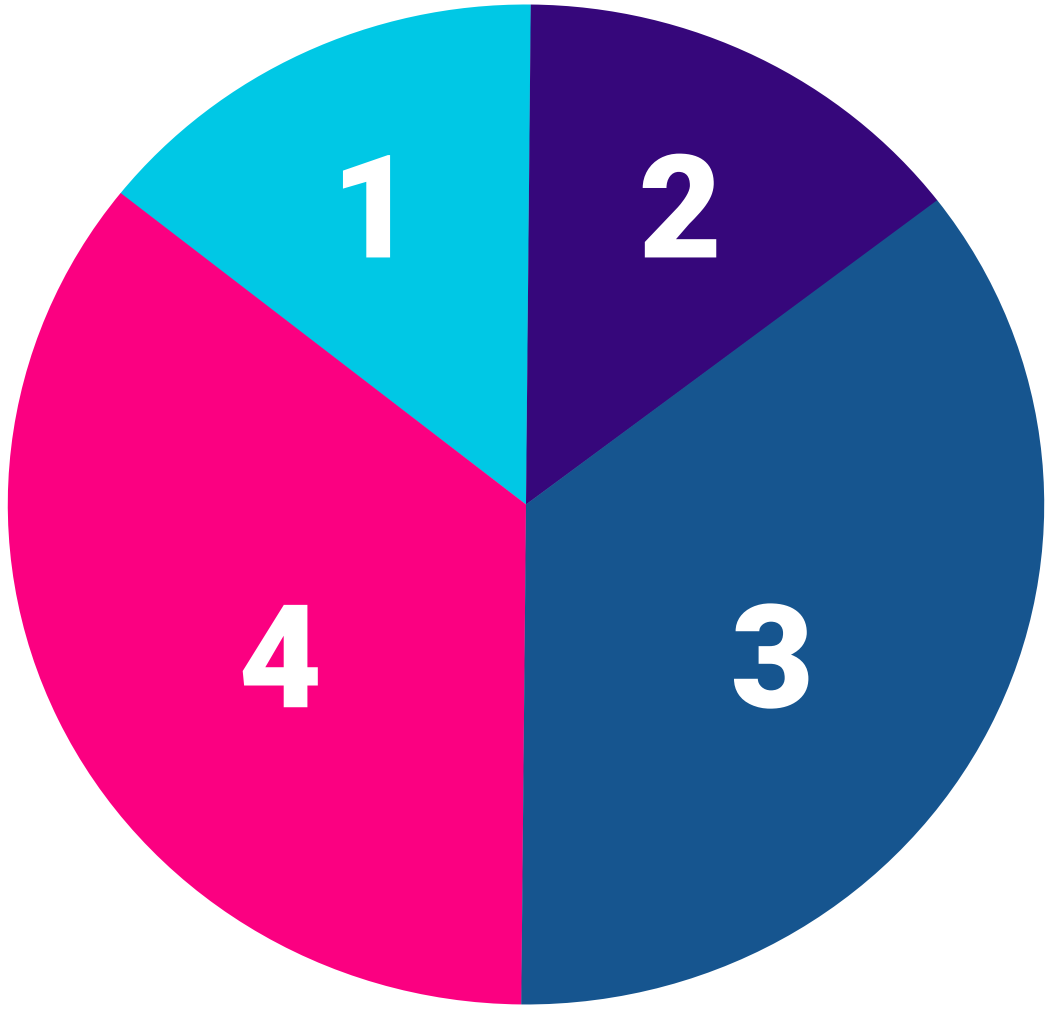
Hands-on ML monitoring flow with NannyML

by Wojtek Kuberski

Agenda

ML Monitoring Flow

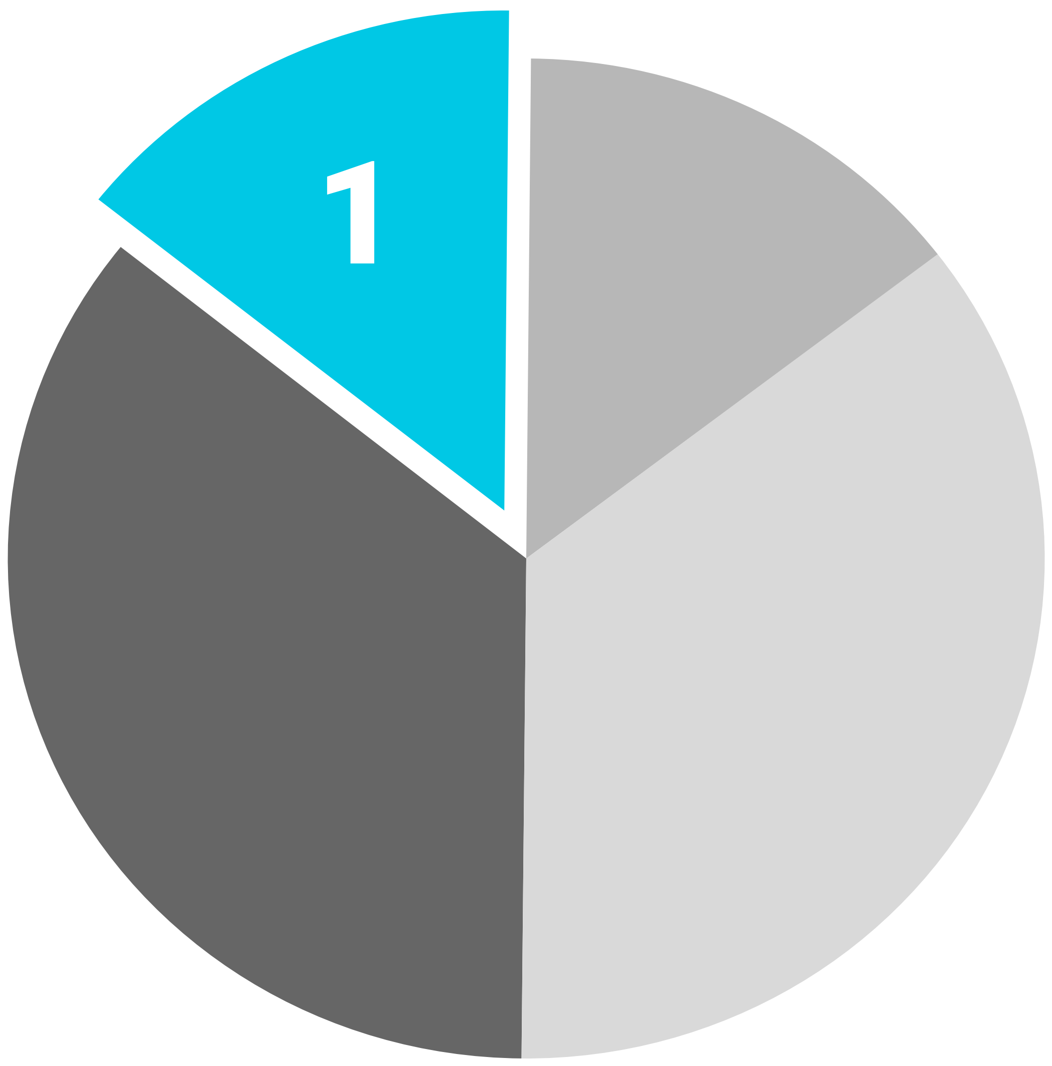
Performance Monitoring



Issue Resolution

Root Cause Analysis

The ML Monitoring flow



ML Monitoring Flow

The goals

1. Maintain the **business impact** of the model

2. Reduce **risk** of the model

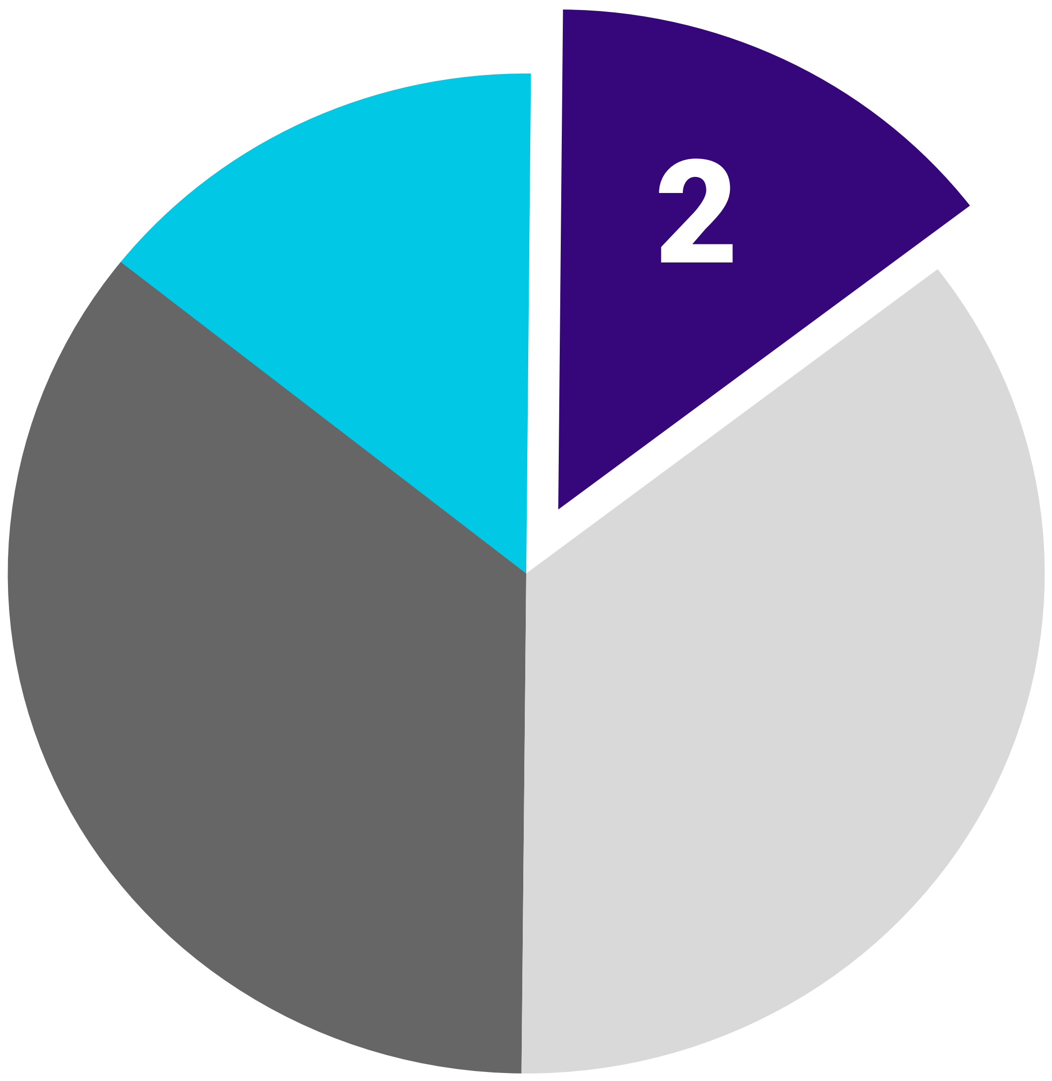
3. Increase **visibility** of the model

ML Monitoring Flow

The process

Performance Monitoring → Root Cause Analysis → Issue Resolution

Performance Monitoring



ML Monitoring Flow

The process

Performance Monitoring → Root Cause Analysis → Issue Resolution

Performance Monitoring

A crucial step in the flow

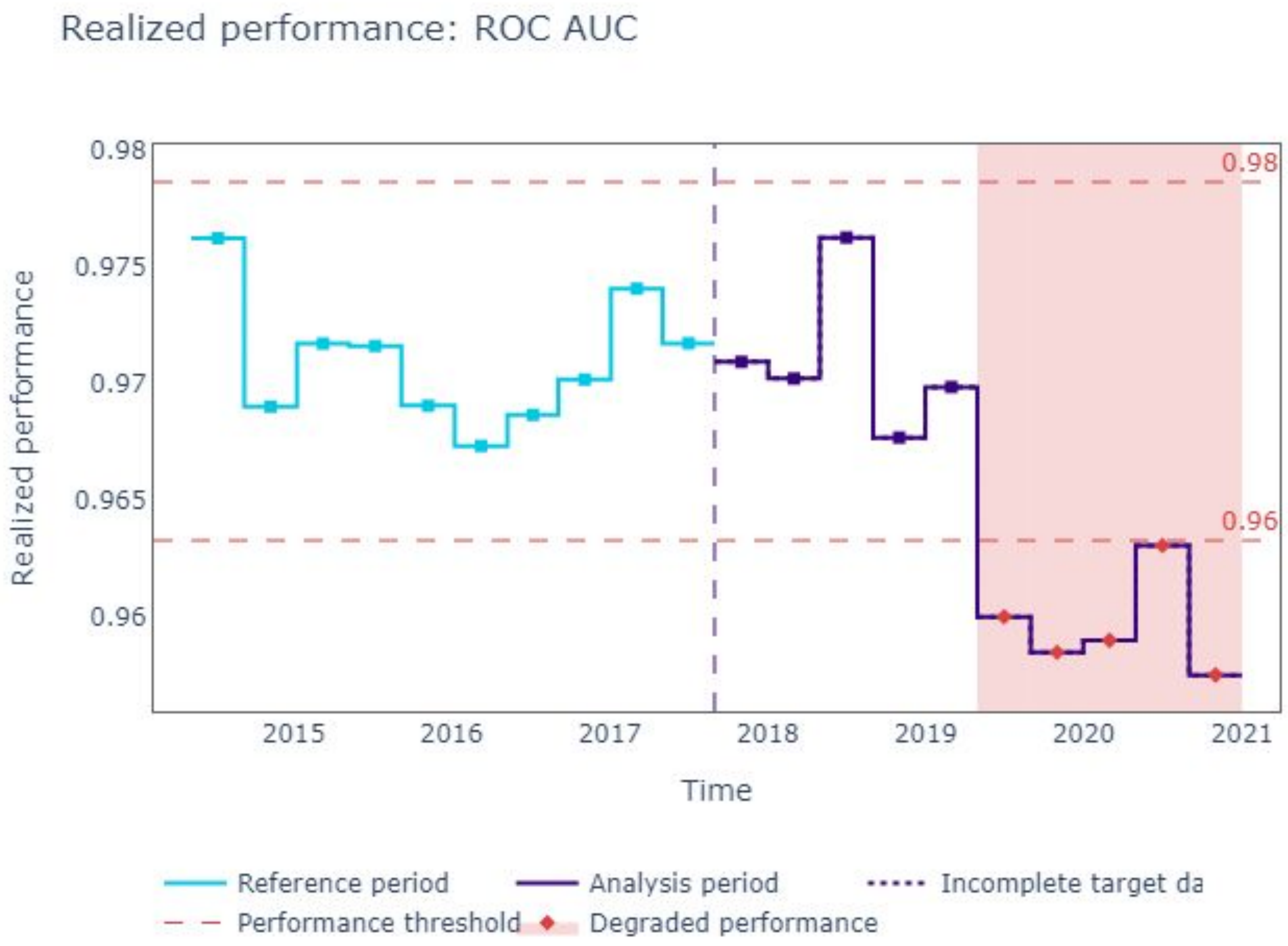
- Proxy for Business Impact
- Identifies potential issues
- Trigger for Root Cause Analysis

