Unlocking Data Insights - Introduction to Data-Centric AI

Data-centric Explainable AI (DCXAI)

Funded by the European Union NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
FAIR Future Artificial Intelligence Research

UniBa UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO
dib
KDDE

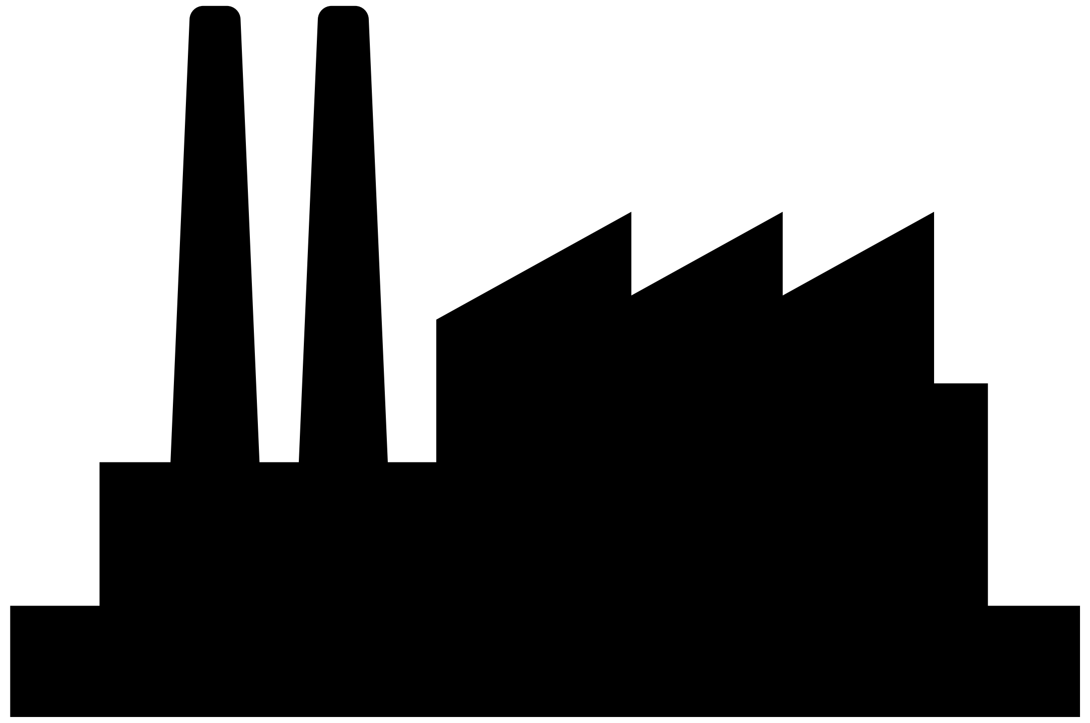
Reference: Applied Machine Learning Explainability Techniques - Aditya Bhattacharya

Data-centric Explainable AI (DCXAI)
Executive summary

- Introduction to DCXAI
- Thorough data analysis and profiling process
- Monitoring and anticipating drifts
- Checking adversarial robustness

