FIELDBOX

BORDEAUX INP Enseirb-Matmeca

ENSEIRB-MATMECA

8 Nov 2024

CHAIRE IA
DIGNE DE CONFIANCE

Agenda

01

Introduction

02

About Fieldbox

03

What is MLOPS?

04

Imbalance

Introduction

01

Three courses

The three courses will be on three mornings:
→ Nov 8th - 8:30 to 12:30
Nov 22nd - 8:30 to 12:30
Nov 29th - 8:30 to 12:30

If you can't come please let me know by email as soon as you can.

We will have a hands-on lecture, with discussion.

We will alternate between black-boarding and coding in Python.

You can send me your code or commit it to git.
Deadline: end of the course.

At the end of each course I will propose ways to go further if you want.

My name is Julien Budynek and I am VP Data Science at Fieldbox.
I work on Data and AI projects delivered for Fieldbox clients, and on internal projects to advance our internal AI R&D roadmap.

You can reach me at jbudynek@fieldbox.ai

The supporting files for this course are available on my github:
https://github.com/jbudynek/teaching-01-imbalance

About Fieldbox

02

Fieldbox mission

Our mission is to ensure industrial companies can rely on digital, data & AI to deliver sustainable efficiency bringing resilience & agility.

DIGITAL DATA & AI

are required to manage the increasing complexity of industrial operations.

80%

of Data & AI solutions are not put into production and do not produce value.

Confidential - To be shared only with intended recipients.

Go beyond the Wall of Operationalization

Fieldbox helps you secure & accelerate your Data & AI-driven Transformation

INITIAL DIGITAL AWARENESS

- No data initiatives in place yet
- No dedicated staff
- Relying mainly on Excel sheets and manual data entry

FIRST PoC

- First Proof of Concept (PoC)
- No actual implementation
- Team limited to data scientists
- No clear governance structure

STRATEGIC DIGITAL ROADMAP

- Framing and prioritization of use cases that generate the most value
- Definition of roles, workflows, governance, and expected skills

MYRIAD OF PoC SILOS

- Multiple isolated PoCs
- Struggle to scale
- Unable to extract value

FIRST FULL INDUSTRY 4.0 PILOT SITE

- Strategic topic with CEO focus
- 1 Data & AI solution in production
- Operations onboarded
- Pilot Site 4.0

FULL SCALE INDUSTRY 4.0

- Data & AI-driven mindset, driving business decision
- Clear value-driven portfolio of cases
- Full scale deployments & adoption
- Full organization onboarded

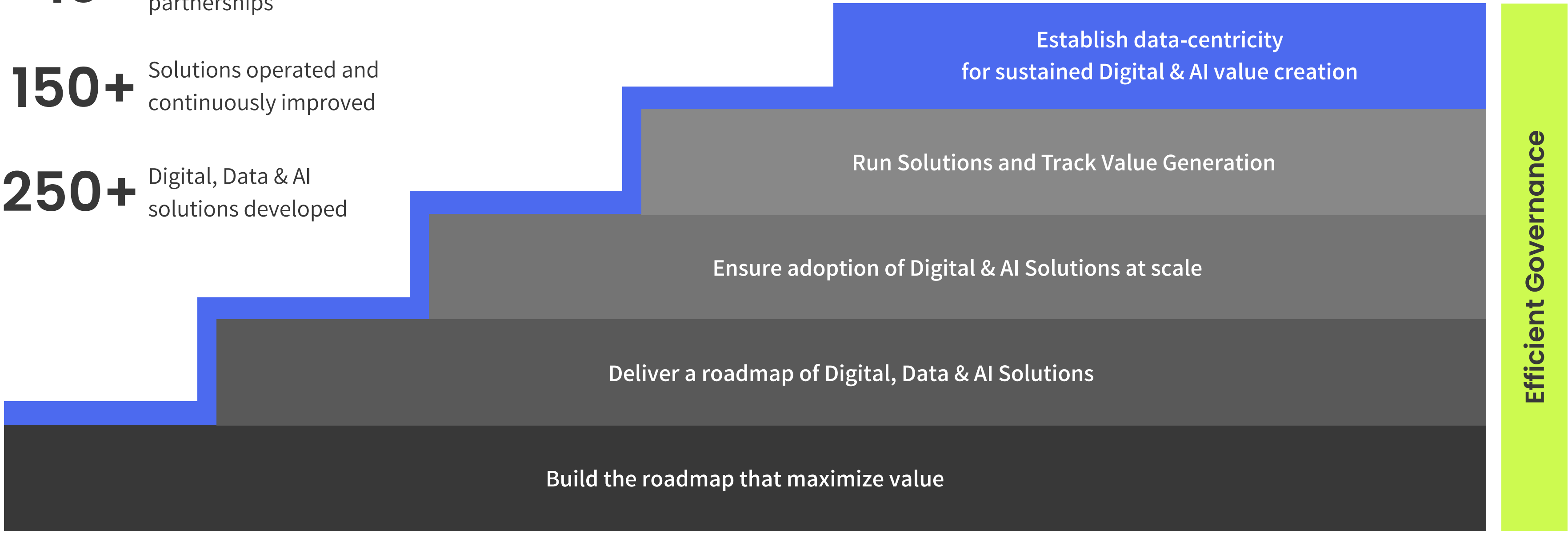
Confidential - To be shared only with intended recipients.

Fieldbox is your Digital & AI Scale-Up Partner

10+ Key strategic partnerships

150+ Solutions operated and continuously improved

250+ Digital, Data & AI solutions developed



Fieldbox support your organization in every steps toward sustainable value generation

Confidential - To be shared only with intended recipients.

Using Digital, Data & AI to Reach Operational Excellence in Industrial Operations

Safety & Risk detection

Energy Management

Predictive Maintenance

Inventory Management

Quality Control

Process Control

Production Optimization

Supply Chain Optimization

Confidential - To be shared only with intended recipients.

Fieldbox: a Decade of Cutting-Edge Data & AI Solutions for Industry

Fieldbox offices

Customer locations operated with Fieldbox solutions

- Operating on 5 continents
- Independent & organic growth

2011
Creation of the company (as IDMOG)

2014
Extension to Africa & South America

2016
New AI capabilities
Expansion to new verticals

2018
New offices in Paris & Singapore

2020
+50% revenue growth

Selected references

Partners

Confidential - To be shared only with intended recipients.

