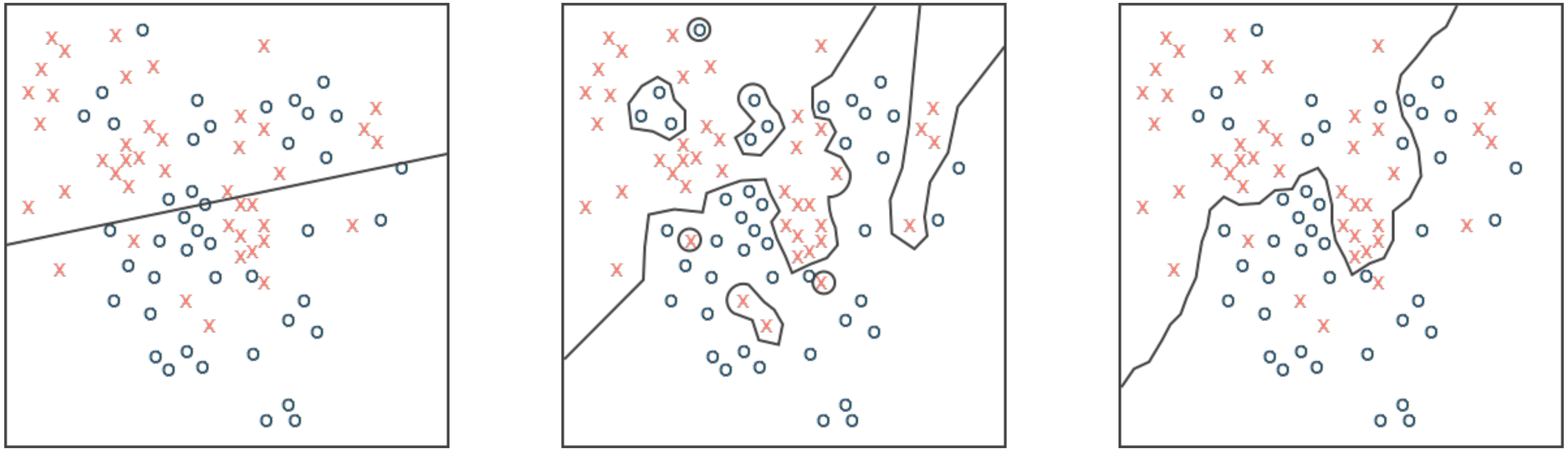
DATAPRED

SPECIFICITIES OF MACHINE LEARNING FOR TIME SERIES

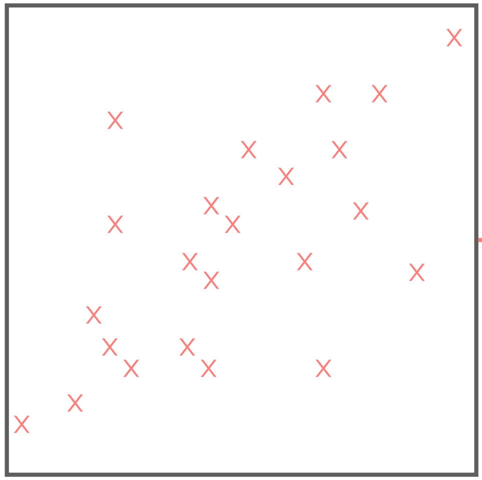
How machine learning works (1/2)

« Machine learning algorithms detect patterns and learn how to make predictions and recommendations by processing data and experience, rather than by receiving explicit programming instructions. » - McKinsey



1. Select an algorithm that can identify patterns in large and diverse data sets
2. Launch the algorithm on your data and let it learn
3. Choose the right balance between accuracy and adaptability

How machine learning works (2/2)



Traditional statistics

- « This looks like y = ax + b
  Let's find a and b »

