06 ML Monitoring

Observability

Concepts

Metrology

- Temporal accumulation of metrics representing the technical state of a system

Supervision

- Metric giving the state of health of a system at the present moment

Alerting

- Ability to send alerts via multi-channel when a rule is triggered (reaching a specific value, exceeding a threshold, etc.)

Logs

- Formatted application logs describing processing states

Tracing

- Ability to correlate pieces of processing distributed in several technological bricks

KPIs, Dashboards

- Calculation of indicators and display of these metrics in a visual interface

Observability

Different needs

Business

- Purpose
  - Have an aggregated and high-level vision of the state of health of an application chain, regardless of the technical bricks it uses
- Philosophy
  - We monitor the state of a data product, there is as little reference as possible with technical elements (tables, files, jobs, topics, models, etc.)
- For who
  - Data product manager
  - Application manager

Technical

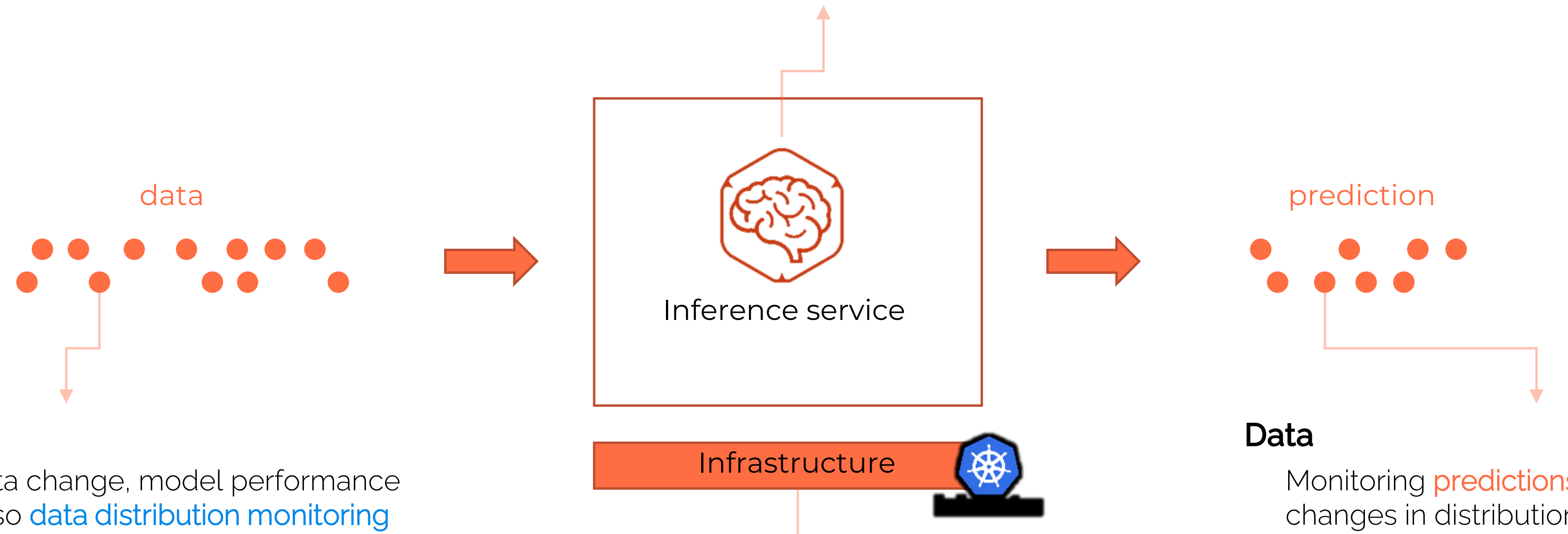
- Purpose
  - Have an exhaustive and precise vision of the state of health of all the technologies behind the data platform
- Philosophy
  - Access to the technical information of the systems, regardless of the data that is handled there
- For who
  - Operations Manager
  - Data manager

ML Observability

What can we track & monitor ?

Model

Follow real accuracy of the model in production or approximations when the ground truth is not available



Data

Usually if input data change, model performance will be impacted, so data distribution monitoring is a great mean to monitor model's performance.

