Introduction to Learning from Streaming Data

KEEPER Workshop Tutorial 2024

Nuwan Gunasekara*[1], Sepideh Pashami [1],
https://nuwangunasekara.github.io/KEEPER2025/

* Corresponding author: heitor.gomes@vuw.ac.nz

https://capymoa.org/

[1] Center for Applied Intelligent Systems Research, Halmstad University, Sweden

Classification algorithms

Random Forest (RF)

Main components:

1. Bagging
2. Random subset of features

Breiman, L. (2001). Random forests. *Machine learning.*

Bagging

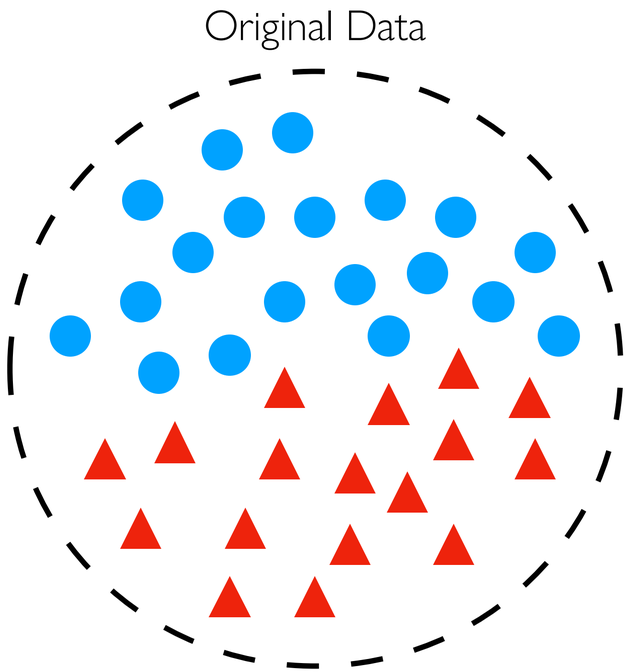
Bootstrap **Agg**regat**ing**

Bagging trains each model of the ensemble with a bootstrap sample from the original dataset.