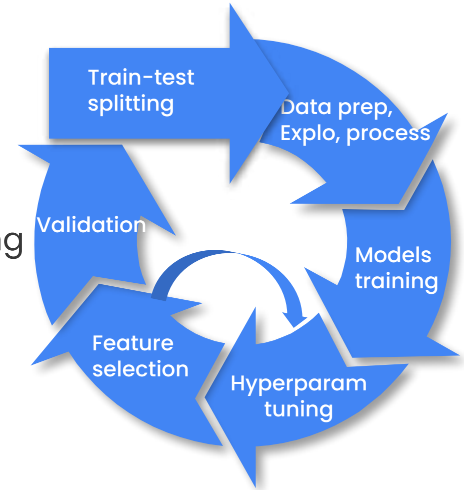
Data project lifecycle

Pre-sales sales | Architect | Data engineers | AI research | DS Team | AI engineers | Software engineers | IT, Ops | Services | Run Team

Build project team: Fieldbox team **& client**

Pre-sales phase

Defining scientific and technical objectives
Global project planning
Data batch gathering



Deliver / Deploy

Live test
Monitor
Maintain
Re-train
Coach

Running phase

~ months

What is MLOPS?

03

What is MLOPS?

There are several ways to look at MLOPS.

Some see it as the complete process governing the lifecycle of a machine learning model: Data collection - Analytic formulation - Modelization - Test - Deployment - Monitoring - and loopback.

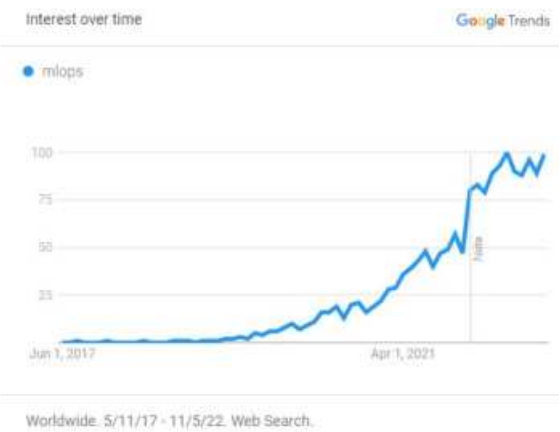
Some say that MLOPS is the practice of making all of those steps easier with tooling.

We will focus on some of the upstream activities, and also on deployment and monitoring.

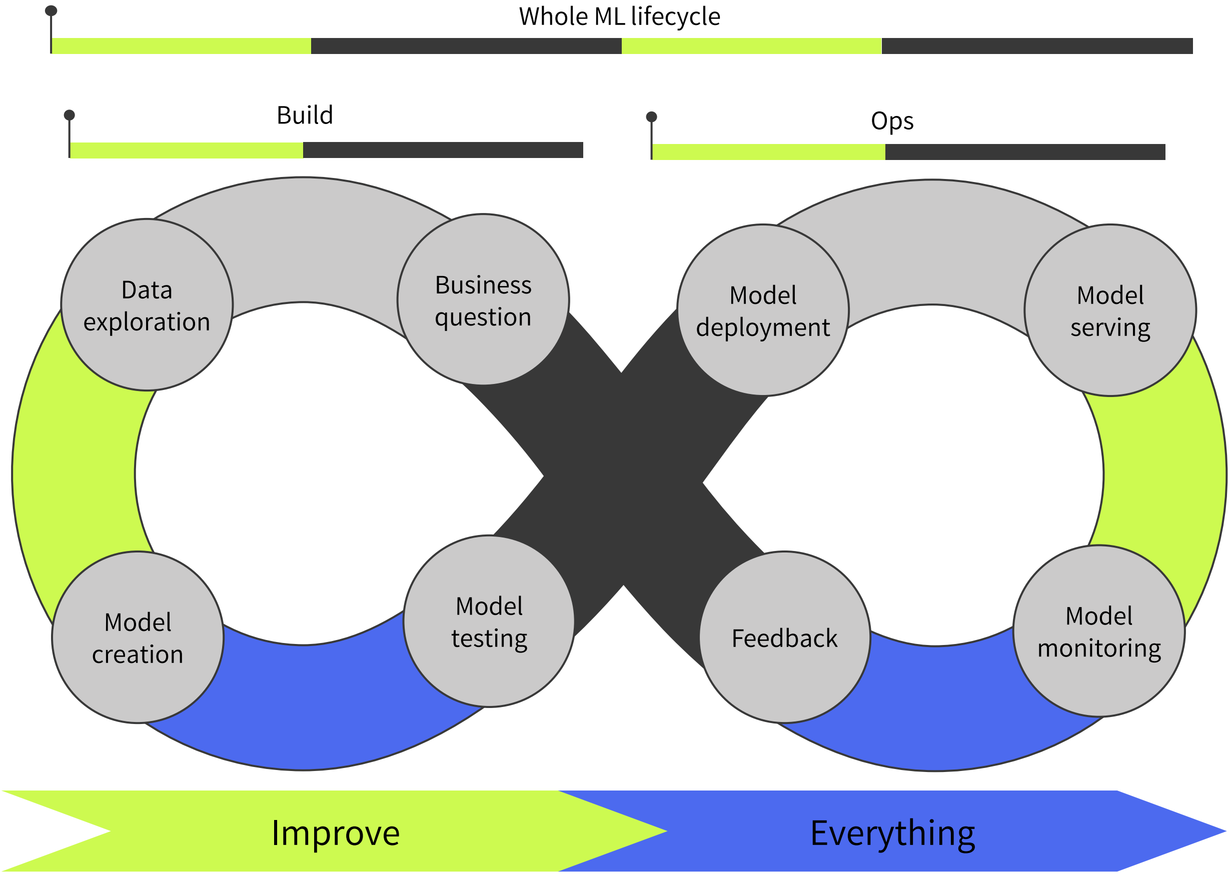
“MLOps is the standardization and streamlining of machine learning lifecycle management.” Wikipedia - march 2022

“MLOps is a set of practices that aims to deploy and maintain machine learning models in production reliably and efficiently”
Wikipedia - nov 2022

https://en.wikipedia.org/wiki/MLOps



MLOPS

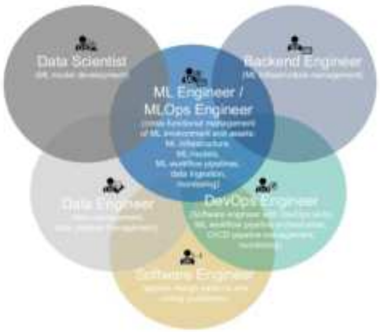

Whole ML lifecycle
Build
Ops
Data exploration
Business question
Model deployment
Model serving
Model creation
Model testing
Feedback
Model monitoring
Improve
Everything

What is MLOPS?

