

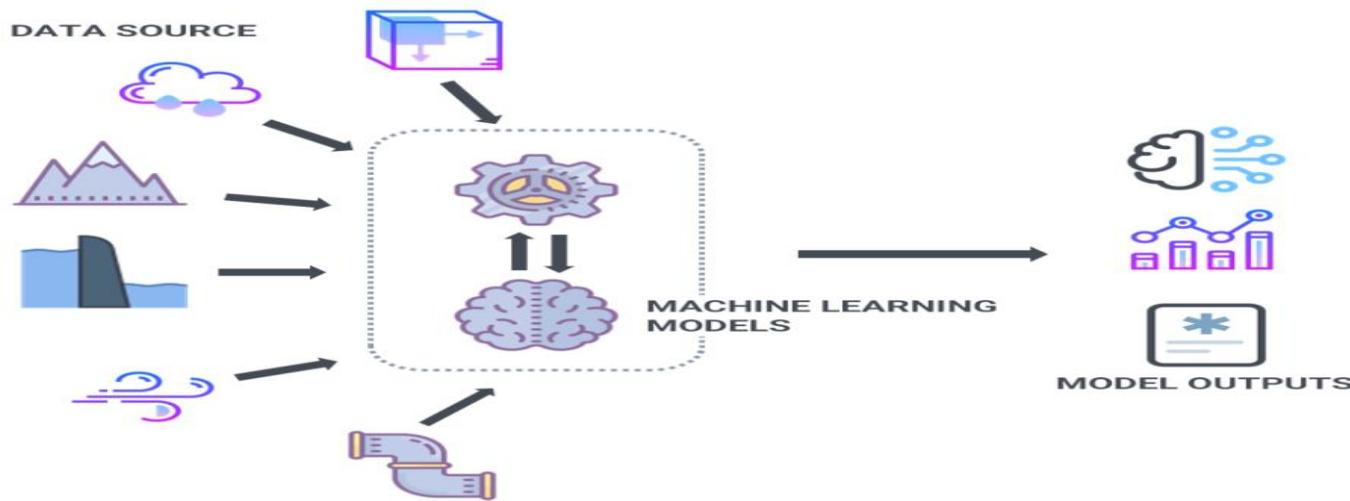
2022/2023

Réalisé par: Naïma Daghfous

# Matrice de confusion

# Matrice de confusion

## ○ Besoin d'évaluer le modèle de ML



## ○ Pour évaluer le modèle de ML Nous avons besoin de la Matrice de confusion

		Reality	
		Confusion matrix	Positive : 1
Prediction	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

Matrice de confusion

# TN, TP, FN, FP

Confusion matrix		Reality	
		Negative : 0	Positive : 1
Prediction	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

*Matrice de confusion*

- ♣ **Vrai négatif ou TN** (True Negative) : La prédiction est négative et c'est la réalité.
- ♣ **Vrai positif ou TP** (True Positive) : La prédiction est positive et c'est la réalité.
- ♣ **Faux positif ou FP** (False positive), erreur de 1ère espèce : La prédiction est positive mais ce n'est pas la réalité.
- ♣ **Faux négatif ou FN** (False negative), erreur de 2ème espèce : La prédiction est négative mais ce n'est pas la réalité.

# Métriques d'évaluation

- Différentes métriques permettent de juger de la qualité de nos prédictions qui sont principalement

$$Precision(i) = \frac{\text{nb d'individus correctement attribués à la classe } i}{\text{nb d'individus attribués à la classe } i} = \frac{\mathbf{tp}^{(i)}}{\mathbf{tp}^{(i)} + \mathbf{fp}^{(i)}}$$

$$Recall(i) = \frac{\text{nb d'individus correctement attribués à la classe } i}{\text{nb d'individus appartenant à la classe } i} = \frac{\mathbf{tp}^{(i)}}{\mathbf{tp}^{(i)} + \mathbf{fn}^{(i)}}$$

$$F1 - Score (i) = 2 * \frac{Precision(i) * Recall(i)}{Precision(i) + Recall(i)}$$

# Matrice de confusion multi-classes

- Soit la matrice de confusion pour un exemple de prédiction de la météo avec 4 classes:

		Reality			
Confusion matrix		Cloudy	Rain	Shine	Sunrise
Prediction	Cloudy	39	6	13	1
	Rain	3	23	0	0
	Shine	6	0	30	1
	Sunrise	12	10	13	68

Matrice de confusion pour la prédiction météo.

