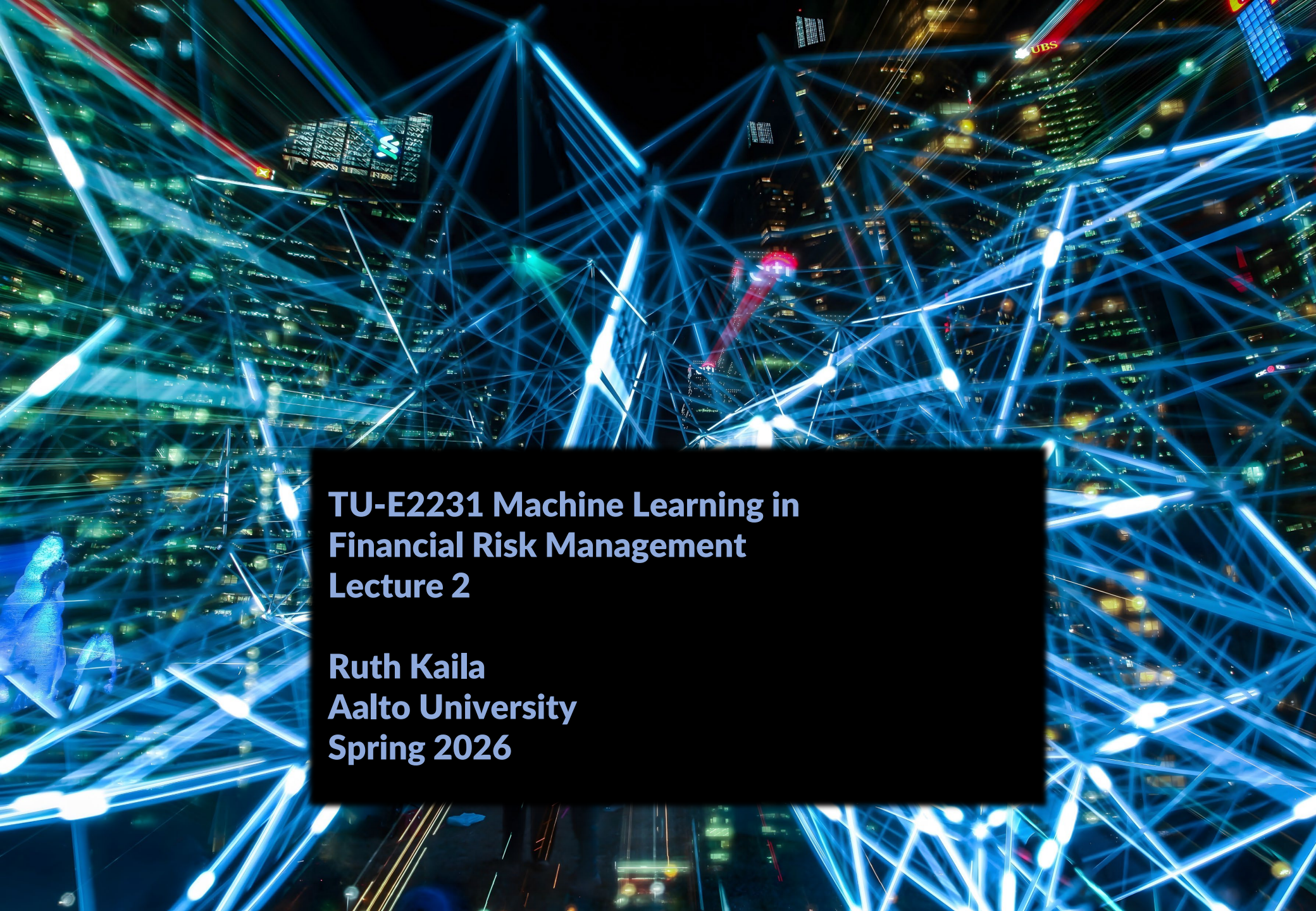
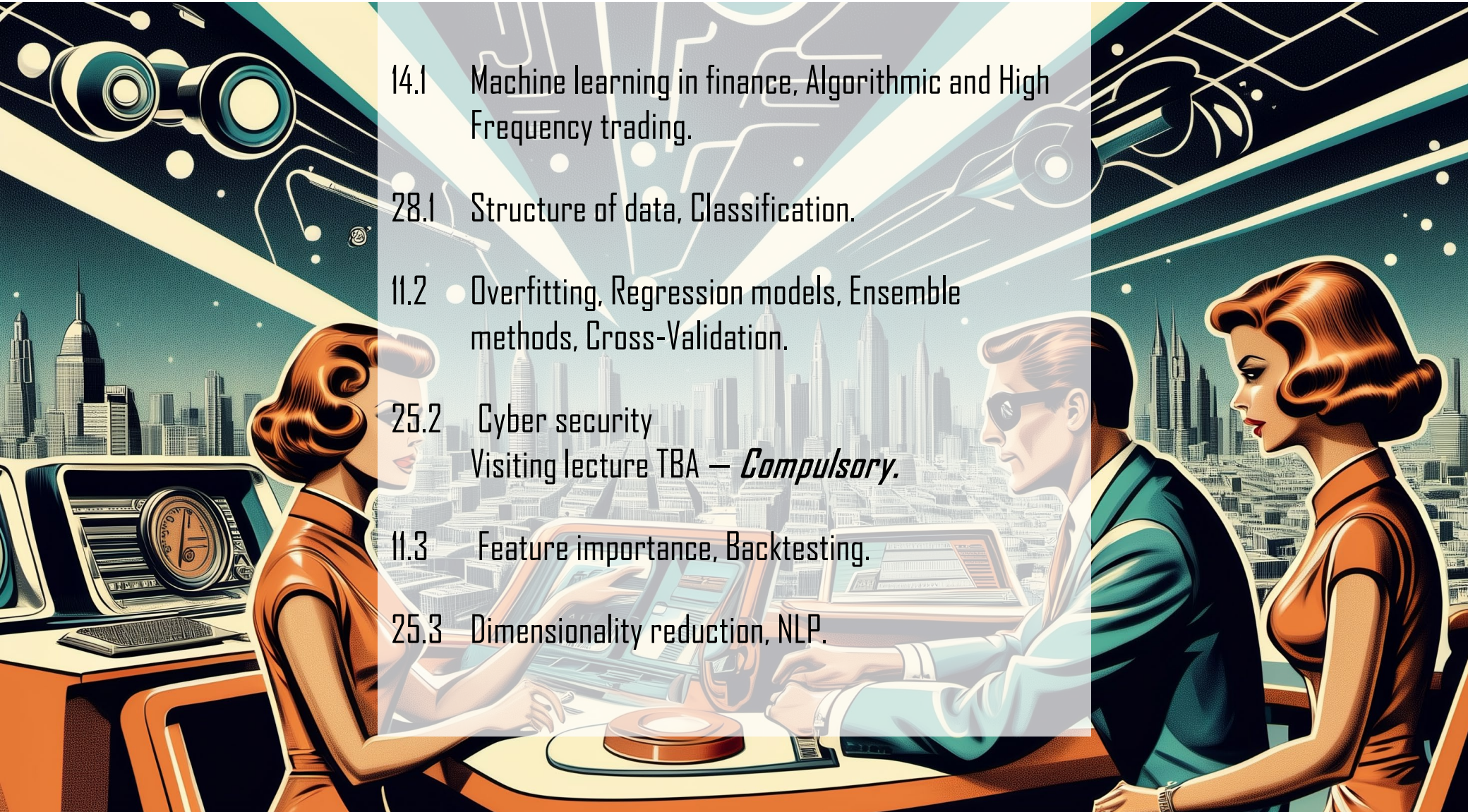


TU-E2231 Machine Learning in Financial Risk Management
Lecture 2

Ruth Kaila
Aalto University
Spring 2026

Schedule



14.1 Machine learning in finance, Algorithmic and High Frequency trading.

28.1 Structure of data, Classification.

11.2 Overfitting, Regression models, Ensemble methods, Cross-Validation.

25.2 Cyber security
Visiting lecture TBA — ***Compulsory.***

11.3 Feature importance, Backtesting.

25.3 Dimensionality reduction, NLP.

Today

Part 1, de Prado

- Data structures
- Labeling

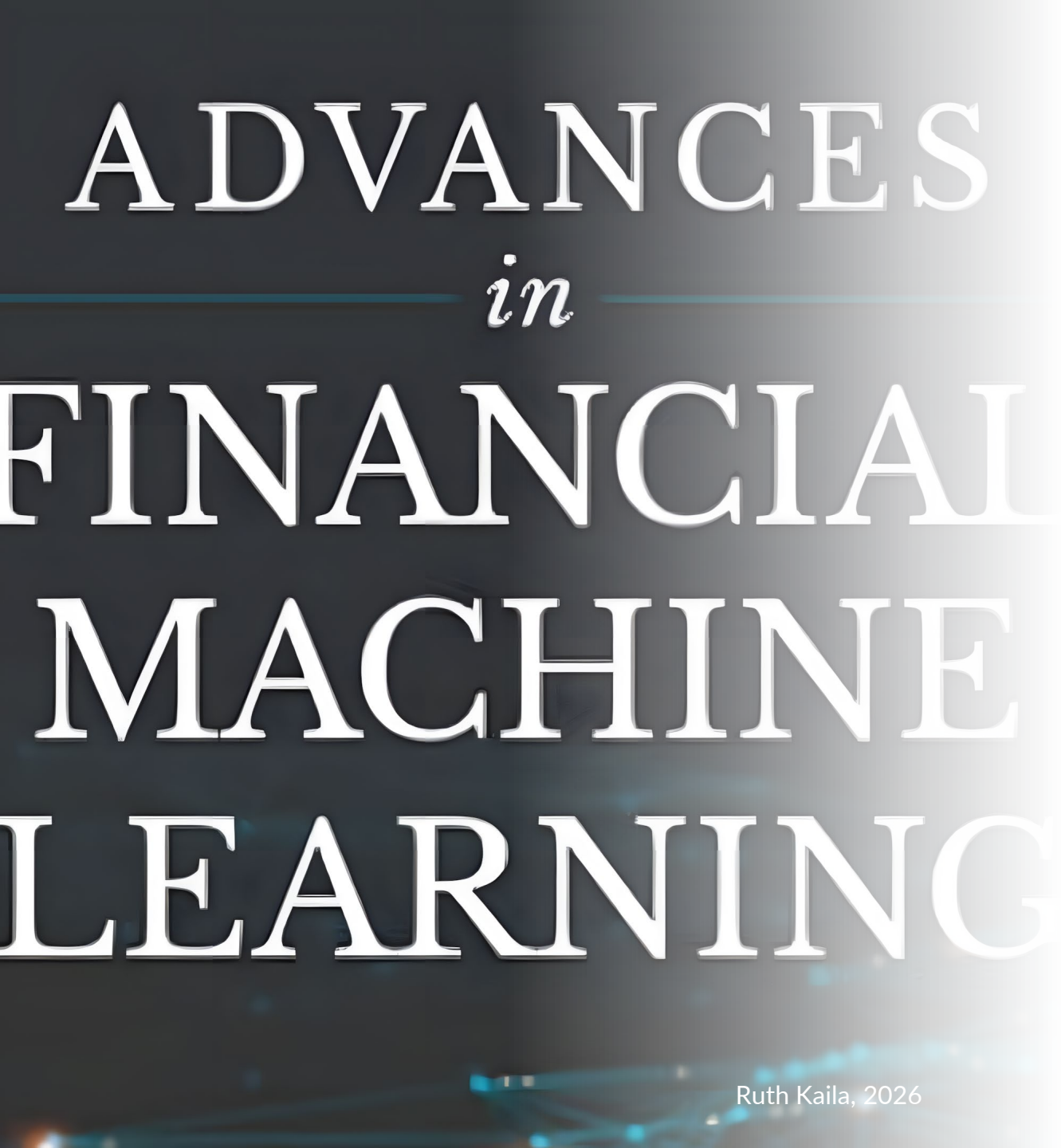
Part 2 Evaluation Metrics for ML Models

- Type 1 and type 2 errors
- Confusion matrix

- Exercise 2C, pairs trading

Main elements of machine learning

- Defining a Task
- Data
  - choose data
  - collect data
  - prepare data
- Mathematical modelling of the task
- Define the loss function (how well a given machine learning model fits the specific data set, or some other objective the model will optimize on the training data)
- Choose a learning algorithm
- Evaluate model performance using appropriate metrics (e.g., precision/recall, mean squared error, economic P&L), not just classification accuracy
- Validate and monitor the model (backtesting, out-of-sample tests, and live monitoring)



Financial data structures
based closely on de Prado's book

Source: de Prado

2. Data structures

Essential types of data

- Fundamental data
  - mainly accounting data
  - Typically reported quarterly or semiannually, i.e., with a lag.
  - Make sure that your analysis uses only information that has been publicly available
  - The actual publication date is often 1–2 months after the period end (and longer for annual reports) — always use the actual release timestamp to avoid look-ahead bias
  - often **backfilled**: missing data is assigned a value that might be completely incorrect or was known only much later
- Market data: quotes and trades disseminated by exchanges and other trading venues (note that not all trading activity is visible in public feeds)
- Analytics: extracted from raw sources; might be expensive
- Alternative data: often primary data – movement of tankers etc.
  - Unique, often hard-to-process data might bring unique opportunities

2.1 Data bars

In order to apply ML algorithms, we often need to:

- Parse / break down unstructured data
- Extract valuable information from the parsed data
- Store the data in a structured format
    - Typically, as tabular data where each row is a bar, though sequence or tensor formats are also used.