Kreuzberger, Dominik, Niklas Kühl, and Sebastian Hirschl. "Machine Learning Operations (MLOps): Overview, Definition, and Architecture." https://arxiv.org/abs/2205.02302

Roles
R1 Business Stakeholder
R2 Solution Architect
R3 Data Scientist
R4 Data Engineer
R5 Software Engineer
R6 DevOps Engineer
R7 ML Engineer/MLOps Engineer

Principles
P1 CI/CD automation
P2 Workflow orchestration
P3 Reproducibility
P4 Versioning
P5 Collaboration
P6 Continuous ML training & evaluation
P7 ML metadata tracking/logging
P8 Continuous monitoring
P9 Feedback loops



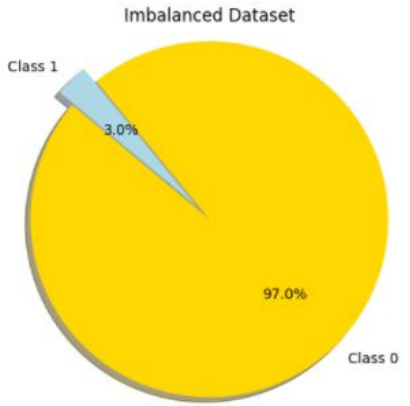
Tech Components
C1 CI/CD Component (P1, P6, P9).
C2 Source Code Repository (P4, P5).
C3 Workflow Orchestration Component (P2, P3, P6).
C4 Feature Store System (P3, P4).
C5 Model Training Infrastructure (P6).
C6 Model Registry (P3, P4).
C7 ML Metadata Stores (P4, P7).
C8 Model Serving Component (P1).
C9 Monitoring Component (P8, P9).

Imbalance

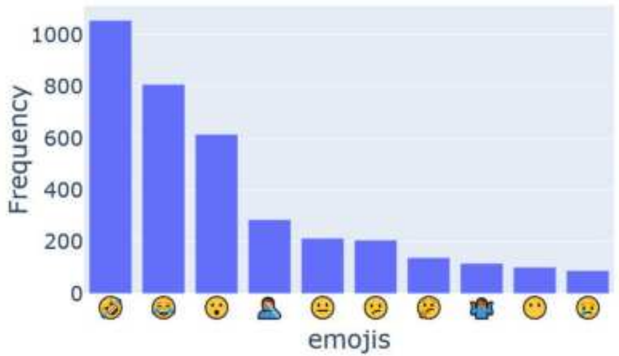
04

About imbalance

We will consider the question of 2-class classification.
In general we consider or we suppose that both classes are equally represented in the dataset.
What is imbalance? It is when one of the classes in less frequent than the other.
In 2-class classification, if one class is <10% we have imbalance.



Freq of emojis in a whatsapp conversation

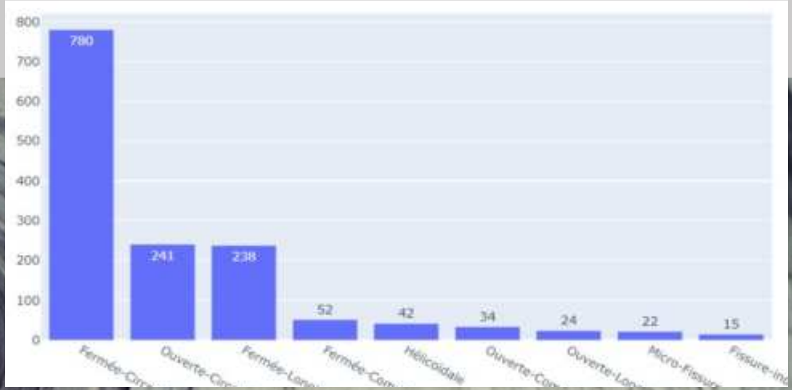


More imbalance

Imbalance shows up all the time in real life, and in particular in industrial use cases.
Predictive maintenance: some equipments have dysfunctions, but sometimes only once every two years!
Quality control: by design, the proportion of samples that do not meet the quality is very low.
The fields of anomaly detection and rare events prediction are linked to imbalanced data.

Finance: fraud detection datasets commonly have a fraud rate of ~1–2%
Ad Serving: click prediction datasets have click-through-rate <1%.
Medical: classification of patients with rare condition
Content moderation: detection of unsafe content

Automated pipe condition analysis and reporting



- Training of a computer vision model on a set of categories and sub-categories
- Application for the automatic detection of anomalies
- Co-development of an off-the-shelf product with the client's SMEs

TIME REDUCTION FOR PIPE INSPECTION BY 50%

REPORTING TIME REDUCTION BY 80%

FieldBox.ai

Hands on coding – expected output

In the following sections we will see three ways to deal with imbalanced data:

1. Dedicated metrics
2. Dedicated models
3. Dedicated data manipulation

Below is the expected output of your notebook:

						Training						Testing					
#ID	Classifier name	Modifiers	Sampling	dataset size	% imbalance	precision	accuracy	recall	f1	balanced accuracy	training size	precision	accuracy	recall	f1	balanced accuracy	testing size
1	Random Forest	N/A	N/A	1000	10												
2	Random Forest	class_weight	N/A	1000	10												
2	Random Forest	class_weight	oversampling	1000	...												