> Parametric

Machine learning

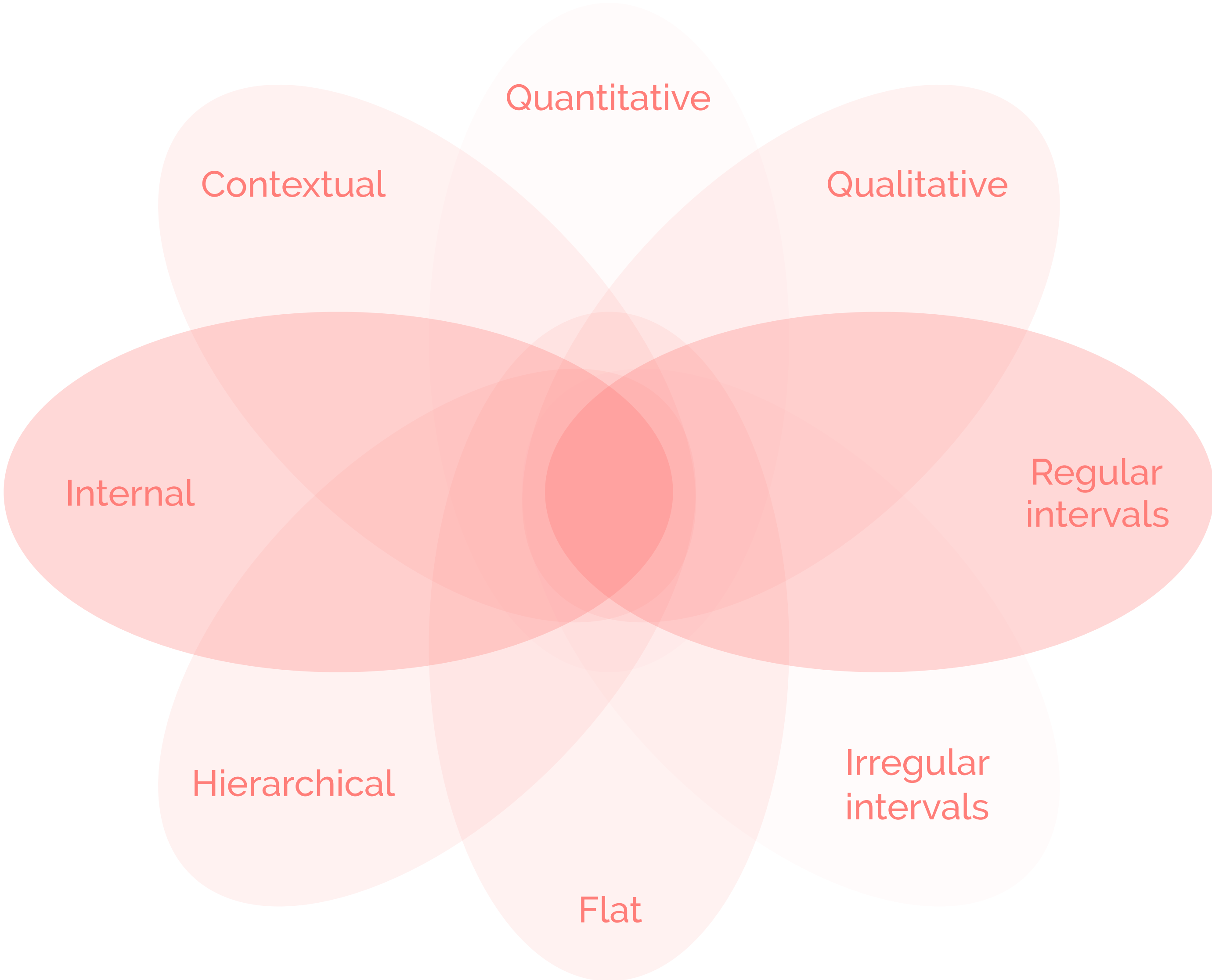
- « Let's use algo xyz to find the dominant pattern here »

> Non-parametric

Time series come in many stripes…

Retail sales
Machine logs
Banking transactions
Sensor data
Logistics scans
Website visits
Calendar events
Weather information
Stock prices
Inventory levels
Satisfaction scores
CRM logs
Transport orders
Energy consumption
Industrial yields
…