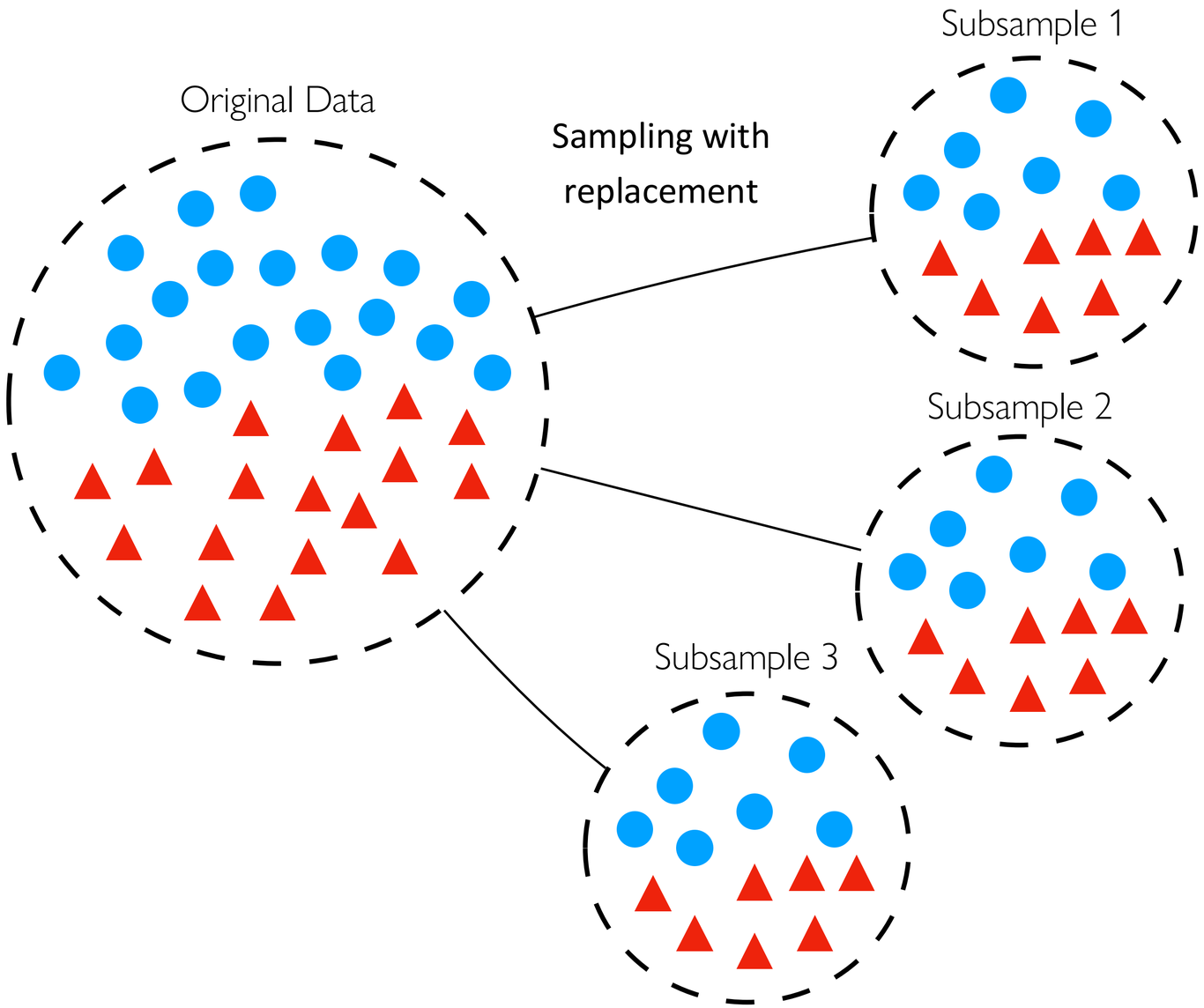
Every bootstrap contains each original sample **K** times, where **Pr(K=k)** follows a **binomial** distribution.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

