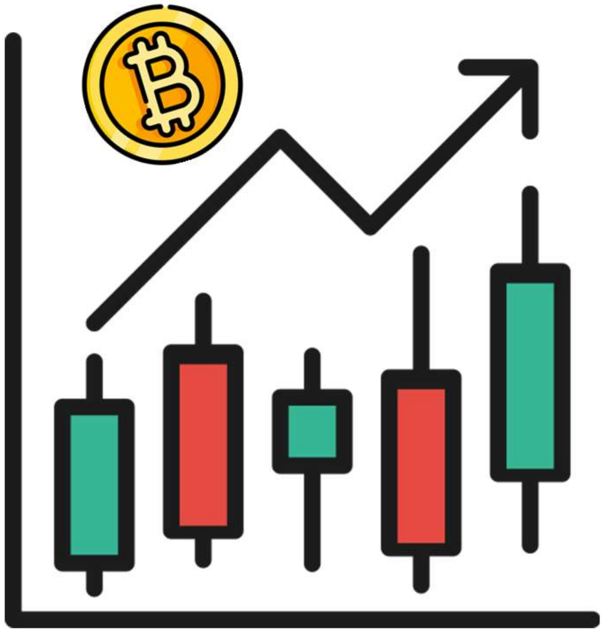


Bitcoin price forecasting

Big Data Computing Project

A.Y. 2022 - 2023

Faculty of Ingegneria dell'informazione, informatica e statistica
Department of Informatica

Danilo Corsi
Matr. 1742375

Outline

- **Introduction**
  - What is bitcoin?
  - Goal of the project

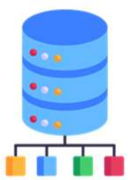
Outline

- **Introduction**
  - What is bitcoin?
  - Goal of the project

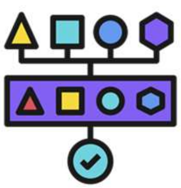
- **Dataset and features**
  - Data collection
  - Features engineering

Outline

- **Introduction**
  - What is bitcoin?
  - Goal of the project



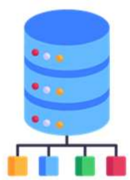
- **Dataset and features**
  - Data collection
  - Features engineering



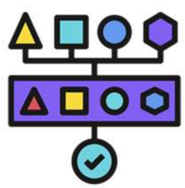
- **Project pipeline**
  - Data crawling / feature extraction
  - Models train / validation
  - Final scores

Outline

- **Introduction**
  - What is bitcoin?
  - Goal of the project



- **Dataset and features**
  - Data collection
  - Features engineering



- **Project pipeline**
  - Data crawling / feature extraction
  - Models train / validation
  - Final scores

- **Conclusions**

Introduction

- **What is Bitcoin?**
  - Decentralized cryptocurrency
  - No central bank behind it
  - Relies on a network of nodes
  - **Transactions**
    - Uses strong cryptography (validity and security)
    - Made by anyone with a Bitcoin address
    - Public ledger constantly updated

Introduction

- **What is Bitcoin?**
  - Decentralized cryptocurrency
  - No central bank behind it
  - Relies on a network of nodes
  - **Transactions**
    - Uses strong cryptography (validity and security)
    - Made by anyone with a Bitcoin address
    - Public ledger constantly updated

- Value determined by the market and the number of people using it
- Price fluctuation can be extremely unpredictable
- **Prediction of Bitcoin prices can be a competitive advantage**

Goal

Analyze machine learning techniques

↓

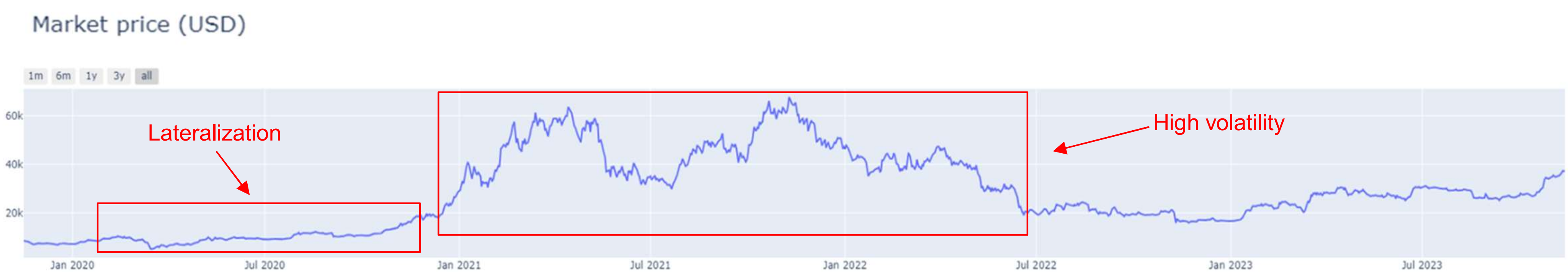
Understand how accurately the price of Bitcoin can be predicted

↓

Can provide added value to cryptocurrency investors and traders?

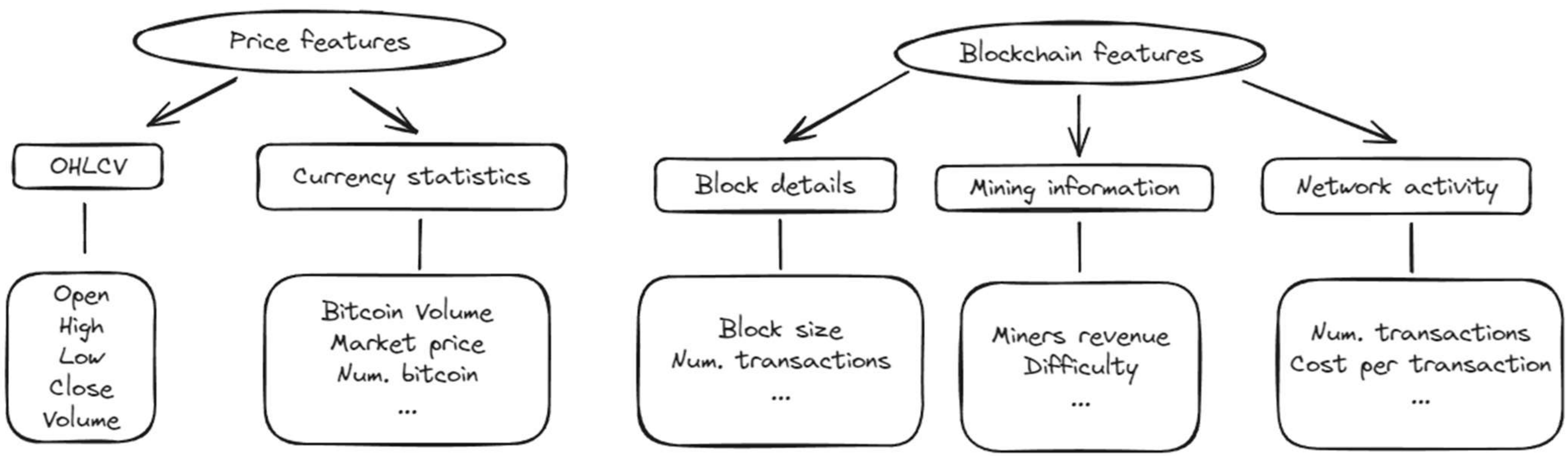
Dataset and features

- **Collecting Bitcoin data:**
  - **Blockchain.org** (for blockchain data)
  - **Binance** and **Kraken** exchanges (for price information)
- Data organized in 15-minute time-frame
- Retrieving the most relevant information from the last four years to current days



Dataset and features

- **Collecting Bitcoin data:**
  - **Blockchain.org** (for blockchain data)
  - **Binance** and **Kraken** exchanges (for price information)
- Data organized in 15-minute time-frame
- Retrieving the most relevant information from the last four years to current days



