Data Drift:

How to Uncover Changes in Your ML Model's Input Data

Emeli Dral
CTO Evidently AI

EVIDENTLY AI

About me

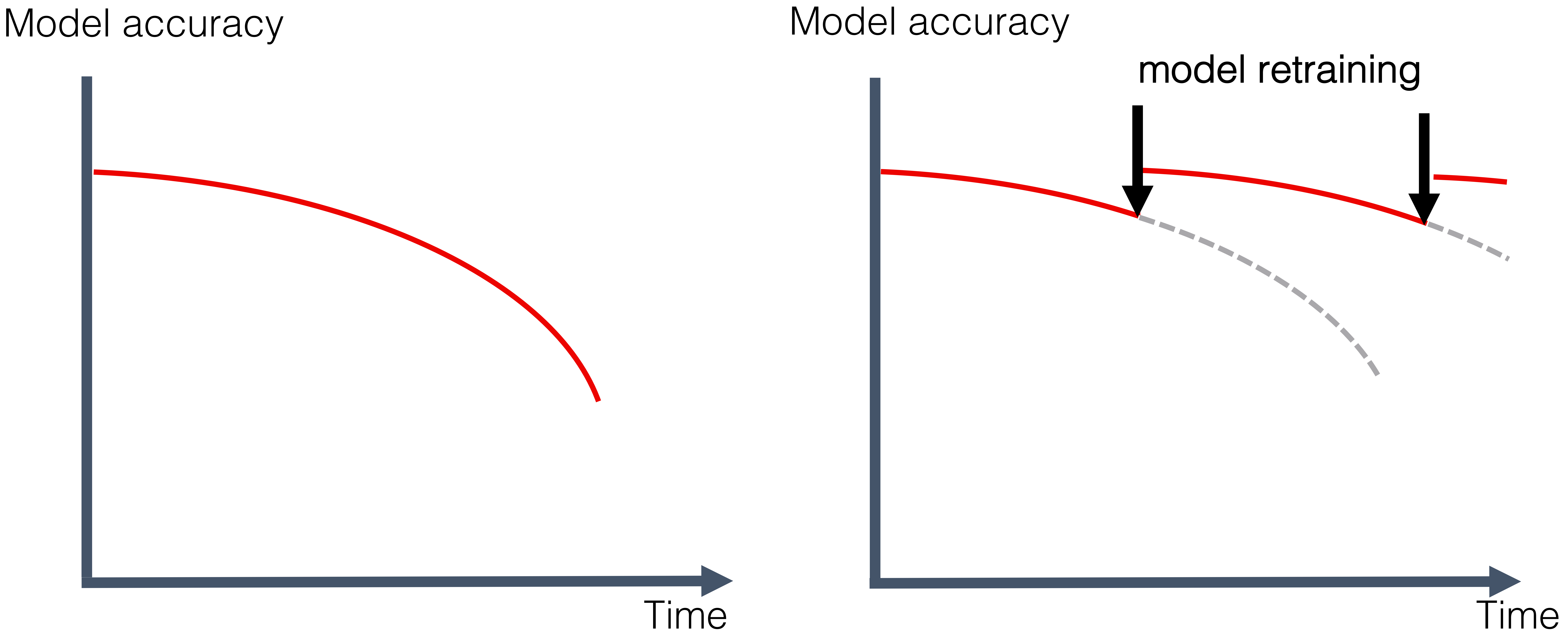
- **Co-founder & CTO** Evidently AI, open-source ML monitoring
- Ex **Chief Data Scientist** at Yandex Data Factory and Mechanica AI
- Co-founder of **Data Mining in Action**, largest offline data science course in Russia
- Co-author of two **Coursera** specializations in data science with > 100K students
- Lecturer at **Harbour.Space** University, GSOM MBA