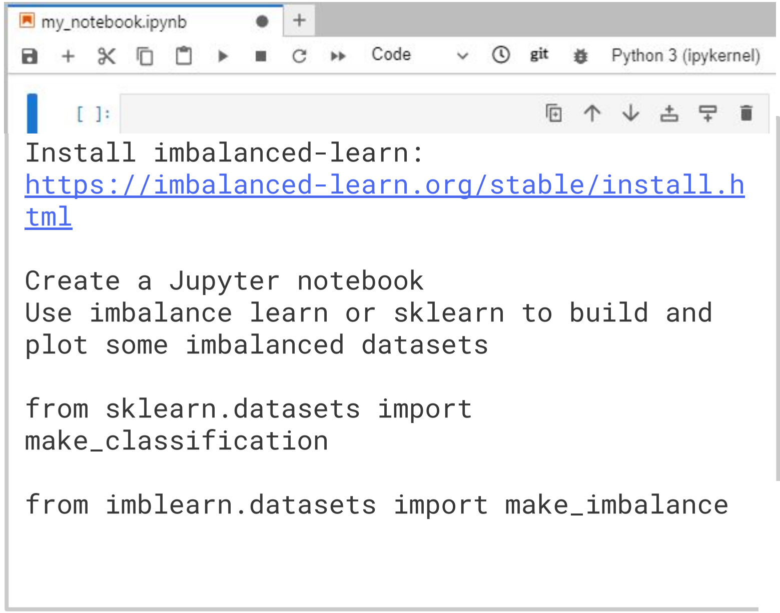
A few sentences of interpretation are also expected.

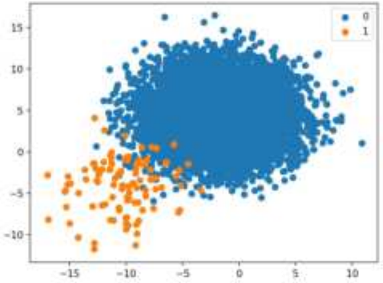
Datasets – synthetic data

Let's create and plot several toy datasets that we will use for this course.

Let’s try 2D/10D/20D data, two classes, 1-2-5-10-20% imbalance, size 10000.

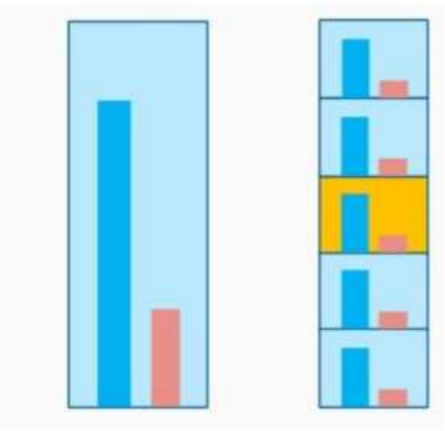
See notebook **00_imbalanced_synthetic.ipynb**





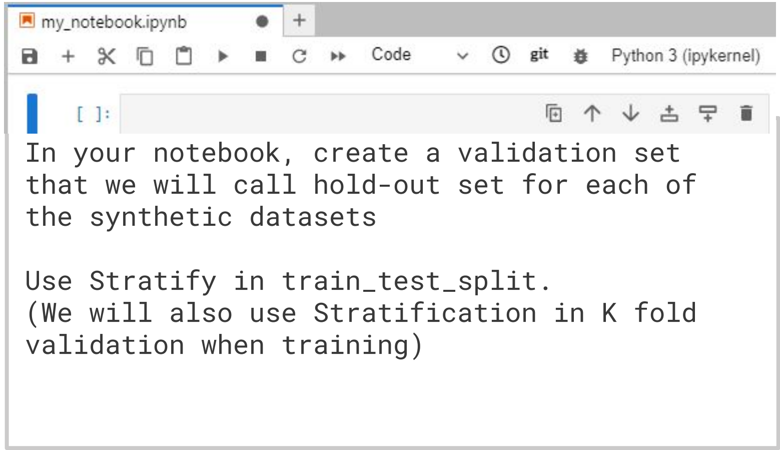
Splitting

Let's split our datasets and keep an validation set on the side. It should be stratified correctly.



Full dataset

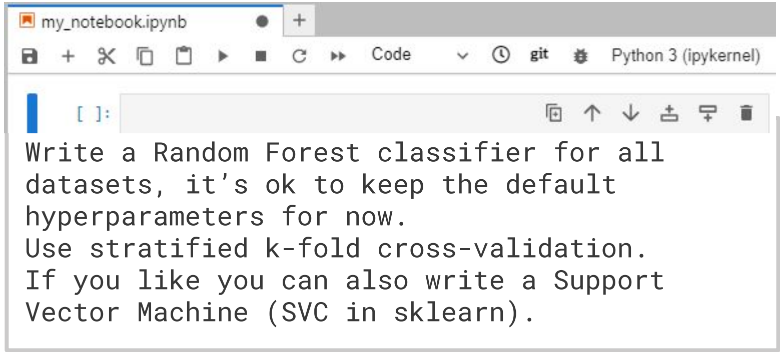
Hold-out for validation (in orange)