Data

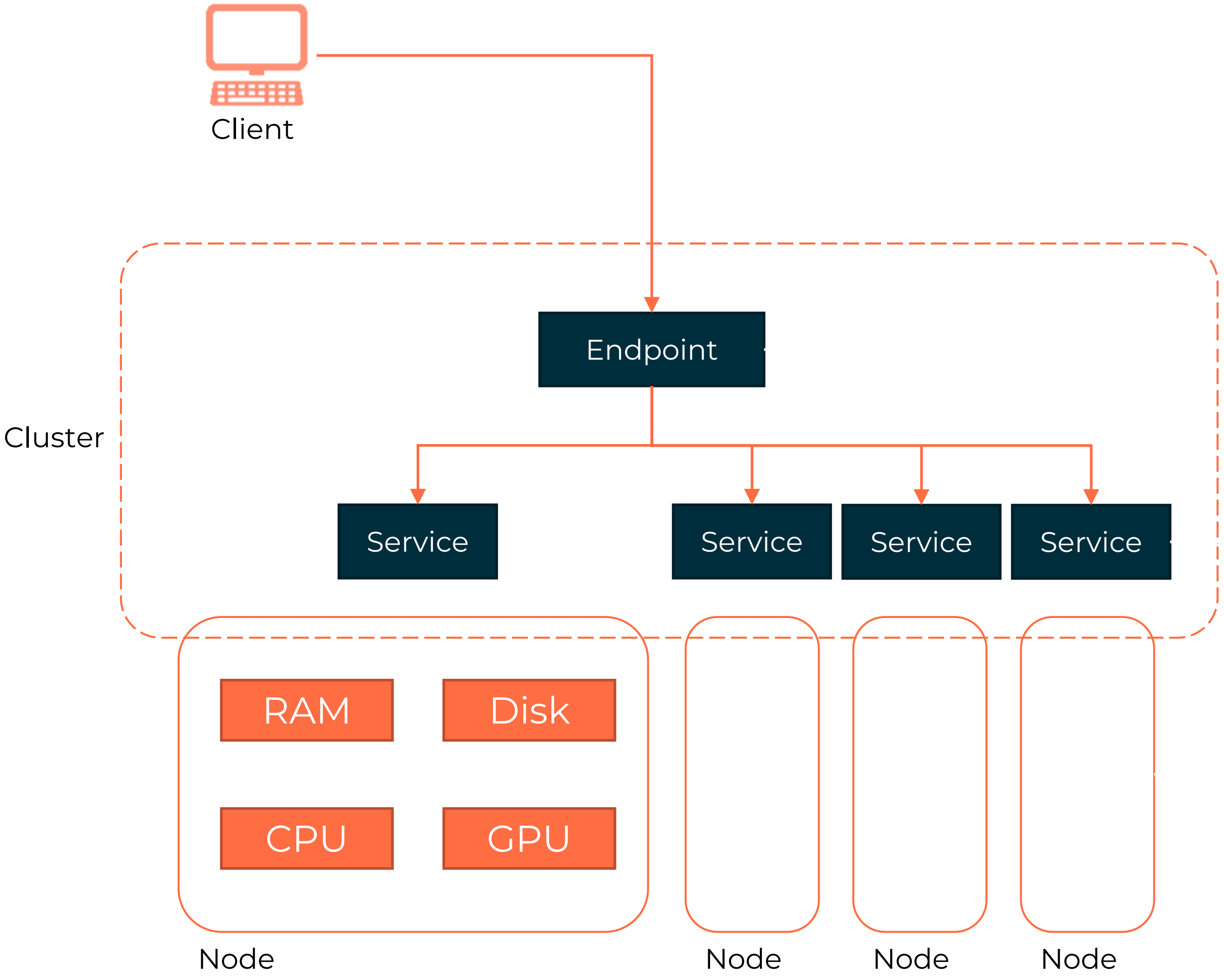
Monitoring predictions and especially changes in distributions can give a clue on model behaviour

System

Technical monitoring of low level hardware (processor, memory) and the upper applicative layers built ontop (number of instances, response time of the service, …) are critical to ensure a good level of service

System Performance

Metrics



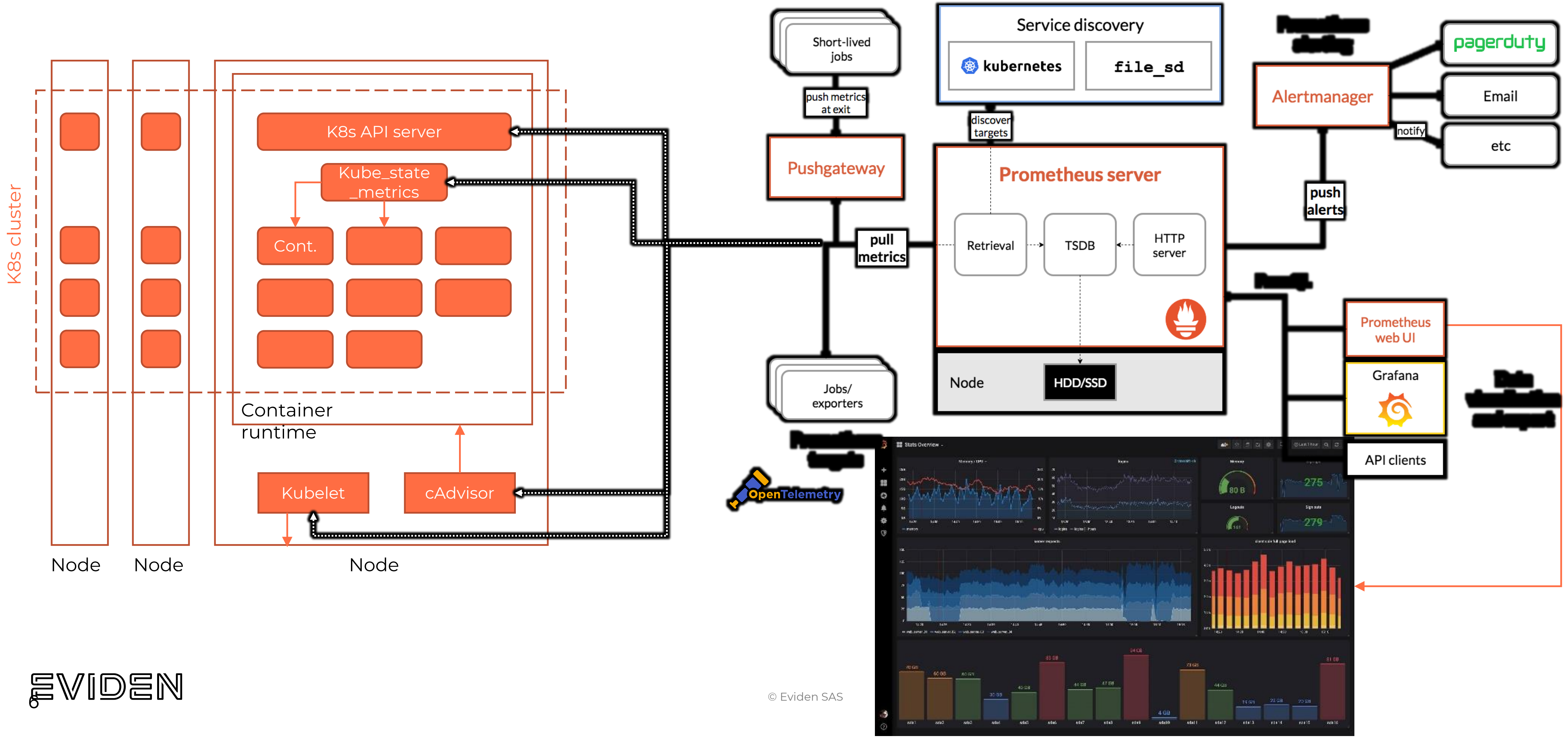
	Example value
Response time : usefull to detect a degradation in service level agreements	5ms
Number of instances : can show a lack of horizontal **scalability**	12
Service RAM usage	255Go
Service CPU usage	34%
Service GPU usage	12%
System RAM usage	4200Go
System Disk usage	88%
System CPU usage	45%
System GPU usage	33%