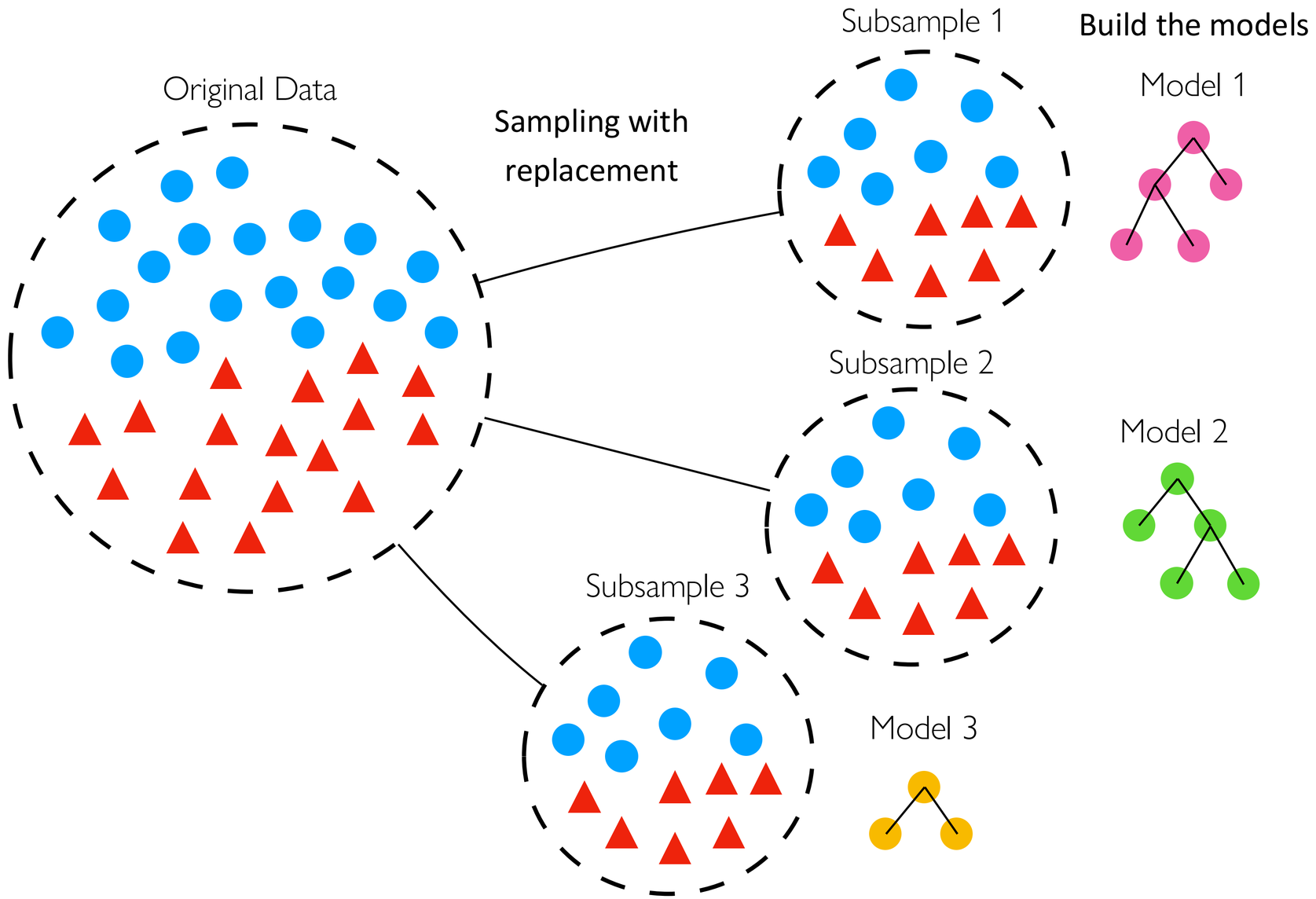
Bagging



Bagging


Original Data
Sampling with
replacement
Subsample 1
Subsample 2
Subsample 3

Bagging


Original Data
Sampling with replacement
Subsample 1
Subsample 2
Subsample 3
Build the models
Model 1
Model 2
Model 3

Bagging

On average **subsample** contains:

~64% of the **original dataset** instances

~37% **repeated** instances

~37% **not present*** instances

of the **original dataset**.

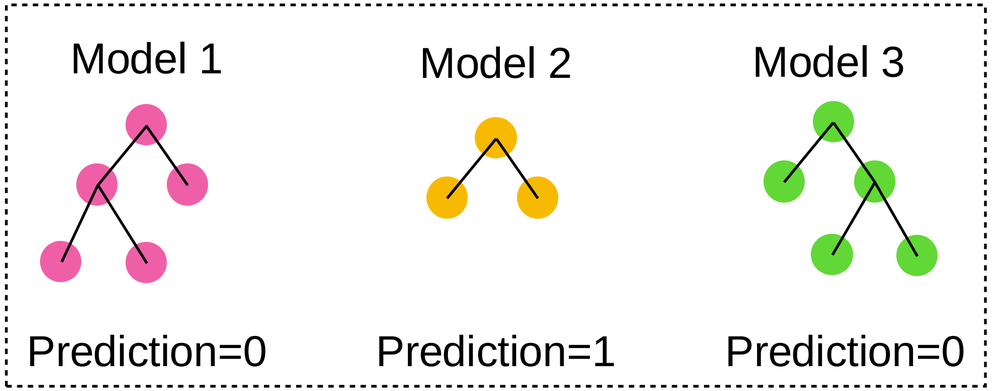
* Out-Of-Bag (OOB)

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

Bagging

The **predictions** of each learner are **aggregated** using **majority vote** to obtain the final prediction.