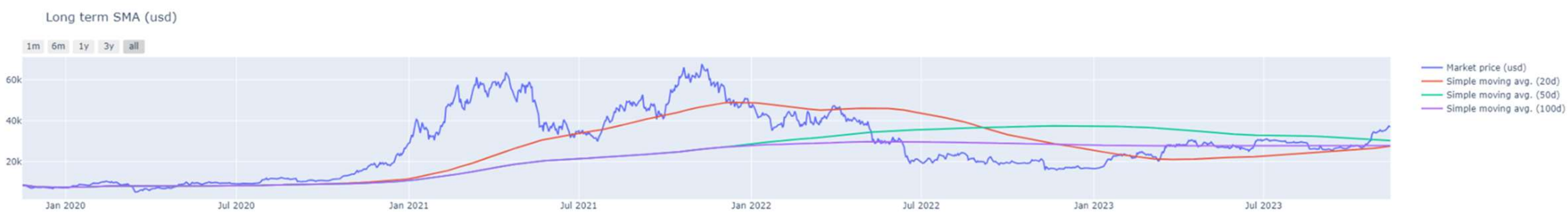
Project pipeline

- **Structure:**
  1. **Data crawling / Feature engineering:** retrieve and process data
  2. **Models' train / validation:** different models and splitting methods
  3. **Final scores:** collect results and draw conclusions

Project carried out with **Apache Spark** (during some phases I converted the Spark dataframe to a Pandas one to make some plots)

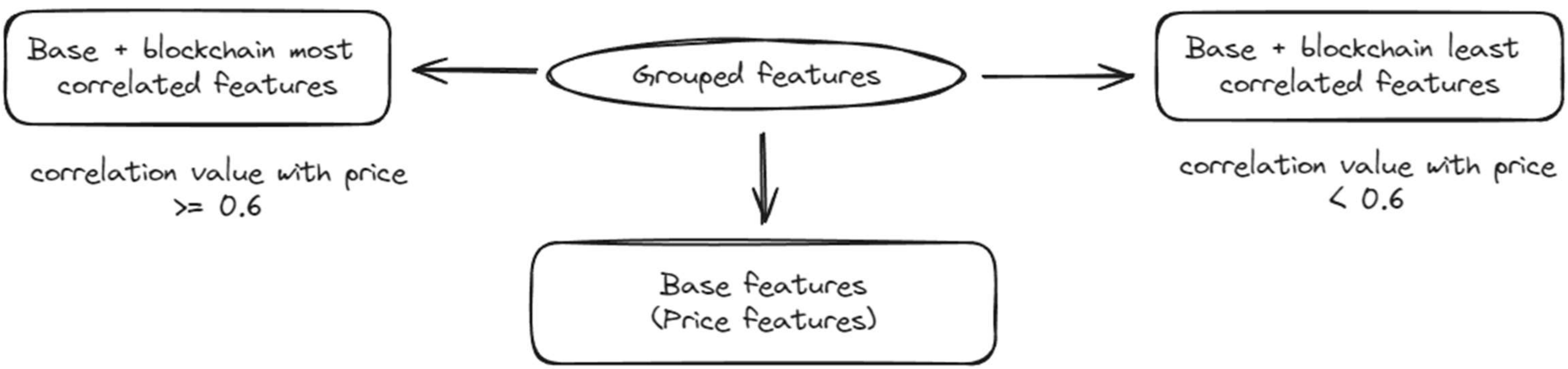
1 - Data crawling / Feature engineering: features

- **Additional features**
  - **Next market price:** next-15 minutes Bitcoin price (will be the **target variable**)
  - **Simple moving avg:** average price over a specified number of days



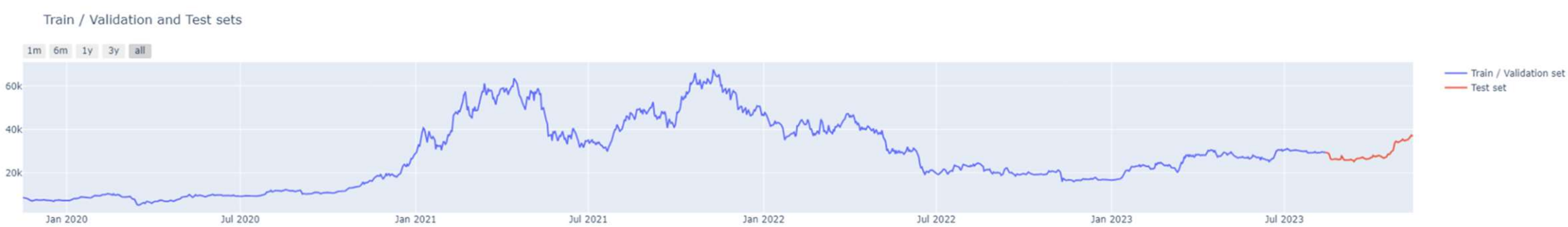
1 - Data crawling / Feature engineering: features

- **Additional features**
  - **Next market price:** next-15 minutes Bitcoin price (will be the **target variable**)
  - **Simple moving avg:** average price over a specified number of days

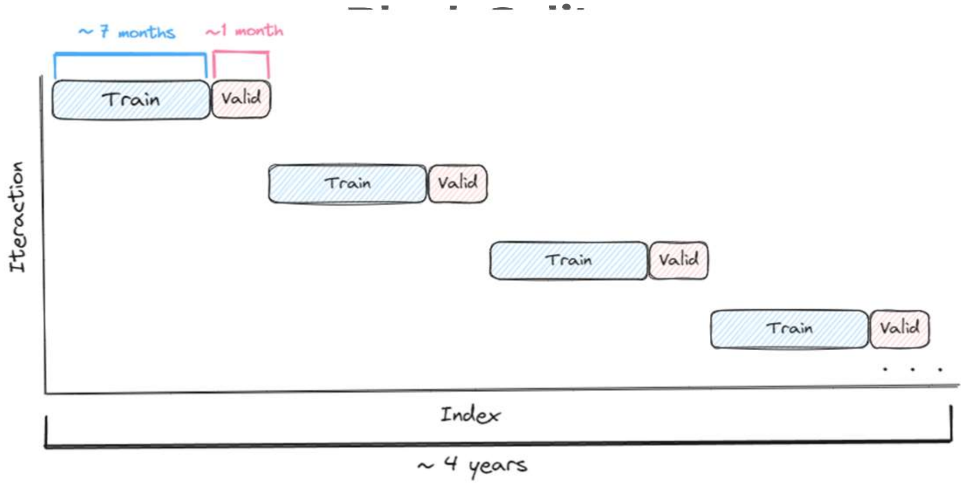


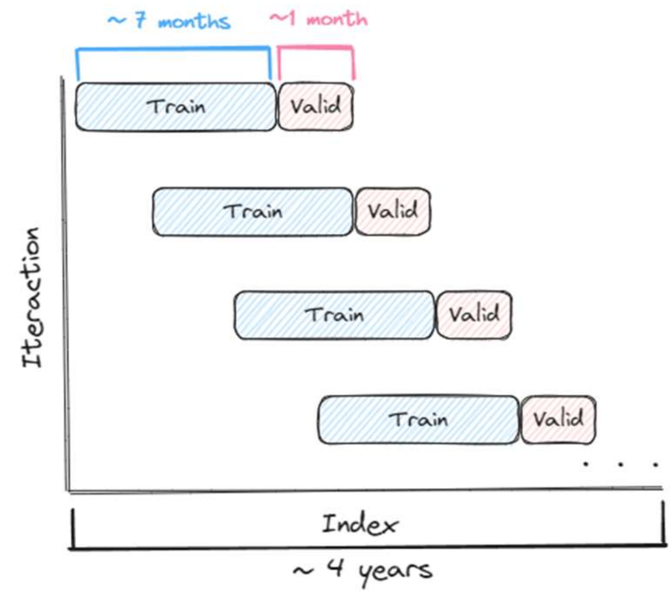
1 - Data crawling / Feature engineering: splitting

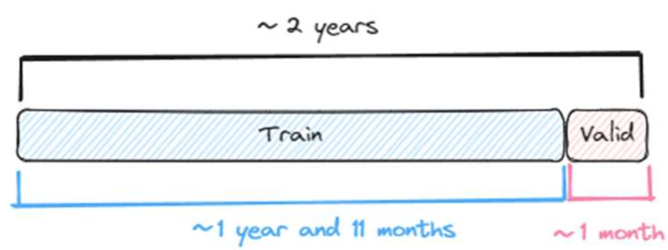
- **Two sets:**
  - **Train / Validation set:** used to train and validate models
  - **Test set:** used to perform price prediction on never-before-seen data (last 3 months of the original dataset will be used)



