Bitcoin candlestick predictions using lagged features and machine learning algorithm in R

Training various machine learning algorithms to predict the next candlestick of the bitcoin price using various lagged features

Sandoche Adittane

2025-04-17

Abstract

This report explores how to get the best accuracy on predicting the next candlestick of the bitcoin chart using the previous ones. It compares different algorithms: Generalized Linear Model, Decision Tree, Random Forest, KNN and Gradient boosting and different number of lagged features. This project is part of the 'Data Science: Capstone' module of HarvardX PH125.9x from the edx platform.

Contents

List of Figures

List of Tables

1 Overview

In this study we will try to predict the direction of the next candlestick of the bitcoin chart. Before starting, it's important to understand what are Bitcoin, candlesticks and what is the goal of this study.

1.1 Introduction to Bitcoin

This last years Bitcoin (BTC) has been gaining attention not only by retail investors but also by institutional investor. In 2025 we've seen the emergence of spot Bitcoin exchange-traded funds (ETF) from institutions such as BlackRock, VanEck, Grayscale. With a market capitalization of about 1.68 billion in dollars at the time of writing, Bitcoin started as a peer-to-peer currency, a free alernative to centralized currencies controlled by central banks. It is now used more as an investment, a store of value and even considered as a strategic reserve assets by some countries.

TODO: Add examples with sources.

Bitcoin ows is decentralization and to it's data structure, the blockchain, a chain of block that contains transaction, and to its consensus, the proof of work. Without going too much into details, it makes a Bitcoin a currency that does not rely on a centralized server. Proof of work is a cryptographic competition where the Bitcoin servers called nodes compete to decide which one is the next block to be added to the blockchain. They go through a process called mining where nodes have to use their computing power to find a number called nonce. This computing power is called the hashrate. The node who succeed at "mining" successfully gets rewarded for that.

TODO: reference to my article

The fact that Bitcoin is defined by its codebase is quite facinating, also having all its ledger visible and publically available gives a lot of data available to analyze. Moreover unlike stocks BTC can be traded any time, there is no opening or closing hours, the bitcoin market never stops and it is very easy for anyone to buy and sell bitcoin. Those are two reasons worth studying bitcoin's candlestick charts instead of other asset.

1.2 What are candlesticks?

Let's talk about the candlestick. The price of assets such as bitcoin is described by a serie of candle stick defined by, an opening price, a close price a high and a low also called OHLC. A candlestick can be "up" / "bullish" if closing price is higher than opening price, or "down" / "bearish" otherwise. You can see this visually with the following figure. " "

https://i0.wp.com/techqualitypedia.com/wp-content/uploads/2024/09/candlestick-components.jpg?w=1491&ssl=1 Source: https://techqualitypedia.com/candlestick-patterns-bullish/

The candle stick chart is defined as a time serie of candles, each candle is defined at a defined time and have a time duration. We will explain more in detail in the exploratory analysis.

1.3 Candlesticks pattern

TODO Talk about chartists and common patterns

1.4 Goal of the study

The goal of the study is to find a model able to predict the direction of a candlestick using N previous candles. This number N will be also part of the research. We will have to not only find N but also find what are the best features to achieve the best accuracy.

1.5 Applications

Why is the direction of a candlestick matter? Because being able to predict the direction of the next candle could enable trader to buy and sell on spot market when the predicted candle is green. Also perpetual futures trader can go both way, they can long when the prediction says "up" and "short" when the predictions says "down".

TODO: Give some resource to learn about spot vs future.

2 Exploratory data analysis

In this section we will see what are the are the different dataset available, see what features are available to train the different models, prepare the data, verify it, and choose different machine learning algorithms we will use and compare.

2.1 Data sets

In order to conduct this study we used as a the main data set the historic rates for the trading pair BTC-USD using Coinbase API. TODO: add reference https://docs.cdp.coinbase.com/exchange/reference/exchangerestapi_getproductcandles

We used the following global variables for the full project:

```
trading_pair <- "BTC-USD"
start_date <- "2024-01-01"
end_date <- "2025-03-29"
candlestick_period <- 3600
set.seed(1)
```

The timeframe is the entire year 2024 and the start of the year 2025 until the day we started the study. Note that since January 2024, Bitcoin ETF has officially been approved. The period `candlestick_period <- 3600` is the time of a candle, the candle closes 1h after it starts. Which means we have 24 candles per day.

I choose this settings to have a dataset of around 10000 candles but also since Bitcoin ETF has been approved the market may have taken a different dynamic than the previous years.

Let's see how the dataset looks like.

Table 1: Overview of the BTC-USD candlestick dataset

time	low	high	open	close	volume
2024-01-01 00:00:00	42261.58	42543.64	42288.58	42452.66	379.1973
2024-01-01 01:00:00	42415.00	42749.99	42453.83	42594.68	396.2019
2024-01-01 02:00:00	42488.03	42625.68	42594.58	42571.32	227.1412
2024-01-01 03:00:00	42235.00	42581.26	42571.32	42325.11	306.0057
2024-01-01 04:00:00	42200.00	42393.48	42325.10	42389.77	296.2336
2024-01-01 05:00:00	42175.65	42396.09	42389.78	42231.47	188.1280

We have 10,873 entries in our candle stick dataset. As described in the overview it contains the OCLH data, timestamp and the volume of each candles.

