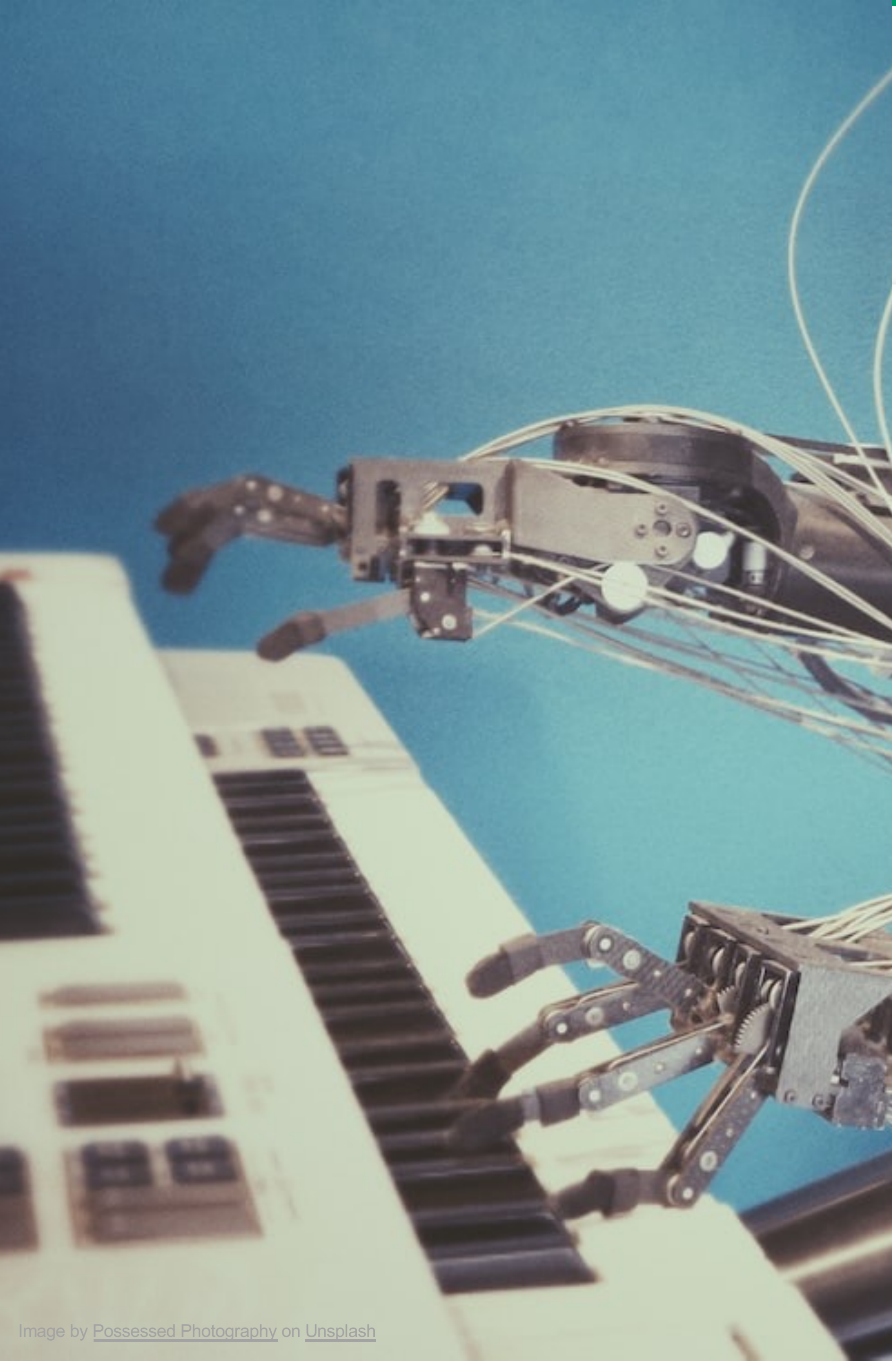


Image by Possessed Photography on Unsplash

# Applied Artificial Intelligence
# 05 - AI Lifecycle: Concept Drift


**Univ.-Prof. Dr-Ing. habil. Niklas Kühl**
**www.niklas.xyz**

University of Bayreuth

Karlsruhe Institute of Technology

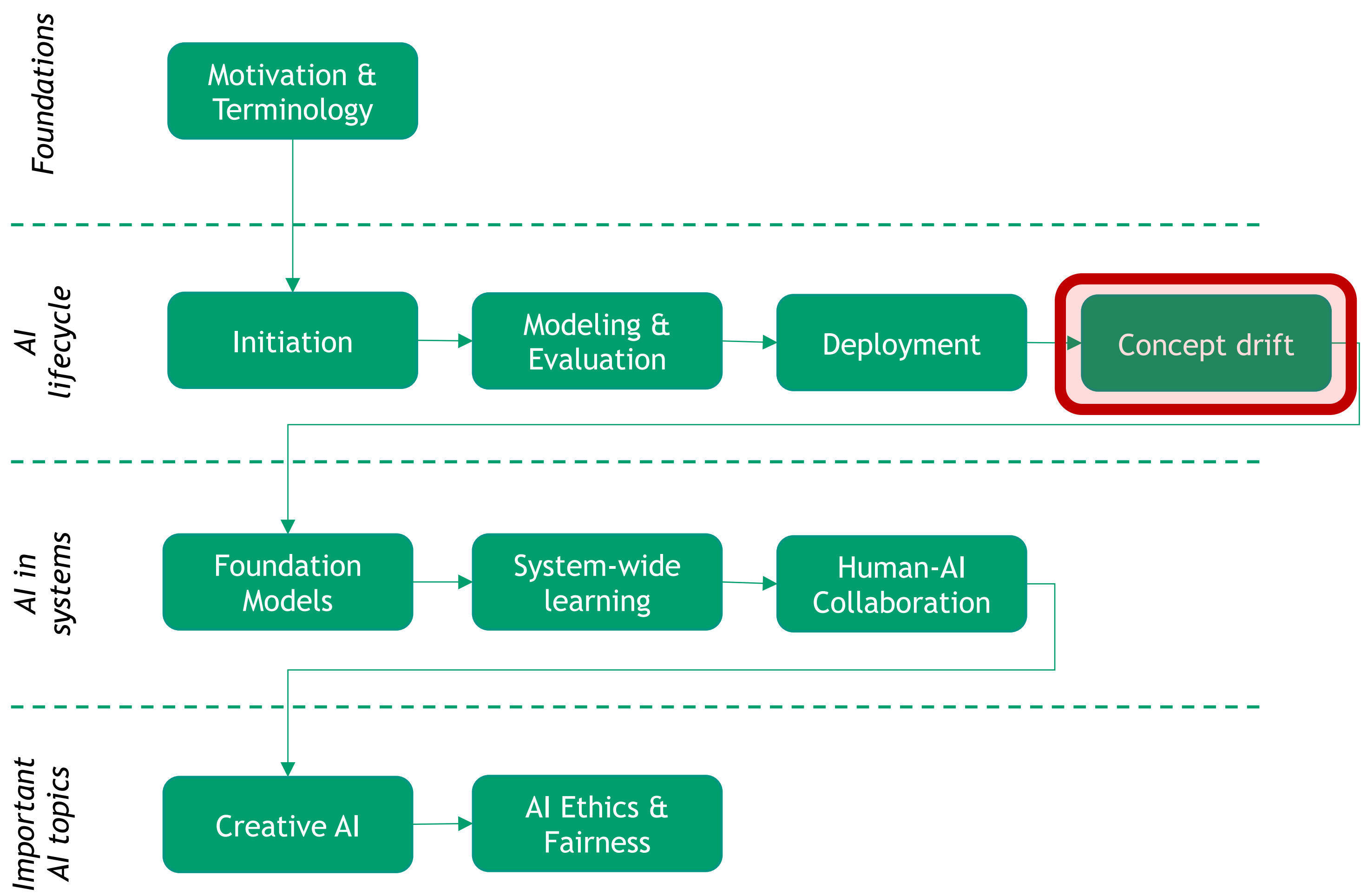
TUM School of Management


www.uni-bayreuth.de | www.kit.edu | www.tum.de | www.fim-rc.de | www.wirtschaftsinformatik.fraunhofer.de

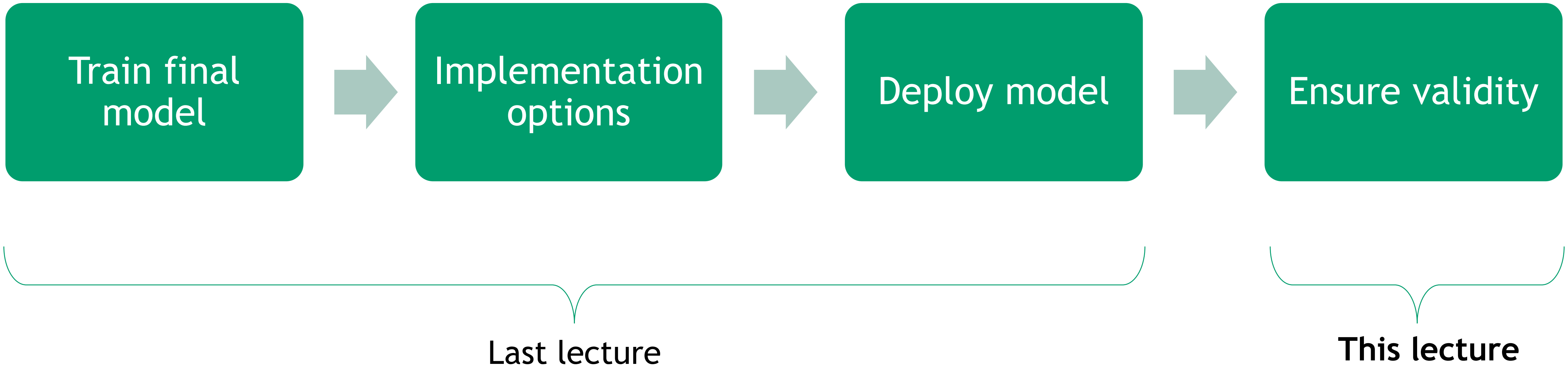
# Repetition

| | |
|---|---|
| **Training vs. deployment** | You learned that model training and testing come before operational deployment, involving iterative fine-tuning. |
| **AI autonomy levels** | You discovered that AI applications can range from making predictions to offering recommendations and even taking direct actions. |
| **Challenges of ML deployment** | You gained insights into dealing with multilingual support, harnessing the power of parallel GPUs, and managing unpredictable costs. |
| **Deployment decisions** | You learned to choose between Monolithic and Microservices, and various deployment options, including On-Premise, IaaS, PaaS, FaaS, and SaaS. |
| **Platforms for ML deployment** | You explored tools like Amazon SageMaker, Kubernetes, Cloud Foundry, and holistic MLaaS solutions. |
| **Concept of MLOps** | You understood that MLOps is a comprehensive paradigm that involves a collaborative team, all working together to streamline the AI deployment process. |

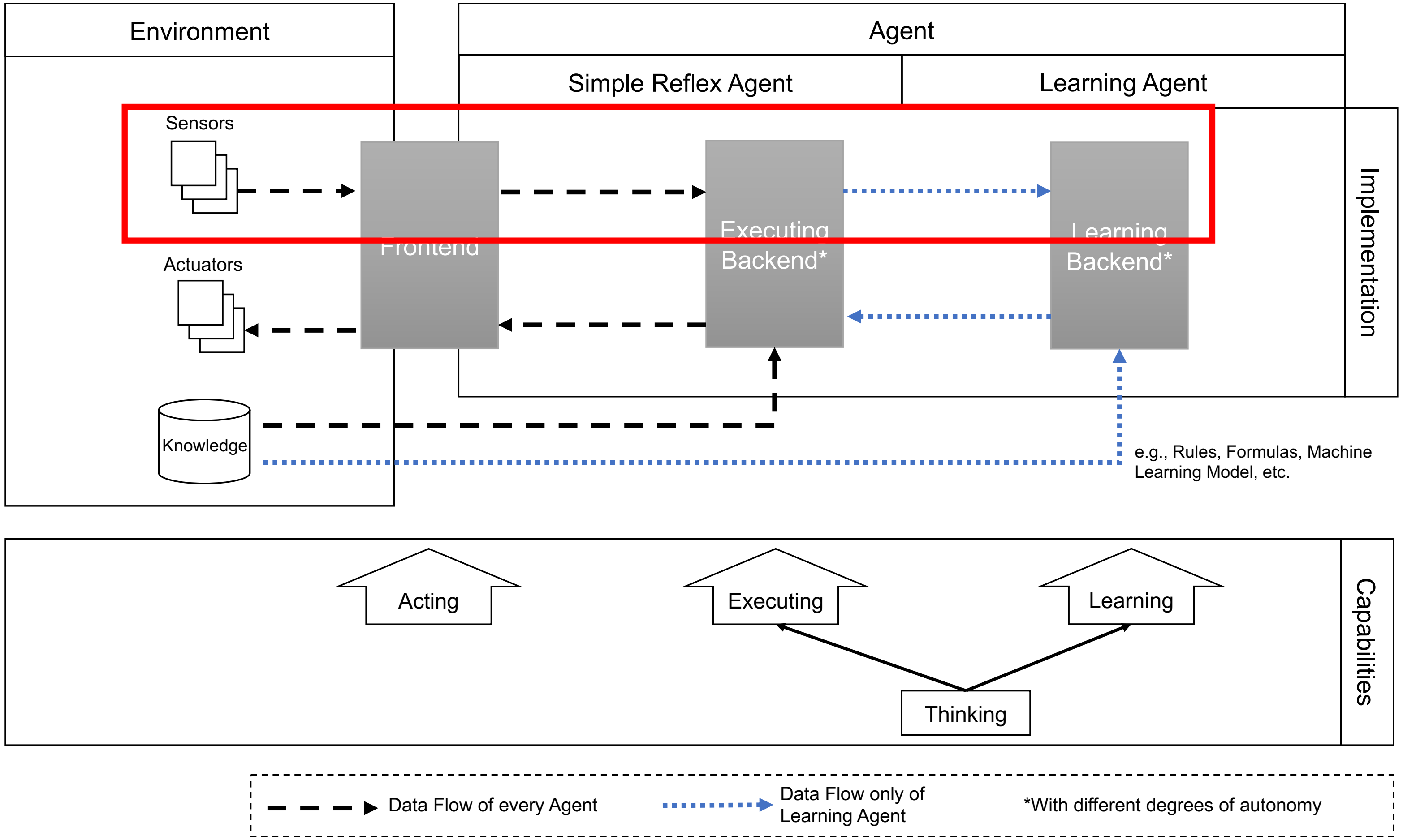
# **Organizational**
# The story of the lecture