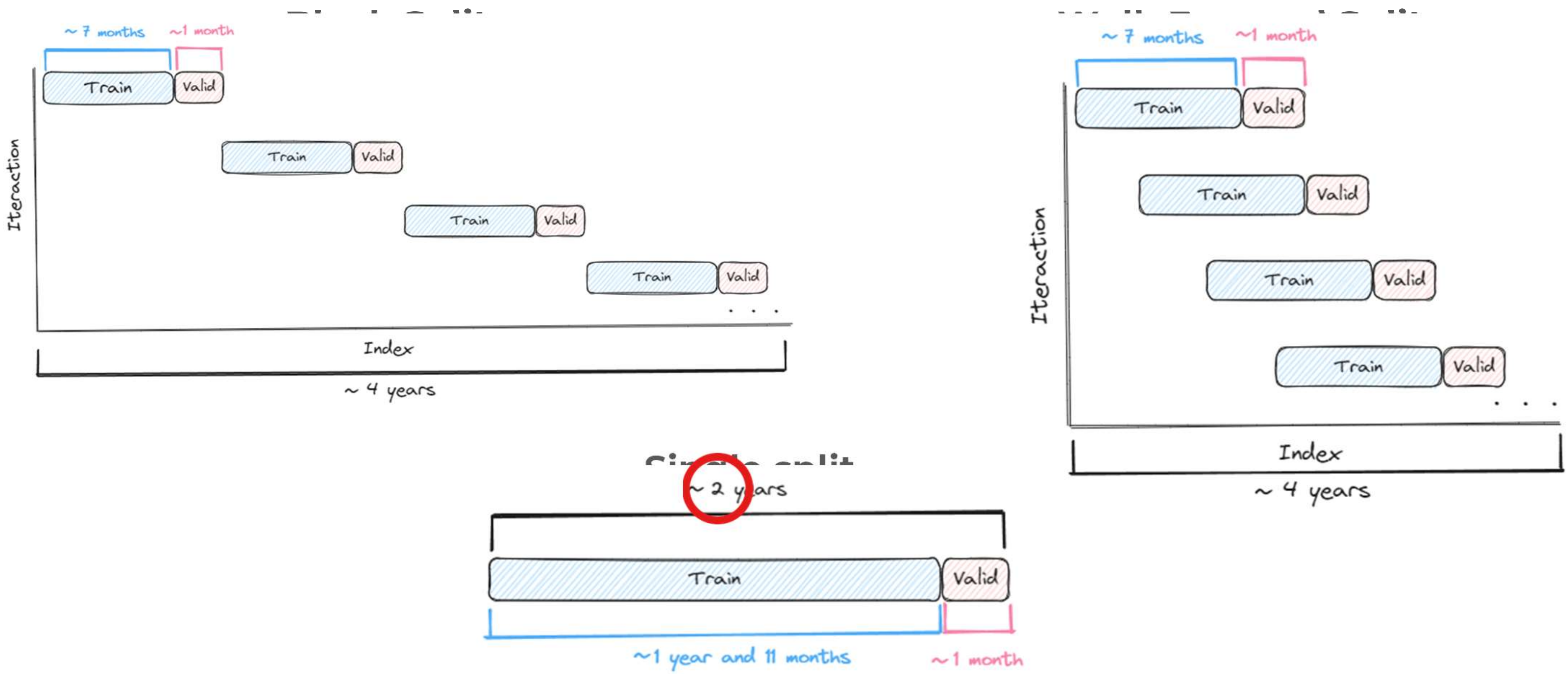
2 - Models train / validation: splitting methods





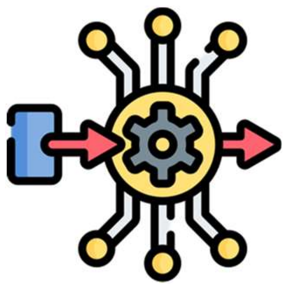


2 - Models train / validation: splitting methods



2 - Models train / validation: models and metrics

- **ML models:**
  - Linear Regression
  - Generalized Linear Regression
  - Random Forest Regressor
  - Gradient Boosting Tree Regressor



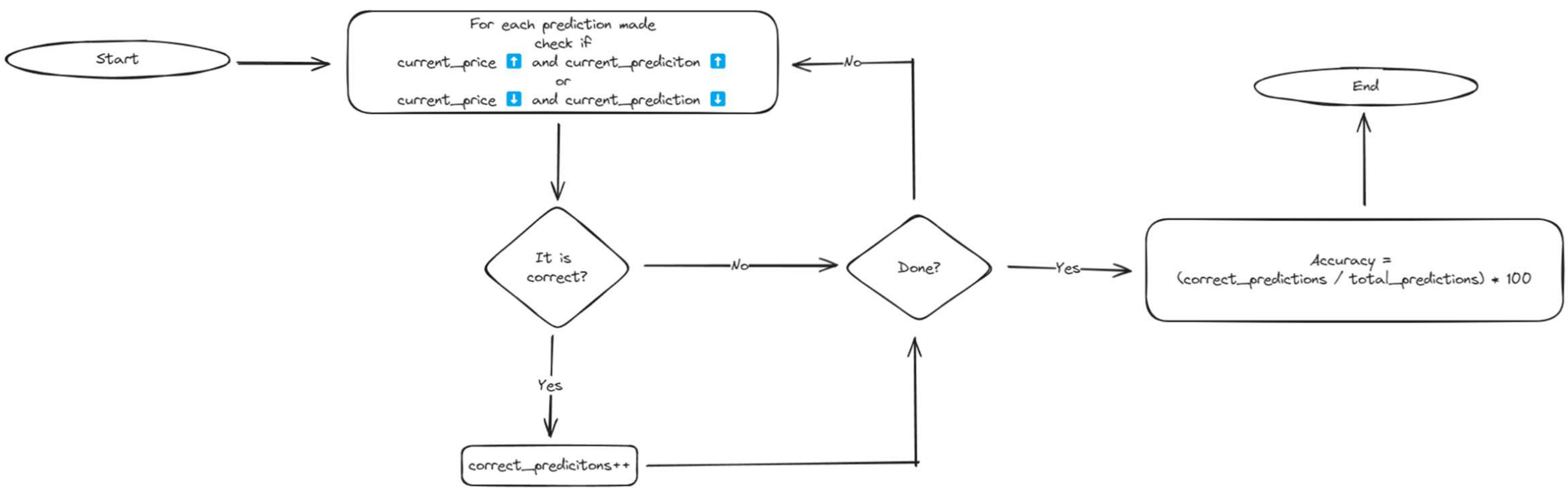
- **Metrics:**
  - RMSE (Root Mean Squared Error)
  - MSE (Mean Squared Error)
  - MAE (Mean Absolute Error)
  - MAPE (Mean Absolute Percentage Error)
  - R2 (R-squared)
  - Adjusted R2