credits: Rohan Makhecha - Unsplash

What’s problem



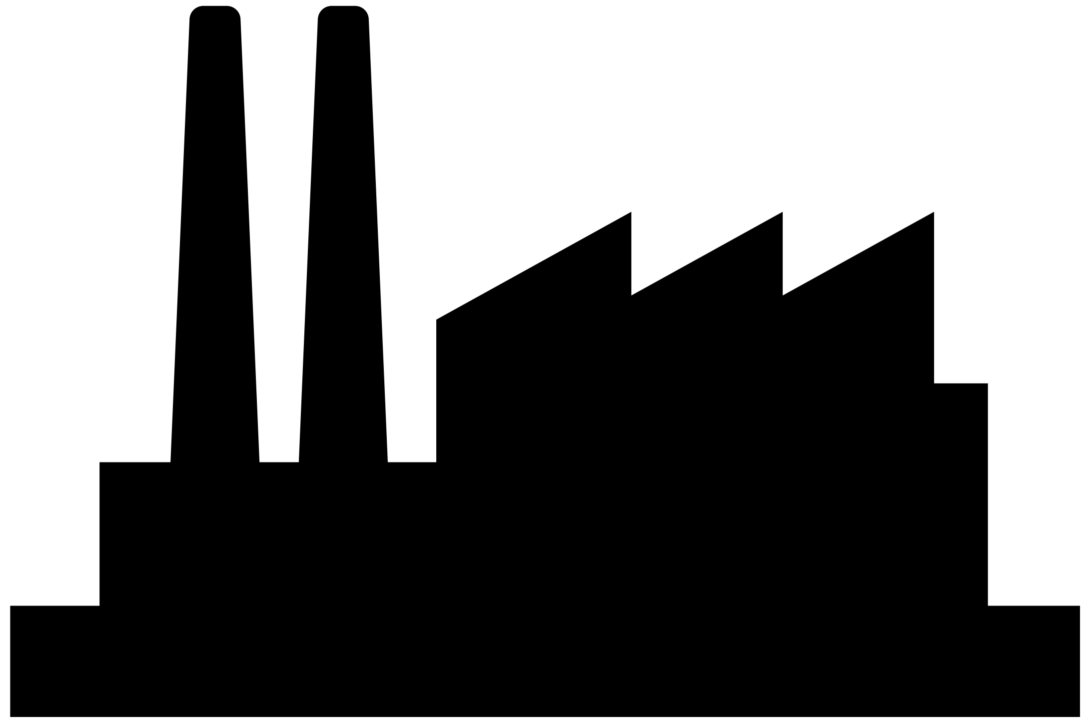
ML production model

are not aligned

DCAI

Principles

What’s problem



ML production
Models

are not aligned

DCAI

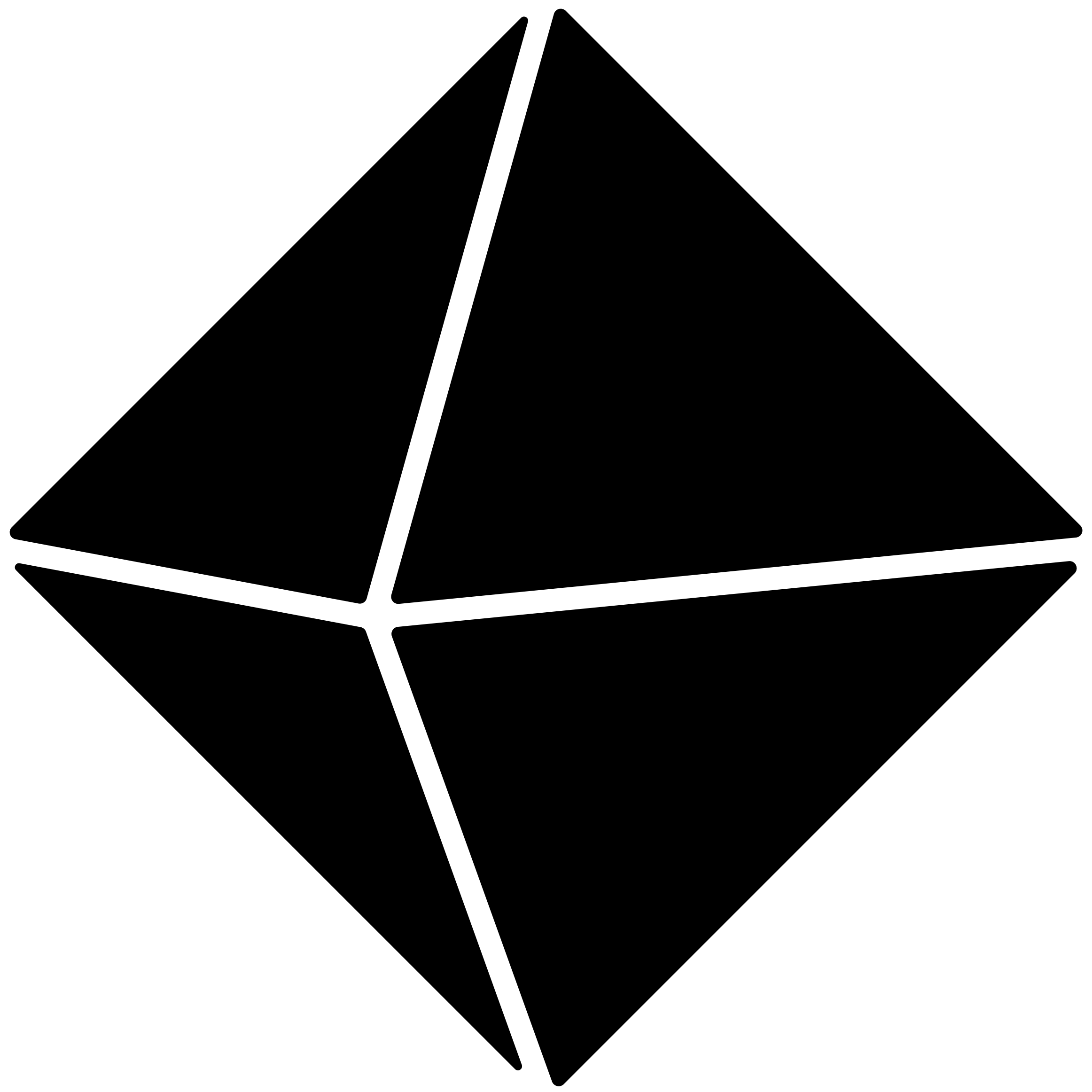
Principles

↓

Data Centric eXplainable AI

Data Centric eXplainable AI

Data properties



Volume

Consistency

Purity

Analyzing data volume

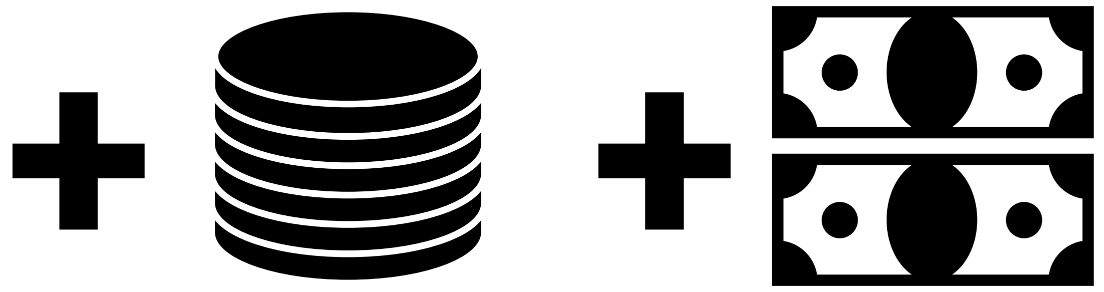
The classical problem of ML

Algorithms

OVERFITTING

Is the model trained on sufficient data?

But for companies



How do we find out if the model was trained on sufficient data?