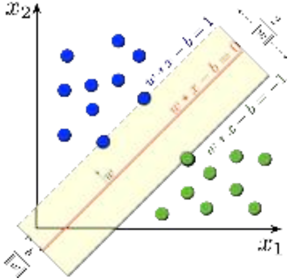


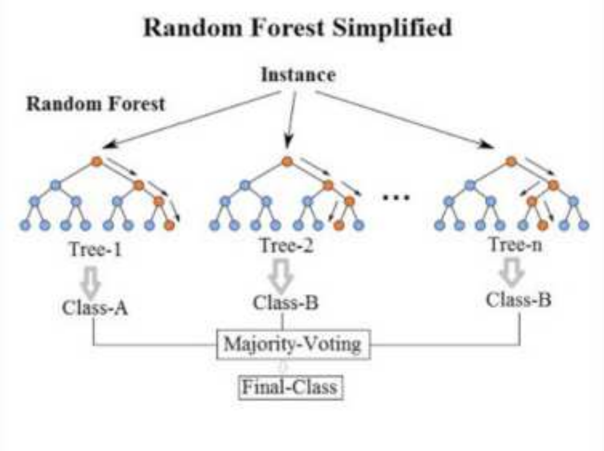
Baseline model

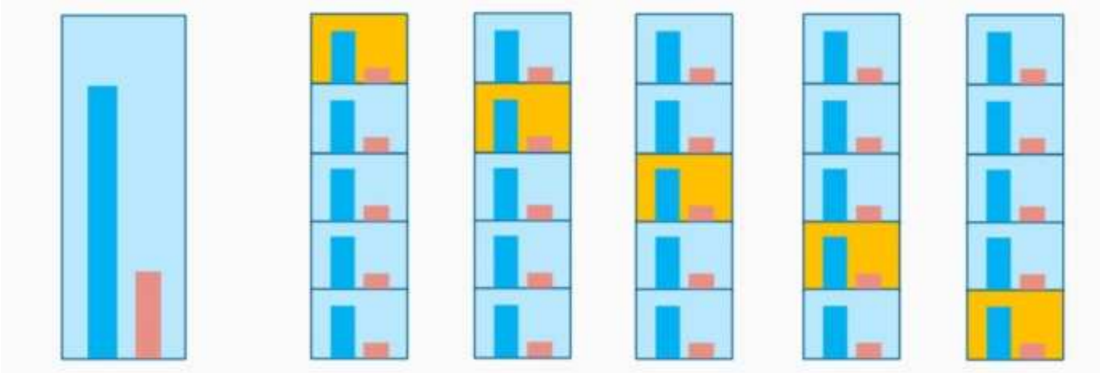
Then we can work on a baseline model, a simple classification algorithm such as Random Forests.

sklearn doc: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html





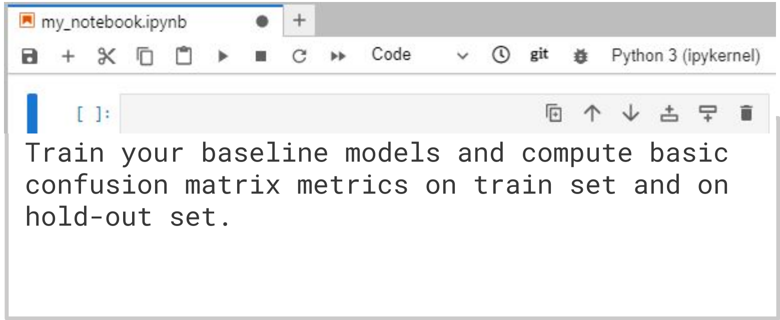




Full dataset Fold 1 Fold 2 Fold 3 Fold 4 Fold 5

Training the baseline models

Let's train it, and measure basic performance on our validation set with a confusion matrix
We have TP, TN, FP, FN.



		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection

Classic Metrics

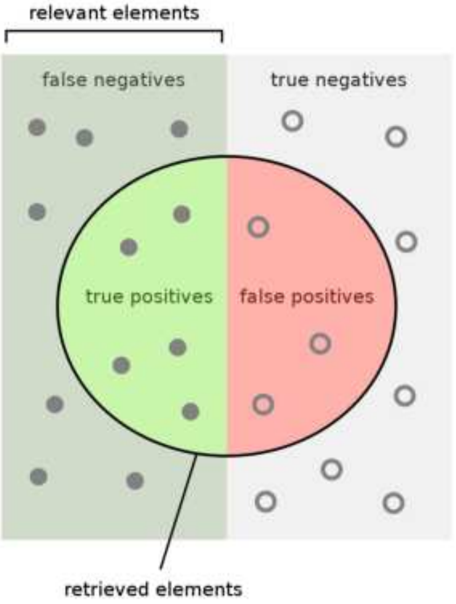
We can compute the usual metrics stemming from a confusion matrix: precision, accuracy, recall.

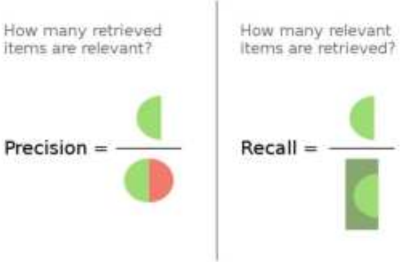
Let's see how it goes for all our datasets. Typically precision and accuracy are very high but do not reflect at all what we want.

You get a high accuracy just by predicting the majority class, but you fail to capture the minority class, which is most often the point of the question.

sklearn doc:
https://scikit-learn.org/stable/modules/model_evaluation.html

Wikipedia:
https://en.wikipedia.org/wiki/Confusion_matrix





precision or **positive predictive value** (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDR}$$

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

sensitivity, **recall**, **hit rate**, or **true positive rate** (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

	Real class 1	Real class 0
Predicted class 1	2	0
Predicted class 0	8	100

True Positives = 2
False Positives = 0
True Negatives = 100
False Negatives = 8

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total \# samples}}$$

$$\text{Accuracy} = \frac{2 + 100}{110}$$

Accuracy = 0.92

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Precision} = \frac{2}{2 + 0}$$

Precision = 1

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Recall} = \frac{2}{2 + 8}$$

Recall = 0.2

$$\text{F1-Score} = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}}$$

$$\text{F1-Score} = \frac{2 \times (0.2 \times 1)}{0.2 + 1} = \frac{0.4}{1.2}$$

F1-Score = 0.33

Alternative metrics more suited to imbalance

Instead of the regular classification metrics, you can use more suited metrics.

F1 score is the harmonic mean of precision and recall.

F-beta score strikes a balance between precision and recall.
If we want to prioritize precision we can use beta=0.5
If we want to prioritize recall we can use beta=2

Balanced accuracy is also a better choice as it takes into account the relative size of the classes.

Let's calculate it for all our datasets.