50+ Industrial applications of machine learning

https://www.linkedin.com/in/emelidral

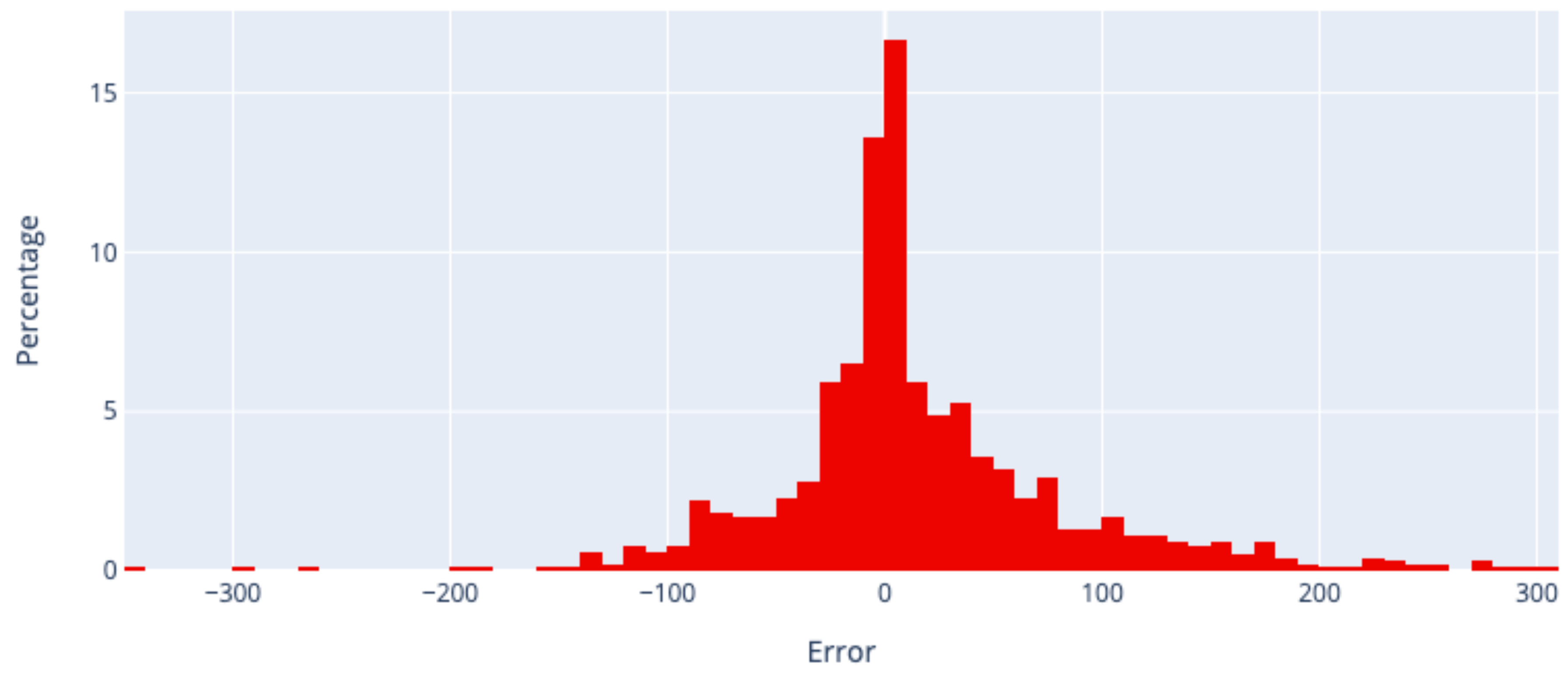
EVIDENTLY AI

ML models degrade and need monitoring



EVIDENTLY AI

Standard ML monitoring: measuring the performance



Model quality or error:
precision, recall, log-loss, MAE, etc…

Business or product metrics:
purchases, clicks, views, etc…

EVIDENTLY AI

Standard ML monitoring is not always enough!

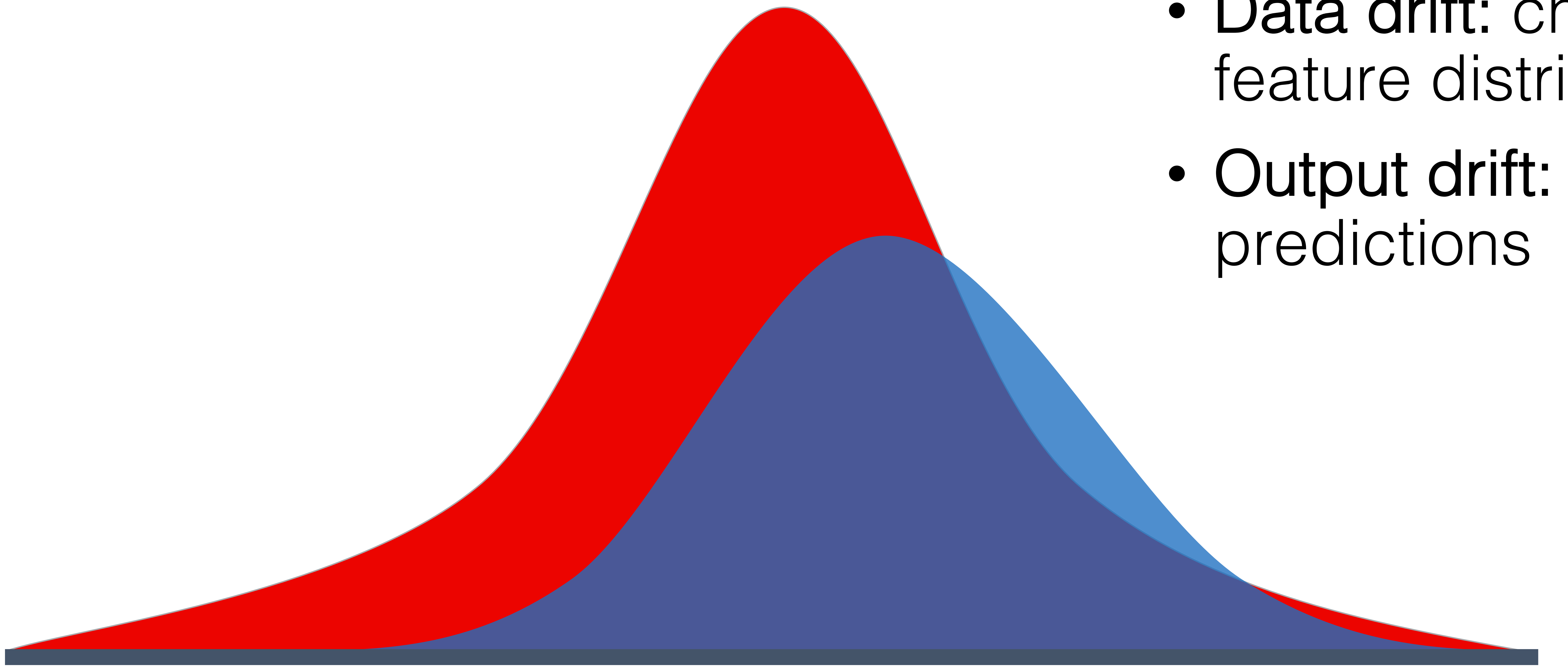
Feedback or ground truth is delayed

Many segments with different quality

The target function is volatile (quality varies)

