Software Analysis and Design for Trading Cryptocurrency

A PROJECT REPORT

Submitted by
Ashish Ucheniya (18114014),
Ankit Aharwal (18114006),
Rahul Sahani (18114062)

for the fullfilment
of
CSN-400B: B.Tech. Project

under the guidance of
Dr. Sandeep Kumar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247667, INDIA

May 2022

Candidate's Declaration

We hereby declare that the work carried out in this dissertation entitled **"Software Analysis and Design for Trading Cryptocurrency"** is presented on behalf of partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science And Engineering submitted to the Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee under the supervision of **Dr. Sandeep Kumar**, Associate Professor, IIT Roorkee. The work presented in this report is authentic to the best of our knowledge and has been done from August 2021 to May 2022.

Date: January 18, 2025 **Place:** Roorkee - 247667

Ashish Ucheniya
(18114014)

Ankit Aharwal
(18114006)

Rahul Sahani
(18114062)

Certificate

This is to certify that the above statement made by the students is correct to the best of my knowledge and belief.

DR. SANDEEP KUMAR
ASSOCIATE PROFESSOR
DEPT. OF CSE
IIT ROORKEE
Date: January 18, 2025

Acknowledgements

In the accomplishment of completion of our project **"Software Analysis and Design for Trading Cryptocurrency"** we would like to convey our special gratitude to Dr. Sandeep Kumar, for his guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

His valuable guidance and suggestions helped us in various phases of the completion of this project. We will always be thankful to him in this regard

Abstract

Nowadays, the popularity of cryptocurrencies have increased among people. And with this rise in popularity more and more people have started to invest in different cryptocurrencies like bitcoin, etherium and many more. In recent years many cryptocurrency trading strategies are being used but it is a challenge to decide which one to choose because of the volatile nature of cryptocurrency.

So in this project, We analysed different traditional strategies and implemented some customized strategies using various machine learning and time based deep learning classification models using currency's close price, different standard technical and customized indicators as features to predict when to buy and sell any currency. We also implemented some optimization technique for finding best hyperparameters values of different strategies and framed closing price into different continuous classification. Finally we evaluated these strategies with different parameters like Sharpe Ratio and Profit Per Day and created visualizations to facilitate the results by making comparisons between the strategies. Our analysis showed that these different framed and optimized classification strategies have a significant improvement over those traditional strategies.

Contents

Chapter 1

Introduction

Cryptocurrencies have become a popular investing option in recent years as traditional fiat currencies are subject to security issues and strict monetary policies and banking regulations. On the other side, digital currencies provide decentralised, immutable, and anonymous alternatives. Cryptocurrencies is also affected by the automation as data becomes more freely available and machine learning models are being developed for trading. Despite their novelty, cryptocurrencies have experienced wide market acceptance and rapid development. Cryptocurrency-related assets are now included in the portfolios and trading strategies of several hedge funds and asset managers. Cryptocurrencies are on the rise and, as we all know, represent a very lucrative way to make money. As an emerging market and research direction, cryptocurrencies and cryptocurrency trading have seen considerable progress and a notable upturn in interest.

Cryptocurrencies market is a huge and highly volatile market. There are so many different currencies are available in the market and also there are so many different trading techniques are exposed to the trader that it become so difficult to chose which trading strategy we should consider for different currencies to get the higher profit.

So in this project, we will analyze and process historical price data to find the most suitable strategies for cryptocurrencies in general and find out which strategies work better than others to achieve best results. We will also implement many different strategies, such as traditional, machine learning-based, and deep learning-based models. And we will then add additional features and optimization techniques to further improve performance.

1.1 Objective

- The objective of this project is to find an effective way to compare different trading strategies for cryptocurrencies and find out which of the strategies are generally the best.
- Further analysing and implementing traditional strategies (SMAC, EMAC, RSI, MACD and BUYN-HOLD) and some customized strategies using various machine learning classification models (Logistic Regression, Random Forest, SVC, KNN, Neural Network, etc.)
- Also implement time-based deep learning models (LSTM, RNN and GRU) with implementation of indicators (Altcoin Season Indicator) and trading indicators (rsi, macd, ma and ema) as features in data for training and framing different classifications to improve training results of machine learning and deep learning.
- Finally evaluating these strategies with different parameters like Sharpe Ratio and Profit Per Day, creating visualizations to facilitate the results by making comparisons between the strategies and finding the best strategies in general with help of different results.

1.2 Problem Statement

Cryptocurrencies are digital decentralized currencies, which use different cryptographic techniques to secure themselves from double spending and counterfeiting. Being decentralized makes it very hard for central authority manipulation. Fiat currencies like Dollar and Rupee are devalued by central authority and are subject to their loss in value over a long time. Cryptocurrency are in trend due to their capability to give big returns in small amounts of time. So, those characteristics make Cryptocurrencies really attractive for an investor or a trader.

This emerging market of crypto has led to use of many trading strategies and many new cryptocurrencies. And choosing which strategy would work best for a particular crypto is a tough task to handle. Also now there is an abundance of data of many cryptocurrency trade histories which could be utilized to gain valuable information about the performance of a cryptocurrency on using different trading strategies. And also can very well be used to predict the variation of the price of a crypto till some extent.

Therefore, we aim on backtesting the historical price data of many cryptocurrencies with different traditional techniques, machine learning models and some deep learning models in order to find some general conclusions about the performance of the strategies.

1.3 Organization

Our rest of the report is organized as follows. In chapter 2, we discuss the previous work that has been done in this area. In chapter 3, we explain the complete methodology of our project and elaborate each different techniques and algorithm that we have used in this project. In chapter 4, we show the different experimentation results that we achieved. Finally in chapter 5, we discuss the overall conclusion of this project.

Chapter 2

Literature Survey

2.1 Literature Review

So much work has been published recently in the different aspects of crypto currency like its price formation, price fluctuations and forecasting, systems dynamics etc such as Marco Ortu [9] focused on predicting the price movements of two major crypto currencies Bitcoin and Ethereum with four different DL algorithms Multi Layers Perceptron (MLP), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Attention Long Short Term Memory (ALSTM) by using past prices, trading indicators and users sentiments as features. Their results showed that the combination of past prices, trading indicators and social indicators as features outperformed the cases in which we used only one of them as features.

Ross C. Phillips [19] attempts to predict crypto currency bubbles using a hidden Markov model previously used to detect flu epidemics with online social media indicators. Their work shows the utility of HMM in identifying financial bubbles and also how social media can be valuable features for predicting price movements in crypto currencies.

For crypto currencies price prediction, Patel Jay [15] suggested a stochastic neural network approach. The random walk theory, which is commonly utilised in financial markets, underpins this technique. To replicate market volatility, this suggested model introduces layer wise randomness into observed feature activations of neural networks, resulting in better prediction than deterministic models.

Erdinc Akyildirim [8] tried to predict the twelve most liquid crypto currencies at daily and minute levels using machine learning classification algorithms with having past price information and technical indicators as model features. They had consistently above the 50% for their various classification models for all crypto currencies.

Jethin Abraham [14] focus on presenting a method for predicting the price movements of the two major cryptocurrencies, Bitcoin and Ethereum, using Twitter and Google Trends data. They found that tweet volume, rather than tweet sentiment, predicts price direction. A person would be better informed to make buy and sell decisions regarding these two currencies by using this model.

McNally [18] tried to predict Bitcoin trends by using Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) by make classification model. He used only past prices and several technical indicators such as the simple moving average for prediction. He got around 51-52% accuracy for predicting changes in the direction of the Bitcoin trend.

Manish Agrawal [3] focused on building up an Evolutionary Deep Learning Model (EDLM) to identify stock trends' prices by using STIs. To establish the concept of Correlation-Tensor, this model also implemented Deep learning model. The proposed showed significant improvement over normal ML models.

V. Derbentsev [10] discussed the problems of short-term forecasting of crypto currency time series using a supervised machine learning approach. They applied Random forests and Stochastic Gradient Boosting machine models in their approach.

2.2 Observations

Cryptocurrencies is a rapid growing market. It have experienced broad market acceptance and fast development. A lots of research work has been done and much work is currently being done in this field. After going through different research work we observed that relative to the stock market there hasn't been much work done in the crypto market as it is much recent technology and it is much more volatile and show more price fluctuations than stock market. Most of the work for crypto currencies price movements prediction uses social news, people sentiments as inputs features instead of currencies price history and technical indicators values and also most work only focus on some top crypto currencies like Bitcoin and Ethereum. There is not any generalized work for all crypto currencies. These is a lot of work on prediction, Generally these classification models only predict the price movements or its trends whether the price will go up or down but they don't give the general idea of when to buy, sell or hold any crypto currency and also the accuracy of these different prediction models are not that good but many new techniques and work is being done to improve the accuracy further more.

Chapter 3

Methodology

3.1 Workflow of our project

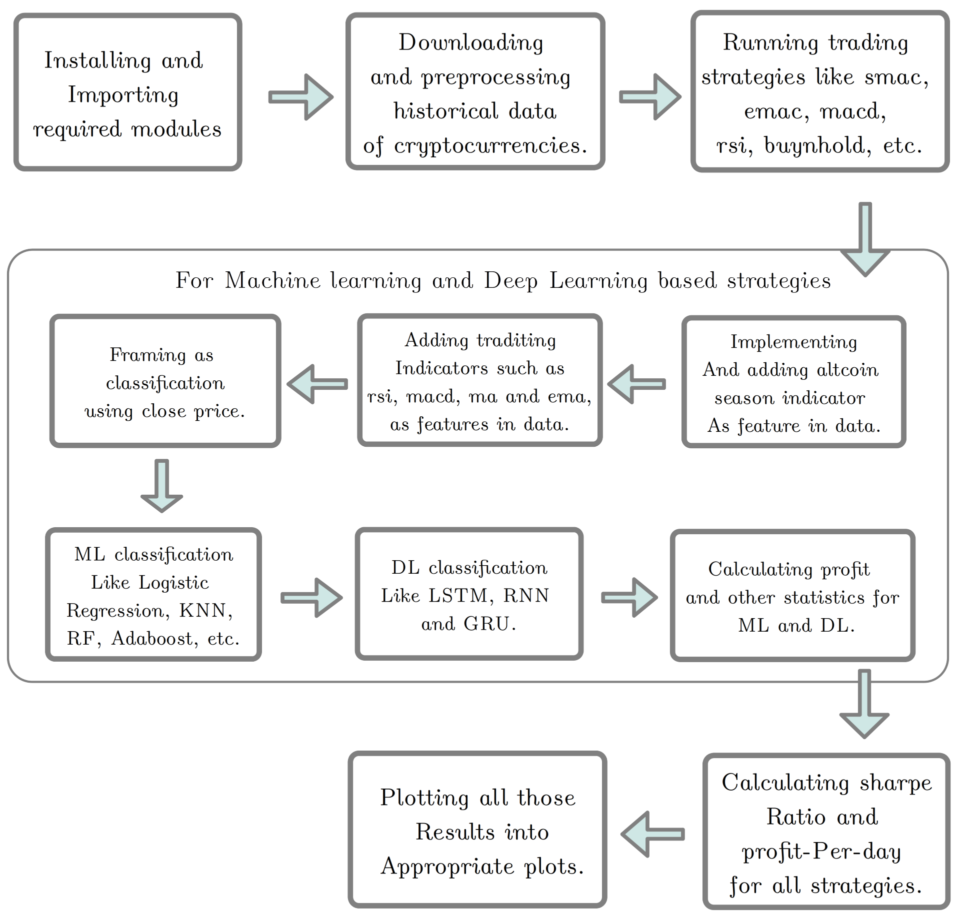

Figure 3.1: Complete flow of our project

The above chart explains the workflow of this project.
First installing required modules (like BeautifulSoup, html_table_extrator and fastquant) and Importing modules (like numpy, pandas, matplotlib, sklearn, keras, etc). Downloading and preprocessing historical data of cryptocurrencies. First scrapping cryptocurrency trading pair names (like BTC/USDT, ETH/USDT, etc). After that downloading the historical data for a particular timeframe for all the cryptocurrency pairs.
Then Running trading strategies like SMAC, EMAC, MACD, RSI and BUYNHOLD using the help of fastquant library. Those strategies are run with combinations of hyperparamters like for SMAC(fast_period and slow_period).
Then Implementing and adding altcoin season indicator (ASI) as a feature in data. In the Implementation, We First calculate the percentage increase of each altcoins. Then rank them based on the percentage increase. After that we standardized the rank. After that Adding trading indicators(such as rsi, macd, ma and ema) on features(close, volume, PCT and ASI) on different periods(1,2,3,4 and 5 days) in data. Then Framing into consecutive classification using close price. Rather than only using local tops and local bottoms, we tried to make use of global tops and bottoms for it. After that Running Machine Learning classification like Logistic Regressing, KNN, Random Forest(RF), Adaboost, Support Vector Classifier(SVC) and Neural Networks(NN) on framed classifications using sklearn. Then Running Deep Learning classification like Long Short-Term Memor(LSTM), recurrent neural network(RNN) and Gated recurrent units(GRU) on framed classfications using keras. Calculating profit for Machine Learning and Deep Learning strategies using the help of fastquant.
After that for evaluation calculating Sharpe-Ratio and Profit-Per-Day for all strategies. Finally, Plotting all those results into appropriate plots and visualize comparison.

