.53
CHFUSD	311	93	41	169	8	-9.79	1873.64	87.36
EURUSD	203	87	63	53	0	-5.50	1219.33	21.93
GBPUSD	240	92	47	100	1	-5.34	1333.20	33.32
NZDUSD	302	89	56	152	5	-9.91	4976.31	397.63
USDCAD	252	93	35	120	4	-5.23	1391.87	39.19
USDJPY	313	93	22	180	18	-8.86	1112.48	11.25

Table 4.5: Trading Performance by Currency Pair

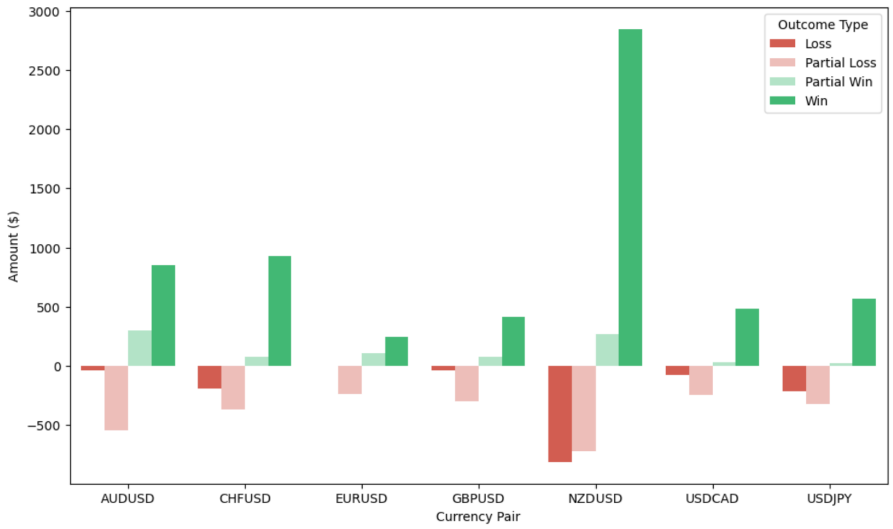

Figure 4.2: Sum of Total Outcomes by Outcomes Type

The monetary sum of the outcomes can also be expressed per currency graphically. It is helpful to visualize as well in terms of percentage of total amount.