2 - Models train / validation: accuracy

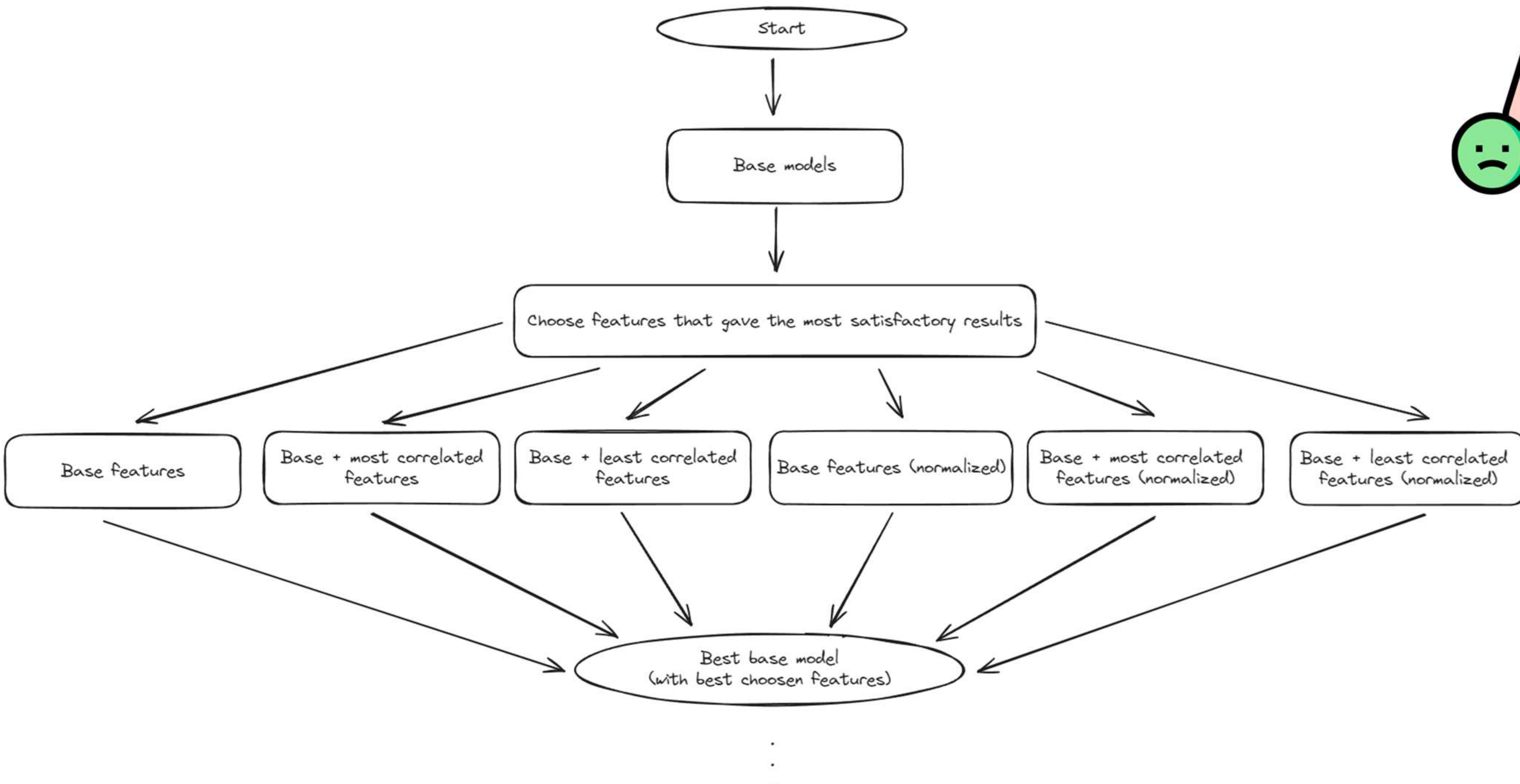
- “How good the models are at predicting whether the price will go up or down?”

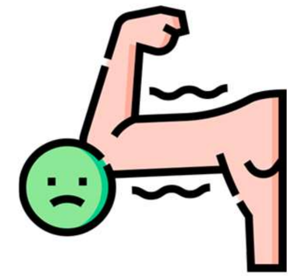
2 - Models train / validation: accuracy

- “How good the models are at predicting whether the price will go up or down?”