Les lignes correspondent aux prédictions du modèle et les colonnes correspondent à la réalité.

# Matrice de confusion multi-classes

○ **Problème** : Les métriques de classification usuelles sont définies dans le cas binaire à partir de la matrice de confusion binaire. Comment utiliser la matrice de confusion multi-classes pour généraliser les métriques binaires ? Nous allons voir les 3 méthodes classiques :

- ♣ L'approche “per class”,
- ♣ L'approche “macro”,
- ♣ L'approche “micro”

Confusion matrix		Reality			
		Cloudy	Rain	Shine	Sunrise
Prediction	Cloudy	39	6	13	1
	Rain	3	23	0	0
	Shine	6	0	30	1
	Sunrise	12	10	13	68

Matrice de confusion pour la prédiction météo.

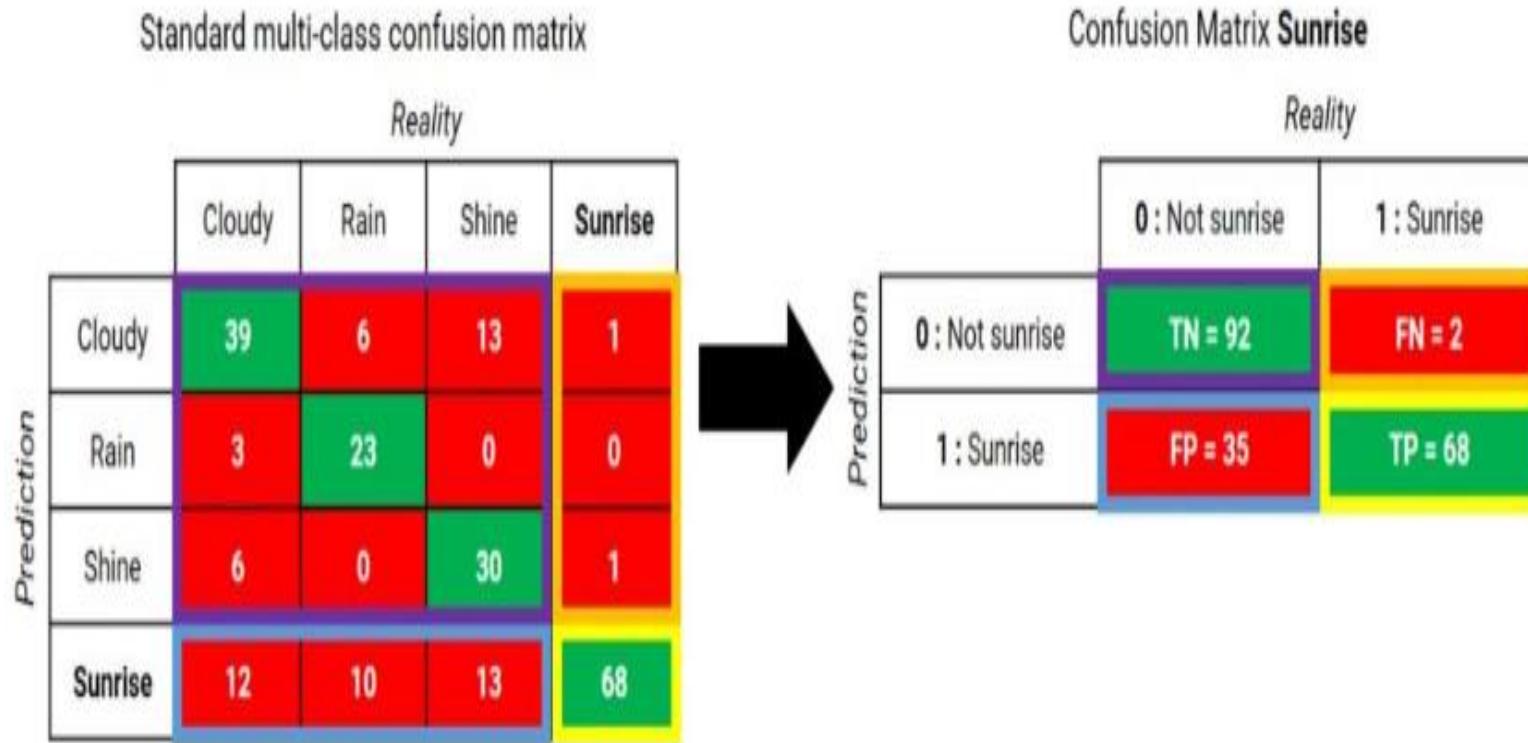
Les lignes correspondent aux prédictions du modèle et les colonnes correspondent à la réalité.

