# INTELLIGENCE ARTIFICIELLE PROJET - EXAMEN ORAL

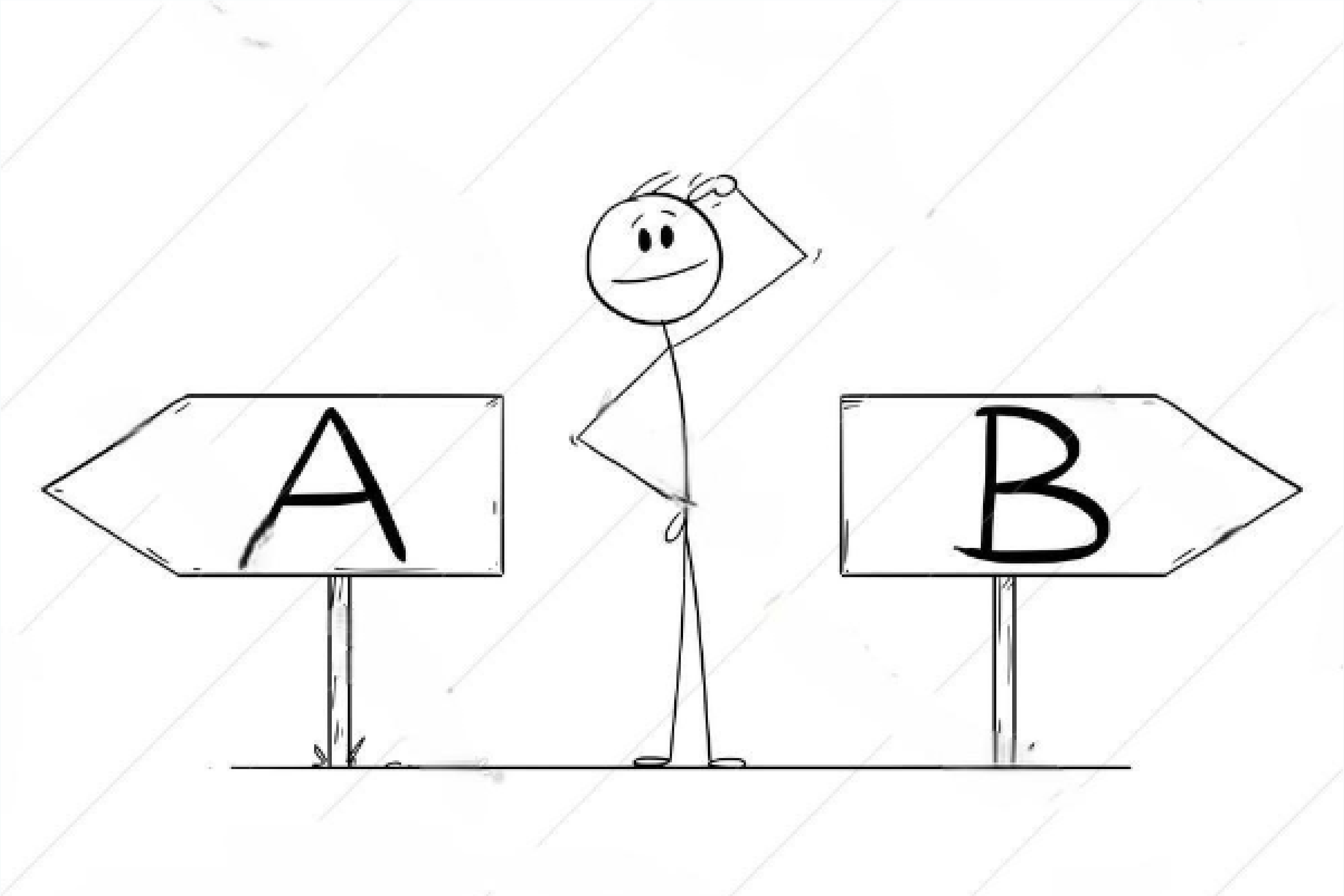
## REGRESSION LOGISTIQUE & NAIVE BAYES

Gregory Sedykh, Leandre Catogni,
Noah Peterschmitt, Noah Munz, Michel Donnet

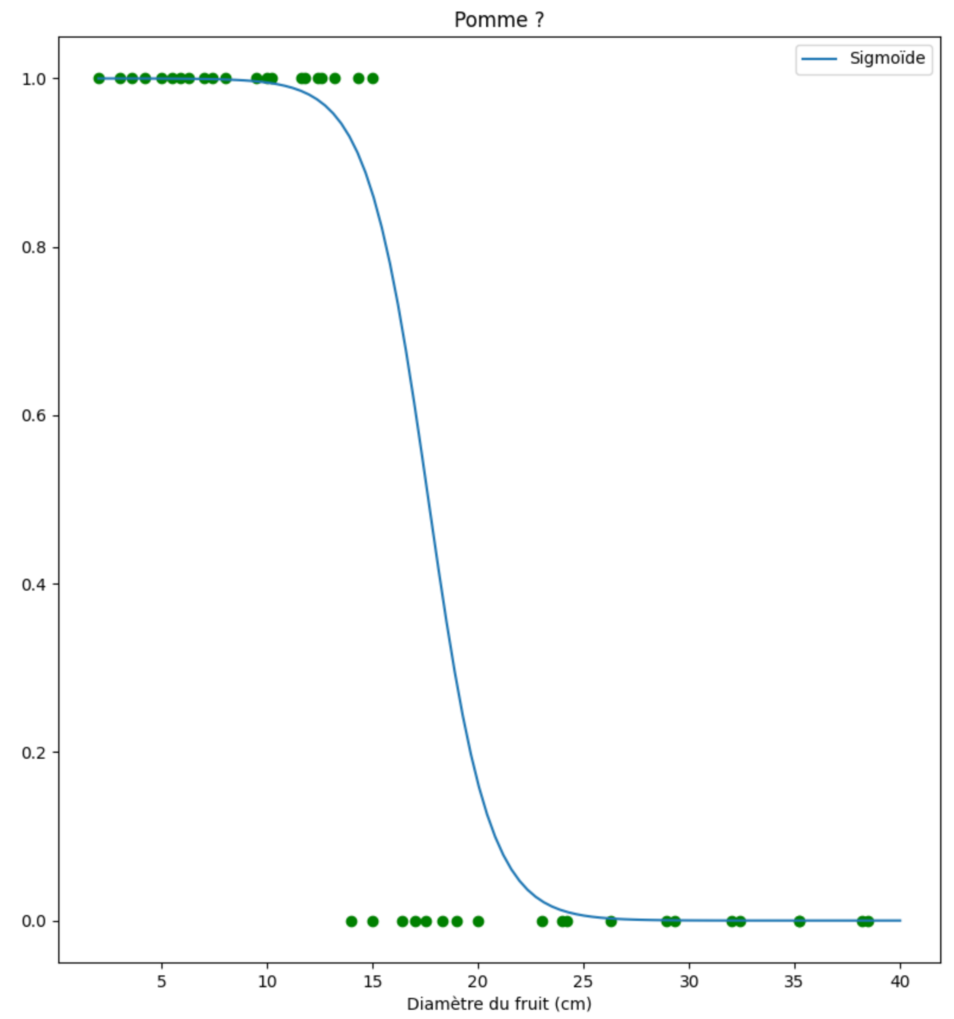
02 Fevrier 2024

# RÉGRESSION LOGISTIQUE

# RÉGRESSION LOGISTIQUE BINAIRE: PRINCIPE



# RÉGRESSION LOGISTIQUE BINAIRE: IDÉE



# RÉGRESSION LOGISTIQUE: FONCTION D'ESTIMATION

Fonction sigmoïde