However

Not easily possible

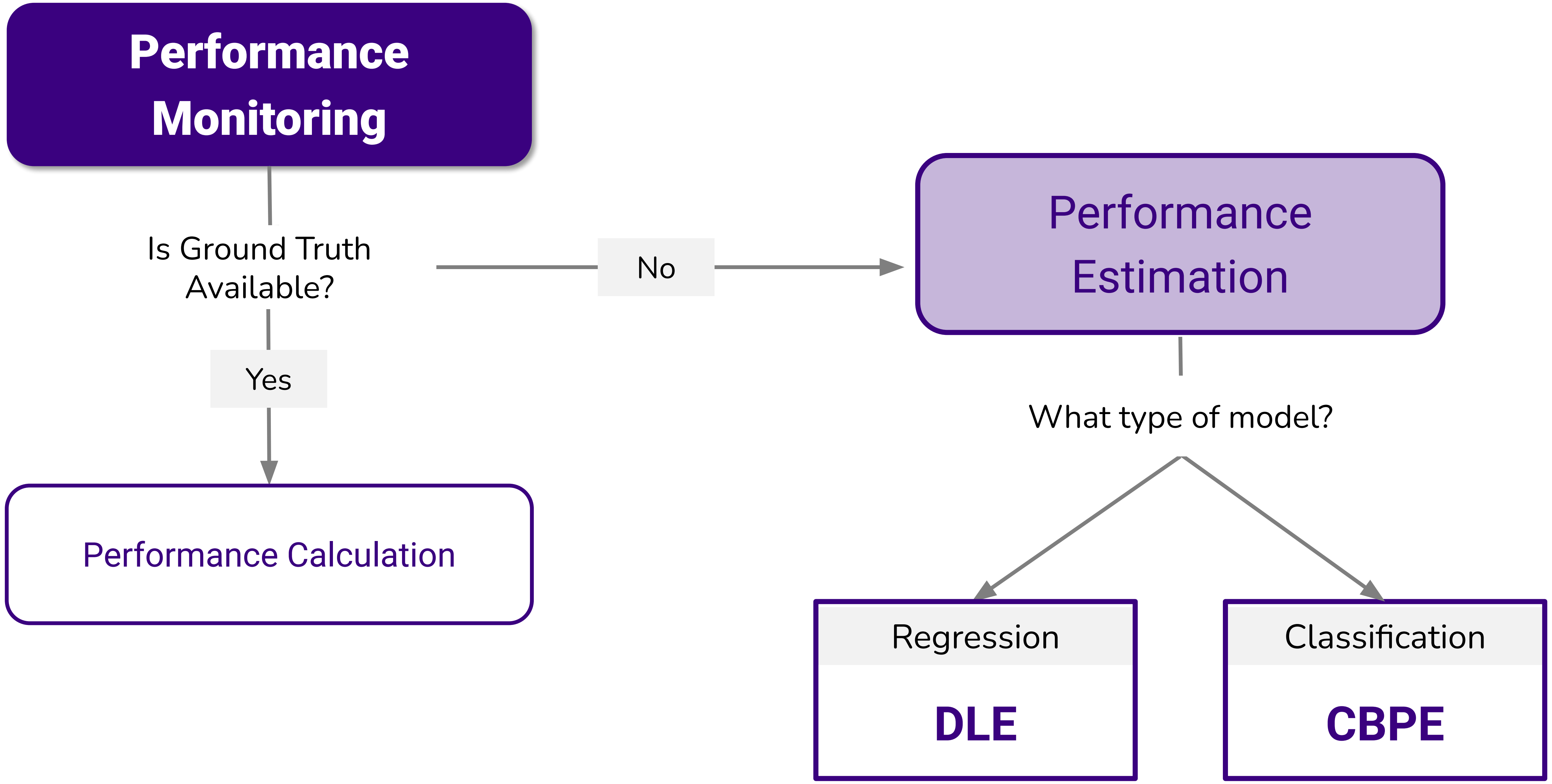
without target data

Performance **Estimation**
is a viable alternative



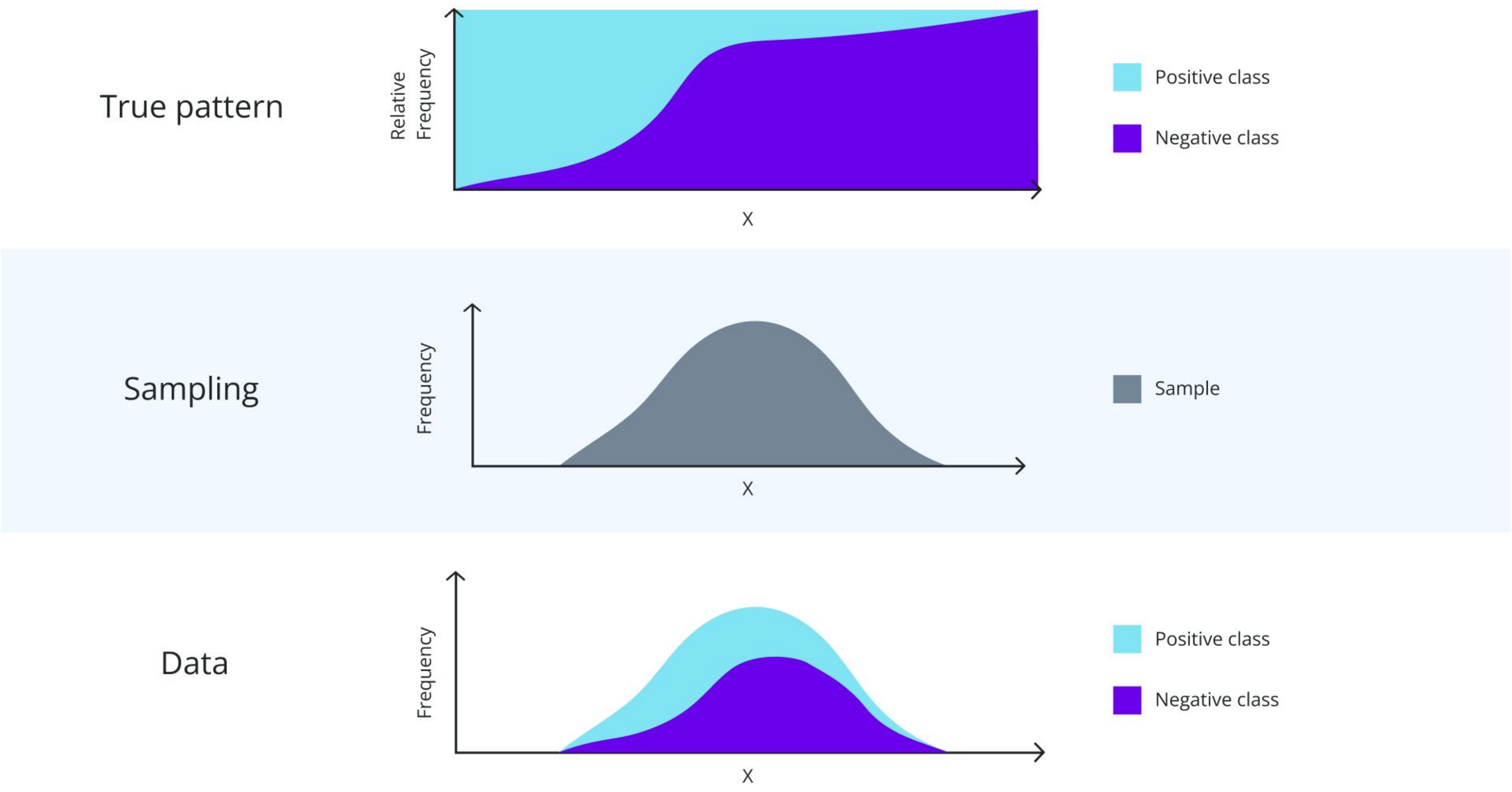
Performance Monitoring

Availability of Ground Truth



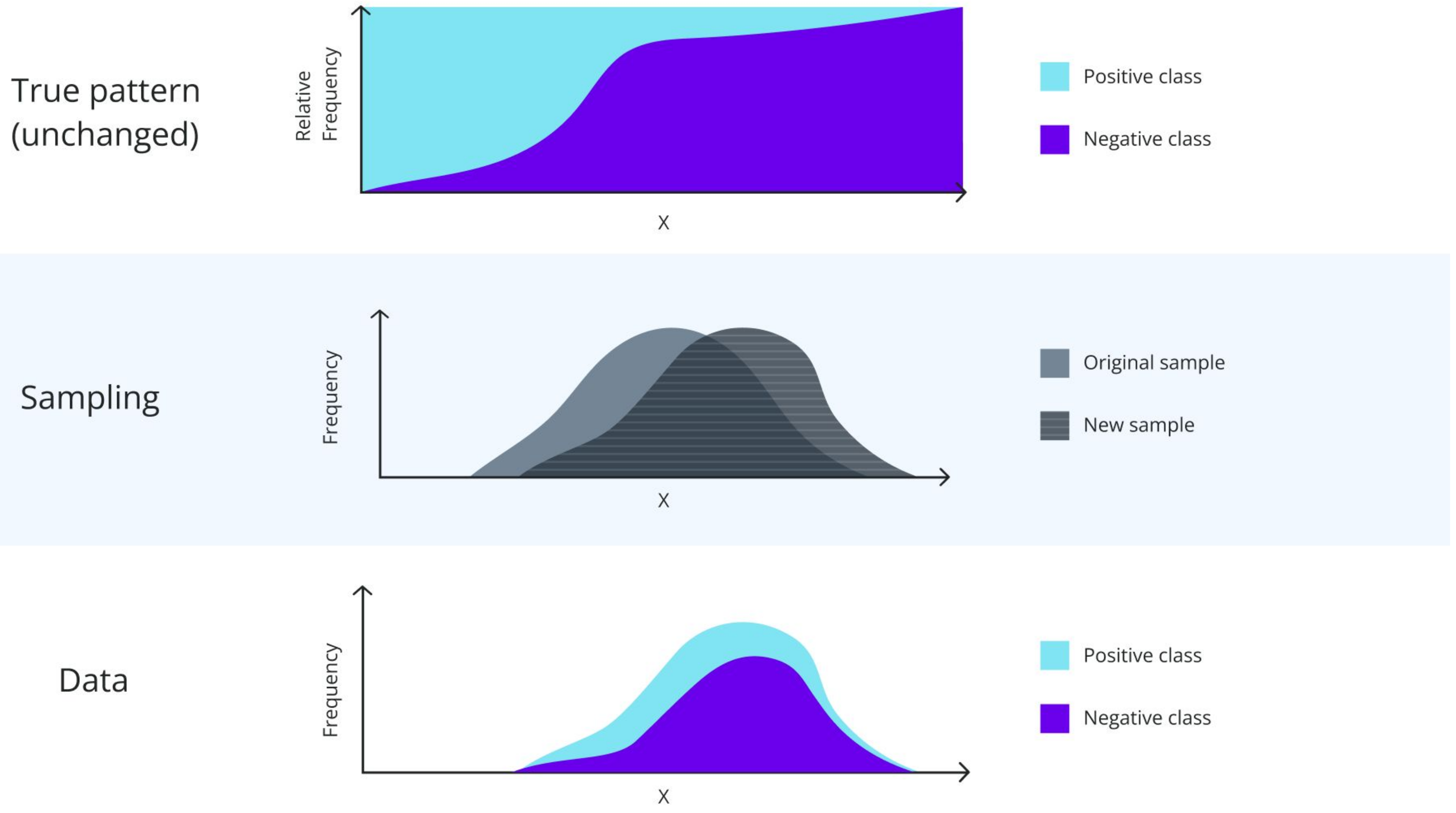
The two causes of silent model failure

The basics



The two causes of silent model failure

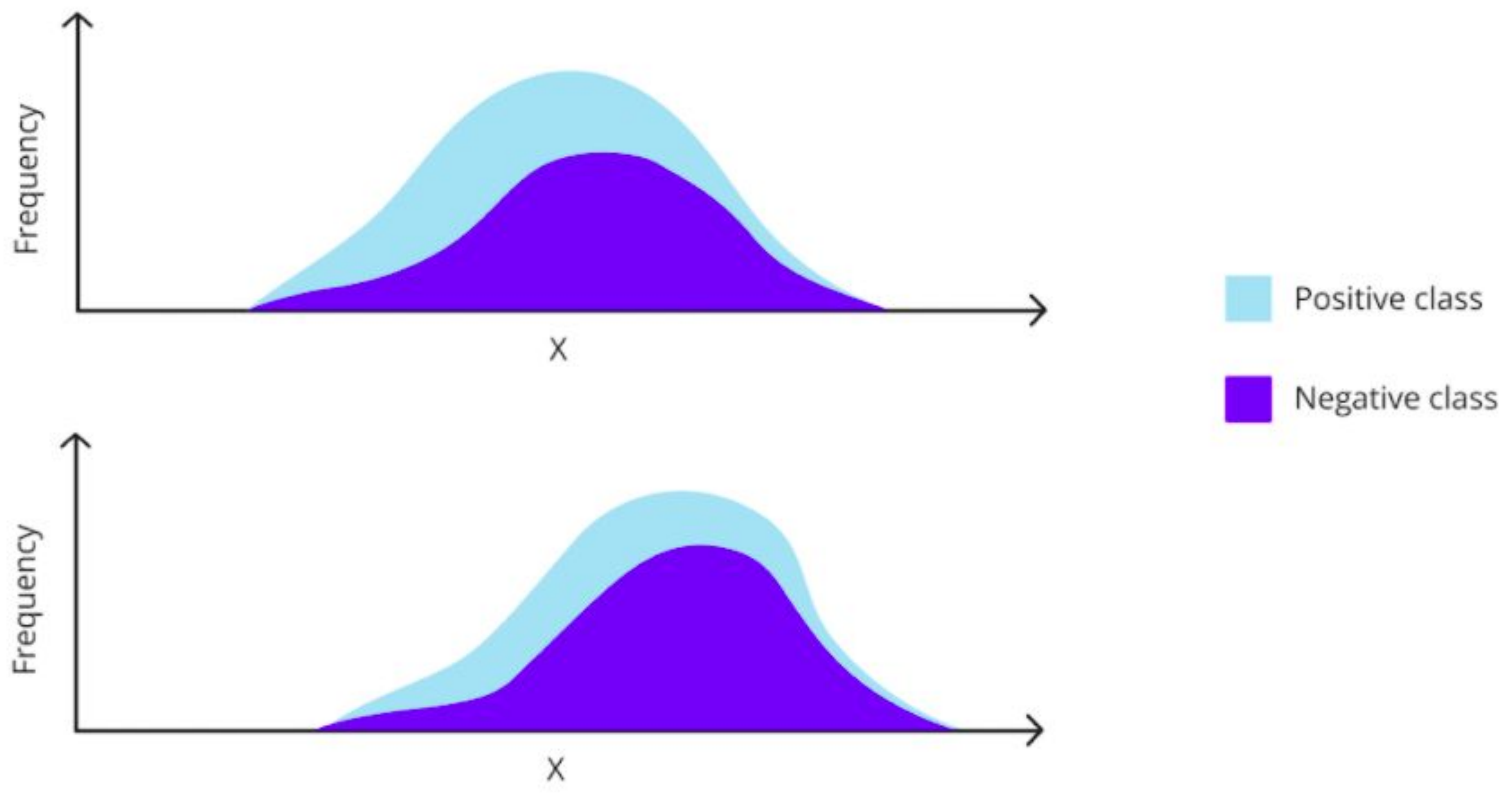
Covariate shift



The two causes of silent model failure

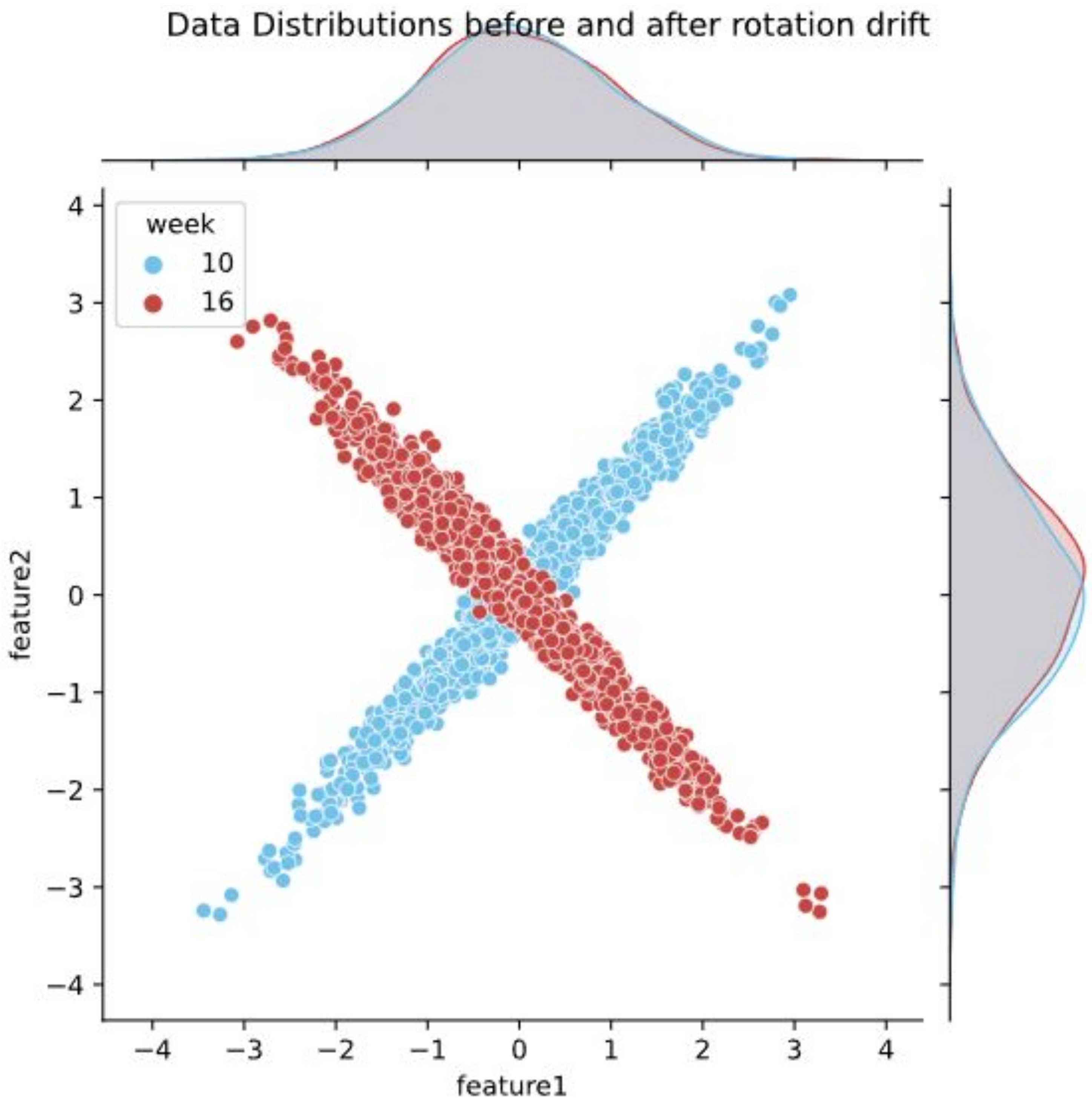
Covariate shift

Change in the joint model input distribution - P(X)