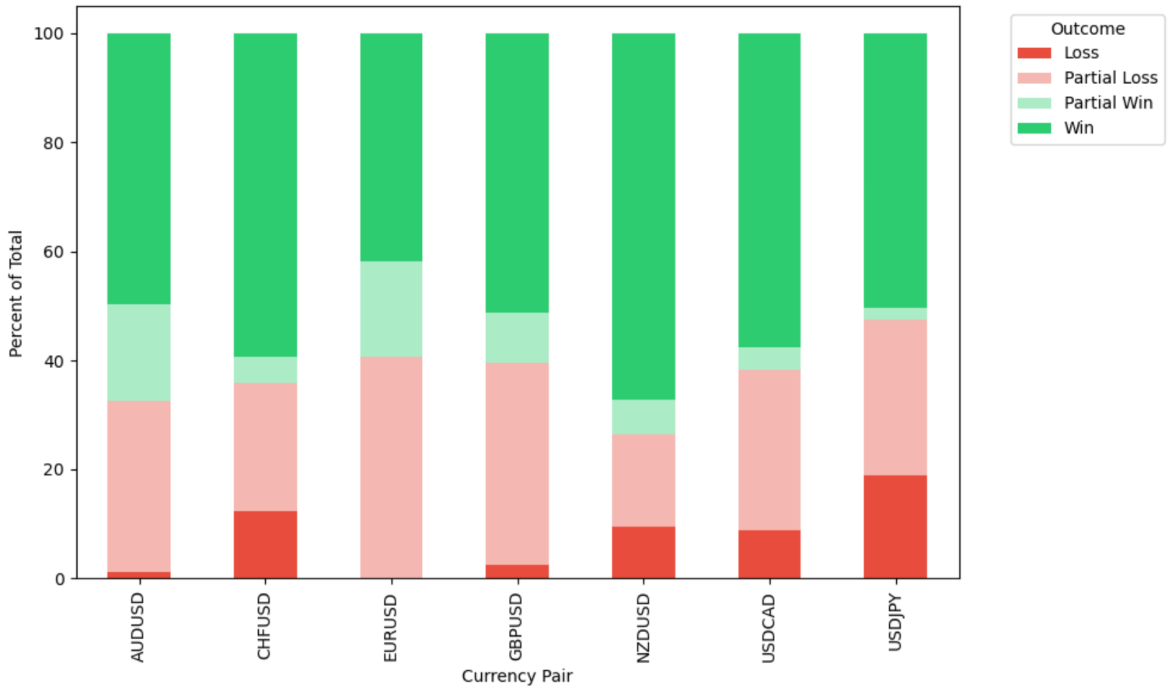

Figure 4.3: Percentage of Total Amount ($) by Outcome Type

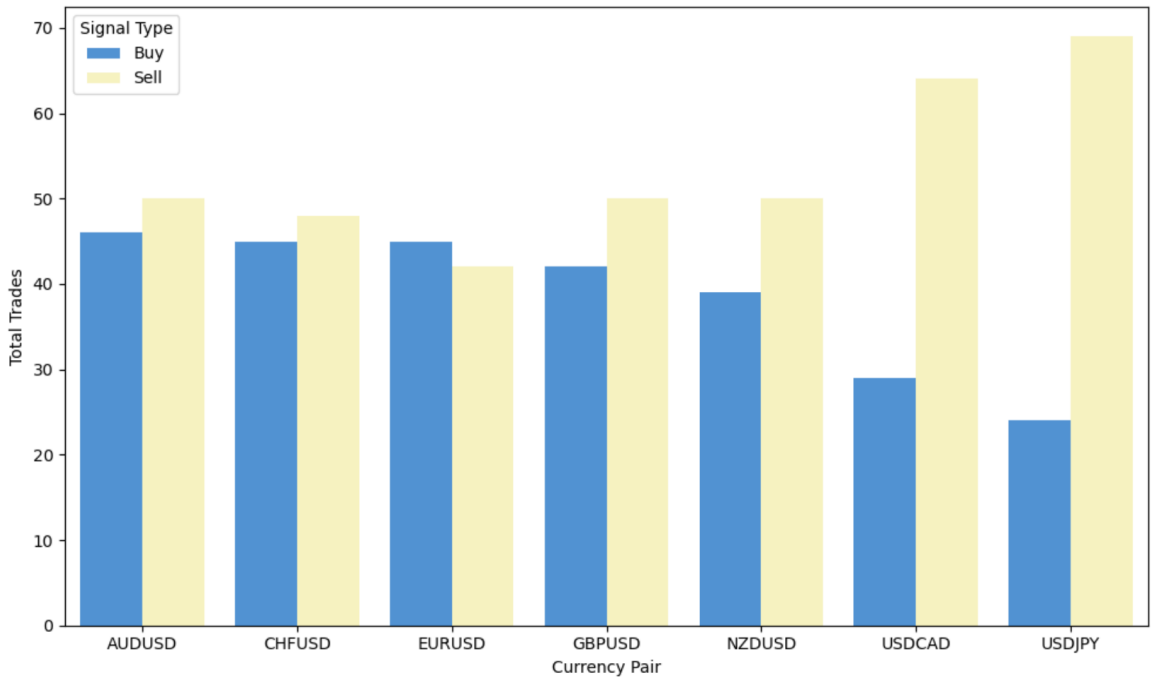

Figure 4.4: Partial Losses by Signal Type

An evolution of the mock portfolio balance over time can also be shown below.

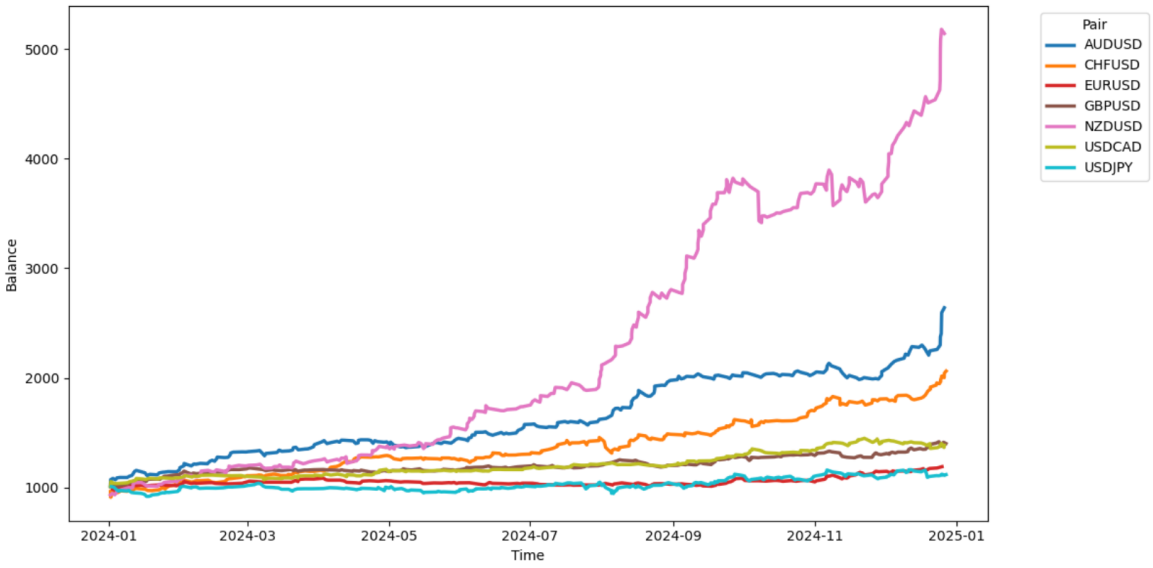

Figure 4.5: Portfolio Balance Evolution by Currency Pair

Not to be forgotten, all features have been ranked by mutual information score. The full list is included in Appendix Table 7 and average mutual information score by variable category is shown below.

Variable Set	Technical Indicator	Cluster Variables	Custom Variables	Pattern Variables
Average Mutual Information Score	0.2279	0.048	0.004	0.006

Table 4.6: Average Mutual Information Scores by Variable Group

Chapter 5

Discussion and Future Work

When considering the initial classifier results with different subsets of variables, it is interesting that the entire set of variables outperforms both the univariate filter and the wrapper method and univariate filter combination. This suggests that each of the models is able to prioritize the most important variables and that the variables with lower importance can still provide additional value, as is the case with the XGBoost classifier, showcasing a five percentage point improvement in accuracy with using all variables as opposed to the univariate filter set. When considering the performance of the tuned models, it can be shown that each model performs slightly better in the area of precision than recall, with a difference of around 1 percent. In context for precision, this means that when a buy, hold, or sell prediction is given, it is more likely to belong to this class. Conversely, when analyzing the ground truth buy hold and sell,s the model is slightly less likely to detect them. The strong predictive accuracy is evident in the confusion matrix of the XGBoost classifier, as the diagonals of the matrix constitute the lion's share of total predictions. Notably, the model classification mistakes that would be costly for a forex trader (predicted buy when the true value is a sell and vice versa) are quite limited. They comprise just .26% of the 43205 model predictions for the year of 2024. Of interest as well are the cases where the model predicts a hold as opposed to a true sell or buy, this happens in roughly 8% of the validation set examples. These are effectively opportunities that are more difficult to spot but present an opportunity to enter into a profitable trade. Similarly, approximately 12.8% of predictions point towards a buy or sell but are in fact a hold. The cost of these errant predictions can be better revealed through analysis of the trading system results. The log loss of.41 along with an F1 score of.826 for the best classifier is encouraging, but shows some room for future improvement. More advanced models such as deep neural network classifiers are unlikely to outperform the XGBoost classifier by a significant margin; it is likely that new features not expressed in this study would improve classification accuracy. Nevertheless, 82.6% accuracy is quite impressive for a market which has been said to at times exhibit a random walk behavior [5].