1. **Sequentially revealed**
2. **Time stamped**
3. **Time critical**



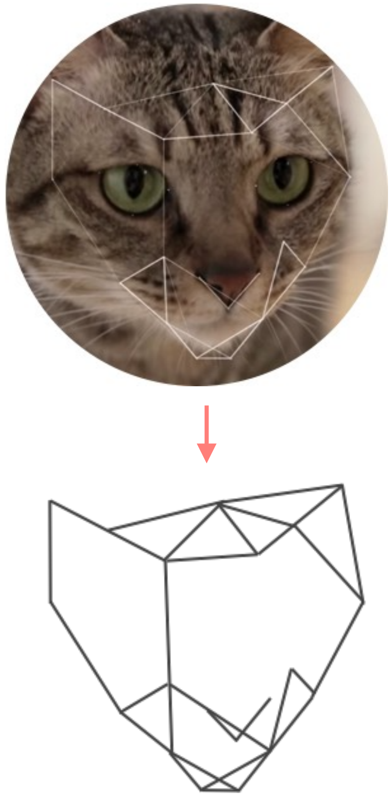
…and their proliferation is transforming every industry

	OLD		NEW
Marketing	Customer segmentation and churn prediction based on stable customer characteristics	▶	Based on real-time consumer patterns (transactions, website visits, in-app navigation…)
Demand prediction	Statistical analysis of historical sales	▶	Sequential machine learning with dozens of internal and contextual data sources
Industrial optimization	« Predictive maintenance » based on static clustering and alert rules	▶	Sequential performance and quality monitoring
Logistics	Operations research everywhere	▶	Dynamic allocation/optimization based on activity predictions
Purchasing	Rule-based replenishment	▶	Dynamic purchase optimization based on raw material price predictions

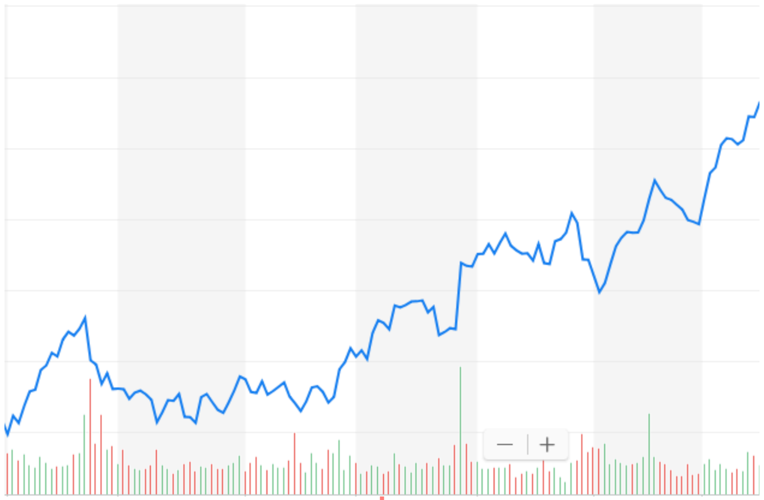
First specificity of time series modeling
No underlying structure (1/2)

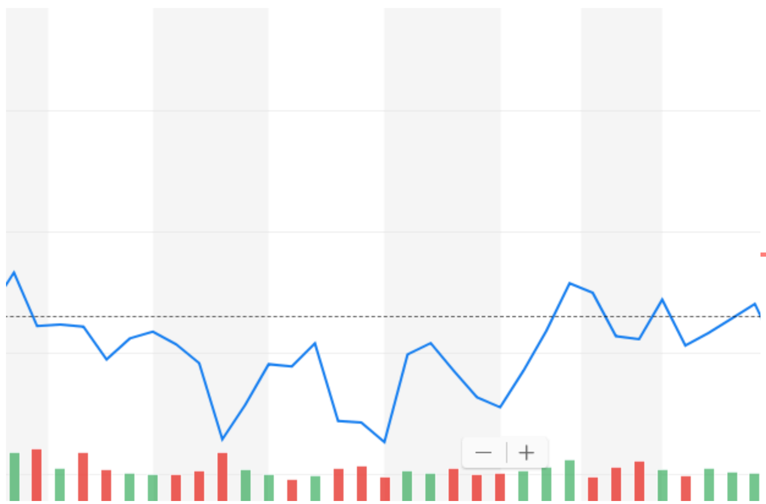
In most cases, data is complex…

…but its underlying structure is stable



First specificity of time series modeling
No underlying structure (2/2)