F1 score

is the harmonic mean of precision and sensitivity:

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$$

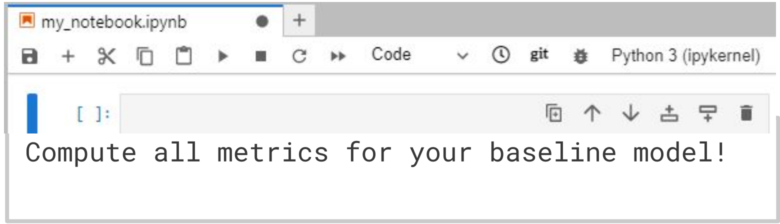
$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

specificity, selectivity or true negative rate (TNR)

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

balanced accuracy (BA)

$$BA = \frac{TPR + TNR}{2}$$

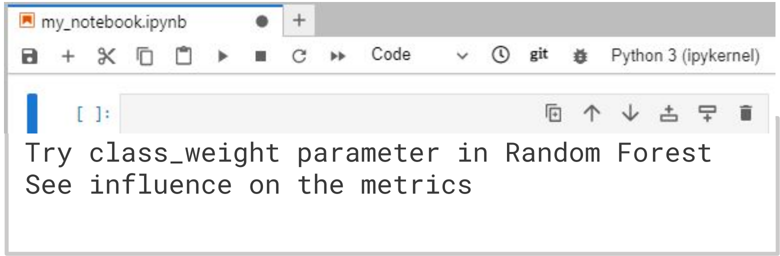


Tweaks to classification models to handle imbalance

Some classification models can behave a bit better than others with imbalanced data, through the use of “class_weight” parameter (Decision Tree, Random Forest, SVC)

The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as
`n_samples / (n_classes * np.bincount(y))`

sklearn doc:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html



Alternative classification models more suited to imbalance

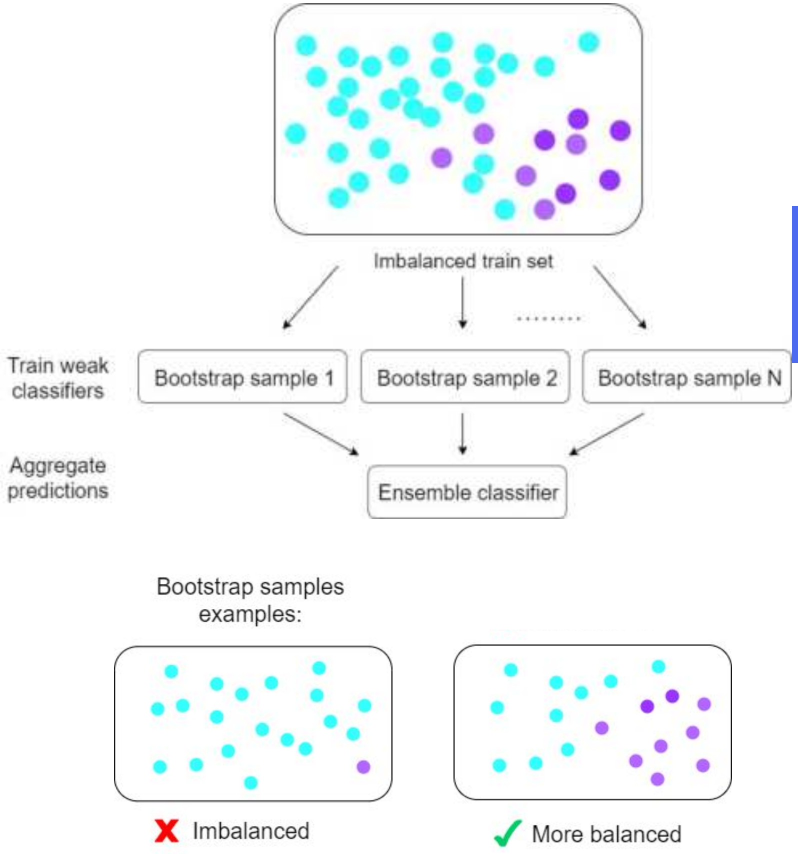
Besides the metrics change you can also use different machine learning models, that are more suited to imbalance.

For instance **Balanced Random Forests** is a variation on Random Forests that deals with class imbalance natively.

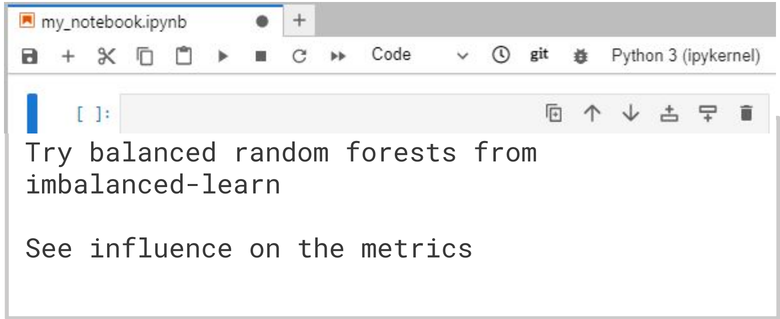
For each iteration of RF, take a bootstrap of minority class, then take the same number of observations in majority class

Reference: Chao **Chen**, Andy Liaw, Leo Breiman, and others. Using random forest to learn imbalanced data. University of California, Berkeley, 110(1-12):24, 2004.

Imbalanced-learn ref:
https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html



Alternative classification models more suited to imbalance



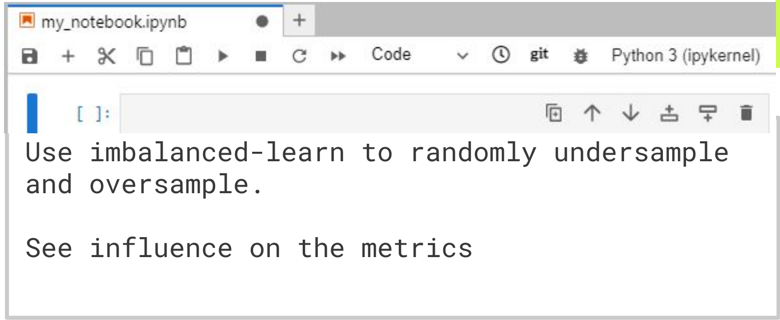
If you are in a strongly imbalanced case, you can decide to use one-class classification models such as isolation forests or single class svm.
You can also use anomaly detection techniques.