From first glance of the trading system outcomes, the results appear quite promising. Using a very low reward to risk ratio (1:4 or 50 pips to 200 pips), each currency is able to deliver at least an 11% annual return or better with a small maximum drawdown amount, and an astonishing 400% return is realized with the NZDUSD pair. The portfolio balance over time in Figure 4.5 shows a promising upward trend for all

currencies. The heterogeneity of the results is quite remarkable, depending on which currency pair is being considered. Some currencies exhibit more modest returns while the AUDUSD and NZDUSD deliver quite strong returns year over year. It appears that the AUDUSD and NZDUSD are the most well predicted currencies by the classifier. These currencies, along with the USDCAD tend to move strongly with the commodities market [4], which perhaps moves in a more predictable manner. However, it is also worth noting that these findings are true only for the validation dataset (2024). A further improvement to this study could be a leave one out cross validation strategy where all years are used as the training and test dataset. This could help provide more robustness to the results, however the computing time required to fully implement this validation strategy proved to be too lengthy for the time permitted to complete this study.

However, from a certain point of view, there is more efficiency to be extracted out of the system than meets the eye. The classifier is able to achieve 82.6% accuracy, yet the money flow by outcome in Figure 4.3 shows that in aggregate, anywhere from 26-47% of total money is lost via losses or partial losses. Partial losses in particular are a key concern for performance. Figure 4.4 shows that apart from the EURUSD pair, sell signals appear more likely to generate losses, though this is likely explained by the imbalanced nature of the validation set, which favored more sells than buys as evidenced in Figure 4.1 of the confusion matrix. It is more likely that buy is a trickier class for the classifier to handle. From analysis of Table 4.4, buy tends to exhibit notably less precision than the hold or sell classes. The most extreme example shows a precision of buy classification of just 61% for the USDCAD currency pair.

Perhaps dynamic risk limits would narrow the gap between the classifier and trading performance, as risk limits of varying sizes could help adequately determine when to exit a trade. This would likely increase the percentage of total losses (where the stop loss was met), but potentially reduce the amount of downside risk present in trades with a high safety margin. These risk limits could be sized using metrics such as average true range, which have been known to help capture the amount of volatility in the market. Furthermore, adjusting the arbitrary twenty-five period trade limit implemented could also help improve the win rate, either by allowing speculative positions to fully materialize, or to stop trades which continue to sink farther from their target objective. Another notable limitation of the trading system is that only one trade is allowed at a time. If multiple trades were allowed (provided that there was adequate available capital), the system would be able to take advantage of the missed opportunities that may be developing while a current trade is in progress. Finally, another strategy which could be explored would be to pile onto a winning trade by opening more positions which are aligned with the initial take profit goal. For example, if a trade is at 50% or 75% of its target take-profit amount, a new position with the same take-profit price amount as the initial position could be added. However, with all of these suggested improvements, the coding implementation for backtesting grows more complex and the assumptions required are more numerous.

Regarding which variables are the most important contributors to the classifier's decision making process, it is abundantly clear that the set of technical indicators of variables outperforms that of the other three classes with respect to mutual information. Variables such as relative strength index, chande momentum oscillator, moving average convergence/divergence have a relatively high mutual information score. Nevertheless, the work which has been done for custom variables such as price_vs_ema20

and the cluster long term behavior variable cluster_up_minus_down_pct_change_30 do demonstrate additional value. With special regard to the clustering work that was done, there is still much room for improvement. As mentioned before, the best clustering fit using k-means dynamic time warping achieved an average silhouette score of just 0.11, suggesting the clusters can be much better separated. Still, it is unclear whether it is possible to achieve this separation, as many years of data was utilized for the clustering algorithm to use.

Returning to the broader discussion, with the right classifiers, the collective sum of the variables with a small contribution can show that having many different variables can lead to stronger classification performance than simply reducing the variable set was shown in early comparisons of model testing. In summary, a forex price classifier with 82.6% accuracy and an F1 score of.826 was able to be produced with profit-making ability of at least 11% annually. The most useful variables in assisting the classifier came mostly from the technical indicators, though moving averages proved to be useful, and to a lesser extent, the cluster feature engineering. The chart pattern variables proved to be the least useful as a variable category. The trading system, when coupled with the classifier, yielded strong annual returns with a reasonable maximum drawdown during the validation testing period, although it shows room for improvement in the area of minimizing partial losses incurred by the trading rules.

References

[1] 5paisa. *Three Black Crows Pattern*. https://www.5paisa.com/finschool/three-black-crows-pattern/. Accessed: 2025-06-19. 2025.

[2] 5paisa. *Three Inside Up/Down Candlestick Pattern*. https://www.5paisa.com/finschool/three-inside-up-down-candlestick-pattern/. Accessed: 2025-06-19. 2025.

[3] Bank for International Settlements. *OTC foreign exchange turnover in April 2022*. https://www.bis.org/statistics/rpfx22_fx.htm. Accessed: 2025-06-19. 2022.

[4] James Chen. *Commodity Pairs*. Accessed: 2025-06-22. Investopedia. Apr. 2024. URL: https://www.investopedia.com/terms/forex/c/commodity-pairs.asp.

[5] James Chen. *Random Walk Theory*. Accessed: 2025-06-22. Investopedia. July 2022. URL: https://www.investopedia.com/terms/r/randomwalktheory.asp.

[6] C. Cliffe. *What are the major currency pairs?* https://www.ig.com/en/trading-strategies/major-currency-pairs-190618. Accessed: 2025-05-22. 2019.

[7] P. D'Urso, L. De Giovanni, and R. Massari. "Trimmed fuzzy clustering of financial time series based on dynamic time warping". In: *Annals of Operations Research* (2019). DOI: 10.1007/s10479-019-03284-1.