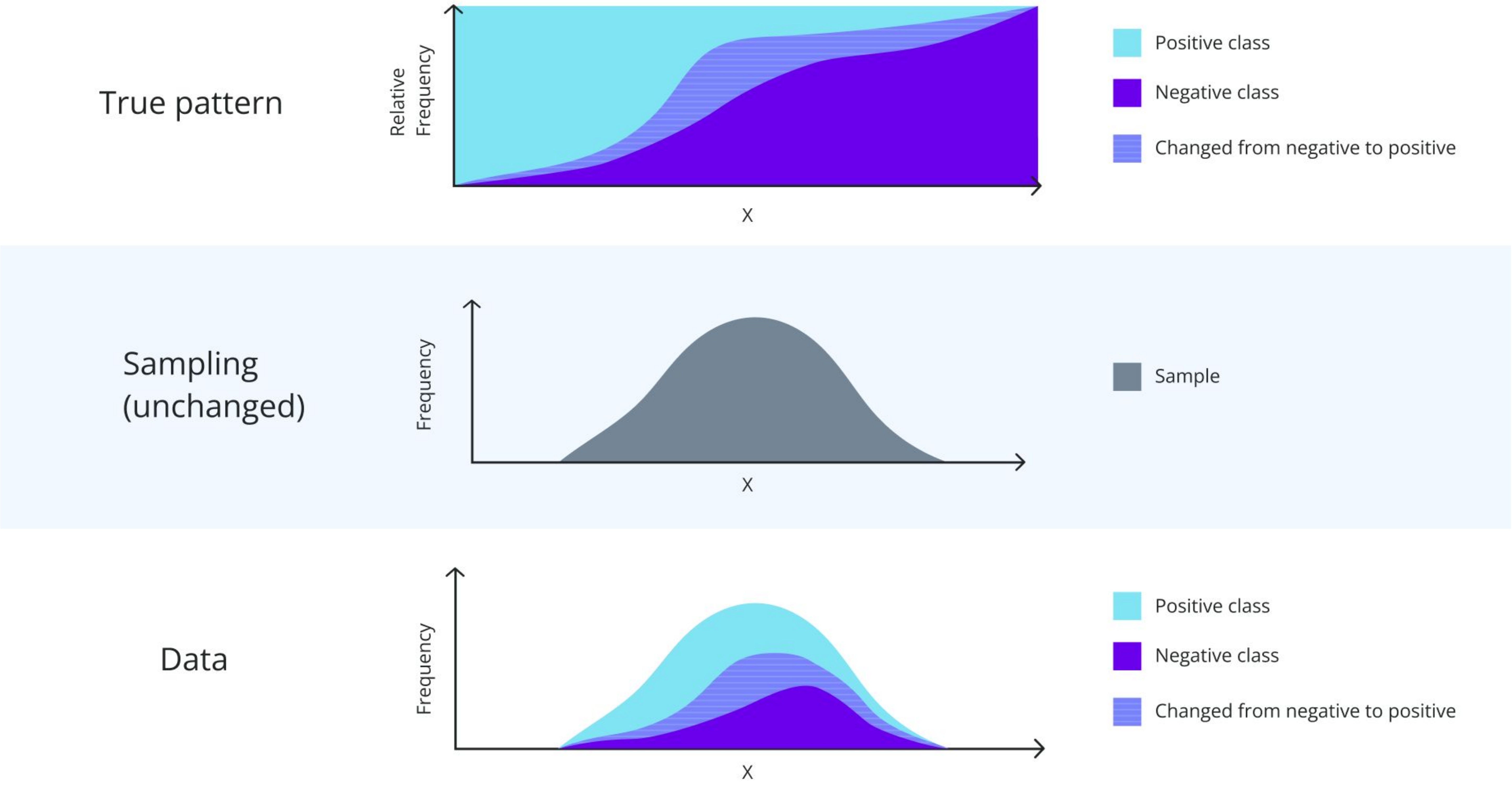
The two causes of silent model failure

Covariate shift: Multivariate example



The two causes of silent model failure

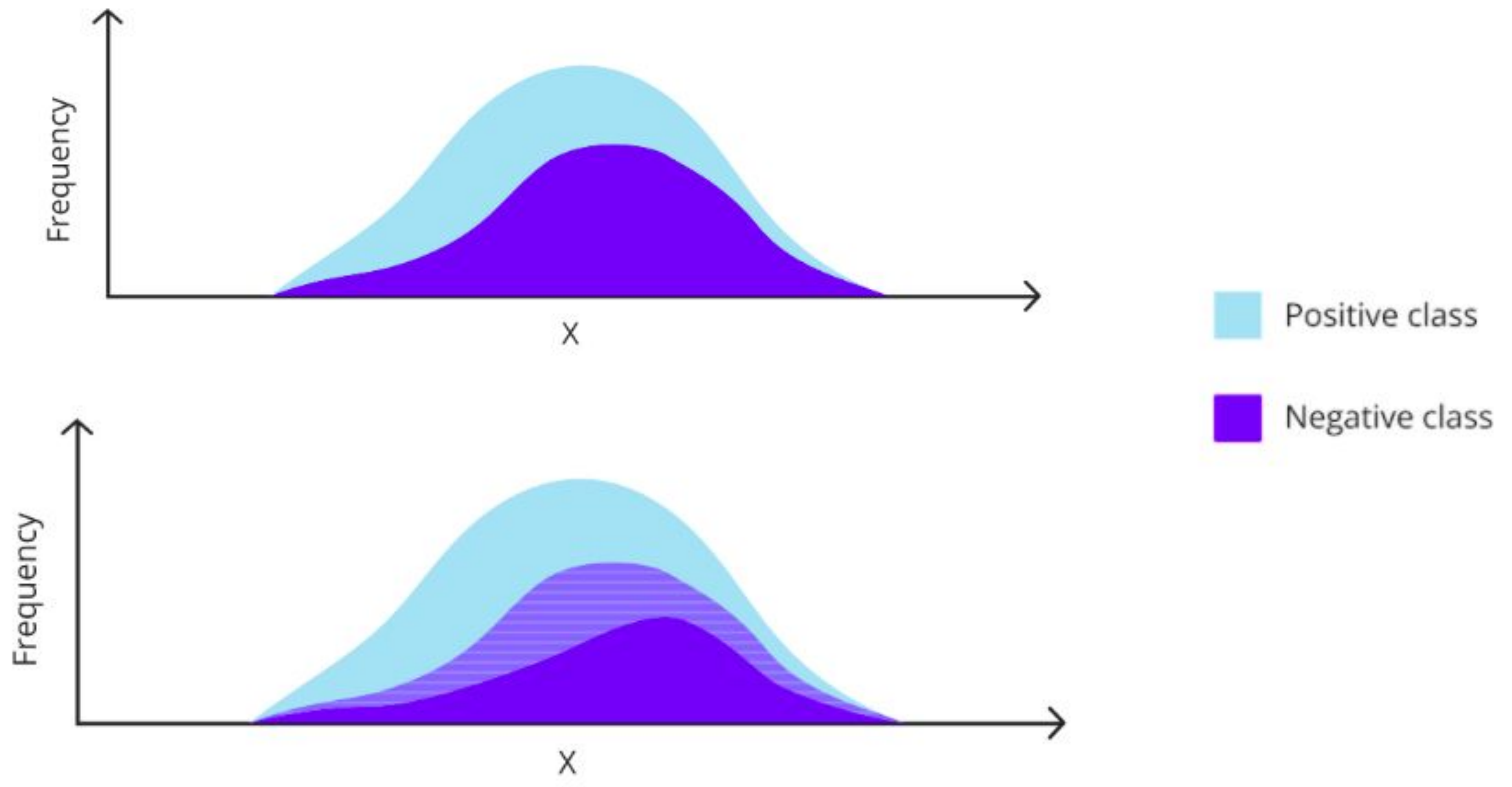
Concept drift



The two causes of silent model failure

Concept drift

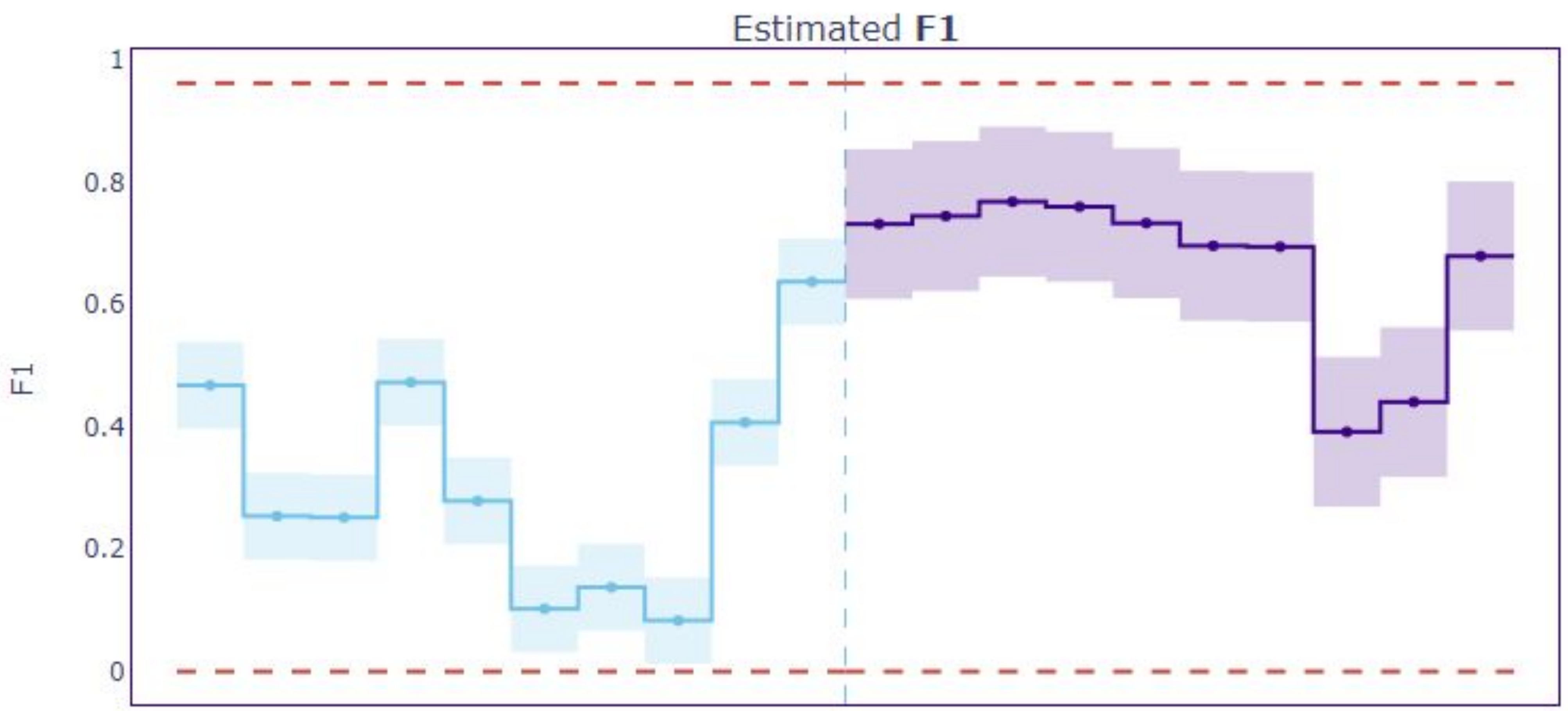
Change in the relationship between the target and the model inputs - P(y|X)



Performance Estimation for Classification

CBPE: What can we calculate?

- Confusion Matrix
- Precision, Recall, Accuracy, Specificity, F1..
- ROC AUC



Performance Estimation for Classification

CBPE: What does it do?