Past performance does not guarantee future results

How to tackle it? Early monitoring



- **Data quality:** any issues with quality and integrity
- **Data drift:** changes in the input feature distributions
- **Output drift:** change in the models predictions

EVIDENTLY AI

Data Quality: with no expectations

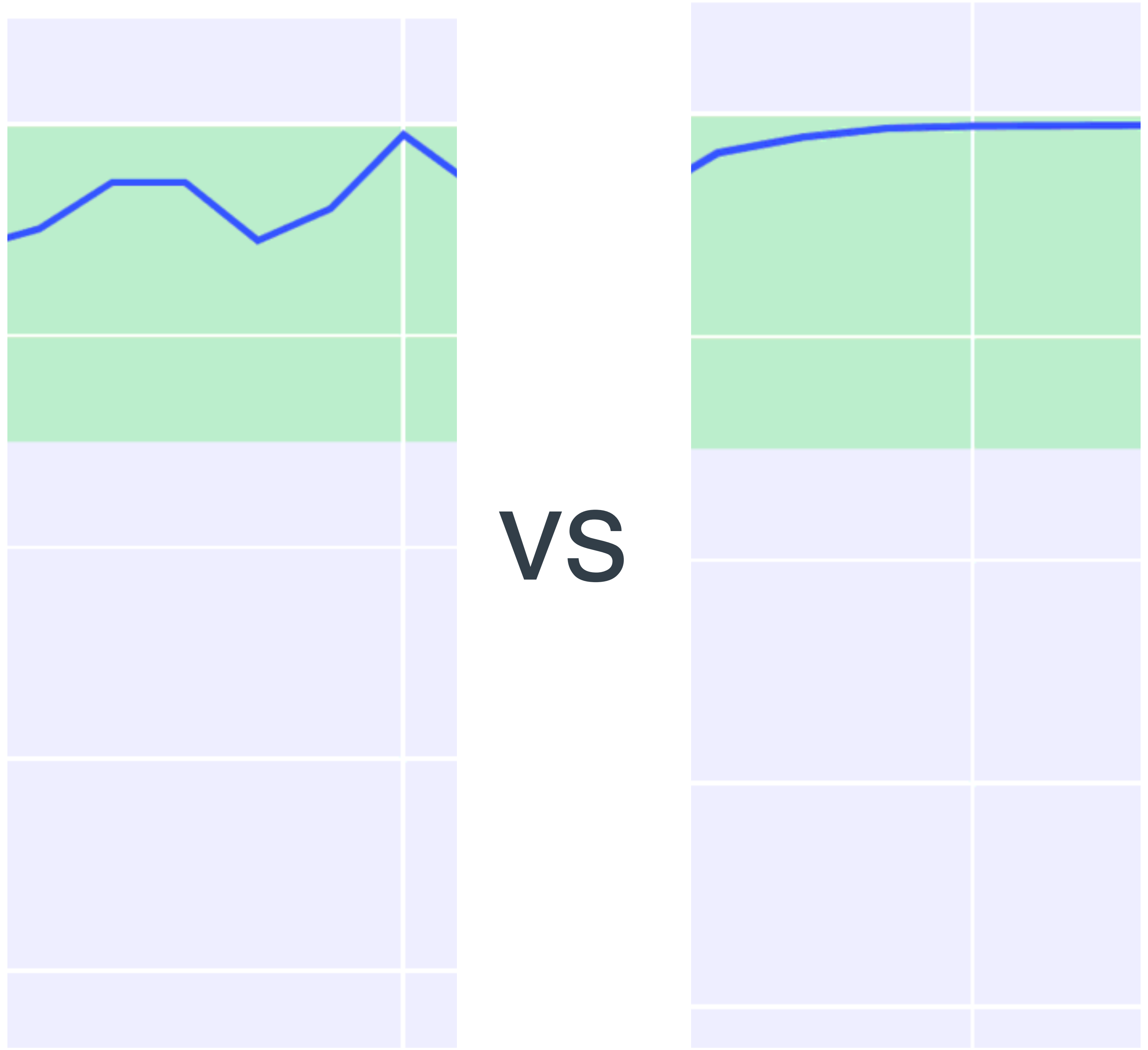
Missing cells	10 (6%)
Constant features	12 (7,5%)
Empty features	1 (0,6%)
Almost constant	12 (7,5%)
Almost empty	0 (0%)

Even if we do not have reference data:

- missing values (or almost missing!)
- constant features (or almost constant!)
- correlations (high correlation between feature and target, feature pairs, etc.)
- range violations (based on the feature context, e.g. negative age)