[8] Y. Dave. "Predicting Forex Pair Movements: Integrating Sentiment Analysis, Technical, and Fundamental Indicators using Machine Learning and Deep Learning Models". Accessed: 2025-06-19. M.Appl.Tech. thesis. School of Computing, Electrical and Applied Technology, New Zealand, 2024. URL: https://www.researchbank.ac.nz/server/api/core/bitstreams/08bb9fd2-bcdb-4d2c-a1e8-a12dc2845b66/content.

[9] Finance Magnates. *Institutional Trading*. https://www.financemagnates.com/terms/i/institutional-trading/. Accessed: 2025-05-22. 2025.

[10] M. Fisichella and F. Garolla. *Can Deep Learning Improve Technical Analysis of Forex Data to Predict Future Price Movements?* 2021. DOI: 10.1109/ACCESS.2021.3127570.

[11] Forex Academy. *How much forex hedge fund made to investors?* https://www.forex.academy/how-much-forex-hedge-fund-made-to-investors/. Accessed: 2025-05-22. 2025.

[12] Z. Gong, C. Ventre, and J. G. O'Hara. *Classifying High-Frequency FX Rate Movements with Technical Indicators and Inception Model*. New York, NY, USA, 2020. DOI: 10.1145/3383455.3422560.

[13] Hantec Markets. *Risk Reward Ratios | Managing Your Risk*. https://hmarkets.com/learn-to-trade/managing-your-risk/risk-reward-ratios/. Accessed: 2025-05-22. 2025.

[14] M. Hauge. *The History of Algorithmic Trading*. https://www.quantifiedstrategies.com/history-of-algorithmic-trading/. Accessed: 2025-06-19. 2023.

[15] imbalanced-learn contributors. *RandomUnderSampler — Version 0.13.0 API reference*. https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html. Accessed: 2025-06-13. 2024.

[16] Investopedia. *Technical Indicator: Definition, Analyst Uses, Types and Examples*. https://www.investopedia.com/terms/t/technicalindicator.asp. Accessed: 2025-05-22. 2025.

[17] S. Kim et al. "Pattern Matching Trading System Based on the Dynamic Time Warping Algorithm". In: *Sustainability* 10.12 (2018), p. 4641. DOI: 10.3390/su10124641.

[18] J. Kozák and P. Juszczuk. *Portfolio Investments in the Forex Market*. 2022. DOI: 10.1007/978-3-031-21743-2_8.8.

[19] M. Mahootchi and Alireza Fereydooni. "An algorithmic trading system based on a stacked generalization model and hidden Markov model in the foreign exchange market". In: *Global Finance Journal* (2023). DOI: 10.1016/j.gfj.2023.100825.

[20] A. H. Moghaddam and S. Momtazi. *Image processing meets time series analysis: Predicting Forex profitable technical pattern positions*. 2021. DOI: 10.1016/j.asoc.2021.107460.

[21] R. S. Oetama et al. *Improving Candle Direction Classification in Forex Market Using Support Vector Machine with Hyperparameters Tuning*. Denpasar, Bali, Indonesia, 2022. DOI: 10.1109/ICIC56845.2022.10006993.

[22] Leslie Ow, David Chek Ling Ngo, and Yunli Lee. "Prediction of forex trend movement using linear regression line, two-stage of multi-layer perceptron and dynamic time warping algorithms". In: *Journal of Information and Communication Technology* 15 (Dec. 2016), pp. 117–140. DOI: 10.32890/jict2016.15.2.6.

[23] K. S. Y. Pande, D. G. H. Divayana, and G. Indrawan. *Comparative analysis of naïve Bayes and KNN on prediction of forex price movements for GBP/USD currency at time frame daily*. 2021. DOI: 10.1088/1742-6596/1810/1/012012.

[24] F. Pedregosa et al. *sklearn.ensemble.RandomForestClassifier*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed: 2025-05-28. 2025.

[25] F. Pedregosa et al. *sklearn.feature$_s$election.SelectPercentile*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html. Accessed: 2025-05-28. 2025.

[26] F. Pedregosa et al. *sklearn.feature_selection.SequentialFeatureSelector*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html. Accessed: 2025-05-28. 2025.

[27] F. Pedregosa et al. *sklearn.linear$_{m}$odel.LogisticRegression*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Accessed: 2025-06-10. 2025.

[28] F. Pedregosa et al. *sklearn.preprocessing.LabelEncoder*. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html. Accessed: 2025-05-28. 2025.

[29] F. Pedregosa et al. *sklearn.preprocessing.StandardScaler*. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. Accessed: 2025-06-19. 2025.

[30] Polygon.io. *Forex API & Crypto API*. https://polygon.io/currencies. Accessed: 2025-04-16. 2025.

[31] A. Sadeghi, A. Daneshvar, and M. M. Zaj. *Combined ensemble multi-class SVM and fuzzy NSGA-II for trend forecasting and trading in Forex markets*. 2021. DOI: 10.1016/j.eswa.2021.115566.

[32] A. Santuci et al. *Evaluation of FOREX trading Strategies based in Random Forest and Support Vector Machines*. 2022. DOI: 10.1109/TLA.2022.9878170.

[33] Scikit-learn developers. *sklearn.metrics.silhouette$_{s}$core*. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html. Accessed: 2025-06-03. 2025.

[34] T. Financial. *TA-Lib: Technical Analysis Library*. https://ta-lib.org/. Accessed: 2025-04-16. 2024.

[35] R. Tavenard et al. "tslearn: A Machine Learning Toolkit for Time Series Data". In: *Journal of Machine Learning Research* 21.118 (2020). Accessed: 2025-06-19, pp. 1–6. URL: https://tslearn.readthedocs.io/en/stable/.

[36] P. Tsinaslanidis et al. *Dynamic Time Warping as a Similarity Measure: Applications in Finance - Kent Academic Repository*. Accessed: 2025-06-19. 2014. URL: https://kar.kent.ac.uk/43498/1/HFAA%20paper%20-%20Tsinaslanidis%20etal.pdf.