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Caractérisiques:

- Comprise entre 0 et 1 $\Rightarrow$ probabilité !
- Point d'inflexion à 0.5

Idée:

- établir un seuil afin de prédire le label $Y$

# ENTRAÎNEMENT DU MODÈLE

**But:**

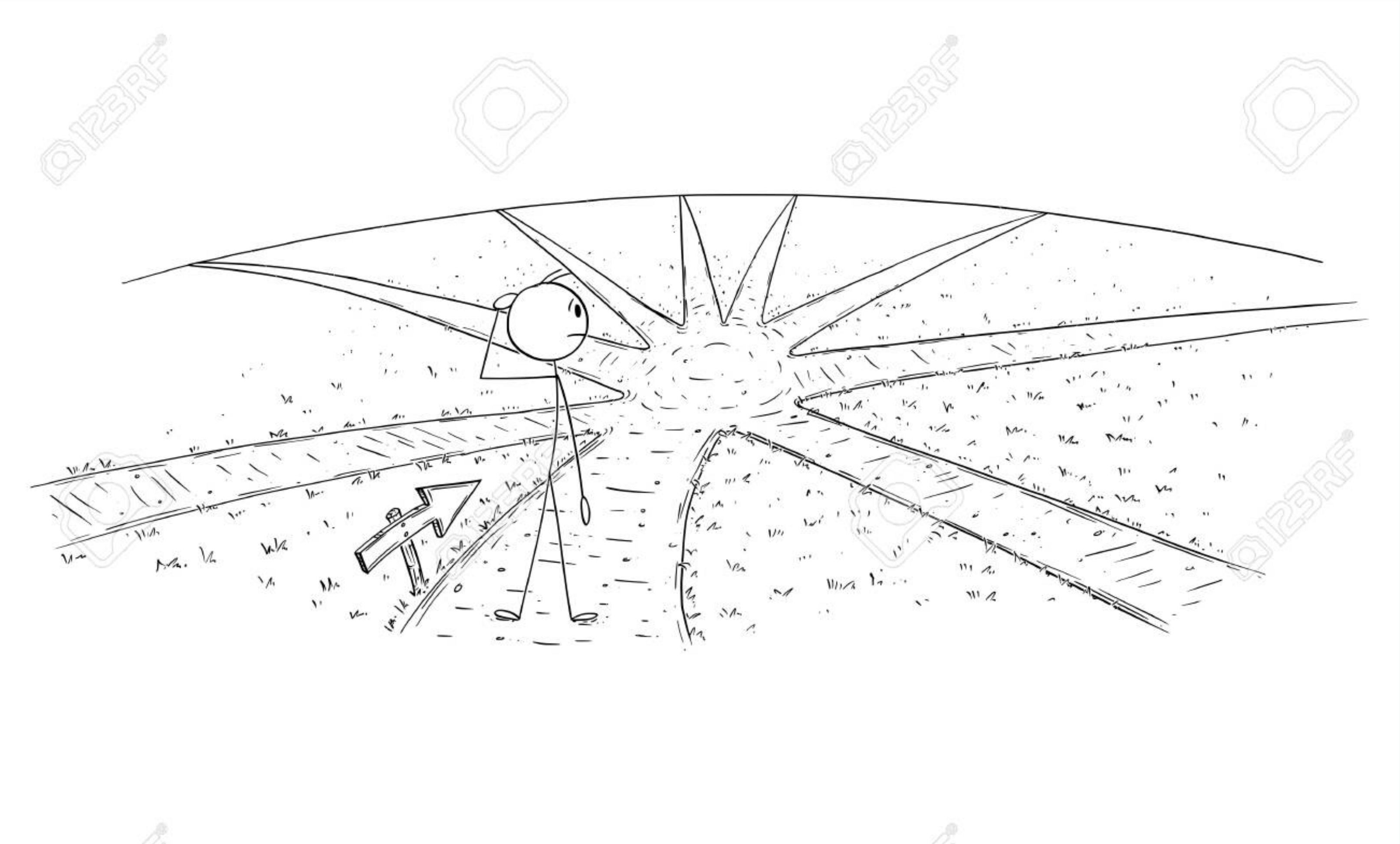
- maximiser la probabilité $P(Y = y|X)$ pour $y$ la valeur d'entrainement du label.