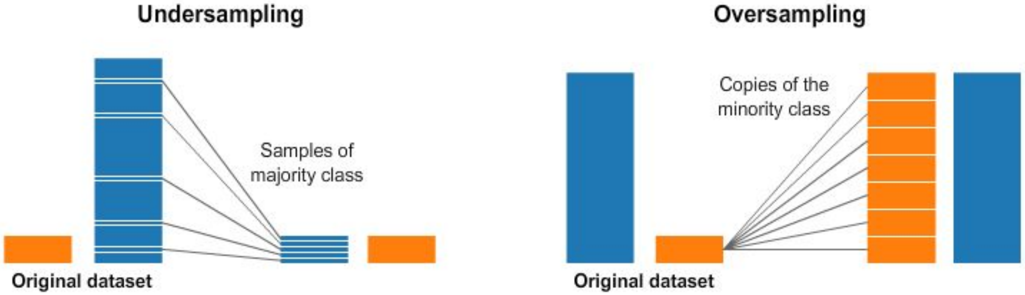
Change the input data

Then, a powerful method to address class imbalance is to use undersampling and oversampling techniques.
If the minority class has a sufficient number of representants, then you can try undersampling the majority class at random.

Otherwise, you must find clever ways to oversample the minority class. You can simply duplicate data.

You could also try to add noise to members of the minority class.





SMOTE

Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357
https://www.jair.org/index.php/jair/article/download/10302/24590

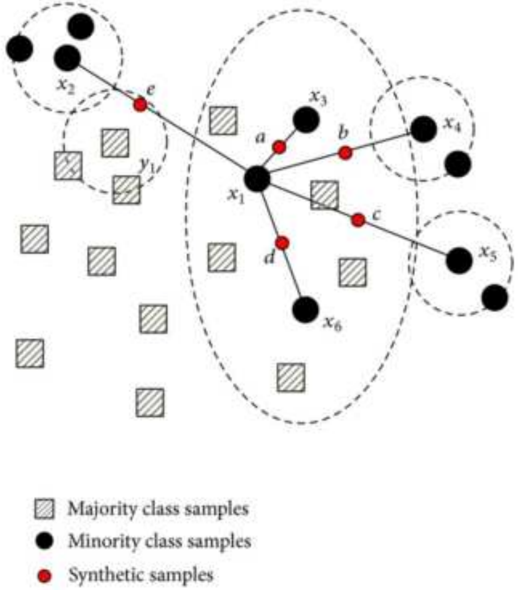
Finally, you could use **SMOTE** -
Synthetic Minority Oversampling TEchnique.

SMOTE allows you to interpolate between members of your minority class in order to create additional data points.

For each sample in the minority class:
Compute its K-Nearest Neighbors (KNN).
Select one of them randomly.
Synthesize a new observation by linear interpolation.

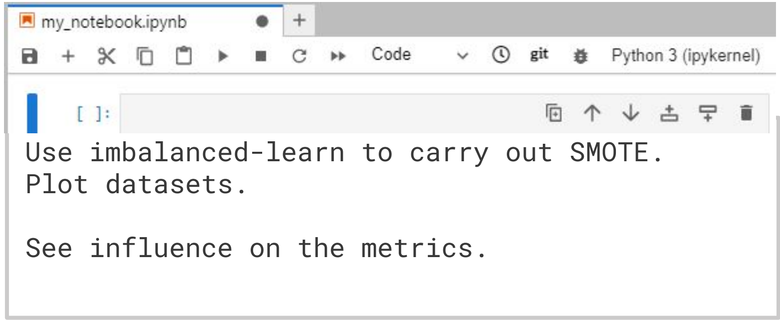
Source of schema - https://www.researchgate.net/publication/287601878_A_Novel_Boundary_Oversampling_Algorithm_Based_on_Neighborhood_Rough_Set_Model_NRSBoundary-SMOTE
Source of algorithm https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/node6.html

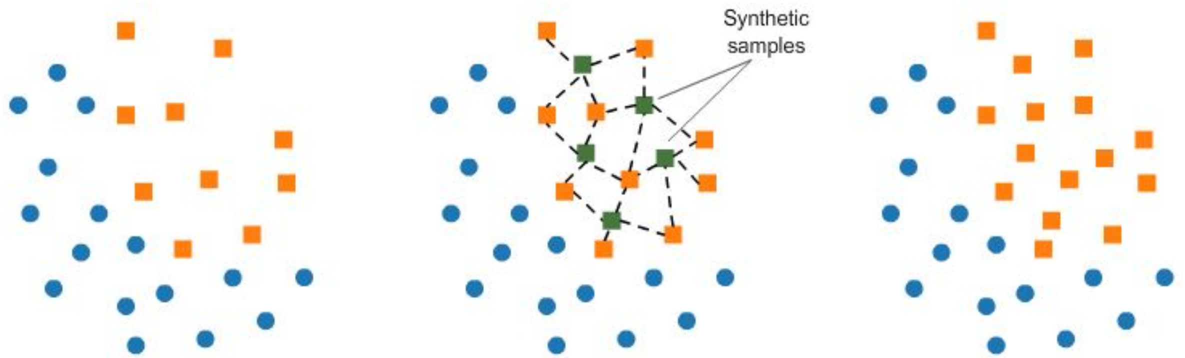
```
Algorithm SMOTE(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE N%;
    Number of nearest neighbors k
Output: (N/100) * T synthetic minority class samples
(* If N is less than 100%, randomize the minority class samples as
only a random percent of them will be SMOTEd. *)
if N < 100
    then Randomize the T minority class samples
        T = (N/100) * T
        N = 100
endif
N = (int)(N/100) (* The amount of SMOTE is assumed to be in
integral multiples of 100. *)
k = Number of nearest neighbors
numattrs = Number of attributes
Sample[ ][ ]: array for original minority class samples
newindex: keeps a count of number of synthetic samples generated,
initialized to 0
Synthetic[ ][ ]: array for synthetic samples
(* Compute k nearest neighbors for each minority class sample only. *)
for i ← 1 to T
        Compute k nearest neighbors for i, and save the indices in
        the nnarray
        Populate(N, i, nnarray)
endfor

Populate(N, i, nnarray) (* Function to generate the synthetic sam-
ples. *)
while N ≠ 0
        Choose a random number between 1 and k, call it nn. This
        step chooses one of the k nearest neighbors of i.
        for attr ← 1 to numattrs
                Compute: dif = Sample[nnarray[nn]][attr] − Sample[i][
                Compute: gap = random number between 0 and 1
                Synthetic[newindex][attr] = Sample[i][attr] + gap *
                dif
        endfor
        newindex++
        N = N − 1
endwhile
return (* End of Populate. *)
End of Pseudo-Code.
```



SMOTE

Let's use SMOTE on our toy datasets and plot them.
Then let's learn on these augmented datasets and see how the test metrics change.