[37] P. E. Tsinaslanidis. "Subsequence dynamic time warping for charting: Bullish and bearish class predictions for NYSE stocks". In: *Expert Systems with Applications* 94 (2018), pp. 193–204.

[38] XcalData Blog. *Two Crows Pattern: Navigating Bearish Signals with Candlestick Analysis*. https://blog.xcaldata.com/two-crows-pattern-navigating-bearish-signals-with-candlestick-analysis/. Accessed: 2025-06-19. 2024.

[39] XGBoost Developers. *xgboost.XGBClassifier*. https://xgboost.readthedocs.io/en/release_3.0.0/python/python_api.html#xgboost.XGBClassifier. Accessed: 2025-05-28. 2025.

Appendix

This Appendix contains different tables referenced throughout the text.

Cluster	cluster_up	cluster_down	cluster_up_minus_down
0	0.47%	0.47%	3.12934E-07
1	0.52%	0.53%	0.0001246
2	0.31%	0.64%	0.003334537
3	0.63%	0.31%	-0.003229968
4	0.44%	0.45%	2.3062E-05
5	0.41%	0.46%	0.000515152
6	0.48%	0.60%	0.001201573
7	0.45%	0.45%	1.67583E-05
8	0.41%	0.43%	0.00017348
9	0.57%	0.46%	-0.001156336
10	0.44%	0.53%	0.000856618
11	0.50%	0.41%	-0.000936502
12	0.55%	0.48%	-0.000604459
13	0.34%	0.53%	0.001852691
14	0.38%	0.56%	0.001757399
15	0.40%	0.58%	0.001798846
16	0.51%	0.40%	-0.00105566
17	0.45%	0.45%	8.18327E-05
18	0.46%	0.56%	0.000946783
19	0.48%	0.55%	0.000730103
20	0.41%	0.50%	0.000920346
21	0.50%	0.52%	0.000163815
22	0.34%	0.65%	0.00305365
23	0.71%	0.45%	-0.002546579
24	0.34%	0.73%	0.003860425
25	0.43%	0.50%	0.000728259
26	0.49%	0.58%	0.000939253
27	0.42%	0.56%	0.001326391
28	0.41%	0.47%	0.000606439
29	0.45%	0.43%	-0.000223413

Table 1: Technical Indicators Overview

Indicator	**Abbr**	**Description**	**Parameters**	**Variables Used**
Relative Strength Index	RSI	Momentum oscillator for identifying changes in momentum and price direction.	timeperiod=14	Close
Avg Directional Movement Index	ADXR	Positive and negative direction indicators to detect trends.	timeperiod=9	High, Low, Close
Absolute Price Oscillator	APO	Difference between fast and slow period moving averages.	fastperiod=12, slowperiod=26, matype=0	Close
Aroon Down / Aroon Up	AROON	Measures time since highest high (Up) or lowest low (Down).	timeperiod=14	High (Up), Low (Down)
Aroon Oscillator	AROONOSC	Combines Aroon Up and Aroon Down into one value.	timeperiod=14	High, Low
Balance Of Power	BOP	Discerns buyer vs seller strength.	–	Open, High, Low, Close
Commodity Channel Index	CCI	Compares current price to historical average.	timeperiod=14	High, Low, Close
Chande Momentum Oscillator	CMO	Measures price momentum based on recent gains and losses.	timeperiod=14	Close
Directional Movement Index	DX	Compares directional price movement between days.	timeperiod=14	High, Low, Close
Moving Average Convergence/Divergence	MACD	Uses short and long term moving averages to detect momentum changes.	fastperiod=12, slowperiod=26, signalperiod=9	Close

Indicator	**Abbreviation**	**Description**	**Parameters Used**	**Variables Used**
Money Flow Index	MFI	Volume-weighted version of RSI.	timeperiod=14	High, Low, Close, Volume
Minus Directional Indicator	MINUS_DI	Component of ADX indicating downward trend strength.	timeperiod=14	High, Low, Close
Momentum	MOM	Measures speed of price change.	timeperiod=14	Close
Plus Directional Indicator	PLUS_DI	Component of ADX indicating upward trend strength.	timeperiod=14	High, Low, Close
Percent Price Oscillator	PPO	Ratio of short to long period exponential moving averages.	fastperiod=12, slowperiod=26, matype=0	Close
Rate of Change (ROC)	ROC	Calculates price change rate over time.	timeperiod=10	Close
Stochastic Oscillator	STOCH	Identifies overbought/oversold conditions.	fastk_period=5, slowk_period=3, slowk_matype=0, slowd_period=3, slowd_matype=0	High, Low, Close
Stochastic Fast	STOCHF	Variant of Stochastic Oscillator with less smoothing.	fastk_period=5, fastd_period=3, fastd_matype=0	High, Low, Close
TRIX	TRIX	Triple smoothed exponential moving average.	timeperiod=30	Close
Ultimate Oscillator	ULTOSC	Combines momentum from multiple timeframes to reduce false signals.	timeperiod1=7, timeperiod2=14, timeperiod3=28	High, Low, Close
Williams' %R	WILLR	Measures close relative to high-low range.	timeperiod=14	High, Low, Close
Average True Range	ATR	Measures market volatility using high-low-close ranges.	timeperiod=14	High, Low, Close

Indicator	**Abbreviation**	**Description**	**Parameters Used**	**Variables Used**
Bollinger Band (Width)	BBANDS	Uses standard deviation of price to define bands.	timeperiod=5, nbdevup=2, nbdevdn=2, matype=0	Close
Chaikin A/D Line	AD	Cumulative volume-based indicator for buying/selling pressure.	–	High, Low, Close, Volume
On Balance Volume	OBV	Uses volume changes relative to price movements.	–	Close, Volume