Mais on a la descente en gradient...

$\Rightarrow$ transformer le problème en problème de minimisation !

$\Rightarrow$ `Negative Logarithm Likelihood`

# RÉGRESSION LOGISTIQUE MULTINOMIALE: PRINCIPE



# GÉNÉRALISATION DE LA FONCTION SIGMOÏDE EN FONCTION SOFTMAX

$$P(Y = k|X) = \frac{1}{1 + e{-X\theta^T}} \rightarrow \frac{e^{X\theta_k^T}}{\sum_i^N e^{X\theta_i^T}}$$

# ENTRAÎNEMENT DU MODÈLE

Même principe que pour la régression logistique binaire

# NAIVE BAYES

# NAIVE BAYES

- Classification probabiliste conditionnelle: théorème de Bayes : $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$
  - Features $X_0, \cdots, X_{K-1}$, Classes $Y_0, \cdots, Y_{C-1}$ (ici $C = 3, K = 3$)
- Sepal length $\perp$ sepal width $\perp$ petal length $\perp$ petal width (Hypthèse d'indépendence naïve)
- Calculer la distribution empirique des features indépendemment des autres
  - $X_i$ continue $\Rightarrow X_i \sim \mathcal{N}(\mu_i, \sigma_i)$, i.e. pour notre colonne de data, on calcule, la moyenne et standard deviation $\Rightarrow$ on dit que ce sont les paramètres de la loi normale qui modélise comment les données de la colonne $i$ sont réparties
  - $X_i$ binaire $\Rightarrow X_i \sim \mathcal{B}(p_i)$
- On calcul l'impact qu'ont les V.A. de lois inférés de la répartition de chaque colonne sur le label que l'on veut prédire.
  E.g. Comment la répartition de $X_2$ (longueur du pétale) nous donne une information sur le type de fleur?
- Intuitivement $\Rightarrow$ Comment la probabilité que la longueure du pétale aie une certaine valeur influe sur le type de fleur $Y_i$?
- En se prenant l'information non pas donnée par la répartition de $X_2$ mais par la répartition de tous les $[X_{j\in[0,3]}] \Rightarrow$ on obtient le principe du classifier bayesien. (Chaque $X_j$ a un poids équivalent)

# NAIVE BAYES - FORMELLEMENT

- $P(cause|\text{effet}) = \frac{P(\text{effet}|cause)P(cause)}{P(\text{effet})}$
- $P(class|\text{donnée}) = \frac{P(\text{donnée}|class)P(class)}{P(\text{donnée})} = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$
- On aimerait (intuitivement) : Calculer les probabilités que notre label ait telle ou telle classe connaissant notre sample, et prendre le max $\tilde{y}$ i.e.

$$\tilde{y} = \arg\max_{y \in \mathcal{Y}} P(y|\mathbf{x})$$

- Ici on part du principe qu'on connaît $\mathbf{x} \Rightarrow$ perd principe de la prédiction puisque ça impose le fait que l'on doit avoir déjà observé *exactement* ce $\mathbf{x}$.
- On utilise que, le $y$ qui maximise la formule du théorème de bayes est aussi le $y$ qui maximise $P(\mathbf{x}|y)P(y)$, (car $P(y|x) \propto P(x)P(y)$)

# NAIVE BAYES - CONCLUSION

- $\tilde{y} = \arg\max_{y \in \mathcal{Y}} P(y|\mathbf{x})$
- $P(y|x) \propto P(x)P(y)$
- Donc $\tilde{y} = P(\mathbf{x}|y)P(y)$
- Sepal length $\perp$ sepal width $\perp$ petal length $\perp$ petal width (Hypthèse d'indépendence naïve)
- $$P(\mathbf{x}|y) = P(x_1|y)\prod_{k=2}^{K} P(x_k|x_{k-1}, \cdots, x-1, y) = P(x_1|y)\prod_{k=2}^{K} P(x_k|y) = \prod_{k=1}^{K} P(x_k|y)$$
- En on conclut donc que

$$\boxed{\tilde{y} = \arg\max_{y \in \mathcal{Y}} \left[\prod_{k=1}^{K} P(x_k|y)\right]}$$

- On prédit la classe $\tilde{y}$ d'un sample $\mathbf{x}$, en calculant le maximum de la probabilité conditionnelle $P(\mathbf{x}|classe)$ pour chaque classe.

# EVOLUTION DES MÉTRIQUES

# CONTEXTE :

Les mesures d'évaluations permettent d'analyser les performances de prédictions d'un modèle, à l'aide d'un "test set".