We can use

- standard bars
- more informative bars.

Source: de Prado

2.1 Standard bars

Typical information of bars

- Timestamp
- Volume-weighted average price
- Open price
- Close price
- High price
- Low price
- Volume traded
- Etc.

Source: de Prado

```
Date,Open,High,Low,Close,Adj Close,Volume
2000-01-03,68.199997,69.599998,68.199997,38.762444,68.900002,4501230
2000-01-04,68.000000,68.000000,67.000000,38.143608,67.800003,6111790
2000-01-05,65.699997,67.400002,65.599998,37.355972,66.400002,7268180
2000-01-06,67.500000,69.000000,67.400002,38.706192,68.800003,8504905
2000-01-07,69.400002,70.599998,69.400002,39.325039,69.900002,9159290
2000-01-10,70.199997,71.199997,70.000000,39.943890,71.000000,5719630
2000-01-11,70.599998,70.900002,70.400002,39.831371,70.800003,7115810
2000-01-12,70.199997,71.199997,70.199997,39.831371,70.800003,8344375
2000-01-13,70.800003,72.599998,70.500000,40.450214,71.900002,4385670
2000-01-14,71.800003,72.599998,71.800003,40.618999,72.199997,5425405
2000-01-17,71.900002,72.300003,71.800003,40.675255,72.300003,2520860
2000-01-18,72.300003,72.800003,72.000000,40.562737,72.099998,6579610
2000-01-19,72.000000,73.400002,72.000000,40.844025,72.599998,3117025
2000-01-20,73.000000,73.000000,71.699997,40.675255,72.300003,4488870
2000-01-21,72.000000,72.000000,70.400002,39.943890,71.000000,4731020
2000-01-24,71.000000,72.000000,71.000000,40.281437,71.599998,3327035
2000-01-25,71.000000,71.000000,68.599998,38.706192,68.800003,4810635
2000-01-26,69.400002,69.500000,68.000000,38.537415,68.500000,4298665
2000-01-27,68.699997,69.199997,67.800003,38.424900,68.300003,2983245
2000-01-28,68.199997,68.199997,65.199997,37.355972,66.400002,3611035
```

Here, we have daily prices. But new price information might come every millisecond. We cannot use all the data. How to sample it?

Time bars

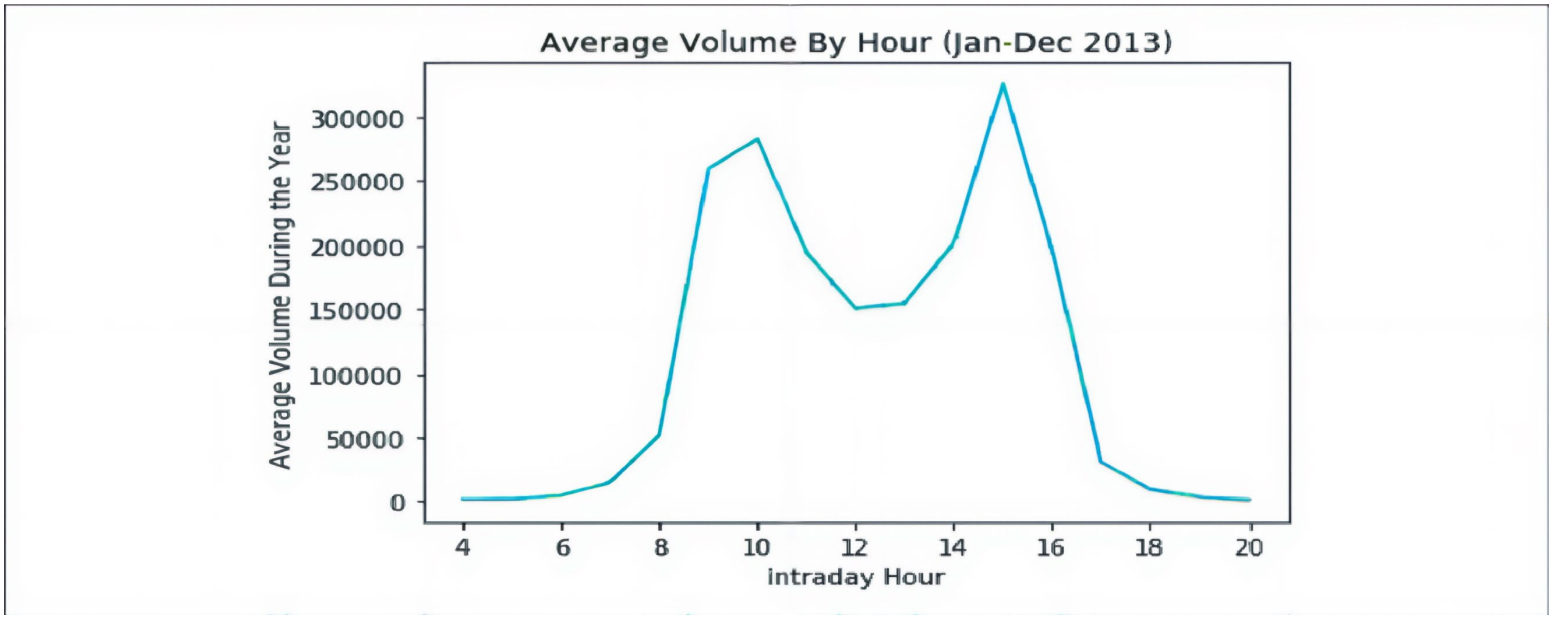
New bar is obtained by sampling information at fixed time intervals, for example, every minute.

Time bars are the most popular bars, BUT

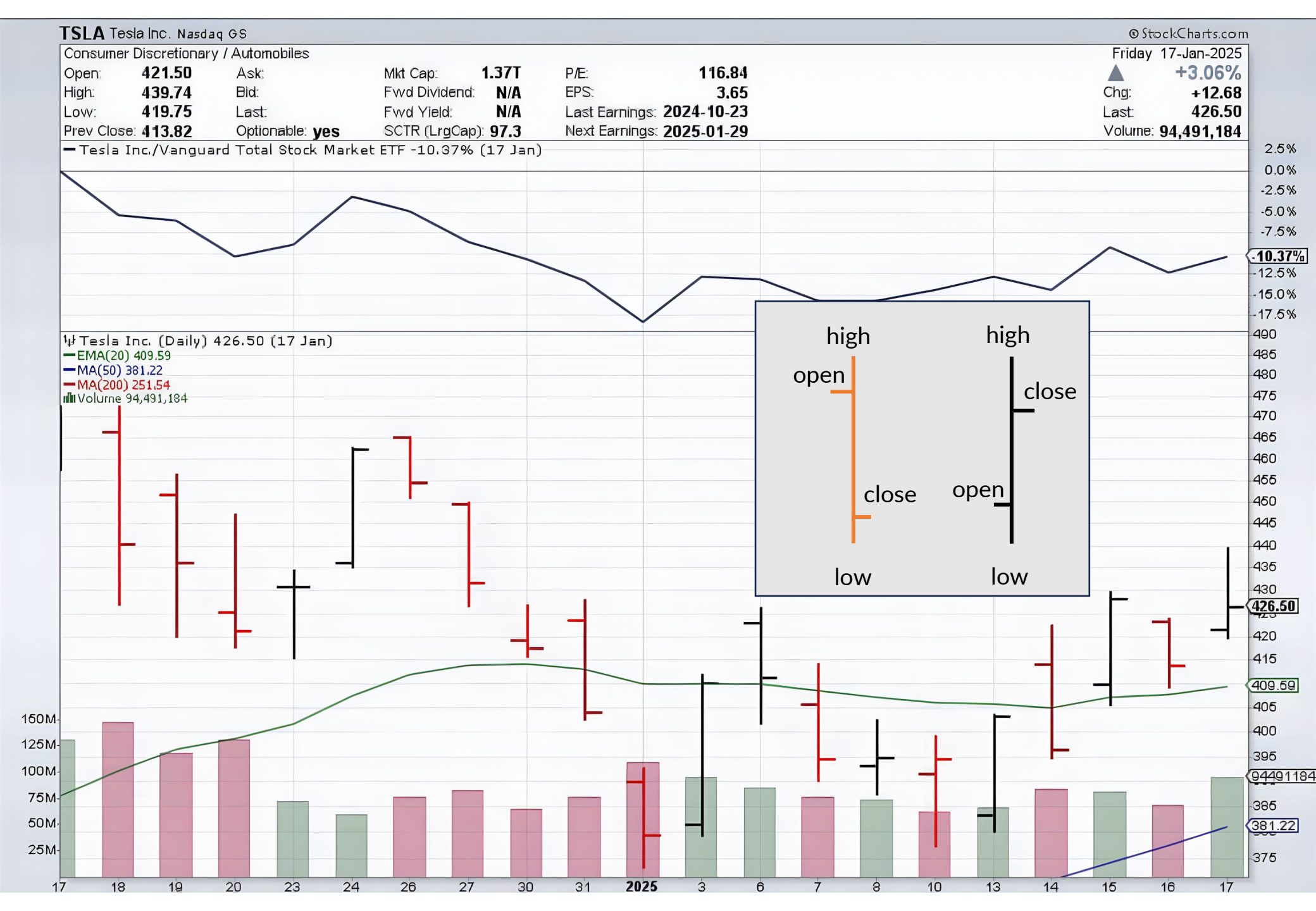
- Markets don't process information at constant time intervals
    - the first hours of the day are much more active than the time around the midday.
- Today, algo's trade with light human supervision – CPU/GPU/FPGA processing cycles are even more important.
- The statistical properties of time-sampled series is often poor (serial correlation, heteroscedasticity, non-normality of returns, because of market microstructure and volatility clustering).
    - GARCH models were partly developed to cope with the heteroscedasticity associated with incorrect sampling. However, they are useful regardless of the sampling scheme, as some heteroskedasticity is inherent to financial markets, not just as a consequence of time bars.

One possibility in improving statistical propertied is to form the bars based on **trading activity.**

Source: de Prado



Picture: Libnam, Haber, Schabs


TSLA Tesla Inc. Nasdaq GS
StockCharts.com
Consumer Discretionary / Automobiles
Friday 17-Jan-2025
Open: 421.50
High: 439.74
Low: 419.75
Prev Close: 413.82
Ask:
Bid:
Last:
Optionable: yes
Mkt Cap: 1.37T
Fwd Dividend: N/A
Fwd Yield: N/A
SCTR (LrgCap): 97.3
P/E: 116.84
EPS: 3.65
Last Earnings: 2024-10-23
Next Earnings: 2025-01-29
+3.06%
Chg: +12.68
Last: 426.50
Volume: 94,491,184
Tesla Inc./Vanguard Total Stock Market ETF -10.37% (17 Jan)
Tesla Inc. (Daily) 426.50 (17 Jan)
EMA(20) 409.59
MA(50) 381.22
MA(200) 251.54
Volume 94,491,184
high
open
close
low
high
close
open
low
426.50
409.59
381.22
94491184
-10.37%
160M
125M
100M
75M
50M
25M
17 18 19 20 23 24 26 27 30 31 2025 3 6 7 8 10 13 14 15 16 17

Tick bars

Update the sample variables (timestamp, open price etc.) each **time a pre-defined number of transactions takes place**, e.g., 10 000 ticks.

- Price changes over a fixed number of ticks can produce time series which are closer to Gaussian properties, and which are more homoscedastic.
- Return distributions still have heavy tails and volatility clustering
- We can better approximate price log-returns with normal distributions.

Outliers: many exchanges have an auction at the open and an auction at the close of the day.

As a result, for a period of time, the order book accumulates bids and offers without matching them.

- After the auction, a large trade is published at the clearing price. This auction trade is reported as one tick, even if it could be the equivalent of thousands of ticks.

Source: de Prado

Note:

tick represents a transaction of a buyer matching a seller

But also: A tick is the minimum incremental amount at which you can trade a security. The minimum tick size for stocks trading above $1 is one cent.

