Francesco Scolz

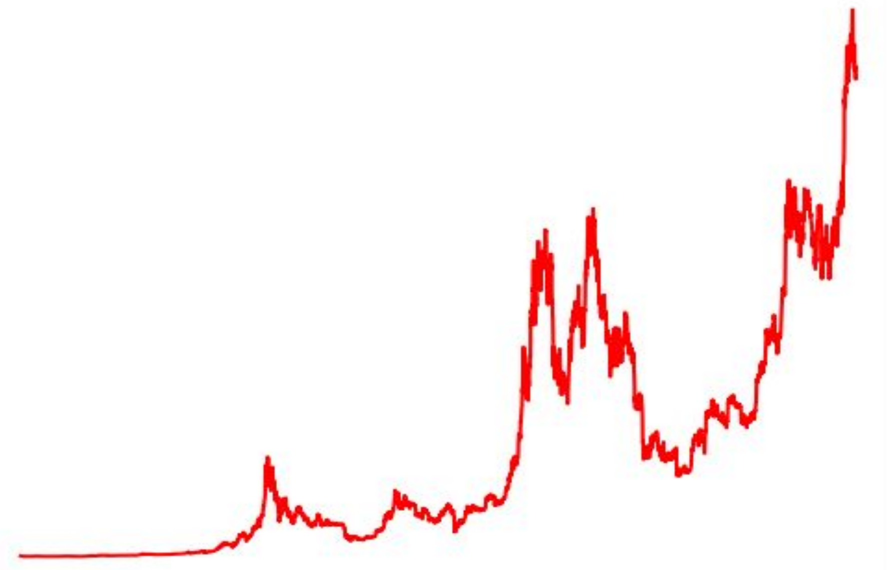
Attimis, 30/08/2025

Bitcoin Price Predictor

How machines can still fail

Goal of the project

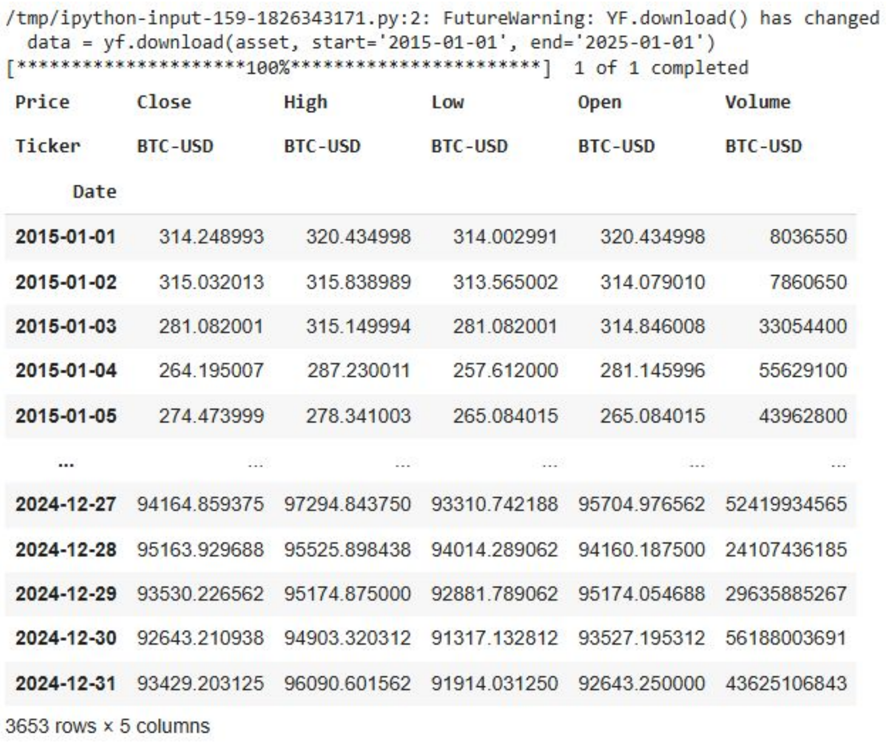
We want to create a **predictive model** of **Bitcoin prices** that can **reliably** predict **daily price movements**



Why solve this problem?