Analyzing data consistency

Don't forget to understand the

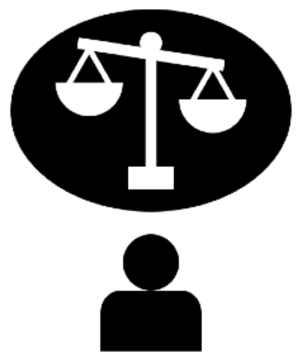
Data Distribution

Data

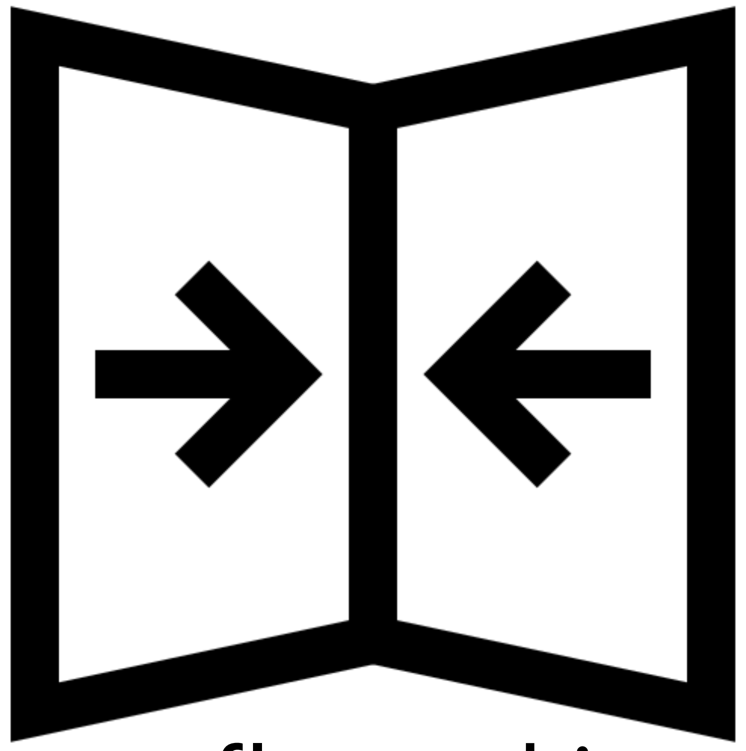
Is skewed toward a particular direction

is not evenly distributed

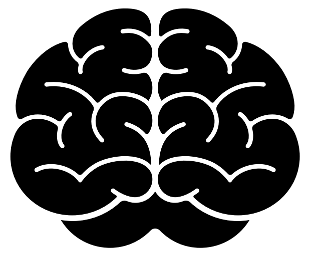
Class imbalance



Bias



reflected in



Model

Analyzing data consistency

For production system

Data observed in inference time have some variance with training data

→ **Data drift**

↗ Change in data structure

↘ Change in statistical data properties

Analyzing data consistency

Data drift
Change in data structure

Whole dataset

One o more features

Population Stability Index **(PSI)**

Kullback-Leibler Divergence **(KL Divergence)**

Wasserstein metric **(Earth Movers Distance)**

Statistical methods
Measure the distance between two data distribution

Data consistency *is an important parameter for* ***root cause analysis*** *inspection when interpreting black box models*

Analyzing data purity

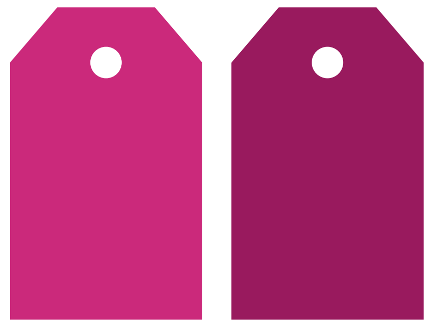
Real data are very often noisy, but …

What if a black-box ML model is trained on a dataset with less purity and, hence, perform poorly?

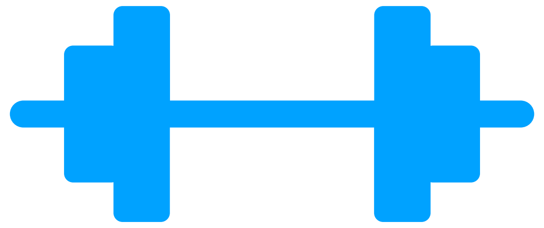
Analyzing data purity

Most common integrity issues

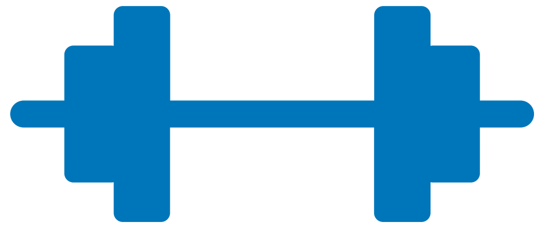
Label ambiguity



Dominant Features Frequency Change (DFCC)