Table 2: Technical Indicators Overview

Pattern Name	**Abbr**	**Description**
Two Crows	CDL2CROWS	Bearish reversal pattern indicating potential top.
Three Black Crows	CDL3BLACKCROWS	Strong bearish reversal pattern with three consecutive long black candles.
Three Inside Up/-Down	CDL3INSIDE	Bullish or bearish reversal pattern within a three-candle formation.
Three-Line Strike	CDL3LINESTRIKE	Reversal pattern with three candles in trend direction followed by a large opposite candle.
Three Outside Up/-Down	CDL3OUTSIDE	Bullish or bearish reversal pattern with engulfing characteristics.
Three Stars In The South	CDL3STARSINSOUTH	Bullish reversal pattern with three small-bodied candles.
Three Advancing White Soldiers	CDL3WHITESOLDIERS	Strong bullish reversal pattern with three consecutive long white candles.
Abandoned Baby	CDLABANDONEDBABY	Bullish or bearish reversal pattern with a doji gap between two candles.
Advance Block	CDLADVANCEBLOCK	Bearish reversal pattern with weakening upward momentum.
Belt-hold	CDLBELTHOLD	Single candle pattern indicating potential reversal.
Breakaway	CDLBREAKAWAY	Five-candle pattern signaling trend reversal.
Closing Marubozu	CDLCLOSINGMARUBOZU	Strong continuation pattern with no shadow on closing side.
Concealing Baby Swallow	CDLCONCEALBABYSWALL	Rare bullish reversal pattern in a downtrend.
Counterattack	CDLCOUNTERATTACK	Reversal pattern with two candles of opposite color and similar close.
Dark Cloud Cover	CDLDARKCLOUDCOVER	Bearish reversal pattern where a black candle opens above previous close and closes below midpoint.
Doji	CDLDOJI	Neutral pattern indicating indecision in the market.
Doji Star	CDLDOJISTAR	Potential reversal pattern with a doji following a long candle.
Dragonfly Doji	CDLDRAGONFLYDOJI	Bullish reversal pattern with long lower shadow and no upper shadow.

Pattern Name	**Abbreviation**	**Description**
Engulfing Pattern	CDLENGULFING	Reversal pattern where a larger candle engulfs the previous smaller candle.
Evening Doji Star	CDLEVENINGDOJISTAR	Bearish reversal pattern with a doji between two candles.
Evening Star	CDLEVENINGSTAR	Bearish reversal pattern with a small-bodied candle between two others.
Up/Down-gap side-by-side white lines	CDLGAPSIDESIDEWHITE	Continuation pattern with two white candles gapping up or down.
Gravestone Doji	CDLGRAVESTONEDOJI	Bearish reversal pattern with long upper shadow and no lower shadow.
Hammer	CDLHAMMER	Bullish reversal pattern with long lower shadow and small body.
Hanging Man	CDLHANGINGMAN	Bearish reversal pattern similar to hammer but in an uptrend.
Harami Pattern	CDLHARAMI	Reversal pattern with a small candle within the previous large candle's body.
Harami Cross Pattern	CDLHARAMICROSS	Reversal pattern with a doji inside the previous candle's body.
High-Wave Candle	CDLHIGHWAVE	Indecision pattern with long upper and lower shadows.
Hikkake Pattern	CDLHIKKAKE	Continuation pattern with a false breakout followed by a true breakout.
Modified Hikkake Pattern	CDLHIKKAKEMOD	Variation of Hikkake pattern indicating continuation.
Homing Pigeon	CDLHOMINGPIGEON	Bullish reversal pattern with a smaller candle within a larger black candle.
Identical Three Crows	CDLIDENTICAL3CROWS	Bearish reversal pattern with three black candles of similar size.
In-Neck Pattern	CDLINNECK	Bearish continuation pattern with a small white candle closing near previous low.
Inverted Hammer	CDLINVERTEDHAMMER	Bullish reversal pattern with long upper shadow and small body.
Kicking	CDLKICKING	Reversal pattern with opposite marubozu candles gapping away from each other.

Pattern Name	**Abbreviation**	**Description**
Kicking by Length	CDLKICKINGBYLENGTH	Variation of Kicking pattern considering the length of marubozu candles.
Ladder Bottom	CDLLADDERBOTTOM	Bullish reversal pattern with a series of declining candles followed by a bullish candle.
Long Legged Doji	CDLLONGLEGGEDDOJI	Indecision pattern with long upper and lower shadows.
Long Line Candle	CDLLONGLINE	Single candle with long body and short shadows indicating strong movement.
Marubozu	CDLMARUBOZU	Strong bullish or bearish candle with no shadows.
Matching Low	CDLMATCHINGLOW	Bullish reversal pattern with two candles having the same low price.
Mat Hold	CDLMATHOLD	Bullish continuation pattern with a brief consolidation.
Morning Doji Star	CDLMORNINGDOJISTAR	Bullish reversal pattern with a doji between two candles.
Morning Star	CDLMORNINGSTAR	Bullish reversal pattern with a small-bodied candle between two others.
On-Neck Pattern	CDLONNECK	Bearish continuation pattern with a small white candle closing near previous low.
Piercing Pattern	CDLPIERCING	Bullish reversal pattern where a white candle closes above the midpoint of the previous black candle.
Rickshaw Man	CDLRICKSHAWMAN	Indecision pattern with long upper and lower shadows and small body.
Rising/Falling Three Methods	CDLRISEFALL3METHODS	Continuation pattern with small candles within a larger trend.
Separating Lines	CDLSEPARATINGLINES	Continuation pattern with a gap between two candles of the same color.
Shooting Star	CDLSHOOTINGSTAR	Bearish reversal pattern with long upper shadow and small body.
Short Line Candle	CDLSHORTLINE	Single candle with short body and shadows indicating low volatility.
Spinning Top	CDLSPINNINGTOP	Indecision pattern with small body and longer shadows.