EVIDENTLY AI

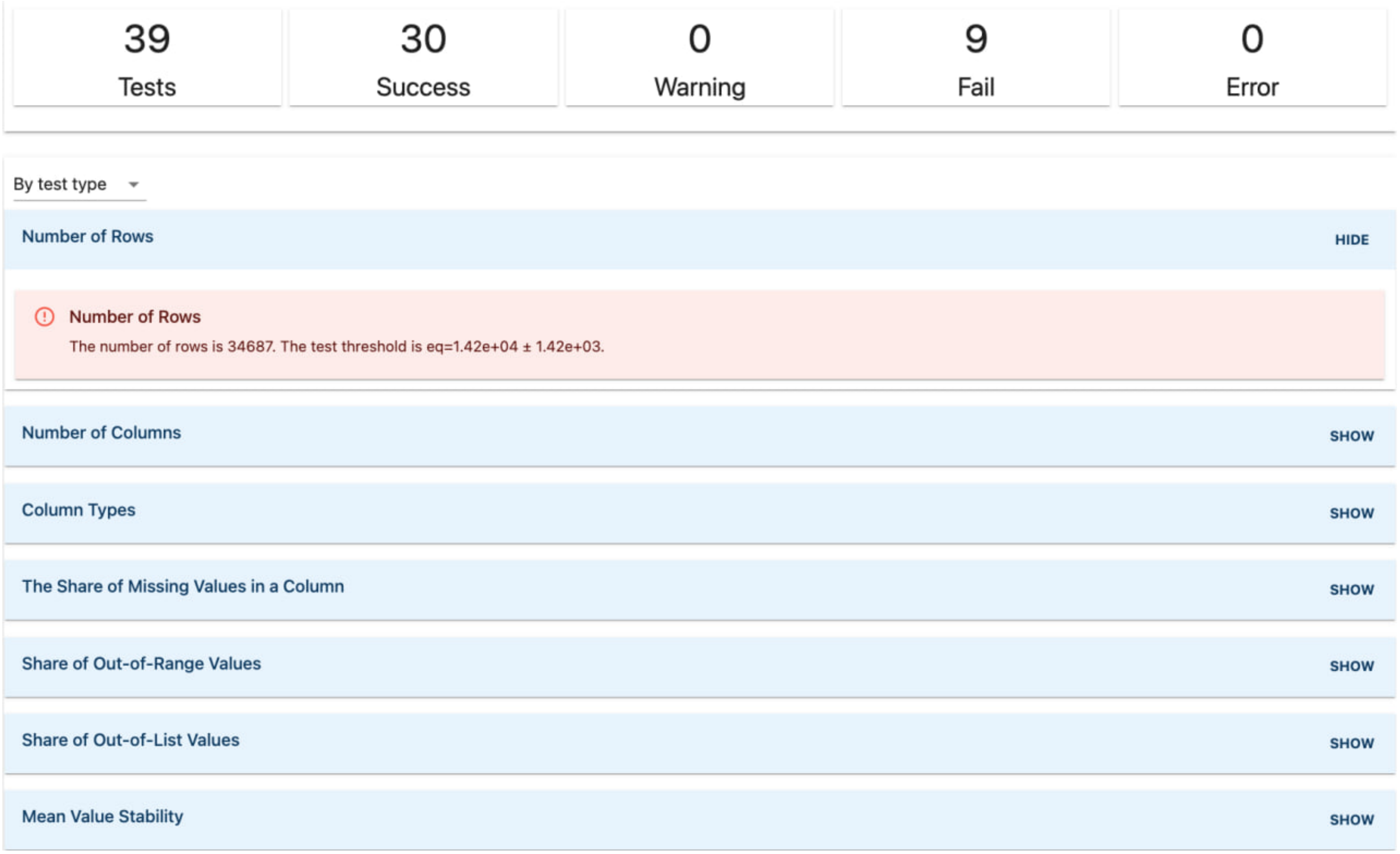
Data Quality: if we have expectations



Based on training data or past batch:

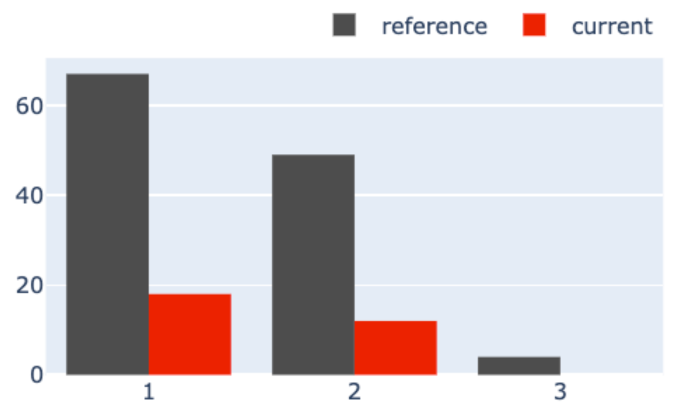
- expected data quality (e.g. 80% non-constant)
- data distribution type (e.g. normality)
- descriptive statistics: averages, median, quantiles, min-max for individual features
  - point estimation as a simple solution
  - statistical test to get confidence interval