3.2 Traditional strategies Analysis

Traditional strategies are the strategies which uses the different values of the technical indicator like **Simple Moving Avarage(SMA), Exponential Moving Avarage (EMA), Relative Strength Index (RSI), Moving Average Convergence/Divergence indicator (MACD)** etc to get the buy and sell signals and do trading based on those signals.

Stock Technical Indicators (STIs)	Mathematical Formula
Moving Average (MA) of n days	$\frac{1}{n}\sum_{i=1}^{n} C_i$
Exponential Moving Average (EMA)	$EMA_t = C_t \left(\frac{2}{T+1}\right) + EMA_{t-1}\left(1 - \frac{2}{T+1}\right)$
Moving Average Convergence Divergence (MACD): most common is 12/26 MACD	$MACD = [(12 - day\ EMA) - (26 - day\ EMA)]$
Relative Strength Index (RSI)	$RSI = 100 - \left[\left(\frac{100}{1+\left(\frac{AG}{AL}\right)}\right)\right]$

Figure 3.2: Technical Indicator Mathematical Formula

We ran all these Traditional strategies by using the *Fastquant.backtest*() python module and for each different strategy we made a line chart which tells us about the closing price of any particular crypto currency and it also shows the different buy and sell points based on the values of different indicators. Below we have shown a table containing the details related to all technical indicators and the different hyperparametes combinations that we used in making the line chart and after that we have shown the line chart for the

SMAC-Simple moving average crossover strategy. In the chart We can see the closing price of Bitcoin (BTC) and the volume that traded on each day. We also can see different buy and sell based the cross over of the fast and slow period SMA values. We have shown the line chart for the all remaining strategies after this.

Table 3.1: Traditional Strategies with different hyperparameters combinations

Model	**Hyperparameters combinations**	**About**
SMAC	l_fast_period=[2,3,5] l_slow_peroid=l_fast_period*[2,3]	Simple moving average crossover: If the fast SMA crosses the slow SMA, there is a trading signal. If the fast SMA is above it, buy, else sell.
EMAC	l_fast_period=[2,3,5] l_slow_peroid=l_fast_period*[2,3]	Exponential moving average crossover: If the fast EMA crosses the slow EMA, there is a trading signal. If the fast EMA is above it, buy, else sell.
MACD	default	Moving Average Convergence Divergence: If the MACD crosses a signal, there is a trading signal. If the MACD is above it, buy, else sell.
RSI	default	Relative Strength Index: If RSI falls below a certain threshold, buy, and if RSI rises above a certain threshold, sell.
BUYNHOLD	default	Buy and Hold: Buy at start and never sell