Train

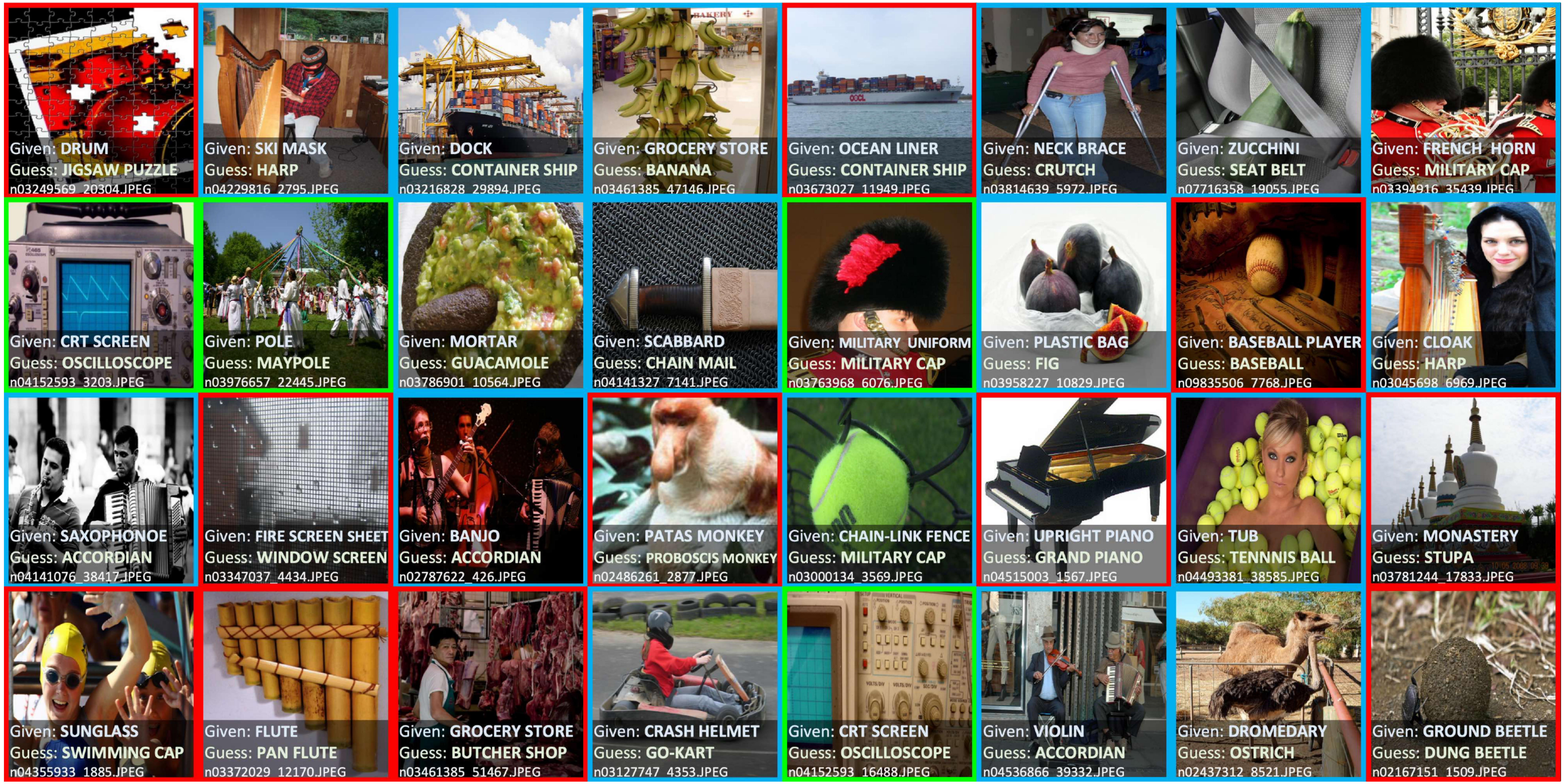


Test

Other data purity issues:

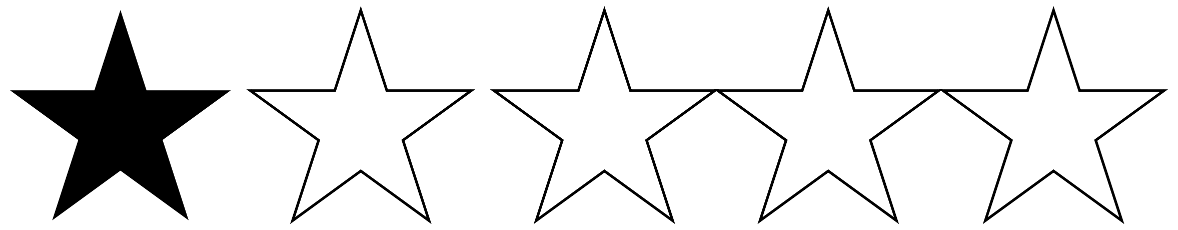
New label, new feature category or out of bound values (anomalies) for particular feature in inference set

Errors, errors everywhere



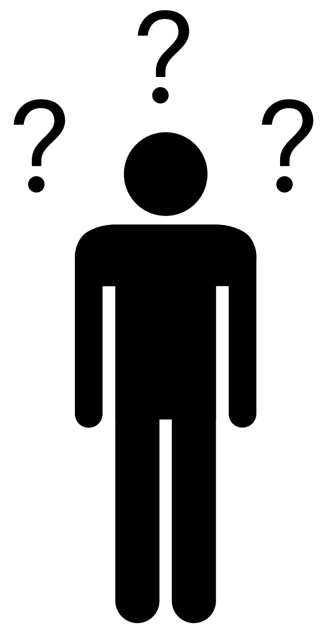
Top 32 label issues in the 2012 ILSVRC ImageNet train set. Label Errors are boxed in **red.**

Sources of Noisy Labels

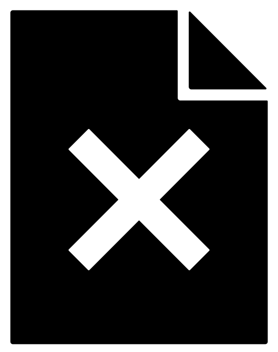


Clicked the wrong button

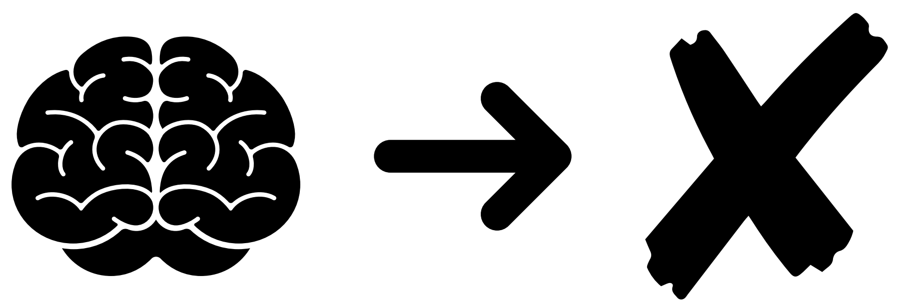
Mismeasurement



Incompetence



Mistakes



Another ML model's bad predictions

Nice tool for your data

deepchecks.

Status	Check	Condition	More Info
✖	Single Value in Column	Does not contain only a single value	Found 1 out of 14 columns with a single value: ['Is Ripe']
✖	Feature-Feature Correlation	Not more than 0 pairs are correlated above 0.9	Correlation is greater than 0.9 for pairs [('4046', 'Total Volume'), ('4225', 'Total Volume'), ('Total Bags', 'Total Volume'), ('Small Bags', 'Total Volume'), ('Small Bags', 'Total Bags')]
✖	Identifier Label Correlation	Identifier columns PPS is less or equal to 0	Found 1 out of 1 columns with PPS above threshold: {'Date': '0.03'}
!	String Mismatch	No string variants	Found 1 out of 2 columns with amount of variants above threshold: {'type': ['organic']}
!	Data Duplicates	Duplicate data ratio is less or equal to 5%	Found 13.5% duplicate data

Example of Data integrity checks using the Deepchecks framework

Source: https://docs.deepchecks.com/stable/tabular/auto_tutorials/quickstarts/plot_quick_data_integrity.html#sphx-glr-tabular-auto-tutorials-quickstarts-plot-quick-data-integrity-py

Thorough data analysis and profiling process

Let's assume we have a baseline trained ML model

But it is not meeting the benchmark accuracy

Model-Centric AI

Hyperparameter tuning, complex model, etc.