Example from the Evidently OSS library



weathersit
cat

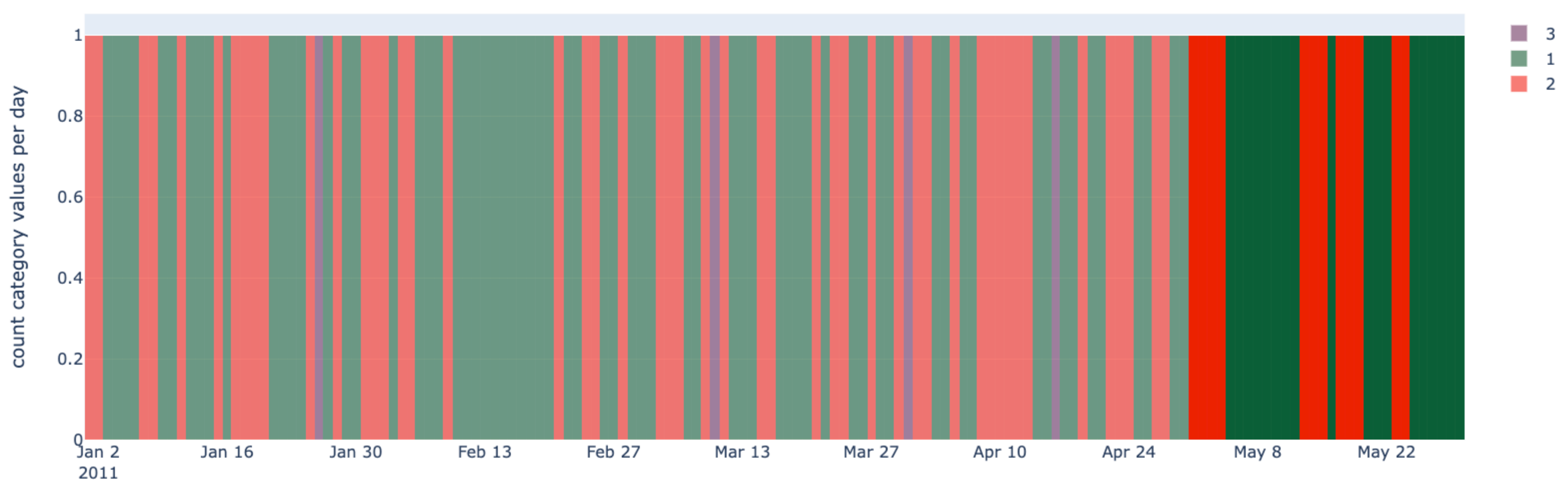
	reference	current
count	120	30
unique	3 (2.5%)	2 (6.67%)
most common	1 (55.83%)	1 (60.0%)
missing	0 (0.0%)	0 (0.0%)
new categories		0
missing categories		1