Prediction for a given instance X…



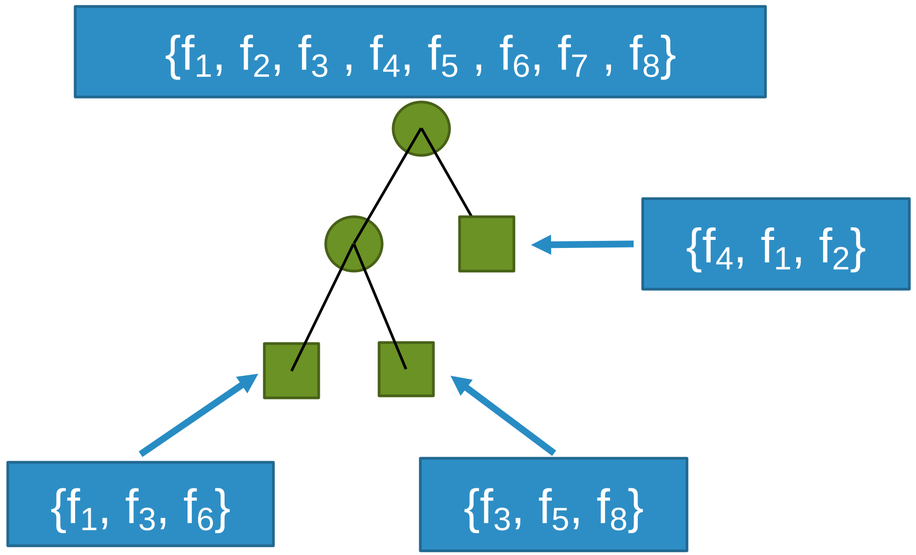
Ensemble

Prediction=0

Randomizing the feature set

Local randomization

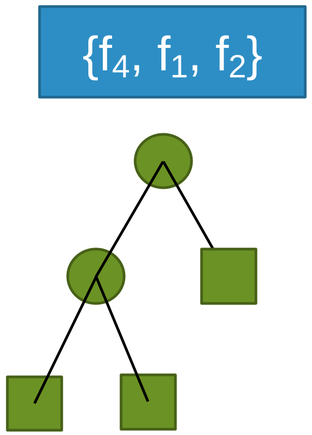
Random Forest



Global randomization

Random Subspaces

Random Patches



Adaptive Random Forest (ARF)

Streaming version of the original Random Forest by Breiman

Main differences:

Online base learner, **Online** bagging & **drift adaptation**

Overview:

1. Online bagging
2. Random subset of features
3. Drift detector for each tree

Breiman, L. (2001). Random forests. *Machine learning.*

Gomes, H. M., Bifet, A., Read, J., …, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning.*

Online base learner

- Uses a variation of the **Hoeffding Tree** (HT)
- HT builds a tree that **converges to** the **tree built by a batch learner** given **sufficiently large** data

Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. In *ACM SIGKDD*.

Online Bagging

- We cannot apply Bagging directly to data streams…
- Unfeasible to store all data before creating each bootstrap subsample

We need to build the subsamples online

N. Oza and S. Russel "Online bagging and boosting" Artificial Intelligence and Statistics, 2001

Online Bagging

- Given a dataset with **N** samples
- In Bagging, every bootstrap contains each original sample **K** times, where **Pr(K=k)** follows a binomial distribution
- Oza and Russel found out that for large **N**, the **binomial** distribution tends to a **Poisson(1)** distribution
- Online Bagging instead of sampling with replacement, gives each example a weight according to **Poisson(1)** distribution