Data-Centric AI

Data augmentation, Data profiles, Adversarial robustness

Thorough data analysis and profiling process

Building robust data profiles

A Statistical Data Profile of dataset is a collection of certain statistical measure of its feature values segmented by the target variable class

Thorough data analysis and profiling process

Building robust data profiles

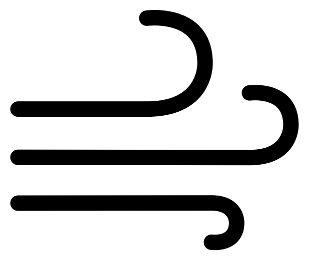
Class	Mean_feat1	Median_feat1	AvgVar_feat1	Mean_feat2	Median_feat2	AvgVar_feat2
0	34.5	37.0	-3.5	128.0	103.5	4.0
1	23.8	23.8	7.8	73.9	102.8	2.2
2	49.0	40	-2.3	101.5	101.5	-1.8

We can create the statistical profiles for validation and test set

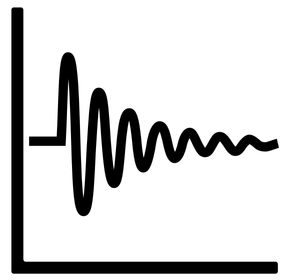
If the absolute percentage change between the value significantly higher (say, >20%), then this indicates the presence of data drift

Monitoring and anticipating drifts

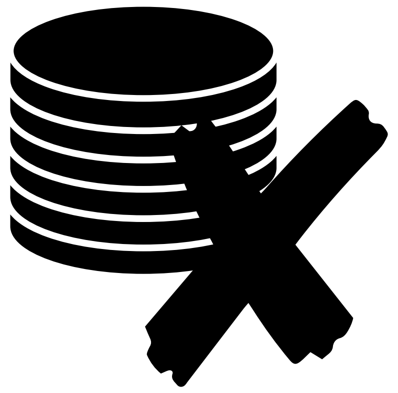
Data consistency for real-time systems is a challenging problem



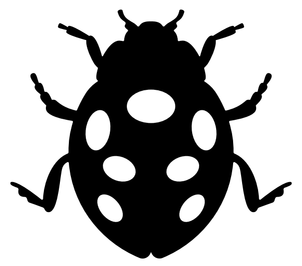
Change in external environmental conditions



The natural wear and tear of sensors



Physical damage caused to the system collecting the data



Bug in the software program that process data

Thorough data analysis and profiling process

Detecting drifts

- What is the best way to identify the presence of a drift?
- What happens when we detect the presence of a drift?

Solution

Comparing correlations of the feature with the target outcome

Thorough data analysis and profiling process

Selection of statistical measures

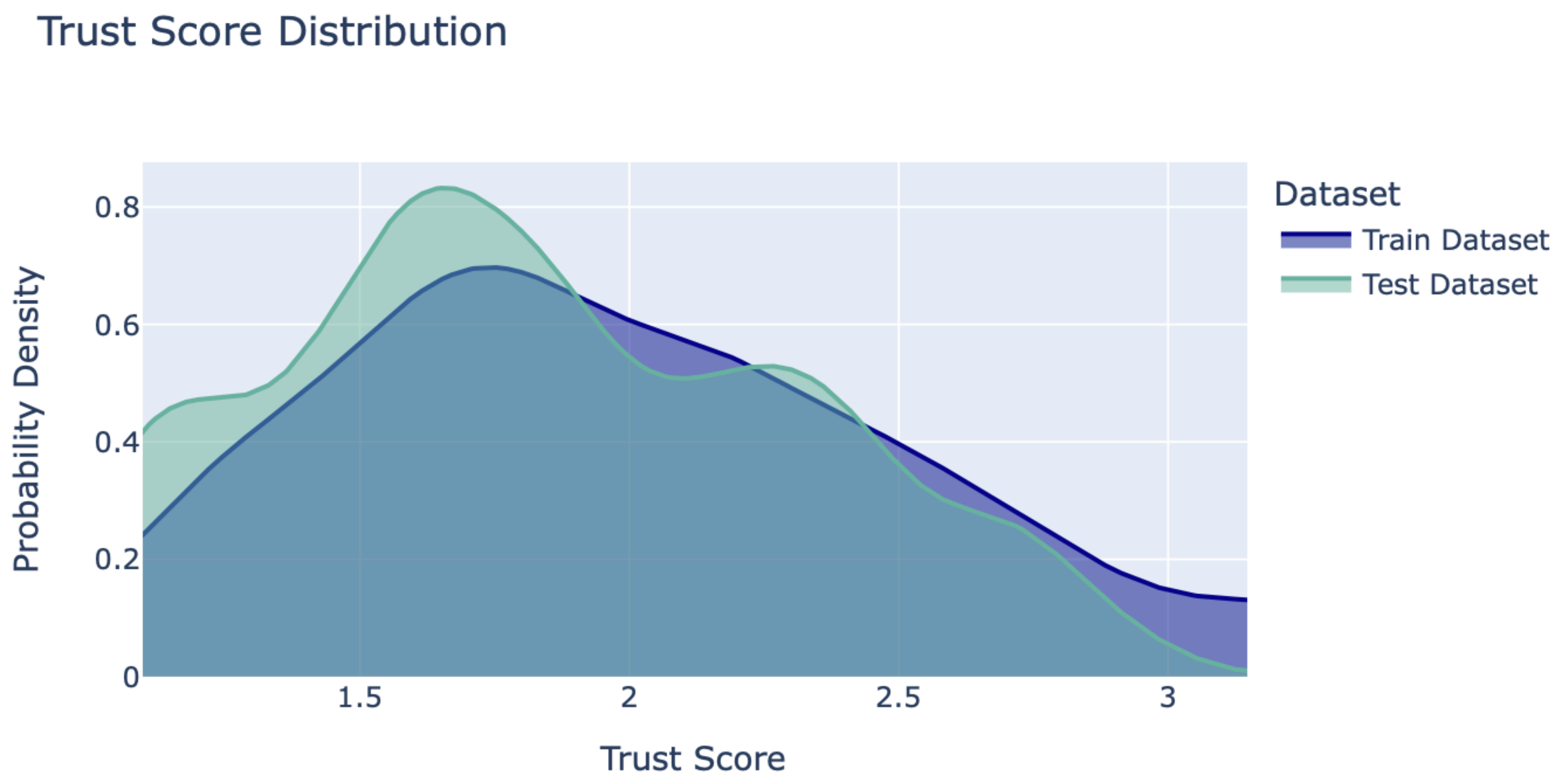
How to quantify the drift?

