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 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.

1.4 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 100000 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.

```
candles <- read_csv(paste0("data/", trading_pair, "_candles_",
    start_date, "_", end_date, "_", candlestick_period, ".csv"))
```

```
## Rows: 10873 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (5): low, high, open, close, volume
## dttm (1): time
##
## 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.
```

```r
head(candles)
```

```
## # A tibble: 6 x 6
##   time                  low   high   open  close volume
##   <dttm>              <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 2024-01-01 00:00:00 42262. 42544. 42289. 42453.   379.
## 2 2024-01-01 01:00:00 42415  42750. 42454. 42595.   396.
## 3 2024-01-01 02:00:00 42488. 42626. 42595. 42571.   227.
## 4 2024-01-01 03:00:00 42235  42581. 42571. 42325.   306.
## 5 2024-01-01 04:00:00 42200  42393. 42325. 42390.   296.
## 6 2024-01-01 05:00:00 42176. 42396. 42390. 42231.   188.
```

```r
count_candles <- nrow(candles)
```

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/

```r
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.
```

```r
fear_and_greed_index <- fear_and_greed_index %>%
    mutate(value = as.numeric(value))
head(fear_and_greed_index)
```

```
## # A tibble: 6 x 3
##   value value_classification timestamp
##   <dbl> <chr>                <dttm>
## 1    26 Fear                 2025-03-29 00:00:00
## 2    44 Fear                 2025-03-28 00:00:00
```

```
## 3    40 Fear                2025-03-27 00:00:00
## 4    47 Neutral             2025-03-26 00:00:00
## 5    46 Fear                2025-03-25 00:00:00
## 6    45 Fear                2025-03-24 00:00:00
```

```
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"))
head(hash_rate)
```

```
##    timestamp hash_rate
## 1 2024-01-01 501122294
## 2 2024-01-02 509303882
## 3 2024-01-03 505213088
## 4 2024-01-04 520042217
## 5 2024-01-05 545098332
## 6 2024-01-06 538450791
```

```
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"))
head(average_block_size)
```

```
##    timestamp avg_block_size
## 1 2024-01-01       1.653640
## 2 2024-01-02       1.718455
## 3 2024-01-03       1.771466
## 4 2024-01-04       1.782402
## 5 2024-01-05       1.774551
## 6 2024-01-06       1.847959
```

```
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"))
head(n_transactions)
```

```
##    timestamp n_transactions
## 1 2024-01-01         657752
## 2 2024-01-02         367319
## 3 2024-01-03         502749
```

```
## 4 2024-01-04         482557
## 5 2024-01-05         420884
## 6 2024-01-06         382140
```

```
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))
head(utxo_count)
```

```
##    timestamp utxo_count
## 1 2024-01-01  135771648
## 2 2024-01-01  135985967
## 3 2024-01-02  136204295
## 4 2024-01-03  136425266
## 5 2024-01-03  136647884
## 6 2024-01-04  136871780
```