- Captures the full impact of covariate shift on performance
- Assumes no concept drift
- Assumes there is no covariate shift to previously unseen regions

Performance Estimation for Classification

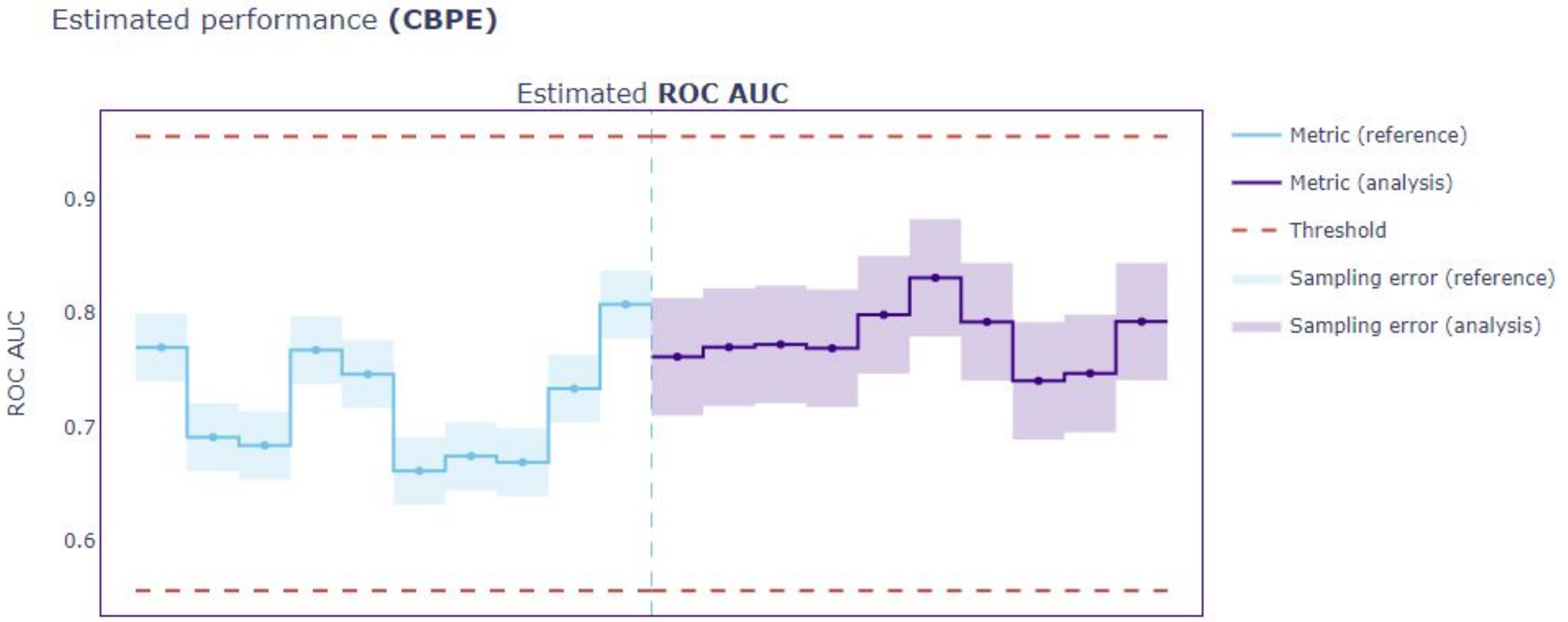
CBPE: Inputs

```
estimator = nml.CBPE(
   y_pred_proba = 'y_pred_proba',
   y_pred = 'y_pred',
   y_true = 'target',
   metrics = ['roc_auc', 'f1'],
   problem_type = 'classification_binary'
)
```

- Model scores (predicted probabilities)
- Model predictions
- Ground truth / target **(fitting only)**

Performance Estimation for Classification

CBPE: Results

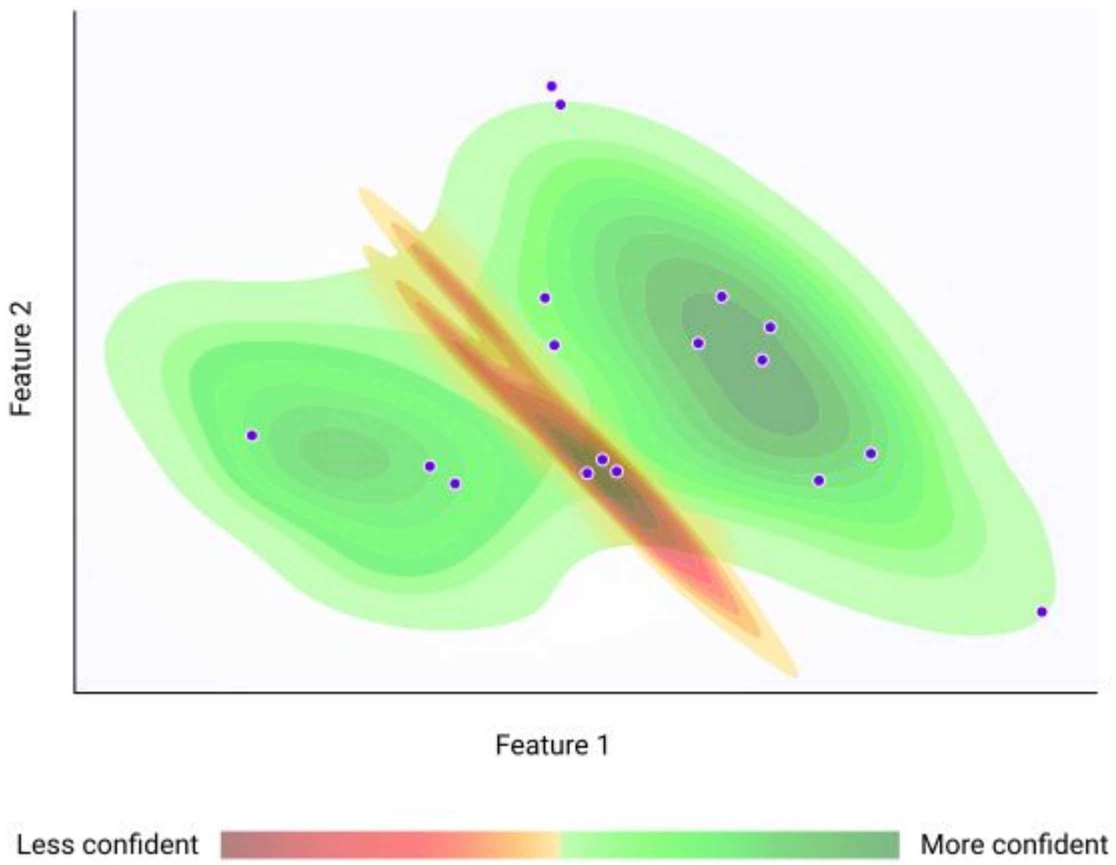


- Estimated metric
- Confidence bands
- Thresholds

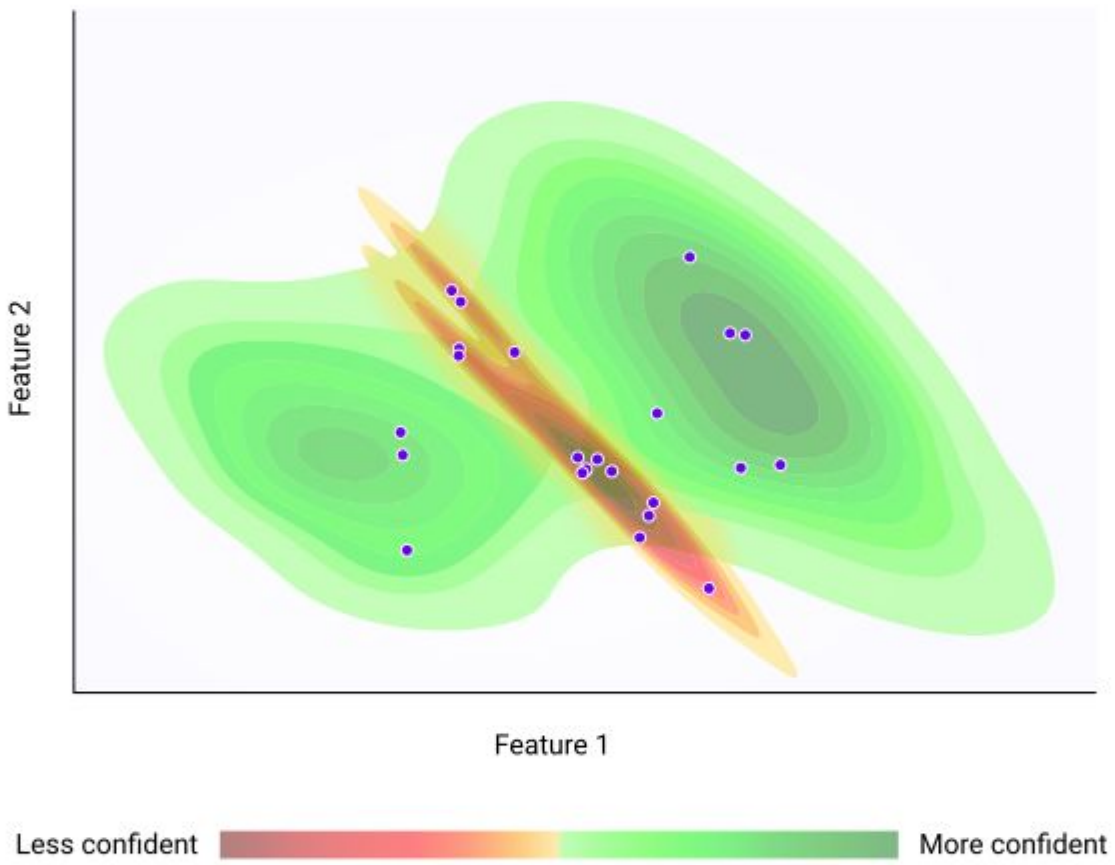
Performance Estimation for Classification

CBPE: Intuition

Test data



Production data



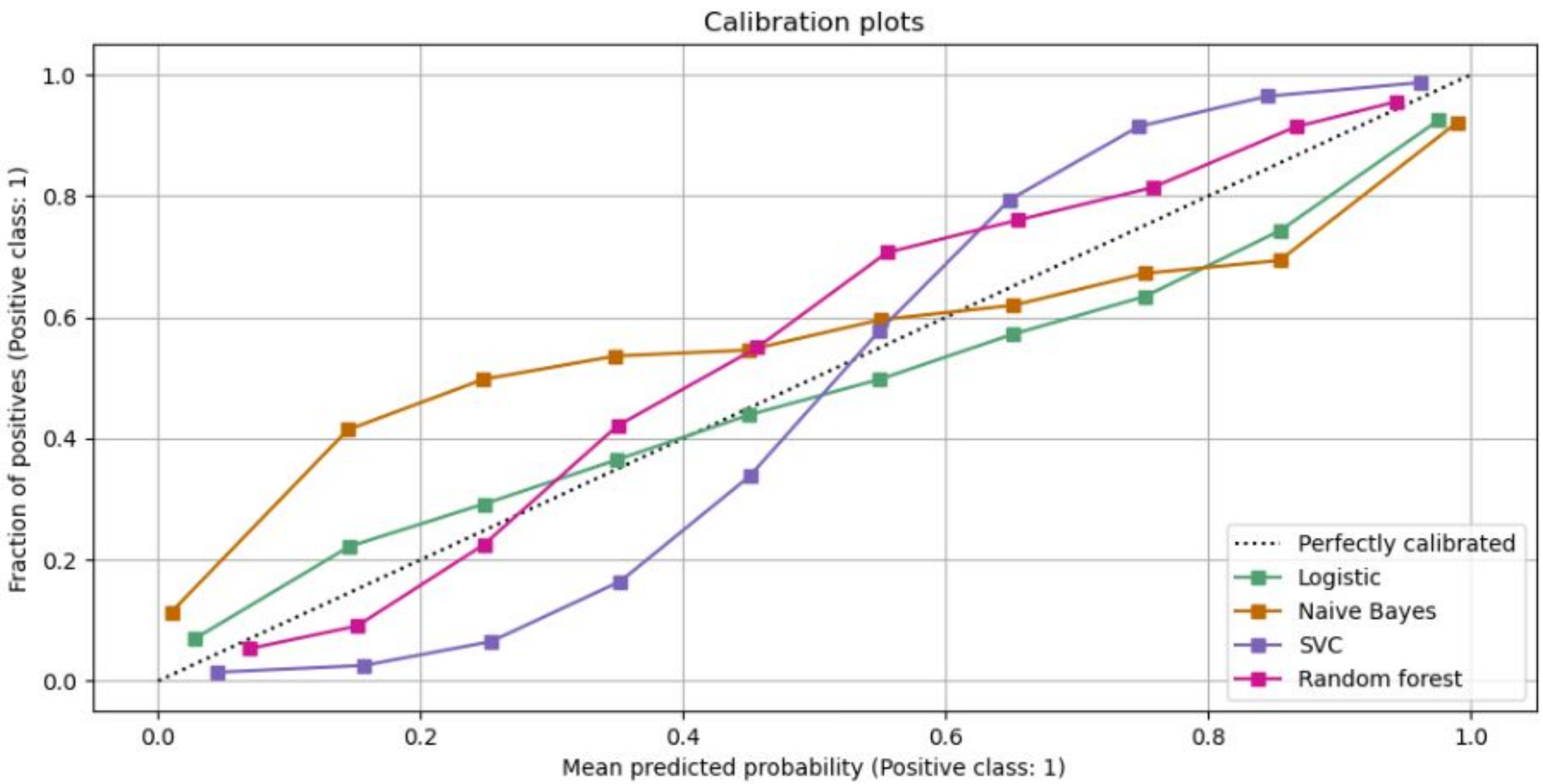
Performance Estimation for Classification

CBPE: Algorithm

1. Calibrate probabilities
2. Choose a threshold
3. **Find expected confusion matrix for every point**
4. Get aggregate confusion matrix
5. Compute a metric