# Process


Train final model
Implementation options
Deploy model
Ensure validity
Last lecture
This lecture

# Positioning of the “concept drift” within an intelligent agent’s architecture




Kühl et al. (2022), Artificial intelligence and machine learning [1]

# Objectives
What are the learning goals of this lecture?

| EXPLORE | UNDERSTAND | INTENSIFY | APPLY |
| --- | --- | --- | --- |
| Learn about the term concept drift | Understand the importance of continuous monitoring of deployed AI solutions | Familiarize with concept drift detection | Be able to adapt prediction models |

1 Introduction

2 Theoretical foundations of concept drift & adaption mechanism

3 Two examples for concept drift detection algorithms

4 Real-world examples for concept drift handling

# Introduction
How to keep machine learning (microservices) correct?

- changing context or data over time
- own action taken on machine learning services

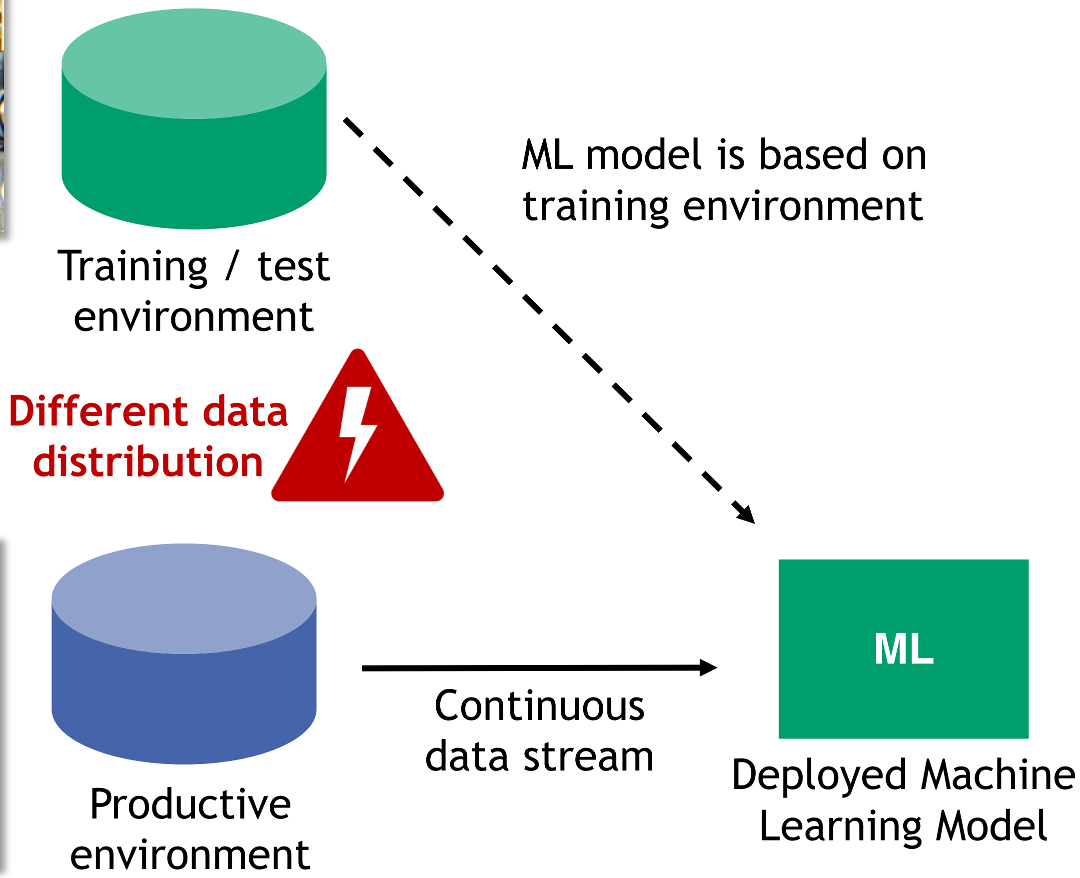



Image source DALL-E

# Introduction

## Data streams may change in various application domains

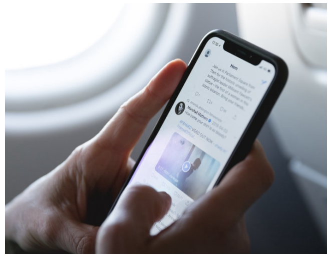
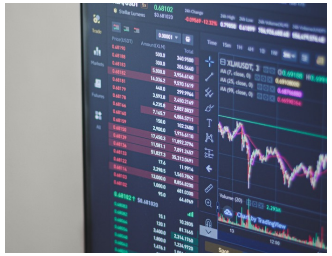
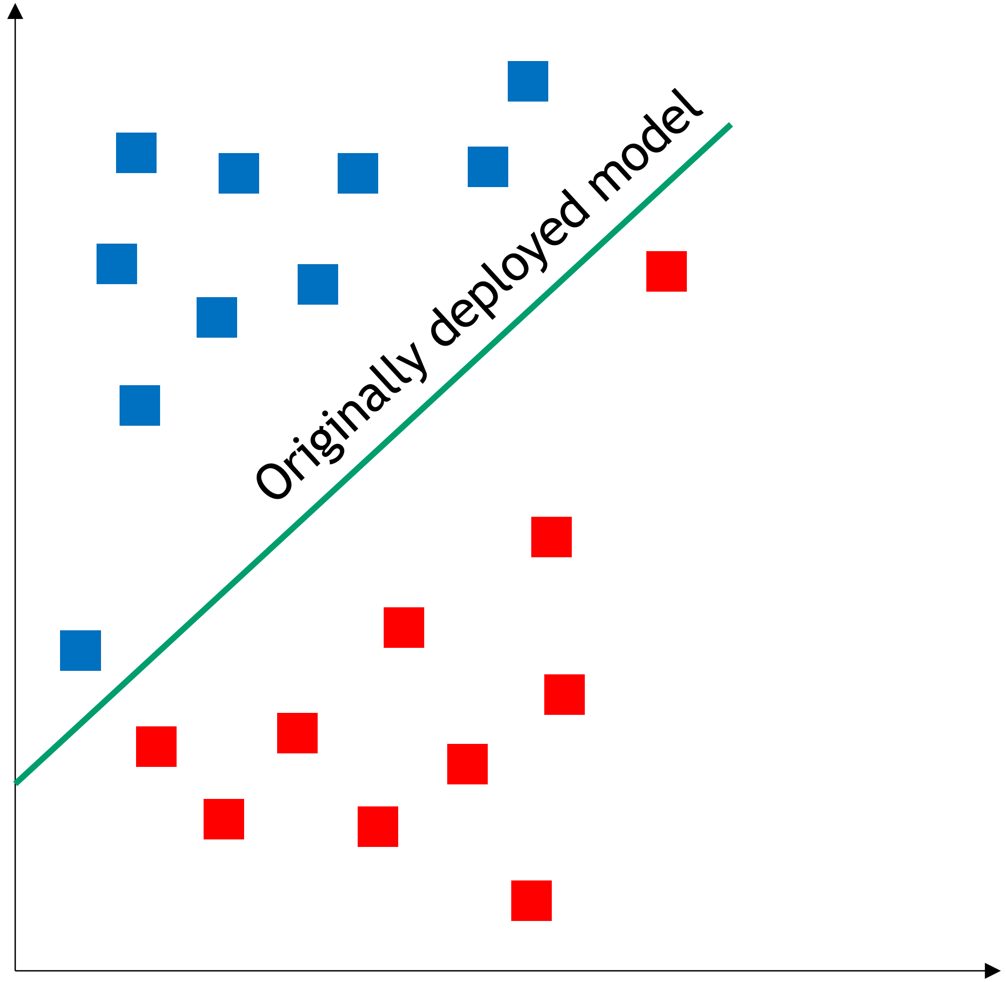
| Changing context/concept of data | | Machine learning | Result |
| --- | --- | --- | --- |
|  | **Sentiment prediction:** Different communication pattern on Twitter (e.g. doubling of characters) | ⇨ | ? |
|  | **Prediction of prices:** Policy changes in electricity markets | ⇨ | ? |
|  | **Prediction of downtimes:** Change of machine parameters in industrial context | ⇨ | ? |

Image source https://unsplash.com, free licence

# Introduction

## Ongoing validity of machine learning models can be ensured with model adaptations




**Constant monitoring**

… of data stream with dedicated change detection module (e.g. based on statistical properties)

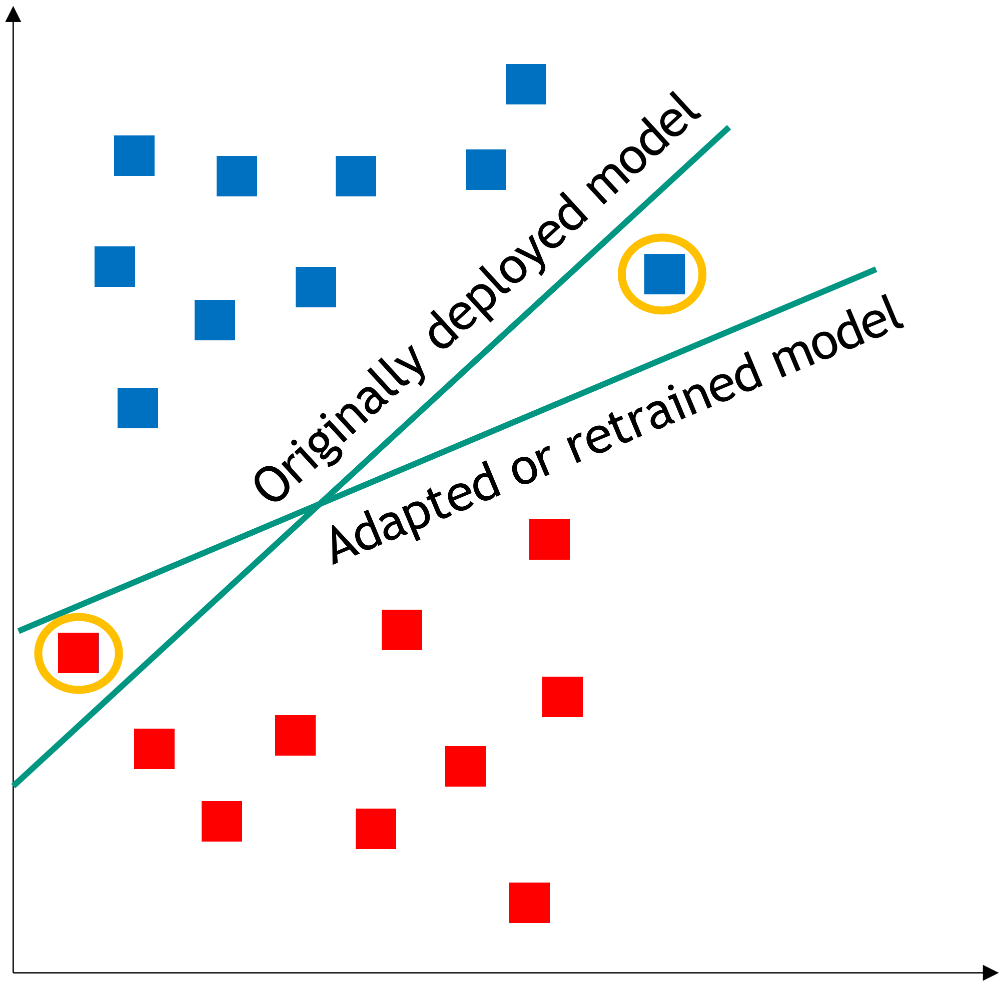
**Change detection**

system decides whether model adaptation or retraining is necessary

- **Adaptation:** Machine learning model is slightly adapted based on the new data
- **Retraining:** Machine learning model is completely retrained from scratch

# Introduction

## Ongoing validity of machine learning models can be ensured with model adaptations




**Constant monitoring**

… of data stream with dedicated change detection module (e.g. based on statistical properties)

**Change detection**

system decides whether model adaptation or retraining is necessary

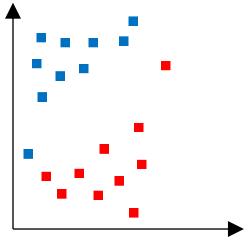
**Adaptation:** Machine learning model is slightly adapted based on the new data

**Retraining:** Machine learning model is completely retrained from scratch

# Theoretical foundations of concept drift & adaption mechanism

Concept drift describes the phenomenon of changing data in computer science

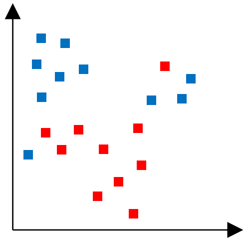
Definition of **concept:**



$$\text{Concept} = P(X, y)$$

**Problem:** In the real world, concepts are not stable but change with time

Definition of **concept drift** between two time points:



$$P_{t0}(X, y) \neq P_{t1}(X, y)$$

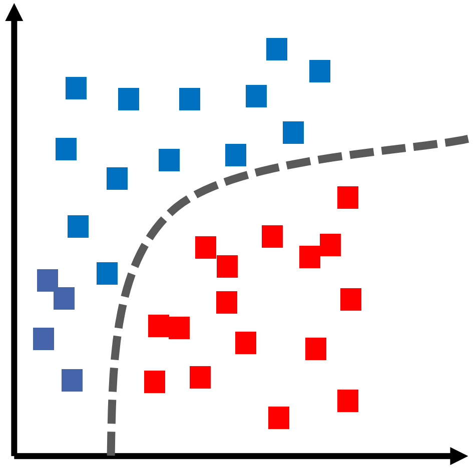
Closely related to **streaming data** with time stamps (e.g., social media data)

Learning algorithm has access to data with time stamps prior to $t$, but needs to be applied to data elements with subsequent time stamps