Performance issues ?

System Performance

Metrics

System Model Data

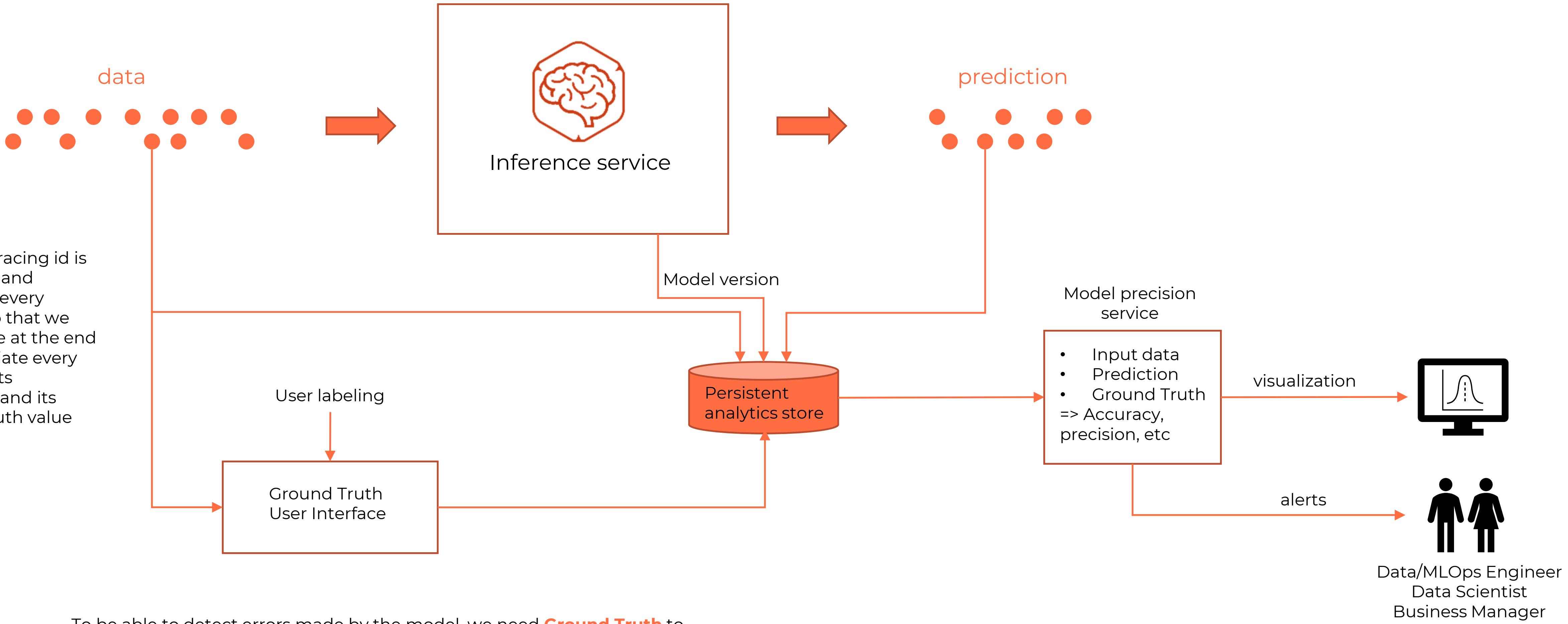


Model Precision

Need for feedback, ground truth

System Model Data

A unique tracing id is generated and forward in every services, so that we can be able at the end to reconciliate every data with its prediction and its Ground Truth value