Typical information content of bars

- Timestamp
- Volume-weighted average price
- Open price
- Close price
- High price
- Low price
- Volume traded
- Etc.

```
09/28/2009,09:30:00,50.79,50.7,50.79,100
09/28/2009,09:30:00,50.71,50.7,50.79,638
09/28/2009,09:31:32,50.75,50.75,50.76,100
09/28/2009,09:31:32,50.75,50.75,50.76,100
09/28/2009,09:31:33,50.75,50.75,50.76,100
09/28/2009,09:31:33,50.75,50.75,50.76,100
09/28/2009,09:31:33,50.75,50.75,50.76,100
09/28/2009,09:31:33,50.75,50.75,50.76,100
09/28/2009,09:31:33,50.75,50.75,50.76,100
09/28/2009,09:31:33,50.75,50.75,50.76,100
09/28/2009,09:31:33,50.75,50.72,50.75,100
09/28/2009,09:31:33,50.75,50.72,50.75,100
09/28/2009,09:31:50,50.75,50.73,50.76,300
09/28/2009,09:31:51,50.75,50.74,50.76,300
09/28/2009,09:32:06,50.78,50.76,50.78,300
09/28/2009,09:32:06,50.78,50.76,50.78,500
09/28/2009.09:32:06.50.78.50.76.50.78.100
```

Above, we have information based on transactions. We could sample variables each **time a pre-defined number of transactions takes place**, e.g., 10 000 ticks.

What challenges are there with tick bars?

Source: de Prado

Volume bars

The order fragmentation (into many small trades) introduces some randomness in the number of ticks per unit of traded volume.

E.g., someone sells 100 shares of a stock.

- Problem: this can be executed as 1 tick of size 100 OR 100 ticks of size 1
- Solution: volume bars

Volume bars sample every time **a pre-defined amount of the security's units** has been exchanged.

- security's units could be shares, futures contracts, etc.

We would sample prices every time 1000 futures contracts have been exchanged.

Stock returns are even closer to Gaussian distributions than when sampling by tick bars.

Several market microstructure theories study the interaction between prices and volume → sampling as a function of these variables is natural.

Source: de Prado

Dollar bars

Dollar bars: An observation in samples every time a **pre-defined market value is exchanged.**

Example: We want to analyze a stock that has doubled over a time period T.

We initially bought an amount A of the shares with 1000 dollars.

- The value of the shares doubles over a time period T.
- Selling 1000 dollars worth of this stock at the end of the period requires trading half the number of shares it took to buy them, i.e. A/2.

The number of shares traded is a function of the actual value of the shares.

→ sampling bars in terms of dollar value exchanged is particularly **good in case of significant price fluctuations.**

Motivation for the use of dollar bars:

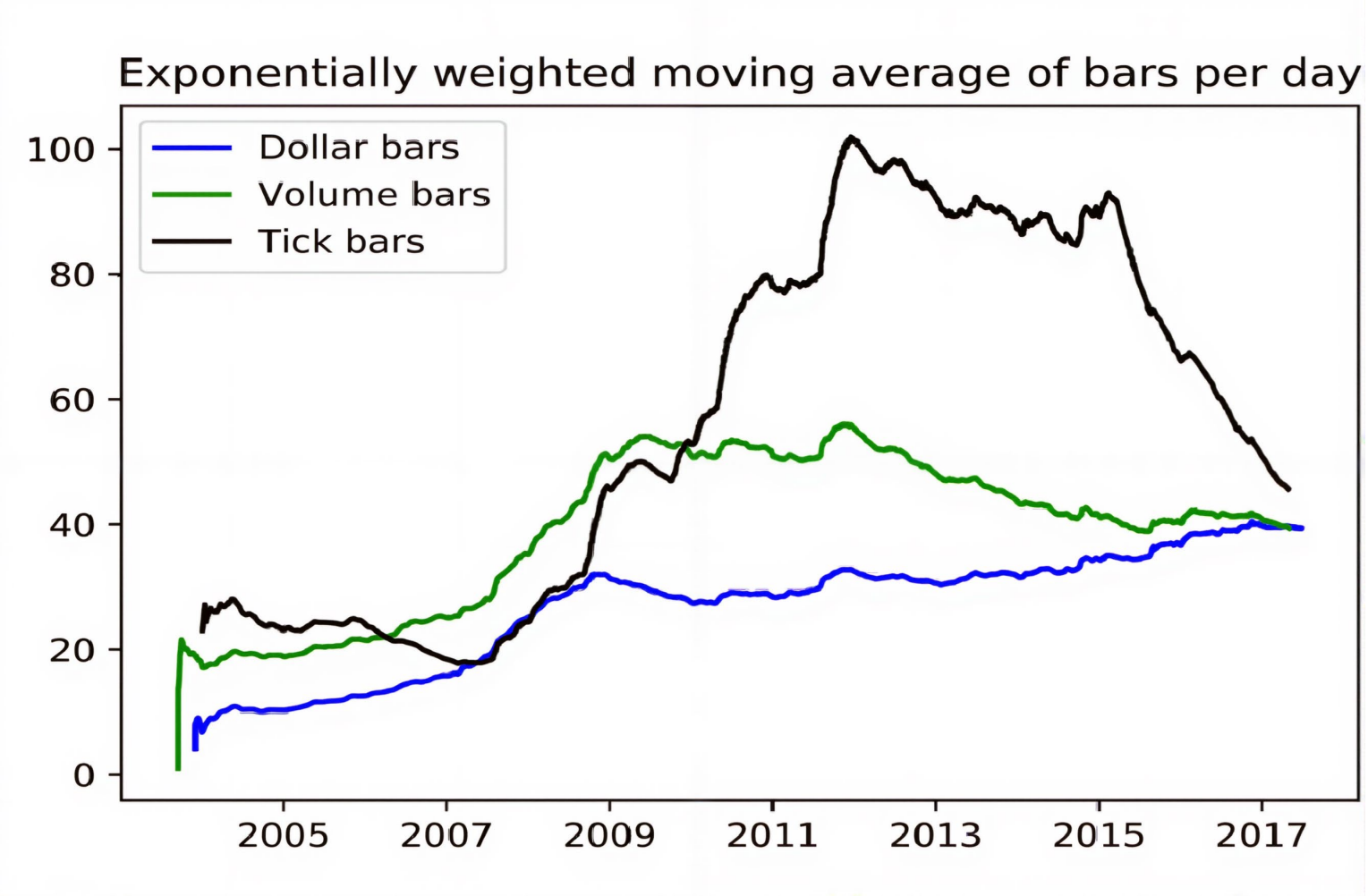
1. If you compute tick bars and volume bars on e.g. E-mini S&P futures, the number of bars per day will vary a lot over the year.

→ number of dollar bars is more stable.

2. The number of outstanding shares often changes multiple times over the security's lifetime, for example due to corporate actions. (stock splits etc.)

Source: de Prado

Dollar bars



Source: de Prado

Information driven bars

The aim is to sample more frequently as new information becomes available.

We want to detect order-flow imbalances (e.g., more buyer-initiated trades than seller-initiated ones) so that we can anticipate a possible change in trend before the market reaches a new equilibrium. We assume that persistent imbalances are partially the result of the transactions of informed traders, although noise traders also contribute.

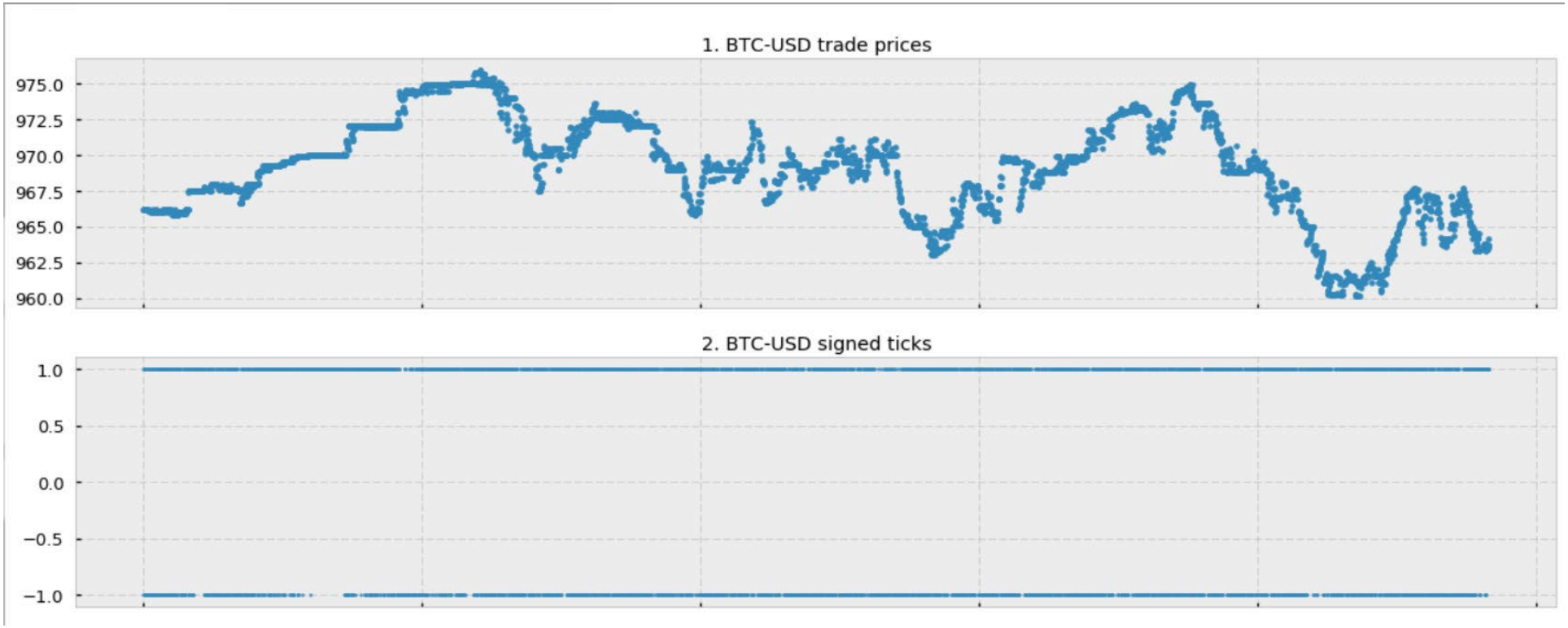
→ ***imbalance bars***

Imbalance bars can be applied to tick, volume or dollar data.

We create a threshold and create a new bar whenever the imbalance exceeds the threshold.

Tick imbalance bars

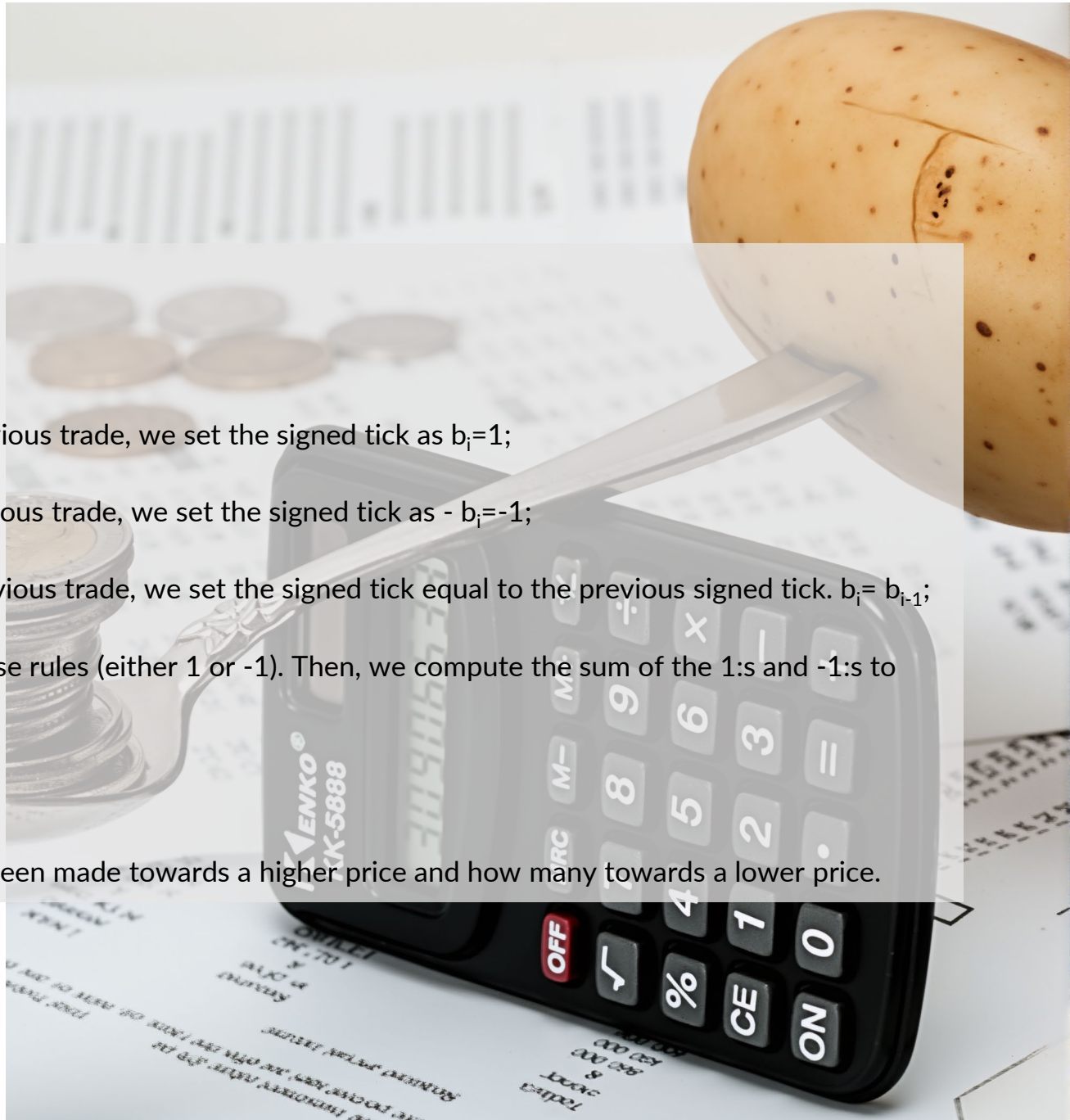
- The idea is to sample bars whenever tick's imbalance exceeds a certain threshold.
- Bars are created more frequently during periods of high market activity and less frequently during quiet periods.
- These bars can better reflect the dynamics of supply and demand.
- They capture meaningful order flow and reduces noise.
- We assume that the imbalance is the result of the transactions of informed traders.
- Good for trend detection, volatility estimation



Picture: G Martinez

1. Bitcoin-dollar prices, 5000 trades
2. We transformed the trades to 1 (up) or −1 (down)
3. We compare $\sum 1$ and $\sum -1$.