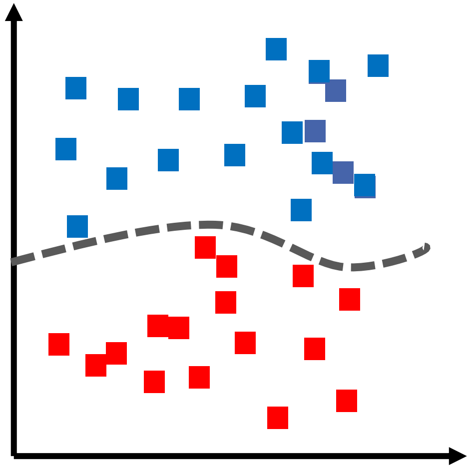
Source: Webb et al. (2016), Characterizing Concept Drift

# Theoretical foundations of concept drift & adaption mechanism

## Different types of concept drift exists

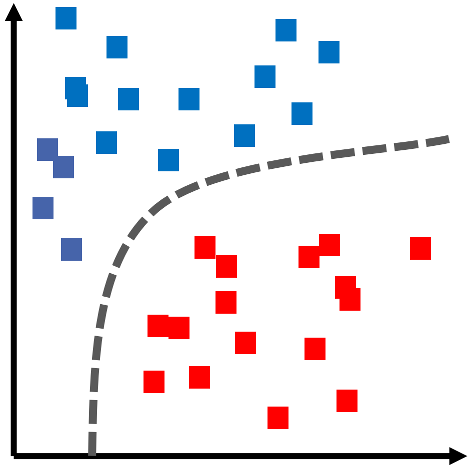


**Original data**



**Real concept drift**

changes in $P(y|X)$, $P(X)$ might stay constant or not



**Virtual drift**

$P(X)$ changes without affecting $P(y|X)$

Source: Gama et al. (2014), A survey on concept drift adaptation

# Theoretical foundations of concept drift & adaption mechanism
## Example: News classification

**Task:**
Classification of online news stream of articles on real estate as relevant or not relevant for a given user

**Scenario:** User is looking for a new apartment

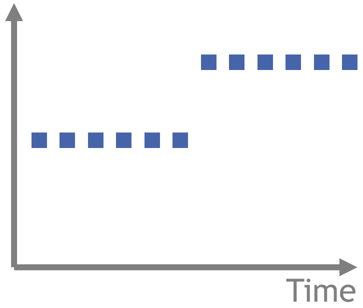
**Virtual drift:**

| | $t$ | |
|---|---|---|
| News on city apartments is relevant; news on holiday homes is not relevant | Editor of news portal changes → writing style | City apartments remain relevant |

**Real concept drift:**

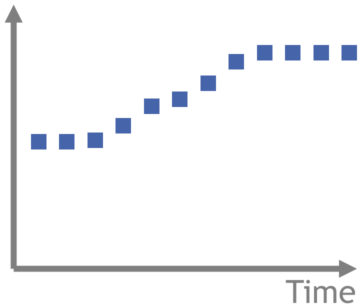
| | $t$ | |
|---|---|---|
| News on city apartments is relevant; news on holiday homes is not relevant | User buys a house → starts looking for holiday home | City apartments become irrelevant, holiday homes are relevant |

# Theoretical foundations of concept drift & adaption mechanism
## Data changes over time in various ways



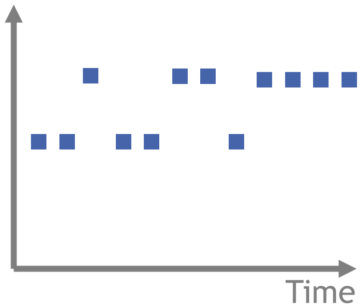

**Sudden changes**

Exchange of a sensor in IOT setting



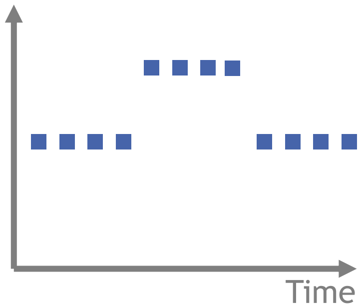

**Slow changes**

Sensor that is slowly degrading, becomes less accurate



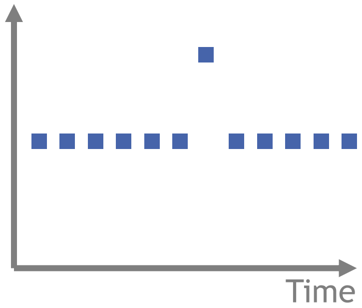

**Gradual change**

User behavior, user interested in finance, later in sports, but keeps looking back at finance




**Reoccurring concepts**

Seasonal pattern for sales forecasting




**Outliers**

Difficulties to not misclassify outliers as concept drift; Otherwise, danger of false adaptation

Source: Zliobaite (2010), Learning under Concept Drift

# Theoretical foundations of concept drift & adaption mechanism

## Challenges for deployed machine learning services in concept drift environments

**Detect**
concept drift as soon as possible

**Adapt**
the model quickly if needed

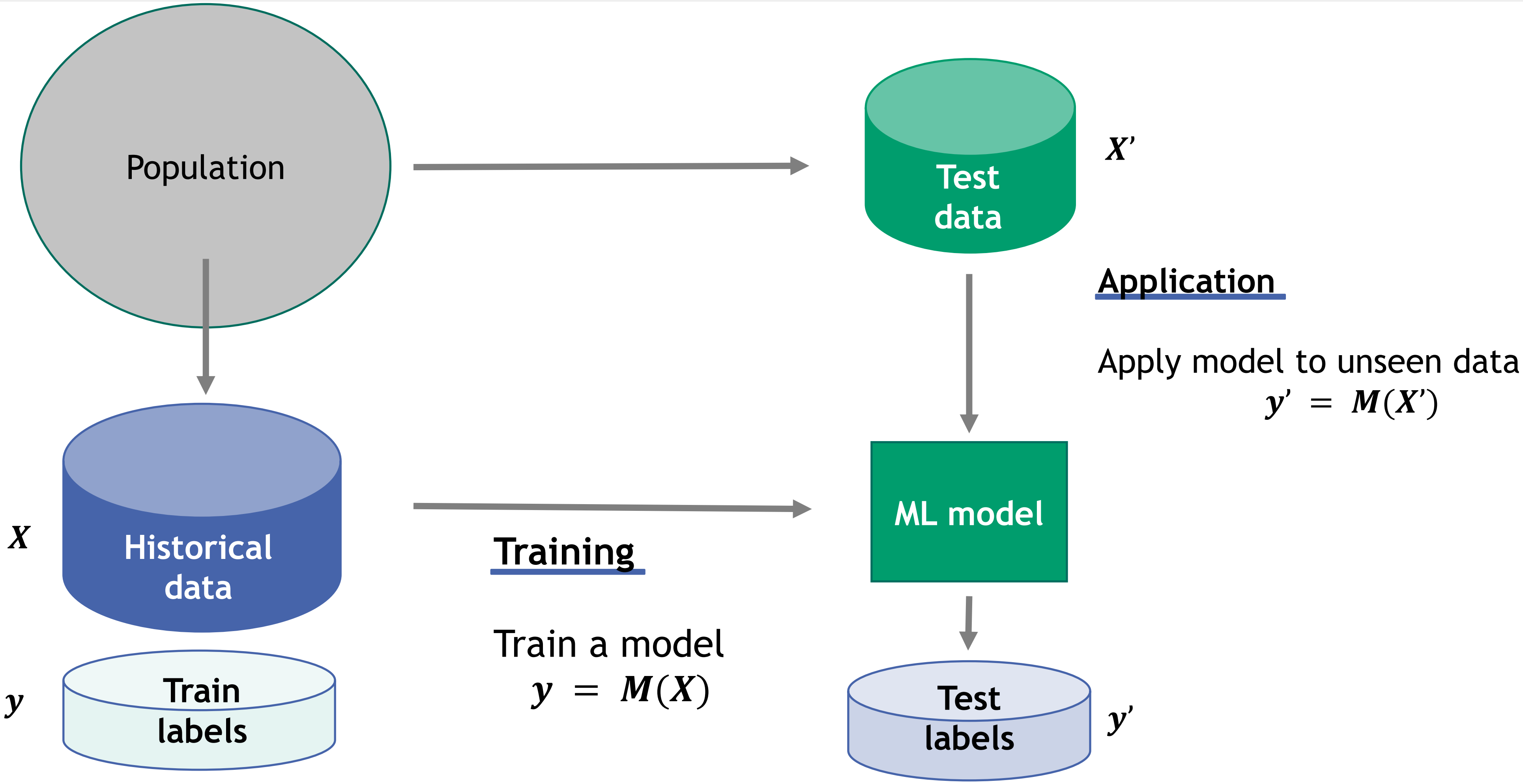
**Distinguish**
drifts from noise; robustness to noise

**Recognize**
recurring contexts (e.g. seasons)

**Operate**
in less than example arrival time and with limited storage available

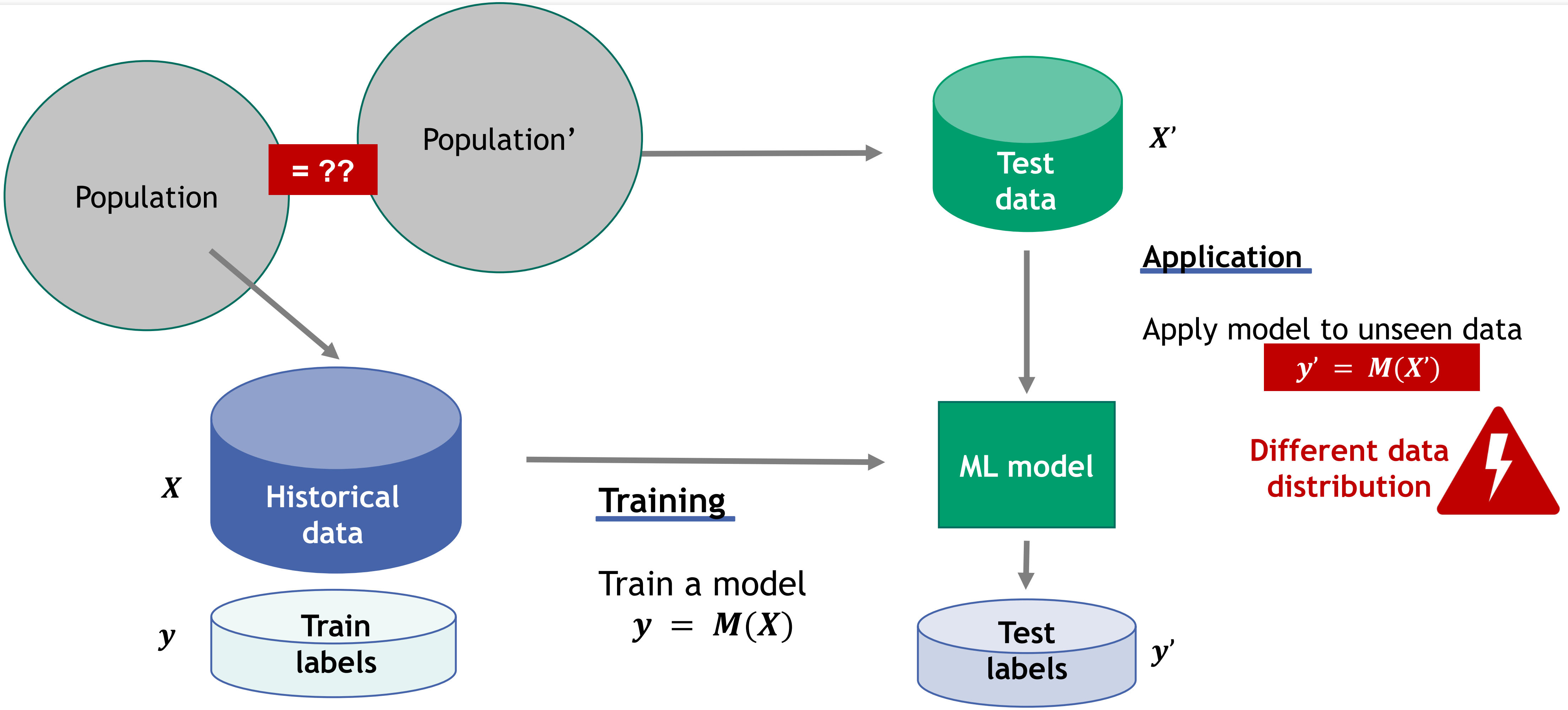
Source: Gama et al. (2014), A survey on concept drift adaptation Tsymbal (2004), The problem of concept drift: definitions and related work

# Theoretical foundations of concept drift & adaption mechanism
## Recap: Supervised learning




Source: Bifet et al. (2010), Handling Concept Drift

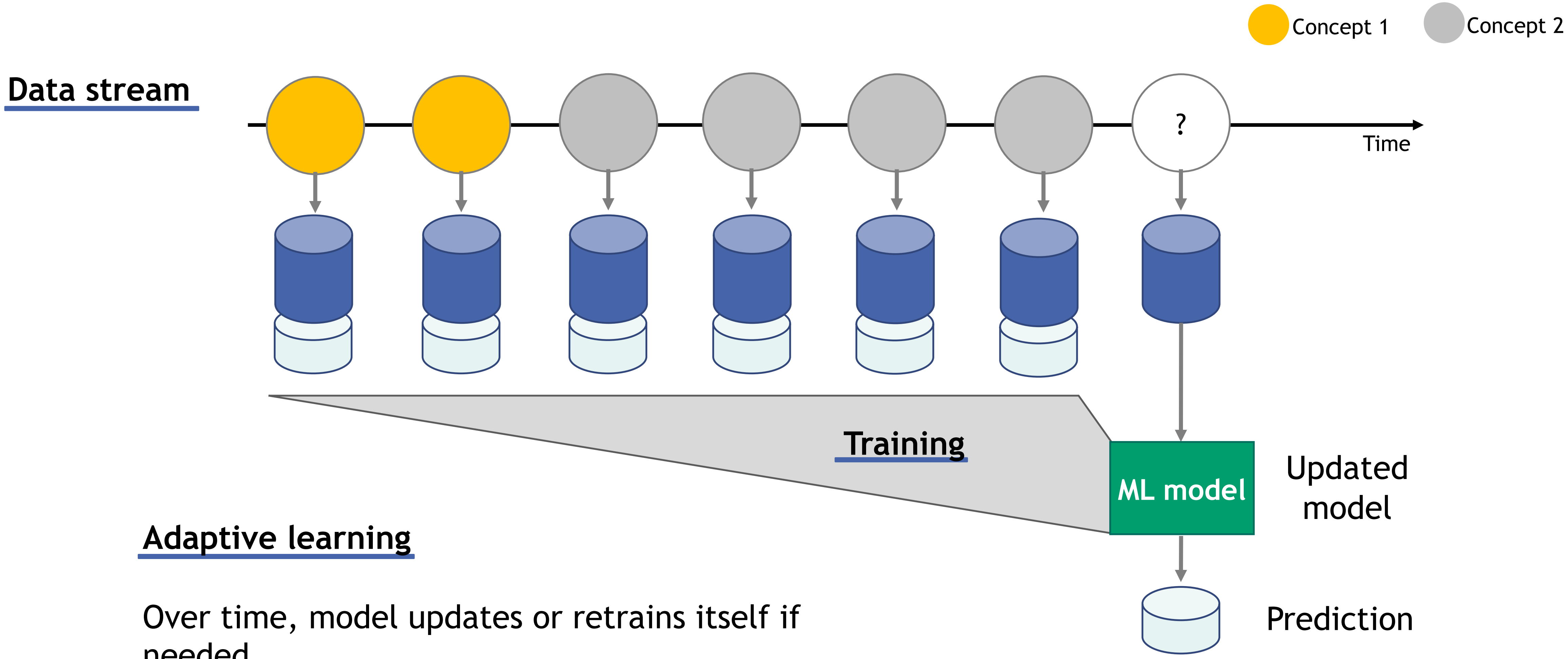
# Theoretical foundations of concept drift & adaption mechanism
Recap: Supervised learning