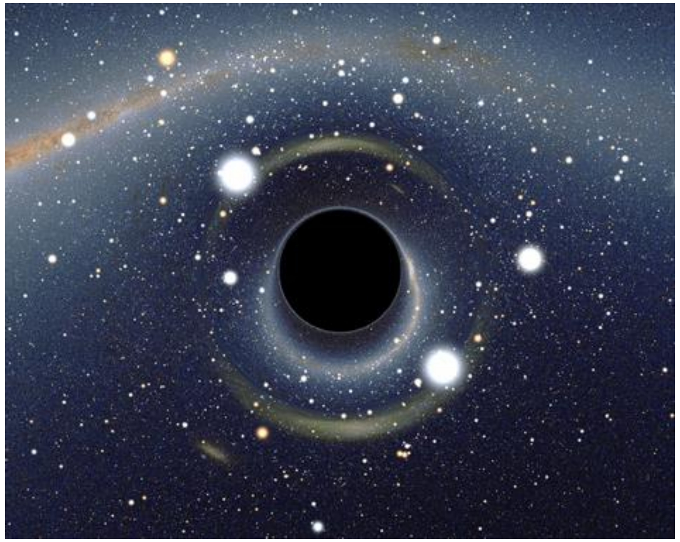
To be able to detect errors made by the model, we need **Ground Truth** to compare it with model predictions.
In most cases, only humans can provide this information (exceptions are for instance with Financial regression, Ad clicking).

Drifts

Proxy Metrics – Alternative when feedback is unavailable

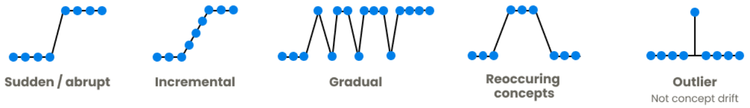
System Model Data

How to “see” a black hole ?



Same technic is used with AI model without Ground Truth access…

- **Ground Truth is often not available** so we use Proxy Metrics to monitor the health of our AI applications : a **Proxy Metric** is a metric that aims to **approximate** or **point out the same information** as another metric that cannot be directly calculated
- Indirect model monitoring is based on the observation of **changes in data distribution** also called **drifts** around the model



- For each type of data, there’s a particular drift
  - Input data => Feature drift
  - Predicted data => Prediction drift
  - Label data => Concept drift

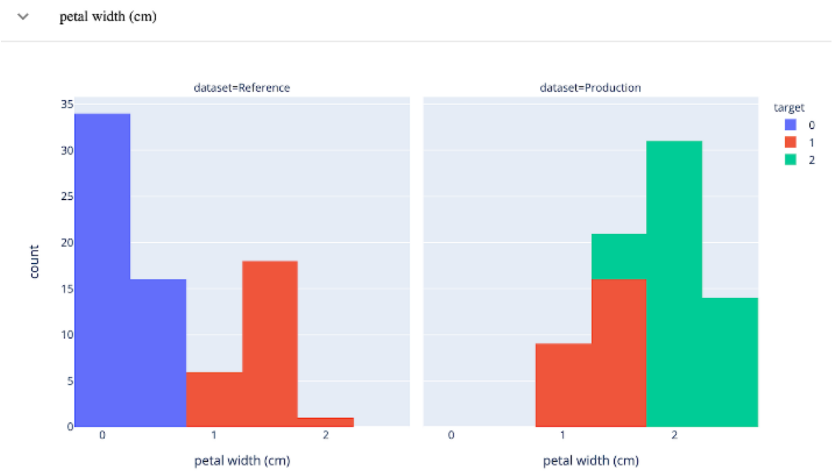
Drifts

Skew vs drift

System Model Data

Skew

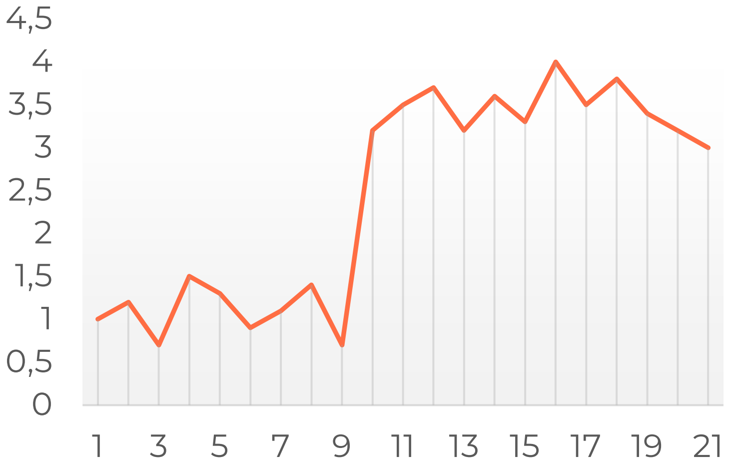
Also know as training-serving skew, refers to differences between training phase (lab, reference) and production. This skew can appear on every types of drifts (data, concept, prediction)



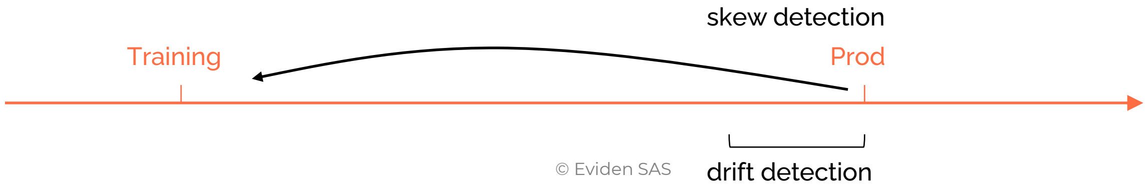
Example of training-serving skew

Drift

Change in distribution of a data (input data, Ground Truth, prediction, feature importance, ...) over time.