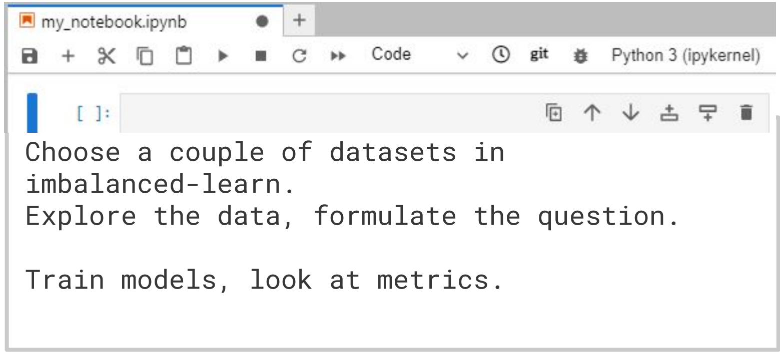
Datasets – real data

There are several public datasets that exhibit imbalance, now that we went all the way with synthetic data, we can use them too.

A few of them are available in **imbalanced-learn** directly.

https://imbalanced-learn.org/stable/references/generated/imblearn.datasets.fetch_datasets.html

See notebook
01_imbalanced_public_dataset.ipynb



SECOM dataset

https://archive.ics.uci.edu/ml/datasets/SECOM

A complex modern **semi-conductor manufacturing process** is under consistent surveillance via the monitoring of signals/variables collected from sensors and/or process measurement points. The dataset presented in this case represents a selection of such measurements where each example represents a single production entity with associated measured features and the labels represent a simple **pass/fail yield** for in house line testing, where –1 corresponds to a pass and 1 corresponds to a fail and the data time stamp is for that specific test point.

Quality control question.

See notebook **02_imbalanced_secom.ipynb**

my_notebook.ipynb

Code

Python 3 (ipykernel)

```
[ ]:
```

Download the data.

Explore the data, formulate the question.

Choose one model, metric, and data augmentation technique.
Train your model **with cross-validation**, look at metrics.

```
import pandas as pd
dataset_url =
"http://archive.ics.uci.edu/ml/machine-learning-databases/secom/secom.data"
label_url =
"http://archive.ics.uci.edu/ml/machine-learning-databases/secom/secom_labels.data"
data_df = pd.read_csv(dataset_url, sep=" ",
header=None, names=[f"F{i+1:03d}" for i in
range(590)])
label_df = pd.read_csv(label_url, sep=" ",
header=None, names=["Label", "Time"],
parse_dates=["Time"], dayfirst=True)
  df = pd.concat([label_df, data_df], axis=1)
```

NASA Turbofan dataset

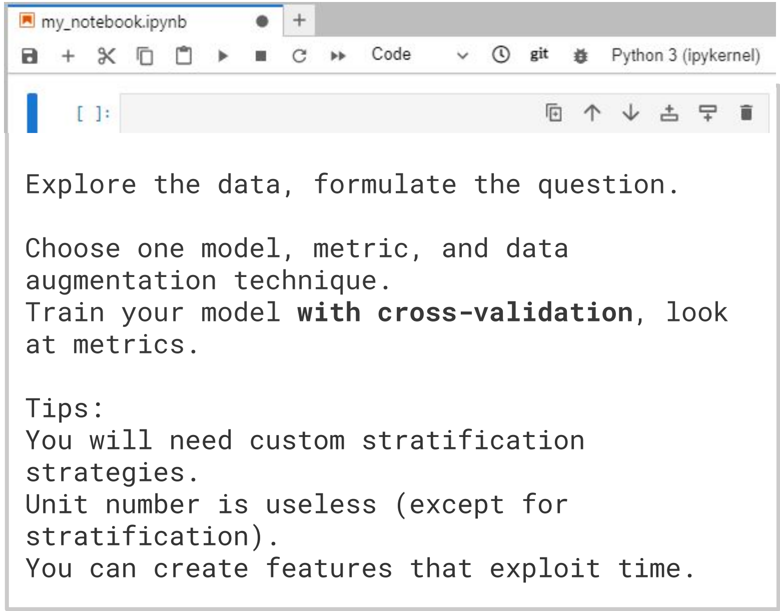
https://www.kaggle.com/datasets/behrad3d/nasa-cmaps

Use only FD001

Dataset presents Run-to-Failure simulated **data from turbo fan jet engines**. It consists of multiple multivariate time series. Each time series is from a different engine. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user.
The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the dataset, the fault grows in magnitude until system failure.

We want to predict whether the unit will fail within the next 5 cycles.

See notebook **03_imbalanced_turbofan.ipynb**



Conclusion and perspectives

Data imbalance is frequent and you can deal with it in several ways.
You can :

- change your data,
- use appropriate models,
- use appropriate metrics.

More about SMOTE:

- There are other approaches that can be used for regression questions, in particular one is SMOTER.
- At Fieldbox we also studied an algorithm for sequence-to-sequence questions, which is called SMOTEST.
- SMOTE has been discussed in a recent paper on medical trials and seem to be counterproductive sometimes.

Ideas of complementary work:

- Find other datasets and work with them
- Study SMOTER and implement it in Python
- Study this paper that tempers SMOTE:

van den Goorbergh, Ruben, et al. "The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression." Journal of the American Medical Informatics Association (2022).

- Watch this video of Guillaume Lemaitre at euroscipy2023 about some advanced scikit-learn features, some of which deal with imbalance.

FIELDBOX

Thank you

ENSEIRB-MATMECA
8 Nov 2024