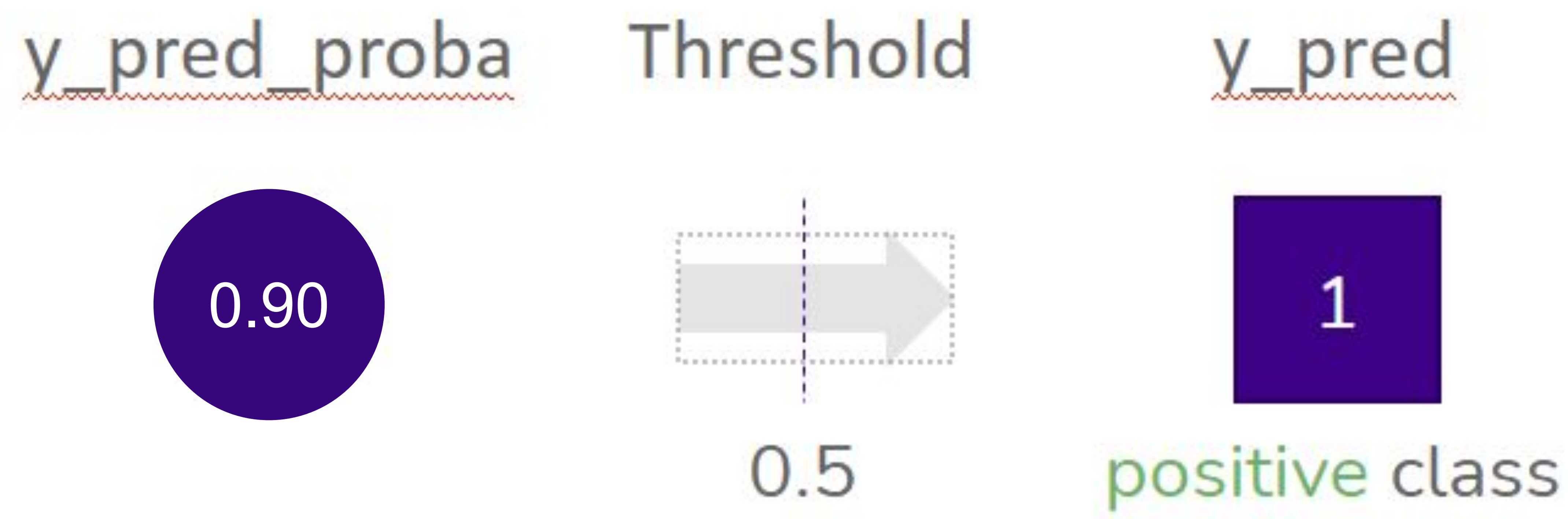
Classification with CBPE

Calibrate Probabilities



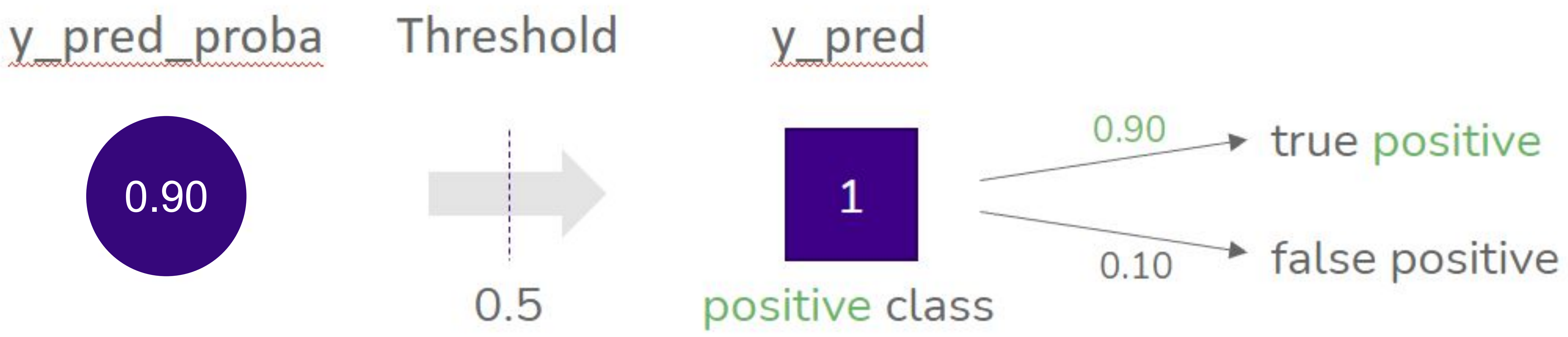
Classification with CBPE

Choose a threshold



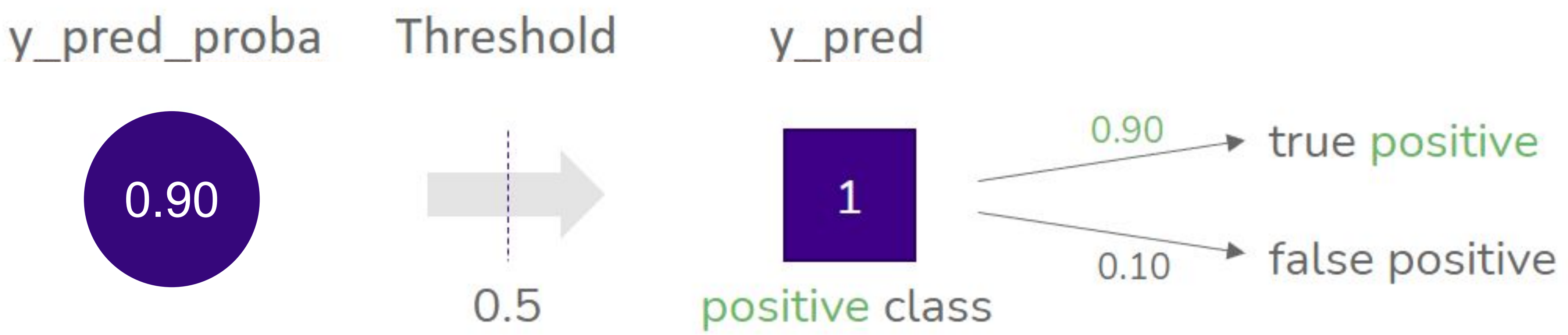
Classification with CBPE

Find expected confusion matrix for every point



Classification with CBPE

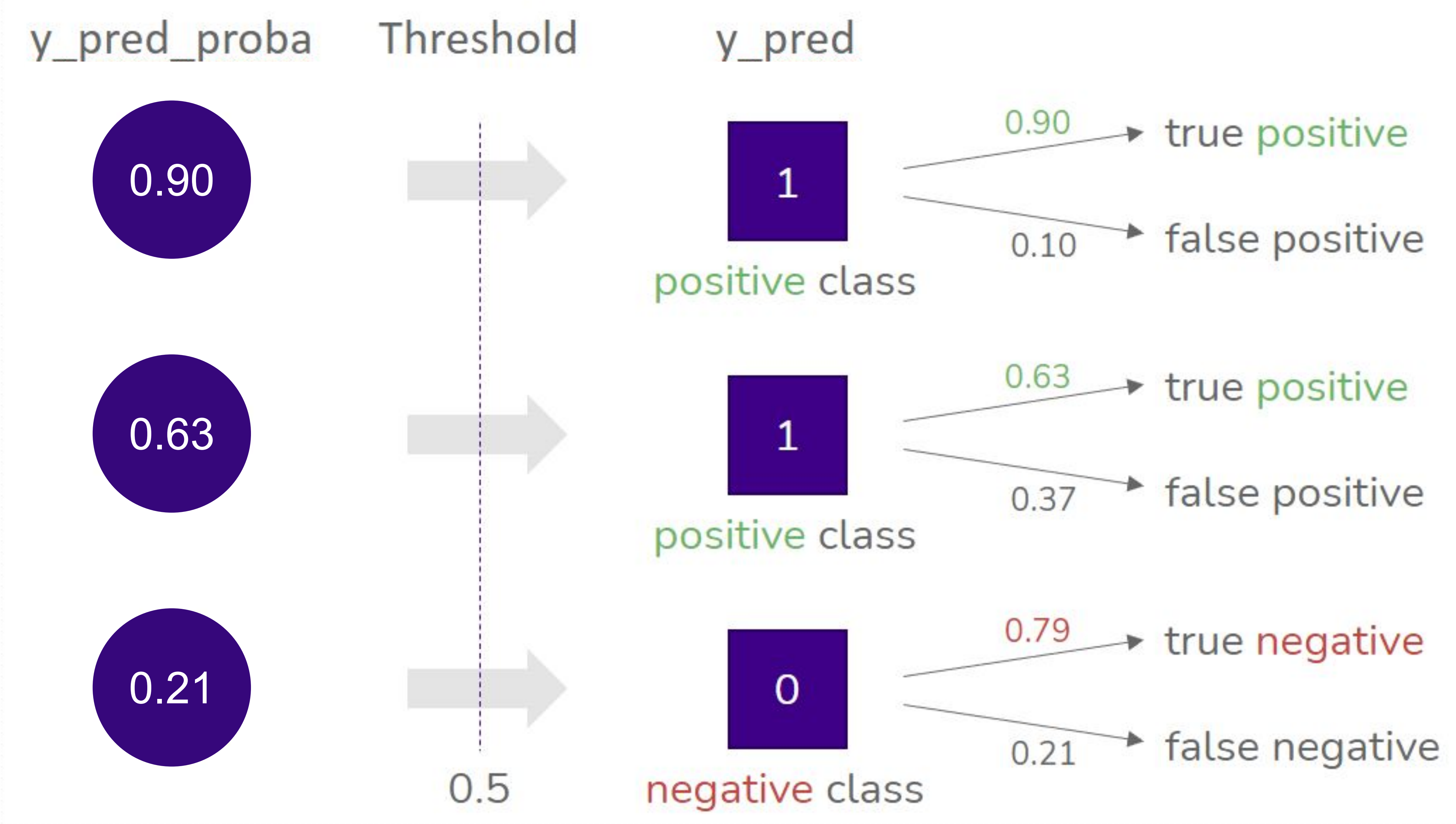
Find expected confusion matrix for every point



TP	TN	FP	FN
0.9	-	0.1	-

Classification with CBPE

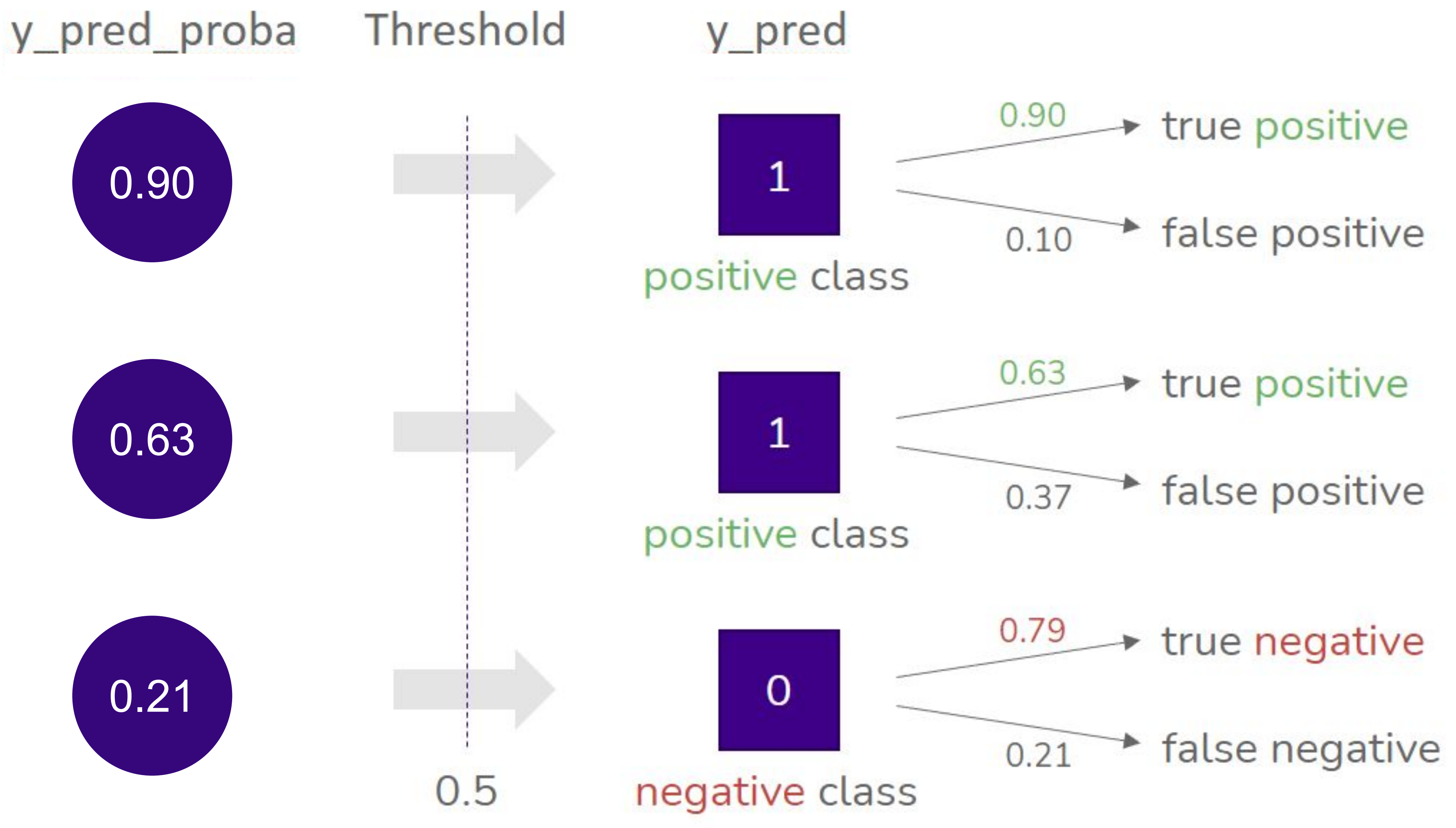
Find expected confusion matrix for every point



TP	TN	FP	FN
0.9	-	0.1	-
0.63	-	0.37	-
-	0.79	-	0.21

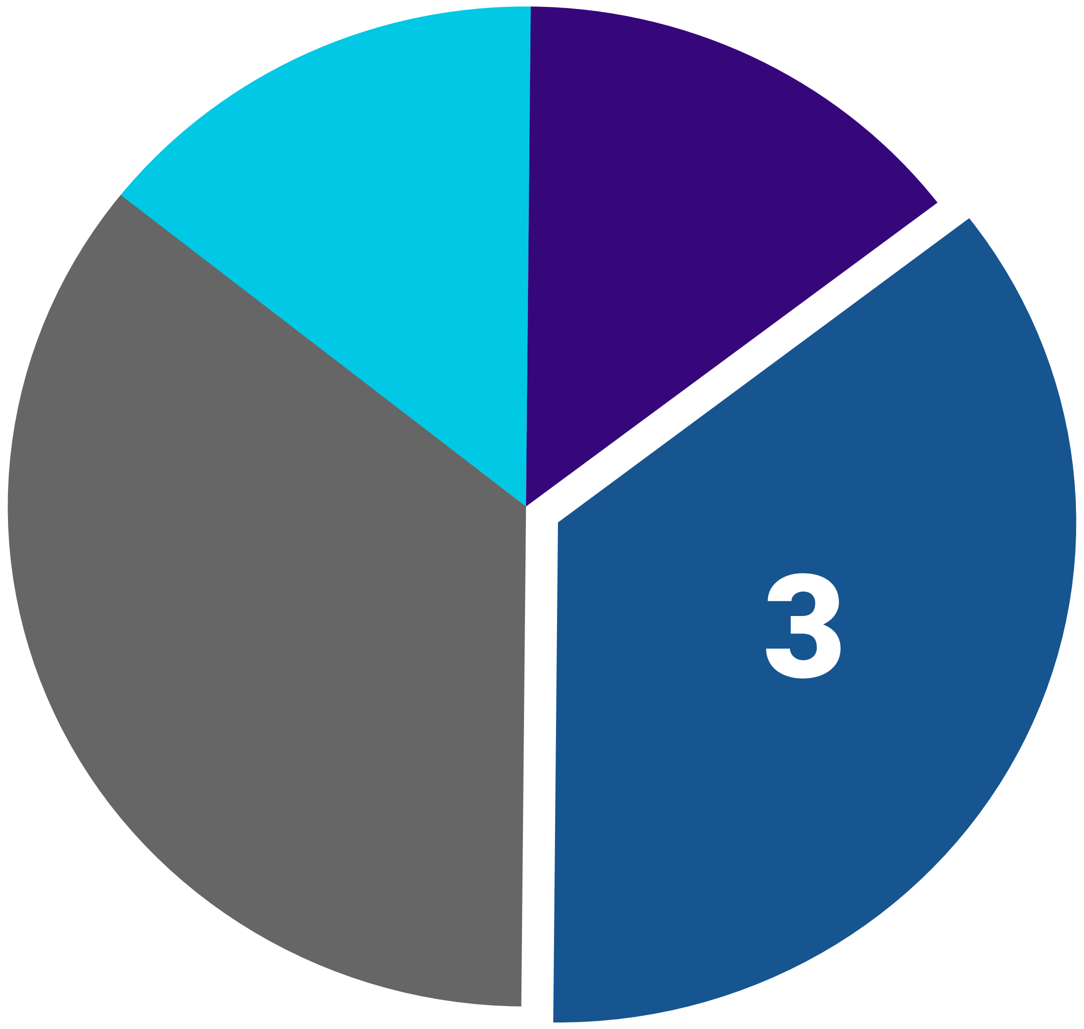
Classification with CBPE

Find expected confusion matrix for every point



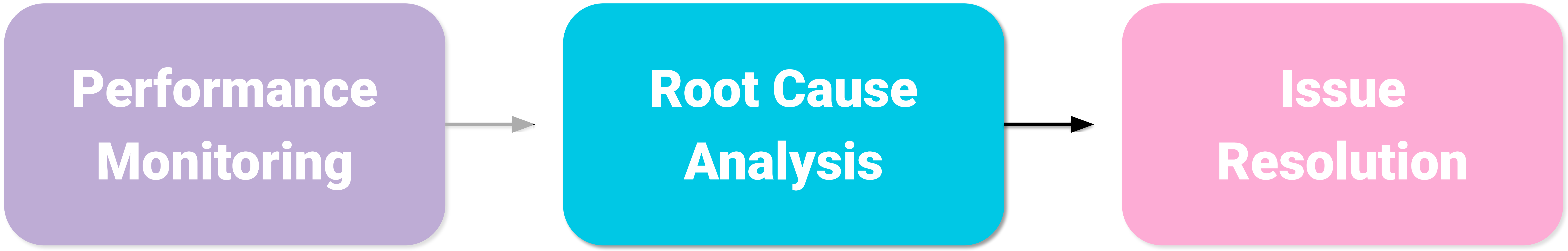
	TP	TN	FP	FN
	0.9	-	0.1	-
	0.63	-	0.37	-
	-	0.79	-	0.21
∑	1.53	0.79	0.47	0.21

Root Cause Analysis



Root Cause Analysis

The second step in the monitoring flow



Root Cause Analysis

The framework

Run drift detection and data quality monitoring → Correlate with Performance → Use domain knowledge to determine the root cause