Imbalance bars



What is tick imbalance?

For each trade:

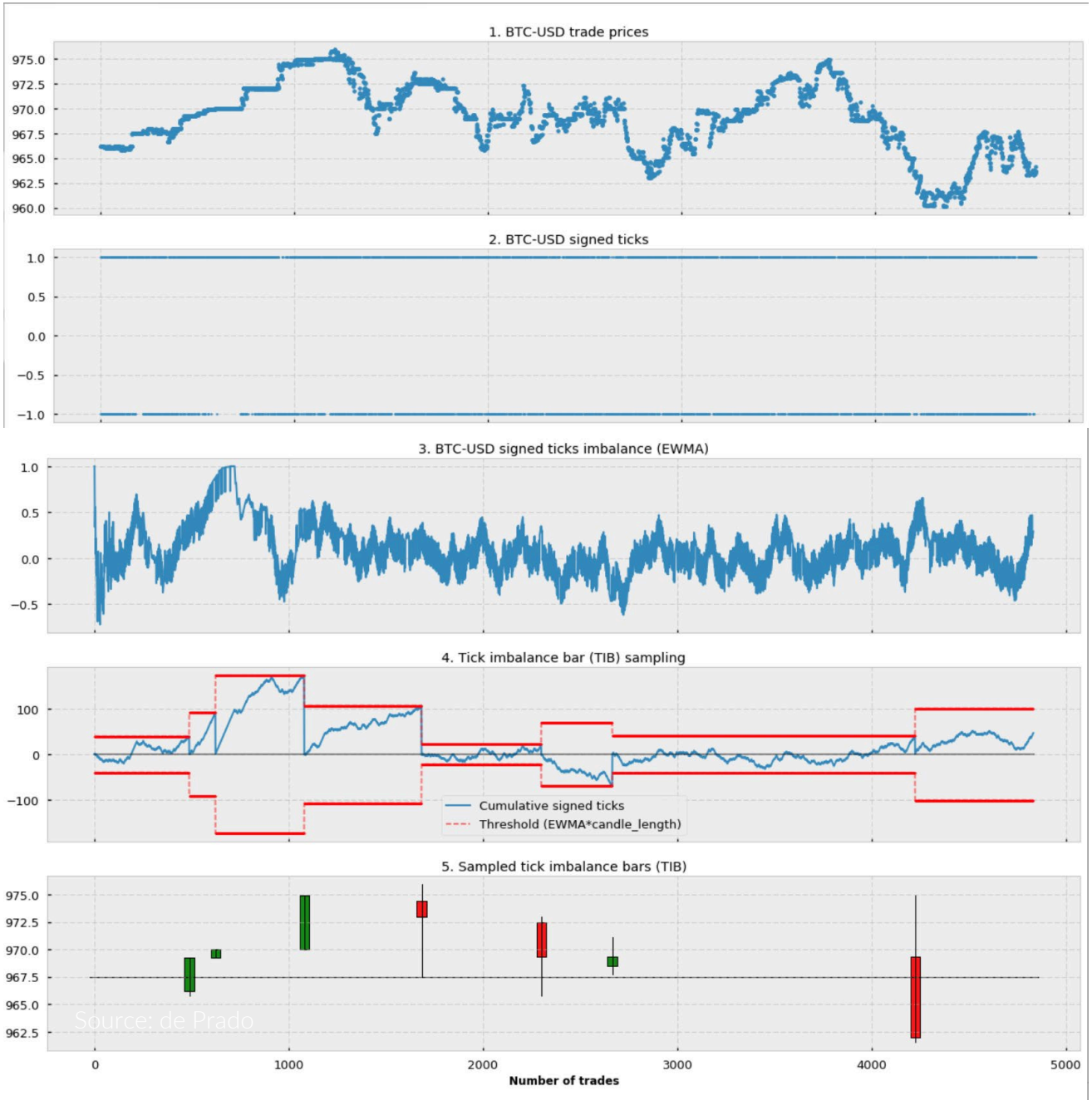
1. If the price is higher than in the previous trade, we set the signed tick as b_i=1;
2. If the price is lower than in the previous trade, we set the signed tick as - b_i=-1;
3. If the price is the same as in the previous trade, we set the signed tick equal to the previous signed tick. $b_i = b_{i-1}$;

We transform all trades according to these rules (either 1 or -1). Then, we compute the sum of the 1:s and -1:s to calculate how imbalance the market is.

Market imbalance $\theta = \sum_{i=1}^{n} b_i$.

We want to see how many trades have been made towards a higher price and how many towards a lower price.

Source: de Prado



Picture: G Martinez

Figure 1. Example of tick imbalance bars for the BTC-USD pair.

1. Bitcoin-D prices, 5000 trades
2. We transformed the trades to 1 or -1
3. We applied an exponential moving average EWMA to the whole sequence of ticks.
   - stochastic fluctuation between -1 and 1
   - a bit the same kind of cumulative information than the market imbalance

$$\theta = \sum_{i=1}^{n} b_i$$

4. the red thresholds are based on the expected number of ticks and the imbalance of b_i sequence at the beginning of the bar (i=1)
5. the generated tick imbalance bars

Feature is data that's used as the input for ML models to make predictions

Sampling features

We have now seen how to produce continuous, structured bar data from unstructured economic data.

BUT

- Several ML algorithms (e.g., kernel methods, exact k-nearest neighbors) do not scale will with sample size (this will be discussed later).
  - In high-frequency data, we may have many more data points than are needed to estimate a stable model
- ML algorithms should be trained with relevant data.
  - All data points are not relevant; many may be redundant or dominated by noise.

Source: de Prado

Sampling for reduction

The aim is to reduce the amount of data used to fit the ML algorithm. — down-sampling

- Sequential sampling as a constant step size (linspace sampling)
  - Simple
  - The step size is arbitrary; the outcomes may vary depending on the seed bar (the index of the first bar used).
- Sampling randomly using a uniform distribution (uniform sampling)
  - Sampled uniformly across the entire set of bars.
  - May fail to include rare but very important events

The samples don't necessary contain the subset of most relevant observations in terms of their predictive power.

Event-based sampling

Instead of taking observations at fixed intervals, observations are taken only when a meaningful event or price movement occurs.

Portfolio managers typically place a bet after some meaningful event takes place.

- structural break
  - extracted signal (momentum, volatility)
  - microstructural phenomena (order-book phenomena, bd-ask spread change..)

In our trading exercises, we don't react to each price change. We label events based on when the cumulative sum of price changes exceeds a predefined threshold.

Source: de Prado

Event-based sampling, CUSUM filter

The CUSUM filter (explained in detail in your exercise) is designed to detect a shift in the mean value of a measured quantity away from a target value.

- a tool used for identifying significant price changes by filtering out noise
- works by accumulating price changes until a predefined threshold is reached, which **signals a meaningful move in the price series.**

Advantages

- We label only market movements that exceed our chosen threshold (their relevance depends on how well we chose the threshold for the strategy.
- We get information on market volatility by observing the labels.
- Less noise, clearer signal

We assume:

- P_t is the price series,
- $r_t = P_t - P_{t-1}$ is the price change at time t,
- threshold $= 1.0$ (a price change of 1 unit is considered significant)

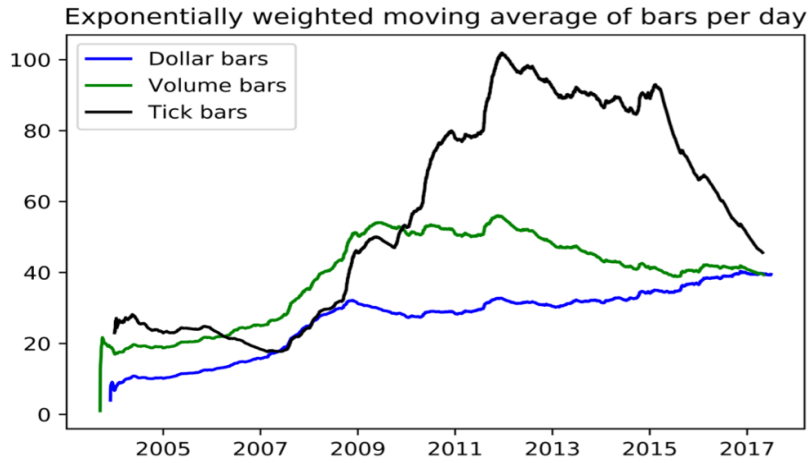
Steps:

- At $t=0$, $S_0=0$.
- We accumulate changes ($S_t = S_{t-1} + r_t$) as long as $|S_t| < 1.0$.
- When $|S_t| \geq 1.0$, we record t as an event and reset S_t to 0 and continue.

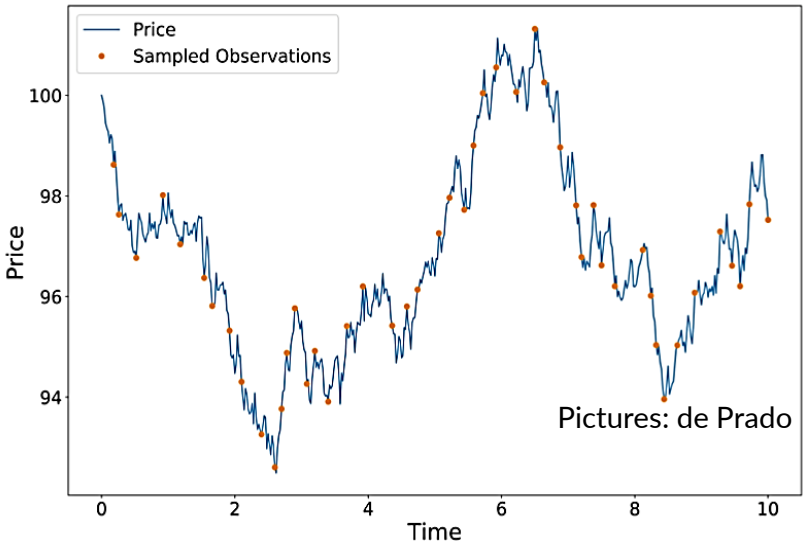
CUSUM=cumulative sum control chart

What did we learn about bars and the information arriving to the markets?