Source: Bifet et al. (2010), Handling Concept Drift

# Theoretical foundations of concept drift & adaption mechanism

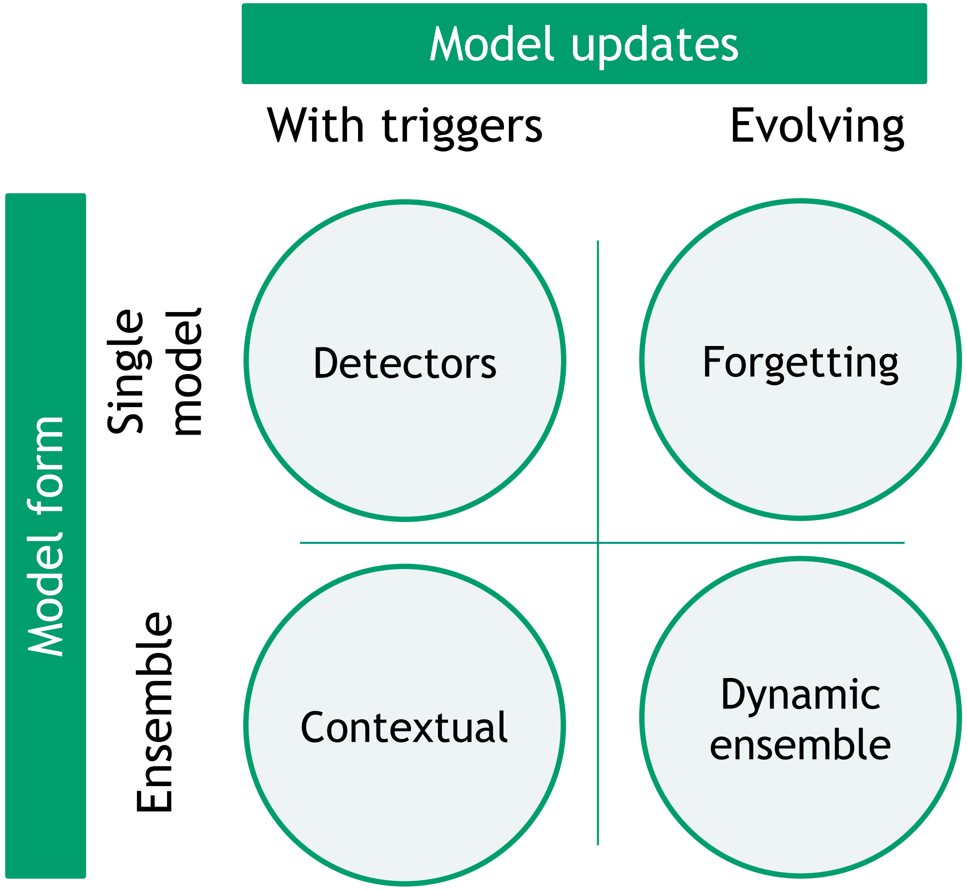
We have to consider the machine learning model in a data stream / online setting




**Adaptive learning**

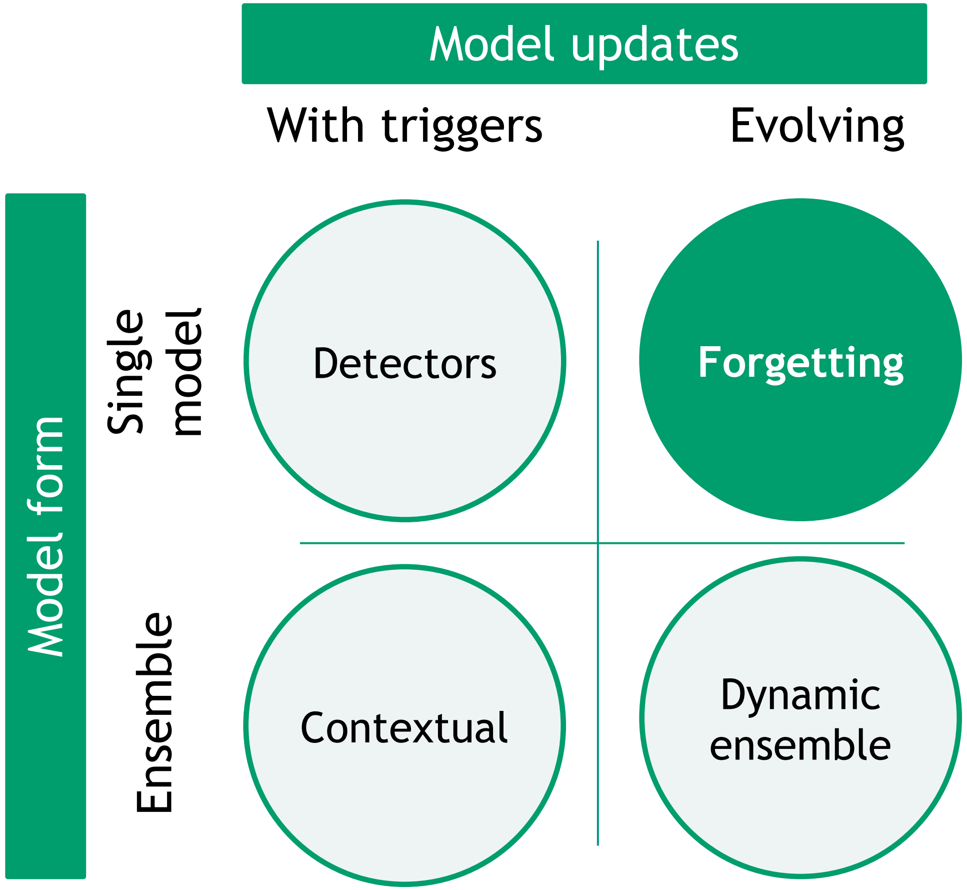
Over time, model updates or retrains itself if needed

# Theoretical foundations of concept drift & adaption mechanism
## Different adaptive learning strategies are available



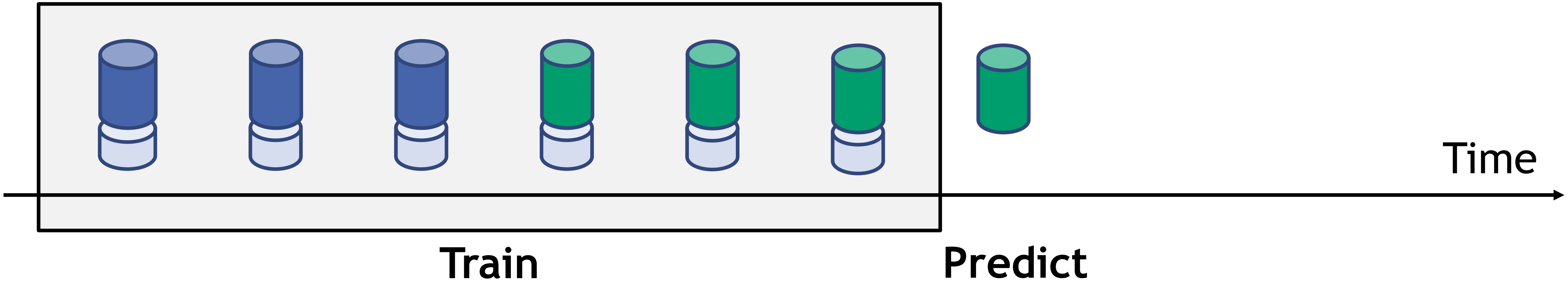

Source: Zliobaite (2010), Learning under Concept Drift

# Theoretical foundations of concept drift & adaption mechanism
## Different adaptive learning strategies are available

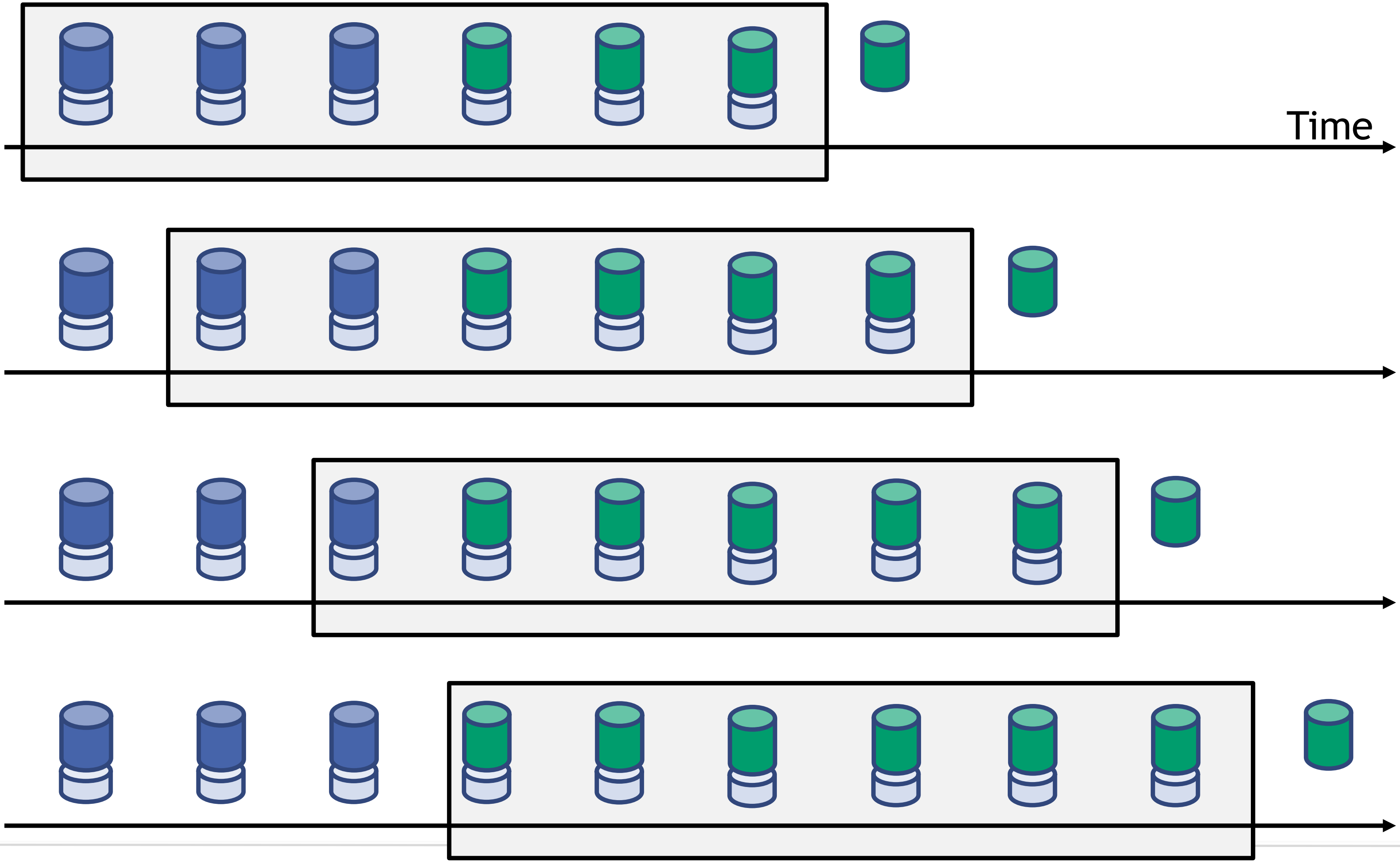



Source: Zliobaite (2010), Learning under Concept Drift

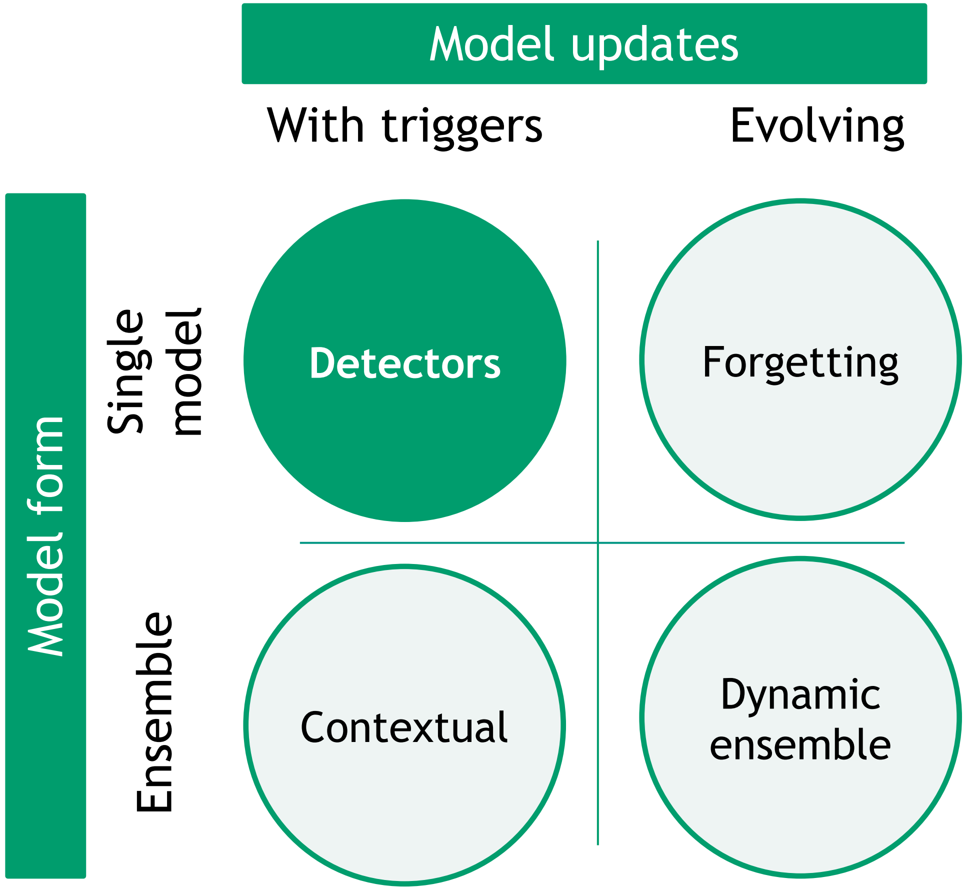
## Theoretical foundations of concept drift & adaption mechanism
A fixed training window ensures forgetting