Pattern Name	**Abbreviation**	**Description**
Stalled Pattern	CDLSTALLEDPATTERN	Bearish reversal pattern with weakening upward momentum.
Stick Sandwich	CDLSTICKSANDWICH	Bullish reversal pattern with two black candles surrounding a white candle.
Takuri	CDLTAKURI	Bullish reversal pattern with long lower shadow indicating buying pressure.
Tasuki Gap	CDLTASUKIGAP	Continuation pattern with a gap followed by a candle in the opposite direction.
Thrusting Pattern	CDLTHRUSTING	Bearish continuation pattern with a white candle closing within previous black candle's body.
Tristar Pattern	CDLTRISTAR	Reversal pattern with three doji candles indicating indecision.
Unique 3 River	CDLUNIQUE3RIVER	Bullish reversal pattern with a specific three candle pattern.

Table 3: Candlestick Chart Patterns

Variable	**Type**
Relative Strength Index	Indicator
Avg Directional Movement Index	Indicator
Absolute Price Oscillator	Indicator
Aroon Down / Aroon Up	Indicator
Aroon Oscillator	Indicator
Balance Of Power	Indicator
Commodity Channel Index	Indicator
Chande Momentum Oscillator	Indicator
Directional Movement Index	Indicator
Moving Average Convergence/Divergence	Indicator
Money Flow Index	Indicator
Minus Directional Indicator	Indicator
Momentum	Indicator
Plus Directional Indicator	Indicator
Percent Price Oscillator	Indicator
Rate of Change (ROC)	Indicator
Stochastic Oscillator	Indicator
Stochastic Fast	Indicator
TRIX	Indicator
Ultimate Oscillator	Indicator
Williams' %R	Indicator
Average True Range	Indicator
Bollinger Band (Width)	Indicator
Chaikin A/D Line	Indicator
On Balance Volume	Chart Pattern
Two Crows	Chart Pattern
Three Black Crows	Chart Pattern
Three Inside Up/Down	Chart Pattern
Three-Line Strike	Chart Pattern
Three Outside Up/Down	Chart Pattern
Three Stars In The South	Chart Pattern
Three Advancing White Soldiers	Chart Pattern
Abandoned Baby	Chart Pattern
Advance Block	Chart Pattern
Belt-hold	Chart Pattern
Breakaway	Chart Pattern
Closing Marubozu	Chart Pattern
Concealing Baby Swallow	Chart Pattern
Counterattack	Chart Pattern
Dark Cloud Cover	Chart Pattern
Doji	Chart Pattern
Doji Star	Chart Pattern
Dragonfly Doji	Chart Pattern
Engulfing Pattern	Chart Pattern
Evening Doji Star	Chart Pattern
Evening Star	Chart Pattern
Up/Down-gap side-by-side white lines	Chart Pattern

Variable	**Type**
Gravestone Doji	Chart Pattern
Hammer	Chart Pattern
Hanging Man	Chart Pattern
Harami Pattern	Chart Pattern
Harami Cross Pattern	Chart Pattern
High-Wave Candle	Chart Pattern
Hikkake Pattern	Chart Pattern
Modified Hikkake Pattern	Chart Pattern
Homing Pigeon	Chart Pattern
Identical Three Crows	Chart Pattern
In-Neck Pattern	Chart Pattern
Inverted Hammer	Chart Pattern
Kicking	Chart Pattern
Kicking - bull/bear determined by longer marubozu	Chart Pattern
Ladder Bottom	Chart Pattern
Long Legged Doji	Chart Pattern
Long Line Candle	Chart Pattern
Marubozu	Chart Pattern
Matching Low	Chart Pattern
Mat Hold	Chart Pattern
Morning Doji Star	Chart Pattern
Morning Star	Chart Pattern
On-Neck Pattern	Chart Pattern
Piercing Pattern	Chart Pattern
Rickshaw Man	Chart Pattern
Rising/Falling Three Methods	Chart Pattern
Separating Lines	Chart Pattern
Shooting Star	Chart Pattern
Short Line Candle	Chart Pattern
Spinning Top	Chart Pattern
Stalled Pattern	Chart Pattern
Stick Sandwich	Chart Pattern
Takuri (Dragonfly Doji with very long lower shadow)	Chart Pattern
Tasuki Gap	Chart Pattern
Thrusting Pattern	Chart Pattern
Tristar Pattern	Chart Pattern
Unique 3 River	Chart Pattern
dayofweek	Custom Feature
hour (UTC)	Custom Feature
is_usd_jpy	Custom Feature
is_eur_usd	Custom Feature
is_gbp_usd	Custom Feature
is_chf_usd	Custom Feature
is_aud_usd	Custom Feature
is_nzd_usd	Custom Feature

Variable	**Type**
Price - EMA(20)	Custom Feature
EMA(200) - EMA(50)	Custom Feature
cluster_up_minus_down_pct_change_30	Custom Feature
Better Option	**Target Output**

Table 4: All Variables to be used

Variable/Model	**Logistic Regression**	**Random Forest**	**XGBoost**
IND_cmo	**X**	**X**	**X**
IND_rsi		**X**	**X**
IND_price_vs_ema20	**X**	**X**	**X**
IND_macd			
IND_roc		**X**	**X**
IND_mom			
IND_ppo	**X**	**X**	**X**
IND_apo			
IND_macdsignal			
IND_willr	**X**	**X**	**X**
IND_aroonosc		**X**	**X**
IND_cci	**X**	**X**	**X**
IND_mfi	**X**		
IND_trix	**X**	**X**	**X**
IND_minus_di	**X**		
IND_plus_di	**X**		
IND_slowd	**X**	**X**	**X**
IND_aroonup		**X**	**X**
IND_aroondown		**X**	**X**
IND_ultosc			
IND_slowk	**X**	**X**	**X**
IND_fastd		**X**	**X**
IND_dx	**X**		**X**
IND_macdhist			
IND_bb_width	**X**	**X**	**X**
PAT_CDLLONGLINE	**X**		
IND_fastk	**X**	**X**	**X**
PAT_CDLBELTHOLD			
IND_adxr	**X**		**X**
PAT_CDLCLOSINGMARUBOZU			
PAT_CDL3OUTSIDE		**X**	
PAT_CDLADVANCEBLOCK		**X**	**X**
IND_200_minus_50_ema	**X**		
PAT_CDLHARAMI			
IND_atr			
PAT_CDLMARUBOZU		**X**	
PAT_CDLDOJISTAR	**X**		
cluster_up_minus_down_pct_change_30			