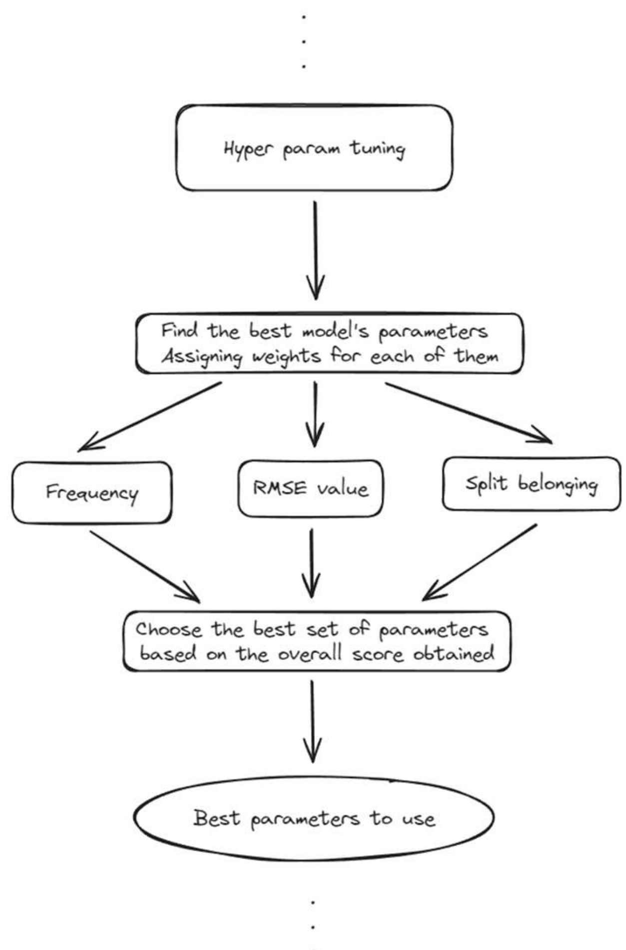


2 - Models train / validation: pipeline

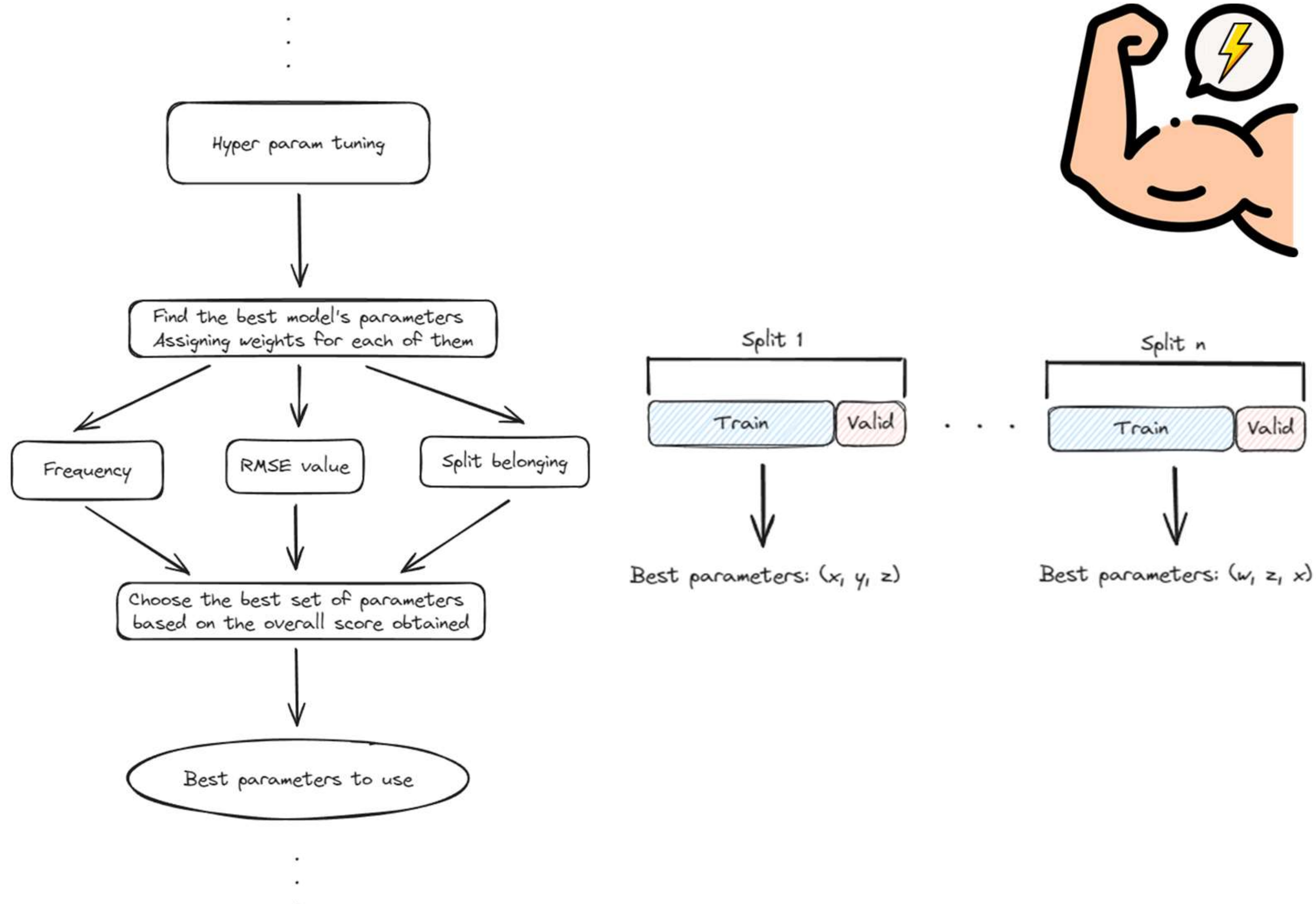




2 - Models train / validation: pipeline

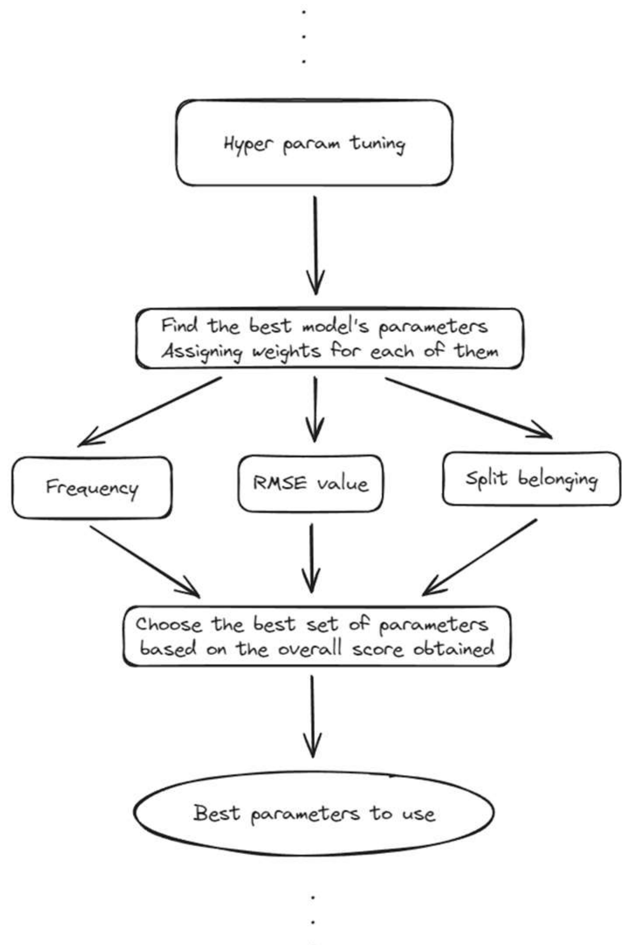


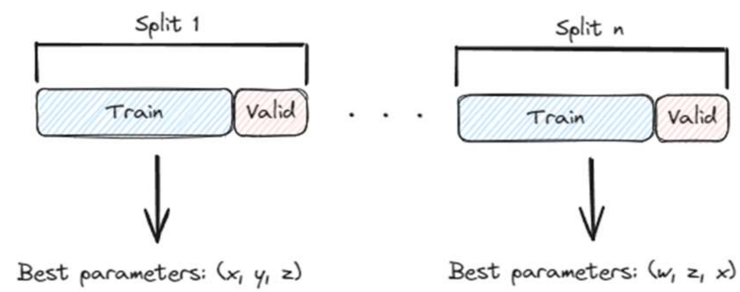
2 - Models train / validation: pipeline



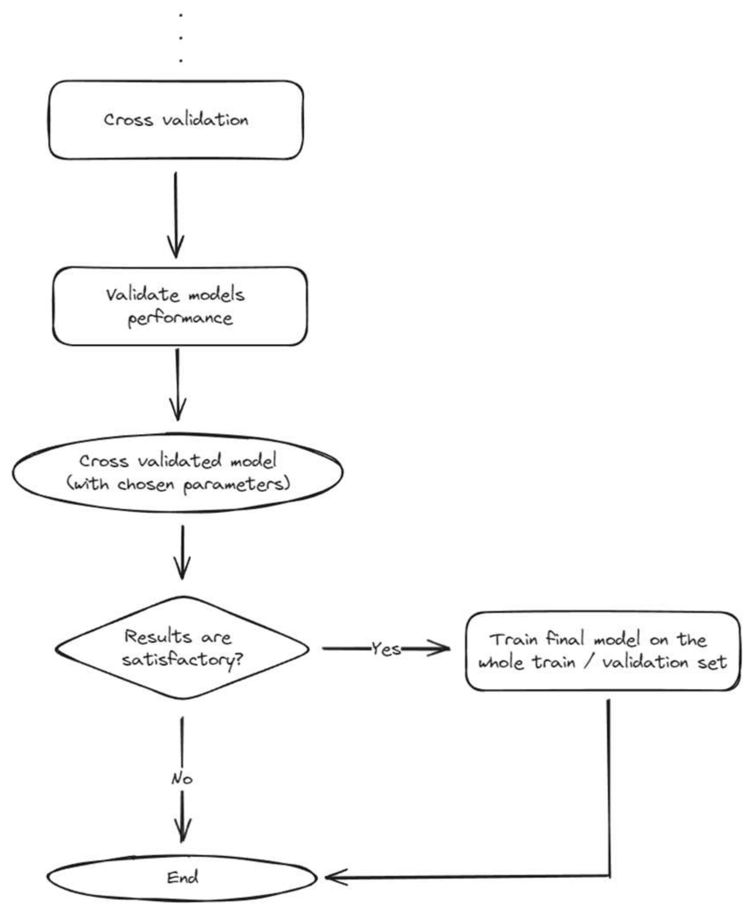
2 - Models train / validation: pipeline

Params.	Split	RMSE	Frequency	Score
(x, y, z)	0.75	0.90	1.0	0.67
...	...	...	...	...
(w, z, x)	0.68	0.69	0.50	0.23