# Theoretical foundations of concept drift & adaption mechanism
## A fixed training window ensures forgetting

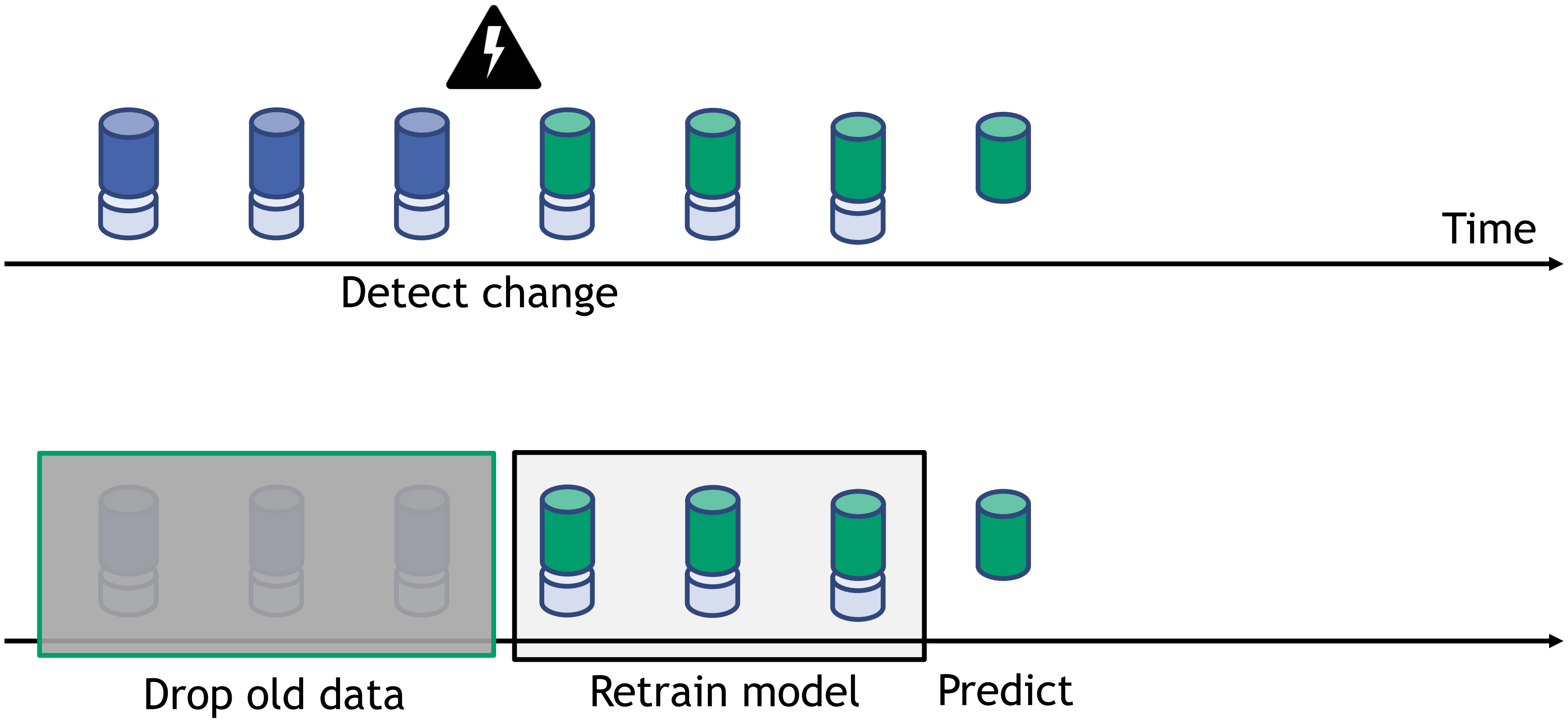


# Theoretical foundations of concept drift & adaption mechanism
Different adaptive learning strategies are available

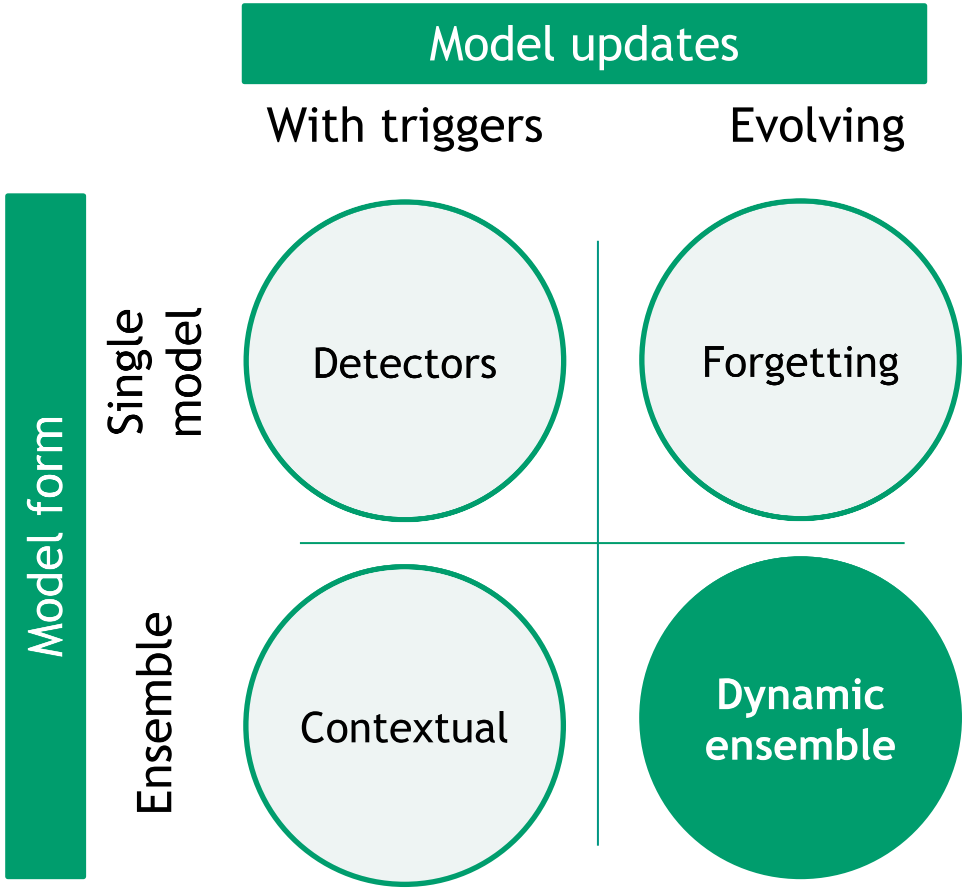



Source: Zliobaite (2010), Learning under Concept Drift

# Theoretical foundations of concept drift & adaption mechanism
## Detection of change leads to deletion of old data

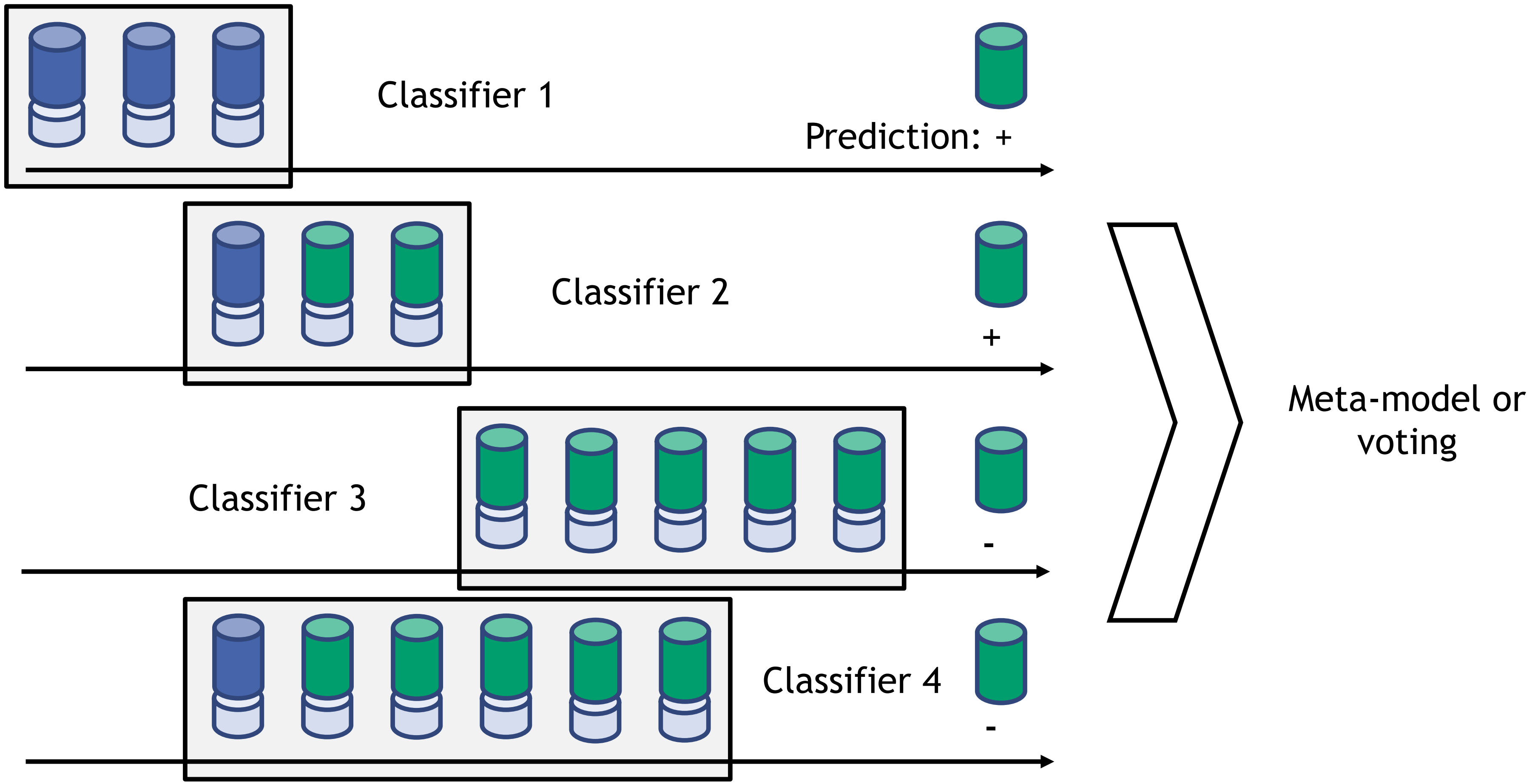


# Theoretical foundations of concept drift & adaption mechanism
## Different adaptive learning strategies are available

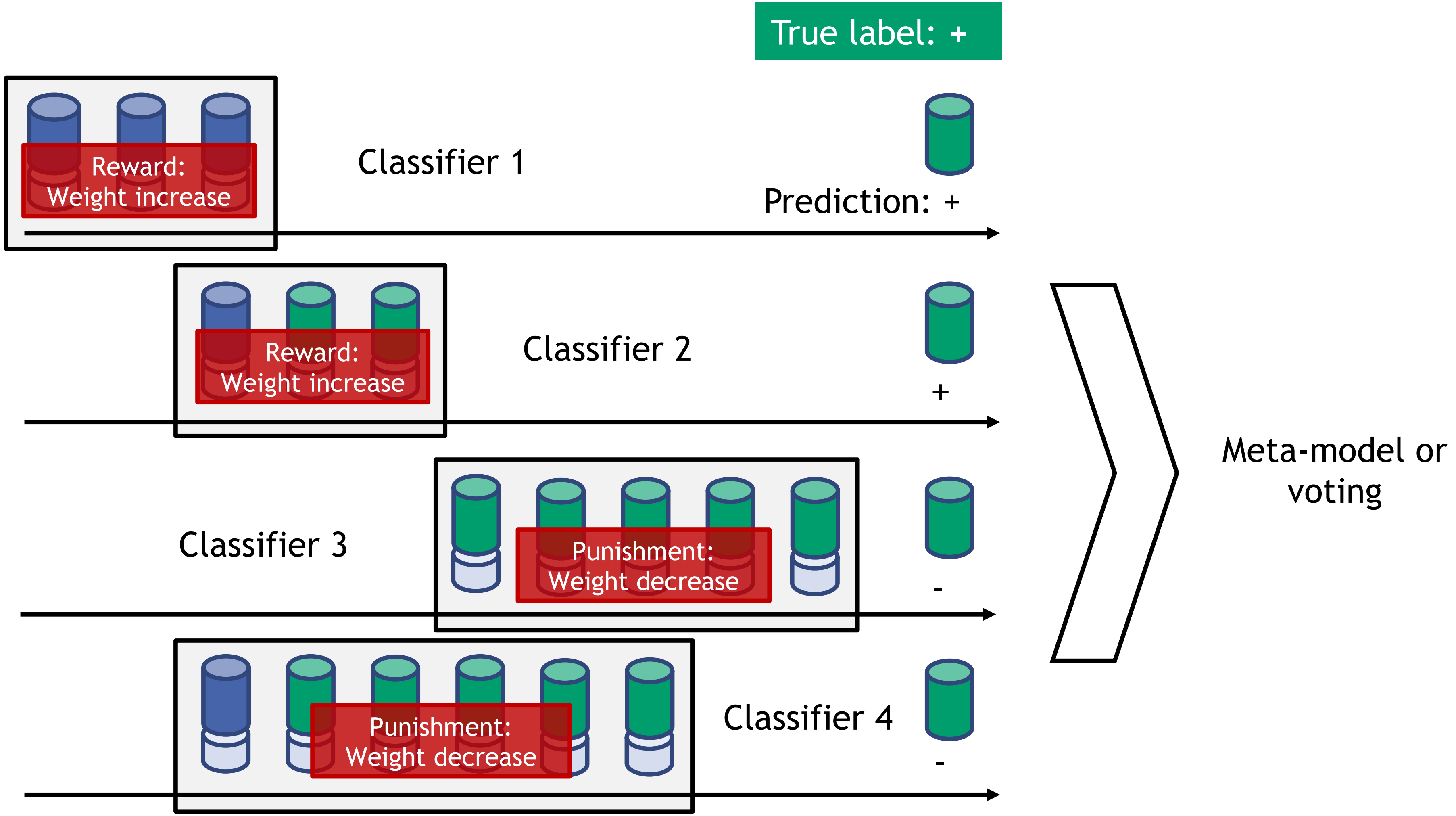



Source: Zliobaite (2010), Learning under Concept Drift

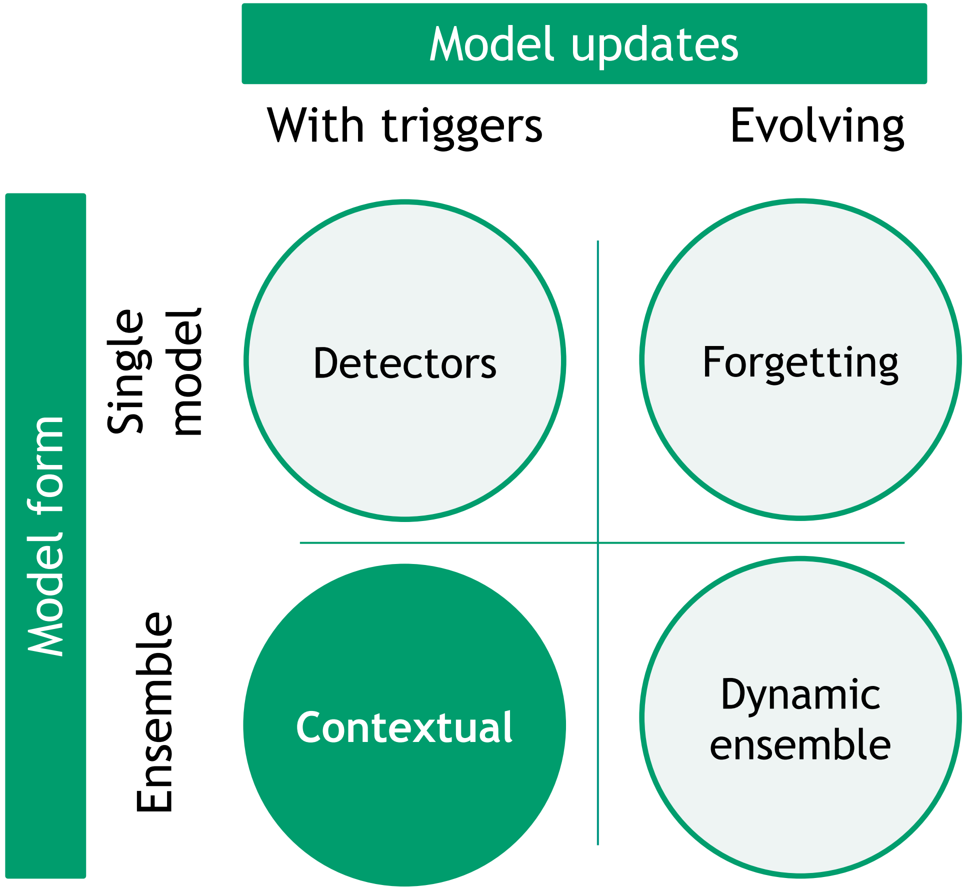
# Theoretical foundations of concept drift & adaption mechanism
Dynamic ensembles consider several ML models in parallel



# Theoretical foundations of concept drift & adaption mechanism
## Dynamic ensembles consider several ML models in parallel



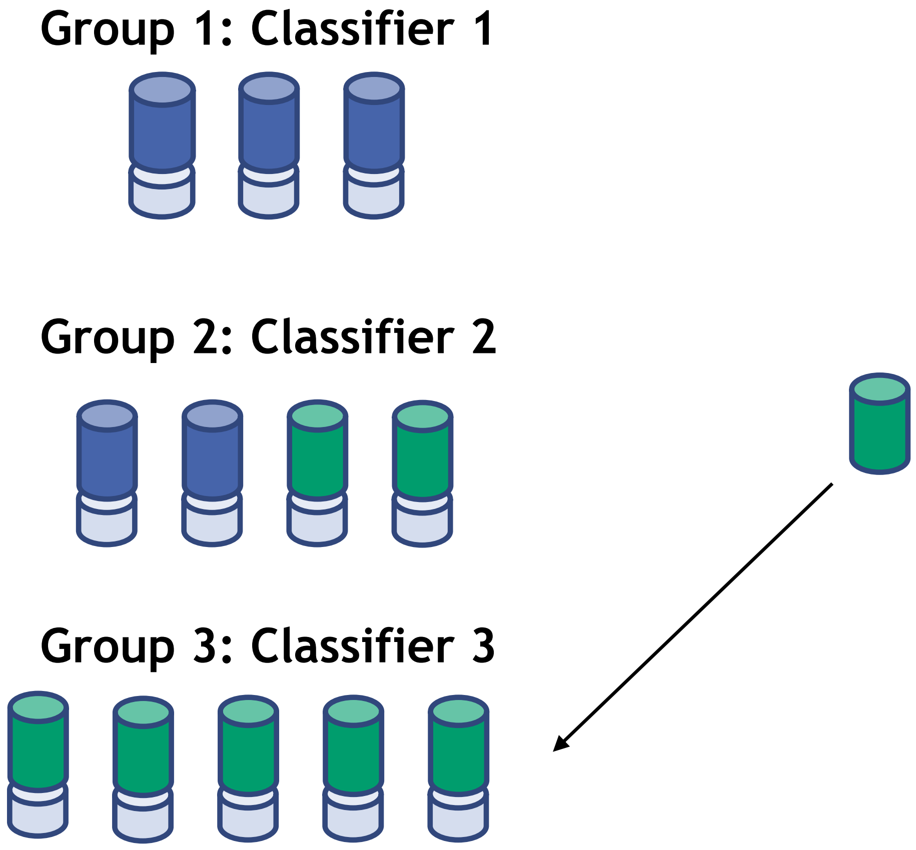
# Theoretical foundations of concept drift & adaption mechanism
Different adaptive learning strategies are available




Source: Zliobaite (2010), Learning under Concept Drift

# Theoretical foundations of concept drift & adaption mechanism

Contextual approaches start by identifying the group affiliation of a new data instance