Popular distribution metrics to detect presence of data drift using a quantitative approach

Trust Score Distribution (TSD)
Population Stability Index (PSI)
Predictive Power Score (PPS)
...

Thorough data analysis and profiling process

Trust Score Distribution



An Example of the Trust Score Distribution between the training dataset and the inference dataset

Thorough data analysis and profiling process

Trust Score Distribution

The Trust Score is a distribution metric used to measure the agreement between the ML classifier on the training set and an updated k-Nearest Neighbor (kNN) classifier on the inference data

Thorough data analysis and profiling process

Trust Score Distribution

=

Ideally

Same distribution for both train and test set

←

Left

The trained model has less confidence in the inference data **(drift?)**

→

Right

High probability of the data leakage

Thorough data analysis and profiling process

Data leakage

Incorrect Preprocessing: Data preprocessing must be done separately for training and test sets. For example, calculating the mean or variance on the entire dataset (training + test) and then using these values to normalize the data can introduce leakage.

Target Leakage: This happens when the variables to be predicted are present, in some form, in the input variables. For instance, if predicting the risk of a customer defaulting on a loan includes information on missed payments recorded afterwards, it can lead to leakage.

Thorough data analysis and profiling process

Data leakage

Future Information: If data that will only be available in the future (e.g., future outcomes) is used during model training, the model can learn from information it wouldn't have access to when making predictions.

Shared Data Between Training and Test Sets: If the training and test sets are not properly separated, some information from the test set might influence the model during training.

Thorough data analysis and profiling process

Population Stability Index

To detect feature drifts on categorical features, the popular choice is the **Population Stability Index (PSI)**

This statistical method used to measure the shift in a variable over a period of the time. If the over a period of time. If the overall drift score is more than 0.2 or 20%, then the drift is considered significant **(feature drift)**

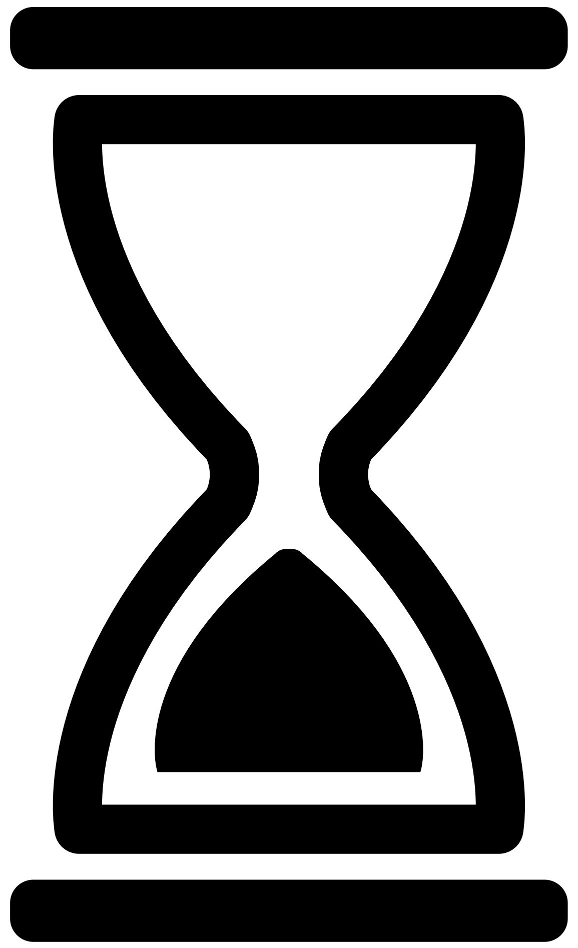
Thorough data analysis and profiling process

Wasserstein metric

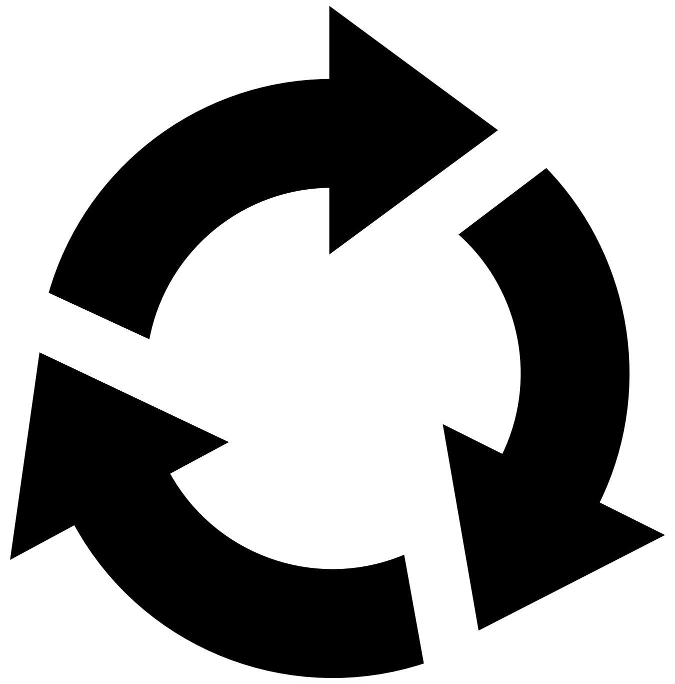
To detect feature drifts on numerical features, the popular choice is the **Wasserstein metric**

This is a distance function for measuring the distance between two probability distribution. Similar to PSI, if the drift score using Wasserstein metric is higher than 20%, then the drift is considered significant **(feature drift)**