Bitcoin is used by 3 types of users:

- Traders — they are interested by the price and make profit
- Users — using the currency to do payments or to transfer money around the world
- Miners — they mine bitcoin to sell it, their interest is that the price of bitcoin is higher than the cost of mining bitcoin

Keeping this in mind, I tried to find other dataset that could represent each of the type of users that could eventually help in our predictions and I picked the following:

- Fear and greed index — represents the overall mood of the market (traders)
- Hash-rate — defines the overall mining power (miners)
- Average block size — the higher it is the more transactions are happening (users)
- Number of transactions — defines the activity of the network (users)
- Number of unspent transaction outputs (UTXO) — defines how many addresses contains bitcoin, and reflects the network activity (users)

https://www.blockchain.com/explorer/charts/total-bitcoins https://alternative.me/crypto/fear-and-greed-index/

```
fear_and_greed_index <- read_csv(paste0("data/", trading_pair,
    "_fear_and_greed_index_", start_date, "_", end_date, ".csv"))
```

```
## Rows: 453 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): value_classification
## dbl  (1): value
## dttm (1): timestamp
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
fear_and_greed_index <- fear_and_greed_index %>%
    mutate(value = as.numeric(value))
knitr::kable(head(fear_and_greed_index), format = "simple", caption = "Overview of the BTC fear and gre
```

Table 2: Overview of the BTC fear and greed index dataset

value	value_classification	timestamp
26	Fear	2025-03-29
44	Fear	2025-03-28
40	Fear	2025-03-27
47	Neutral	2025-03-26
46	Fear	2025-03-25
45	Fear	2025-03-24

This dataset is a time serie of the daily fear and greed index, it is a value between 0 and 100, 0 being the most fearful and 100 being the most greedy. The data set contains 453 entries.

```
hash_rate <- jsonlite::fromJSON("data/hash-rate.json")$`hash-rate` %>%
    rename(timestamp = x, hash_rate = y) %>%
    mutate(timestamp = as.POSIXct(timestamp/1000, origin = "1970-01-01",
        tz = "UTC")) %>%
    filter(timestamp >= as.POSIXct(start_date, origin = "1970-01-01",
        tz = "UTC") & timestamp <= as.POSIXct(end_date, origin = "1970-01-01",
        tz = "UTC"))
knitr::kable(head(hash_rate), format = "simple", caption = "Overview of the BTC hash rate dataset")
```

Table 3: Overview of the BTC hash rate dataset

timestamp	hash_rate
2024-01-01	501122294
2024-01-02	509303882
2024-01-03	505213088
2024-01-04	520042217
2024-01-05	545098332
2024-01-06	538450791

This dataset is a time serie of the daily hash rate, it is a value in TH/s. The data set contains 454 entries.

```
average_block_size <- jsonlite::fromJSON("data/avg-block-size.json")$`avg-block-size` %>%
    rename(timestamp = x, avg_block_size = y) %>%
    mutate(timestamp = as.POSIXct(timestamp/1000, origin = "1970-01-01",
        tz = "UTC")) %>%
    filter(timestamp >= as.POSIXct(start_date, origin = "1970-01-01",
        tz = "UTC") & timestamp <= as.POSIXct(end_date, origin = "1970-01-01",
        tz = "UTC"))
knitr::kable(head(average_block_size), format = "simple", caption = "Overview of the BTC average block s
```

Table 4: Overview of the BTC average block size dataset

timestamp	avg_block_size
2024-01-01	1.653640
2024-01-02	1.718455
2024-01-03	1.771466
2024-01-04	1.782402
2024-01-05	1.774551
2024-01-06	1.847959

This dataset is a time serie of the daily average block size, it is a value in bytes. The data set contains 454 entries.

```
n_transactions <- jsonlite::fromJSON("data/n-transactions.json")$`n-transactions` %>%
    rename(timestamp = x, n_transactions = y) %>%
    mutate(timestamp = as.POSIXct(timestamp/1000, origin = "1970-01-01",
        tz = "UTC")) %>%
    filter(timestamp >= as.POSIXct(start_date, origin = "1970-01-01",
        tz = "UTC") & timestamp <= as.POSIXct(end_date, origin = "1970-01-01",
        tz = "UTC"))
knitr::kable(head(n_transactions), format = "simple", caption = "Overview of the BTC number of transacti
```

Table 5: Overview of the BTC number of transactions dataset

timestamp	n_transactions
2024-01-01	657752
2024-01-02	367319
2024-01-03	502749
2024-01-04	482557
2024-01-05	420884
2024-01-06	382140

This dataset is a time serie of the daily number of transactions, it is a value in transactions. The data set contains 454 entries.

```
utxo_count <- jsonlite::fromJSON("data/utxo-count.json")$`utxo-count` %>%
    rename(timestamp = x, utxo_count = y) %>%
    mutate(timestamp = as.POSIXct(timestamp/1000, origin = "1970-01-01",
        tz = "UTC"), timestamp = as.Date(timestamp)) %>%
    filter(timestamp >= as.Date(start_date) & timestamp <= as.Date(end_date)) %>%
    group_by(timestamp) %>%
    summarise(utxo_count = mean(utxo_count))  # Take average for each date
knitr::kable(head(utxo_count), format = "simple", caption = "Overview of the BTC UTXO count dataset")
```

Table 6: Overview of the BTC UTXO count dataset

timestamp	utxo_count
2024-01-01	135878807
2024-01-02	136204295
2024-01-03	136536575
2024-01-04	136871780
2024-01-05	137209298
2024-01-06	137552822

This dataset is a time serie of the daily number of UTXO, it is the count of UTXO in the network. The data set contains 454 entries. We had to group by timestamp since some dates had more than 1 value.

As we can see `fear_and_greed_index` seems to miss one data point. We will see fix that in the next sections.

We have different dataset that we will use for the predictions but we are still missing one important data used by traders, technical analysis indicator.