DETAILS

WEATHERSIT IN TIME

WEATHERSIT BY TARGET

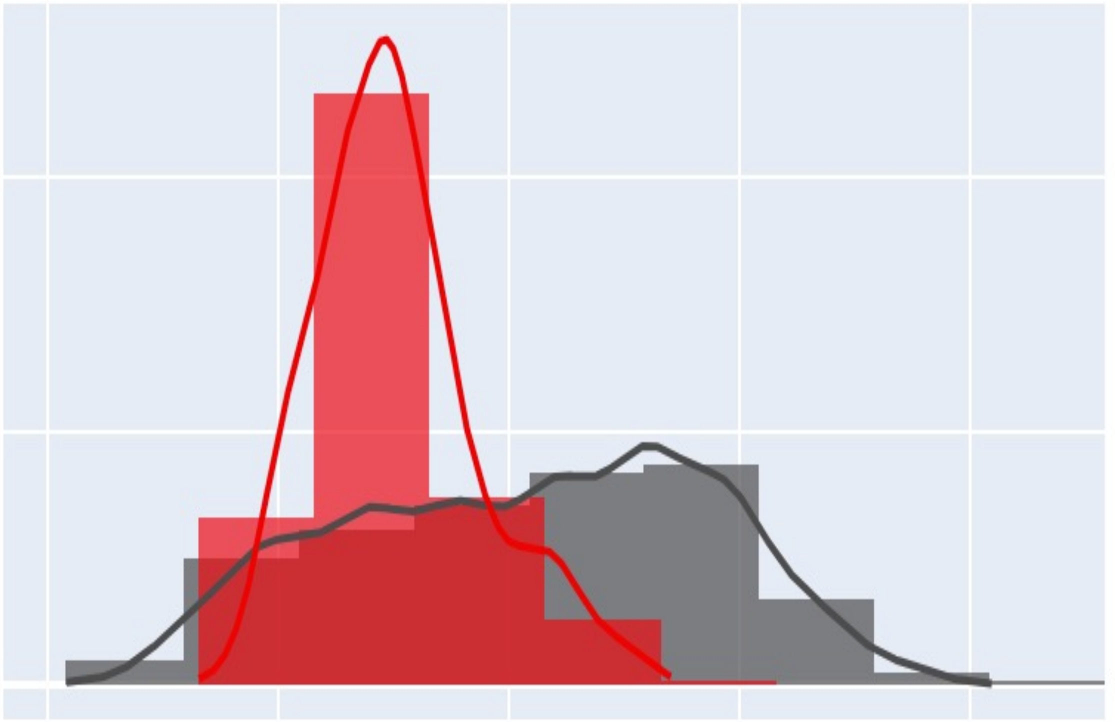


Test against expectations

Explore changes

EVIDENTLY AI

Data Drift: compare feature distributions



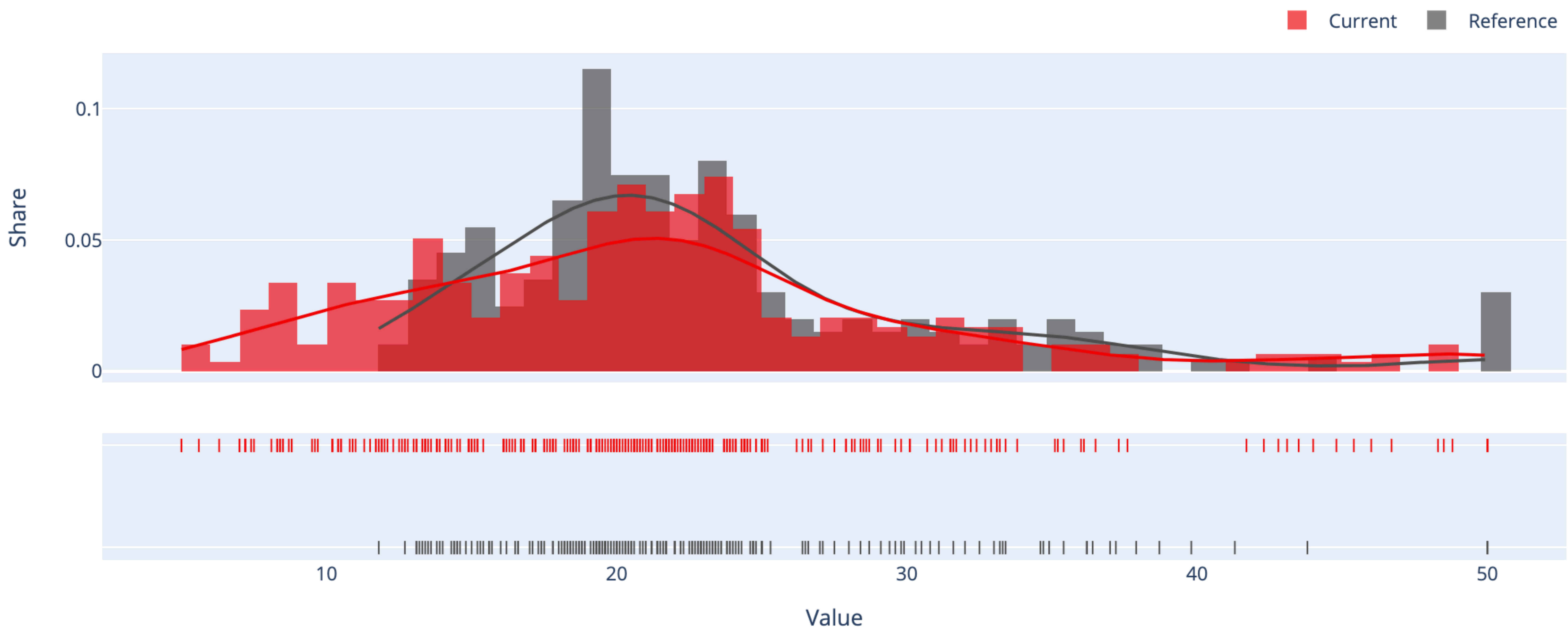
Data can drift without performance decay. But if the **key features** change, this can be an issue!

It is important to define:

- Optimal statistical tests, distance metrics or rules
- Reasonable confidence for statistical tests
- Alert conditions based on feature importance and the share of the drifting features

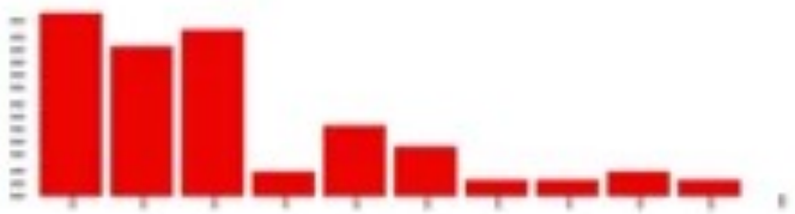
Prediction Drift: compare output distributions

Target Drift: detected, p_value=0.002266



EVIDENTLY AI

How do we define that data has drifted?

Reference Distribution	Production Distribution	Data drift
		DETECTED
		DETECTED
		DETECTED
		NOT DETECTED

Drift detection: parametric tests

EXAMPLES OF PARAMETRIC TESTS

One-sample:

- Z-test & T-test for mean (m = m0)
- One proportion Z-test (p = p0)

Two-samples:

- Two-proportions Z-test
- Two-samples Z-test T-test for means (normally distributed samples)

SOME CONSIDERATIONS:

- Require different tests for different features
- More sensitive to drift than non-parametric tests
- Hard to fine-tune if you have a lot of features

Makes sense if you have a small number of interpretable features and critical use cases (e.g., in healthcare).