Example of data drift



Feature Drift

Focus on Input data

Synonyms

Data drift

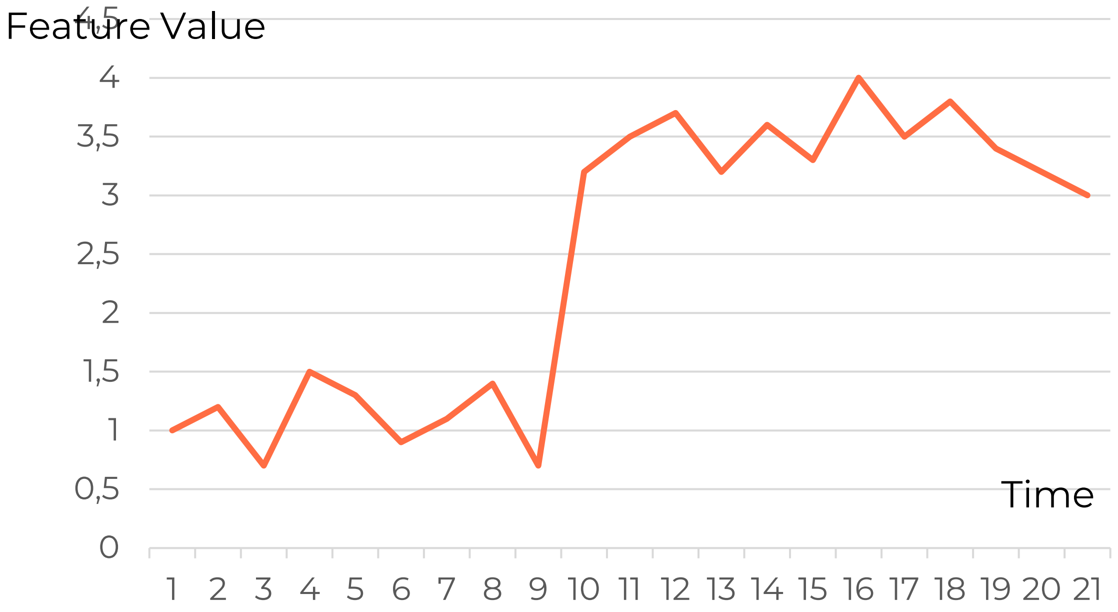
Input drift

Covariate shift

Feature drift

System Model Data

Example of a distribution change (**Feature Drift**)



Lots of algorithms exist for drift detection, ex :
https://github.com/SeldonIO/alibi-detect#drift-detection

Definition

Feature Drift occurs when one or multiple features from the **input data** progressively or suddenly go through **significant distribution changes**

Origins

- **environmental changes**
- **acquisition method shifts**
- pipeline modifications or errors

Example

A supermarket wants to predict its client loyalty based on the price they pay when they come. We will have a feature drift if the supermarket's prices go up or down (environmental change) or if the VAT suddenly is being counted in the total price (acquisition change)

How to detect

- **monitor** input data
- use of **drift detectors** (ex Kolmogorov-Smirnov test)

What to do in case of drift

- **Understand** why data has drifted : could be only errors in data ingestion, no need for retrain
- If really needed, **retrain**
- Use models less affected by feature drift such as random forest or gradient-boosting model