Based on a previous research and blog post I wrote, I decided to include a few indicators that are very common in trading:

- Moving average convergence divergence (MACD)
- Rate of change (ROC)
- Bolinger Bands (BB)
- Relative Strenght Index (RSI)

TODO: add https://medium.com/learning-lab/become-a-better-crypto-trader-with-technical-and-chart-analysis-1496b2fc6b85

Before preparing the dataset let's see what would be the features we can extract from the OHLC candlestick data.

2.2 Features

Our candlesticks dataset from coinbase gives us values that are based on the price of bitcoin, but used itself for a machine learning algorithm it will be hard to use those raw absolute values since the price always fluctuate, so we have to think about what defines the candlesticks we have seen above?

If we isolate a candle we can see the following features:

- Size of the body
- Size of the upper shadow / wicks
- Size of the lower shadow / wicks
- Direction of the candle (up or down)
- Closing price

We are now ready to prepare the dataset for the study.

2.3 Preparation

```
enhance_dataset <- function(candles_data, fear_and_greed_index_data, hash_rate_data, average_block_size_
  candles_enhanced <- candles_data %>%
    mutate(date_only = as.Date(time)) %>%
    left_join(fear_and_greed_index_data, by = c("date_only" = "timestamp")) %>%
    left_join(hash_rate_data, by = c("date_only" = "timestamp")) %>%
    left_join(average_block_size_data, by = c("date_only" = "timestamp")) %>%
    left_join(n_transactions_data, by = c("date_only" = "timestamp")) %>%
    left_join(utxo_count_data, by = c("date_only" = "timestamp")) %>%
    mutate(
      body_size = abs(close - open),
      upper_shadow_size = high - pmax(close, open),
      lower_shadow_size = pmin(close, open) - low,
      direction = ifelse(close > open, "up", "down"),
    ) %>%
    tq_mutate(
      select = close,
      mutate_fun = ROC,
      n = 14,
      col_rename = "roc"
    ) %>%
    tq_mutate( # https://www.keenbase-trading.com/find-best-macd-settings/#t-1719588154943
      select = close,
      mutate_fun = MACD,
      nFast = 12,
      nSlow = 26,
      nSig = 9,
      col_rename = c("macd", "signal")
    ) %>%
    tq_mutate(
      select = close,
      mutate_fun = RSI,
```

```
      n = 14,
      col_rename = "rsi"
    ) %>%
    tq_mutate(
      select = close,
      mutate_fun = BBands,
      n = 20,
      sd = 2,
      col_rename = "bband"
    )

  candles_enhanced
}

candles_enhanced <- enhance_dataset(candles, fear_and_greed_index, hash_rate, average_block_size, n_tra
```

```
## Warning in coerce_to_tibble(ret, date_col_name, time_zone, col_rename): Could not rename columns. Th
##   Is the length of `col_rename` the same as the number of columns returned from the `mutate_fun`?
```

```
knitr::kable(head(candles_enhanced), format = "simple", caption = "Overview of the candlestick dataset
```

time	low	high	open	close	volume	date_only	value	value_classification	h
2024-01-01 00:00:00	42261.58	42543.64	42288.58	42452.66	379.1973	2024-01-01	65	Greed	5
2024-01-01 01:00:00	42415.00	42749.99	42453.83	42594.68	396.2019	2024-01-01	65	Greed	5
2024-01-01 02:00:00	42488.03	42625.68	42594.58	42571.32	227.1412	2024-01-01	65	Greed	5
2024-01-01 03:00:00	42235.00	42581.26	42571.32	42325.11	306.0057	2024-01-01	65	Greed	5
2024-01-01 04:00:00	42200.00	42393.48	42325.10	42389.77	296.2336	2024-01-01	65	Greed	5
2024-01-01 05:00:00	42175.65	42396.09	42389.78	42231.47	188.1280	2024-01-01	65	Greed	5

We have now a table with all the features discussed above. Before adding the lagged features let's explore our set.

First let's see if we have NAs.

```
na_indexes <- which(apply(candles_enhanced, 1, function(x) any(is.na(x))))
knitr::kable(candles_enhanced[na_indexes, ], format = "simple",
    caption = "NAs of the dataset")
```

time	low	high	open	close	volume	date_only	value	value_classification
2024-01-01 00:00:00	42261.58	42543.64	42288.58	42452.66	379.197253	2024-01-01	65	Greed
2024-01-01 01:00:00	42415.00	42749.99	42453.83	42594.68	396.201924	2024-01-01	65	Greed
2024-01-01 02:00:00	42488.03	42625.68	42594.58	42571.32	227.141166	2024-01-01	65	Greed
2024-01-01 03:00:00	42235.00	42581.26	42571.32	42325.11	306.005694	2024-01-01	65	Greed
2024-01-01 04:00:00	42200.00	42393.48	42325.10	42389.77	296.233644	2024-01-01	65	Greed
2024-01-01 05:00:00	42175.65	42396.09	42389.78	42231.47	188.128033	2024-01-01	65	Greed
2024-01-01 06:00:00	42199.63	42463.83	42231.47	42400.90	327.010976	2024-01-01	65	Greed