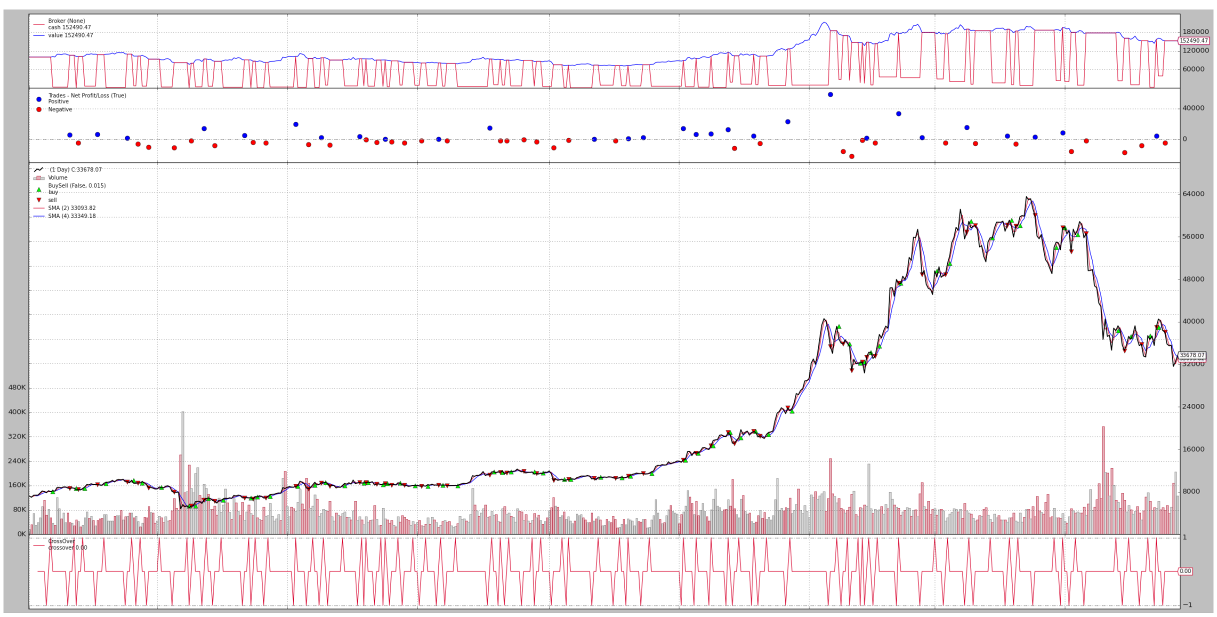

Figure 3.3: Running SMAC Strategy

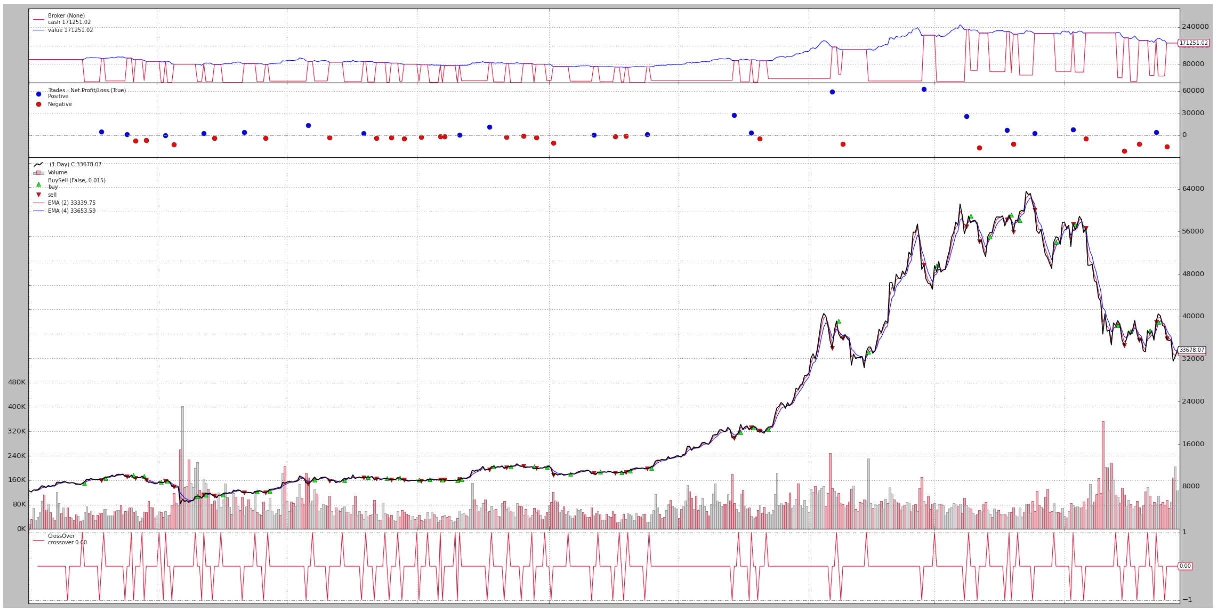

Figure 3.4: Running EMAC Strategy

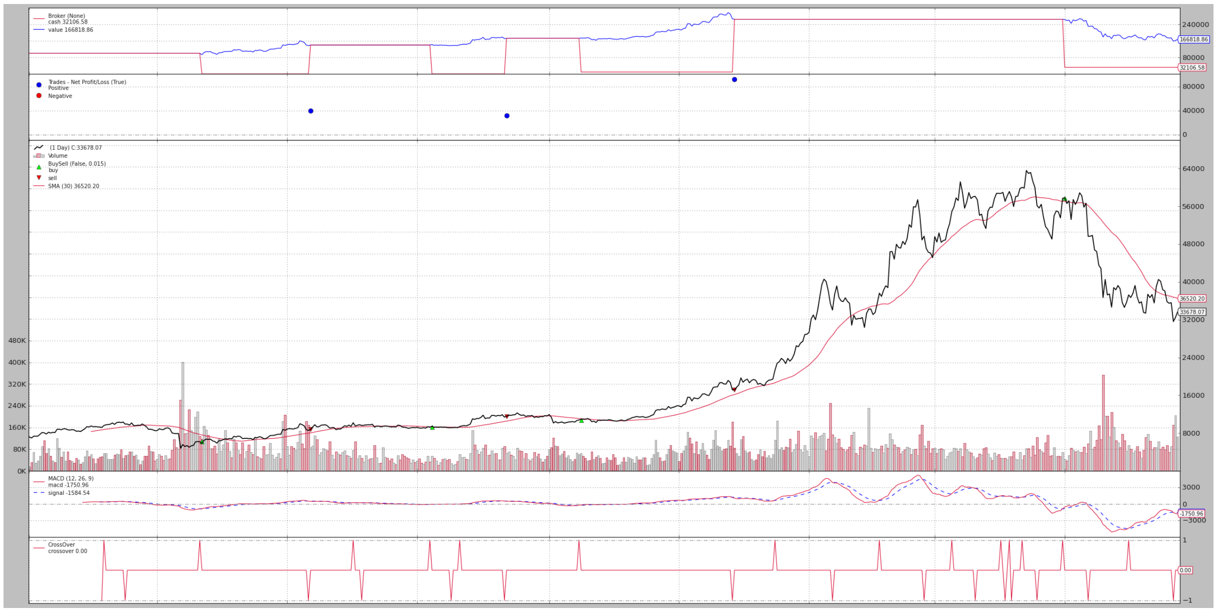

Figure 3.5: Running MACD Strategy

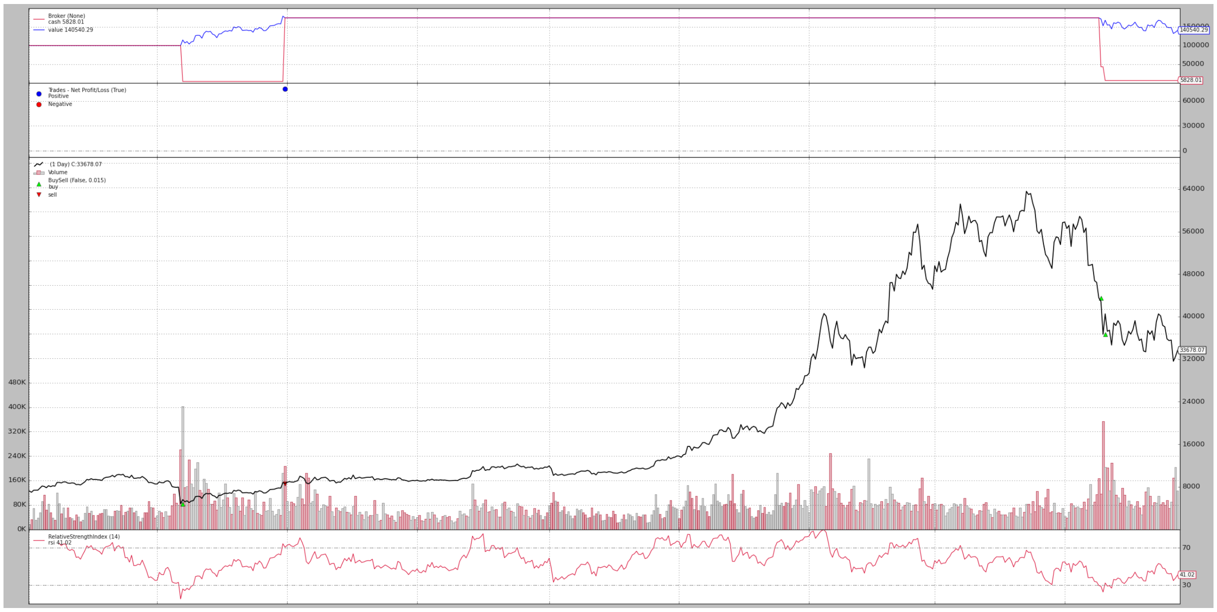

Figure 3.6: Running RSI Strategy

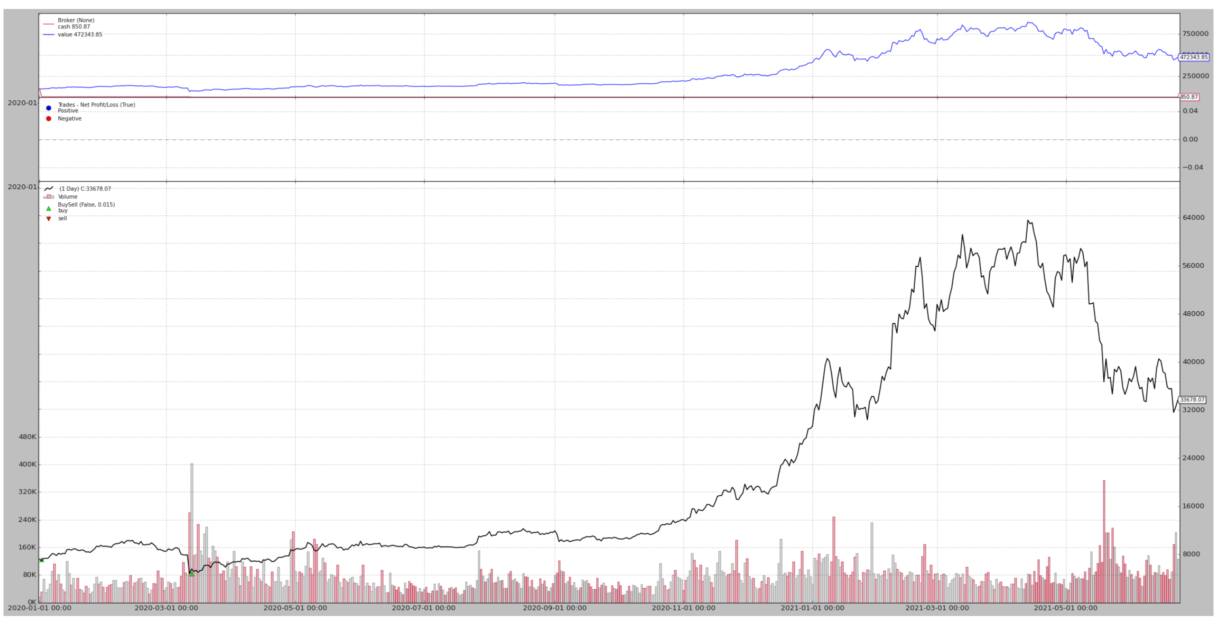

Figure 3.7: Running Buynhold Strategy

3.3 Hyperparameter Optimization For All Strategies

This technique basically means we have used strategies with multiple hyperparameter values. For example in the SMAC strategy we had two hyperparameters($fast\_period$ and $short\_peroid$). So, we used some combination of $fast\_period$ and $slow\_period$. So, in comparison the better parameters will automatically come at top.

In this particular case of SMAC $fast\_period = 5$ and $slow\_period = 15$ is the best hyperparameter.

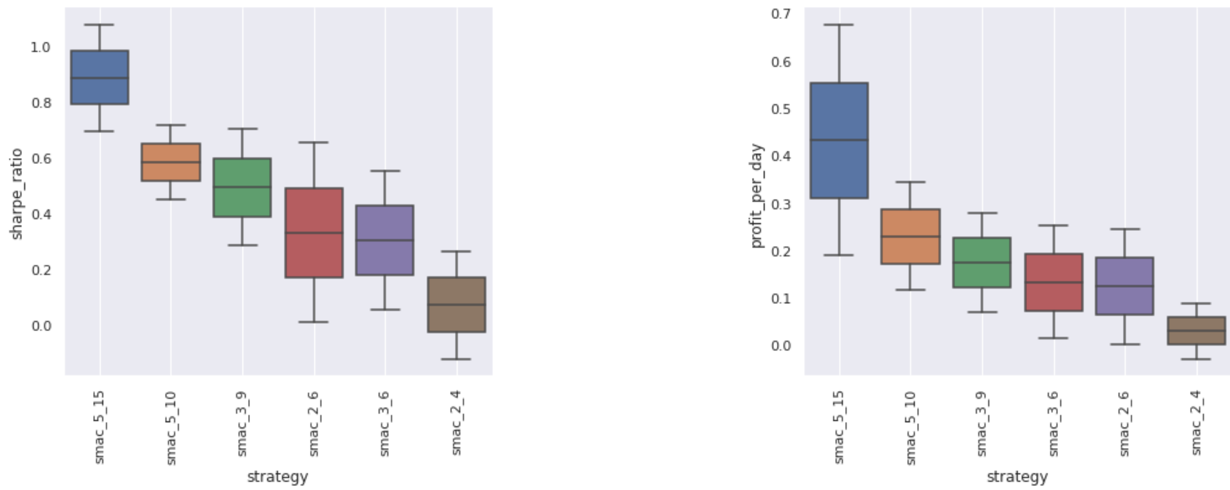

Figure 3.8: Comparison of different Hyperparameters combinations for SMAC

3.4 Machine Learning Based Strategies Analysis

Our each Machine learning strategy is based on a Classification model in which we first calculated the local minima and local maxima for each cryptocurrency in given dataset interval. These local minima and local maxima were used as class label for our classification model. Local minima represent the buy signal, Local maxima represent the sell signal and all other represent hold signal. We considered currency's past price, Technical indicator like sma, ema, rsi, macd etc and custom indicator like altcoin season with different hyperparameters values as features to our classification models. We used the sklearn library to implement our ML models.

We divided the dataset into some parts. Then each time we tested the model with 1 part and trained the model with the rest of the part. Then finally find out the profit-per-day and sharpe-ratio for the tested prediction for each classification model.

Table 3.2: Machine Learning Strategies with different hyperparameters combinations

Model	**Hyperparameters combinations**	**About**
KNN	"n_neighbors":[3,5,10]	It classifies the data point according to how its k nearest neighbors are classified.
LogisticRegression	"C":[10,100]	It is a linear classification model. It uses sigmoid as the activation function.
RandomForest	"criterion":["gini","entropy"]	Bagging ensemble of decision trees.
Adaboost	"learning_rate":[1]	Boosting ensemble of decision trees.
SVC	"kernel": ["linear","rbf","sigmoid"]	Support Vector Classifier provides a best fitting hyperplane that categorizes our data.
NN	"hidden_layer_sizes": [(50,50,5),(50,5)]	A Neural Network is a network of perceptrons connected in a layered structure. It has an input layer, a hidden layer, and an output layer. It also has non-linear activation functions after each layer such as Relu or Softmax.

In the above table, Hyperparameter combination refers to the different hyperparameters we used in the models, in order to obtain the best configuration.

We ran all Machine learning based strategies by using the Fastquant.backtest() python module and made a line chart for each strategies. Below we have shown the line chart for the KNN classification model based strategy. In the chart, We can see the closing price of Bitcoin (BTC) and the volume that traded on each day. We also can see different buy and sell point represented with green and red color respectively based on the prediction of the KNN classification model. We have shown the line chart for the remaining ML based strategies after this.