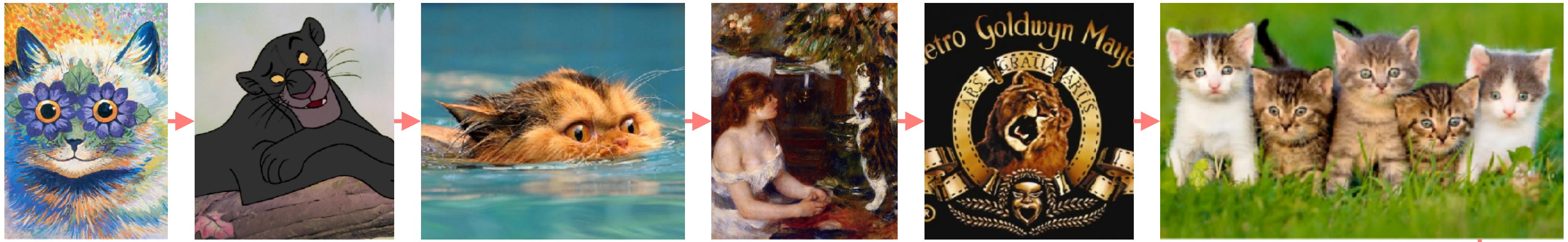




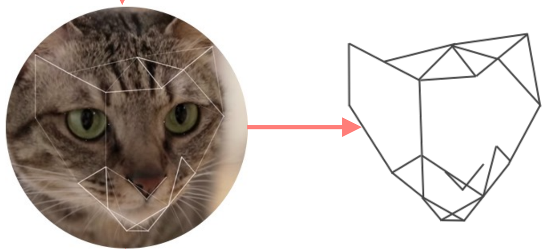
- Same stock price at different time periods, over different time intervals

> Where is the underlying structure?

Second specificity of time series modeling
Sequence matters for learning (1/2)



Two sequences of the same images
➔ same lesson: this is how a cat face is structured

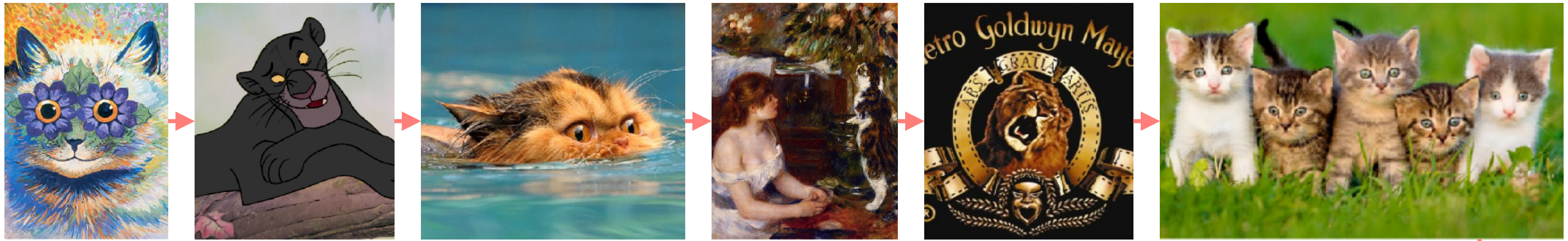


Second specificity of time series modeling
Sequence matters for learning (2/2)

« This looks like a stock price. Let's try to predict future values based on past patterns. »

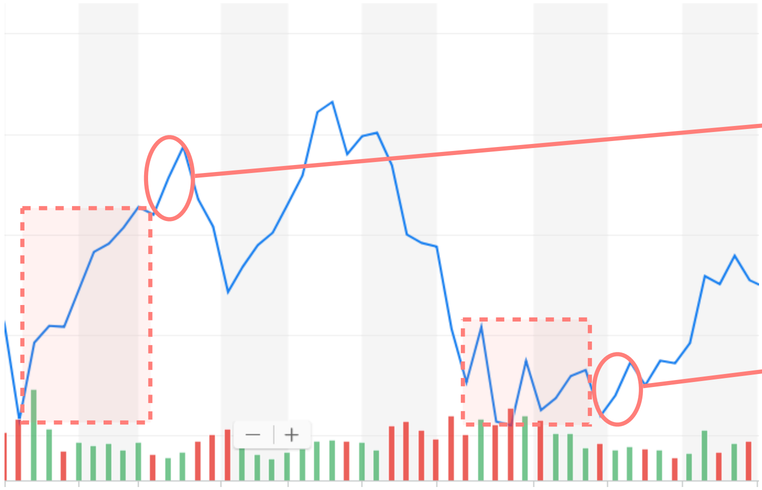
??? Nothing to learn

Third specificity of time series modeling
Sequence matters for interpretation (1/2)