Drift detection: non-parametric tests

EXAMPLES OF NON-PARAMETRIC TESTS

- Kolmogorov–Smirnov test:
  - the equality of distributions for continuous data
- K-sample Anderson–Darling tests:
  - can several collections of observations be modelled as coming from a single population?
- Pearson's chi-squared test:
  - the equality of distributions for categorical data
- Fisher's/Barnard's exact test for small samples

SOME CONSIDERATIONS:

- Can use heuristics to choose tests based on the feature type, e.g. numerical, categorical, binary
- Less sensitive to drift than parametric tests

Drift detection: distance-based approaches

EXAMPLES OF METHODS

- Wasserstein distance:
  - distance between probability distributions; shows how much effort it takes to turn one distribution into another
- Jensen-Shannon divergence:
  - distance between probability distributions; based on Kullback-Leibler divergence, but it is always finite and symmetric
- Population Stability Index (PSI):
  - reflects the relative size of a drift for numand cat features
- Domain classification:
  - applicable for different data types, including unstructured and multinomial data; ML-model based

SOME CONSIDERATIONS:

- Roughly any metric that shows difference/similarity between distributions can be used as a drift detection method
- Often it makes more sense to pick an interpretable metric rather than a statistical test

Example from the Evidently OSS library

For small datasets (<1000):

- **Numerical** features: two-sample Kolmogorov-Smirnov test.
- **Categorical** features, chi-squared test.
- **Binary categorical** features: the proportion difference test for independent samples based on Z-score.

EVIDENTLY AI

Example from the Evidently OSS library

CURRENT: CHARACTERISTIC WORDS
disappointed
money
poorly
returning
unraveled
cheap
completely

VS

REFERENCE: CHARACTERISTIC WORDS
feminine
dressed
elegant
hi
little
issues
pleated

For text data:

- Content drift: domain classifier for text features
- Text descriptors drift: distribution shift in text characteristics

EVIDENTLY AI

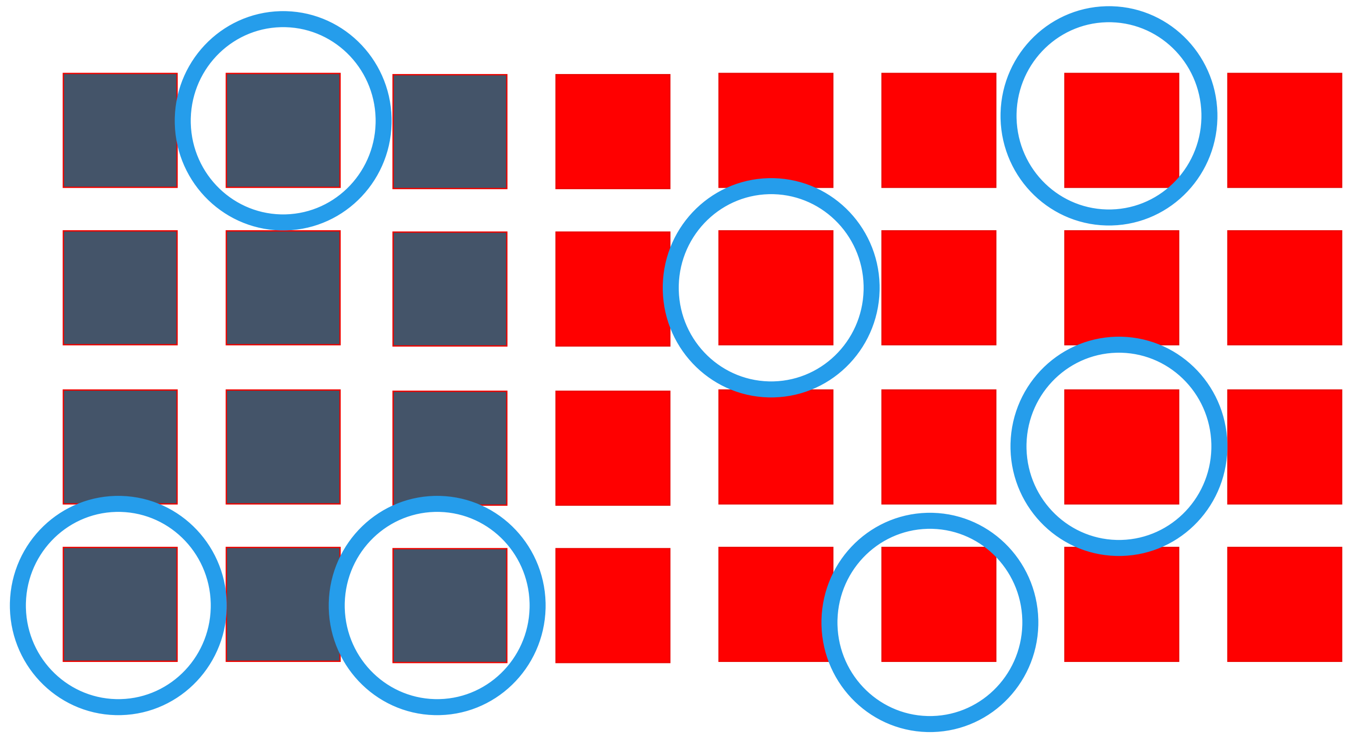
Drift detection: large datasets

If we have a lot of objects and/or a lot features, tests can be “too sensitive”.

NOTE: statistics was made to work with samples!