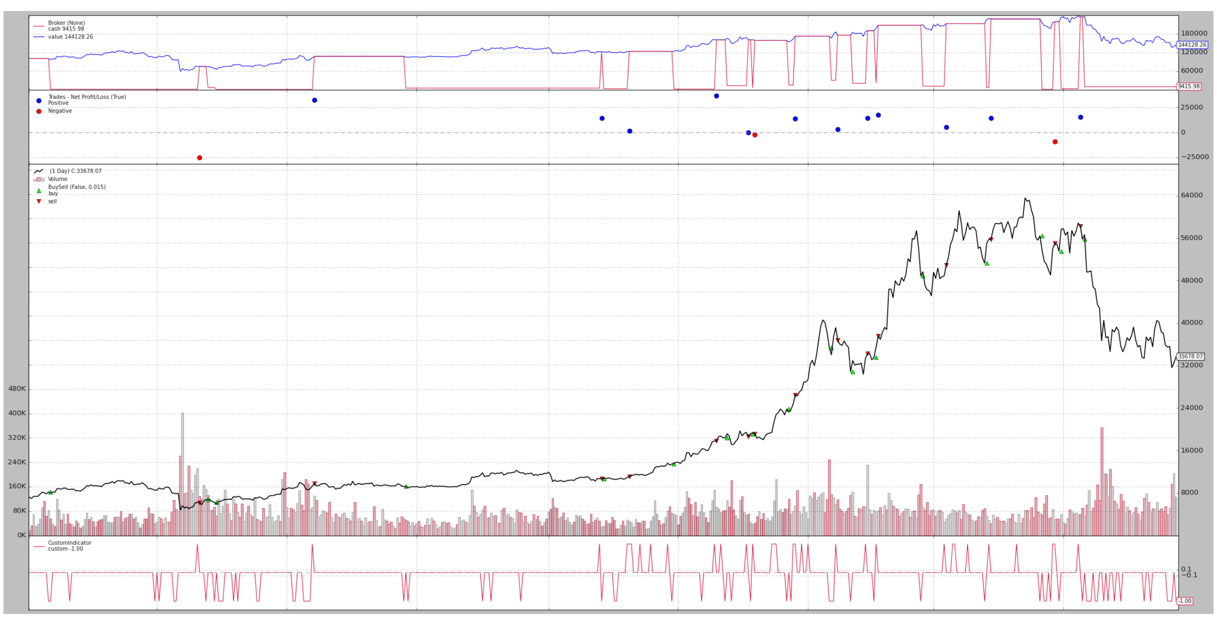

Figure 3.9: Running K-Nearest-Neighbour Strategy

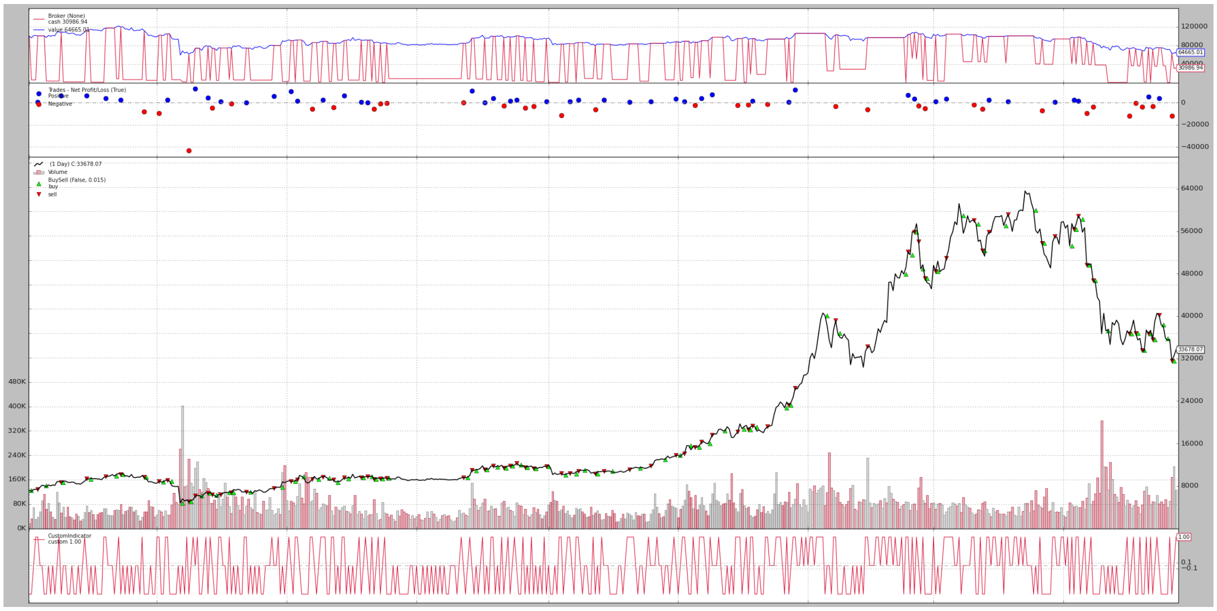

Figure 3.10: Running Logistic Regression Strategy

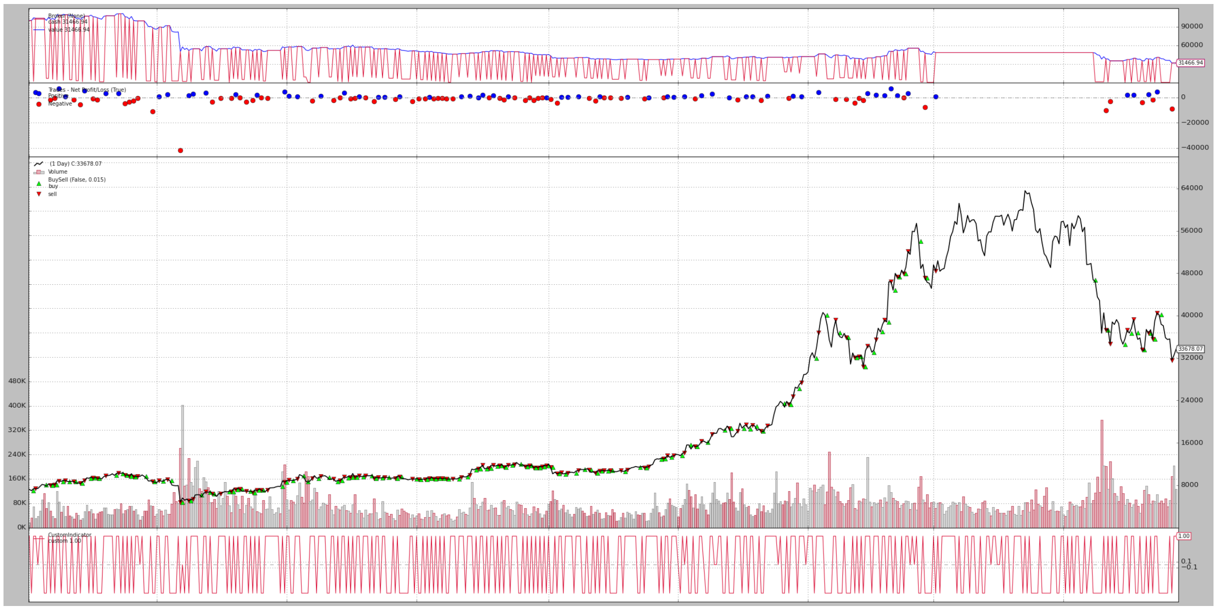

Figure 3.11: Running Neural Network Strategy

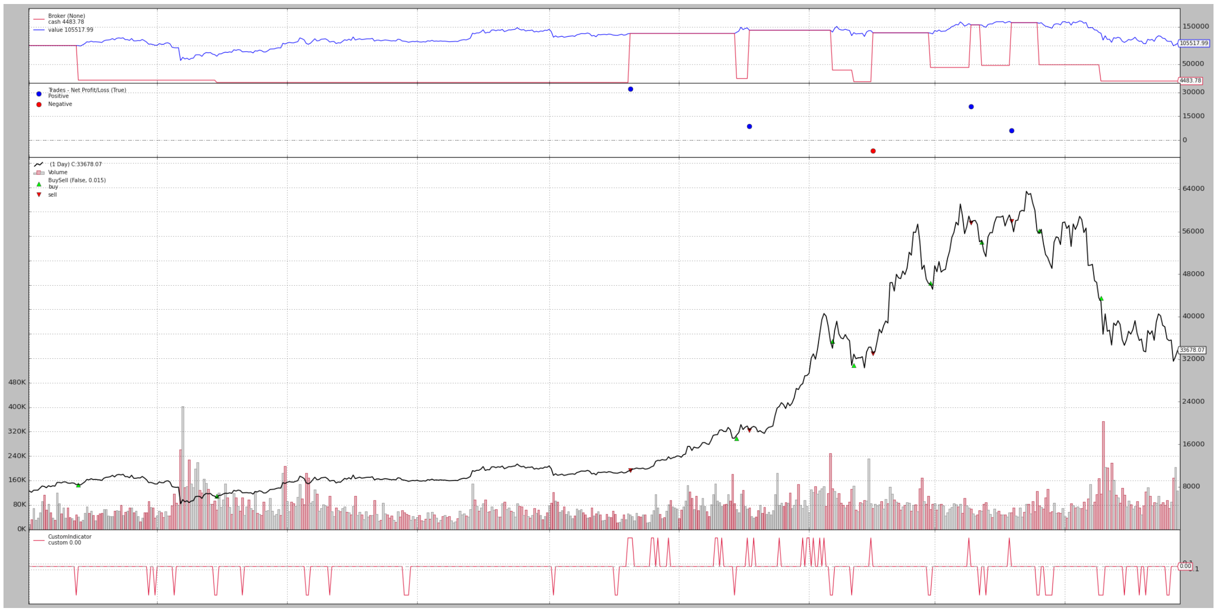

Figure 3.12: Running Random Forest Strategy

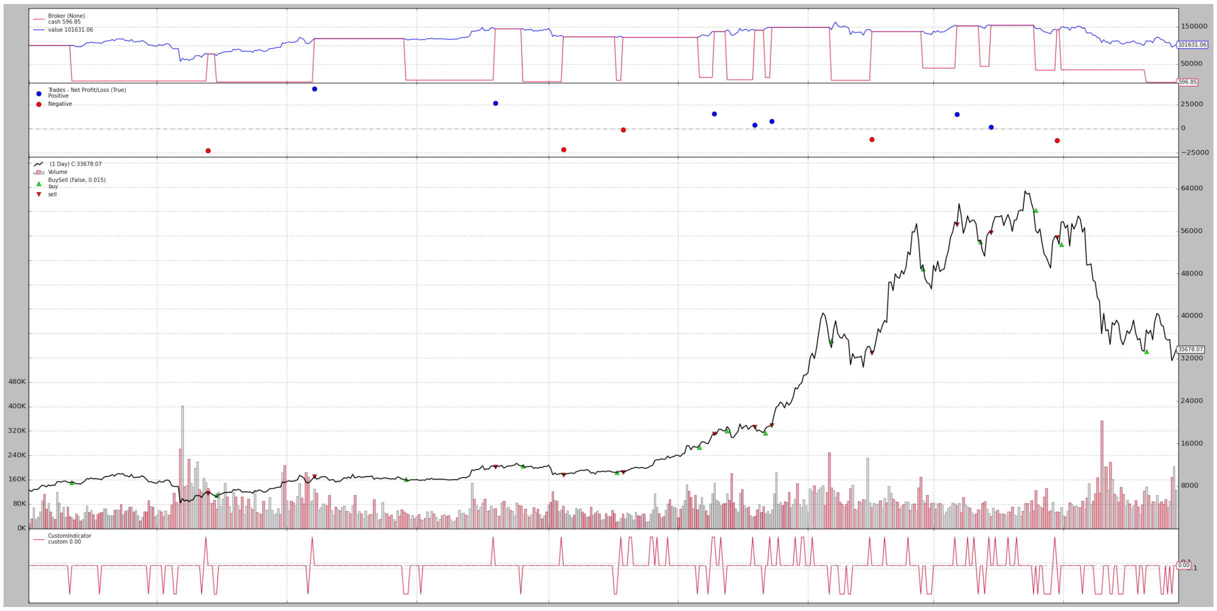

Figure 3.13: Running Support Vector Classifier Strategy

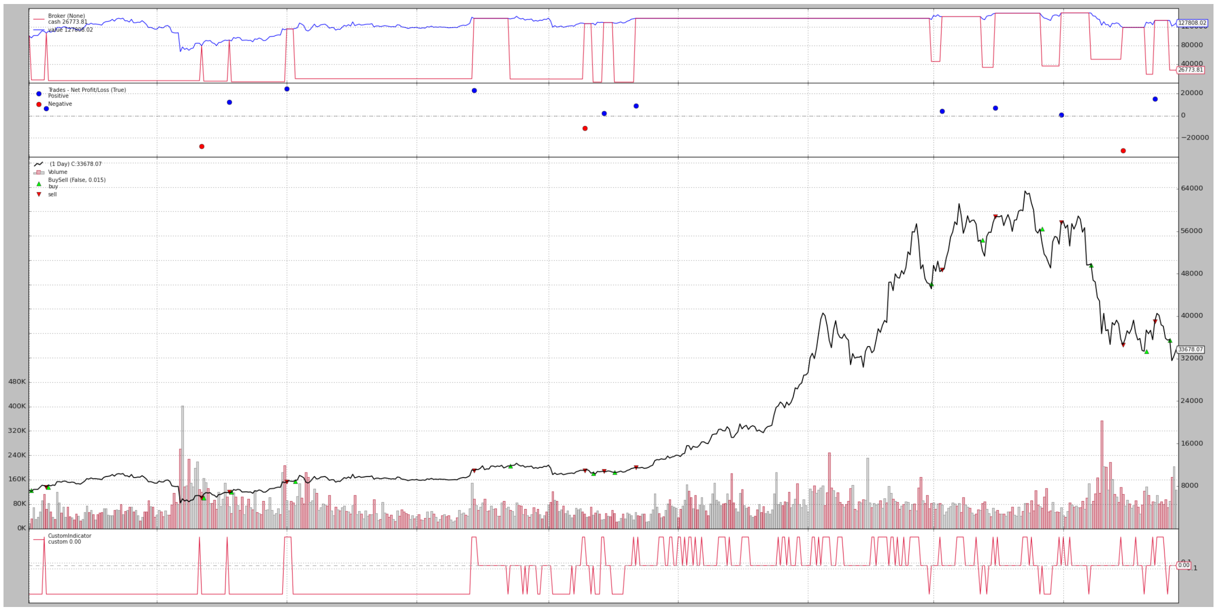

Figure 3.14: Running Adaboost Strategy

3.5 Feature Importance

Since we are using Random Forest we can also find which features are being used for prediction. Here we can see the feature importance for a classification. There were too many features so we decided to only show top and bottom 5 of the feature importance.

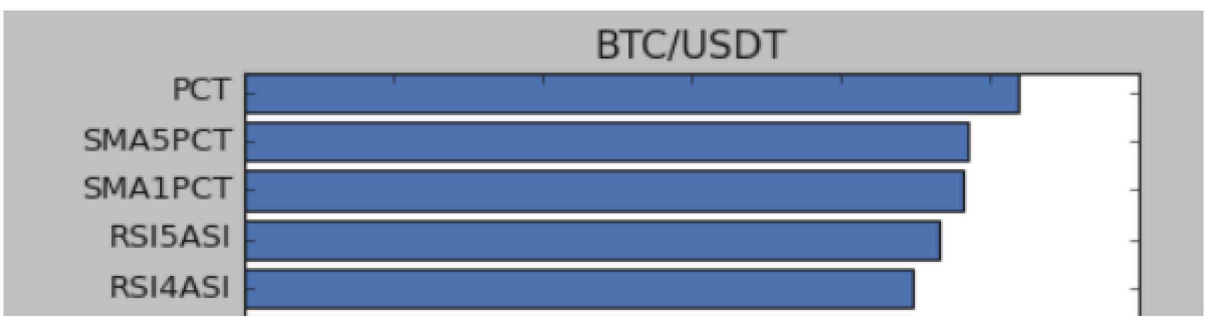

Figure 3.15: Top 5 Most Important Features

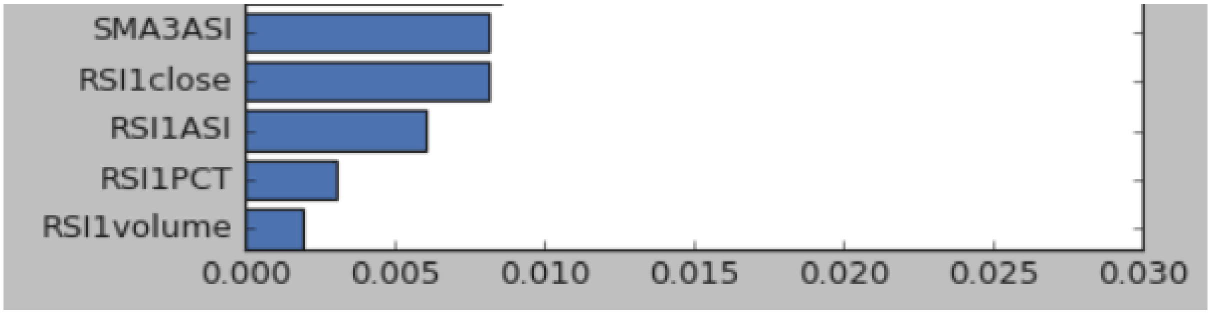

Figure 3.16: Bottom 5 Least Important Features

3.6 Deep learning based strategies Analysis

We implemented three Deep learning models Long Short-Term Memory (LSTM), recurrent neural network (RNN) and Gated recurrent units (GRUs) using python keras library.

Table 3.3: Deep Learning(sequence based) Strategies with different hyperparameter combination

Model	Hyperparameters combinations	About
LSTM	"sequence_length":[5]	Long short-term memory networks are a variant of RNNs that are able to learn long-term dependencies in sequence prediction.
RNN	"sequence_length":[5]	Recurrent neural networks are deep learning based models capable of learning sequence-based dependencies.
GRU	"sequence_length":[5]	Gated recurrent units are basically like LSTM with a forget gate. They have fewer parameters than LSTM and do not have an output gate.

Same as ML strategies, Our each DL strategy is also based on a classification model which predict different buy, sell or hold signals. The line charts for each different DL strategies which we made using Fastquant.backtest() python module are shown below.