Root Cause Analysis

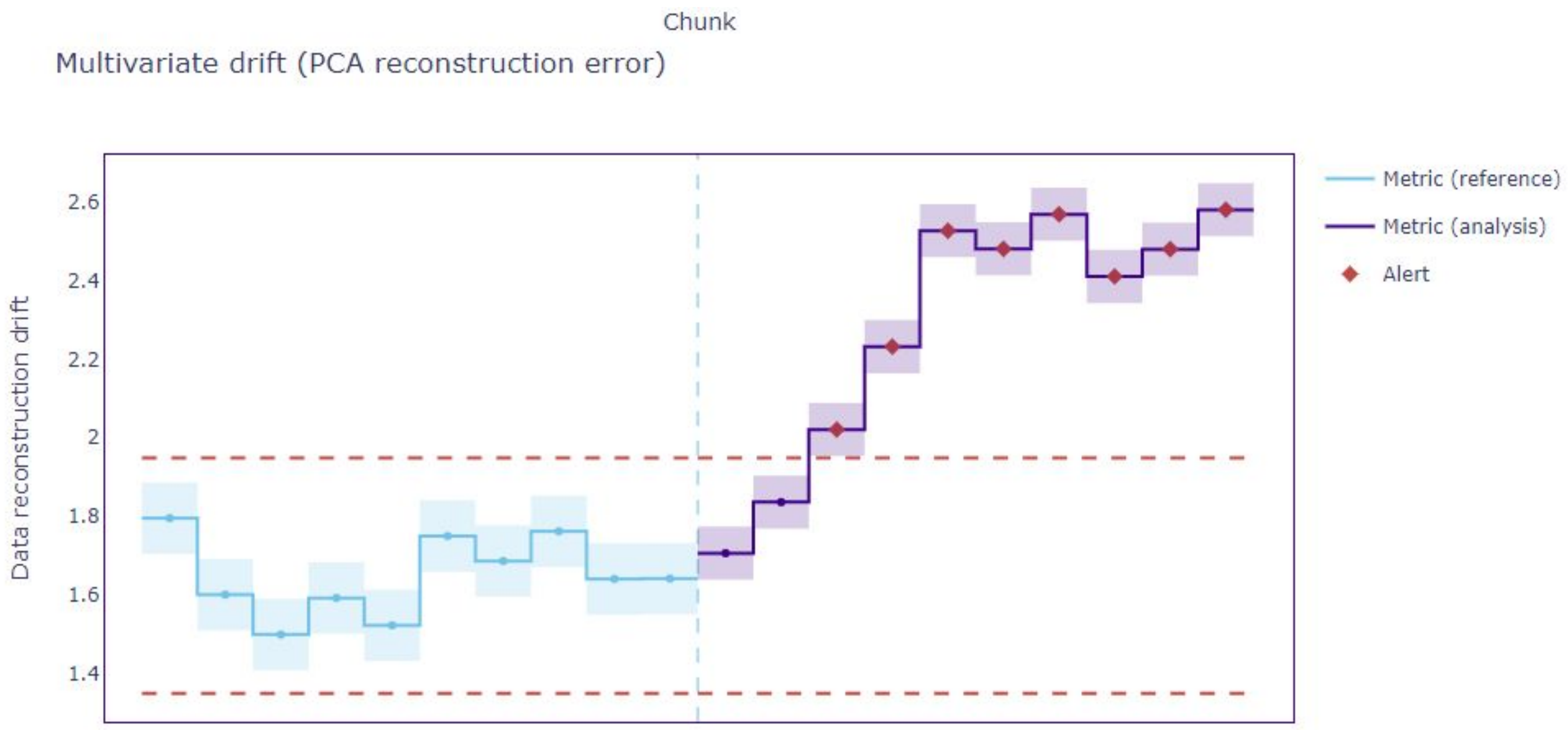
Drift Detection and Data Quality Monitoring

Run drift detection and data quality monitoring

- Univariate Drift Detection
  - Categorical Methods
  - Continuous Methods
- Multivariate Drift Detection
- Data Quality
  - Unseen Data
  - Missing Data

Multivariate Drift Detection
Results

- Reconstruction error
- Confidence bands
- Thresholds



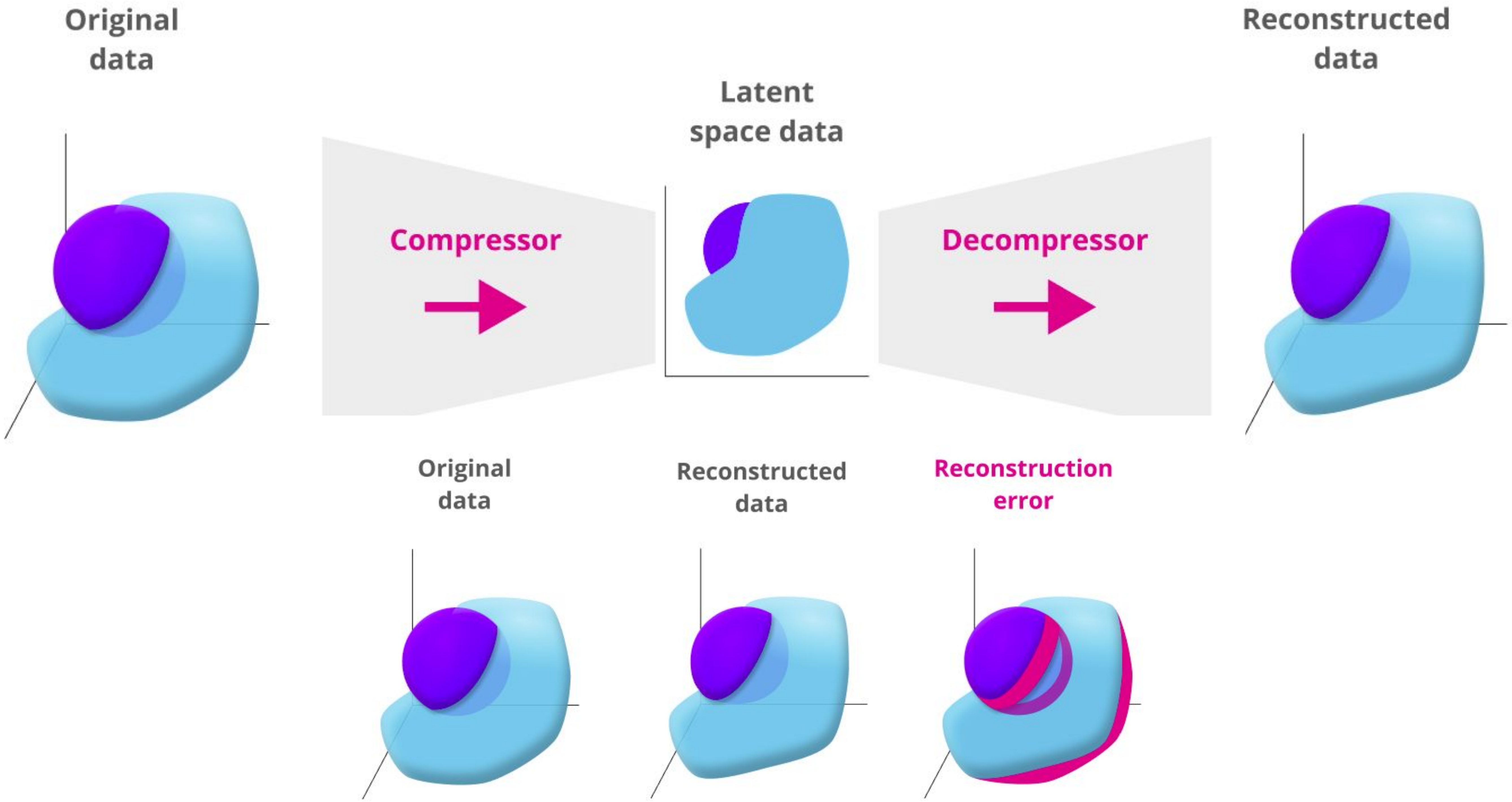
Multivariate Drift Detection

Characteristics

- Captures any linear change in relationship between features
- Captures changes in single feature distributions
- Requires at least 2 features to work
- Not easily interpretable

Multivariate Drift Detection

Intuition



Multivariate Drift Detection
Algorithm training

1. Prepare the data: impute missing values, encode categorical features, scale the data
2. Train PCA on reference data
3. Compress and decompress the reference data using trained PCA
4. Compute the distance between original and reconstructed points
5. Compute the reconstruction error: the average of those distances

Multivariate Drift Detection

Algorithm drift detection

1. Compress and decompress the analysis data using trained PCA
2. Compute the distance between original and reconstructed points
3. Compute the reconstruction error: the average of those distances
4. Compare this reconstruction error with the reconstruction error we got on the reference data

Univariate Drift Detection

At-a-glance comparison

Name	Categorical	Continuous	Range
Jensen-Shannon Distance	Yes	Yes	[0,1]
Hellinger	Yes	Yes	[0,1]
Wasserstein	No	Yes	[0,+inf]
L-Infinity	Yes	No	[0,1]
Kolmogorov-Smirnov	No	Yes	[0,1]
Chi-squared	Yes	No	[0,+inf]

Univariate Drift Detection

Structure of the Overview

1. **Basics**: what the method is
2. **Intuition**: how the method works
3. **Results**: example results on the same dataset

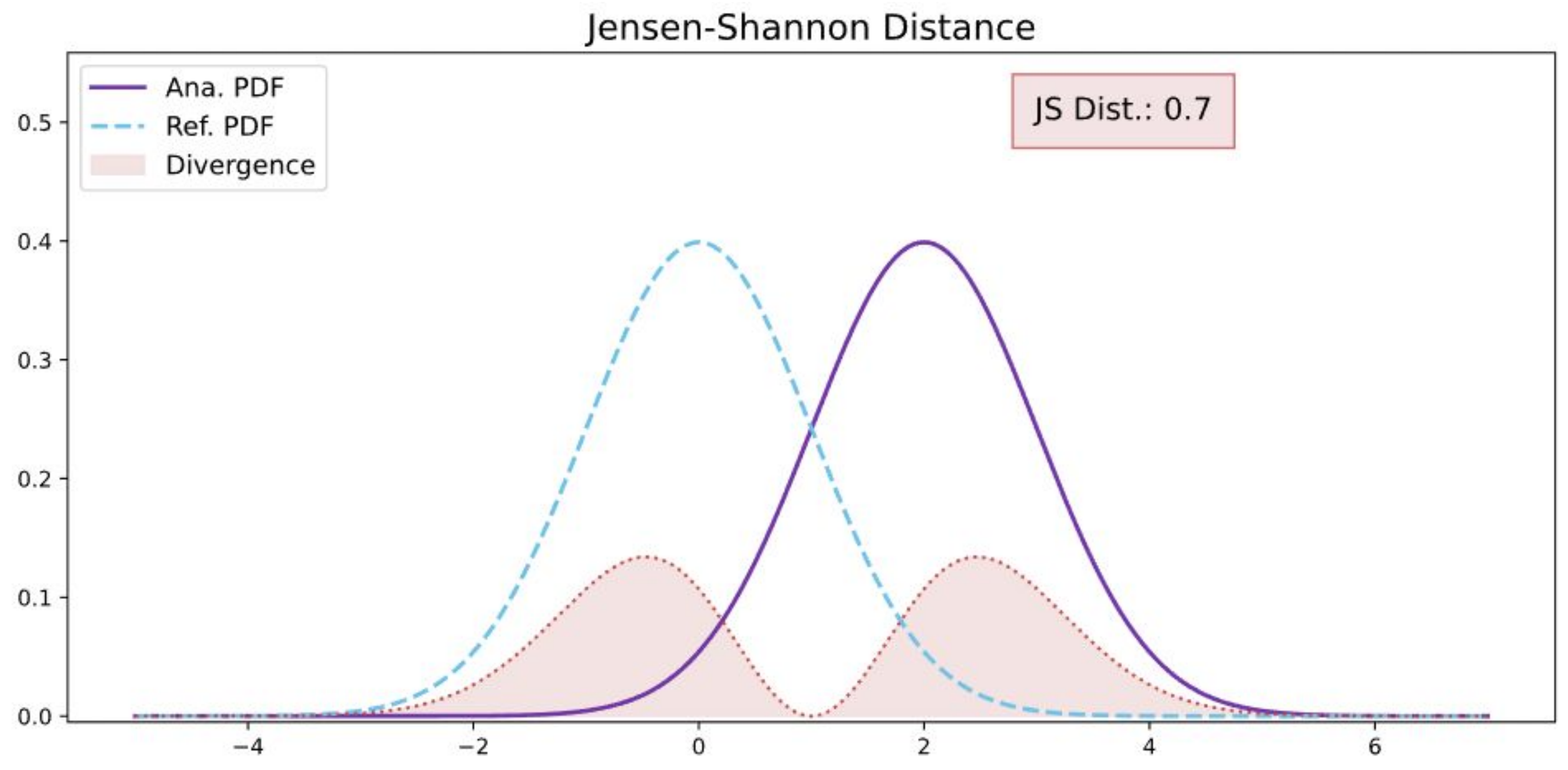
Univariate Drift Detection

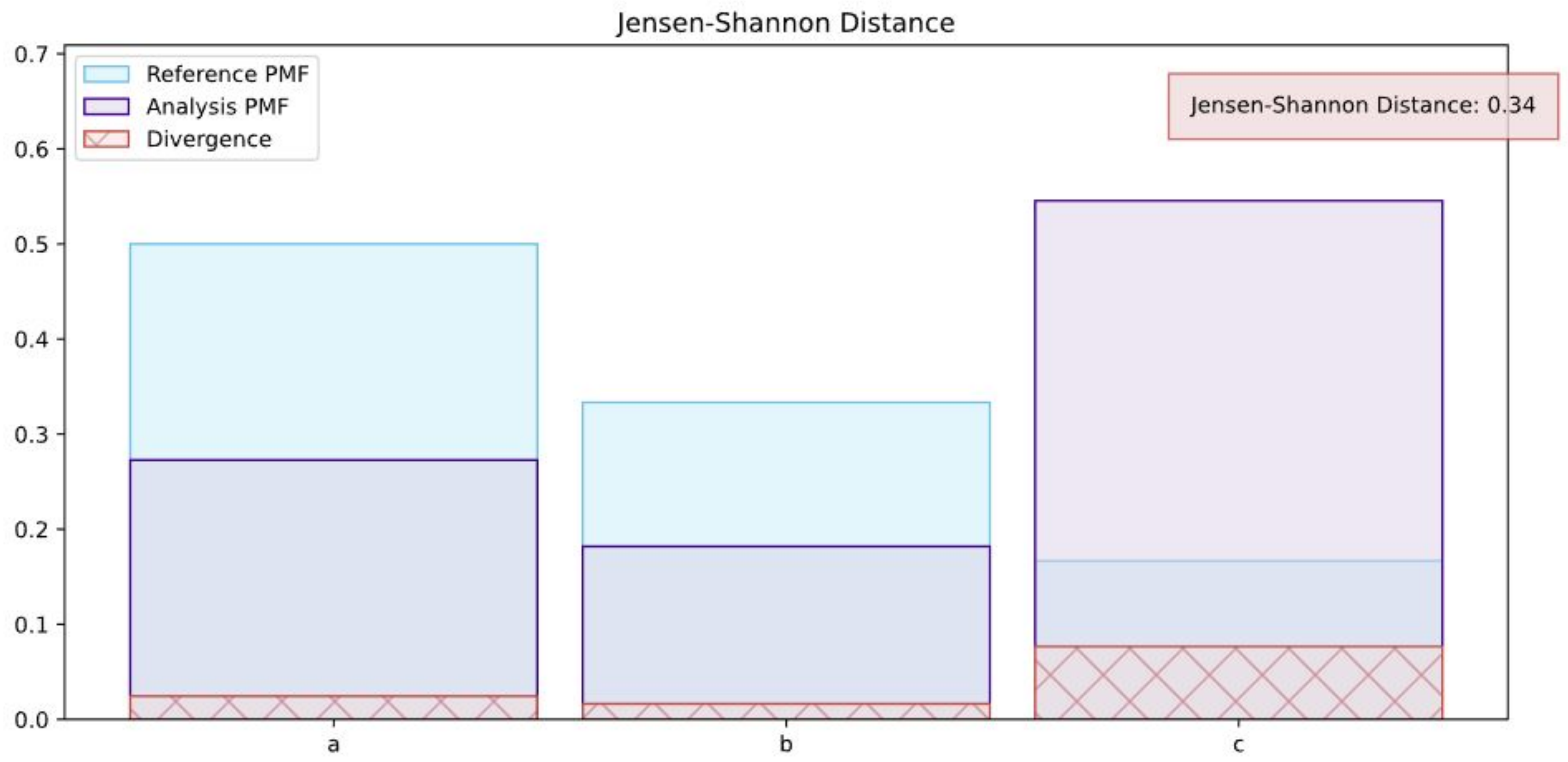
Jensen-Shannon distance: Basics

Data Type	Intuition	Range
continuous and categorical	Based on KL-divergence	[0,1]