## Train

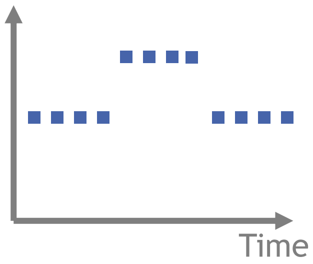
Partition the training data into several groups and build separate models for each group

## Predict

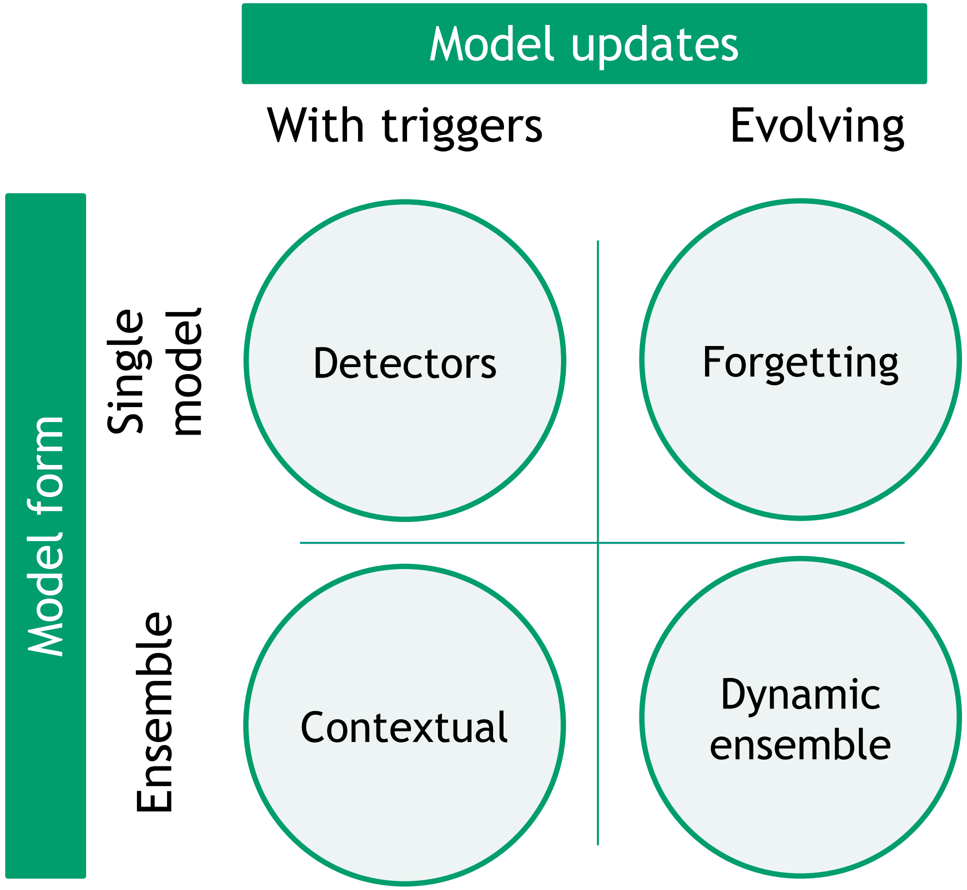
New instance is assigned to one group and corresponding model is applied

## Use case

Especially suited for reoccurring concepts (e.g. seasonal pattern in sales forecasts)



# Theoretical foundations of concept drift & adaption mechanism
Different adaptive learning strategies are available




Source: Zliobaite (2010), Learning under Concept Drift

1 Introduction

2 Theoretical foundations of concept drift & adaption mechanism

3 Two examples for concept drift detection algorithms

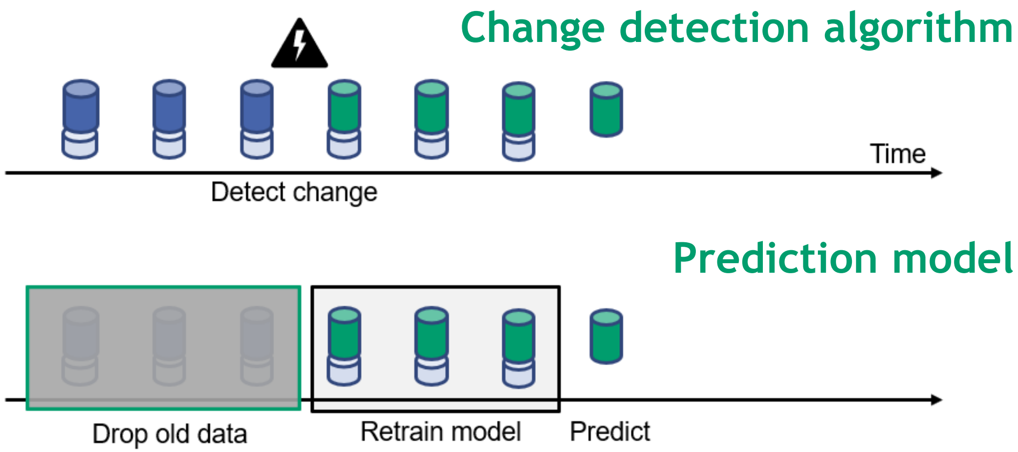
4 Real-world examples for concept drift handling

# Two examples for concept drift detection algorithms
## Wide variety of change detection / adaptation approaches available

## Informed methods
Next!

(Detector-based methods)

Explicit change detection




Change detection usually works on prediction error:

- Sequential Analysis: *Page-Hinkley*
- Monitoring two distributions: *ADWIN*

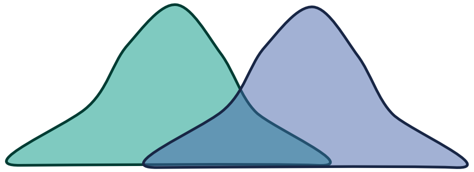
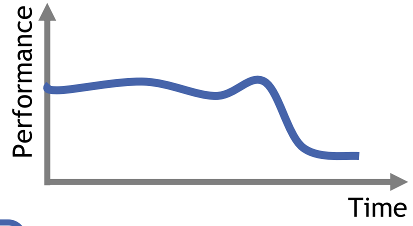
## Blind methods

(Forgetting-based methods)

Model which adapt incrementally or are frequently retrained


- Incremental weight decrease for older observations
- *CVFDT (Concept-adapting very fast decision trees)*