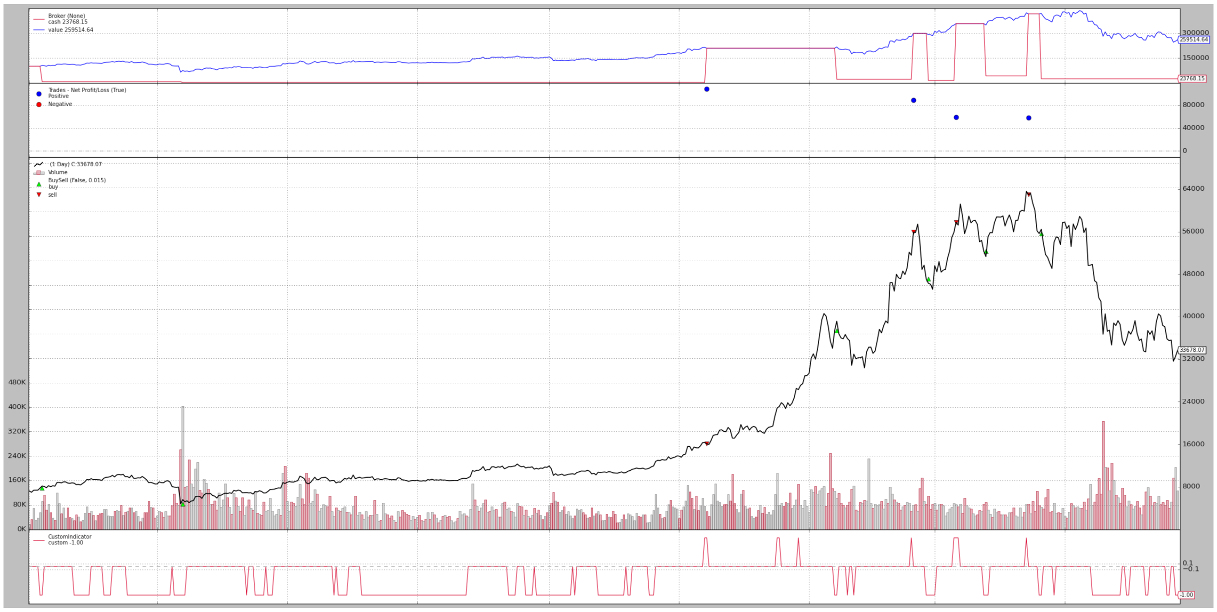

Figure 3.17: Running Gated recurrent units (GRUs) Strategy

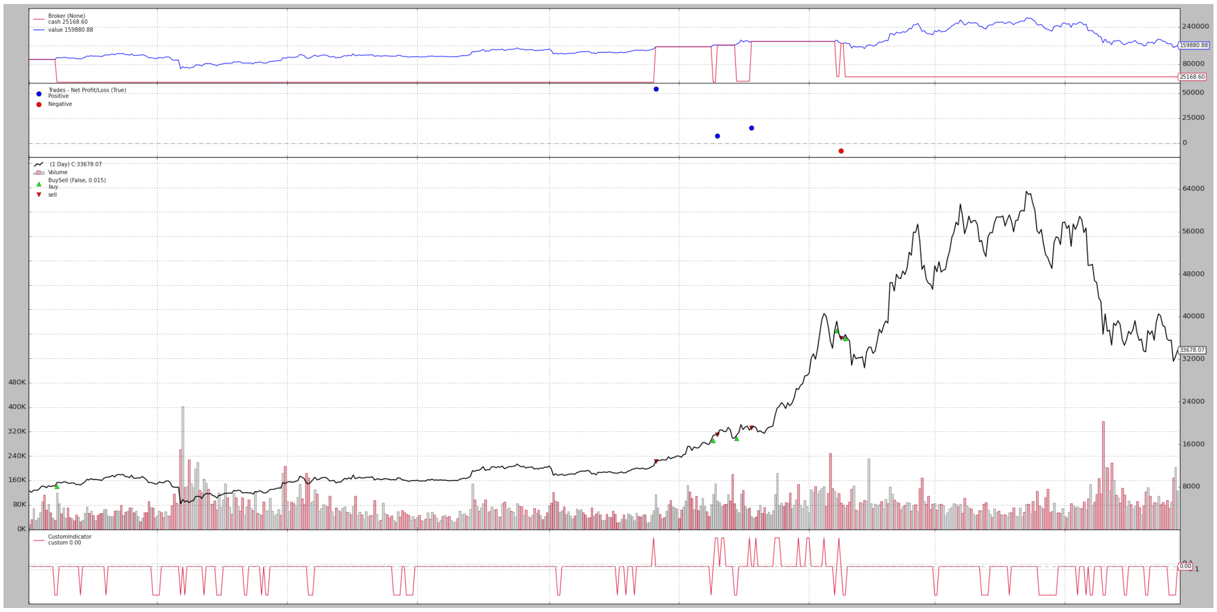

Figure 3.18: Running Long short-term memory (LSTM) Strategy

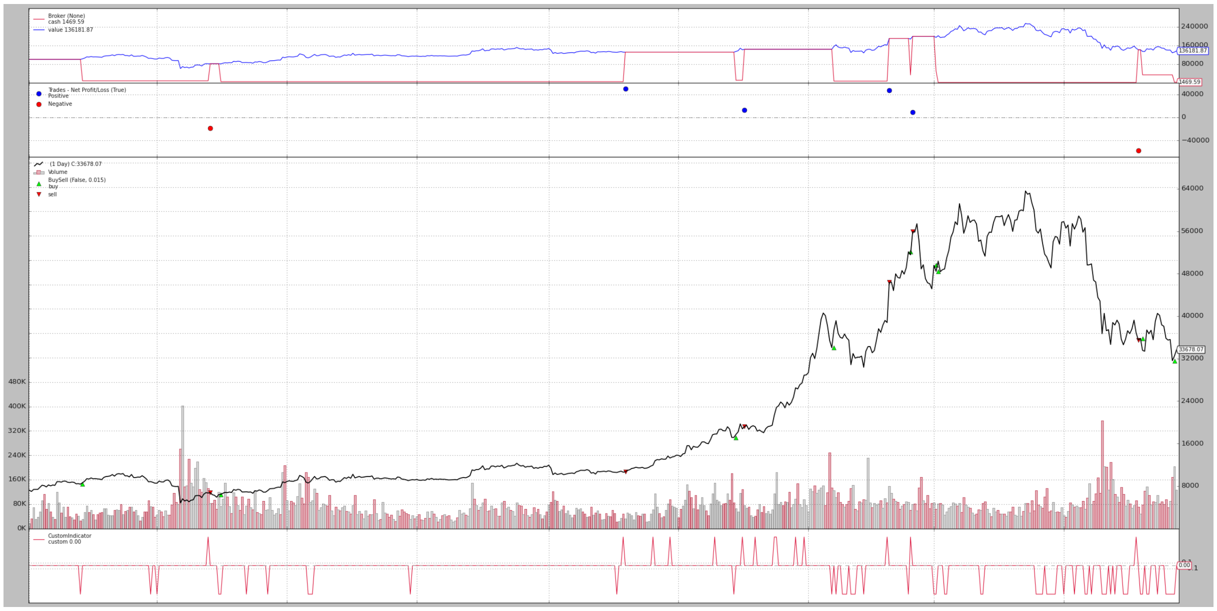

Figure 3.19: Running recurrent neural network (RNN) Strategy

We can see in all the above charts that when we are buying the crypto currency and when we are selling it based on the our different deep learning classification models prediction.

3.7 Evaluation metrics

3.7.1 Profit-Per-Day

The Profit-Per-Day is a measure of positive return which is adjusted by dividing by the number of days of running strategy.

$$\text{Profit Per Day} = \frac{R_p}{N_d}$$

Where,
R_p = Return of portfolio
N_d = Number of Days of running strategy

3.7.2 Sharpe Ratio

Y Liu and CR Harvey [7] showed that how Sharpe Ratio can be used to evaluate different trading strategies. The Sharpe Ratio is the measure of the return which is risk adjusted of a trading strategy. A strategy with a higher Sharpe ratio is considered better than its peers.

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

Where,
R_p = Return of portfolio
R_f = Risk-Free rate
σ_p = Standard deviation of portfolio's excess return

3.8 Altcoin Season Indicator

We Implemented and Integrated an indicator called the altcoin season indicator for all the cryptocurrencies. It can be used as a feature for machine learning strategies.

In the Implementation, We First calculate the percentage increase of each altcoins. Then rank them based on the percentage increase. After that we normalized the rank (using standard scaling) hence it is of mean 0 and std 1. If value is high then the altcoin increases the highest and if value is low then altcoin increases lowest on a particular day.

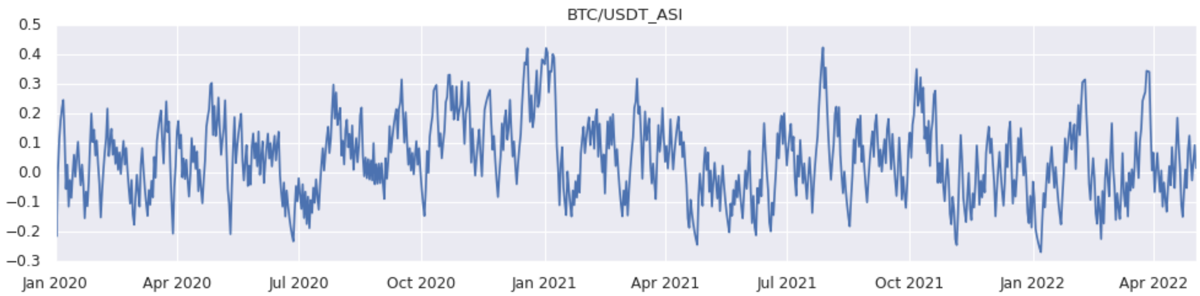

Figure 3.20: Altcoin Season Indicator

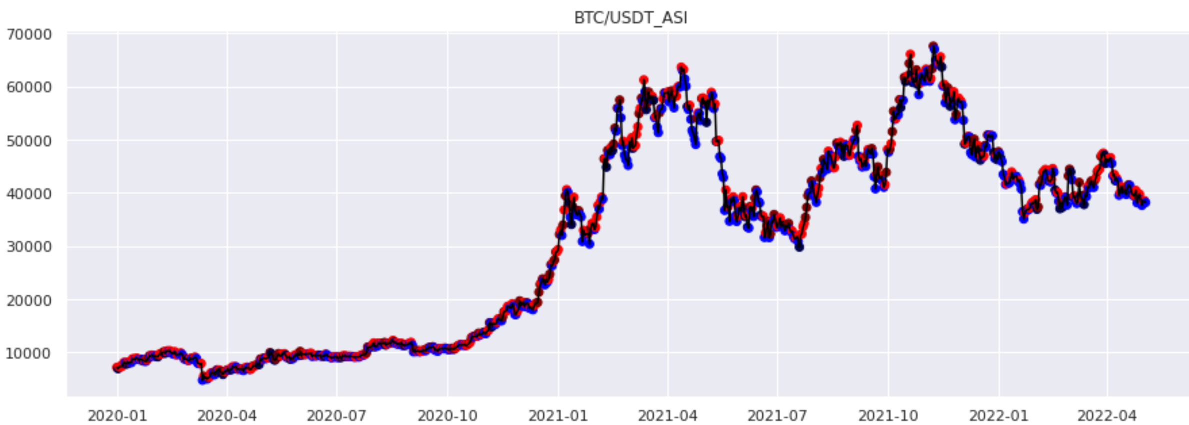

Figure 3.21: Price Graph with Altcoin Season Indicator

The above graph is the price of BTC/USDT. The redder portion means value of altcoin-season-indicator is high and bluer portion means value of altcoin-season-indicator is low.

3.9 Framing into consecutive classifications

Earlier we used local tops and local bottoms in closing price as sell and buy signals but this approach was not that realistic and was not giving better results, So Rather than only using local tops and local bottoms we tried to make use of global tops and bottoms for it.

In the implementation, We first smooth the close price using simple moving average.. But since smoothing is a lagging indicator it would not give appropriate results. To solve that, we apply the same smoothing in the opposite direction also. Then, find the top and bottom. Finally, we define top, and (its k neighbors) as the sell signal and bottom, and (its k neighbors) as the buy signal. We can use different values of k for finding different results.