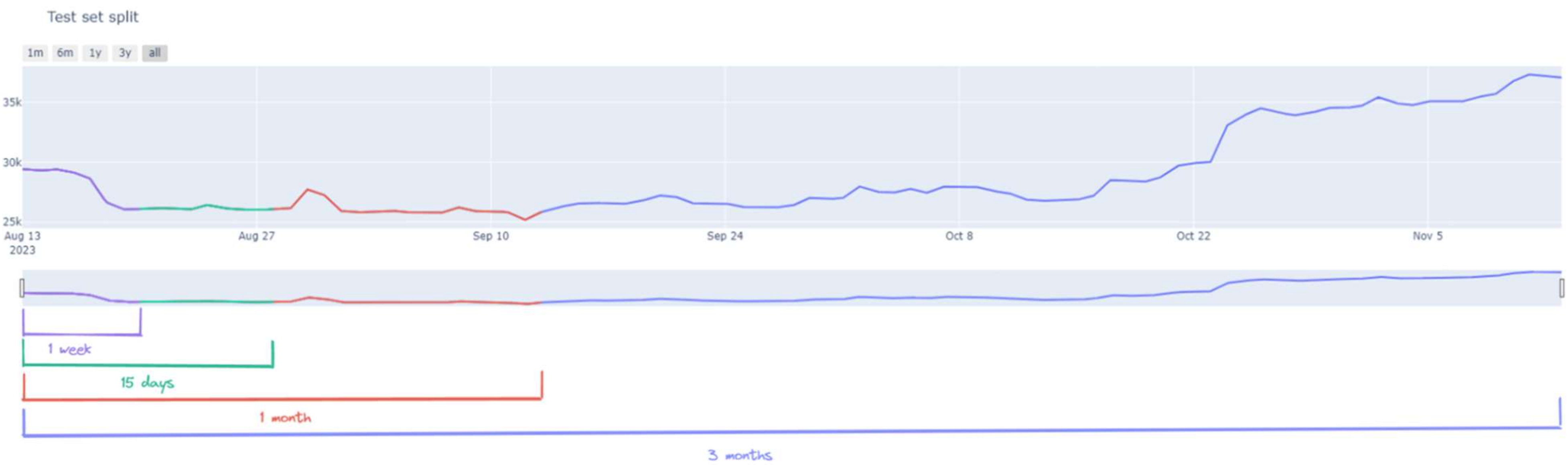


2 - Models train / validation: pipeline

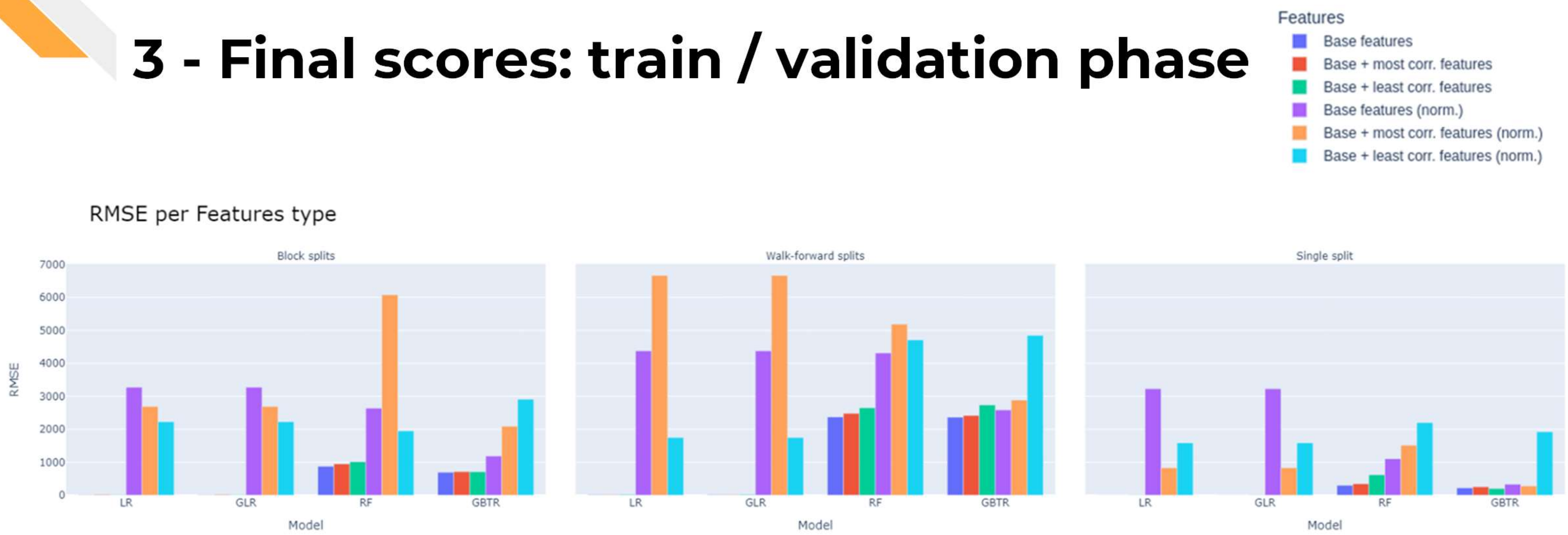


3 - Final scores

- **Comparison** between final results
- **Prediction** on the test set (splitted)
- See how models' performance **changes** as time increases



3 - Final scores: train / validation phase



- **Walk-forward splits** return lower performance than **Block splits** and **Single splits**
    - Benefiting from a shorter time horizon
- **Normalised** features produce suboptimal results (**high RMSE values**)
    - Benefits varies between models

3 - Final scores: train / validation phase

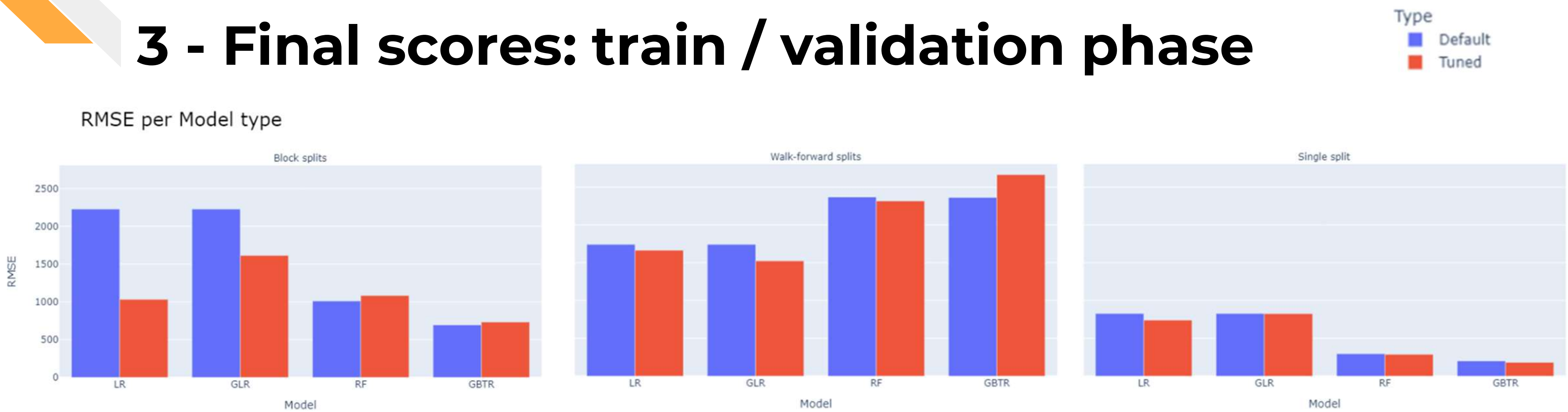
- Helps reduce **overfitting** but presents **problems** in other scenarios
- **Blockchain features** produces a modest improvements
  (persistent influence of **price-based** features)

3 - Final scores: train / validation phase

- Helps reduce **overfitting** but presents **problems** in other scenarios
- **Blockchain features** produces a modest improvements (persistent influence of **price-based** features)

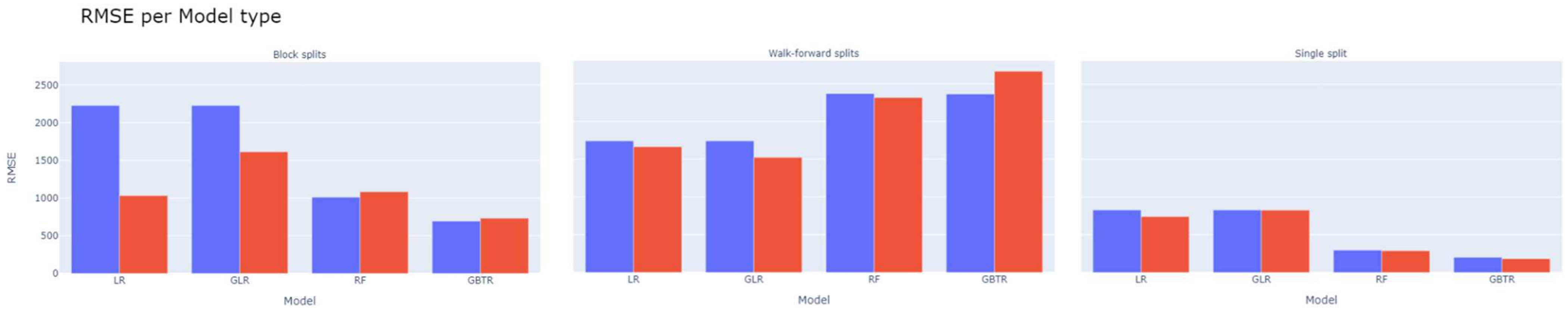
- **Chosen features**
  - **LR:** Base + most corr. (norm.)
  - **GLR:** Base + most corr. (norm.)
  - **RF:** Base (no norm.)
  - **GBTR:** Base + least corr. (no norm.)

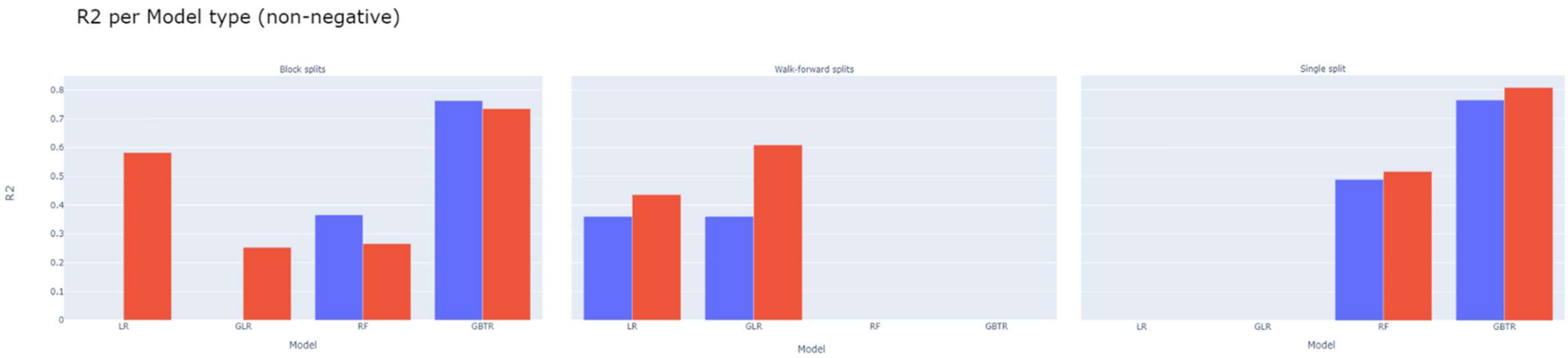
3 - Final scores: train / validation phase