Two sequences of the same images
➔ same interpretation: we are looking at cats

Third specificity of time series modeling
Sequence matters for interpretation (2/2)



Sharp increase consolidating the preceding gains.

Sharp increase continuing a period of high volatility.

The journey to nonstop intelligence

1970s-1980s	1990s-2000s	From mid-2000s	Today
Business reporting	Business intelligence	Batch machine learning	Sequential machine learning
> Collecting, centralizing and sorting business data	> Finding, quantifying and visualizing relationships in structured business data	> Same as BI, but with non-parametric models capable of handling big and unstructured data	> Learning models and delivering insights continuously
Financial consolidation	Market segmentation, dynamic reporting	Recommendation engines, search engines	Sequential predictions, dynamic allocation and anomaly detection
IBM, JD Edwards, Crystal Reports…	Business Objects, Cognos, Hyperion, SAS…	AWS, Google, Microsoft…	

It's not about algorithms, it's about how you train and test them

Many algorithm families in machine learning

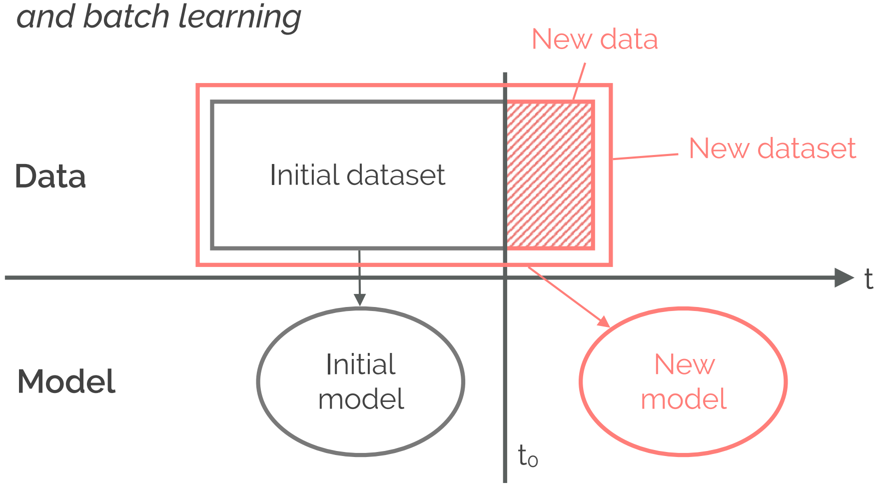
- Decision trees, support vector machines, logistical regressions, neural networks, generalized additive models…

They can all be batch or sequential, depending on how they are trained and tested