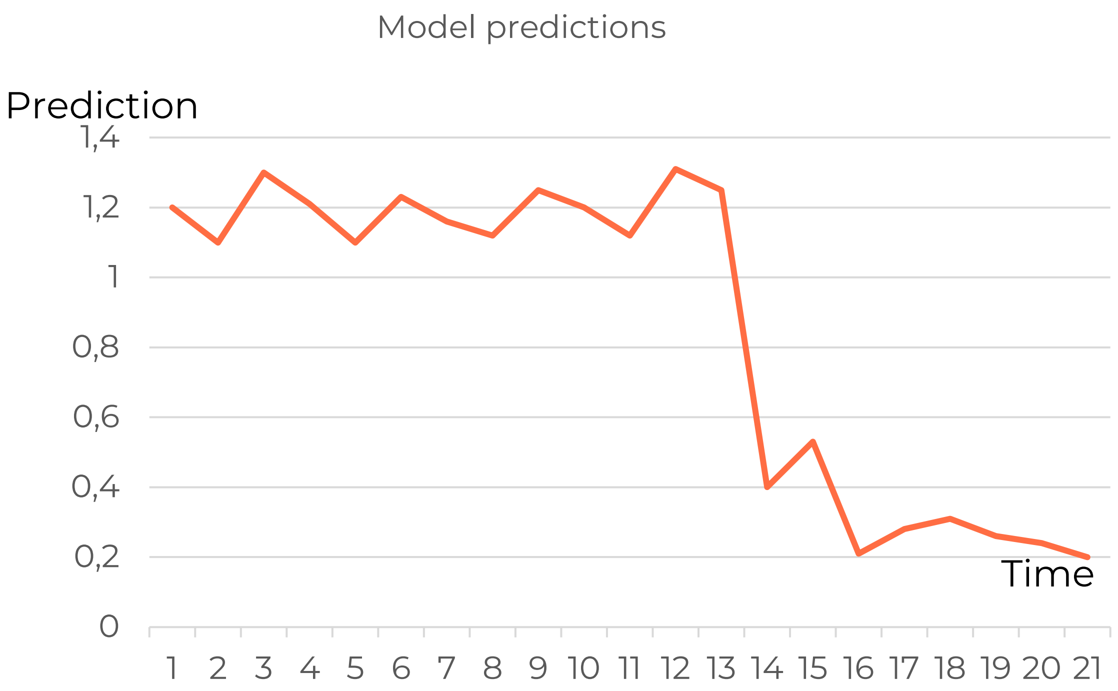
Prediction Drift

Focus on predictions

Synonyms

Model drift   Output drift

System   Model   Data



Definition

Prediction Drift occurs when the **output of the model** progressively or suddenly goes through **significant distribution changes**

Origins

- **Same** than for feature drift
- **Bad retrain** of the model

How to detect

- **monitor** output data (predictions)
- use of **drift detectors** (ex: Page-Hinkley test) like for input data

What to do in case of drift

- Analyze input data to see if there's also a feature drift
- Retrain model with fresh data

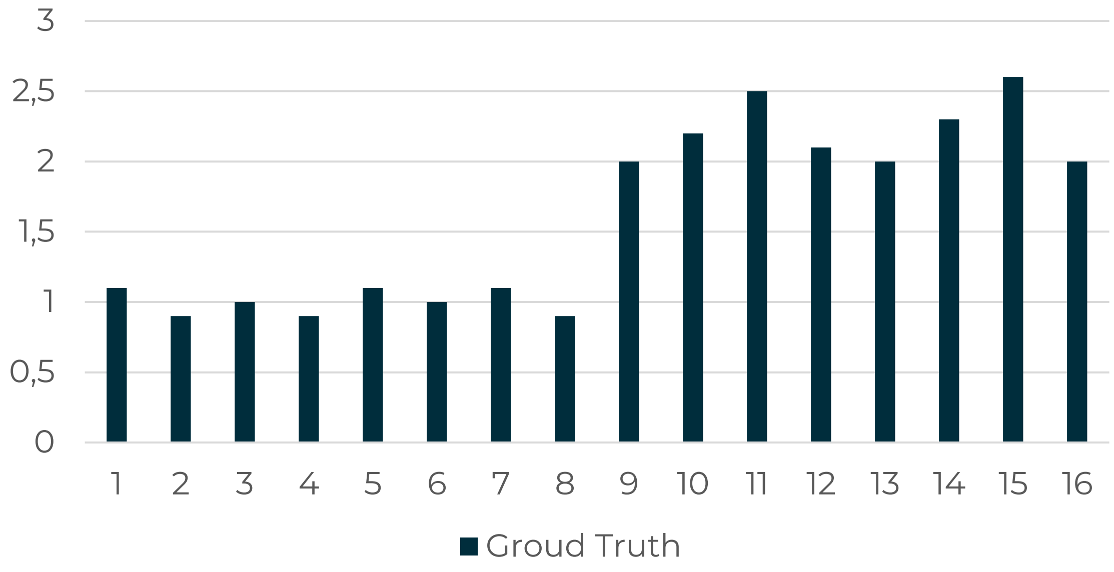
Concept Drift

Focus on Ground Truth

Synonyms
Ground Truth drift
Target drift
Label drift
Annotation drift

System Model Data

Example of a **Target Drift**



Definition
Concept Drift occurs when the **ground truth** progressively or suddenly goes through **significant distribution changes**

Origins
- Change in **human judgement** annotating GT
- Introduction of **new categories**, merging, splitting of existing ones
- **Environmental** changes

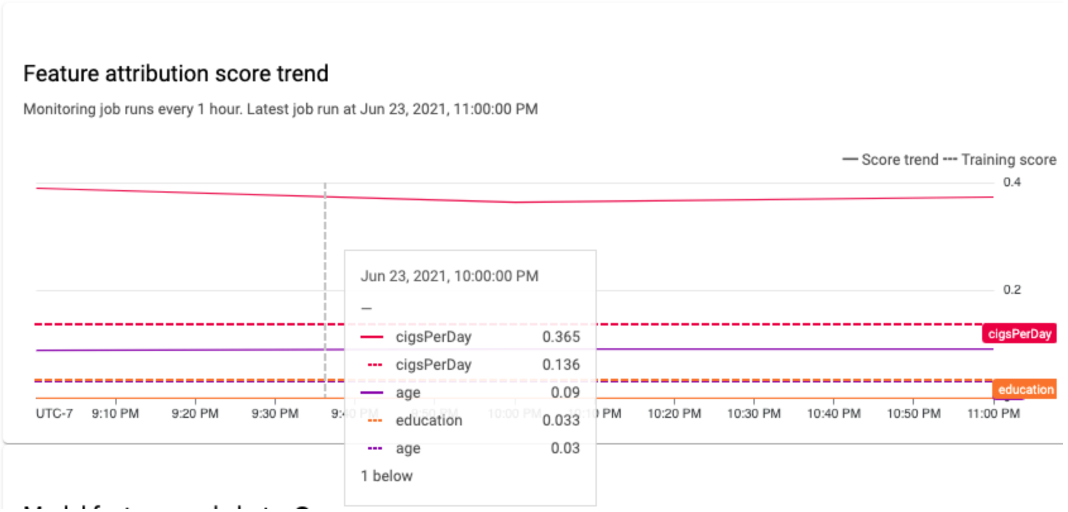
Example
Customer sentiment analysis based on customer reviews. If the criteria for determining whether a review is positive or negative changes, then it causes label drift.