**Rappel :**

Test set : Données dont on connaît les labels exacts, que l'on cachera afin de tester les prédictions faites par le modèle.

# DÉFINITIONS UTILES :

Dans le contexte multinomial considérons un label positif et des labels nétagifs (ie. ceux qui diffèrent du label positif), on a alors :

- **True positive (TP)** : Labels positifs qui ont été correctement prédits comme tel
- **False Positive (FP)** : Labels négatifs prédits comme positifs
- **True negative (TN)** : Labels négatifs prédits comme négatifs
- **False Negative (FN)** : Labels positfs prédits comme négatifs

# PRÉCISION

- **Intuition** : Proportion des prédictions positives correctes (TP) par rapport à toutes les prédictions positives (TP + FP).
- **Cas multinomial** : Moyenne des précisions pour chaque label positif possible.
- **Définition** :

$$\frac{1}{|L|} \cdot \sum_{l \in L} \frac{TP_l}{TP_l + FP_l}$$

où $L$ est l'ensemble des labels

# RAPPEL :

- **Intuition** : Proportion des prédictions positives correctes (TP) par rapport aux positifs réels (du test set) (TP + FN).
- **Cas multinomial** : Moyenne des rappels pour chaque label positif possible.
- **Définition Formelle** :

$$\frac{1}{|L|} \cdot \sum_{l \in L} \frac{TP_l}{TP_l + FN_l}$$

où $L$ est l'ensemble des labels

# F1 SCORE :

- **Intuition** : Combinaison de la précision et du rappel (moyenne harmonique)
- **Définition** :

$$\frac{2}{rappel^{-1} + precision^{-1}} = 2 \cdot \frac{precision \cdot rappel}{precision + rappel}$$

# ACCURACY :

- **Intuition** : Proportion des prédictions correctes parmi l'ensemble total des prédictions.
- **Définition** :

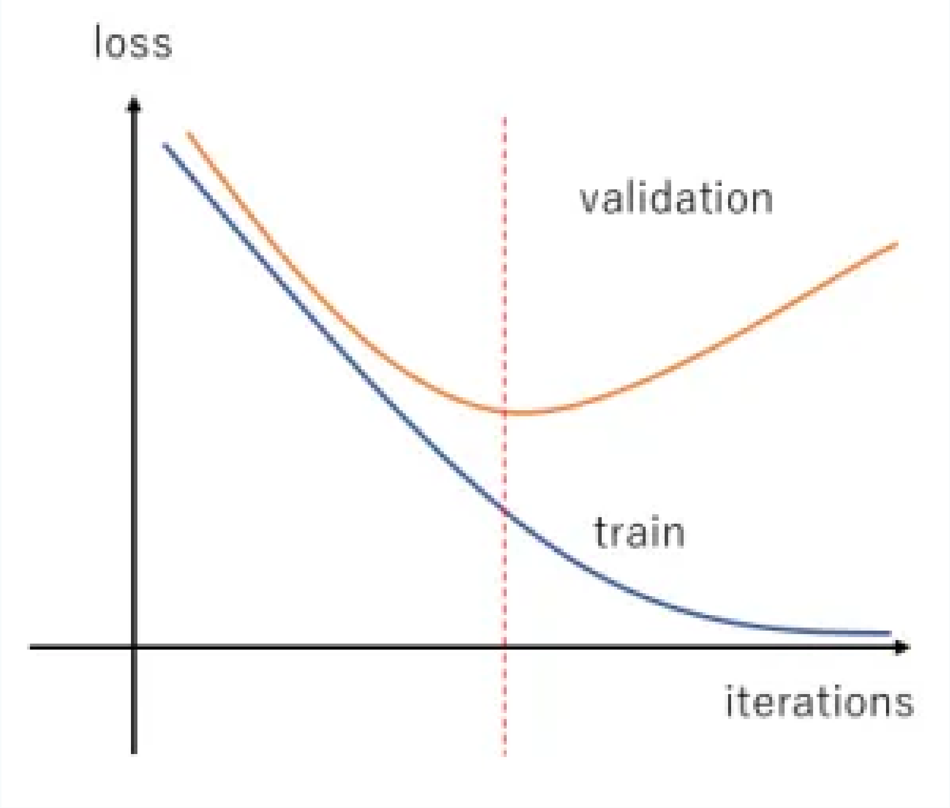
$$\frac{\text{Nombre de predictions correctes}}{\text{Nombre total de prediction}} = \frac{TP + TN}{TP + TN + FP + FN}$$

# OVERFITTING

On ne veut pas apprendre le bruit des données d'apprentissage !