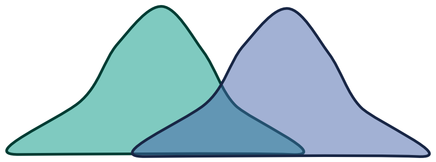
# Two examples for concept drift detection algorithms
## Informed methods can be distribution- or error rate-based methods

| | Data distribution-based drift detection | Error rate-based drift detection |
|---|---|---|
| Goal | Quantify **dissimilarity between distribution** of old data and new data with distance function | Detect changes by considering the error rate of the **underlying machine learning model** |
| Concept |  |  |
| Example Algorithms | Possible distances: Kullback-Leibler divergence or Kolmogorow-Smirnow | Next! ADWIN, Page-Hinkley-Test |
| In practice | Consistency checks for input data (mean, variance) Next! | Often manual supervision of prediction quality and impact on KPIs |
| Disadvantage | Computationally intensive | Requires true labels for drift detection |

# Two examples for concept drift detection algorithms
# **Cost functions** evaluate jumps in distribution characteristics

## Data distribution-based drift detection

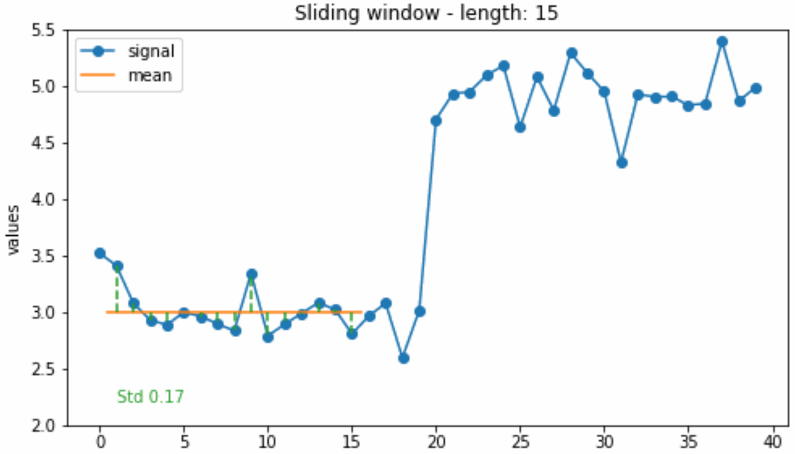
Concept



### Cost functions – a simple example

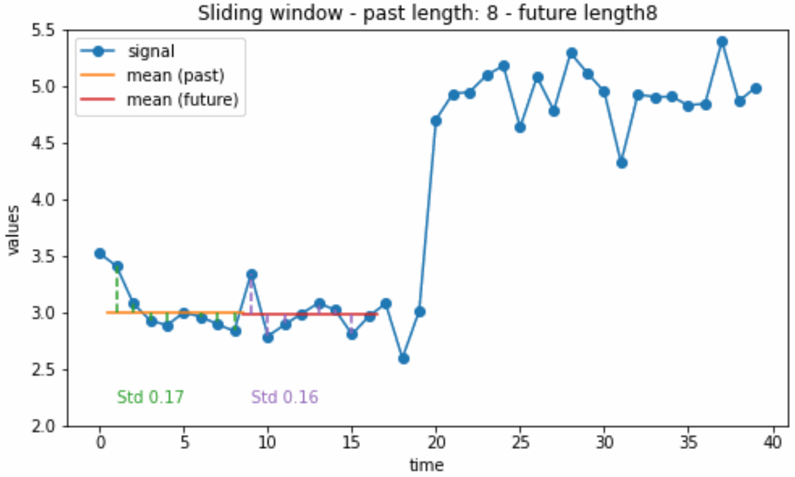
- Change point detection uses a **sliding window** with a cost function to identify changes in the signal
- Standard deviation can **detect changes in mean**, rising when the **signal jumps**.
- Change points are detected via **(a)** comparison to a fixed threshold or **(b)** comparing a second sliding windows

**(a)** Change points are marked if costs **exceed a threshold** in std.




[1]

**(b)** change point can be detected by **comparing the costs** of these two windows




[1]

Source: Gama et al. (2014), A Survey on Concept Drift Adaptation
[1] https://www.iese.fraunhofer.de/blog/change-point-detection/

# Two examples for concept drift detection algorithms
# Adaptive Sliding Window (ADWIN) compares sliding windows

## Adaptive Sliding Window (ADWIN) Algorithm

ADWIN: ADAPTIVE WINDOWING ALGORITHM

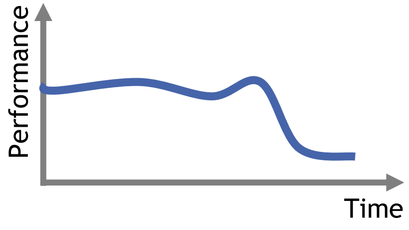
Initialize Window $W$
**for** each $t > 0$
  **do** $W \leftarrow W \cup \{x_t\}$ (i.e., add $x_t$ to the head of $W$)
    **repeat** Drop elements from the tail of $W$
      **until** $|\hat{\mu}_{W_0} - \hat{\mu}_{W_1}| \geq \epsilon_{cut}$ holds
        for every split of $W$ into $W = W_0 \cdot W_1$
    output $\hat{\mu}_W$

Figure 1: Algorithm ADWIN.

Next!

- Uses a detection window $\boldsymbol{W}$ which iteratively adapts
- Whenever two large (sub)windows of $\boldsymbol{W}$ exhibit distinct enough means, algorithm drops older elements
- Threshold $\boldsymbol{\varepsilon_{cut}}$ defined by Hoeffding bound

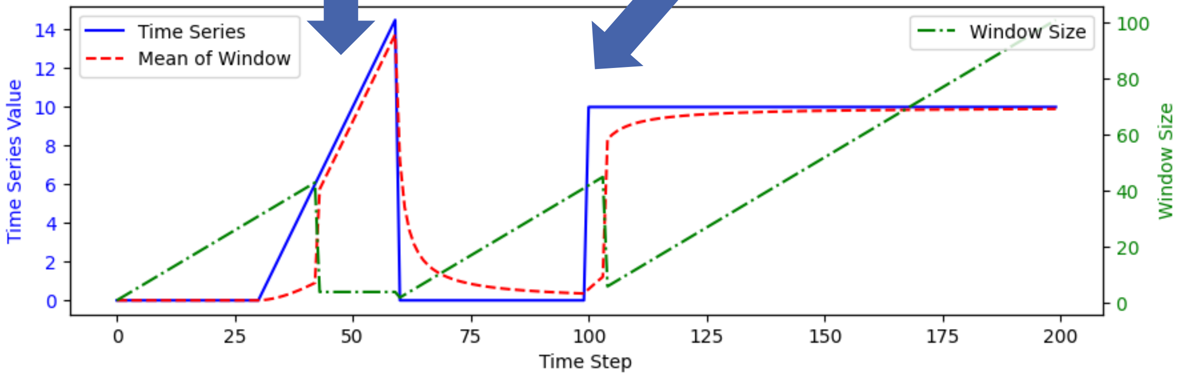
## Error rate-based drift detection




Concept

Incremental change

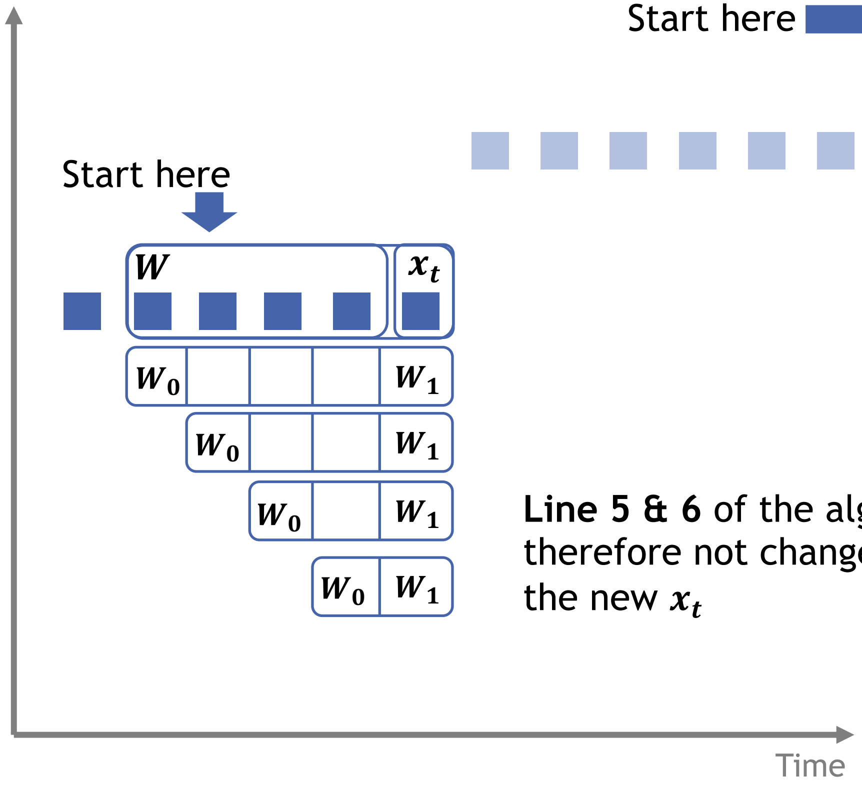
Sudden change



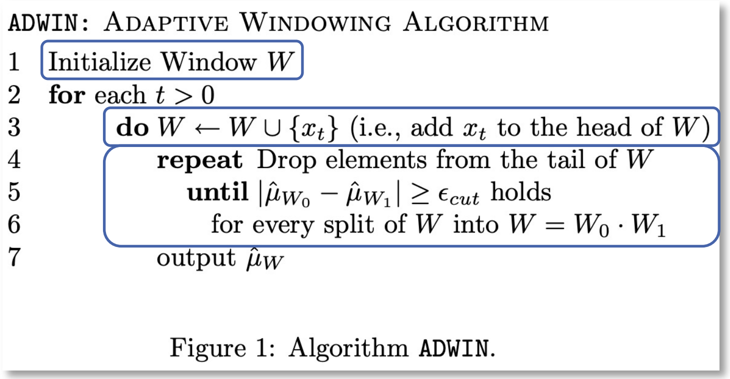

Bifet, Albert, and Ricard Gavalda. "Learning from time-changing data with adaptive windowing." SIAM 2007.

# Two examples for concept drift detection algorithms
# **Adaptive Sliding Window (ADWIN)** compares sliding windows

## Sudden changes




Start here



ADWIN: ADAPTIVE WINDOWING ALGORITHM

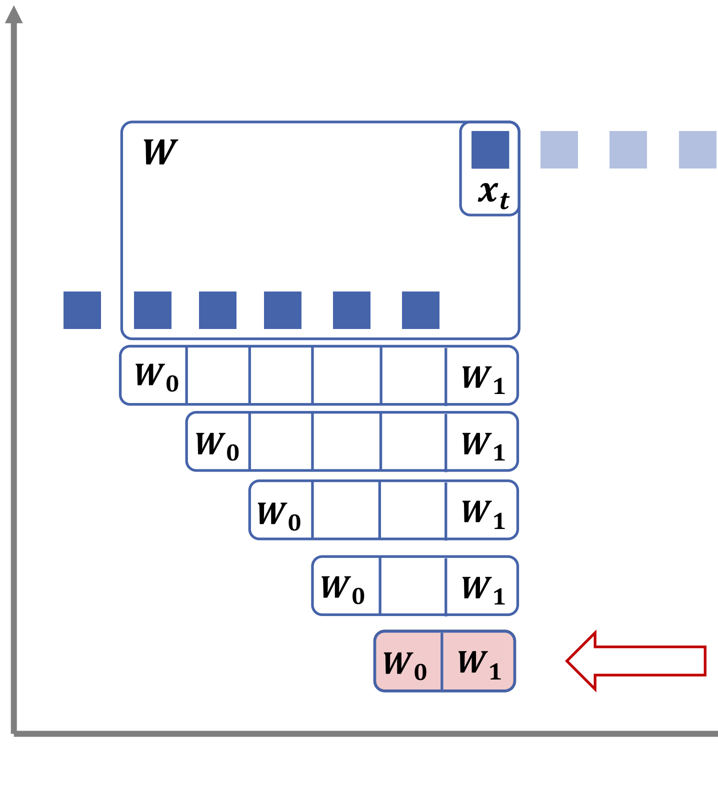
1 Initialize Window $W$
2 **for** each $t > 0$
3 **do** $W \leftarrow W \cup \{x_t\}$ (i.e., add $x_t$ to the head of $W$)
4 **repeat** Drop elements from the tail of $W$
5 **until** $|\hat{\mu}_{W_0} - \hat{\mu}_{W_1}| \geq \epsilon_{cut}$ holds
6 for every split of $W$ into $W = W_0 \cdot W_1$
7 output $\hat{\mu}_W$

Figure 1: Algorithm ADWIN.

**Line 5 & 6** of the algorithm never applied, therefore not change is detected and W grows by the new $x_t$

# Two examples for concept drift detection algorithms
# **Adaptive Sliding Window (ADWIN)** compares sliding windows

## Sudden changes



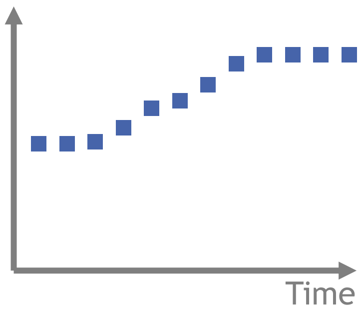

ADWIN: Adaptive Windowing Algorithm

```
1  Initialize Window W
2  for each t > 0
3      do W ← W ∪ {x_t} (i.e., add x_t to the head of W)
4         repeat Drop elements from the tail of W
5            until |μ̂_{W_0} − μ̂_{W_1}| ≥ ε_cut holds
6              for every split of W into W = W_0 · W_1
7         output μ̂_W
```

Figure 1: Algorithm ADWIN.

That's where **line 5 & 6** of the algorithm apply in the case of **sudden change**

## Slow changes




How does ADWIN behave in the case of slow change?