How to detect
- **monitor** ground truth (labels)
- use of **drift detectors** like for input data

What to do in case of drift
- Retrain if needed
- Use active or self-learning algorithms

Explainability related drift

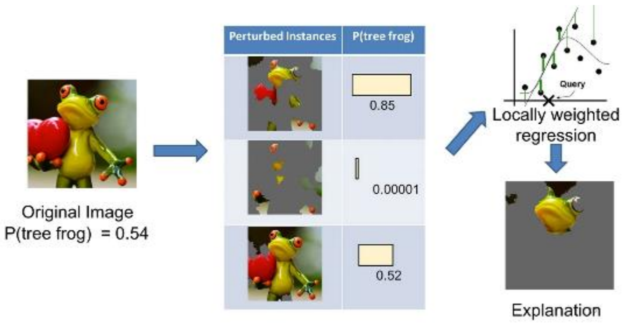
Feature attribution drift

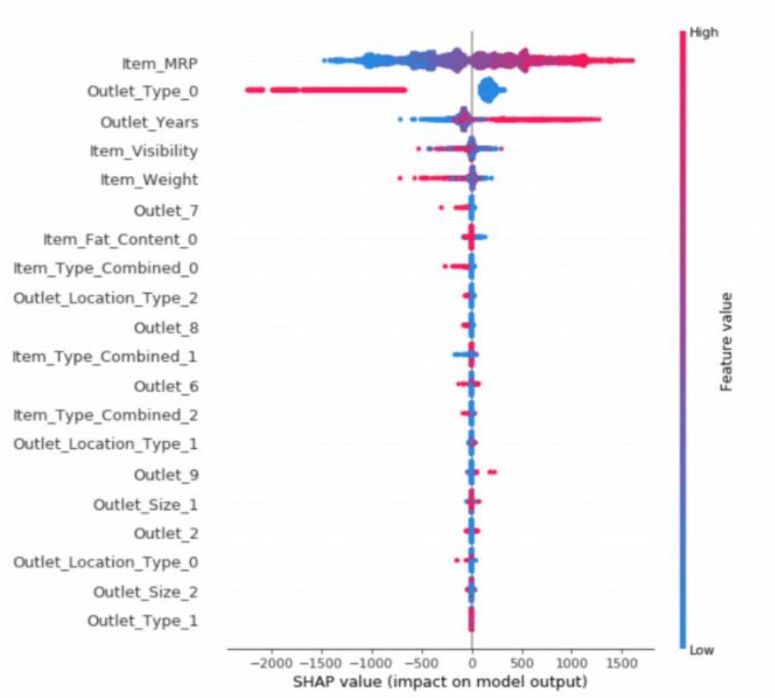


Definition

Feature Attribution is the measurement of the importance of the input features in the decision process of the model to produce the prediction.

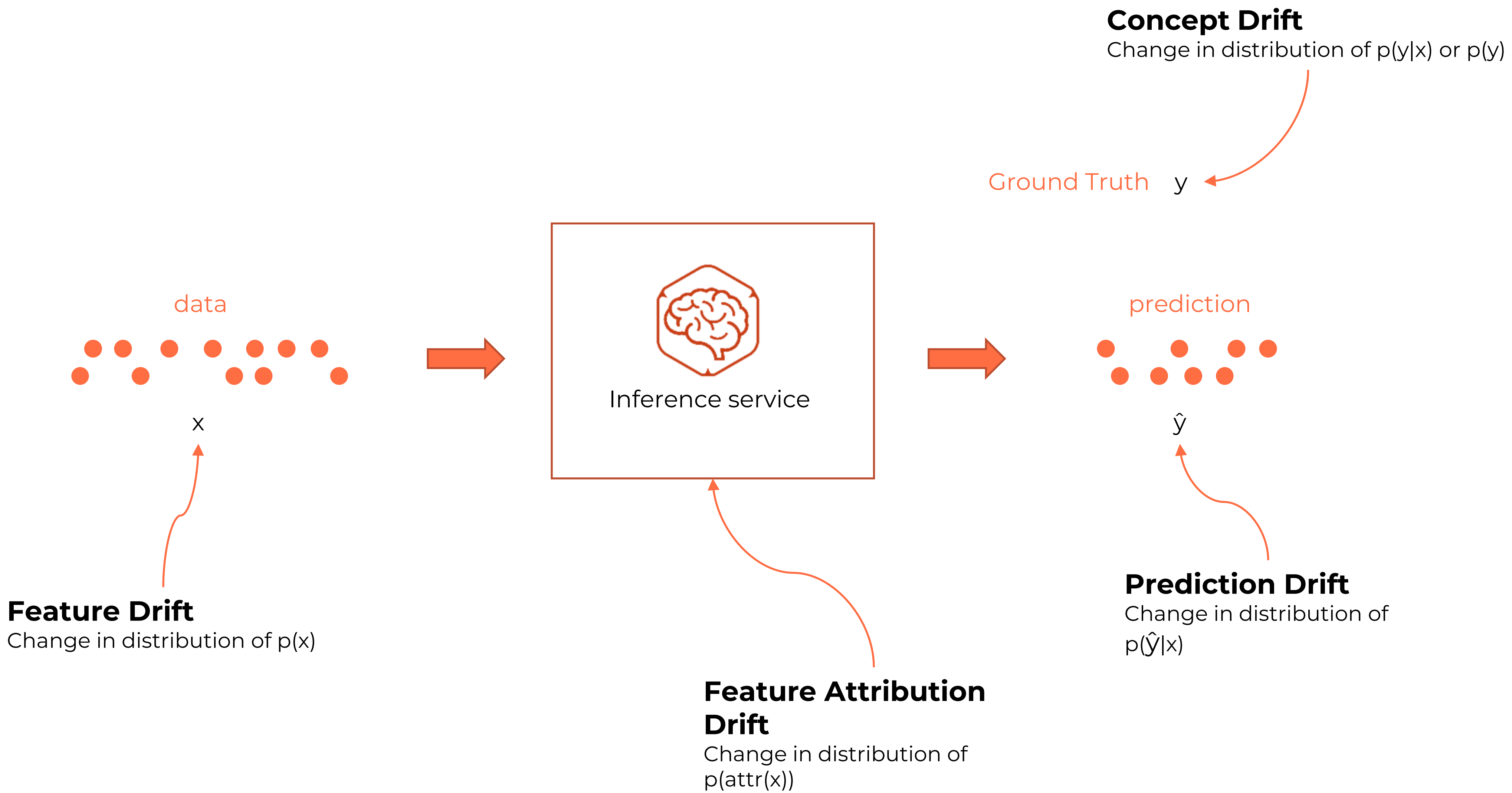
Feature Attribution Drift occurs when the **feature attribution** progressively or suddenly goes through **significant distribution changes**