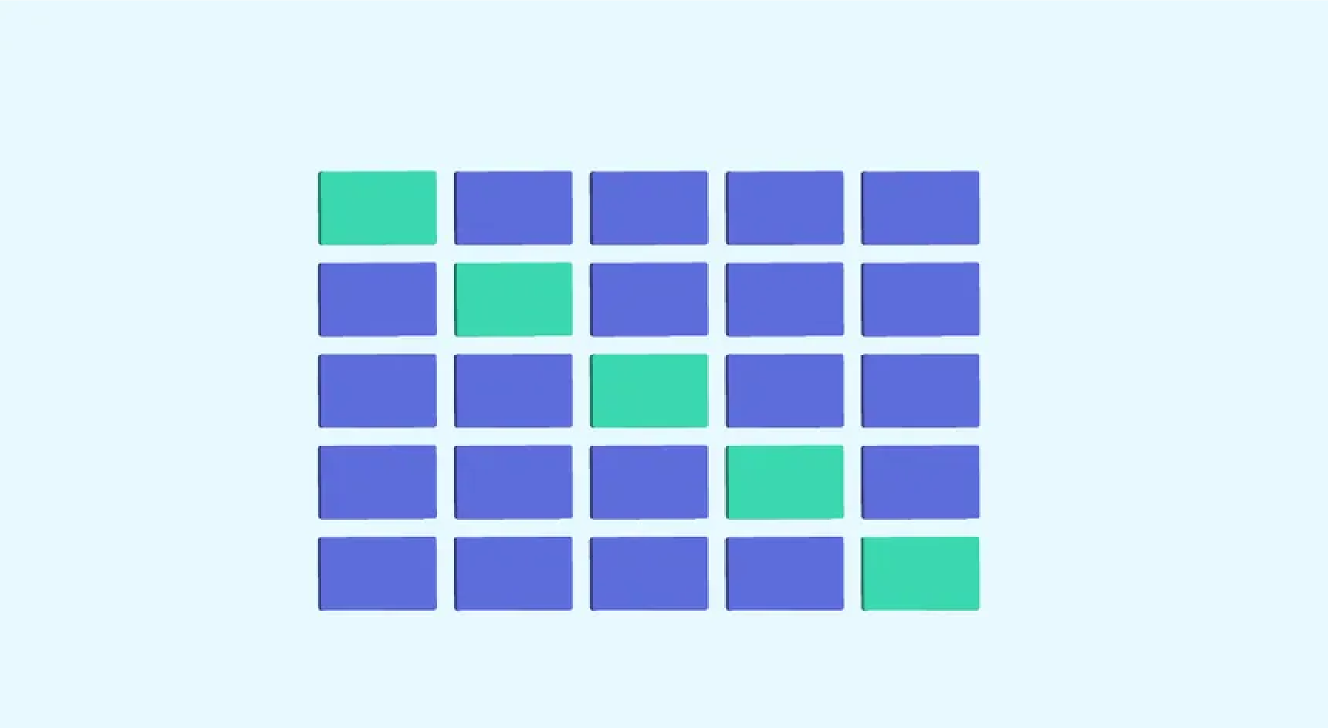
# SUR-APPRENTISSAGE: EXEMPLE

# SUR-APPRENTISSAGE: GRAPHIQUE



# COMMENT ÉVITER LE SUR-APPRENTISSAGE ?

Validation croisée !



# AUTRES TECHNIQUES ?

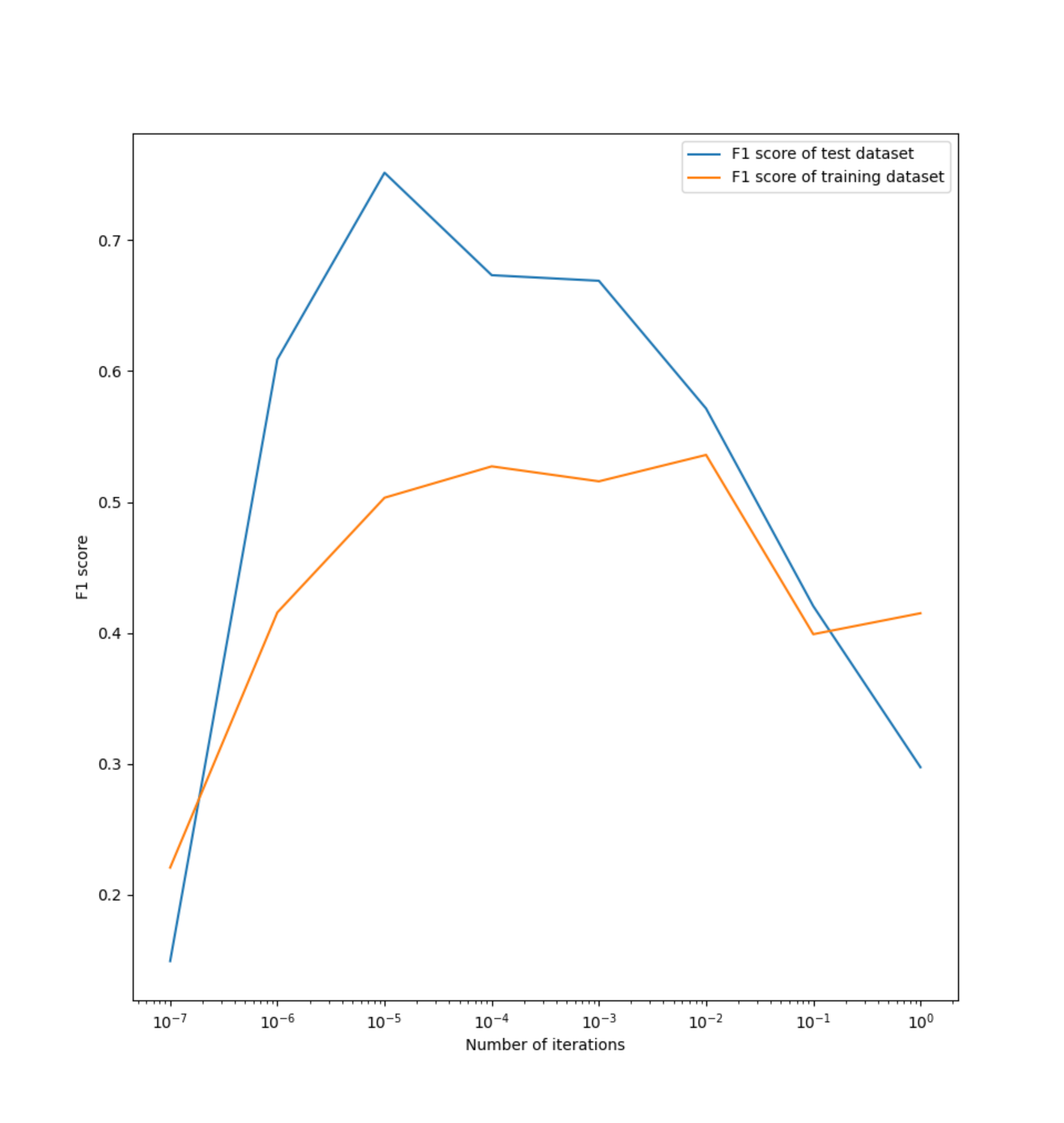
- Ajout données d'apprentissage modifiées (pour plus de généralisation...)
- Retirer des caractéristiques
- ...

