

TECH ALPHARETTA

2972 Webb Bridge Rd, Alpharetta, GA 30009

**Project #11: Bitcoin Time Series**

This project will test your comfort with time series data. It will also test your ability to structure a fairly complex analysis ([walk-forward analysis](https://en.wikipedia.org/wiki/Walk_forward_optimization)).

**Tip:** You don’t need to know much about finance or cryptocurrencies to complete this challenge. This challenge is a great example of why clear, structured thinking is important.

**Background**

Cryptocurrencies have become a widely debated topic over the last several years. Technologists point to the vast disruptive potential of blockchain and traders look for new profit opportunities, while skeptics and regulators approach the topic with a more measured view.

We are not here to enter the debate, but rather to learn from the rich datasets produced by cryptocurrency. With cryptocurrencies, you not only have the price data you get with other asset classes, but you also get interesting time series data such as “mining difficulty” and “average block size.”

**Data**

We have one table called **bitcoin\_time\_series.csv**, and it contains 3 years of daily time series data for 17 Bitcoin indicators.



***Data Dictionary:***

* **avg\_block\_size** – Average block size in MB.
* **avg\_trx\_cost** – Average transaction cost in USD.
* **confirmation\_time** – Median time for a transaction to be accepted.
* **difficulty** – [Mining difficulty](https://bitcoinwisdom.com/bitcoin/difficulty).
* **hash\_rate** – Bitcoin network’s computing power in TH.
* **market\_cap** – Total value of bitcoin supply in circulation in USD.
* **price\_usd** – Market-weighted price in USD.
* **total\_block\_size** – Total size of Bitcoin blockchain in MB.
* **total\_miner\_revenue** – Total revenue paid to miners in USD.
* **total\_supply** – Total number of bitcoins in circulation.
* **total\_trx\_fees** – Total transaction fees paid to miners in USD.
* **trade\_volume** – Total value of daily trading volume in USD.
* **transactions** – Daily Bitcoin transactions.
* **trx\_cost\_percent** – Transaction costs as percentage of volume.
* **trx\_per\_block** – Average number of transactions per block.
* **trx\_volume** – Daily total value of Bitcoin transactions in USD.
* **unique\_addresses** – Total number of unique Bitcoin addresses.

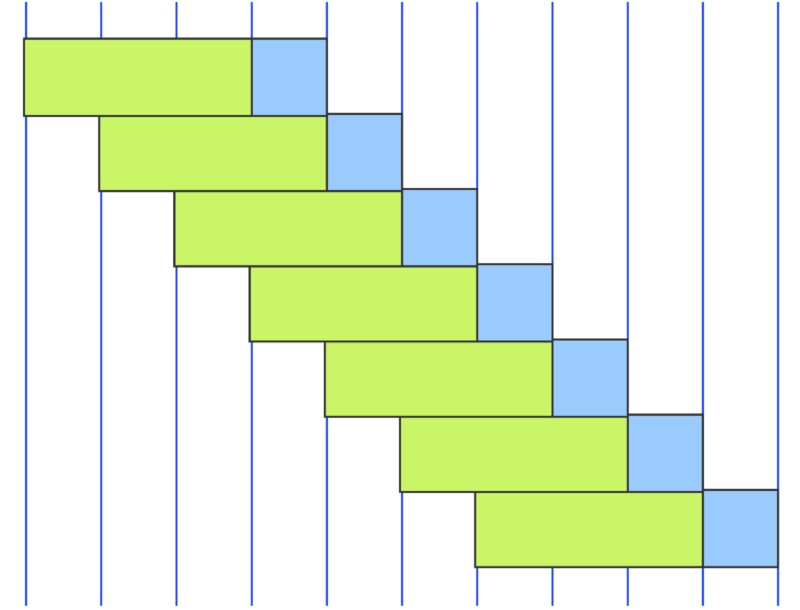
***Note****: You can find primary sources for crypto time series at*[*Blockchain.info*](https://blockchain.info/)*,*[*CoinMarketCap.com*](https://coinmarketcap.com/)*,*[*BitInfoCharts.com*](https://bitinfocharts.com/)*, or many other publicly-available sites..*

**Objectives**

Our goal is to build a model to predict whether the price of Bitcoin will grow by over 2% over the next day using **only** changes in its indicators from the previous day. In other words, if we are at time *t*, then our goal is to predict whether *y* = (*t*+1)/*t*> 1.02 using only data from time *t*-1 to time *t*.

* Perform a full walk-forward analysis for 2017. At the start of every month in 2017, train three different predictive models to predict *y*. The three models should be:
  + (1) A model trained on just previous month of data.
  + (2) A model trained on the previous 3 months of data.
  + (3) A model trained on all of the data you have up til then.
  + **Tip:** To keep things simple and straightforward, each model should be trained using the same algorithm and hyper-parameters.
* For each window in the walk-forward analysis, evaluate all three models. For example, for the window that starts January 1st, 2017, the test set would include all of the days in January 2017. You would train (1) a model using only data from December 2016 (2) a model using data from October, November, and December 2016 and (3) a model using all available data before January 1st, 2017.
* **Tip:**If you’re given this type challenge and discover your models are not predictive, don’t panic! Employers are looking for the way you structure your analysis and whether you can avoid errors… they are ***not*** expecting you to build a profitable trading algorithm in under 4 hours!

A **walk-forward analysis** uses moving windows for training and test sets to evaluate the robustness and performance of a modeling methodology over time. For example, for method #2 above, the training (green) and test (blue) windows would look like this:



Each vertical line represents the start date of a new month.