Having such **powerful market knowledge** and **tools** at hand can be especially useful for **finance businesses**, **traders** and **investing funds**

Finding and importing the data



```
/tmp/ipython-input-159-1826343171.py:2: FutureWarning: YF.download() has changed
  data = yf.download(asset, start='2015-01-01', end='2025-01-01')
[*********************100%***********************]  1 of 1 completed
```

Price	Close	High	Low	Open	Volume
Ticker	BTC-USD	BTC-USD	BTC-USD	BTC-USD	BTC-USD
Date					
2015-01-01	314.248993	320.434998	314.002991	320.434998	8036550
2015-01-02	315.032013	315.838989	313.565002	314.079010	7860650
2015-01-03	281.082001	315.149994	281.082001	314.846008	33054400
2015-01-04	264.195007	287.230011	257.612000	281.145996	55629100
2015-01-05	274.473999	278.341003	265.084015	265.084015	43962800
...	...	...	...	...	...
2024-12-27	94164.859375	97294.843750	93310.742188	95704.976562	52419934565
2024-12-28	95163.929688	95525.898438	94014.289062	94160.187500	24107436185
2024-12-29	93530.226562	95174.875000	92881.789062	95174.054688	29635885267
2024-12-30	92643.210938	94903.320312	91317.132812	93527.195312	56188003691
2024-12-31	93429.203125	96090.601562	91914.031250	92643.250000	43625106843

3653 rows × 5 columns

We used the **open-source API** from **Yahoo finance**

It contains **Prices** and **Volumes** for a large number of **traded assets**, including **Bitcoin**

We selected the **time range** that goes from **2015** to **2025**, and we **imported the data**

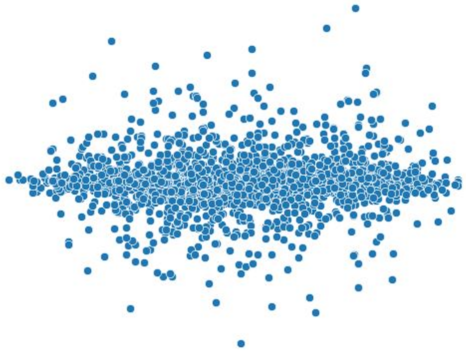
Extrapolating insights and features

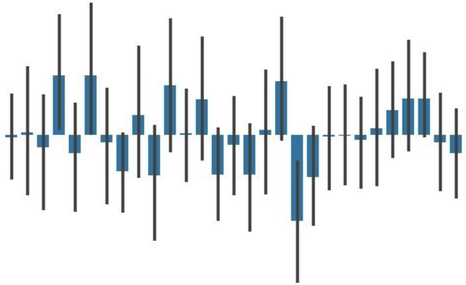
We then **extracted important indicators and features** from our data, making sure to get **the most** out of our **model**

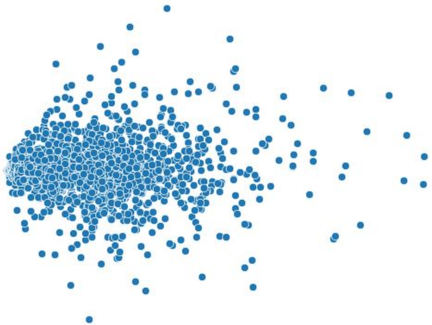
Examples include: **Weekday**, **Month**, **Relative Strength Index (RSI)**, and others

First analysis results

Our **first analysis** revealed how almost **every one** of these features **didn’t have an obvious correlation** with the **target**, so we concluded that **complex models** were surely needed