time	low	high	open	close	volume	date_only	value	value_classification
2024-01-01 07:00:00	42396.80	42534.49	42400.90	42496.49	407.835097	2024-01-01	65	Greed
2024-01-01 08:00:00	42451.00	42560.26	42496.58	42552.70	134.066714	2024-01-01	65	Greed
2024-01-01 09:00:00	42533.77	42692.84	42552.70	42650.97	128.157349	2024-01-01	65	Greed
2024-01-01 10:00:00	42625.75	42750.00	42650.97	42688.50	118.457396	2024-01-01	65	Greed
2024-01-01 11:00:00	42598.94	42767.60	42686.73	42690.00	135.177672	2024-01-01	65	Greed
2024-01-01 12:00:00	42610.77	42778.74	42690.00	42647.83	143.378020	2024-01-01	65	Greed
2024-01-01 13:00:00	42608.98	42750.00	42647.85	42715.88	101.798625	2024-01-01	65	Greed
2024-01-01 14:00:00	42581.54	42723.28	42717.53	42635.19	254.331002	2024-01-01	65	Greed
2024-01-01 15:00:00	42601.88	42868.74	42633.57	42797.33	323.614895	2024-01-01	65	Greed
2024-01-01 16:00:00	42680.01	42880.97	42799.37	42742.35	330.024110	2024-01-01	65	Greed
2024-01-01 17:00:00	42720.76	42846.42	42738.88	42833.66	254.872245	2024-01-01	65	Greed
2024-01-01 18:00:00	42835.63	43228.37	42835.63	43120.92	625.461696	2024-01-01	65	Greed
2024-01-01 19:00:00	43106.97	43567.46	43123.82	43547.61	506.451809	2024-01-01	65	Greed
2024-01-01 20:00:00	43537.04	43849.90	43537.04	43701.58	559.329005	2024-01-01	65	Greed
2024-01-01 21:00:00	43467.97	43800.00	43703.06	43632.21	313.991915	2024-01-01	65	Greed
2024-01-01 22:00:00	43389.00	43677.06	43631.79	43546.06	247.539449	2024-01-01	65	Greed
2024-01-01 23:00:00	43545.99	44240.80	43545.99	44220.78	1273.322823	2024-01-01	65	Greed
2024-01-02 00:00:00	44195.13	45250.00	44220.78	45093.17	3023.793096	2024-01-02	71	Greed
2024-01-02 01:00:00	44714.89	45417.45	45093.14	44894.58	1983.082789	2024-01-02	71	Greed
2024-01-02 02:00:00	44891.40	45500.00	44891.41	45485.31	1913.577949	2024-01-02	71	Greed
2024-01-02 03:00:00	45178.34	45601.00	45485.32	45473.97	1512.833935	2024-01-02	71	Greed
2024-01-02 04:00:00	45204.47	45544.10	45476.27	45218.83	681.159462	2024-01-02	71	Greed
2024-01-02 05:00:00	45131.00	45370.54	45218.84	45216.32	484.732085	2024-01-02	71	Greed
2024-01-02 06:00:00	45166.39	45361.61	45214.41	45208.54	495.637602	2024-01-02	71	Greed
2024-01-02 07:00:00	45209.48	45707.69	45209.48	45504.64	976.921273	2024-01-02	71	Greed
2024-01-02 08:00:00	45382.34	45899.96	45504.40	45808.07	759.882169	2024-01-02	71	Greed
2024-10-26 00:00:00	66413.18	66754.02	66564.51	66635.55	359.487900	2024-10-26	NA	NA
2024-10-26 01:00:00	66430.80	66711.88	66637.60	66597.10	226.587448	2024-10-26	NA	NA
2024-10-26 02:00:00	66331.95	66930.14	66594.88	66728.09	162.061446	2024-10-26	NA	NA
2024-10-26 03:00:00	66580.85	66890.00	66730.12	66816.54	122.871792	2024-10-26	NA	NA
2024-10-26 04:00:00	66687.79	66903.91	66814.44	66855.95	148.712344	2024-10-26	NA	NA
2024-10-26 05:00:00	66851.79	67156.74	66855.94	67049.34	163.124225	2024-10-26	NA	NA
2024-10-26 06:00:00	66959.24	67159.97	67049.33	67086.89	108.339046	2024-10-26	NA	NA
2024-10-26 07:00:00	66913.98	67108.03	67086.89	66926.56	105.386323	2024-10-26	NA	NA
2024-10-26 08:00:00	66920.30	67098.13	66926.56	67058.44	94.345883	2024-10-26	NA	NA
2024-10-26 09:00:00	66973.03	67188.55	67058.44	66973.03	92.454048	2024-10-26	NA	NA
2024-10-26 10:00:00	66920.07	67108.34	66968.89	66977.74	69.774037	2024-10-26	NA	NA
2024-10-26 11:00:00	66876.82	67083.85	66977.74	67055.68	93.974564	2024-10-26	NA	NA
2024-10-26 12:00:00	66906.75	67101.59	67056.84	66946.40	99.399923	2024-10-26	NA	NA
2024-10-26 13:00:00	66784.25	67031.28	66946.40	66808.06	92.616172	2024-10-26	NA	NA
2024-10-26 14:00:00	66644.83	66874.66	66803.39	66713.12	126.183413	2024-10-26	NA	NA
2024-10-26 15:00:00	66675.24	66920.88	66712.82	66795.54	87.307429	2024-10-26	NA	NA
2024-10-26 16:00:00	66781.74	66870.57	66800.48	66818.88	2.195708	2024-10-26	NA	NA
2024-10-26 17:00:00	66388.20	67055.10	66864.73	66974.50	49.094828	2024-10-26	NA	NA
2024-10-26 18:00:00	66926.63	67069.99	66974.49	66942.16	99.453480	2024-10-26	NA	NA
2024-10-26 19:00:00	66936.07	67103.18	66942.16	67100.49	223.657084	2024-10-26	NA	NA
2024-10-26 20:00:00	67050.49	67365.18	67100.50	67173.56	144.763009	2024-10-26	NA	NA
2024-10-26 21:00:00	66999.83	67186.23	67173.56	67089.21	121.258639	2024-10-26	NA	NA
2024-10-26 22:00:00	67015.71	67163.50	67088.99	67042.50	55.228574	2024-10-26	NA	NA
2024-10-26 23:00:00	66993.44	67069.68	67039.92	67012.56	100.942655	2024-10-26	NA	NA

We can see in the table above that there are 2 types of NAs:

1. Technical analisis indicators
2. Fear and greed index

The technical analysis indicators are located at the start of the table which is normal since to calculate these indicators you need to have a certain number of previous values. Clearing those NAs will be enough.