What about downsampling?



bar type	basis	
time	fixed intervals, 1 min, 5 min	Simple; active vs inactive periods
tick	Fixed number of trades	Ignores trade size
volume	Fixed traded volume	Ignores buy/sell imbalance
Tick imbalance	Imbalance between buy and sell pressure	Meaningful order flow, reduces microstructure noise



Revision

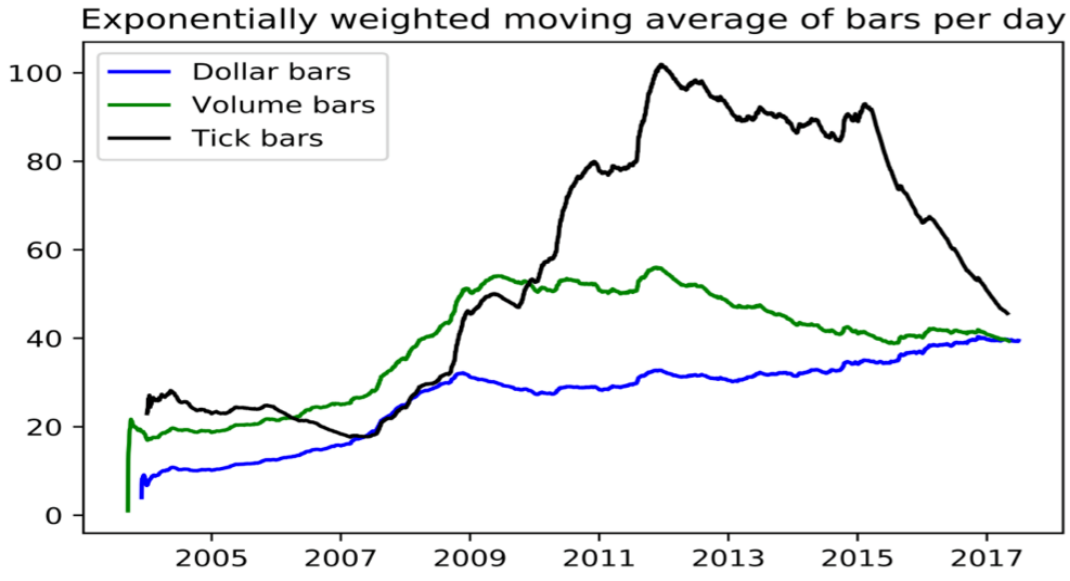
Forming bars

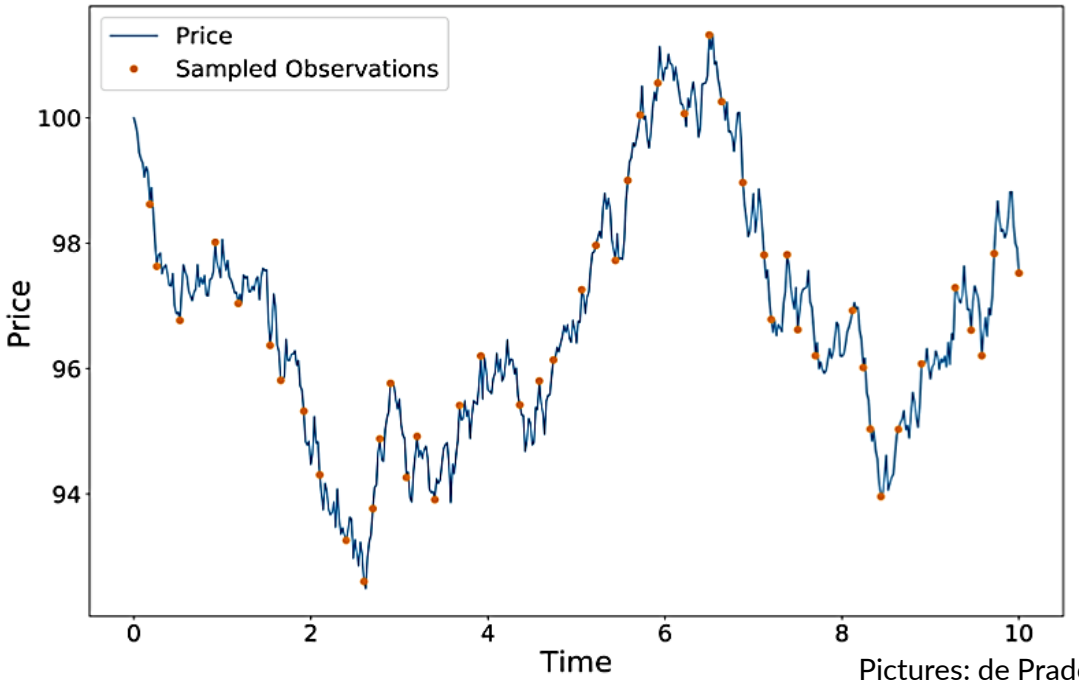
- the information does not arrive to markets at a constant rate
- the information content of data sampled at equal time intervals is not constant
    - Tick bars, volume bars, dollar bars, tick imbalance bars.

We might try to downsample our data (matrix consisting of bars) so that the

subset of bars is as informative as possible

- Several ML algorithms do not scale with sample size.
- ML algorithms should be trained with relevant data.
- CUMSUM filter for event-based sampling of large price moves.





Pictures: de Prado

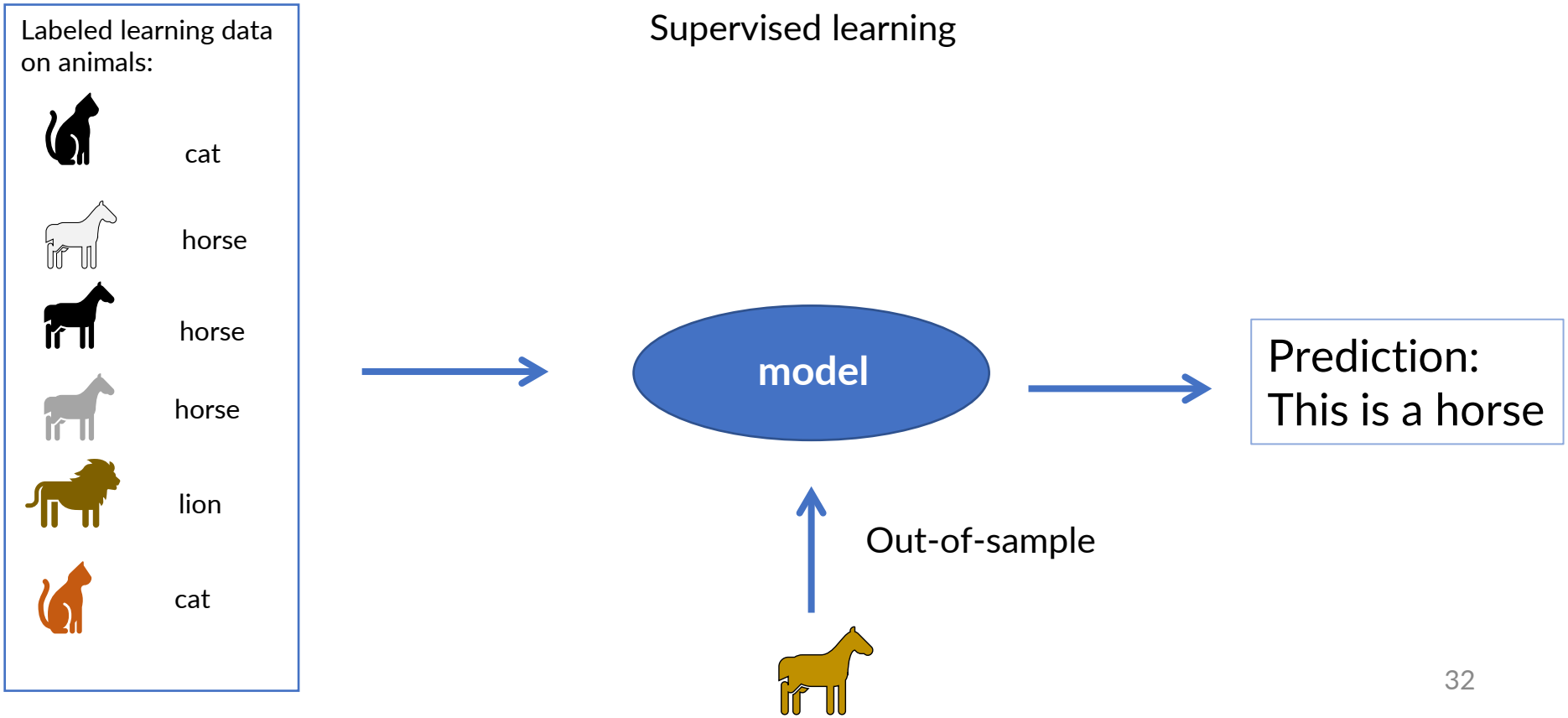
LABELING

Input: feature
Output: label

Labeling

- *Unsupervised ML methods can learn the patterns from the sample data*
- *Supervised ML methods require that the data is labelled. The ML methods can then predict the labels of unseen out-of-sample observations.*

Label is the outcome we want the model to learn.



Labeling: The Fixed-time horizon method

Sell or buy label based on price return and time-bars

Consider a features matrix X with I rows, $\{X_i\}_{i=1,\dots,I}$, drawn from some bar index $t = 1, \dots, T$, where $I \leq T$.

An observation X_i is assigned a label $\mathrm{y_i} \in \{-1, 0, 1\}$,

$$y_i = \begin{cases} -1 & \text{if } \mathrm{r}_{t_{i,0},t_{i,0}+\mathrm{h}} < -\tau \\ 0 & \text{if } |\mathrm{r}_{t_{i,0},t_{i,0}+\mathrm{h}}| \leq \tau \\ +1 & \text{if } \mathrm{r}_{t_{i,0},t_{i,0}+\mathrm{h}} > \tau \end{cases},$$

where τ is a pre-defined constant threshold, $t_{i,0}$ is the index of the bar immediately after X_i takes place, $t_{i,0} + h$ is the index of the hth bar after $t_{i,0}$, and $r_{t_{i,0},t_{i,0}+h}$ is the price return over the bar horizon h,

$$r_{t_{i,0},t_{i,0}+h} = \frac{P_{t_{i,0}+\mathrm{h}}}{P_{t_{i,0}}} - 1$$

X_i refers here to one bar.

The labels $-1,0,1$ refer to

-1: sell

0: do nothing

1: buy

Consider a features matrix X with I rows, $\{X_i\}_{i=1,\dots,I}$, drawn from some bar index $t = 1, \dots, T$, where $I \leq T$.

An observation X_i is assigned a label $\mathrm{y_i} \in \{-1, 0, 1\}$,

$$y_i = \begin{cases} -1 & \text{if } \mathrm{r}_{t_{i,0},t_{i,0}+\mathrm{h}} < -\tau \\ 0 & \text{if } |\mathrm{r}_{t_{i,0},t_{i,0}+\mathrm{h}}| \leq \tau \\ +1 & \text{if } \mathrm{r}_{t_{i,0},t_{i,0}+\mathrm{h}} > \tau \end{cases},$$

where τ is a pre-defined constant threshold, $t_{i,0}$ is the index of the bar immediately after X_i takes place, $t_{i,0} + h$ is the index of the hth bar after $t_{i,0}$, and $r_{t_{i,0},t_{i,0}+h}$ is the price return over the bar horizon h,

$$r_{t_{i,0},t_{i,0}+h} = \frac{P_{t_{i,0}+\mathrm{h}}}{P_{t_{i,0}}} - 1$$

Problems:

- the time-bars don't exhibit good statistical properties (normal/lognormal etc.)
- the same threshold τ for the *price return* is applied regardless the observed volatility (the *risk* taken).

What would be better?

Problems:

- the time-bars don't exhibit good statistical properties
- the same threshold τ is applied regardless the observed volatility.

Better:

- We label using a varying threshold $\tau_{t_{(i,0)}}$ (estimated using a rolling exponentially weighted standard deviation of returns (volatility)).
- Alternatively, we use volume or dollar bars. Their volatilities are more stable.

Even better:

- We use dynamic thresholds.

Source: directly de Prado

Computing Dynamic Thresholds

*We want to set **profit taking and stop-loss limits** that are a function of the risks involved in a bet — the volatility.*

The triple barrier method:

We set

- **two horizontal barriers**, defined by profit-taking and stop-loss limits, which are a **dynamic function of estimated volatility**.
- one vertical barrier, defined in terms of number of bars elapsed since the position was taken.

If the upper limit is touched first, we label the observation as a 1.

If the lower limit is touched first, we label the observation as a -1.

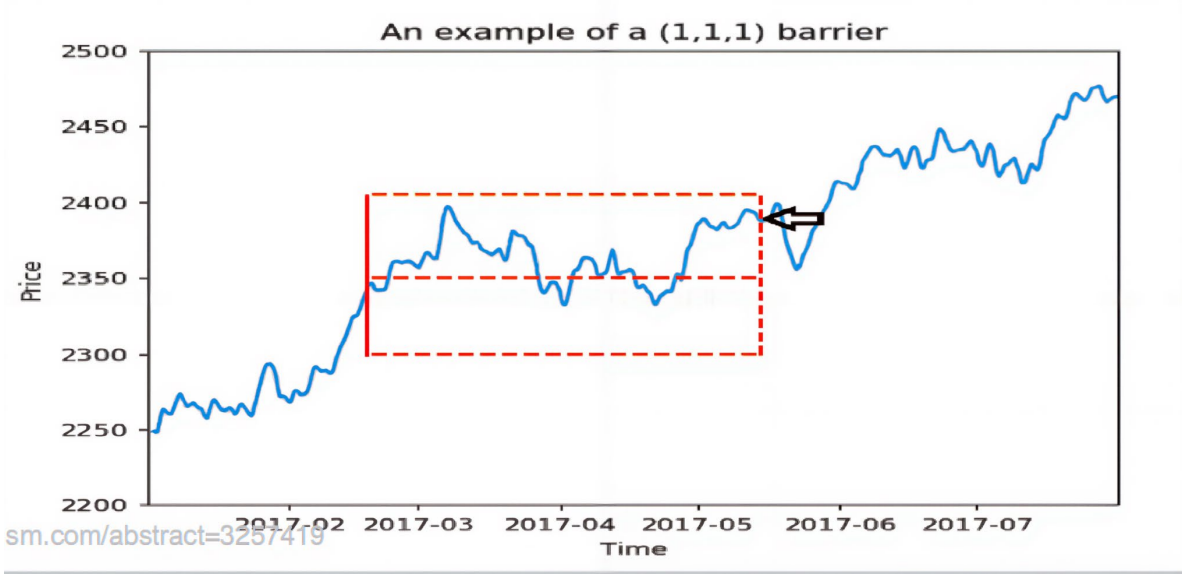
If the vertical barrier is touched first, we have two choices: the sign of the return, or a 0.