Univariate Drift Detection

Jensen-Shannon distance: Intuition

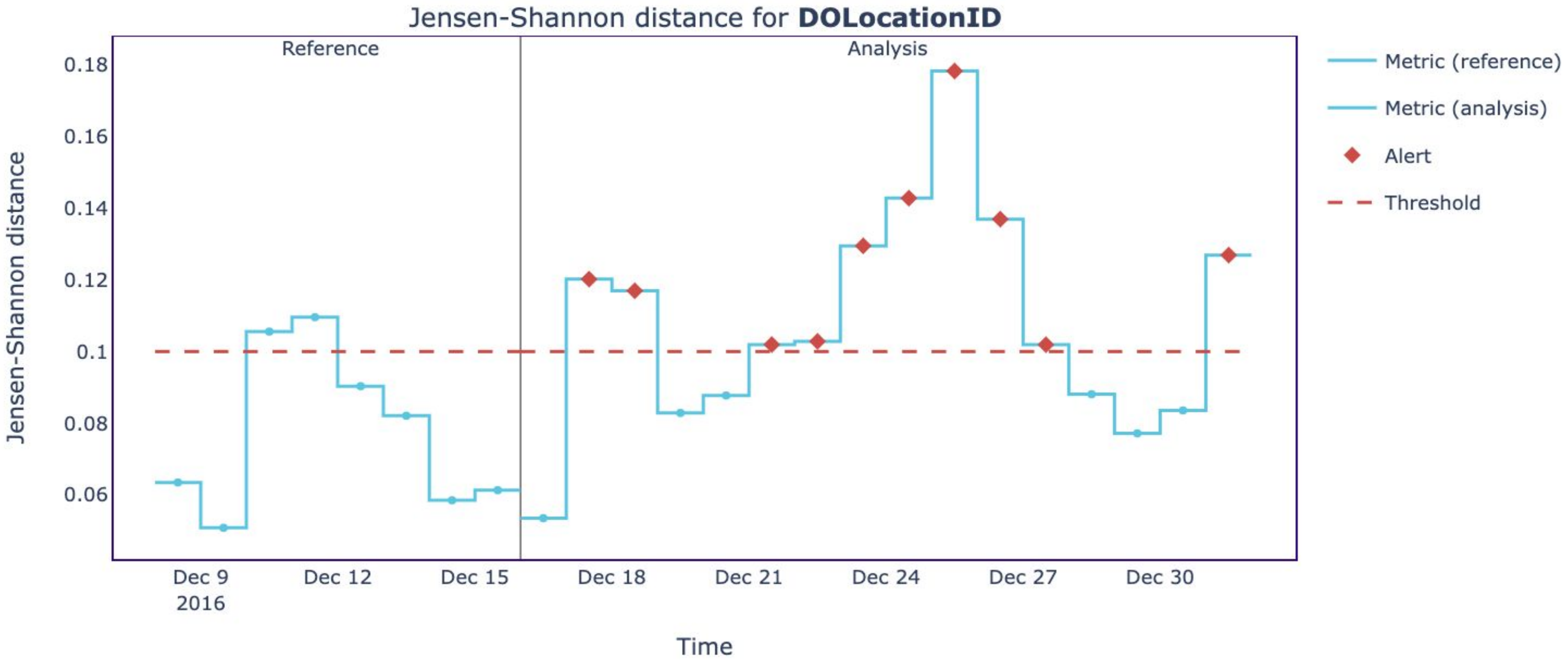




- Captures the “amount of overlap”
- “bi-directional” KL divergence
- Categorical: an average of all changes in relative frequencies of categories

Univariate Drift Detection

Jensen-Shannon distance: Results



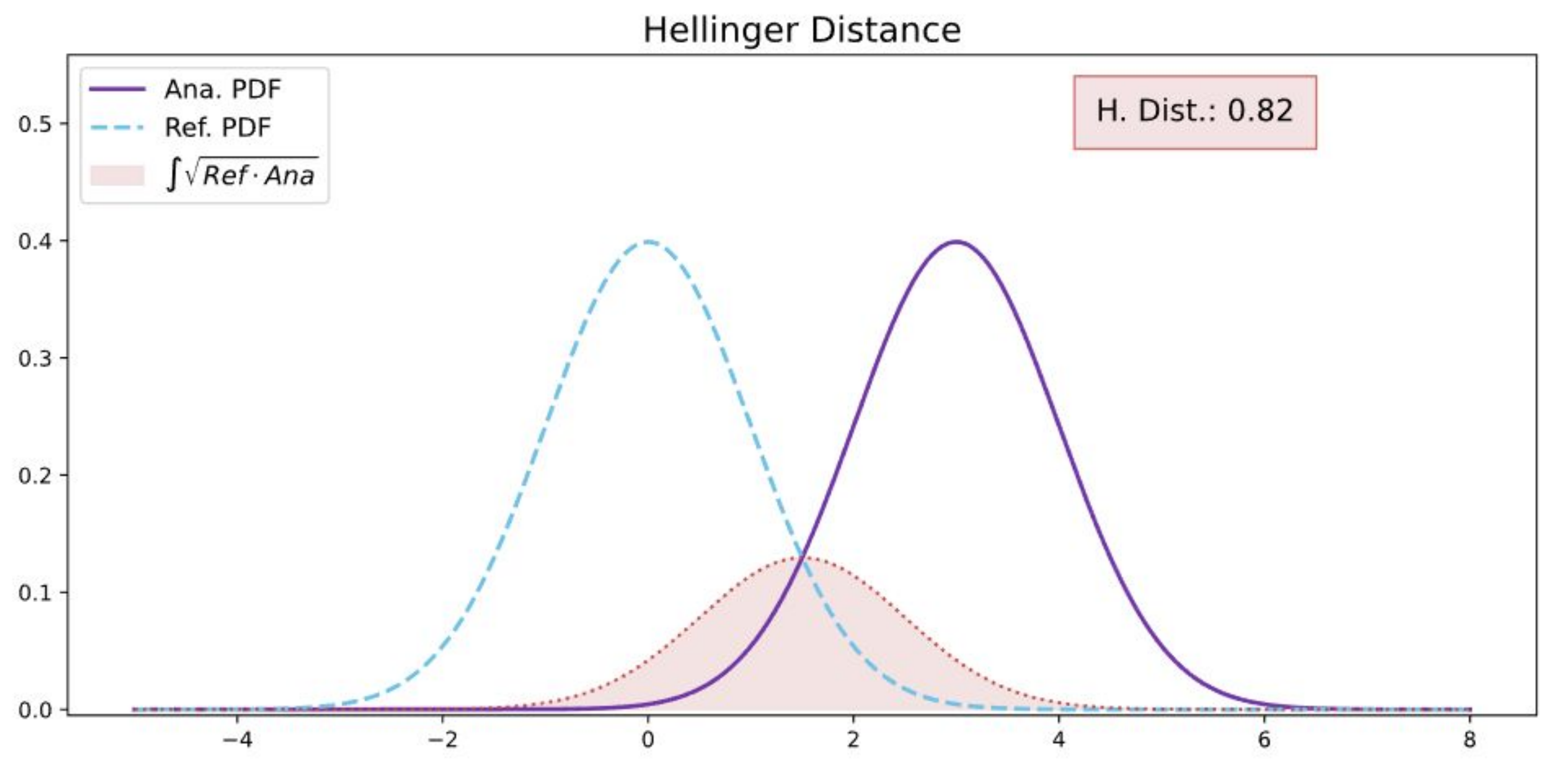
Univariate Drift Detection

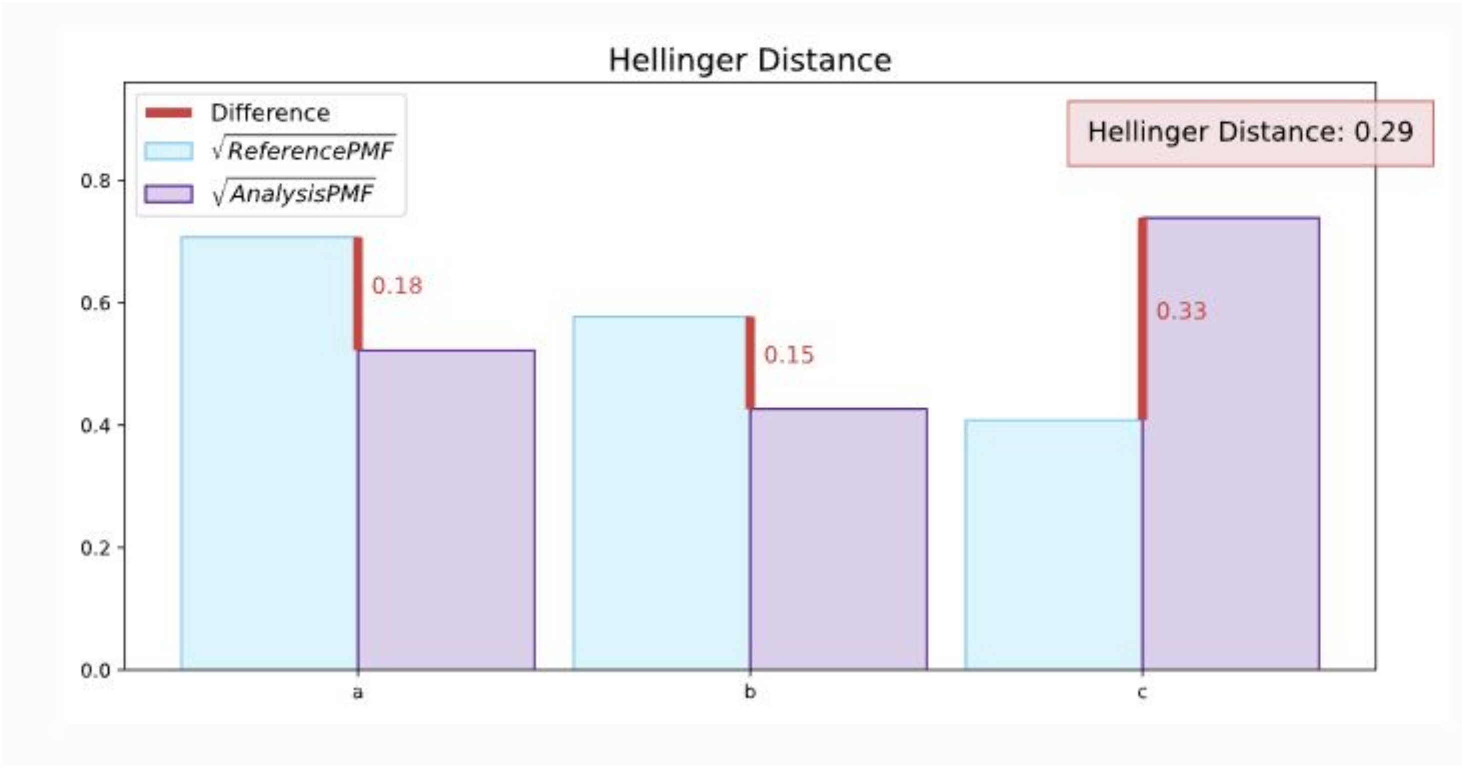
Hellinger distance: Basics

Data Type	Intuition	Range
continuous and categorical	Similar to Bhattacharyya Coefficient $H^2(P,Q) = 2(1 - BC(P,Q))$	[0,1]

Univariate Drift Detection

Hellinger distance: Intuition





- Overlap between the probabilities assigned to the same event by two distributions
- Doesn’t differentiate between very strong shifts
- Qualitatively similar to Jensen-Shannon

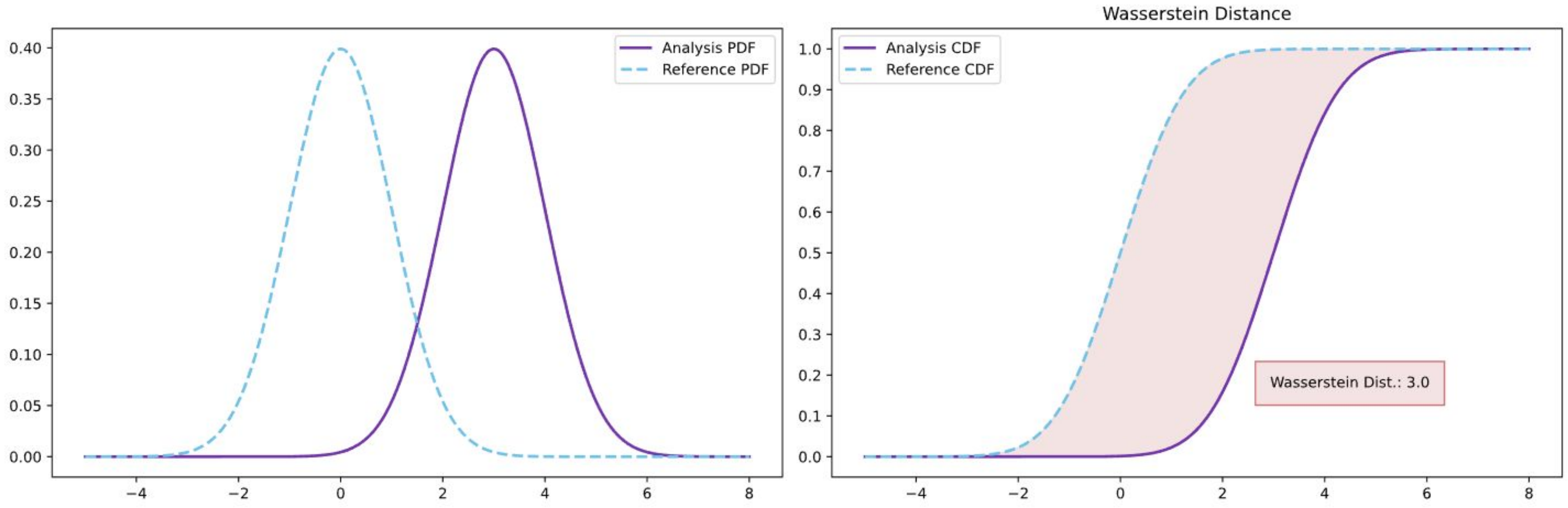
Univariate Drift Detection

Wasserstein distance: Basics

Data Type	Intuition	Range
continuous	Also known as earth mover distance	[0,+inf]

Univariate Drift Detection

Wasserstein distance: Intuition



- Amount of work needed to transform one distribution into the other
- Area between cumulative distribution functions (CDFs)
- Sensitive to outliers

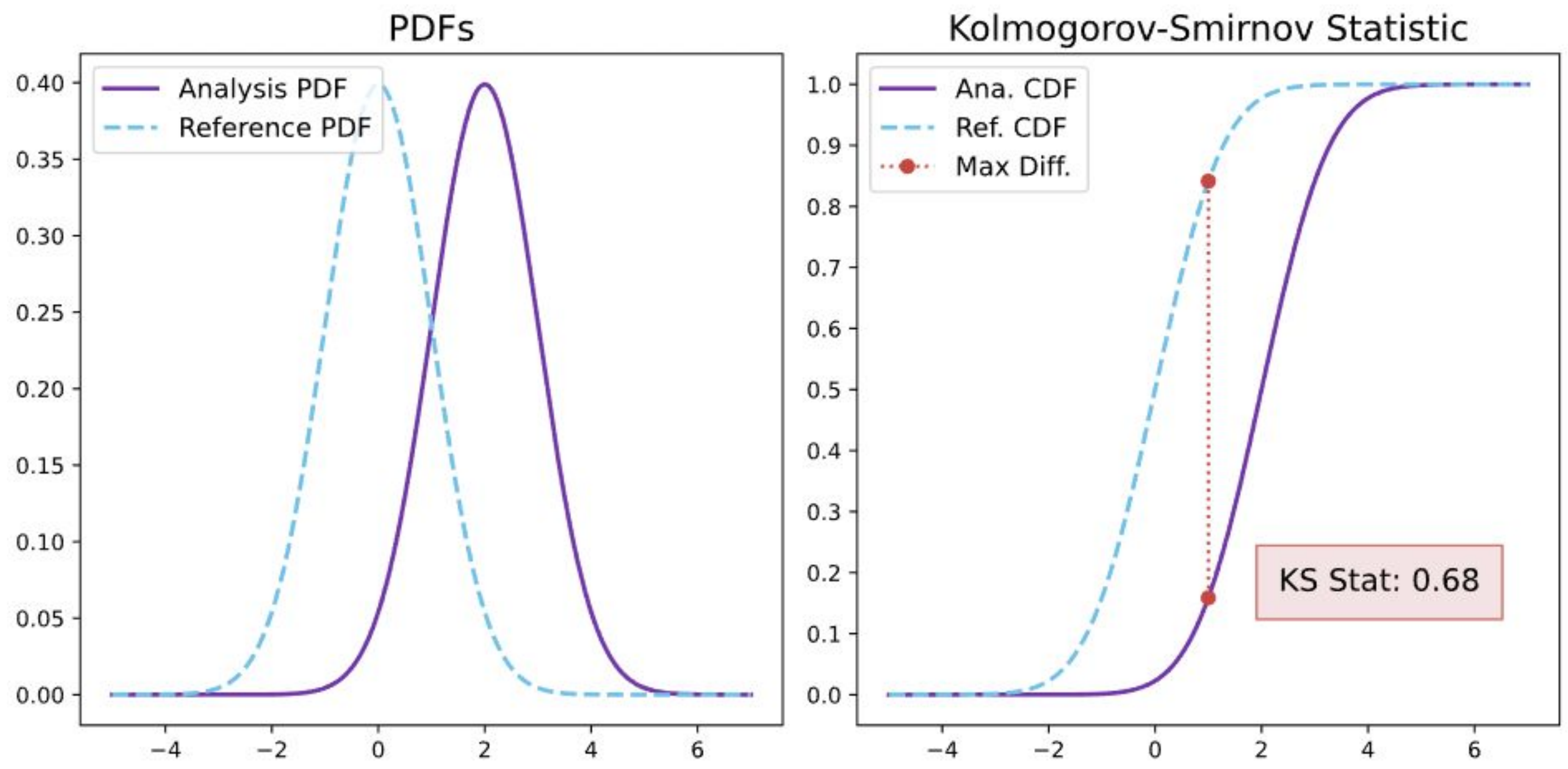
Univariate Drift Detection

Kolmogorov-Smirnov Test: Basics

Data Type	Intuition	Range
continuous	Statistical measure, not a distance	[0,1]