As concern the fear and greed index missing value we can notice that we are missing the values of the day `2024-10-26` so we will add the values of the average between the day before and after manually.

It's important to do so to have coherant lagged values.

```
date_na <- as.Date("2024-10-26")
fear_and_greed_index_date_before_na <- fear_and_greed_index %>%
    filter(timestamp == as.Date("2024-10-25"))
fear_and_greed_index_date_after_na <- fear_and_greed_index %>%
    filter(timestamp == as.Date("2024-10-27"))
fear_and_greed_value_date_na <- mean(c(fear_and_greed_index_date_before_na$value,
    fear_and_greed_index_date_after_na$value))

fear_and_greed_index_corrected <- fear_and_greed_index %>%
    bind_rows(tibble(timestamp = date_na, value = fear_and_greed_value_date_na,
        value_classification = "Greed"))

fear_and_greed_index %>%
    filter(timestamp == date_na) %>%
    nrow()
```

```
## [1] 0
```

```
fear_and_greed_index_corrected %>%
    filter(timestamp == date_na) %>%
    nrow()
```

```
## [1] 1
```

Let's check the NAs again.

```
candles_enhanced_cleaned <- enhance_dataset(candles, fear_and_greed_index_corrected,
    hash_rate, average_block_size, n_transactions, utxo_count)
```

```
## Warning in coerce_to_tibble(ret, date_col_name, time_zone, col_rename): Could not rename columns. The
##   Is the length of `col_rename` the same as the number of columns returned from the `mutate_fun`?
```

```
na_indexes <- which(apply(candles_enhanced_cleaned, 1, function(x) any(is.na(x))))
knitr::kable(candles_enhanced_cleaned[na_indexes, ], format = "simple",
    caption = "NAs of the dataset cleaned")
```

time	low	high	open	close	volume	date_only	value	value_classification
2024-01-01 00:00:00	42261.58	42543.64	42288.58	42452.66	379.1973	2024-01-01	65	Greed
2024-01-01 01:00:00	42415.00	42749.99	42453.83	42594.68	396.2019	2024-01-01	65	Greed
2024-01-01 02:00:00	42488.03	42625.68	42594.58	42571.32	227.1412	2024-01-01	65	Greed
2024-01-01 03:00:00	42235.00	42581.26	42571.32	42325.11	306.0057	2024-01-01	65	Greed
2024-01-01 04:00:00	42200.00	42393.48	42325.10	42389.77	296.2336	2024-01-01	65	Greed
2024-01-01 05:00:00	42175.65	42396.09	42389.78	42231.47	188.1280	2024-01-01	65	Greed
2024-01-01 06:00:00	42199.63	42463.83	42231.47	42400.90	327.0110	2024-01-01	65	Greed
2024-01-01 07:00:00	42396.80	42534.49	42400.90	42496.49	407.8351	2024-01-01	65	Greed
2024-01-01 08:00:00	42451.00	42560.26	42496.58	42552.70	134.0667	2024-01-01	65	Greed
2024-01-01 09:00:00	42533.77	42692.84	42552.70	42650.97	128.1573	2024-01-01	65	Greed
2024-01-01 10:00:00	42625.75	42750.00	42650.97	42688.50	118.4574	2024-01-01	65	Greed
2024-01-01 11:00:00	42598.94	42767.60	42686.73	42690.00	135.1777	2024-01-01	65	Greed
2024-01-01 12:00:00	42610.77	42778.74	42690.00	42647.83	143.3780	2024-01-01	65	Greed
2024-01-01 13:00:00	42608.98	42750.00	42647.85	42715.88	101.7986	2024-01-01	65	Greed
2024-01-01 14:00:00	42581.54	42723.28	42717.53	42635.19	254.3310	2024-01-01	65	Greed
2024-01-01 15:00:00	42601.88	42868.74	42633.57	42797.33	323.6149	2024-01-01	65	Greed
2024-01-01 16:00:00	42680.01	42880.97	42799.37	42742.35	330.0241	2024-01-01	65	Greed
2024-01-01 17:00:00	42720.76	42846.42	42738.88	42833.66	254.8722	2024-01-01	65	Greed
2024-01-01 18:00:00	42835.63	43228.37	42835.63	43120.92	625.4617	2024-01-01	65	Greed
2024-01-01 19:00:00	43106.97	43567.46	43123.82	43547.61	506.4518	2024-01-01	65	Greed
2024-01-01 20:00:00	43537.04	43849.90	43537.04	43701.58	559.3290	2024-01-01	65	Greed
2024-01-01 21:00:00	43467.97	43800.00	43703.06	43632.21	313.9919	2024-01-01	65	Greed
2024-01-01 22:00:00	43389.00	43677.06	43631.79	43546.06	247.5394	2024-01-01	65	Greed
2024-01-01 23:00:00	43545.99	44240.80	43545.99	44220.78	1273.3228	2024-01-01	65	Greed
2024-01-02 00:00:00	44195.13	45250.00	44220.78	45093.17	3023.7931	2024-01-02	71	Greed
2024-01-02 01:00:00	44714.89	45417.45	45093.14	44894.58	1983.0828	2024-01-02	71	Greed
2024-01-02 02:00:00	44891.40	45500.00	44891.41	45485.31	1913.5779	2024-01-02	71	Greed
2024-01-02 03:00:00	45178.34	45601.00	45485.32	45473.97	1512.8339	2024-01-02	71	Greed
2024-01-02 04:00:00	45204.47	45544.10	45476.27	45218.83	681.1595	2024-01-02	71	Greed
2024-01-02 05:00:00	45131.00	45370.54	45218.84	45216.32	484.7321	2024-01-02	71	Greed
2024-01-02 06:00:00	45166.39	45361.61	45214.41	45208.54	495.6376	2024-01-02	71	Greed
2024-01-02 07:00:00	45209.48	45707.69	45209.48	45504.64	976.9213	2024-01-02	71	Greed
2024-01-02 08:00:00	45382.34	45899.96	45504.40	45808.07	759.8822	2024-01-02	71	Greed