N. Oza and S. Russel "Online bagging and boosting" Artificial Intelligence and Statistics, 2001

Subsamples

Batch bagging

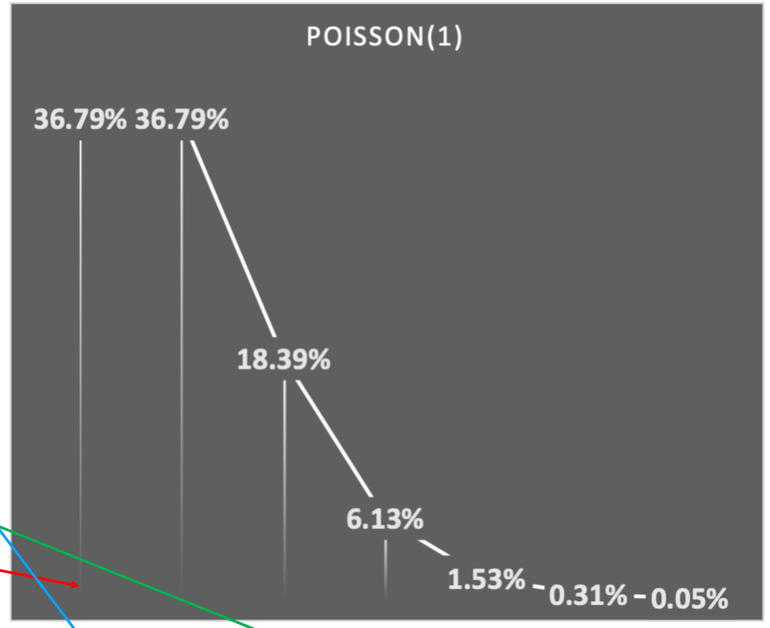
~64% from the original dataset

~37% are repeated

~37% are not present

Online bagging

Probability mass function



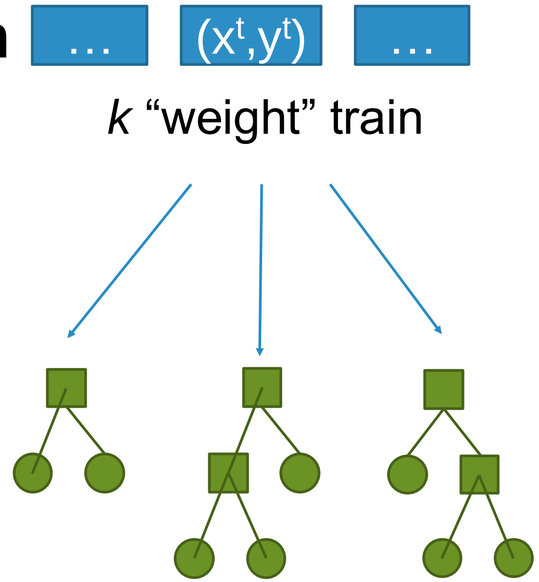
$k = 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad \ldots$

k is the number of occurrences

N. Oza and S. Russel "Online bagging and boosting" Artificial Intelligence and Statistics, 2001

Online Bagging

$k \leftarrow Poisson$ (λ=1)
if $k > 0$ **then**
  $l \leftarrow FindLeaf(t, x)$
  $UpdateLeafCounts(l, x, k)$



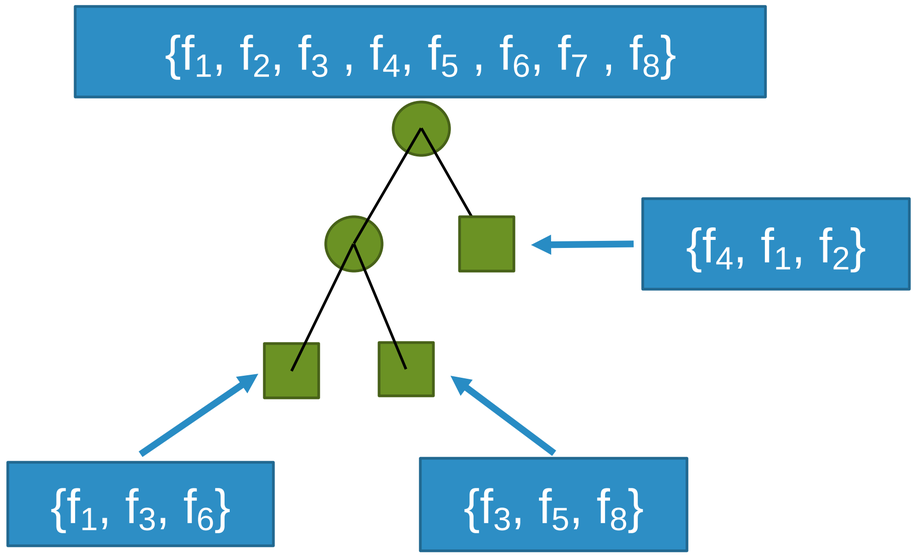
Practical effect: train learners multiple times or update counters multiple times