We currently only used:

- Local top and bottom (local)
- k=1 days (local1
- k=4 days (local4)

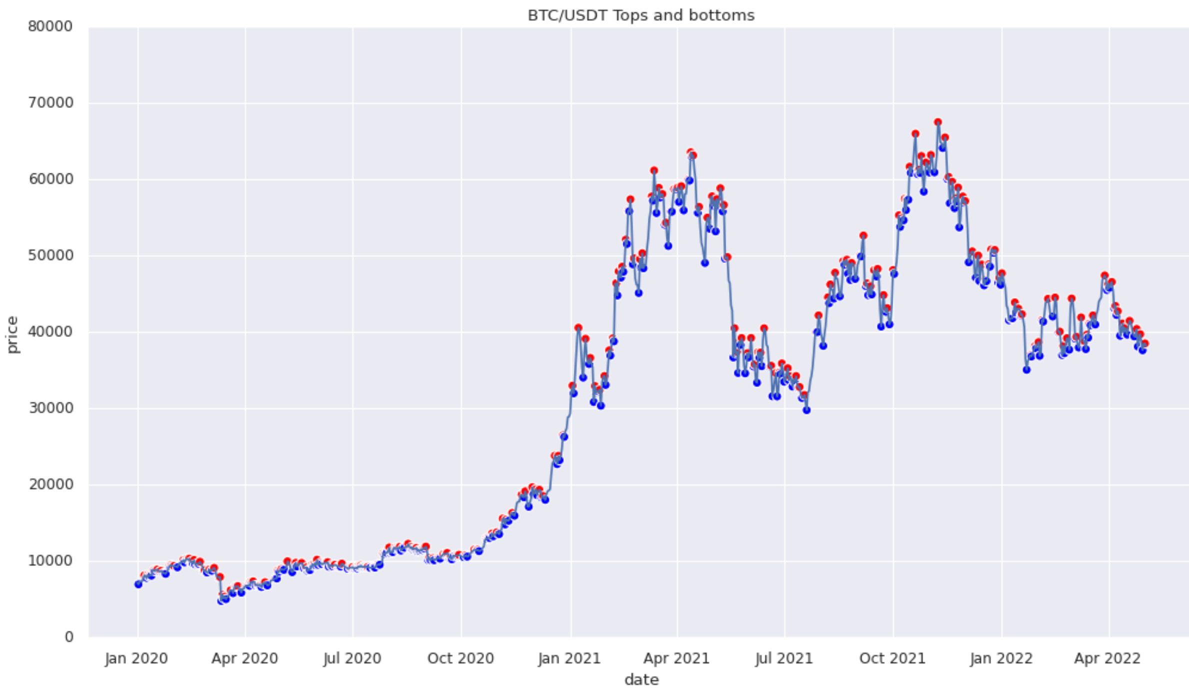

Figure 3.22: Local top and bottom (local)

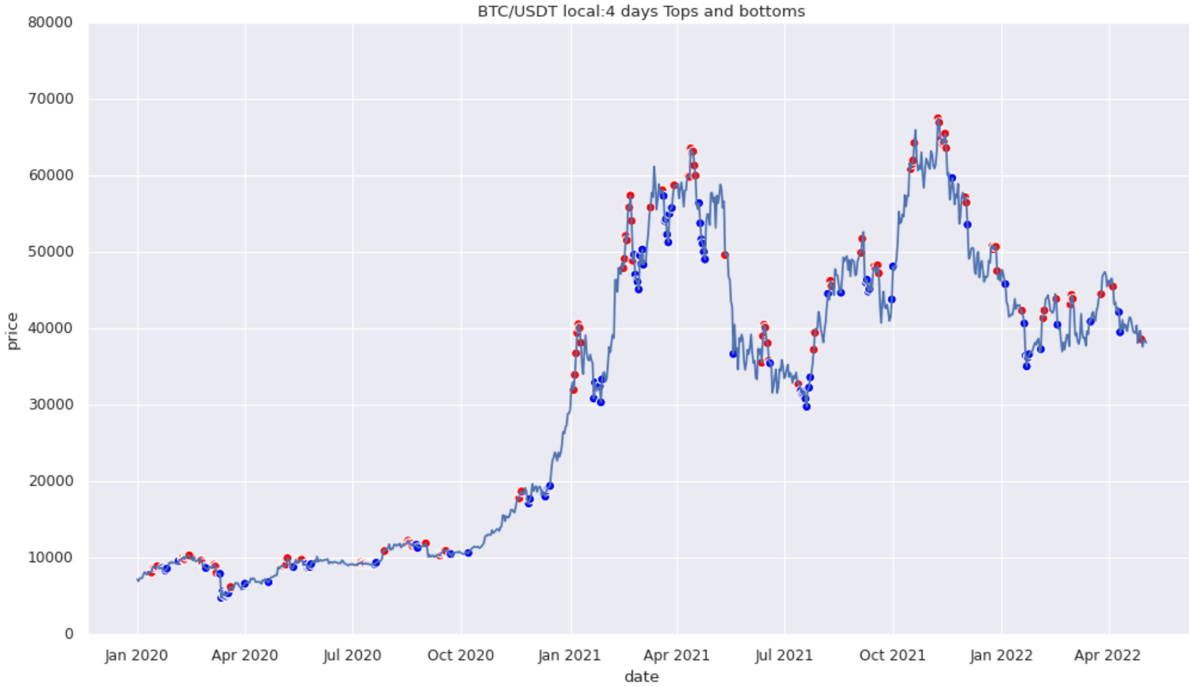

Figure 3.23: k=4 days (local4)

The above two graphs are the price of BTC/USDT(a random altcoin), and the blue and sell dots represent buy and sell signals respectively. As we can see in local the buy and sell signals are quite discrete. And in local4 techniques the buy and sell signals are very consecutive.

Chapter 4

Experimentation And Results

4.1 Experimental Setup

Google colab: Colab allows anyone to write and run arbitrary Python code(.ipynb) in the browser, and is particularly well suited for machine learning, data analysis, and education.

4.1.1 Installing and Importing required Modules

We installed following three modules:

- BeautifulSoup: Library to handle the "https://coinmarketcap.com/exchanges/" html page and its content.
- html_table_extractor: Finding the names of trading pairs on different exchanges. From there we got the names of different trading pairs for USDT.
- fastquant: For downloading the historical data for different trading pairs and for running traditional trading strategies, etc.

We imported following modules:

- For data handling: numpy and pandas.
- For visualizations: matplotlib and seaborn.
- For trading strategies and machine learning models: fastquant, scikit learn and keras.
- For data collection: BeautifulSoup, html_table_extractor and fastquant.
- Others: os (provides functions for interacting with the operating system), time (getting current date), datetime (time related feature handling), requests (requesting url).

4.1.2 Data collection

- First we got a list of 5 trading pairs from "https://coinmarketcap.com/exchanges/" using library requests, BeautifulSoup and html_table_extractor. ex:

  l_all = ["BTC/USDT", "APE/USDT", "LUNA/USDT", "ETH/USDT", 'GMT/USDT"].

- Now, we need to download the historical trading data for those trading pairs. For that we will use $fastquant.get\_crypto\_data()$.

  The timeframe for data should be from 2020-01-01 to 2022-05-03(today date). If the data is not available for this timeframe then the trading pair is removed. ex:

  l_remove = [”APE/USDT”, ”LUNA/USDT”, ”GMT/USDT”]

  l_final = [”BTC/USDT”, ”ETH/USDT”].

- number of trading pairs: len(l_final) : 2
  Originally each trading pair of our dataset had 854 observations(days) and 6 features.
  dt(index): date
  open: open price of that day
  high: highest price of that day
  low: lowest price of that day
  close: close price of that day
  volume: volume trade of that day

4.1.3 Data preprocessing

From close price we obtained “PCT” which means percentage change in price.
We also have added “ASI” which means altcoin season indicator that we have mentioned above.
By taking SMA(simple moving average), EMA(exponential moving average), RSI(relative strength indicator) of features(close, volume, PCT and ASI) in different periods(1,2,3,4 and 5 days).
Hence, we got 3*4*5=60 new features for training of Machine learning and deep learning models. ex:

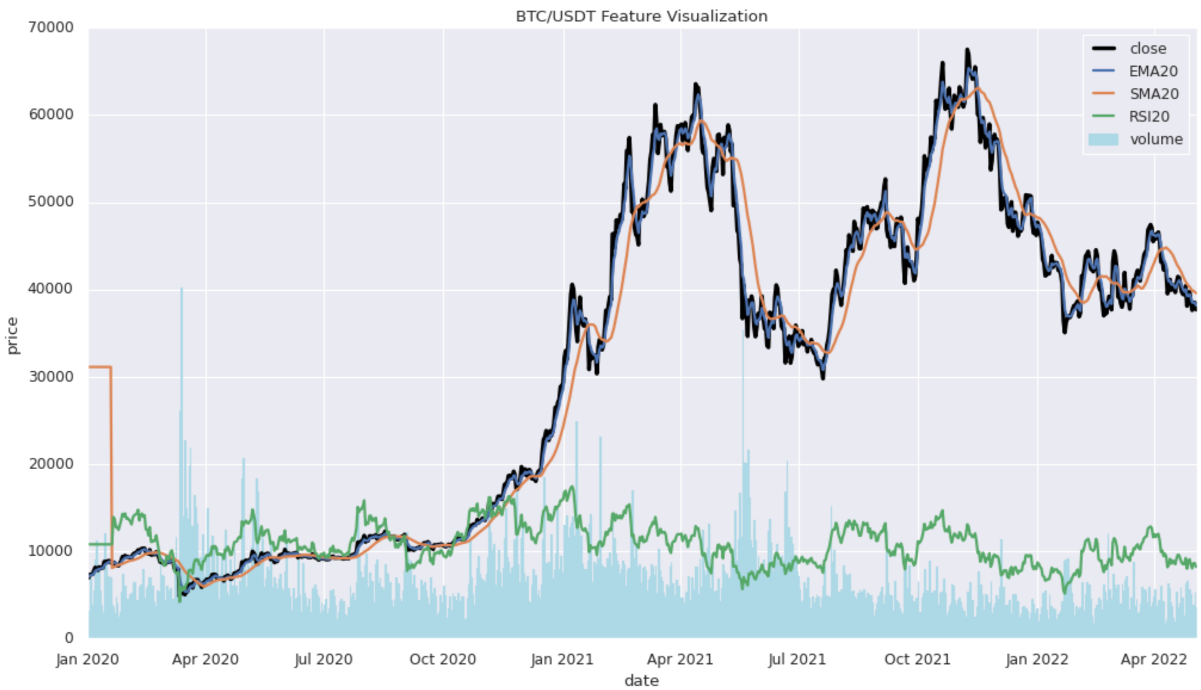

Figure 4.1: Visualization of different features

4.2 Experimental Results

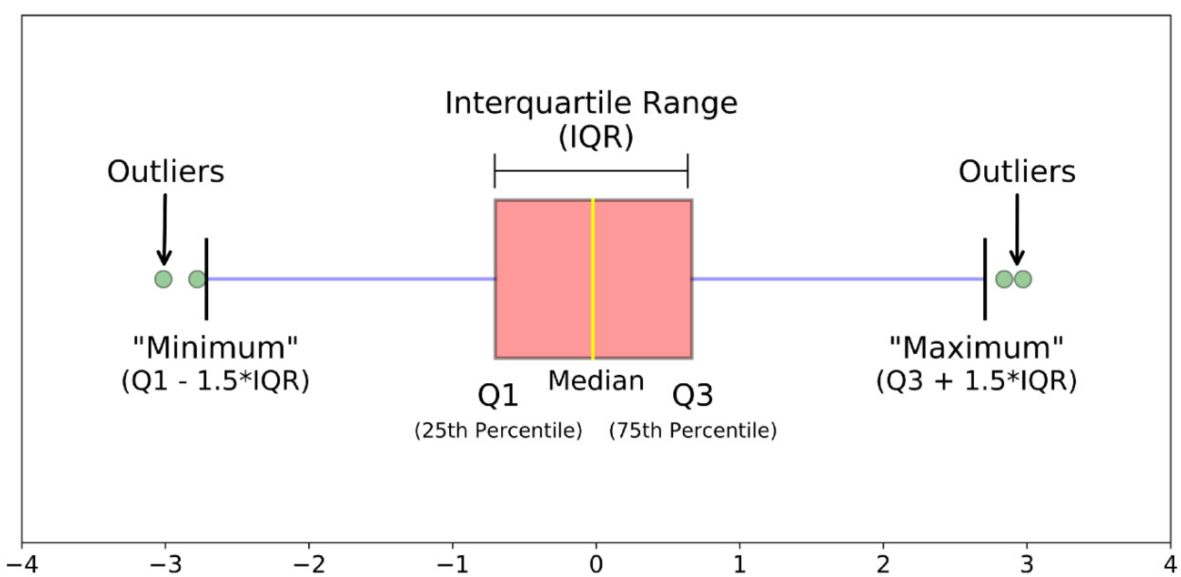

Figure 4.2: Different part of boxplot

For comparison we have used lots of box plots. Here we have shown a image of boxplot. A boxplot provide a standard way to display the distribution of data based on a five number summary ("minimum", first Quartile

(Q1), median, third quartile (Q3), and "maximum"). Boxplot used to find the outliers and their values in any data. It also tell us whether our data is symmetrical, how tightly our data is grouped, and if and how our data is skewed.

Table 4.1: Evaluation metrics of all Strategies

S.no	**Crypto**	**Strategy**	**Profit-per-day**	**Sharpe ratio**
1.	ETH/USDT	NN:hidden_layer_sizes:(50, 50, 5)-local4	1.416074	1.044589
2.	ETH/USDT	NN:hidden_layer_sizes:(50, 50, 5)-local	1.416074	1.044589
3.	ETH/USDT	AdaBoost:learning_rate:1-local	1.416065	1.044589
...	**...**	**...**	**...**	**...**
123.	BTC/USDT	RandomForest:criterion:entropy-local	-0.134326	-1.044069
124.	BTC/USDT	NN:hidden_layer_sizes:(50, 50, 5)-local	-0.138378	-1.093765
125.	BTC/USDT	NN:hidden_layer_sizes:(50, 5)-local	-0.138696	-1.150444

This table contains the final results we are getting for all strategies. This table has total 126 observations. Now based on this table, we will be plotting different categorical plots which we have shown below.