Practical solutions:

- Sampling (“pick” representative observations)
- Bucketing (“aggregate” all observations)



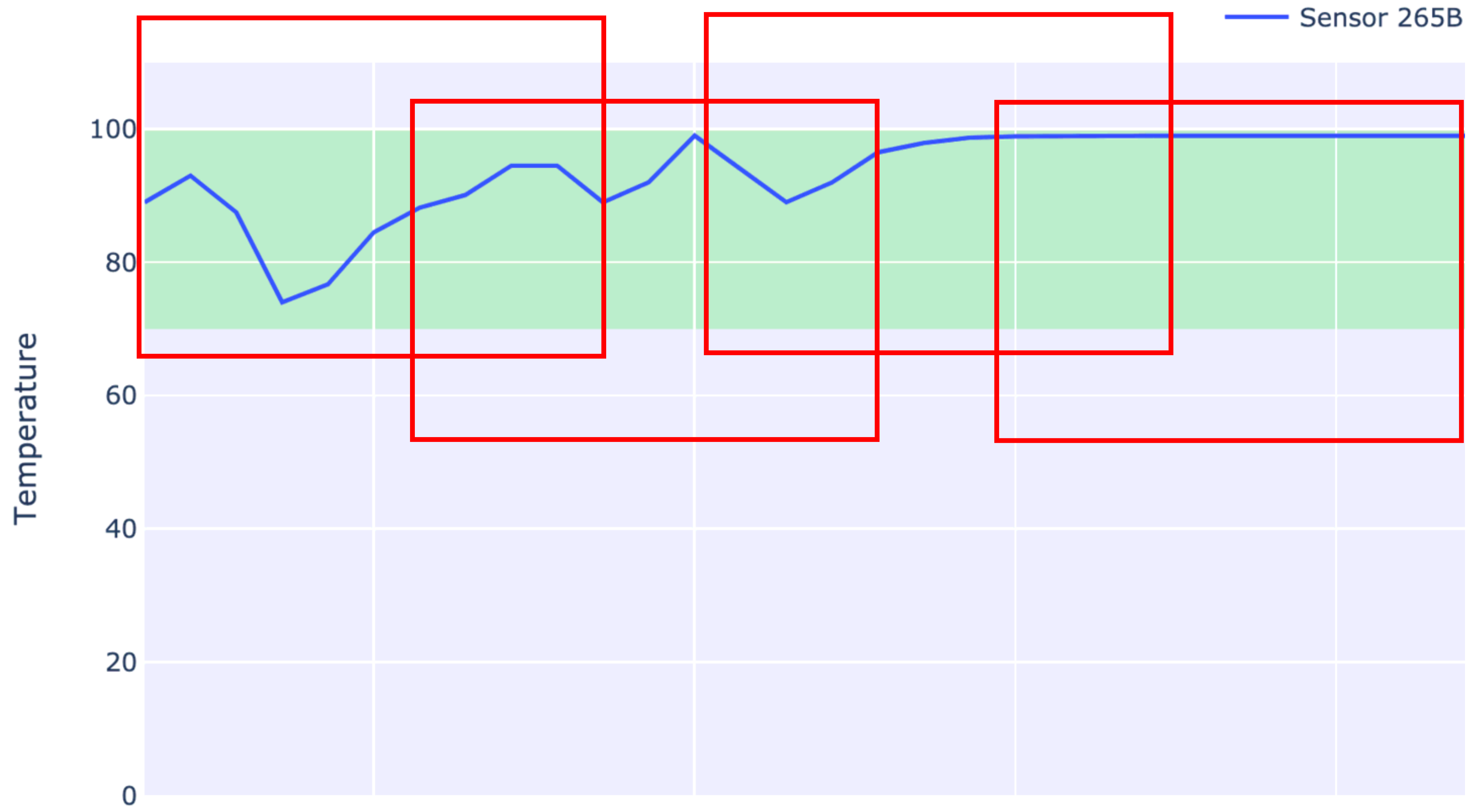
Drift detection: non-batch models

Descriptive statistics and quality:

- calculate metrics continuously or even incrementally

Statistical tests on a continuous data stream:

- pick a window function (e.g. moving windows with or without moving reference)



How to choose metrics and tests?

Option 1: go with defaults

- Pick a reasonable statistical test as a heuristic (e.g. K-S for numerical features)
- Start monitoring
- Adjust based on false alarms and your sensitivity

Option 2: experiment

Use past data to pick the most suitable test and drift conditions

Example of the experiment design:

- Pick a stable period when no drift
- Define candidate statistical tests
- Apply all tests and pick the one with the highest sensitivity (lowest p-value) that does not detect drift.
- If you have known past drift periods, make sure the test catches them.
- You can experiment with tests, confidence threshold, window size, sampling and bucketing parameters

Nuanced interpretations of drift

Data Drift: detected. Prediction drift: not detected.

☺

Positive interpretation:

- Important features did not change.
- Model is robust enough to survive drift.
- No need to intervene.

☹

Negative scenario:

- Important features changed.
- Model should have reacted, but did not. It does not extrapolate well.
- We need to intervene.

Nuanced interpretations of drift

Data Drift: detected. Prediction drift: detected.

Positive interpretation:

- Important features changed.
- Model reacts and extrapolates well (e.g. prices lower > higher sales)
- No need to intervene.

Negative scenario:

- Important features changed.
- Model behavior is unreasonable.
- We need to intervene.

Drift Detected, what is next?

Data Quality:

- run before acting on the model predictions
- solve data quality issues if detected!

Prediction and Data Drift:

- interpret it: sometimes drift is OK if the real world changed!
- (label new data), and retrain the model
- calibrate or rebuild the model
- switch to an alternative process (fallback, rules, manual)
- change business logic or model post-processing (higher decision threshold, exclude certain segments, etc.)

Questions?

Emeli Dral
CTO Evidently AI
emeli@evidentlyai.com

Evidently on GitHub:
https://github.com/evidentlyai/evidently