This method is path-dependent.

Source: directly de Prado

The triple barrier method by de Prado

The observations are labelled according to the first barrier touched out of three barriers. The label is determined by the whole path of the stock price. The volatility, i.e., the risk affects the path.



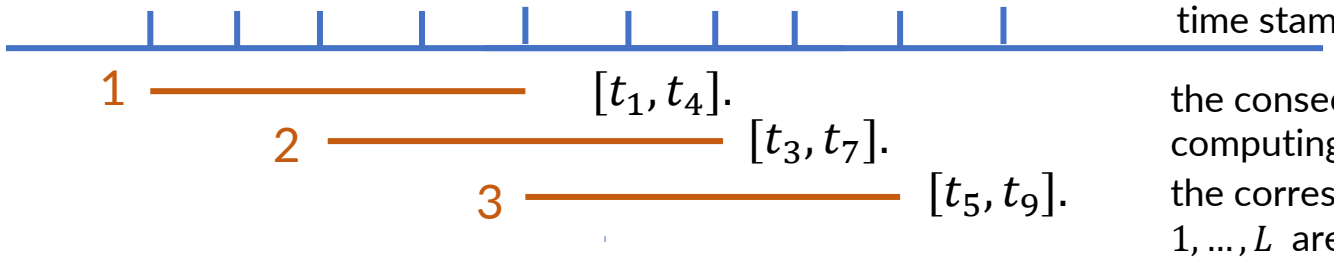
Source: de Prado

- The upper horizontal barrier is defined by **profit-taking limit**. This limit depends on the volatility. If touched first, we set Label=1.
- The lower horizontal barrier is defined by **stop-loss limit**. This limit depends on the volatility. If touched first, we set Label=-1.
- **The vertical barrier is an expiration limit**, defined in terms of number of bars elapsed since the position was taken. If touched first, we set Label=0.

Overlapping labels

Problem:

Often, the observations are not generated by independent and identically distributed (IID) processes.



time stamps of the stock price X_i

the consecutive time intervals for computing the returns $r_{t_i,t_{i+h}}$ and the corresponding labels $\{yj\}, j = 1, \ldots, L$ are overlapping.
Assume that a macro shock happens at t_4. This single event affects the labels 1, 2, 3

We assign a label y_i to an observed feature, stock price X_i, where the label y_i is a function of price bars within the interval $[t_{i,0}, t_{i,1}]$. (Could be return for example)

If $t_{i,1} > t_{j,0}$ for $i < j$, then y_i and y_j will both depend on the same return $r_{t_{j,0},\min(t_{j,1})}$, which spans the interval $[t_{j,0}, \min(t_{j,1})]$.

Consequently, the series of labels $\{y_i\}, i = 1, \ldots, I$ are not independent and identically distributed (IID) whenever consecutive outcomes overlap.

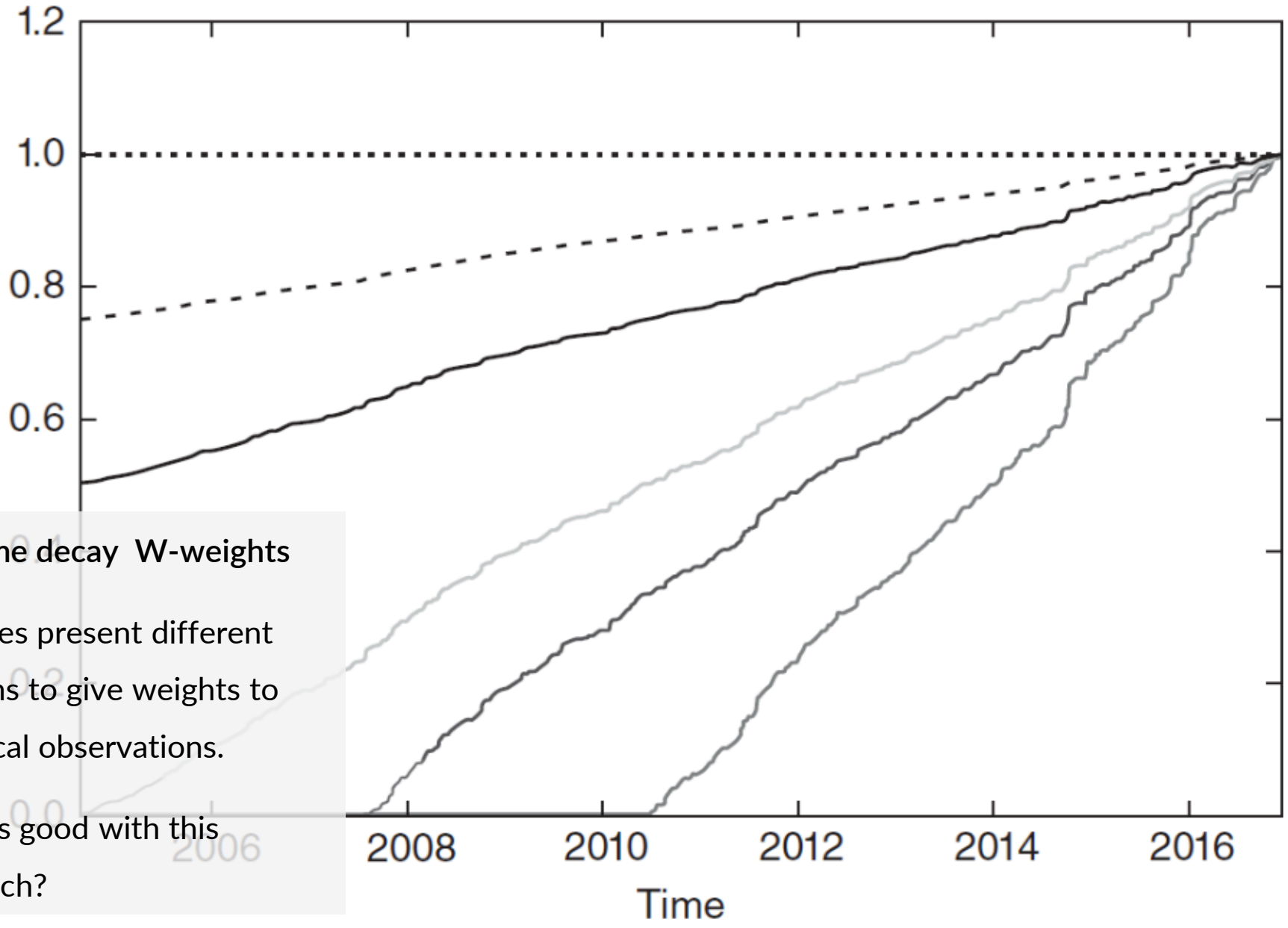
How to cope with this problem?
See the Appendix

Source: de Prado

As markets evolve, older data points are less relevant than newer ones.

One possibility is to give more weight to more recent observations.

We'll practice doing this in the exercises.



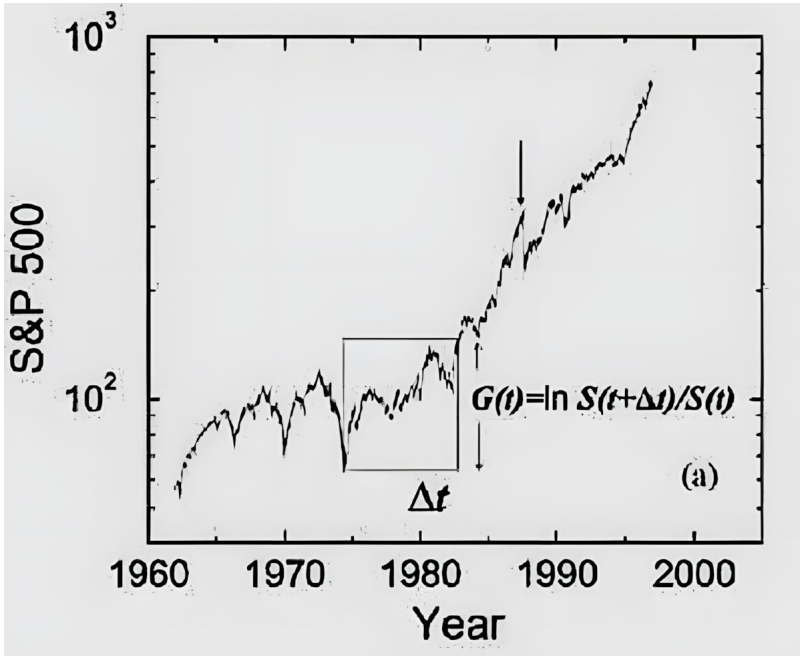
The time decay W-weights

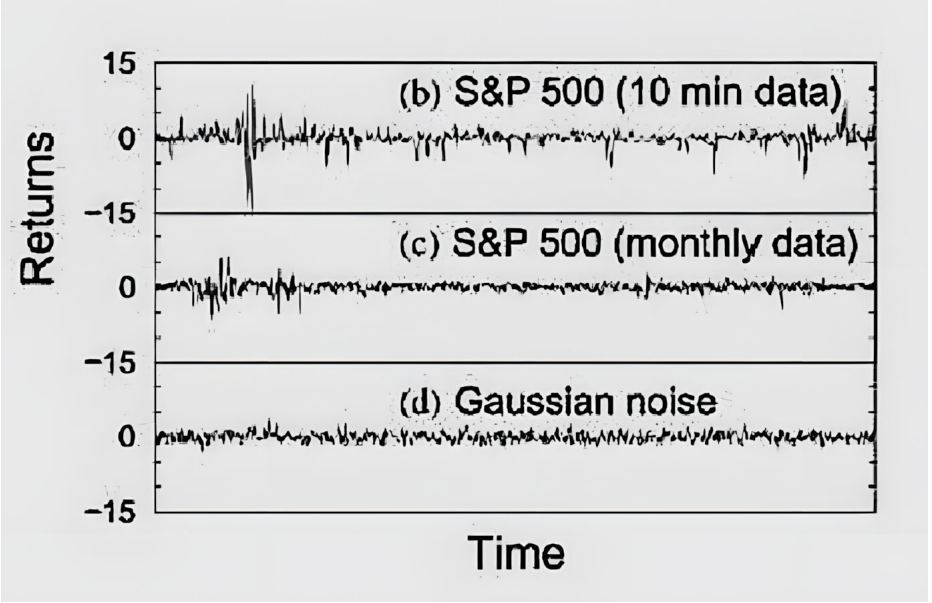
The lines present different patterns to give weights to historical observations.

What is good with this approach?

What is a problem when applying ML to

1. stock prices,
2. returns?





Pictures : de Prado

Stationarity vs Memory

Supervised learning algorithms typically require stationary features: To infer a label for a previously unseen and unlabelled observation, we need to map this observation to a collection of labelled examples. We cannot do the mapping if the features are not stable.

Stationarity is a necessary condition for many well performing ML algorithm.

However, it is not sufficient. **We can make a series more stationary through differentiation, but this comes at the cost of losing memory.**

Stationarity vs Memory

Researches typically need to work with processes whose statistical properties are approximately time-invariant (stationary), and so focus on quantities such as:

- Changes in returns or log-returns
- Changes in yield
- Changes in volatility

The resulting series are closer to stationary, but memory (long-term dependence) from the original series has been reduced.

However, the model's predictive power is based on the memory.

e.g., how far the price has developed from the long-time average.

- **returns are closer to stationary but lack long-term memory**
- **prices have long-term memory but are non-stationary**

We have to search for a minimum amount of differentiation that makes a price series stationary while preserving as much memory as possible.

Part 2
Evaluation Metrics for ML Models

- Type 1 and type 2 errors
- Accuracy, Precision and Recall; F1-measure
- Confusion matrix

Part 3, Exercise 2 C, Pairs trading

Two types of data sets

Training set: (typically 2/3 of the data)

- we select the features
- we fit model parameters

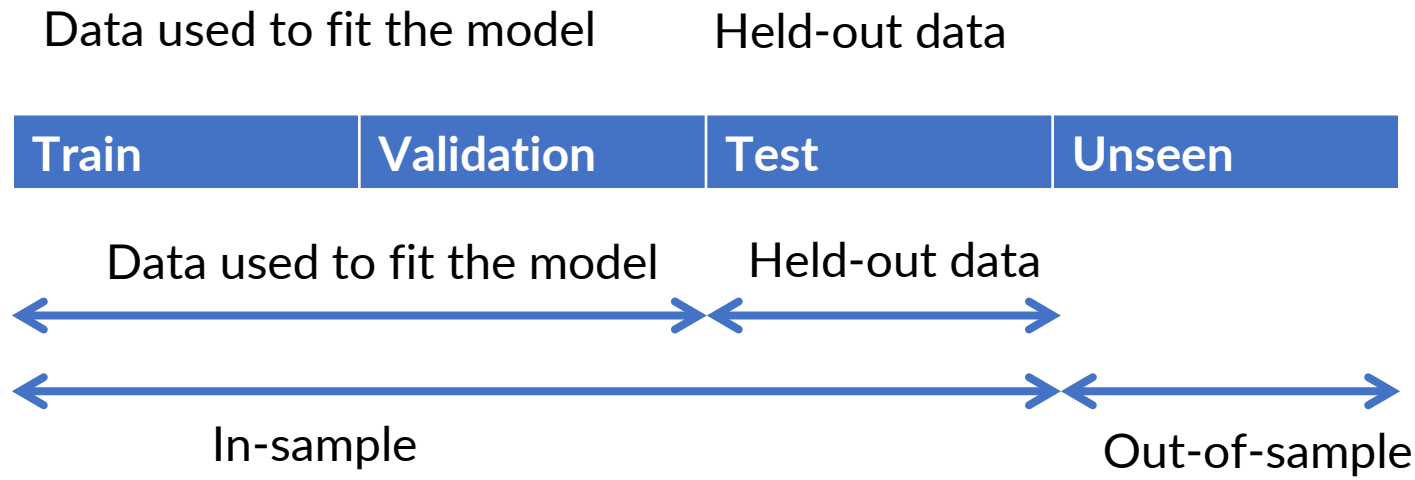
Training set errors:

- errors estimated from the train set

Test set (unseen data):

- not used for model calibration

How to measure the performance of the model?