# CONCRÊTEMENT, DANS LE PROJET

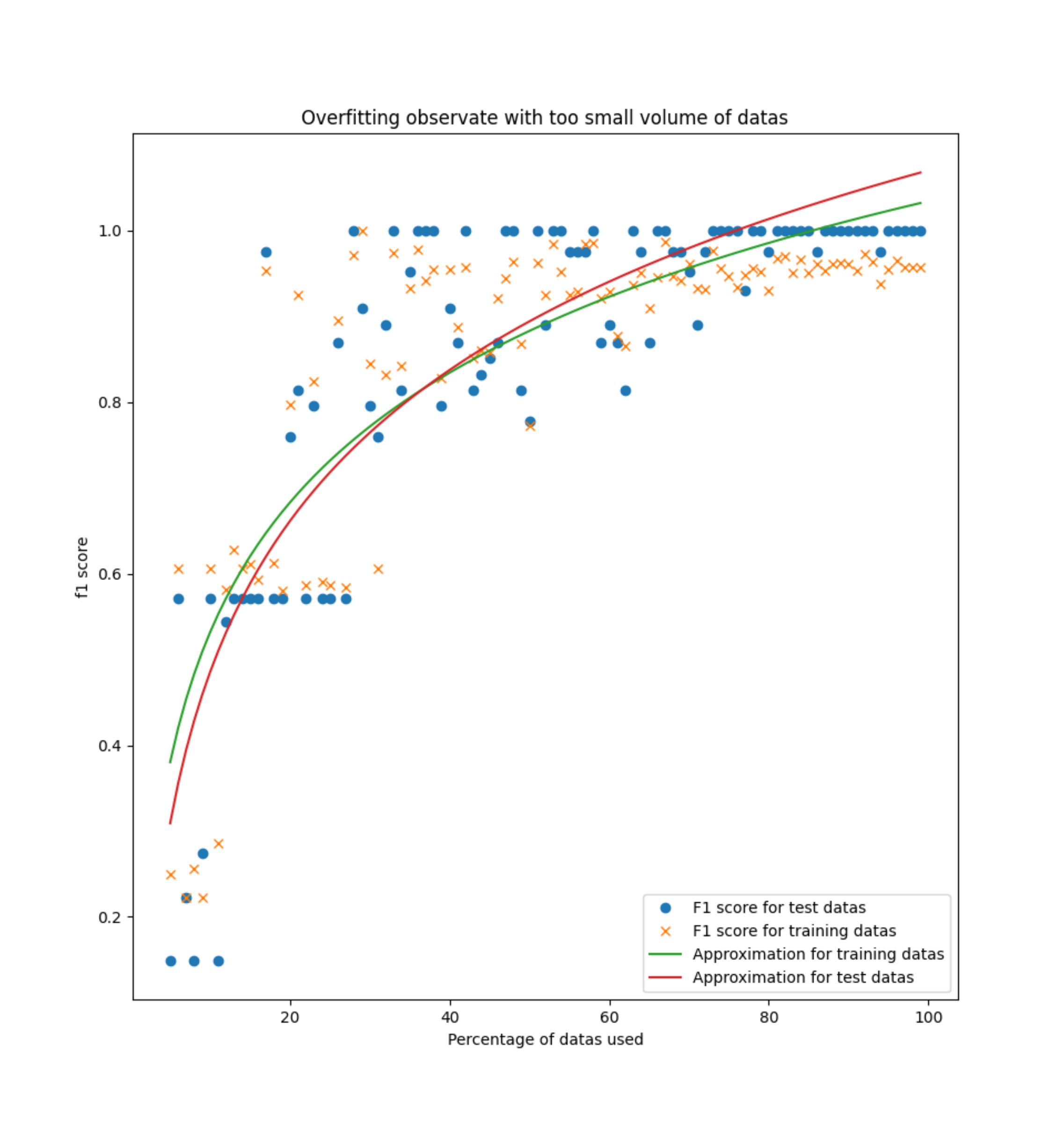
Dans le projet, pour montrer le phénomène de sur-apprentissage:

- Ajout de bruits aux données d'apprentissage
- Volume réduit de données
- Modification du nombre d'itérations