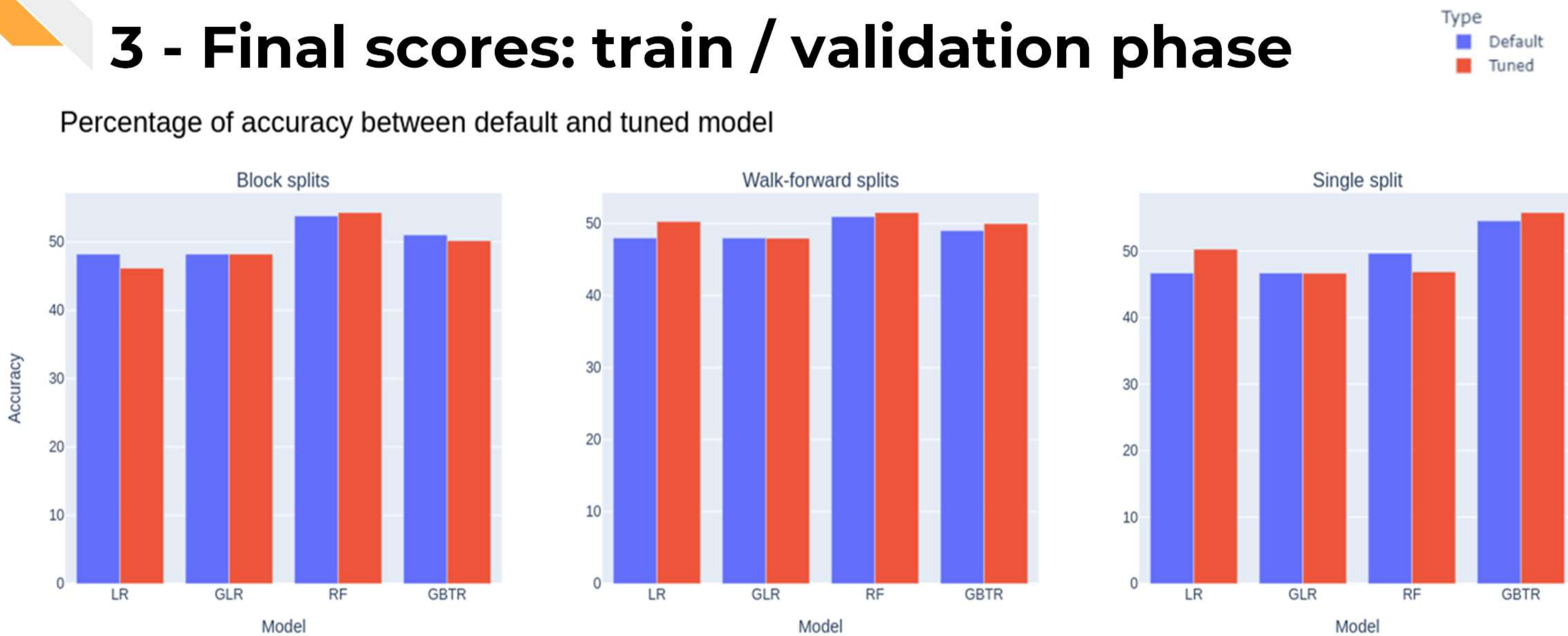
- **Single split** is the best method on which to train / validate the models
- **Hyper parameter tuning** brought some improvements
- **Tree-based model** returned the best results

3 - Final scores: train / validation phase



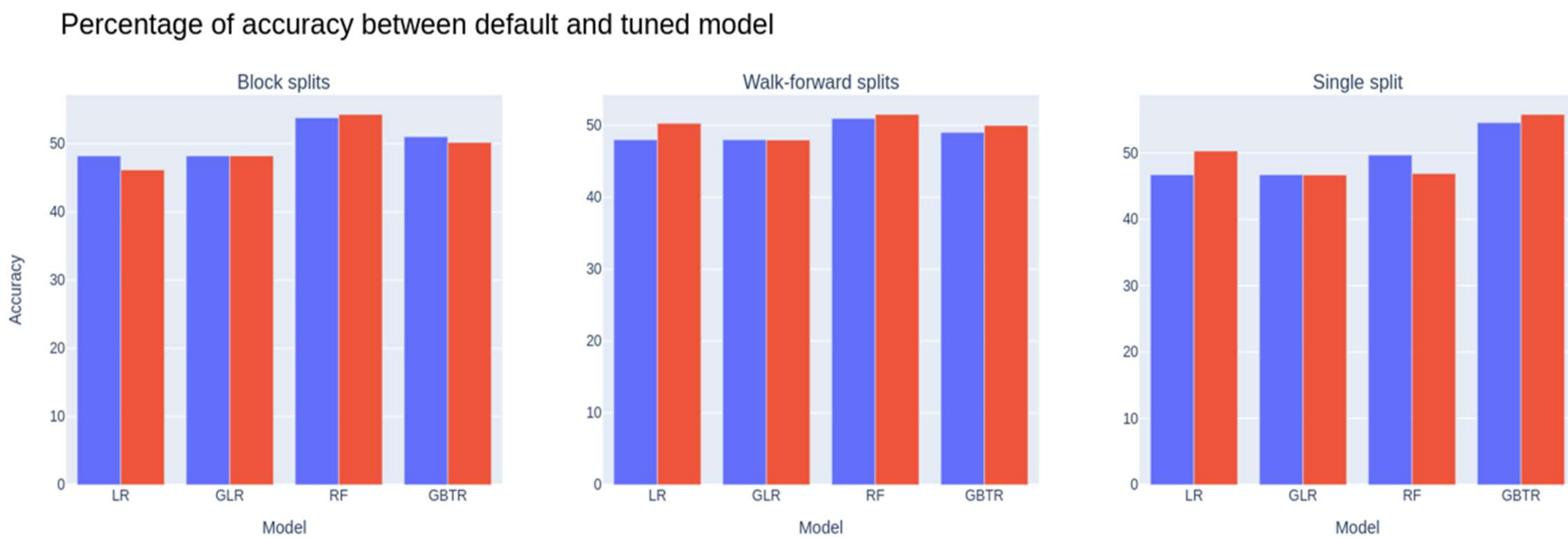


3 - Final scores: train / validation phase



- **Accuracy** has remained the same among (~50%)
- Probably due to the period taken into consideration being too long

3 - Final scores: train / validation phase



- **Accuracy** has remained the same among (~50%)
- Probably due to the period taken into consideration being too long

- **Conclusions**
  - **Best splitting method:** single split
  - **Best models type:** tree-based models

3 - Final scores: testing phase

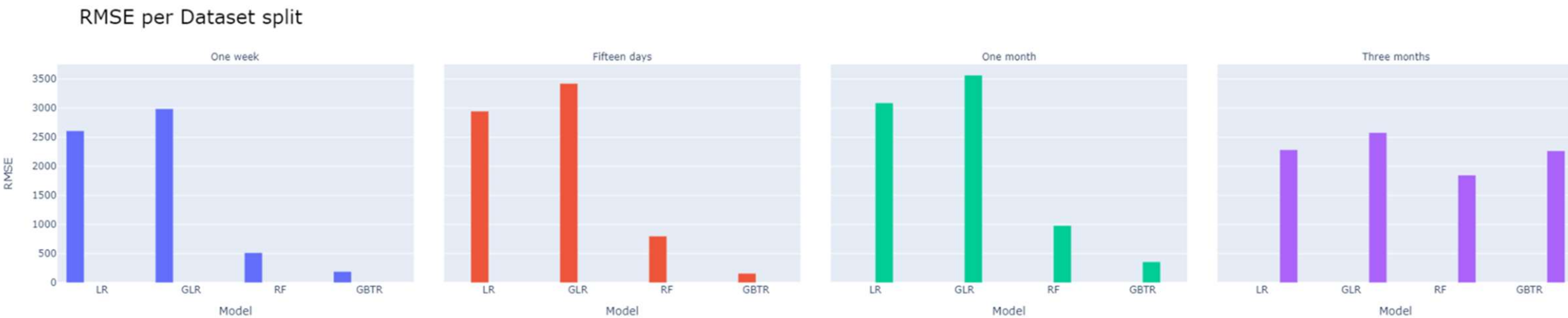
Short term: [one week, fifteen days]
Short-mid term: [one week, one month]
Long term: three months

3 - Final scores: testing phase

Short term: [one week, fifteen days]
Short-mid term: [one week, one month]
Long term: three months