Univariate Drift Detection

Kolmogorov-Smirnov Test: Intuition



- Maximum distance of the cumulative distribution functions (CDFs)
- Prone to false positives, especially in bigger samples
- Outputs d-statistic and p-value

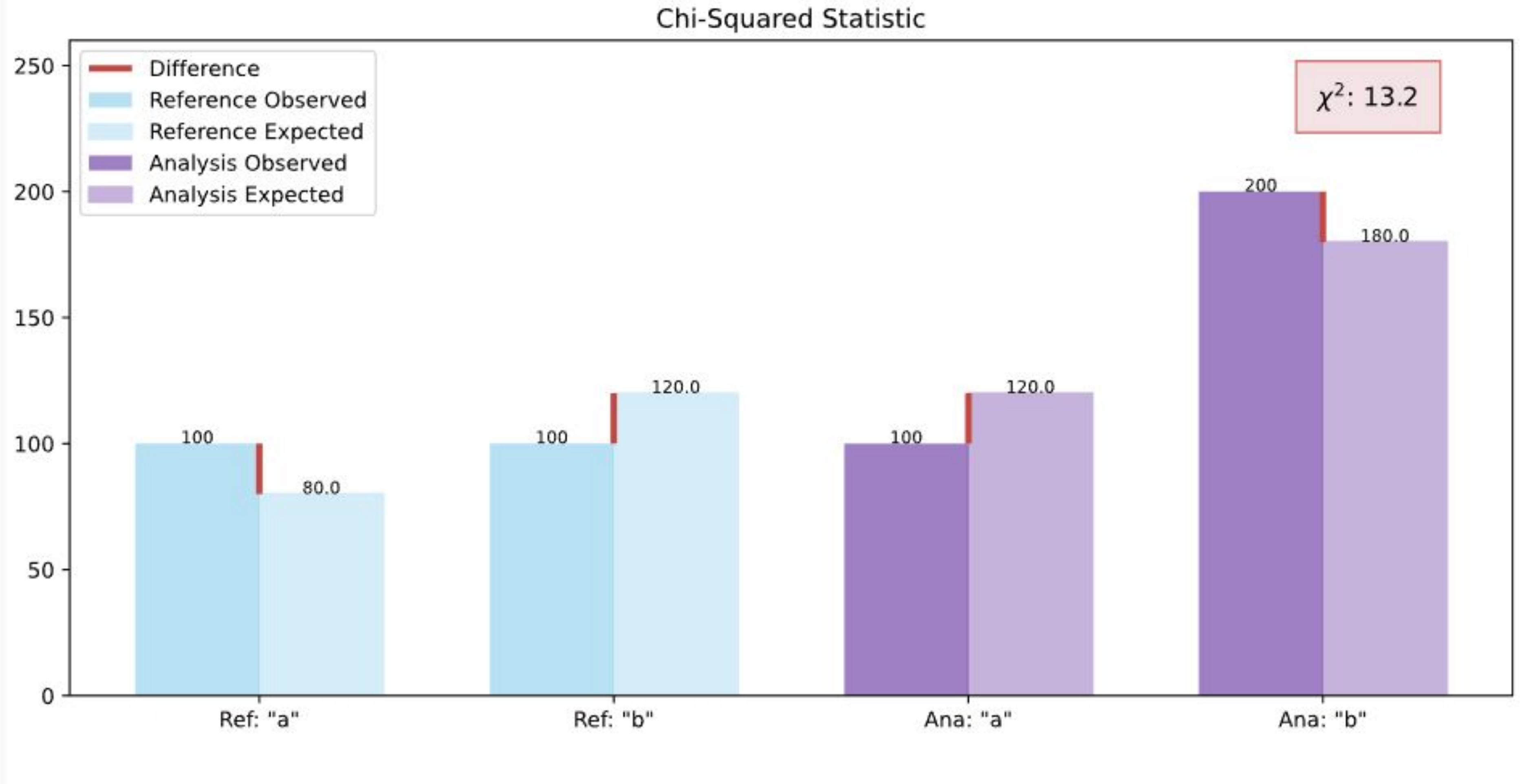
Univariate Drift Detection

Chi-squared Test: Basics

Data Type	Intuition	Range
categorical	Statistical measure, not a distance	[0,+inf]

Univariate Drift Detection

Chi-squared Test: Intuition



- Sample size influences the statistic for the same drift magnitude
- Sensitive to changes in low-frequency categories
- Outputs chi-squared statistic and p-value

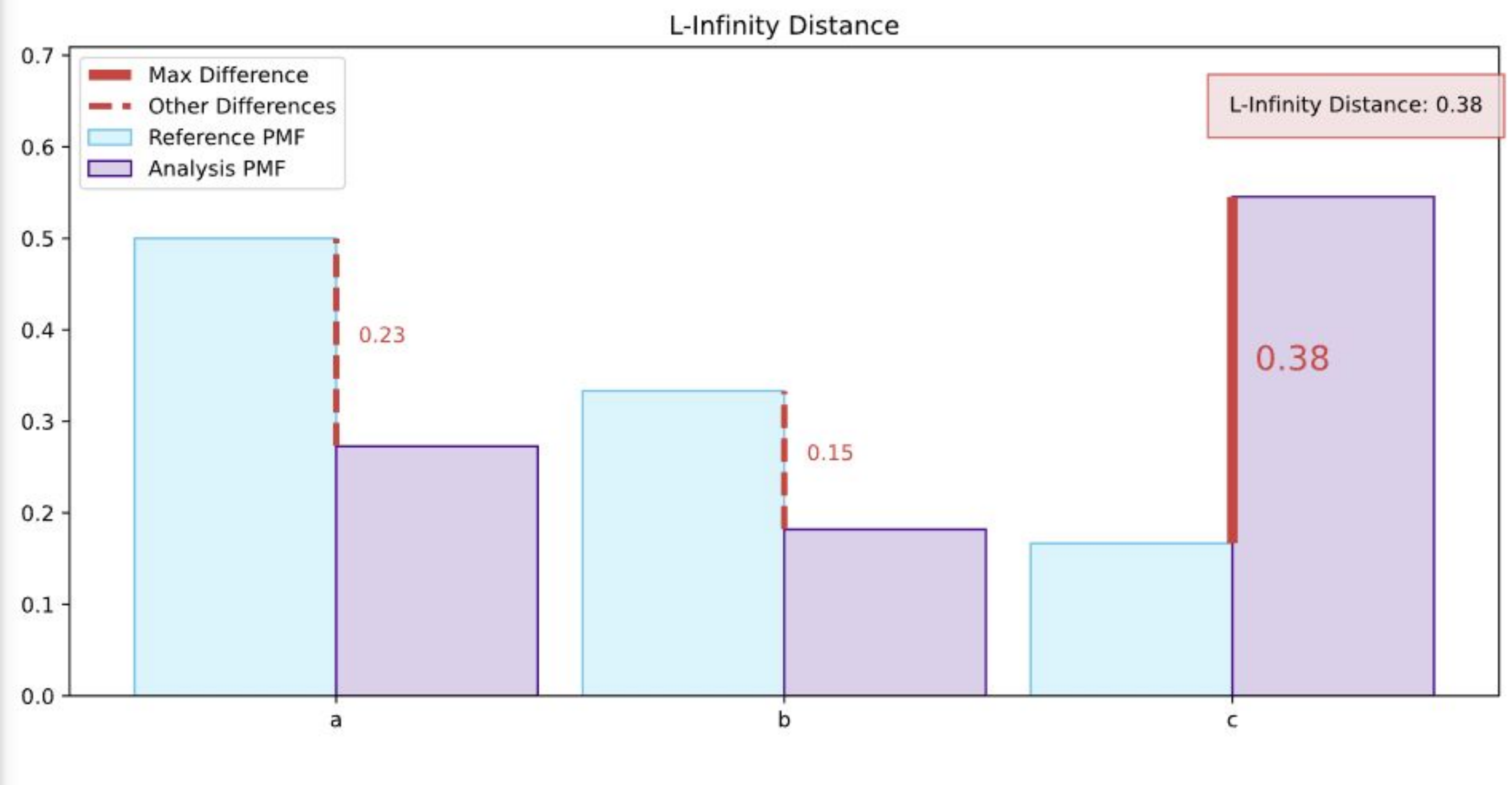
Univariate Drift Detection

L-Infinity Distance: Basics

Data Type	Intuition	Range
categorical	Similar to Euclidean Distance	[0,1]

Univariate Drift Detection

L-Infinity Distance: Intuition



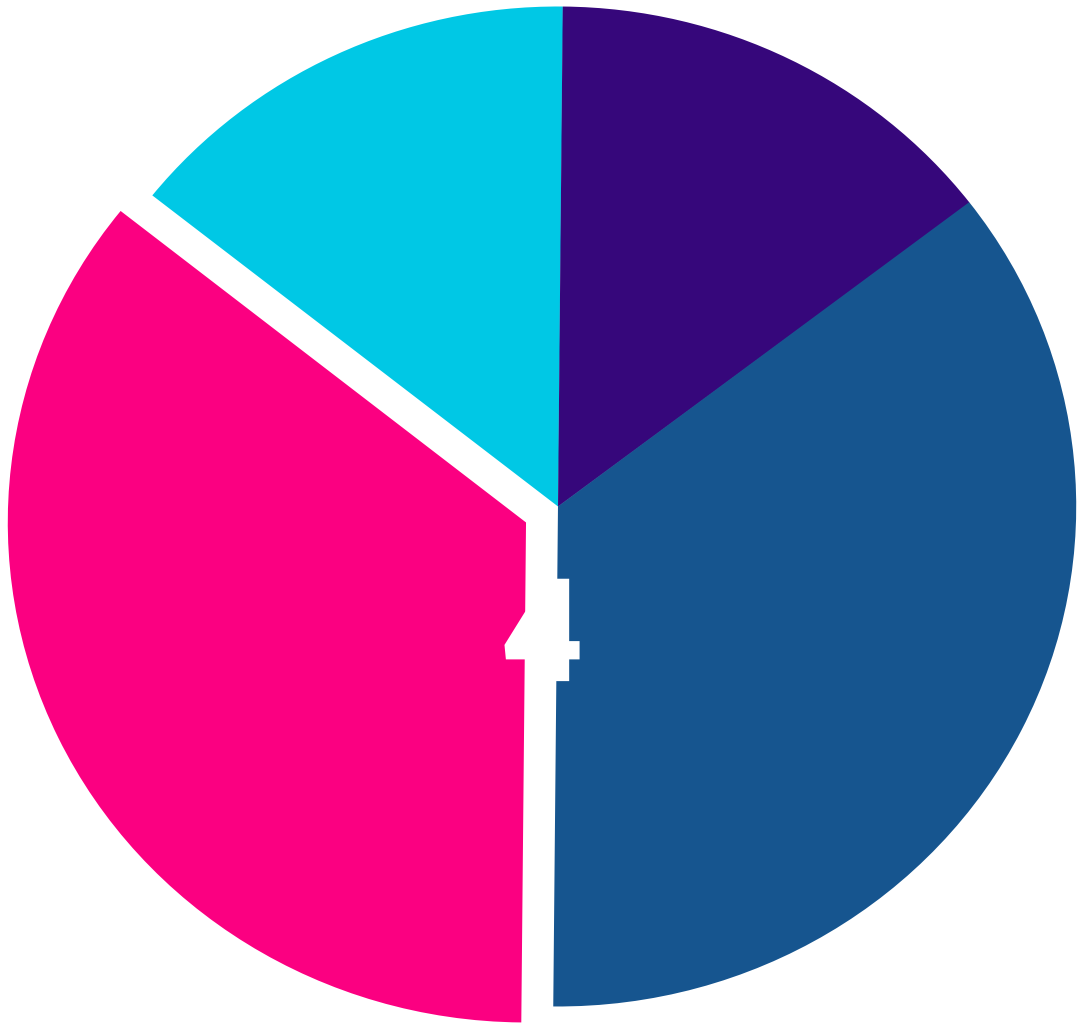
- Maximum of the absolute difference between the relative frequencies
- Robust to noise in features with many categories
- Sensitive to changes to just one category

Univariate Drift Detection

At-a-glance comparison

Name	Categorical	Continuous	Range	Distance	Recommended use	Characteristics
Jensen-Shannon Distance	Yes	Yes	[0,1]	Yes	Versatile	Doesn’t differentiate between very strong and extreme drifts
Hellinger	Yes	Yes	[0,1]	Yes	Medium-strength shifts	Breaks down in extreme shifts
Wasserstein	No	Yes	[0,+inf]	Yes	Work as a shift measure is relevant	Sensitive to outliers
L-Infinity	Yes	No	[0,1]	Yes	Works well with many categories	Sensitive to big changes to one category
Kolmogorov-Smirnov	No	Yes	[0,1]	Statistical Test	Statistical significance	False positives, insensitive to changes in tails
Chi-squared	Yes	No	[0,+inf]	Statistical Test	Statistical significance	False positives, function of sample size

Issue resolution



Issue Resolution

The final step in the monitoring flow

Performance Monitoring → Root Cause Analysis → Issue Resolution

Issue Resolution

5 possible solutions

Retrain
the model

Adjust downstream
processes

Revert to a
previous model

Refactor
the use case

Do
nothing

Issue Resolution

Comparison of methods

	MOSTLY USED	WORKS WHEN	BE MINDFUL
Retrain the model	*Default first step*	There is no concept drift	Will not resolve problems due to covariate shift. Might make matters worse if root cause is data quality issue.
Adjust downstream processes	*To minimize business impact of the performance drop*	Performance degradation cannot be fixed	Generally involves human checks or adjustments to actions done based on predictions
Revert to a previous model	*If you automatically retrain models*	Need quick fix that works well enough	Check if the previous model is still performant
Refactor the use case	*Last-resort, due to cost*	In most cases	Ideally more labelled data should be gathered Might not work if the main cause was covariate shift
Do nothing	*Important to consider it a real option*	Fixing costs higher than leaving the problem as is	Sometimes it's not possible to fix issues due to covariate shift Important to consider it a real option

ML Monitoring Flow

Key Takeaways

- 3-step **monitoring flow** ensures continued performance of your models
- Performance **monitoring and estimation** are crucial and cannot be replaced with just data drift detection
- Data Drift detection is mainly a **root cause analysis** tool
- Retraining won't always fix your problems and can even be a source of an **issue**

Thanks for Listening!

Give NannyML a try (and a star):

https://github.com/NannyML/nannyml