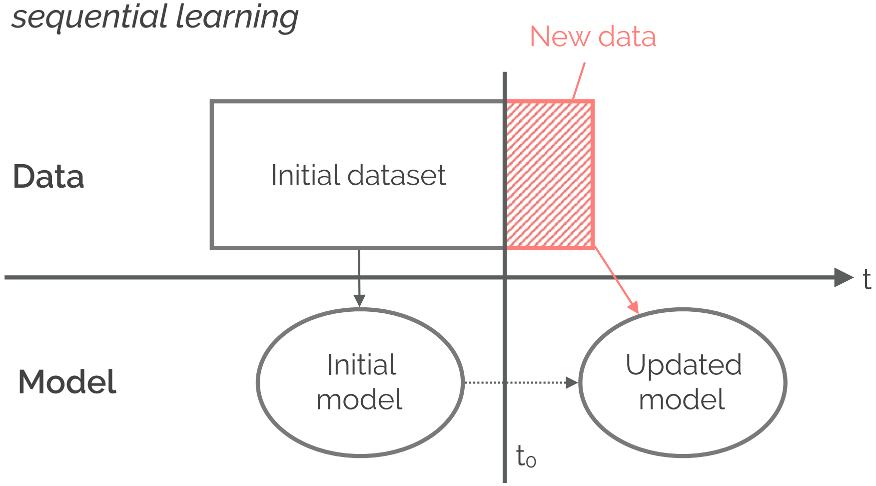
- Don't be blindfolded by buzzwords

The fundamental difference between batch and sequential learning

Time series and batch learning

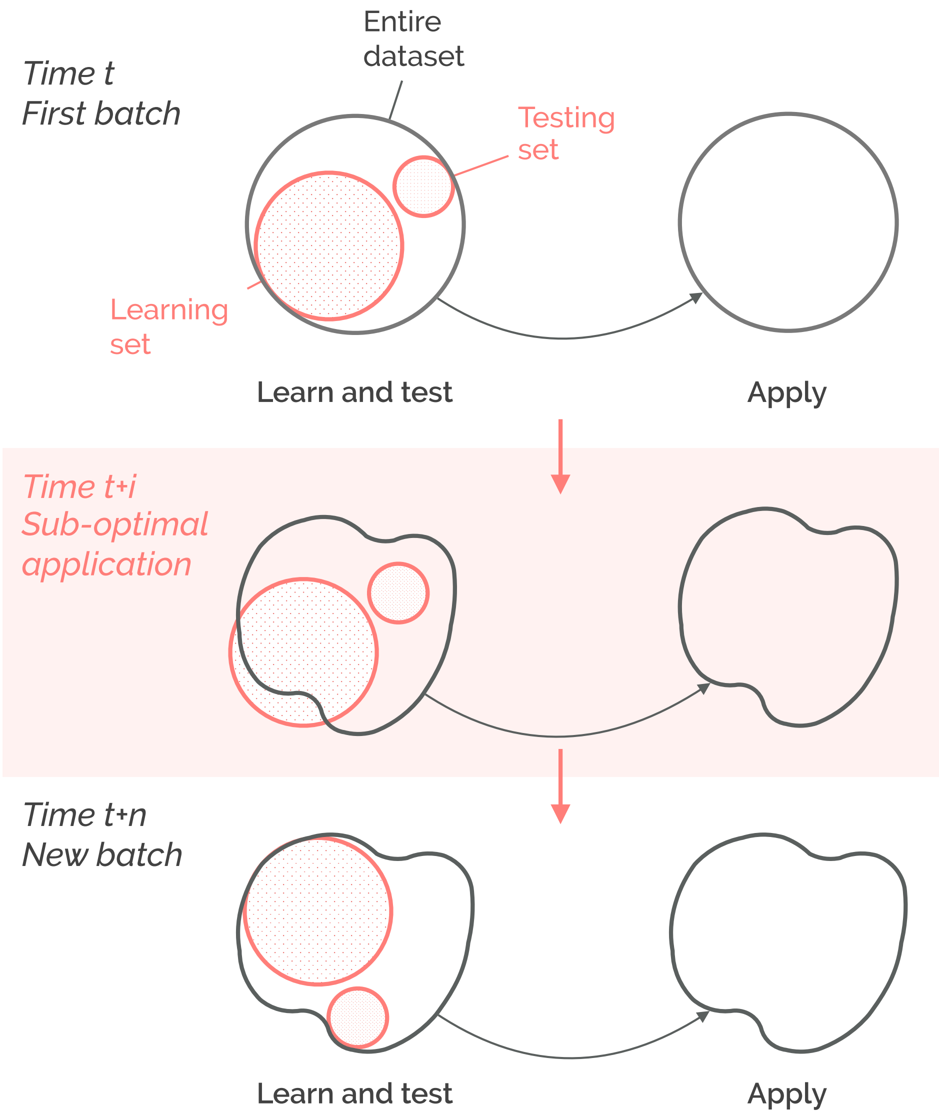


Time series and sequential learning



- Batch learning
  - Gather all the data available at time t_0
  - Learn and test on that data, making sure your training and testing sets are <u>distinct</u>
  - Apply your model for a while
  - Start all over again
- Sequential learning
  - Gather all the data available at time t_0
  - Learn and test, making sure your testing set is <u>posterior</u> to the training set
  - Update your model sequentially every time next data appears