In our project, we have implemented many different strategies starting from traditional strategies to some custom Machine and Deep Learning strategies. We have also implemented some others techniques to improve the overall performance of these strategies. It is hard to show all those results in one go. So, we will try to first show the results in different categories and finally the combined result and comparison. Our Possible categories results is as follows:

4.2.1 Traditional strategies

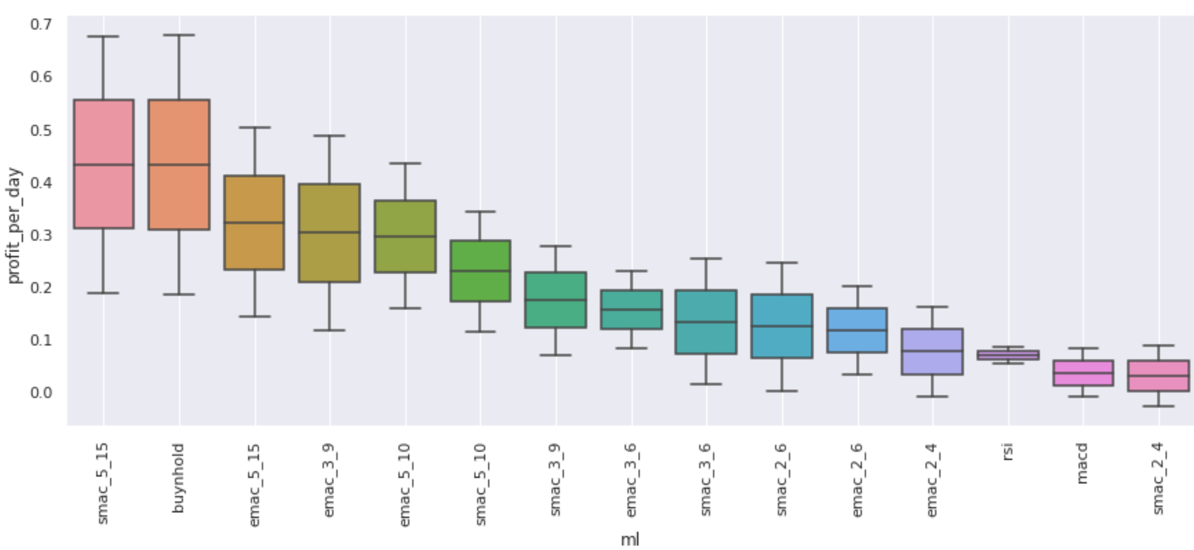

Figure 4.3: Profit Per Day comparison of different traditional strategies

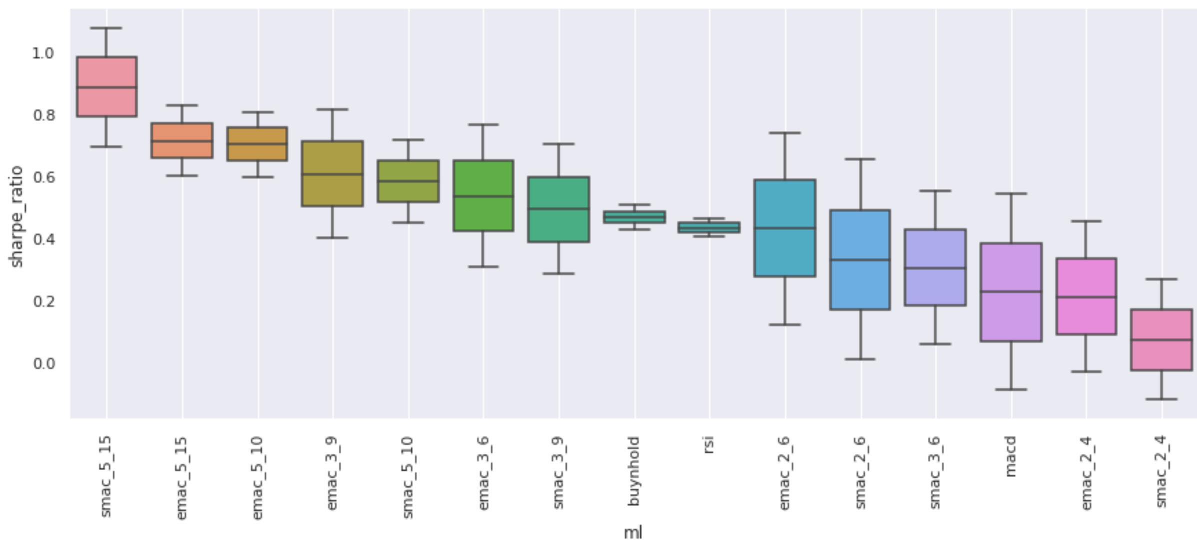

Figure 4.4: Sharpe Ratio comparison of different traditional strategies

As we can see here, These strategies are in decreasing order of their medians. Hence, the strategy at the leftmost place is best and strategies at the rightmost place are worst.

In this particular case **SMAC_5_15** is the **best** strategy as both its profit_per_day and sharpe_ratio values are highest and **SMAC_2_4** is the **worst** strategy as both its profit_per_day and sharpe_ratio values are lowest.

4.2.2 Machine Learning and Deep Learning strategies

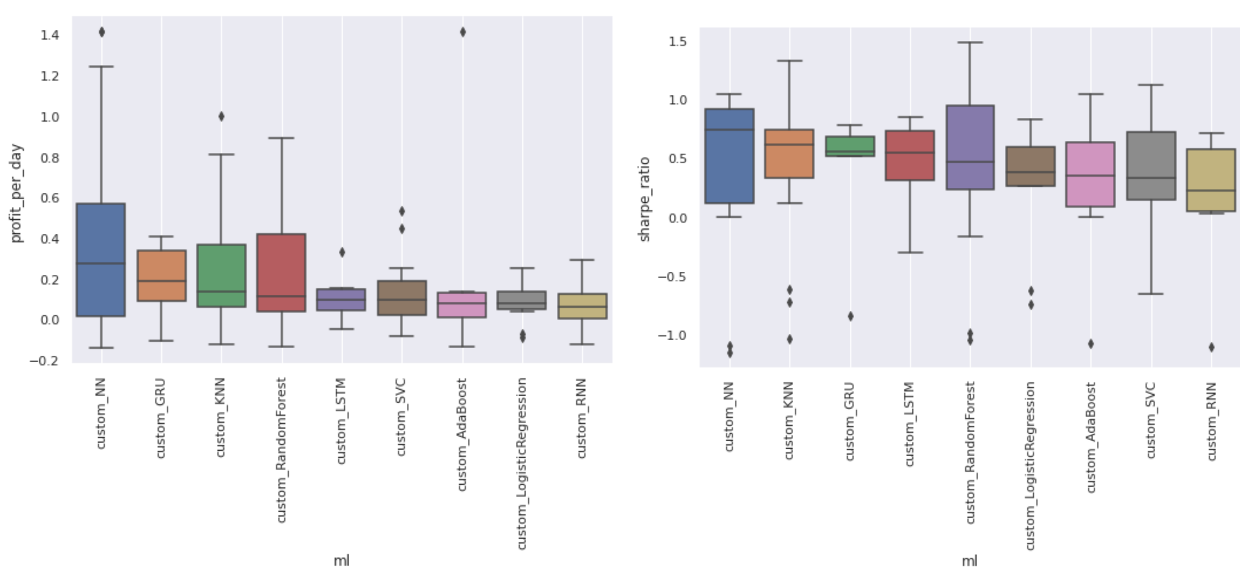

Figure 4.5: Comparison of different ML and DL based strategies

If both the evaluation metrics agree on the rank of a particular strategy then we assign that rank to that strategy and if there are two strategies that don't cant be compared by taking both evaluation metrics at the same time we don't assign order and write them both together.

Based on analyzing both the graph, we can conclude that:

NN > GRU, KNN > RF, LSTM > SVC, Adaboost, LR > RNN

4.2.3 Framing into Consecutive Classification

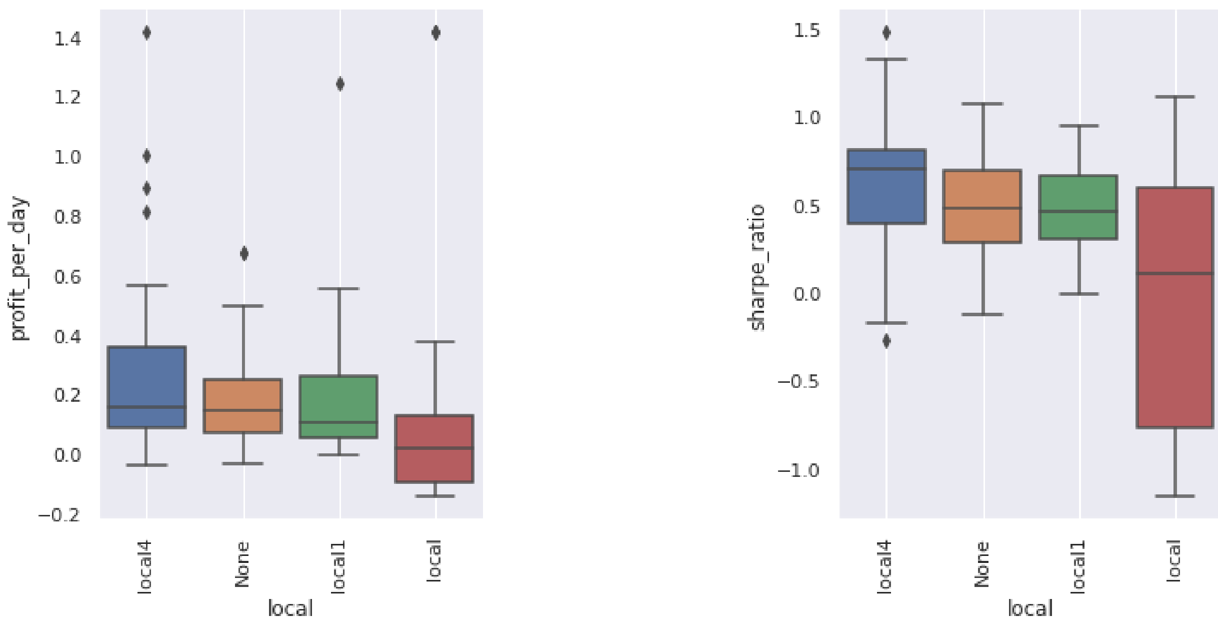

Figure 4.6: Comparison of different classification framing methods

This above graph explains the comparison between different classification framing methods. Here, None means traditional strategies or non-ML/DL strategies. Since both evaluation metrics have **same order**, we can say that:

local4 > None > local1 > local

4.2.4 Cryptocurrency Comparison

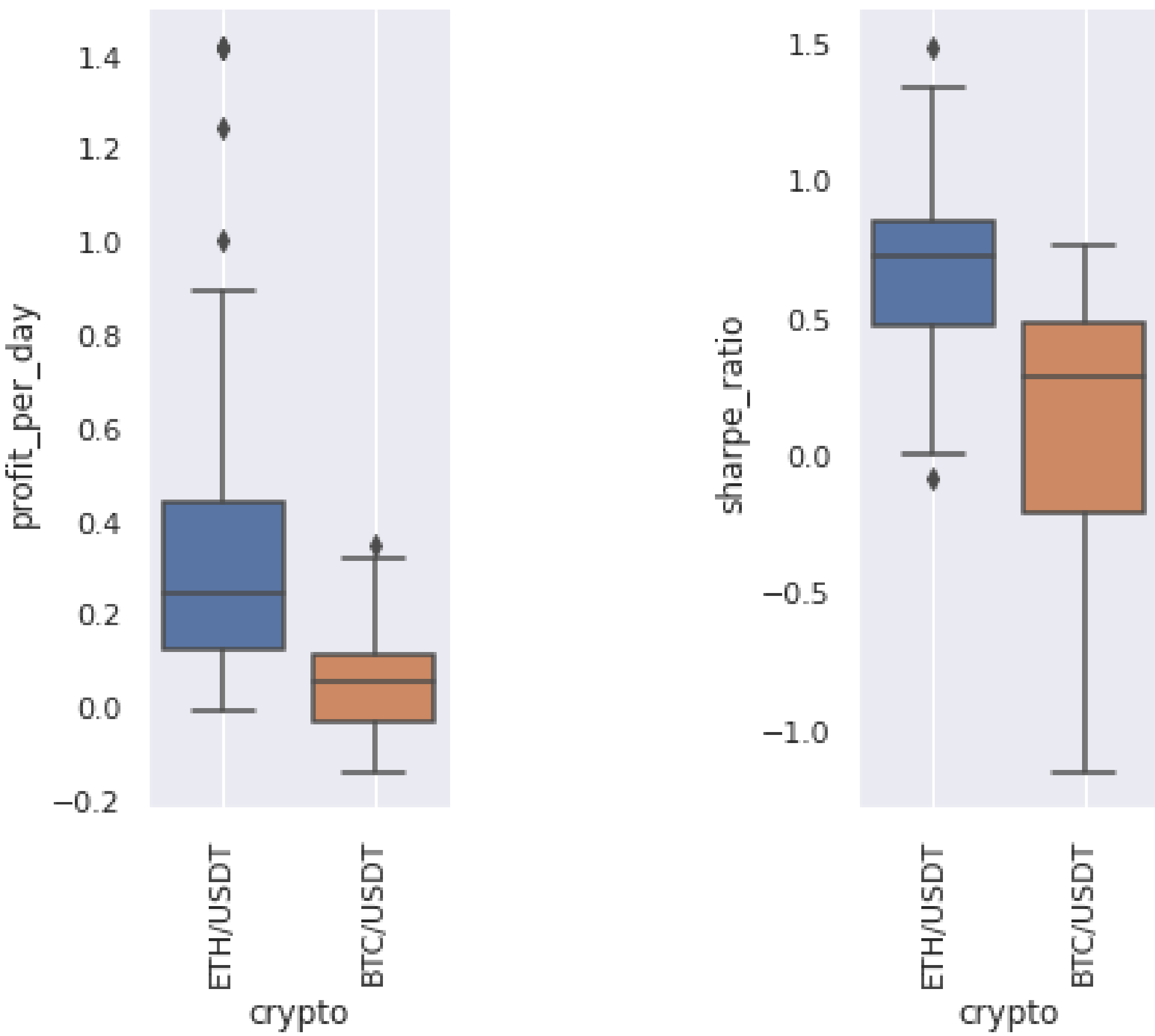

Figure 4.7: Comparison between different crypto currencies

Since, both evaluation metrics have **same order** hence, we can directly conclude that:

ETH/USDT > BTC/USDT

4.3 Comparative Analysis

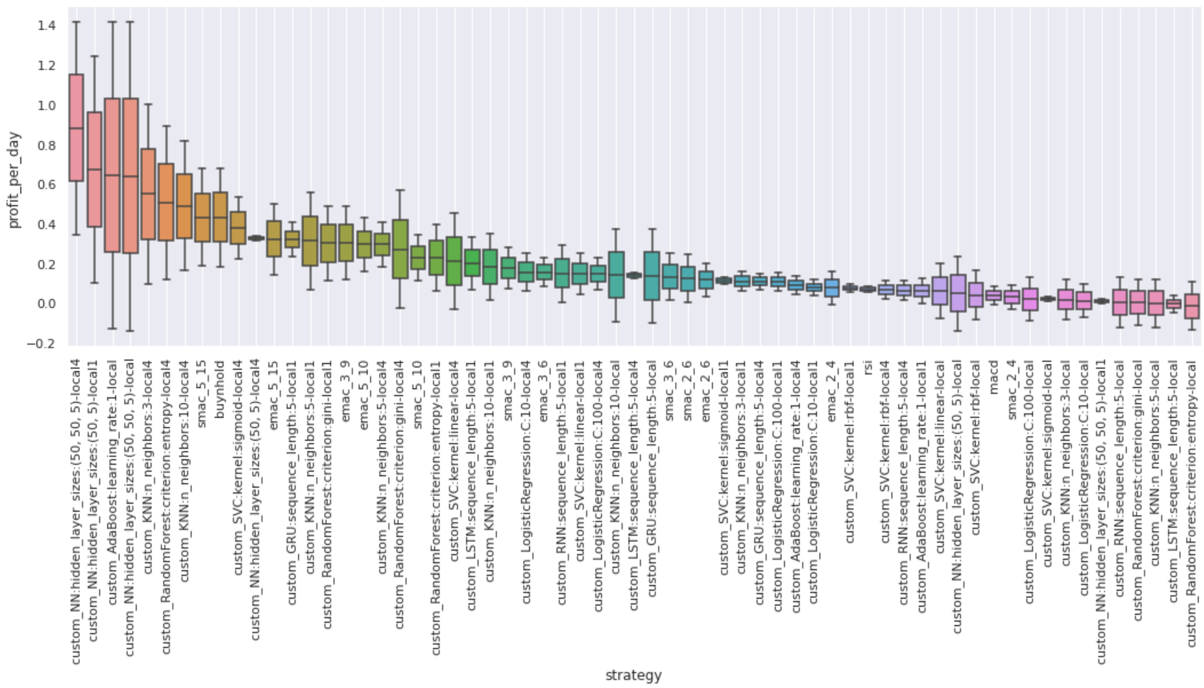

Figure 4.8: Profit per day comparison of all strategies

By looking at the **profit-per-day** we can say that the **best** strategy is **NN_(50,50,5)_local4** and the **worst** strategy is **RandomForest_entropy_local**.

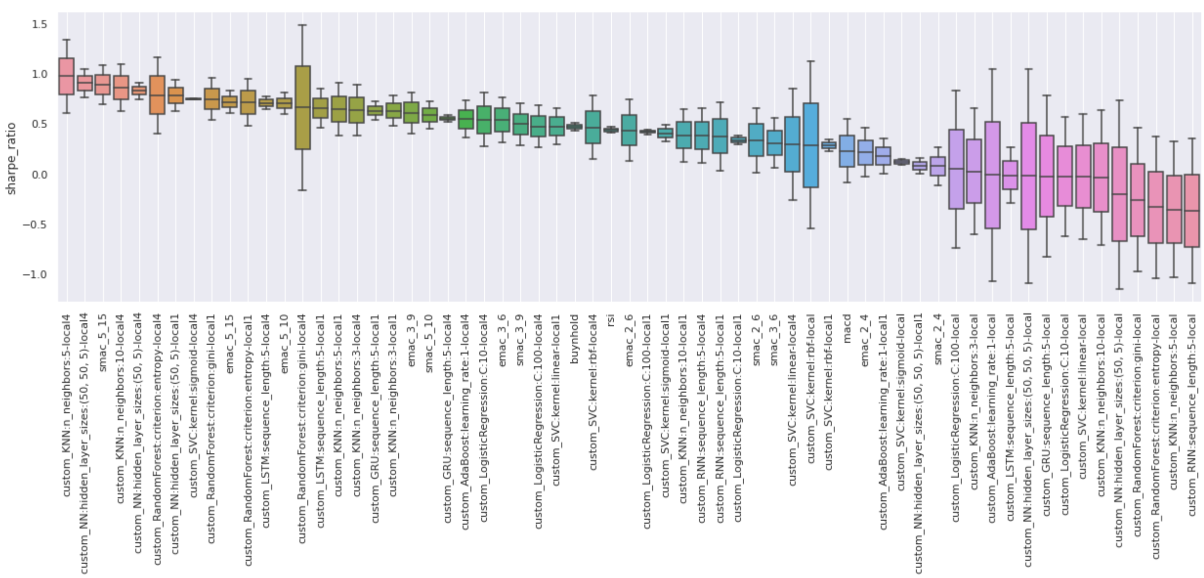

Figure 4.9: Sharpe ratio comparison of all strategies

By looking at the **sharpe_ratio** we can say that the **best strategy** is **KNN_k5_local4** and the **worst** strategy is **RNN_sequence_length5_local**.

Here, we are getting very different results from the two evaluation metrics. The probable reason for that is **Profit-per-day only focuses on returns** while **Sharpe-Ratio penalize inconsistent returns**.

Chapter 5

Conclusions

From this study we can say there are many conclusions emerging, some of those may be summarized as follows. There is no single bullet in determining the best strategies. It depends on what we need.

If we want the highest **returns**, the best trading pair is **ETH/USDT with strategy NeuralNetwork, where hidden_layer_sizes is (50, 50, 5) and local4**, whereas the worst strategy would be **BTC/USDT with strategy NeuralNetwork, where hidden_layer_sizes is (50, 5) and local**.
If we want the highest and **consistent returns**, the best trading pair is **ETH/USDT with strategy RandomForest, where criterion is gini and local4**. The worst trading pair would be **BTC/USDT with strategy NeuralNetwork, where hidden_layer_sizes is (50, 5) and local**.
This maybe the case since Neural network and Random forest is a strategy which generalizes very well. ETH/USDT with local4 is among the best combination meanwhile BTC/USDT with local is among the worst combination. So, the network just learnt the combinations it was given.

In all **traditional strategies** the **SMAC** with fast_period is 5 and slow_period is 15 performed the best. This might be because SMAC tries to identify tops and bottoms using simple moving average(SMA). Also,fast period(5 days) and slow period(15 days) for SMAC does capture trend very well.
The best **Machine learning** strategy is **NeuralNetwork** with hidden_layer_sizes is (50,50,5). As among all the machine learning techniques Neural network generalizes the best.
The best **Deep learning** strategy is **GRU** with sequence length is 5 respectively. This maybe the case since GRU is simple comparison to LSTM and works well with smaller datasets.

Out of all **framing consecutive classification** the **local4** (k is 4 days) perform the best in general, while **local** (local top and bottom) performs the worst. This is justified since its hard to generalize local top and bottoms. Also, by the use of consecutive similiar signals it would capture signals even if triggers are slightly delayed.

The overall best **cryptocurrency** for trading in general is **ETH/USDT** on the other hand **BTC/USDT** is worst. This maybe because ETH/USDT increased much more compared to BTC/USDT percentage wise.

Bibliography

[1] A.-S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: forecasting and trading the taiwan stock index," *Computers Operations Research*, vol. 30, no. 6, pp. 901–923, 2003. Operation Research in Emerging Economics.

[2] J. Oliver-Muncharaz and F. García, "Leading research trends on trading strategies," *Finance, Markets and Valuation*, vol. 6, pp. 27–54, 01 2020.

[3] M. Agrawal, A. Khan, and P. Shukla, "Stock indices price prediction based on technical indicators using deep learning model," *International Journal on Emerging Technologies*, vol. 10, 01 2019.

[4] X. Zhang, W. Qi, and Z. Zhan, "A study on machine-learning-based prediction for bitcoin's price via using lstm and svr," *Journal of Physics: Conference Series*, vol. 1732, p. 012027, 01 2021.

[5] M. Agrawal, P. Shukla, R. Nair, A. Nayyar, and M. Masud, "Stock prediction based on technical indicators using deep learning model," *Cmc -Tech Science Press-*, vol. 70, pp. 287–304, 09 2021.

[6] A. Aguirre, R. Medina, and N. Duque, "Machine learning applied in the stock market through the moving average convergence divergence (macd) indicator," *Investment Management and Financial Innovations*, vol. 17, pp. 44–60, 11 2020.

[7] C. Harvey and Y. Liu, "Evaluating trading strategies," *SSRN Electronic Journal*, vol. 40, 09 2014.

[8] E. Akyildirim, A. Göncü, and A. Sensoy, "Prediction of cryptocurrency returns using machine learning," *Annals of Operations Research*, vol. 297, 02 2021.

[9] M. Ortu, N. Uras, C. Conversano, G. Destefanis, and S. Bartolucci, "On technical trading and social media indicators in cryptocurrency price classification through deep learning," 02 2021.

[10] V. Derbentsev, V. Babenko, K. Khrustalev, H. Obruch, and S. Khrustalova, "Comparative performance of machine learning ensemble algorithms for forecasting cryptocurrency prices," *International Journal of Engineering*, vol. 34, no. 1, pp. 140–148, 2021.

[11] T. Guo and N. Antulov-Fantulin, "Predicting short-term bitcoin price fluctuations from buy and sell orders," 02 2018.

[12] A. Dutta, S. Kumar, and M. Basu, "A gated recurrent unit approach to bitcoin price prediction," *Journal of Risk and Financial Management*, vol. 13, p. 23, 02 2020.

[13] Y. Pang, G. Sundararaj, and J. Ren, "Cryptocurrency price prediction using time series and social sentiment data," p. 35–41, 2019.

[14] J. Abraham, D. W. Higdon, J. Nelson, and J. Ibarra, "Cryptocurrency price prediction using tweet volumes and sentiment analysis," 2018.

[15] J. Patel, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar, and M. Alazab, "Stochastic neural networks for cryptocurrency price prediction," *IEEE Access*, 04 2020.

[16] X. Huang, W. Zhang, Y. Huang, X. Tang, M. Zhang, J. Surbiryala, V. Iosifidis, L. Zhen, and J. Zhang, "Lstm based sentiment analysis for cryptocurrency prediction," 03 2021.

[17] A. Bamidele, R. Ogundokun, R. Jimoh, S. Misra, and T. Aro, "Machine learning algorithm for cryptocurrencies price prediction," pp. 421–447, 06 2021.

[18] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," pp. 339–343, 2018.

[19] R. C. Phillips and D. Gorse, "Predicting cryptocurrency price bubbles using social media data and epidemic modelling," pp. 1–7, 2017.