## Real-world examples for concept drift handling
Various threats exists for validity of machine learning microservices

| Trigger | Development vs Deployment | Environment of the service | Application of the service |
|---|---|---|---|
| Explanation | • During development, models are often trained with well-defined and curated data sets<br>• In deployment, data input shows great variance<br>• **Example:** Outliers in input data during deployment | • Change in the environment of the service with corresponding data changes<br>• Influence on prediction quality<br>• **Example:** Sensor of a production machine wears out | • Application of service affects prediction quality<br>• **Examples:**<br>  • Continuous predictive policing service<br>  • Changing user behavior due to machine learning services (different click stream) |

# Real-world examples for concept drift handling
Studies on concept drift have been performed in various domains

| | Monitoring & control | Information management | Analytics & diagnostics |
|---|---|---|---|
| Drift type | • Real time monitoring<br>• Typically fast concept drift | • Time-stamped data<br>• Sudden or gradual concept drift | • Time-stamped data<br>• Slower concept drift (e.g. population drift) |
| Use cases | • **Monitoring for Management:** Monitoring output quality in chemical production<br>• **Automated control:** Soccer playing robots<br>• **Anomaly detection:** Network intrusion detection | • **Personal assistance:** News categorization, spam filtering<br>• **Marketing:** customer segmentation for cars, recommender systems<br>• **Management:** Archiving of documents | • **Forecasting:** Macroeconomic forecasts<br>• **Medicine:** Antibiotic resistance<br>• **Security:** Biometric authentication |

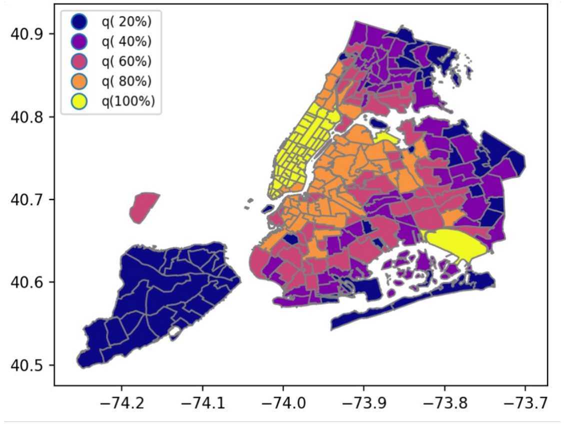
Source: Zliobaite et al. (2016), An Overview of Concept Drift Applications

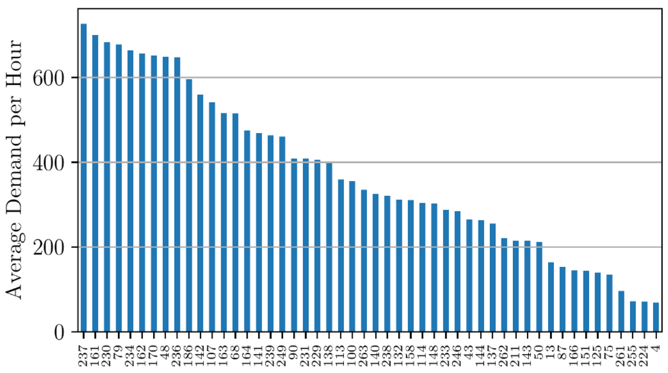
# Real-world examples for concept drift handling
## Exemplary drifts in NYC taxi demand

**Time series examples with sudden drift:**

- Temperature
- Stock prices
- Taxi demand
- Demand for groceries

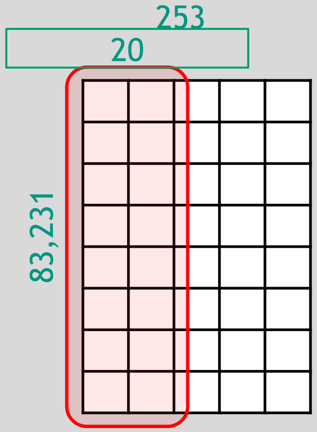
**New York City Taxi data set**




- ~1.4 billion rides (yellow cabs)
- 01/2009 - 06/2018
- Features: information regarding pickup and drop-off, fare and tip
- Various preprocessing steps

**Objective:** Predict hourly taxi demand per taxi zone



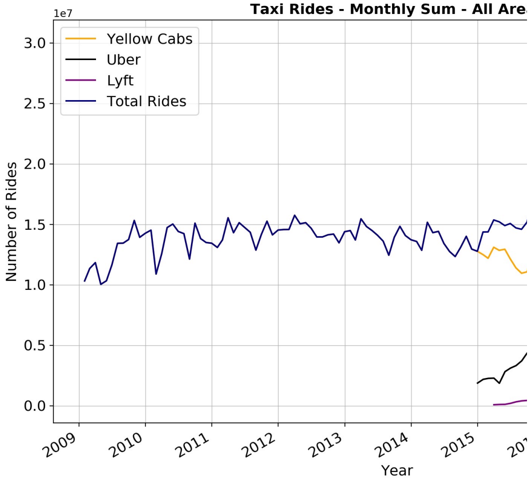

~ 21 million observations

Source: Baier, Hofmann, Kühl, Mohr, Satzger (2020), Handling Concept Drifts in Regression Problems – the Error Intersection Approach

# Real-world examples for concept drift handling
## Exemplary drifts in NYC taxi demand

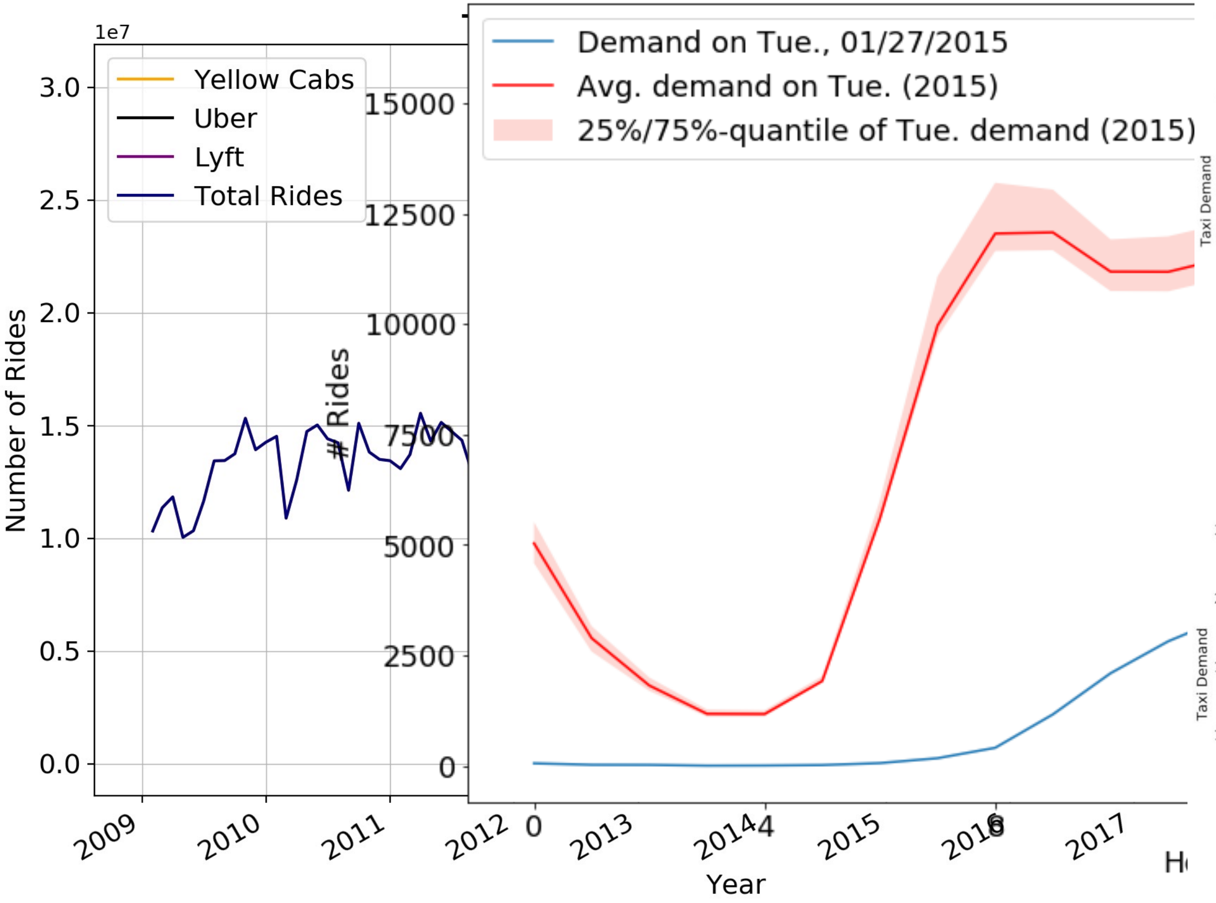
### Incremental drift

- Decreasing demand for yellow cabs over time
- New competitors

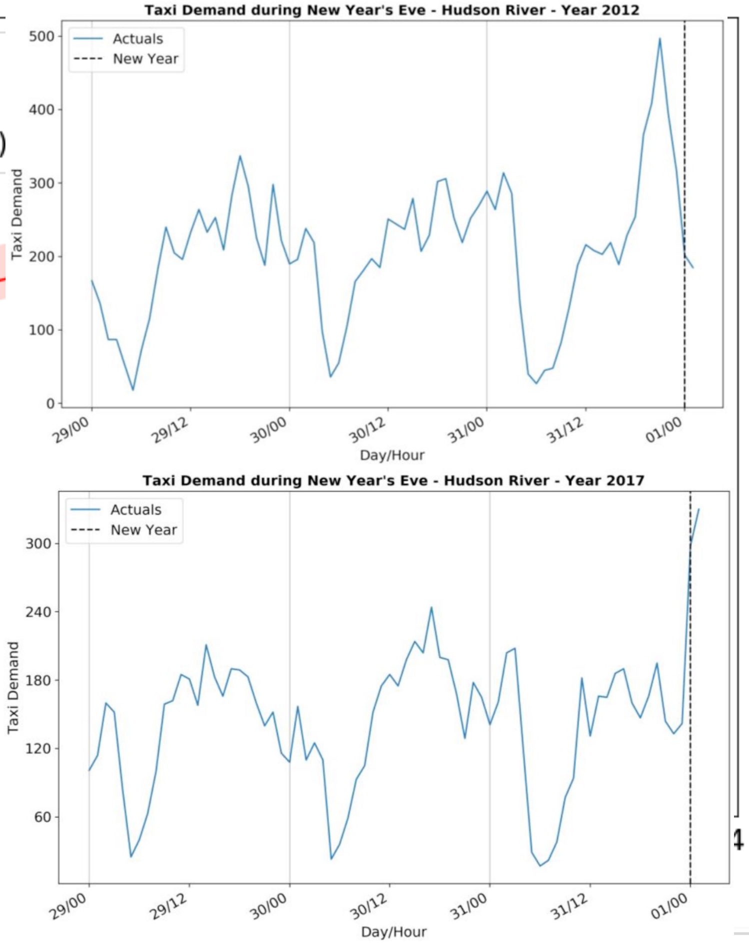



### Sudden drift

- Unusual demand pattern during




### Seasonal drift



# Real-world examples for concept drift handling
## Switch between models to handling concept drift

**Error Intersection Approach:** Switch to simple model in case of sudden drift and use complex model for ordinary days

Data stream (past taxi demand) → Simple model ($M_{simple}$) / Complex model ($M_{complex}$) → Switch between models

### Simple model ($M_{simple}$)

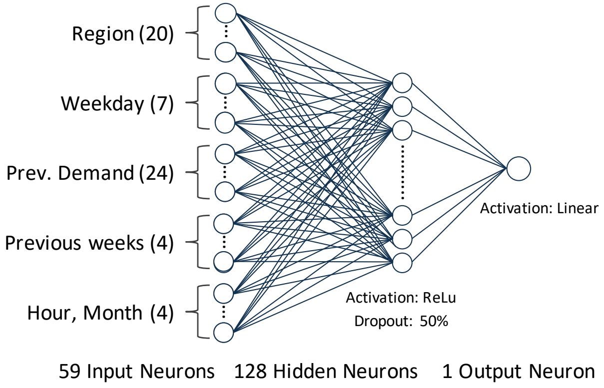
Simple model needs to quickly adapt to changing circumstances

*Random walk model:*

$$\hat{y}_{s,t,r} = y_{t-1,r}$$

*s: simple, r: region*

### Complex model ($M_{complex}$)




### Switching mechanism

$$Err_{M,t,r} = \left|\hat{y}_{M,t,r} - y_{t,r}\right|$$

$$EWMA_{M,t,r} = \begin{cases} Err_{M,1,r}, & t = 1 \\ \alpha * Err_{M,t,r} + (1-\alpha) * EWMA_{M,t-1,r}, & t > 1 \end{cases}$$

$$\alpha = \frac{2}{N+1}, \qquad N = 6$$

$$E_{M,t} = \frac{1}{R}\sum_{r=1}^{R} EWMA_{M,t,r}$$

$$M \in \{s(imple), c(complex)\}$$

$$choose \ \underset{s,c}{\operatorname{argmin}}(E_{s,t-1}, E_{c,t-1})$$

*r: region, N: parameter*

Source: Baier, Hofmann, Kühl, Mohr, Satzger (2020), Handling Concept Drifts in Regression Problems – the Error Intersection Approach

# Real-world examples for concept drift handling
## EIA performs well with unusual demand patterns

| Date | Abs. RMSE improvement | Rel. RMSE improvement | # Predictions by $M_{simple}$ | Probable Drift Cause |
|---|---|---|---|---|
| 2012-07-04 | 5.07 | 3.96% | 14/24 | 4th of July |
| 2012-10-29 | 24.41 | 31.02% | 22/24 | Hurricane Sandy |
| **2012-12-25** | **9.22** | **12.61%** | **17/24** | **Christmas Day** |
| 2013-08-01 | 9.35 | 8.61% | 10/24 | ? (unknown) |
| … | … | … | … | … |
| **2016-01-23** | **66.61** | **61.14%** | **16/24** | **Blizzard** |
| 2017-06-25 | 5.48 | 10.12% | 10/24 | ? (unknown) |
| 2018-03-21 | 15.21 | 17.43% | 14/24 | Cyclone (Nor'easter) |

*Table: Excerpt of days with frequent use of $M_{simple}$*

- We also find evidence for days where we cannot explain the drift cause
- Influencing factors are easy to identify from hindsight and might not be obvious in real time

## Summary

### Constant Monitoring

A **constant monitoring of deployed machine** learning solutions is crucial for ensuring their proper functionality / validity over time

### Application domain

Concept drift is a phenomenon which can be observed in nearly **all application domains**

### Implementation

The **change detection and adaptation options are manifold** and different choices must be made during the implementation

### Use-case specific

Real-world solutions for machine learning services usually need to be **tailored to the specific use case**