# Les Approches

- ♣ L'approche “**per class**”, permet de s'intéresser à chaque classe individuellement mais il est difficile d'en tirer une synthèse car on obtient une métrique par classe.
- ♣ L'approche “**macro**” qui permet de résumer les métriques obtenues avec l'approche “per class”. Il existe 2 variantes “macro”, une variante classique robuste au déséquilibre de classe et une variante pondérée, plus représentative des données.
- ♣ L'approche “**micro**”, qui utilise directement les valeurs de la matrice de confusion multi-classe pour produire une métrique qui résume la performance du modèle.

# L' Approche “Per class”

- Retrouver la matrice de confusion de chaque classe.



# L' Approche “Per class”

- Retrouver la matrice de confusion de chaque classe.

		Reality	
		0 : Not cloudy	1 : Cloudy
Prediction	0 : Not cloudy	TN = 121	FN = 21
	1 : Cloudy	FP = 20	TP = 39

		Reality	
		0 : Not rain	1 : Rain
Prediction	0 : Not rain	TN = 137	FN = 16
	1 : Rain	FP = 3	TP = 23

		Reality	
		0 : Not shine	1 : Shine
Prediction	0 : Not shine	TN = 130	FN = 26
	1 : Shine	FP = 7	TP = 30

		Reality	
		0 : Not sunrise	1 : Sunrise
Prediction	0 : Not sunrise	TN = 92	FN = 2
	1 : Sunrise	FP = 35	TP = 68

# L' Approche “Per class”

- Calculer une métrique (recall par exemple) pour chaque classe.

		Reality	
Confusion Matrix Cloudy		0 : Not cloudy	1 : Cloudy
Prediction	0 : Not cloudy	TN = 121	FN = 21
	1 : Cloudy	FP = 20	TP = 39

		Reality	
Confusion Matrix Rain		0 : Not rain	1 : Rain
Prediction	0 : Not rain	TN = 137	FN = 16
	1 : Rain	FP = 3	TP = 23

		Reality	
Confusion Matrix Shine		0 : Not shine	1 : Shine
Prediction	0 : Not shine	TN = 130	FN = 26
	1 : Shine	FP = 7	TP = 30

		Reality	
Confusion Matrix Sunrise		0 : Not sunrise	1 : Sunrise
Prediction	0 : Not sunrise	TN = 92	FN = 2
	1 : Sunrise	FP = 35	TP = 68

$$Recall(i) = \frac{tp(i)}{tp(i) + fn(i)}$$

- ♣ Cloudy : 65%
- ♣ Rain : 59%
- ♣ Shine : 54%
- ♣ Sunrise : 97%

# L' Approche “Per class”

- Les métriques “per class” permettent de comprendre le détail des performances du modèle sur chaque classe individuellement. Dans certains cas cependant, on peut préférer une métrique qui résume les performances sur l'ensemble des classes, pour comparer des modèles entre eux par exemple.