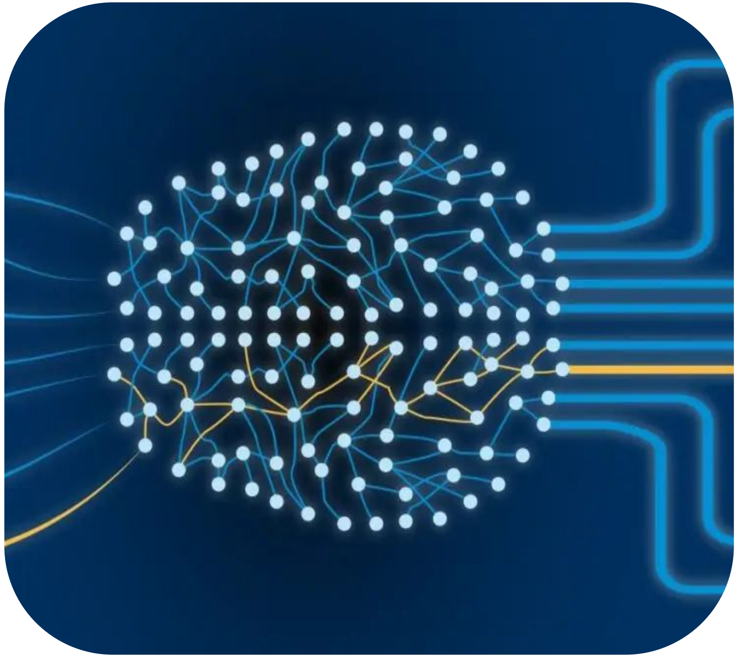


How we modeled | The choices

We created **4 different models**:

- A **simple** one
- A **moderately complex** one
- A **complex** one
- And an **extremely complex** one

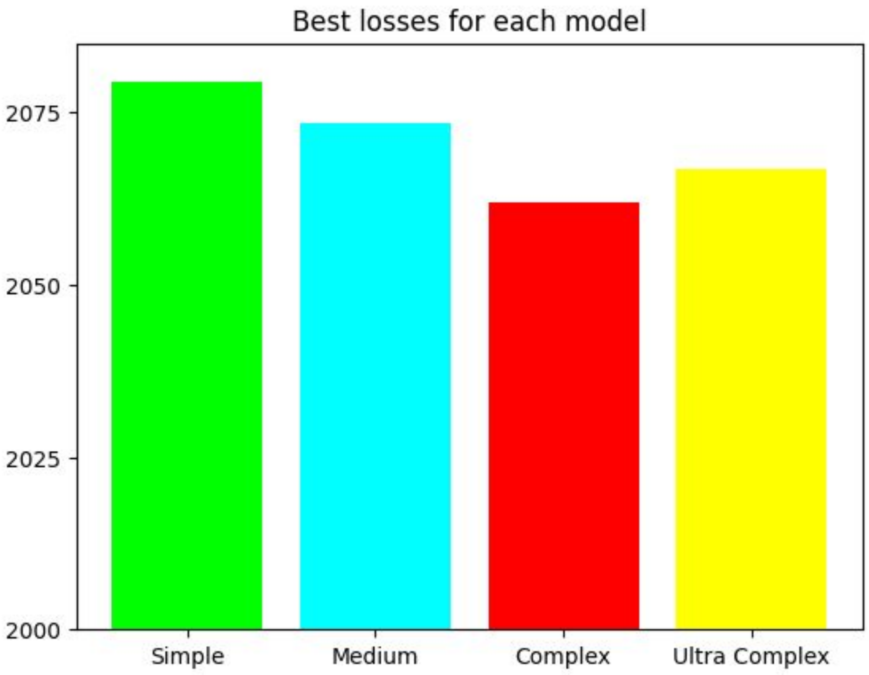
The **goal** was to **understand** what **level of complexity** would **perform better** on a test, returning the lowest error and **highest accuracy**



How we modeled | The results

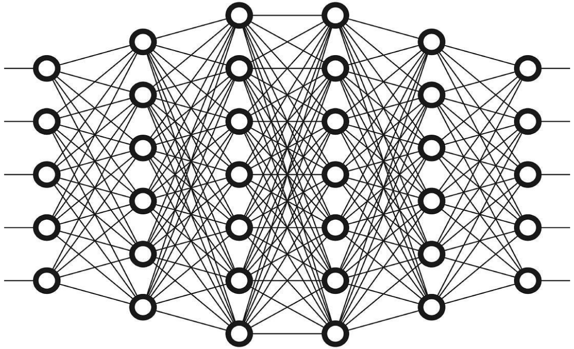
In the end, the one with the **lowest error** was the **complex one**, while only **predicting** in a **narrow range**, causing the error to be **still very high** and **not reliable enough** to predict every **price fluctuation**

This could be because of the **enormous price range** that **Bitcoin** had in the past **10 years**



Conclusions

Even though we tried **extremely complex and adaptive models**, we **couldn't** produce an **accurate model** to **predict Bitcoin prices** in the future



Possibly because of **lack of samples**, **features** or **simple intrinsic randomness**

Conclusions

This project **doesn’t preclude future work** on the same **goal,** so be sure to follow my GitHub and Linkedin pages to stay **updated** with the **latest works!**

Thank you for your attention

Sincerely,

Scolz F.