We can see now the only NAs are the missing TA values that we can clean with a simple line of code.

```
candles_enhanced_cleaned_no_na <- candles_enhanced_cleaned %>%
    drop_na()
sum(is.na(candles_enhanced_cleaned_no_na))
```

```
## [1] 0
```

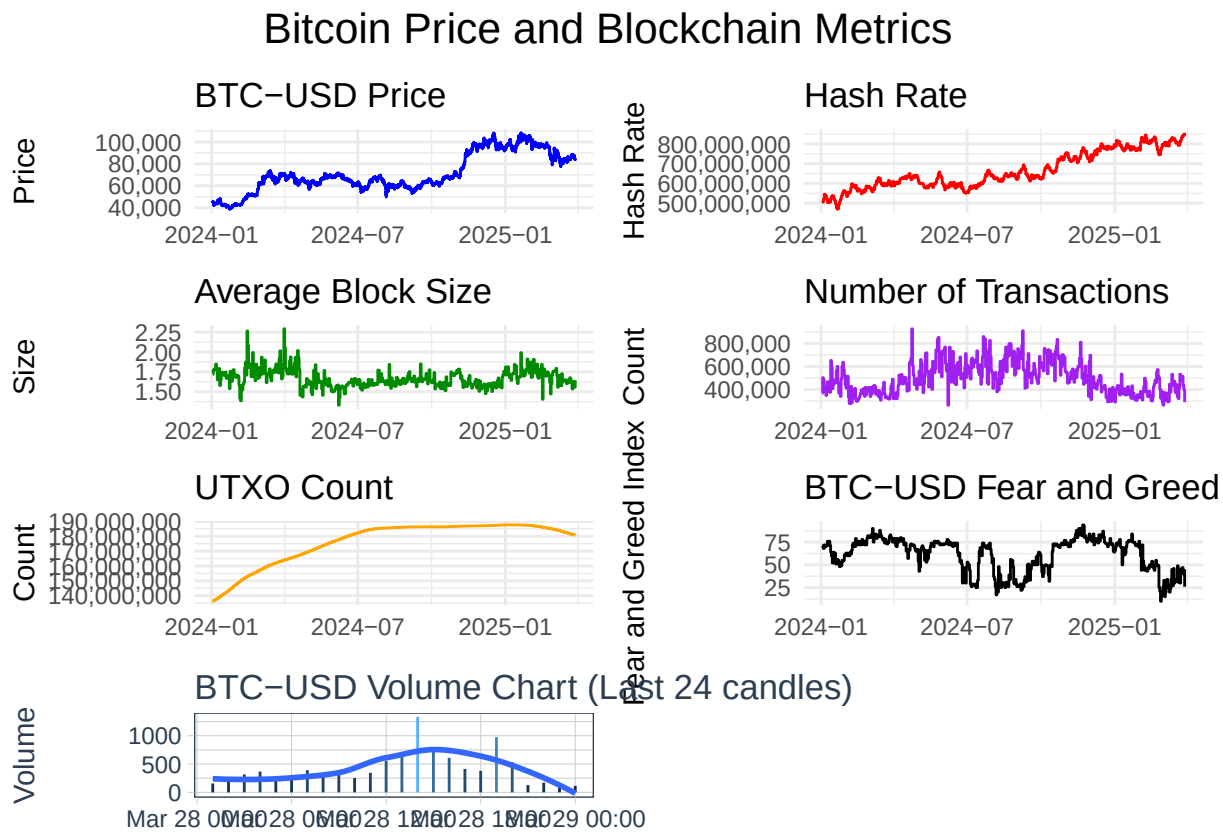
2.4 Visual analysis

First of all let's plot the data to visually verify the data.

```
p1 <- candles_enhanced_cleaned_no_na %>%
    ggplot(aes(x = time, y = close)) + geom_line(color = "blue") +
```

```
    theme_minimal() + labs(title = "BTC-USD Price", y = "Price") +
    scale_y_continuous(labels = scales::comma)

p2 <- candles_enhanced_cleaned_no_na %>%
    ggplot(aes(x = time, y = hash_rate)) + geom_line(color = "red") +
    theme_minimal() + labs(title = "Hash Rate", y = "Hash Rate") +
    scale_y_continuous(labels = scales::comma)

p3 <- candles_enhanced_cleaned_no_na %>%
    ggplot(aes(x = time, y = avg_block_size)) + geom_line(color = "green4") +
    theme_minimal() + labs(title = "Average Block Size", y = "Size") +
    scale_y_continuous(labels = scales::comma)

p4 <- candles_enhanced_cleaned_no_na %>%
    ggplot(aes(x = time, y = n_transactions)) + geom_line(color = "purple") +
    theme_minimal() + labs(title = "Number of Transactions",
    y = "Count") + scale_y_continuous(labels = scales::comma)