=> L'approche “per class” est alors insuffisante et il est nécessaire de faire appel aux méthodes “macro” et “micro”.

# L' Approche “Macro classique”

- L'approche “macro” se base sur **les métriques obtenues avec l'approche “per class”** pour les résumer en une seule valeur. On peut décomposer cette approche en 2 étapes :
  1. Le calcul des métriques “per class”.
  2. Le résumé des métriques “per class” en calculant une moyenne.
- Pour l'exemple météo, le macro-recall sera une moyenne des recalls de chaque classe

$$\text{Recall} = (65\% + 59\% + 54\% + 97\%) / 4 = 68\%$$

*Recall(i)*

- ♣ Cloudy : 65%
- ♣ Rain : 59%
- ♣ Shine : 54%
- ♣ Sunrise : 97%

# L' Approche “Macro-weighted”

- La variante “macro” pondérée revient à faire la moyenne pondérée des métriques “per class”. Chaque métrique a une importance proportionnelle à la proportion d’individus dans la classe correspondante.

**Métrique = MOY( la proportion d’individus de la classe (i)\* Métrique de la classe (i))**

# L' Approche “Macro-weighted”

		Reality	
Confusion Matrix Cloudy		0 : Not cloudy	1 : Cloudy
Prediction	0 : Not cloudy	TN = 121	FN = 21
	1 : Cloudy	FP = 20	TP = 39

		Reality	
Confusion Matrix Rain		0 : Not rain	1 : Rain
Prediction	0 : Not rain	TN = 137	FN = 16
	1 : Rain	FP = 3	TP = 23

		Reality	
Confusion Matrix Shine		0 : Not shine	1 : Shine
Prediction	0 : Not shine	TN = 130	FN = 26
	1 : Shine	FP = 7	TP = 30

		Reality	
Confusion Matrix Sunrise		0 : Not sunrise	1 : Sunrise
Prediction	0 : Not sunrise	TN = 92	FN = 2
	1 : Sunrise	FP = 35	TP = 68

● Pour l'exemple météo,

$$\text{NbExemples}(i) = \text{TP}(i) + \text{FN}(i)$$

♣ NbExemples Cloudy : 60 =>  $60/225=27\%$

♣ NbExemples Rain : 39 =>  $39/225=17\%$

♣ NbExemples Shine : 56 =>  $56/225=25\%$

♣ NbExemples Sunrise : 70 =>  $70/225=31\%$

(Nombre total d'exemples :  $60 + 39 + 56 + 70 = 225$ )

$$\text{Recall(pendéré)} = (65\% * 27\%) + (59\% * 17\%) + (54\% * 25\%) + (97\% * 31\%) / 4 = 71\%$$

# L'Approche "Micro"

- L'approche "micro" utilise directement la matrice de confusion multi-classe pour faire une matrice de confusion "synthétique". Elle reprend les termes de la matrice de confusion binaire : True Positive (TP), True Negative (TN), False Positive (FP) et False Negative (FN). Ces termes sont calculés de la manière suivante :

**Micro Confusion Matrix**

$TN = 0$	$FN = \sum_{i=1}^{n_{class}} FN_i$
$FP = \sum_{i=1}^{n_{class}} FP_i$	$TP = \sum_{i=1}^{n_{class}} TP_i$

*Valeurs utilisées pour le calcul des métriques micro.*

# L' Approche “Micro”

- Pour l'exemple météo

**Micro Confusion Matrix**

$TN = 0$	$FN = 65$
$FP = 65$	$TP = 160$

*Matrice confusion micro sur l'exemple météo.*

$$\text{Recall} = 160 / (160 + 65) = 71\%$$

# Conclusion

- “per class” : permet d'avoir le détail des performances sur chaque classe mais **multiplie le nombre de métriques à surveiller**
- “macro” classique : résume les performances obtenues par classe en restant **robuste au déséquilibre des classes**
- “macro-weighted” : résume les performances par classe en **préservant les proportions des classes**
- “micro” : résume les performances obtenues par individu et **prend en compte les proportions d'individus dans chaque classe**