Table 5: Wrapper Method + Univariate Filter Variables Selected

Variable
IND_dx
IND_macdsignal
IND_macdhist
IND_mfi
IND_mom
IND_plus_di
IND_ppo
IND_slowk
IND_slowd
IND_fastk
IND_fastd
IND_trix
IND_ultosc
IND_willr
PAT_CDL2CROWS
PAT_CDL3BLACKCROWS
PAT_CDL3INSIDE
PAT_CDL3LINESTRIKE
PAT_CDL3OUTSIDE
PAT_CDL3STARSINSOUTH
PAT_CDL3WHITESOLDIERS
PAT_CDLABANDONEDBABY
PAT_CDLADVANCEBLOCK
PAT_CDLBELTHOLD
PAT_CDLBREAKAWAY
PAT_CDLCLOSINGMARUBOZU
PAT_CDLEVENINGSTAR
PAT_CDLGAPSIDESIDEWHITE
PAT_CDLHAMMER
PAT_CDLSHORTLINE
cluster_up_minus_down_pct_change_30

Table 6: Univariate Method Filter Selected Variables

Variable	**Score**
IND_rsi	0.422288
IND_cmo	0.422263
IND_price_vs_ema20	0.398467
IND_macd	0.36913
IND_roc	0.362796
IND_mom	0.358355
IND_ppo	0.335717
IND_apo	0.328587
IND_macdsignal	0.305091
IND_willr	0.29428
IND_aroonosc	0.292418
IND_cci	0.27104
IND_mfi	0.260748
IND_trix	0.255759
IND_minus_di	0.237174
IND_plus_di	0.215751
IND_slowd	0.2041
IND_aroondown	0.196236
IND_aroonup	0.195523
IND_ultosc	0.192543
IND_slowk	0.14975
IND_fastd	0.149648
IND_macdhist	0.110213
IND_dx	0.104631
IND_bb_width	0.077839
PAT_CDLLONGLINE	0.06347
IND_fastk	0.048899
cluster_up_minus_down_pct_change_30	0.047575
PAT_CDLBELTHOLD	0.045711
PAT_CDLCLOSINGMARUBOZU	0.03993
PAT_CDLADVANCEBLOCK	0.031649
PAT_CDL3OUTSIDE	0.029946
IND_adxr	0.029401
PAT_CDLHARAMI	0.018154
IND_200_minus_50_ema	0.017615
PAT_CDLDOJISTAR	0.016357
PAT_CDLMARUBOZU	0.01476
IND_atr	0.012035
PAT_CDLHARAMICROSS	0.009347
IND_bop	0.009211
PAT_CDLSTALLEDPATTERN	0.006451
PAT_CDLENGULFING	0.005971
PAT_CDL3INSIDE	0.005538
PAT_CDLINVERTEDHAMMER	0.005489
PAT_CDLHANGINGMAN	0.005431
PAT_CDLHIKKAKE	0.005339
is_wednesday	0.005162

Variable	**Score**
PAT_CDLDOJI	0.00502
is_tuesday	0.004539
PAT_CDLHOMINGPIGEON	0.004481
is_thursday	0.004389
PAT_CDLHAMMER	0.003927
is_monday	0.003912
PAT_CDLGAPSIDESIDEWHITE	0.003839
is_nzdusd	0.003829
IND_hour	0.003653
is_chfusd	0.003252
PAT_CDLXSIDEGAP3METHODS	0.003122
is_gbpusd	0.003122
PAT_CDLIDENTICAL3CROWS	0.003077
is_eurusd	0.002484
PAT_CDLSHORTLINE	0.002436
is_audusd	0.002391
PAT_CDL3LINESTRIKE	0.002351
PAT_CDLMATCHINGLOW	0.002348
PAT_CDLSPINNINGTOP	0.002332
PAT_CDLMORNINGSTAR	0.002324
PAT_CDLRICKSHAWMAN	0.002314
PAT_CDL3WHITESOLDIERS	0.002248
PAT_CDLMORNINGDOJISTAR	0.002109
PAT_CDLONNECK	0.002108
is_friday	0.001866
PAT_CDLSEPARATINGLINES	0.001705
PAT_CDLTAKURI	0.001643
PAT_CDLHIKKAKEMOD	0.001605
PAT_CDLCOUNTERATTACK	0.001507
PAT_CDLTHRUSTING	0.001472
PAT_CDLGRAVESTONEDOJI	0.001274
PAT_CDLEVENINGSTAR	0.00125
PAT_CDLLONGLEGGEDDOJI	0.00125
PAT_CDLUNIQUE3RIVER	0.001113
PAT_CDLEVENINGDOJISTAR	0.000952
PAT_CDLHIGHWAVE	0.000943
PAT_CDLKICKING	0.000903
PAT_CDLINNECK	0.000853
PAT_CDLSTICKSANDWICH	0.000546
PAT_CDLSHOOTINGSTAR	0.000457
is_usdjpy	0.000345
PAT_CDLTASUKIGAP	0.000288
PAT_CDL2CROWS	0.000275
PAT_CDLKICKINGBYLENGTH	0.00021
PAT_CDLUPSIDEGAP2CROWS	0.000137
PAT_CDLBREAKAWAY	0.000108
PAT_CDLDARKCLOUDCOVER	5.76E-05

Variable	**Score**
PAT_CDLRISEFALL3METHODS	0
PAT_CDLPIERCING	0
PAT_CDLMATHOLD	0
PAT_CDLLADDERBOTTOM	0
is_saturday	0
PAT_CDL3BLACKCROWS	0
PAT_CDLDRAGONFLYDOJI	0
PAT_CDL3STARSINSOUTH	0
PAT_CDLCONCEALBABYSWALL	0
PAT_CDLABANDONEDBABY	0

Table 7: Mutual Information Score for each Variable