Type 1 and type 2 errors

In statistical theory, we test about choosing one of two competing propositions:

1. the null hypothesis, denoted by H_0

   we assume that this is true unless the data shows evidence against it.

2. the alternative hypothesis, denoted by H_1.

Type 1 error: False positive

rejection of true null hypothesis H_0

alpha type error

Type 2 error: False negative

non-rejection of false null hypothesis H_0

beta type error

In financial literature,

Type 1 errors are typically controlled; we incorrectly accept H_1

Type 2 errors are often ignored, even though they matter for the power of a test.

Table of error types

Null hypothesis H_0 is

Decision about null hypothesis H_0

	True	False
Don't reject	Correct inference true negative; we correctly reject H_1 Probability =1-alpha	Type 2 error false negative; we incorrectly reject H_1 Probability = beta
Reject	Type 1 error false positive; we incorrectly accept H_1 Probability = alpha	Correct inference true positive; we correctly accept H_1 Probability =1-beta

The alpha level represents the probability of rejecting the null hypothesis when it is actually true (Type I error)

Neyman-Pearson hypothesis testing (1933), Example with trading strategy

We model

- a null hypothesis H_0: the trading strategy's returns are no better than random, normally distributed with mean 0. (H_0 typically represents the baseline state or reference point (no difference))
- an alternative hypothesis H_1: the trading strategy's returns are better than random. (This is what we want to test)

We set a predefined **cut-off probability** or **significance level** α (e.g. 0,05, 0,01, ..)

- α is the probability of making a **Type I error**, which occurs when we reject H_0 even though it is true.

We compare α with the P-value of H_1.

- The **P-value is the likelihood that the observed statistic occurs due to chance**, given the sampling distribution.
- The P-value is derived from the t-distribution, which depends on the degrees of freedom (df). The t-statistic of the mean returns measures *how far the sample mean return is from zero* (the null hypothesis) in terms of standard error).

If P-value≤ α : Reject H_0 , meaning the result (of the new trading strategy) is statistically significant.

If P-value> α : Fail to reject H_0 , meaning the evidence is insufficient to support H_1.

The P-value does not give the probability that the null hypothesis is true.

Neyman-Pearson hypothesis testing (1933), Example with trading strategy

We model

- a null hypothesis H_0: the trading strategy's returns are no better than random, normally distributed with mean 0. (H_0 typically represents the baseline state or reference point (no difference))
- an alternative hypothesis H_1: the trading strategy's returns are better than random. (This is what we want to test)

We set a predefined **cut-off probability** or **significance level** α (e.g. 0,05, 0,01, ..)

- α is the probability of making a **Type I error**, which occurs when we reject H_0 even though it is true.

We compare α with the P-value of H_1.

- The **P-value is the likelihood that the observed statistic occurs due to chance**, given the sampling distribution.
- The P-value is derived from the t-distribution, which depends on the degrees of freedom (df). The t-statistic of the mean returns measures *how far the sample mean return is from zero* (the null hypothesis) in terms of standard error).

If P-value≤ α : Reject H_0 , meaning the result (of the new trading strategy) is statistically significant.

If P-value> α : Fail to reject H_0 , meaning the evidence is insufficient to support H_1.

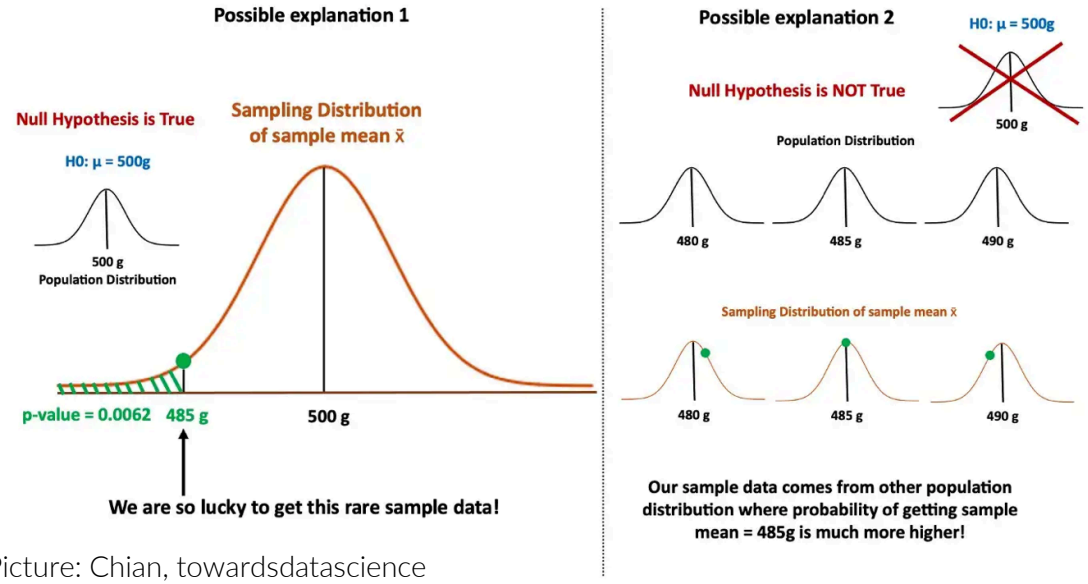
The P-value does not give the probability that the null hypothesis is true.

Neyman-Pearson hypothesis testing (1933)

The P-value is the likelihood that the observed statistic occurs due to chance, given the sampling distribution.

The P-value does not give the probability that the null hypothesis is true.

Below: H_0 – average is 500 g, H_1 –the observed average of the sample is 480, we assume a different distribution.



Picture: Chian, towardsdatascience

P-value <= α :

We reject H_0 in favor of H_1

→ the result is statistically significant

(the result is statistically different from H_0)

P-value > α :

We fail to reject H_0

→ the result is statistically insignificant

Reducing the probability of committing a Type 1 error (false positive, incorrectly accept H_1):

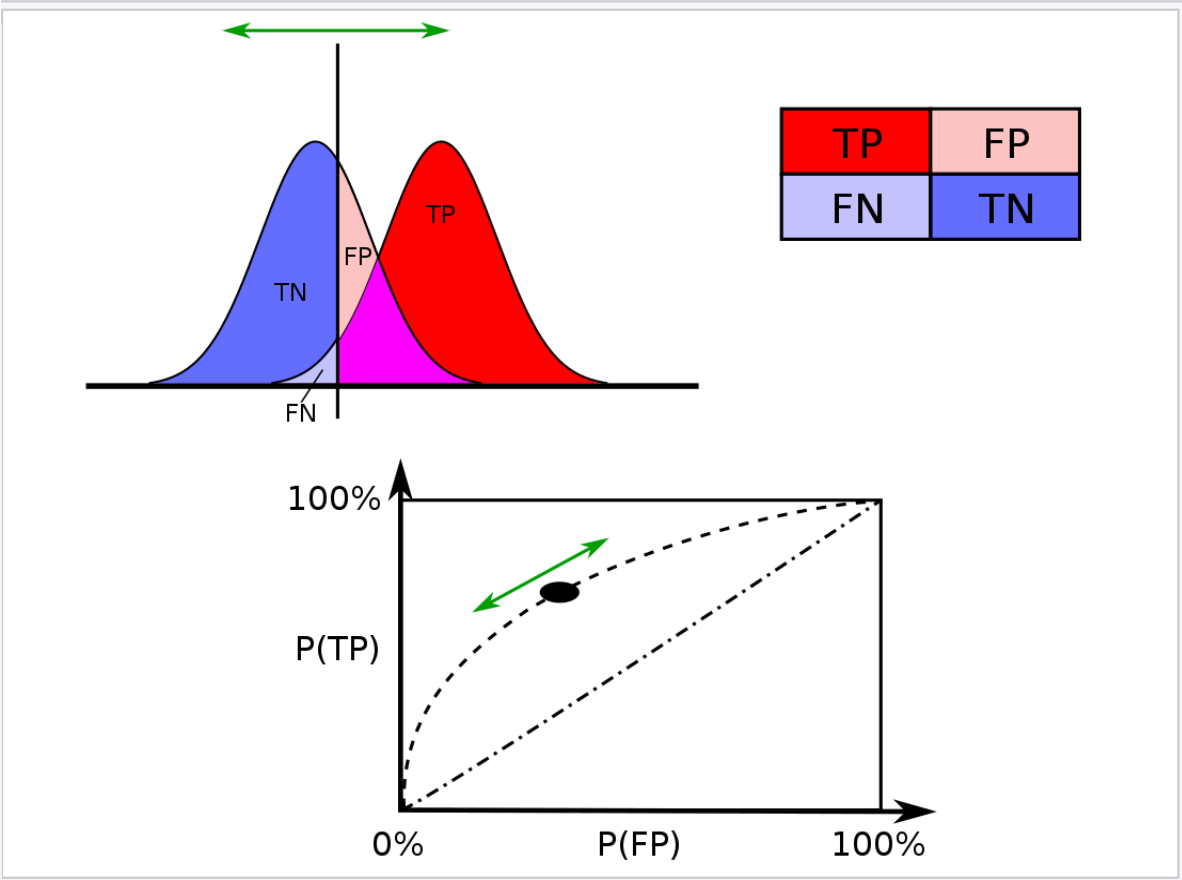
make the $\alpha(p)$ value more rigid.

We say that a test statistic is *robust* if the Type 1 error rate is controlled. (rejection of true null hypothesis H_0)

Reducing the probability of committing a Type 2 error (false negative, no-rejection of false H_0):

- increasing the test's sample size
- relax the α(p)

Varying different threshold (cut-off) value could also be used to make the test either more specific or more sensitive, which in turn elevates the test quality.



The results from the negative sample (left curve) overlap with the results from the positive sample (right curve). By changing the decision threshold (the vertical cutoff line in the test statistic or score distribution), we can trade off the probabilities of false positives (Type I errors) and false negatives (Type II errors).

P-hacking

the practice of manipulating data or statistical analyses to artificially obtain significant p-values (often $p<0.05$)

This might be

- Running multiple regressions or tests until one produces a significant result.
- Selecting or transforming variables post hoc to achieve statistical significance.
- Dropping or including certain observations that shift the p-value in the desired direction.
- Stopping data collection as soon as they obtain a statistically significant result, disregarding the importance of having a pre-established sample size for reliable analysis
- removing outliers if it helps to achieve significance
- slicing or subgroup data in ways that produce significance
- fitting many different regression models (linear regression, multiple linear regression, logistic r etc.)
- reporting the results selectively
- choosing 'wisely' the data (e.g., suitable time interval)

Avoid by

- developing a good research plan
- documenting well
- being ethical

Accuracy, Precision and Recall; F1-measure

Accuracy, precision and recall are used to evaluate the performance of classification or information retrieval systems.

- Accuracy=(true positive + true negative) / (true positive+ false positive + true negative + false negative)

Precision: the fraction of relevant instances among **all retrieved instances**.

Recall (sensitivity): the fraction of retrieved instances among **all relevant instances**.

- **Precision** = ***true positive / (true positive + false positive)***
- **Recall** = ***true positive / (true positive + false negative)***

Both the precision and recall of a perfect classifier equal to 1.

The F1 measure combines precision and recall:

- *F1 = 2 × (Precision × Recall) / (Precision + Recall)*

Confusion matrix

In supervised ML problems,

- we train the model on a set of historical data, using **both inputs and outputs**;
- we test the model with its test data. We **use the inputs** of the test data.

We then compare the predictions given by the trained model with the actual **output values of the test data**.