Consequences of applying batch machine learning to time series



- ✘ Gradual performance degradation as the underlying data structure evolves
- ✘ Translates into falling prediction accuracy or an increasing ratio of false positives
- ✘ Often requires manual re-calibration of the solution
- ✘ Inability to tackle strictly time-dependent challenges, like performance monitoring

No laughing matter
The infamous example of Microsoft's Tay

So cute! → **Hold on…** → **OMG!!!**

> Bad things happen in the real world when ML solutions don't adapt to changing conditions

Key requirements of sequential learning (1/3)
Sequentialized data

Sequential learning and testing sets

- Beware of the future leaking into the past!
- Testing sets should represent diverse data regimes

Sequential feature engineering

- Management of feature time availability
- Management of feature fluctuations

Alignment of predictions and targets

- Management of multiple prediction horizons
- Back and worth between the dates on which/for which predictions are made

Key requirements of sequential learning (2/3) Adapted algorithms

Stationarization

> Fighting the lack of an underlying structure

- Stationarization at every step of the backtests…
- …and for each prediction horizon

Sequential aggregation

> Adapting to regime changes

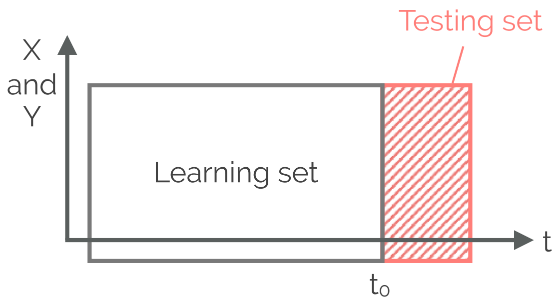
- Managing the portfolio of aggregated models…
- …and the aggregation procedure

Adaptation of batch algorithms

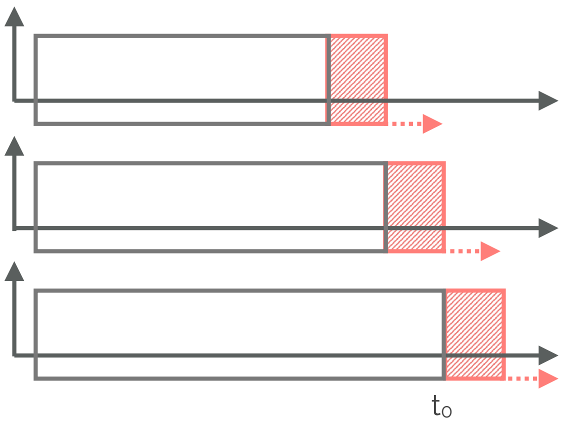
> Learning from the sequence of events

- Teaching batch algorithms to learn sequentially…
- …and to fine-tune in real time

Key requirements of sequential learning (3/3)
Proper backtests



Distinction between learning and testing sets



Sliding window

- The problem: « the future lasts a long time »
  - Testing a model requires its application to data it has never seen
  - For time series, unknown data equals future data
  - But « waiting for the future » to assess a model would mean waiting for quite a while…
- The solution: « backtesting », with two key elements
  - Learning set vs. testing set
  - Sliding window

What's not required 😉

- A brand new « data lake » filled with centuries of perfect data
  - Time-stamped data ≠ data lake
  - Sequential learning becomes applicable with a few dozen data points
  - No real-life data is perfect. Sure, « garbage in, garbage out » is true, but an industry-strength ML solution must be able to cope with the occasional gap or outlier
- Crazy computing power
  - One of the big advantages of sequential over batch learning is its reduced computational power requirements
  - We run most of our projects on standard Linux servers
- Endless soul searching (translating into giant IT projects)
  - ML is eminently practical. Quick feedback loops trump long theoretical reviews…
  - …as long as scalability is kept in mind

Machine learning suite for time series

Datapred secures the 3 steps of time series modeling — data engineering, modeling, and backend optimization — providing unparalleled speed, flexibility and performance for time-dependent challenges.

contact@datapred.com

www.datapred.com