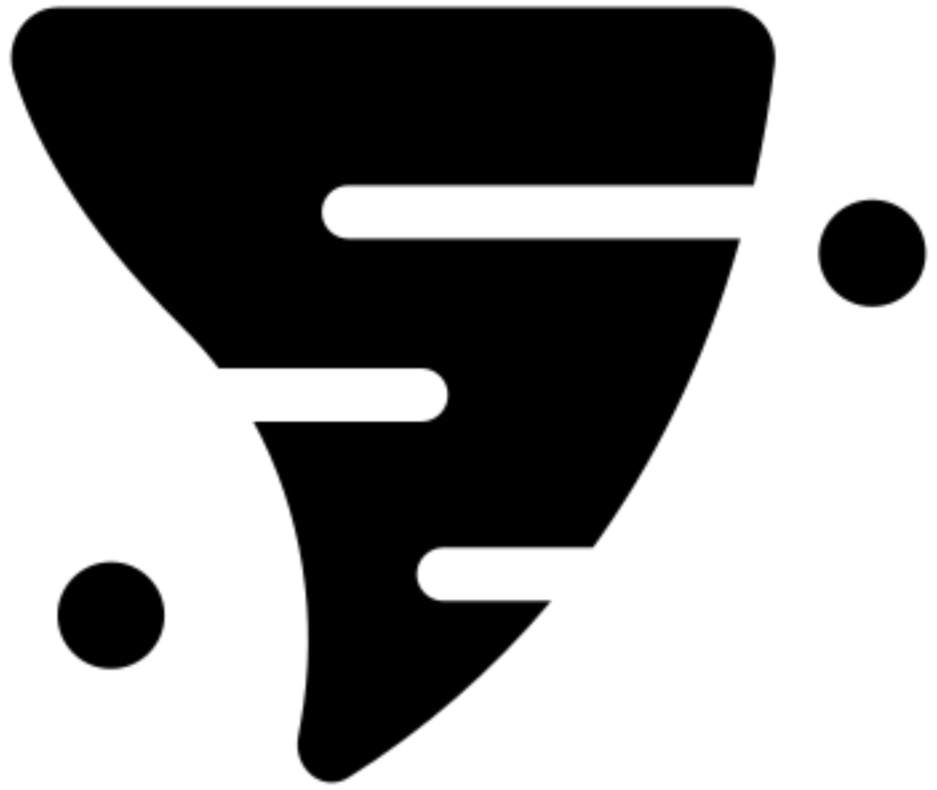
What do we do when we have identified the presence of drifts?



Temporary drift



Recurrent/Seasonal drift

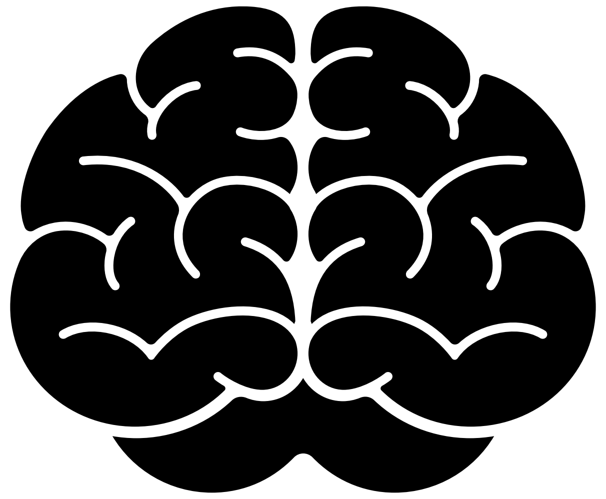


Permanent drift

Checking adversarial robustness

Checking adversarial robustness

Before the model is deployed in production, it is extremely critical to check for the **adversarial robustness**



The degree of adversarial attacks increases with the model’s complexity, as complex models are very sensitive to noisy data samples.

Checking adversarial robustness

We are interested on the impact of adversarial effects on trained ML models

Checking adversarial robustness

There are different types of adversarial attacks that can impact trained ML models:

1. **Fast Gradient Sign Method**
2. **The Carlini & Wagner (C&W) attack**
3. **Targeted adversarial patch attacks**

Checking adversarial robustness

Fast Gradient Sign Method (FGSM)

Method that uses gradients of deep learning models to learn adversarial sample

For image classifiers, this can be a common problem, as FGSM creates perturbations on the pixel values of an image by adding o subtracting pixel intensity values depending on the direction of the gradient descent of the model

Checking adversarial robustness

Fast Gradient Sign Method

The Fast Gradient Sign Method (FGSM) is an adversarial attack technique used in the context of machine learning, particularly in the realm of neural networks. It was introduced to demonstrate the vulnerability of neural networks to adversarial attacks. The basic idea behind FGSM is quite straightforward:

Checking adversarial robustness

Fast Gradient Sign Method

Gradient Calculation: Start with a legitimate input for which you want to generate an adversarial example. Calculate the gradients with respect to the input of the loss function using the model you are attacking

Checking adversarial robustness

Fast Gradient Sign Method

Adversarial Input Generation: Modify the legitimate input by adding a small perturbation in the direction of the gradient. This perturbation is determined by the sign of the gradient multiplied by a small value called epsilon (ε). The goal is to perturb the input in a way that maximizes the loss function

Checking adversarial robustness

Fast Gradient Sign Method

The mathematical formula for FGSM can be expressed as follows, assuming

- x is the original input,
- J is the loss function, and
- ε is the small perturbation:

Checking adversarial robustness

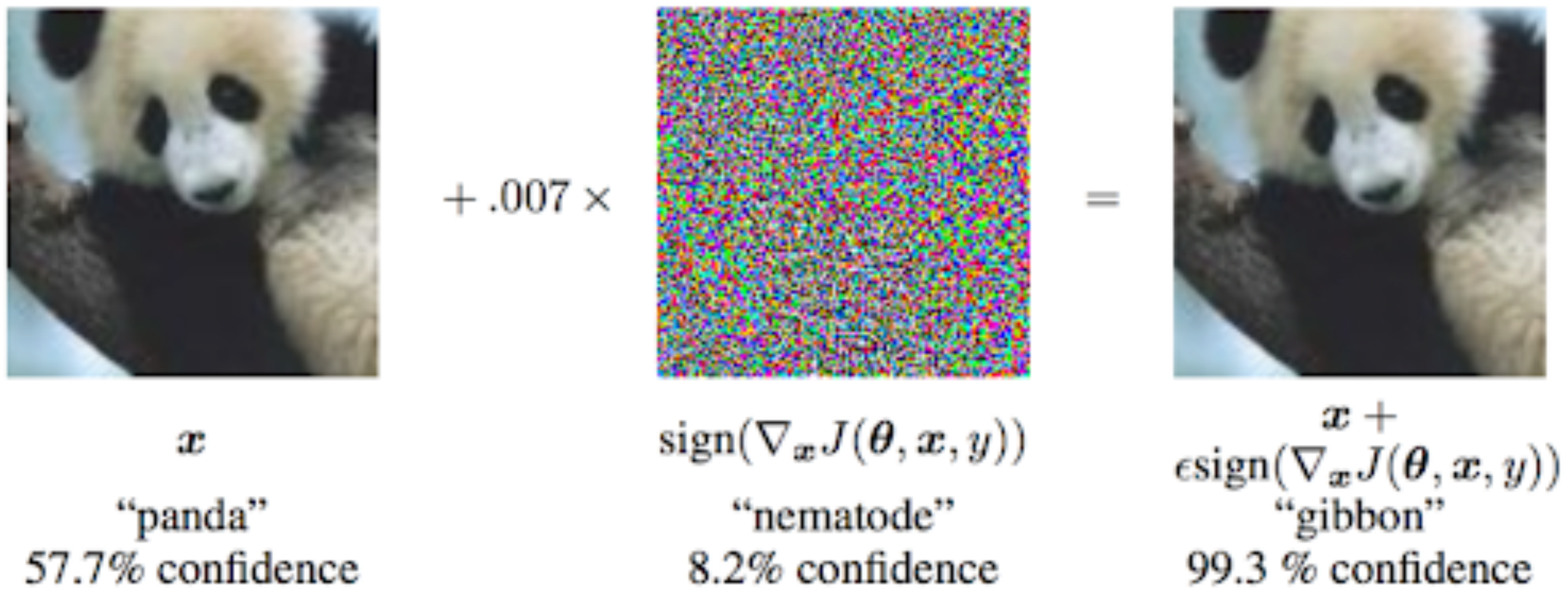
Fast Gradient Sign Method

$$x_{adversarial} = x + \varepsilon \cdot \text{sign}(\nabla_x J(x, y_{true}))$$

- $x_{adversarial}$ is the perturbed input that is hoped to deceive the model
- $\nabla_x J(x, y_{true})$ represents the gradient of the loss function with respect to the input
- sign($\cdot$) returns the sign of the argument, retaining only the information about the direction

Checking adversarial robustness

Fast Gradient Sign Method



Checking adversarial robustness

The Carlini & Wagner (C&W) attack

This method uses the three norm-based distance metrics (L_0, L_2 and L_{inf}) to find adversarial examples, such that the distance between the adversarial example and the original sample is minimal

C&W > **FGSM**

Detecting C&W attacks is more difficult than FGSM attacks

Checking adversarial robustness

Targeted adversarial patch attacks

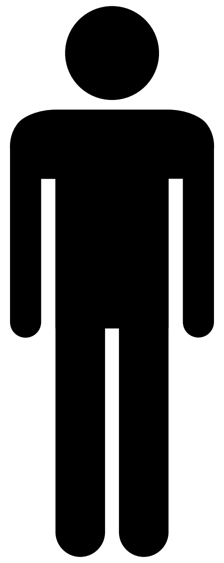
Sometimes, injecting noise into entire image is not necessary. The addition of a noisy image segment to only a small portion of the image can be equally harmful to the model. Targeted adversarial patch attacks can generate a small adversarial patch that is the superimposed with the original sample, thus occluding the key features of the data and making the model classify incorrectly

Checking adversarial robustness

Example of adversarial perturbation with FGSM

Source: https://www.tensorflow.org/tutorials/generative/adversarial_fgsm?hl=it

Problem
for model



No problem
for human

Checking adversarial robustness

Methods to increase adversarial robustness

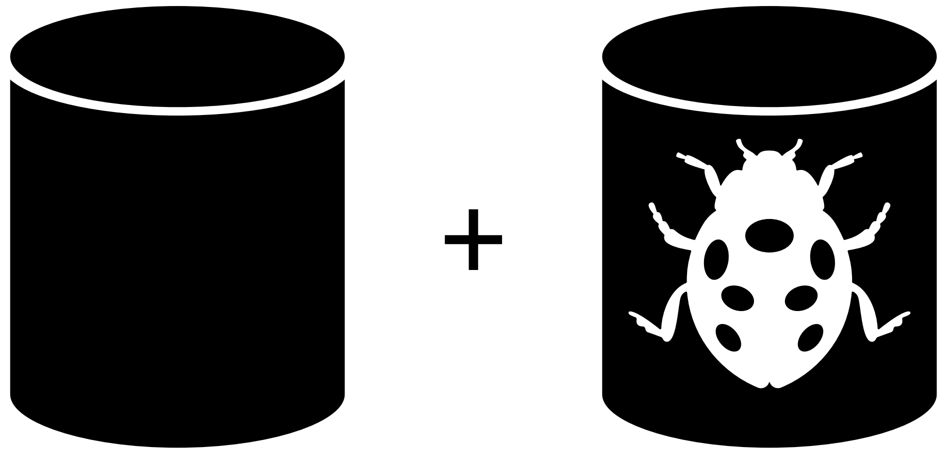
In production system, adversarial attacks can mostly inject noise into inference data. So, to reduce the impact of adversarial attacks, we can adopt different strategies

Checking adversarial robustness

Methods to increase adversarial robustness

Defense mechanism

In order to filter out any abrupt change from any signal, we usually try to apply a smoothing filter such as **Spatial smoothing**.



Adversarial training

By using the technique of **data augmentation,** we can generate adversarial samples from the original data and include the augmented data during the training process

(tips: Fine Tuning the original model with adversarial samples)

Checking adversarial robustness

Evaluating adversarial robustness

How can we measure the adversarial robustness of the models?

↓ Stress testing

↓ Segmented stress testing

Checking adversarial robustness

Stress testing

In Stress testing, adversarial examples are generated by FGSM or C&W methods

Following this, the model's accuracy is measured on the adversarial examples and compared the model accuracy obtained with the original data.

Checking adversarial robustness

Segmented stress testing

In Segmented stress testing, instead of measuring the adversarial robustness of the entire model on the entire dataset, segments of the dataset (either for specific classes or for specific features) are considered to compare the model robustness with the adversarial attack strengths.

Summary

- Data-centric XAI can provide explainability to the black-box model in terms of the data volume, consistency and purity
- Monitoring data drifts for production ML systems is also an essential part of the data-centric XAI process
- Estimating the adversarial robustness of ML models and the detection of adversarial attacks form an important part of the process

Nice Tools