The confusion matrix provides information on

- how successfully the classification was doing and
- where the trained ML-algorithm makes mistakes and becomes confused.

		predicted sell	buy	
actual	sell	15 (Correct)	5	20
	buy	5	20 (Correct)	25

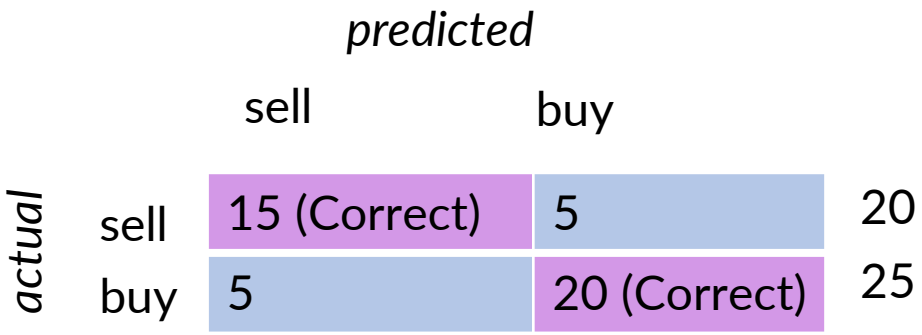
We are interested in selling and buying recommendations. Having calibrated our classification model with the training data, we apply the model to test data. Then, we compare the predicted sell and buy recommendations to the real ones.

The poll can be found at:
presemo.aalto.fi/firma3

Confusion matrix, POLL

What is the overall accuracy of the model?

A. 35/45
B. 20/45
C. 15/25
D. 5/20



We are interested in selling and buying recommendations. Having calibrated our classification model with the training data, we apply the model to test data. Then, we compare the predicted sell and buy recommendations to the real ones.

Multi-class classifiers and the confusion matrix

Example: a model predicting whether a customer invoice will be paid

- on time,
- late, or
- very late.

		predicted			
		on time	late	very late	
actual	on time	42	6	3	51
	late	6	20	3	29
	very late	2	4	7	13

Two types of data sets

Training set: (typically 2/3 of the data)

- we select the features
- we fit model parameters

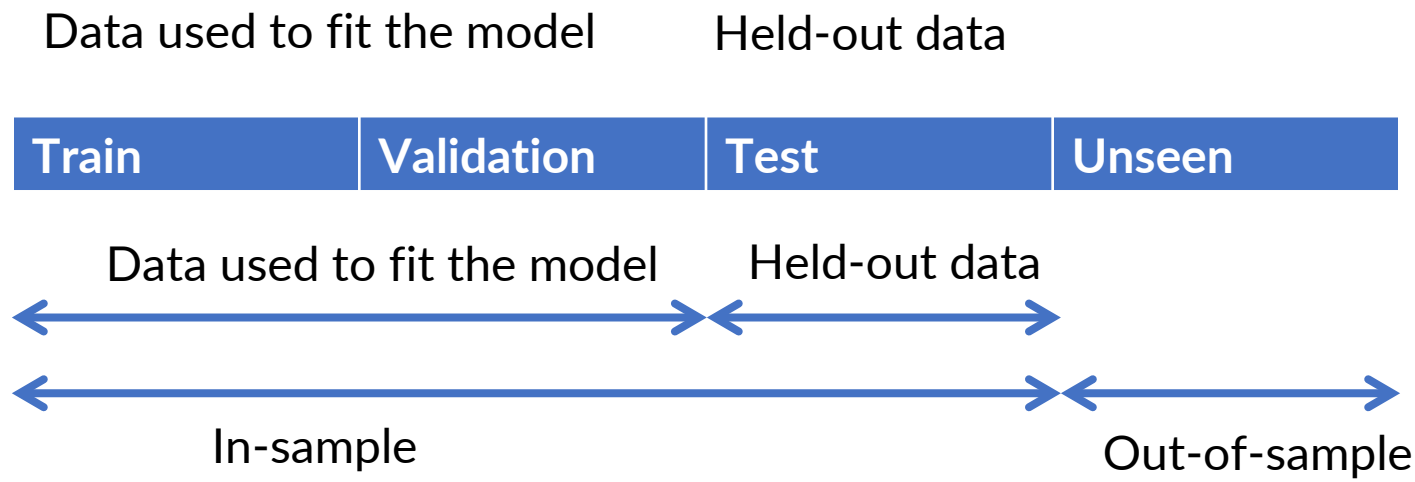
Training set errors:

- errors estimated from the train set

Test set (unseen data):

- not used for model calibration

The amount of overfitting can be estimated from the held-out data.



Exercise 2C: Pairstrading

Equity market neutral is a long/short strategy variation where the investor holds a market neutral portfolio, i.e., it is constructed to have approximately zero net market exposure, so that its performance is intended to be largely uncorrelated or only weakly correlated with overall market movements. This strategy exploits *pricing differences in stocks within the same sector, industry, country*.

This strategy creates a hedge against market factors. If the whole sector moves in one direction or the other, a gain on the long stock is offset by a loss on the short. This strategy depends on the ability of the hedge fund manager to pick the stock in an optimal way.

> *For example, we can take a long position in the 5 forest industry stocks that should outperform and a short position in the 5 forest industry stocks that should underperform.*

A truly equity market neutral portfolio are often very highly diversified (holding hundreds of equities) in order to be able to minimize the *sector exposures* (certain markets or industry) and *country exposures*.

Exercise 2C, Pairstrading example

Coca-Cola and PepsiCo are two strongly correlated stocks, making them suitable for pairs trading.

Market Capitalization (2020):
Coca-Cola: $180 billion
PepsiCo: $151.8 billion

Performance Comparison:
Despite both stocks being considered overpriced, PepsiCo has demonstrated better recent performance, which may suggest potential mean reversion opportunities when compared with Coca-Cola.

Dividend Yield:
Coca-Cola offers slightly higher dividends than PepsiCo, which might influence long-term holding costs in a pairs trading strategy.

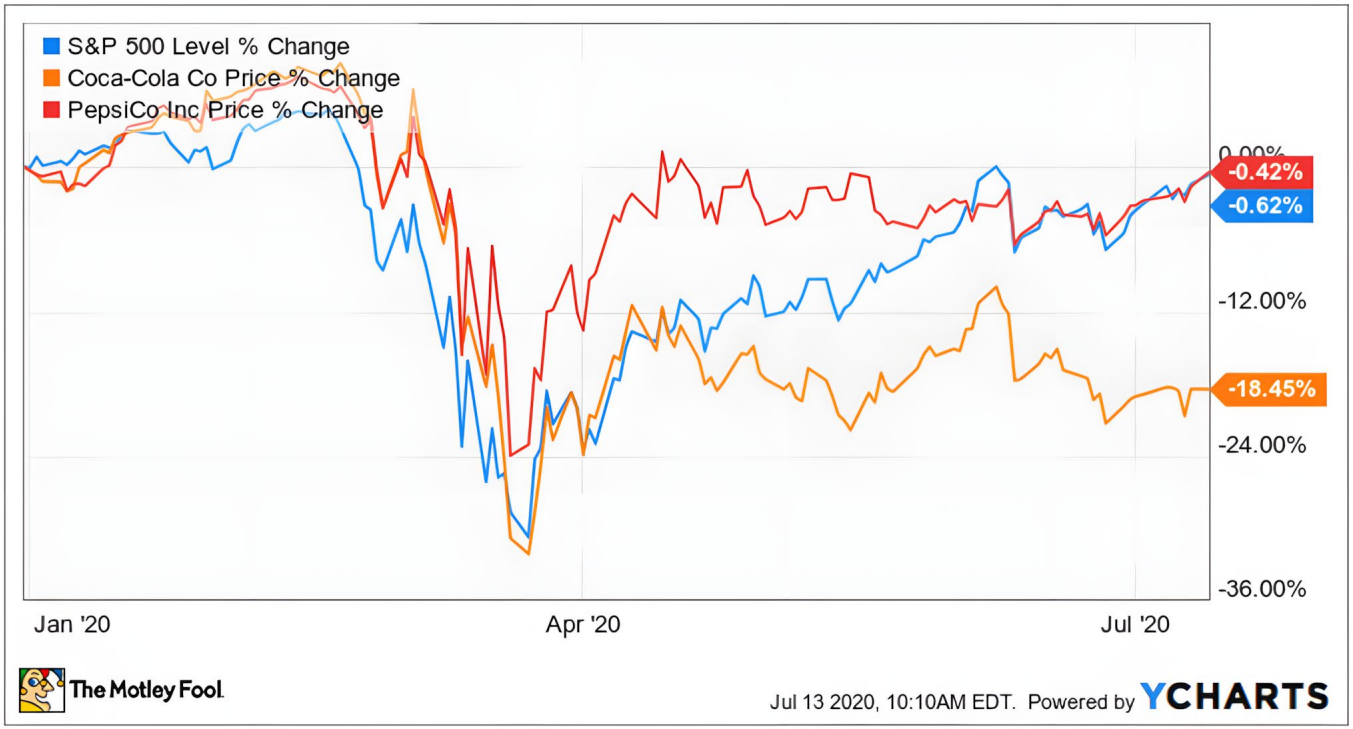
Relative Price Trends:
Historically, the two stocks exhibit high correlation due to their similar industry dynamics, creating potential for arbitrage when one deviates significantly from their usual price ratio.

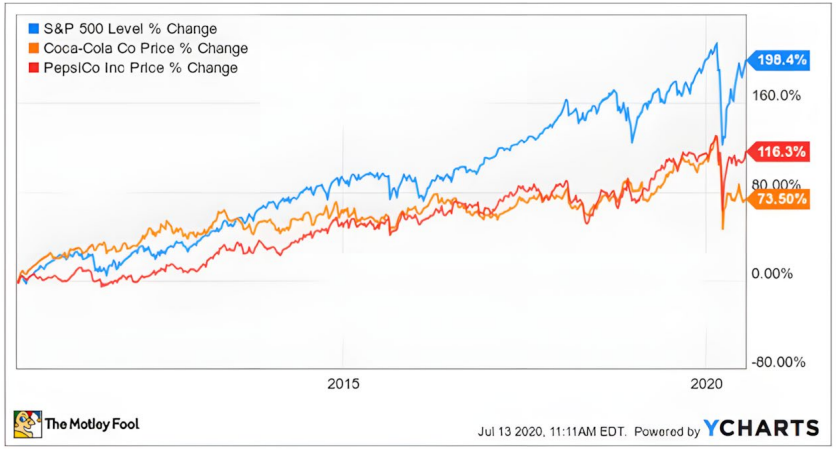
Poll

When following pairs trading, would you take a long position in

- Pepsi cola
- Coca cola

The poll can be found at:
presemo.aalto.fi/firma3


S&P 500 Level % Change
Coca-Cola Co Price % Change
PepsiCo Inc Price % Change
-0.42%
-0.62%
-12.00%
-18.45%
-24.00%
-36.00%
Jan '20
Apr '20
Jul '20
The Motley Fool
Jul 13 2020, 10:10AM EDT. Powered by YCHARTS


S&P 500 Level % Change
Coca-Cola Co Price % Change
PepsiCo Inc Price % Change
198.4%
160.0%
116.3%
80.00%
73.50%
0.00%
-80.00%
2015
2020
The Motley Fool
Jul 13 2020, 11:11AM EDT. Powered by YCHARTS


Next time: Exercises

Self study:
Sample weights, uniqueness

- Assume that we analyze daily stock prices during 1 year (T=252 days).
- Every day, we compute a *h*-day return, with overlapping periods.
- We detect when the return exceed a given threshold and label this event. The total number of labels is L.
- Some of the labels are overlapping as a function of time. The problem of overlapping labels can be addressed with sample weights.

Two labels $y_i$and y_j are concurrent at t, when both are a function of at least one common return $r_{t-1,\,t} = \frac{(p_t - p_{t-h})}{p_{t-h}}$, where p_t is the stock price and *h* means that we are computing a *h*-day return.

- For each observation $t = 1, \ldots, T$, we form a binary array, $\{1_{t,i}\}_{i=1,\ldots,I}$, with $1_{t,i} \in \{0,1\}$ is an indicator function, which gets value 1 if its outcome spans over the return $r_{t-1,t}$, otherwise 0.
- We compute the number of labels concurrent at t, $c_t = \sum_{i=1}^{I} 1_{t,i}$.
- The uniqueness of a label i at time t is $u_{t,i} = 1_{t,i} c_t^{-1}$.

Source: de Prado

The average uniqueness of the label i is the average of $u_{t,i}$ over the label's life span

$$\bar{u}_{t,i} = \left(\sum_{t=1}^{T} u_{t,i}\right)\left(\sum_{t=1}^{T} 1_{t,i}\right)^{-1},$$

where *T* is the number of observations (days). Sample weights can be defined as the sum of the attributed absolute log-returns, $\left|r_{t_{i-1}\, t_i}\right|$, over the event's life span $\left[t_{i,0}, t_{i,1}\right]$,

$$\tilde{w}_i = \left|\sum_{t=t_{i,0}}^{t_{i,1}} \frac{r_{t-1,t}}{c_t}\right|^{-1}$$

$$w_i = \tilde{w}_i L \left(\sum_{j=1}^{L} \tilde{w}_j\right),$$

where *L* is the number of labeled events. The rationale for this method is that we weight an observation as a function of the absolute log returns that can be attributed uniquely to it.

Source: de Prado