p5 <- candles_enhanced_cleaned_no_na %>%
    ggplot(aes(x = time, y = utxo_count)) + geom_line(color = "orange") +
    theme_minimal() + labs(title = "UTXO Count", y = "Count") +
    scale_y_continuous(labels = scales::comma)

p6 <- candles_enhanced_cleaned_no_na %>%
    ggplot(aes(x = time, y = value)) + geom_line() + theme_minimal() +
    labs(title = "BTC-USD Fear and Greed Index Evolution", x = "Time",
        y = "Fear and Greed Index") + scale_y_continuous(labels = scales::comma)

# For more readability we are only plotting the last 100
# candles
p7 <- candles_enhanced_cleaned_no_na %>%
    tail(24) %>%
    ggplot(aes(x = time, y = volume)) + geom_segment(aes(xend = time,
    yend = 0, color = volume)) + geom_smooth(method = "loess",
    se = FALSE) + labs(title = "BTC-USD Volume Chart (Last 24 candles)",
    y = "Volume", x = "") + theme_tq() + theme(legend.position = "none")

combined_plot <- (p1/p2/p3/p4/p5/p6/p7) + plot_layout(ncol = 2,
    heights = c(1, 1, 1, 1)) + plot_annotation(title = "Bitcoin Price and Blockchain Metrics",
    theme = theme(plot.title = element_text(hjust = 0.5, size = 16))) &
    theme(axis.title.x = element_blank())

combined_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



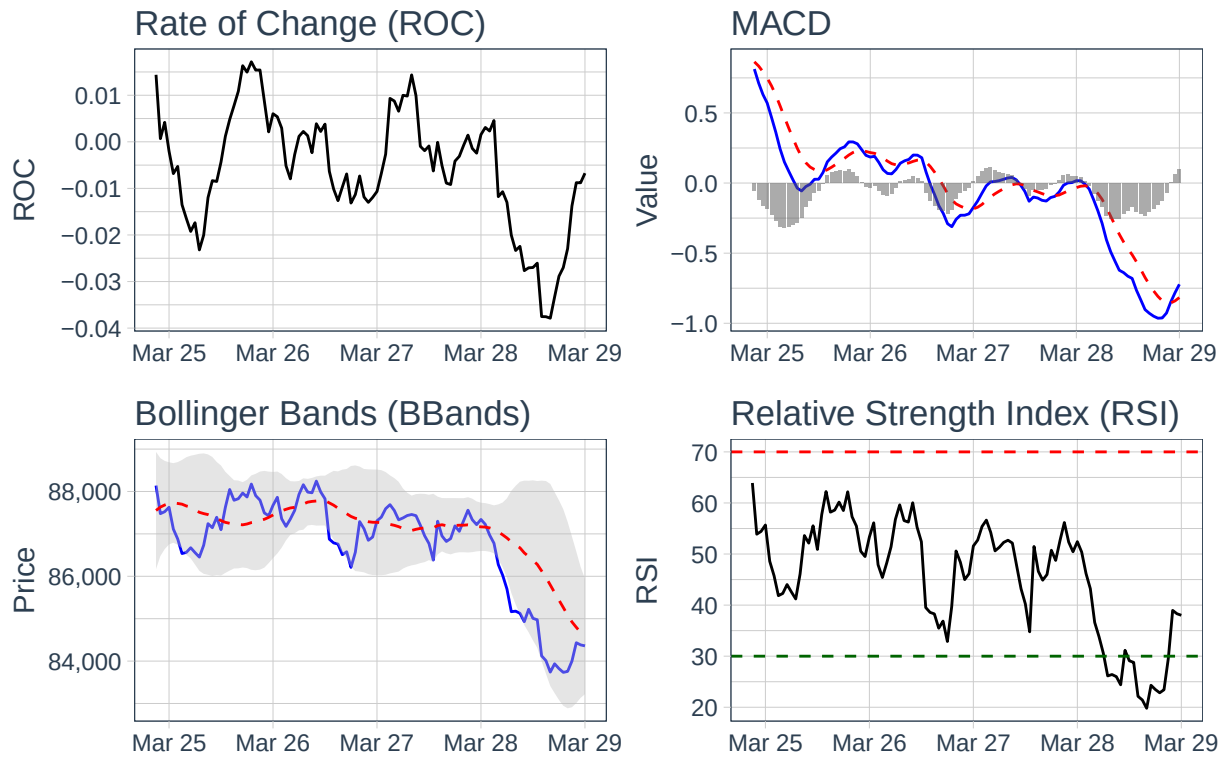
TODO Manually add the one below since it's failing