3 - Final scores: testing phase

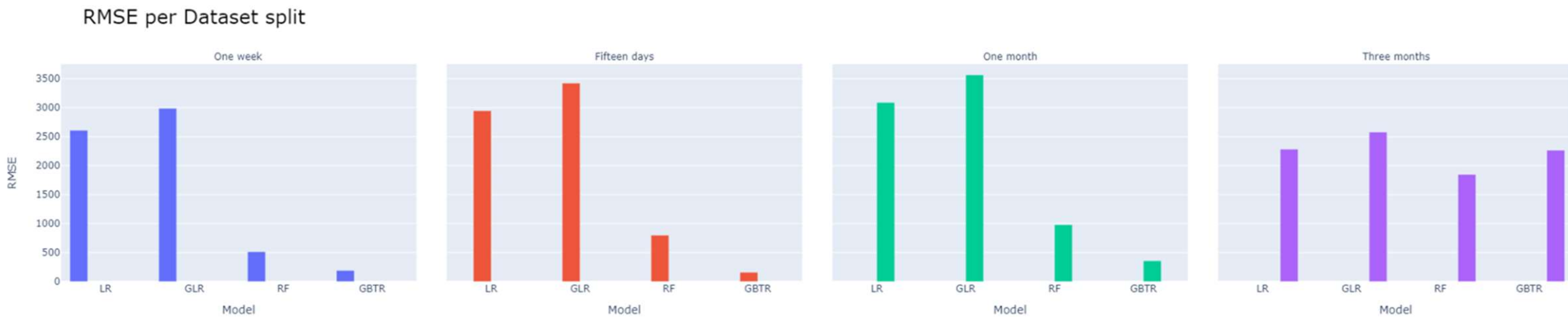
Short term: [one week, fifteen days]
Short-mid term: [one week, one month]
Long term: three months

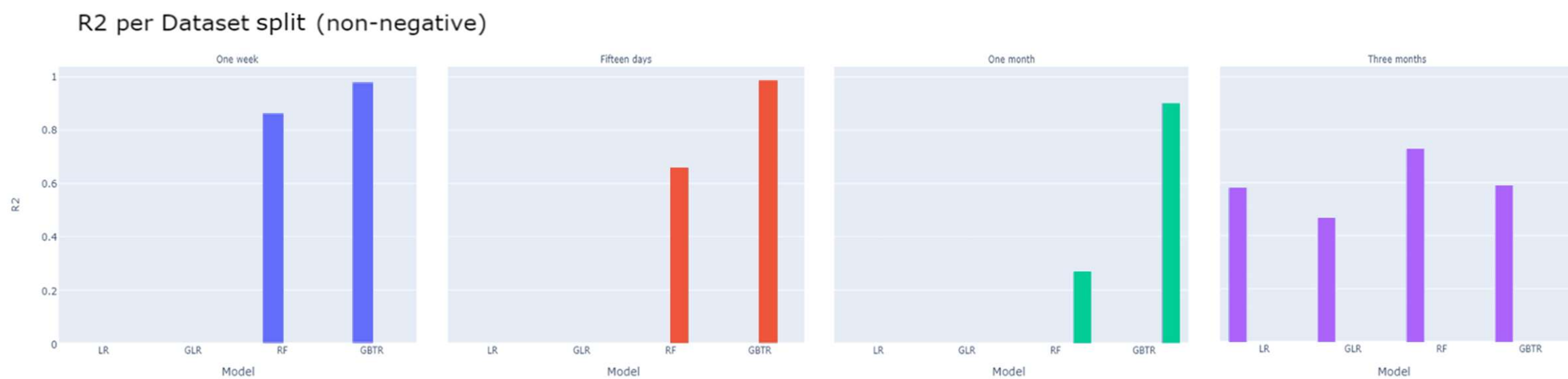


- As **time** taken into consideration **increase** also the **RMSE** values tends to **increase (slowly)**
- **Note**
  - Results were averaged
  - Having more data at each dataset split
  - Periods in which the models did **better (short-mid term)** compensated for the **worst** results in the last period **(long term)**

3 - Final scores: testing phase

Short term: [one week, fifteen days]
Short-mid term: [one week, one month]
Long term: three months





3 - Final scores: testing phase

Short term: [one week, fifteen days]
Short-mid term: [one week, one month]
Long term: three months

- **Higher** in the short-term
- **Lower** in the long-term

3 - Final scores: testing phase

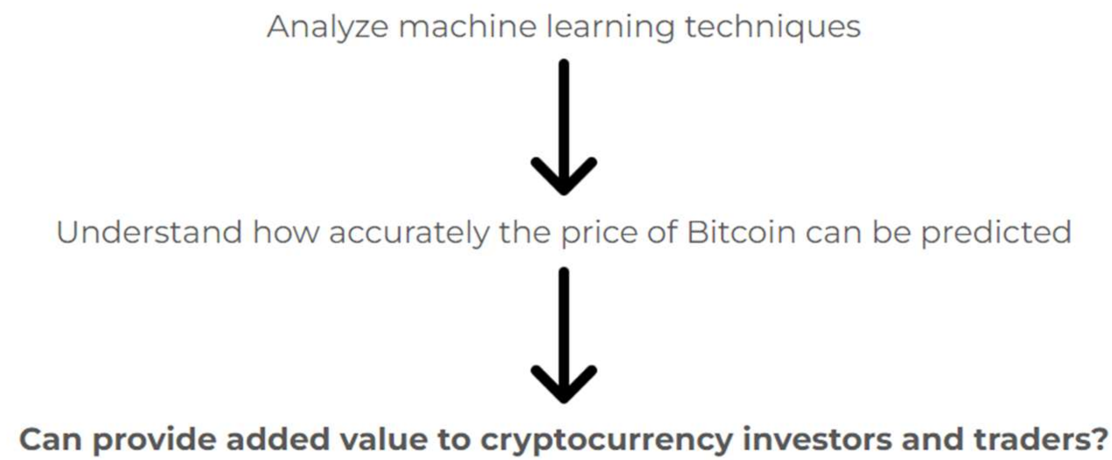
Short term: [one week, fifteen days]
Short-mid term: [one week, one month]
Long term: three months

- **Higher** in the short-term
- **Lower** in the long-term
- Linear models have a **higher accuracy** than tree-based models
    - Probably because because of the smoother curves

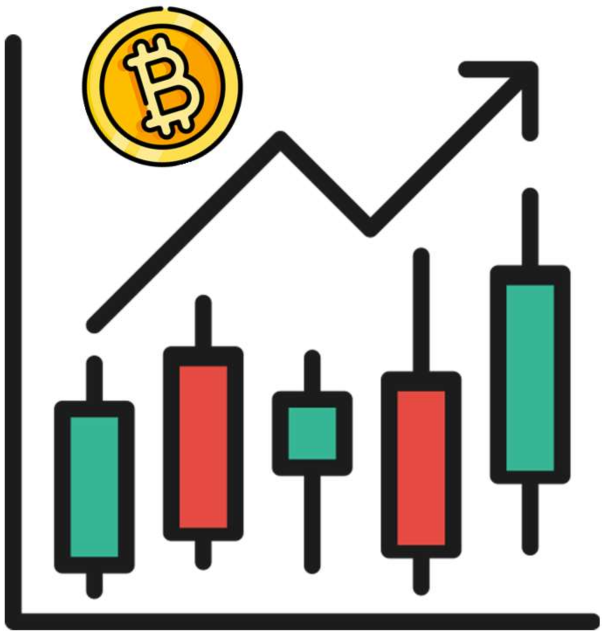
Conclusions

- **Splitting method**
  - Better those that consider a shorter period (e.g. Single Split)
- **Features**
  - Depend on the type of model
  - **In general:** blockchain-related features brought slight improvements
- **Models**
  - Better in the short-medium term (especially **tree-based models**)
  - As time period increase performance begins to degrade

Conclusions



- **Splitting**
  - Bett ... lit)
- **Features**
  - Dep
  - **In g** ... nprovements
- **Models**
  - Bett ... **nodels)**
  - As t
- **Answer to the initial question**
  - **Yes** (as far as the length of the period is concerned)
  - Better to consider a narrower forecast period for higher accuracy
- **Future developments**
  - Create a sliding window on features (additional historical data can be used)
  - Consider events that could influence the price
  - Using deep learning approaches such as CNNs or Transformers



Thanks for the attention

Danilo Corsi

https://github.com/CorsiDanil
o