Shapley values

Sumup drift


Concept Drift
Change in distribution of p(y|x) or p(y)
Ground Truth
y
data
x
Inference service
prediction
ŷ
Feature Drift
Change in distribution of p(x)
Feature Attribution Drift
Change in distribution of p(attr(x))
Prediction Drift
Change in distribution of p(ŷ|x)

Do we really need to retrain ?

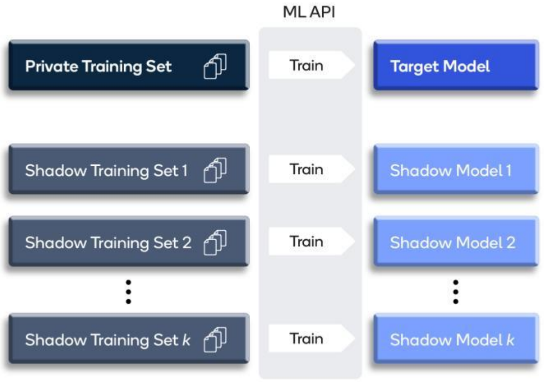
Retraining is not a golden bullet

Feature Drift	Yes	Yes	Yes	No
Prediction Drift	Yes	No	Yes	No
Concept Drift			No	Yes
Need Retrain ?			No	Yes

Need further analysis with business teams to obtain deeper indirect feedback and evaluate the model performance

A drift in data is not always the sign that we should retrain the model.

On the other hand, automatic retrain can be a good practice, but only in a shadow ML strategy : continuously producing candidate models that challenge the one running



Drift detectors

Tooling

Feature available ✓

In development ●

	During Training		During Serving	
	Feature, prediction concept drift	Feature attribution	Feature, prediction concept drift	Feature attribution drift
ALIBI EXPLAIN		✓		✓
ALIBI DETECT	✓		✓	
EVIDENTLY AI	✓		●	
deepchecks.	✓			
Vertex AI based on TensorFlow	✓	✓	✓	✓

Quizz

What we've learned

Question				
Metrology is the storage and analysis of application logs over time	Y	N		
Hardware metrics are used for business observability	Y	N		
Interesting performance metric for ML monitoring is ML service response time	Y	N		
Model monitoring is available without ground truth	Y	N		
Feature drift could be due to errors in ingestion pipeline	Y	N		
Prediction drift can be detected with same methods than feature drift	Y	N		
Concept drift appears when relation between input and ground truth change over time	Y	N		

Quizz

What we've learned

Question				
Metrology is the storage and analysis of application logs over time	Y	N		
Hardware metrics are used for business observability	Y	N		
Interesting performance metric for ML monitoring is ML service response time	Y	N		
Model monitoring is available without ground truth	Y	N		
Feature drift could be due to errors in ingestion pipeline	Y	N		
Prediction drift can be detected with same methods than feature drift	Y	N		
Concept drift appears when relation between input and ground truth change over time	Y	N		

Metrology = metrics

In Practice

Lab Content

- Exo1 Deploy a model and explore **real time monitoring**
  - Create an inference service with a model
  - View simple monitoring embeded into kubeflow
  - Look for more metrics with grafana dashboards
- Exo2 Monitor **drifts**
  - Create a drift by introducing a new category inside your input data
  - Redeploy the model with a stronger inference graph : feature and target drift detectors
  - Observe the drift in target and features
- Exo3 **Fix** the drift
  - Retrain the model taking into account the new category, deploy, and make sure all drifts are gone

EVIDEN