```
# For more readability we are only plotting the last 24 candles
p7 <- candles_enhanced_cleaned_no_na %>%
  tail(24) %>%
  mutate(direction = ifelse(close >= open, "up", "down")) %>%
  ggplot(aes(x = time, y = close)) +
  # The shadows (wicks)
  geom_segment(aes(xend = time, y = low, yend = high, color = direction), size = 0.5) +
  # The body
  geom_segment(aes(xend = time, y = open, yend = close, color = direction), size = 5) +
  scale_color_manual(values = c("up" = "darkgreen", "down" = "red")) +
  theme_tq() +
  theme(legend.position = "none") +
  labs(
    title = "BTC-USD Candlestick Chart (Last 24 Candles)",
    x = "Time",
    y = "Price"
  ) +
  scale_y_continuous(labels = scales::comma)
```

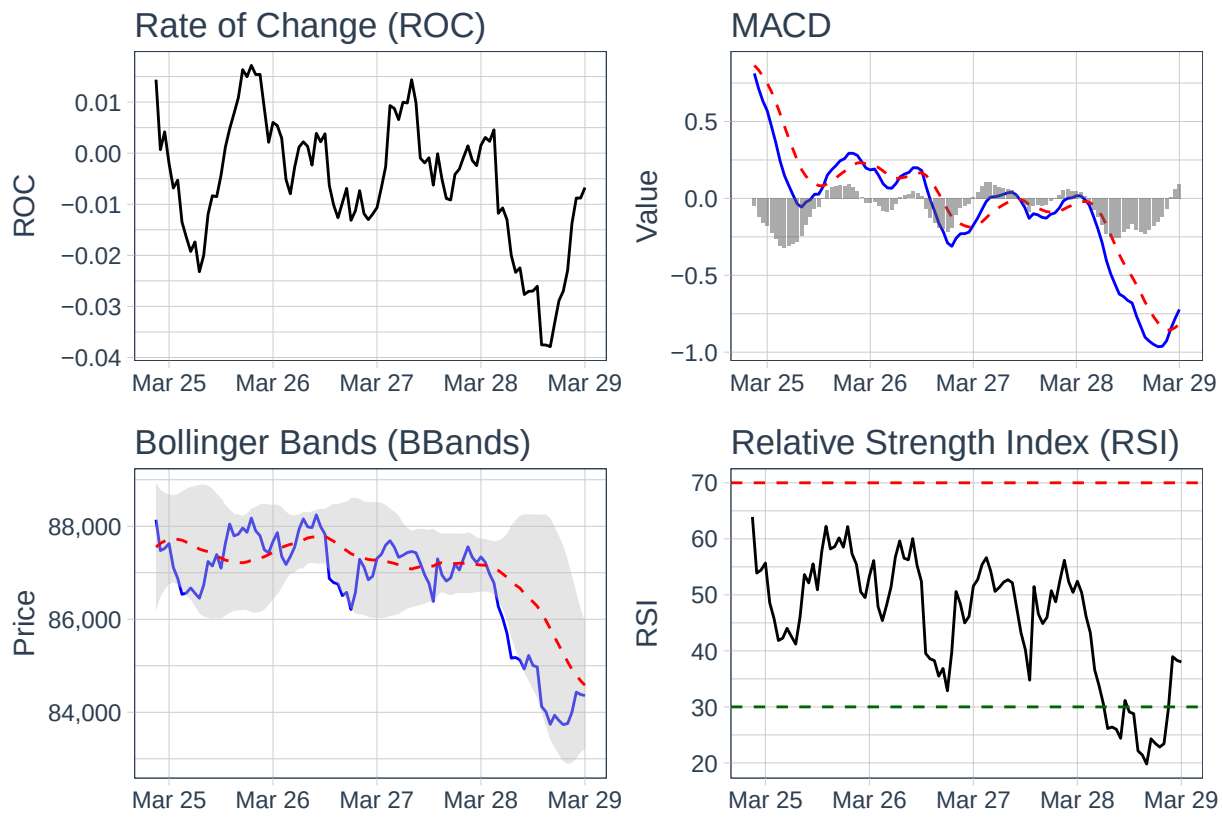
TODO Add chart of TAs TODO When doing the real data don't forget to rm na

```
# Plotting Technical Analysis Indicators for the last 100 candles for better readability

# Subset the data for plotting (last 100 candles)
```

```r
plot_data_ta <- candles_enhanced_cleaned_no_na %>% tail(100)

# ROC Plot
p_roc <- plot_data_ta %>%
  ggplot(aes(x = time, y = roc)) +
  geom_line() +
  labs(title = "Rate of Change (ROC)", y = "ROC") +
  theme_tq() +
  theme(axis.title.x = element_blank())

# Bollinger Bands Plot
p_bbands <- plot_data_ta %>%
  ggplot(aes(x = time, y = close)) +
  geom_line(aes(y = close), color = "blue") + # Close price
  geom_ribbon(aes(ymin = dn, ymax = up), fill = "grey", alpha = 0.4) + # Bollinger Bands area
  geom_line(aes(y = mavg), color = "red", linetype = "dashed") + # Moving Average
  labs(title = "Bollinger Bands (BBands)", y = "Price") +
  theme_tq() +
  theme(axis.title.x = element_blank()) +
  scale_y_continuous(labels = scales::comma)

# MACD Plot
p_macd <- plot_data_ta %>%
  ggplot(aes(x = time)) +
  geom_line(aes(y = macd), color = "blue") +  # MACD line
  geom_line(aes(y = signal), color = "red", linetype = "dashed") + # Signal line
  geom_col(aes(y = macd - signal), alpha = 0.5) + # Histogram of MACD - Signal
  labs(title = "MACD", y = "Value") +
  theme_tq() +
  theme(axis.title.x = element_blank())

# RSI Plot
p_rsi <- plot_data_ta %>%
  ggplot(aes(x = time, y = rsi)) +
  geom_line() +
  geom_hline(yintercept = 70, linetype = "dashed", color = "red") +  # Overbought level
  geom_hline(yintercept = 30, linetype = "dashed", color = "darkgreen") + # Oversold level
  labs(title = "Relative Strength Index (RSI)", y = "RSI") +
  theme_tq() +
  theme(axis.title.x = element_blank())

# Combine TA plots
combined_ta_plot <- (p_roc / p_bbands) | (p_macd / p_rsi)

combined_ta_plot + plot_annotation(
  title = "Technical Analysis Indicators (Last 100 Candles)",
  theme = theme(plot.title = element_text(hjust = 0.5, size = 16))
)
```




```
combined_ta_plot
```



TODO When doing the real data don't forget to rm na