N. Oza and S. Russel “Online bagging and boosting” Artificial Intelligence and Statistics, 2001

Randomizing the feature set

Local randomization

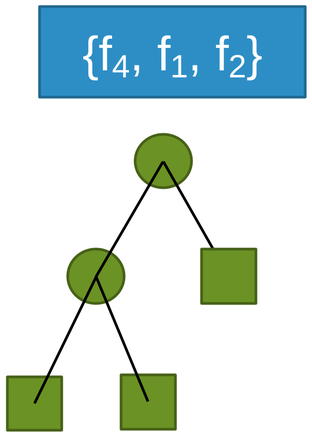
Random Forest



Global randomization

Random Subspaces

Random Patches



ARF: Detect and Adapt

- One **Warning** and one **Drift** detector per base model
- Relies on the **Adaptive WINdow** (ADWIN) algorithm for detection (other algorithms could be used)
- ***Background*** **learners** are started once a warning is detected, their subspace of features may not correspond to the subspace of features used by the *“foreground”* learner.
- Once a drift is detected, the ***background*** **learner replaces** the ***“foreground”*** **learner**.

Ensembles re-cap

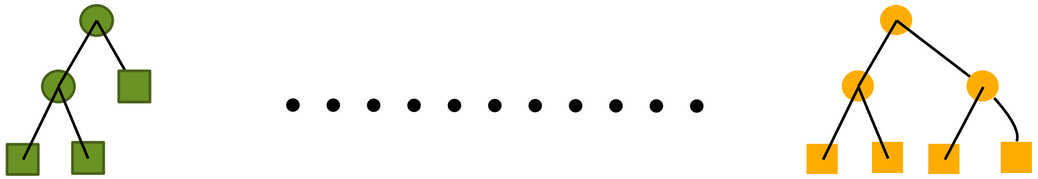
- Use **Poisson** distribution to derive weights for multiple training iterations
- More advance methods use **drift detectors** to adapt to changes
- Latest developments
  - Streaming Gradient Boosting [3]
  - Use **task parallelism** for **bagging ensembles** using **mini-batches** [1, 2]

[1] G. Cassales, H. M. Gomes, A. Bifet, B. Pfahringer and H. Senger, Improving the performance of bagging ensembles for data streams through mini-batching, Information Sciences,Volume 580, 2021, Pages 260-282, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2021.08.085.
[2] G. Cassales, H. M. Gomes, A. Bifet, B. Pfahringer and H. Senger, "Balancing Performance and Energy Consumption of Bagging Ensembles for the Classification of Data Streams in Edge Computing," in IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 3038-3054, Sept. 2023, doi: 10.1109/TNSM.2022.3226505
[3] Gunasekara, N., Pfahringer, B., Gomes, H., & Bifet, A. (2024). Gradient boosted trees for evolving data streams. Machine Learning, 113(5), 3325-3352.

Regression algorithms

Adaptive Random Forest Regression

- Similar to ARF for classification
- builds **regression trees**
- for **prediction**, uses **mean** of **predictions** (by each tree)



Gomes HM, Barddal JP, Ferreira LEB, Bifet A (2018) Adaptive random forests for data stream regression. ESANN

Practical examples

02_KEEPER2025_supervised.ipynb