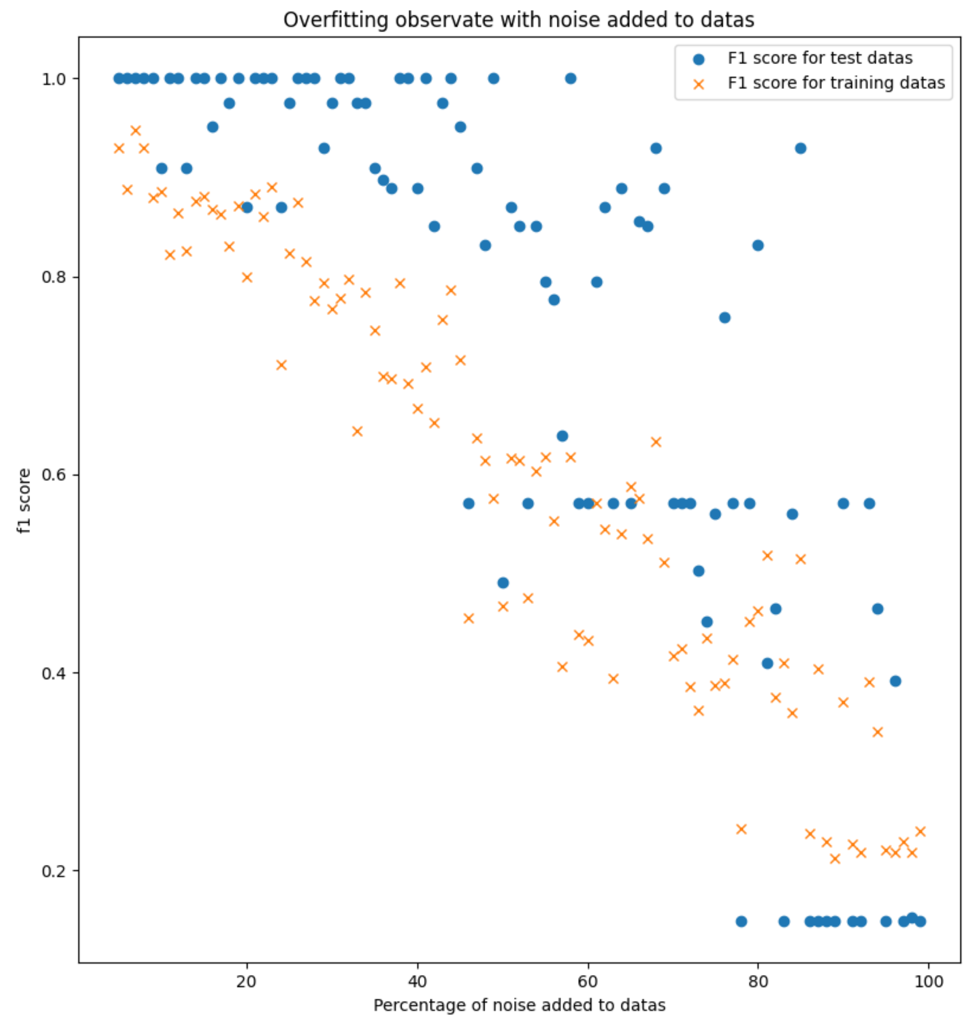
# RÉSULTATS OBTENUS



# RÉSULTATS OBTENUS



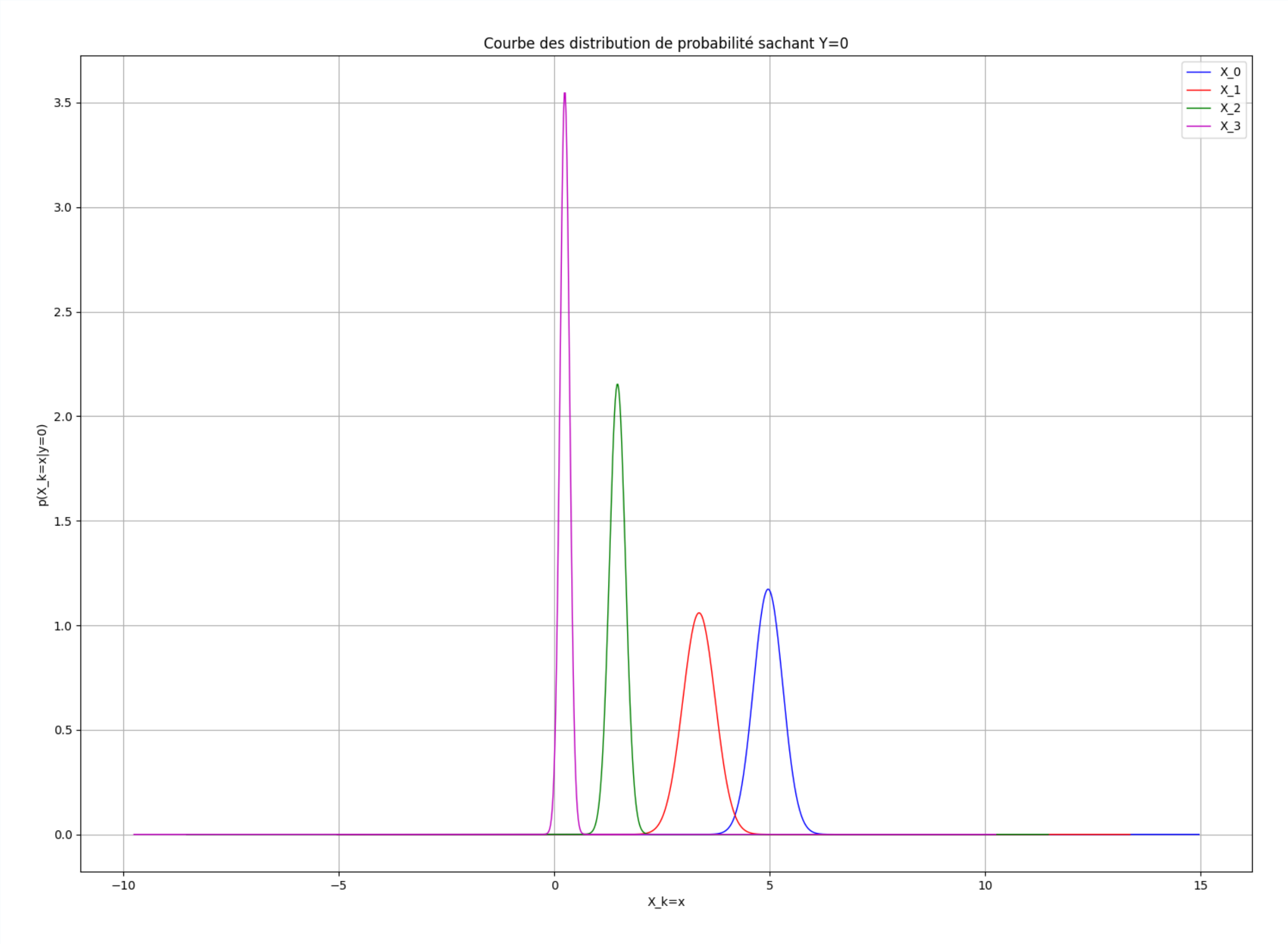
# RÉSULTATS OBTENUS



# ANALYSE DES FEATURES

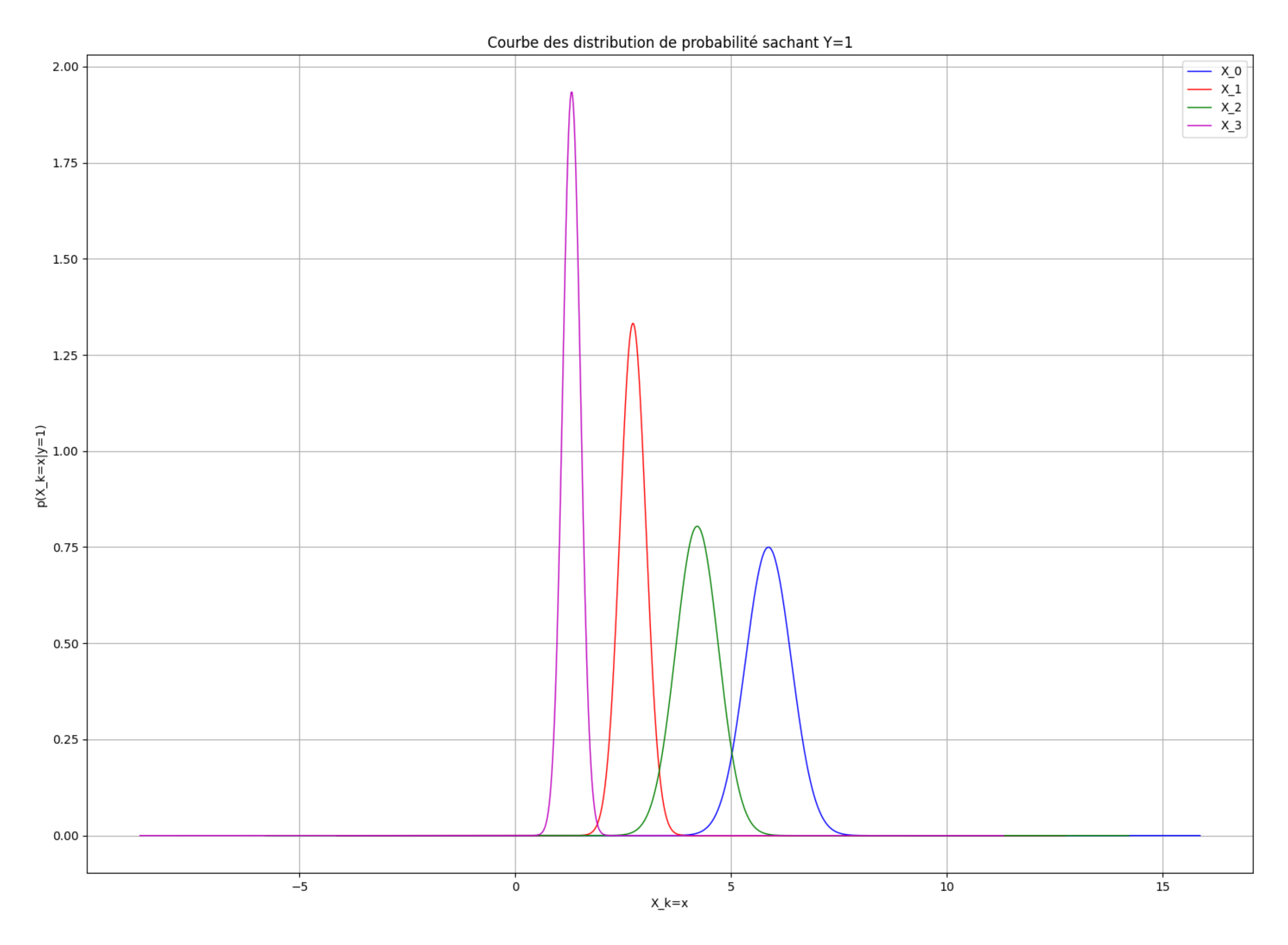
# ANALYSE DES FEATURES

- Pic bleu et rouge faible $\Rightarrow X_0$ et $X_1$ - ont moins d'influence sur la classe.
- Chevauchement faible $\Rightarrow$ peu interdépendance



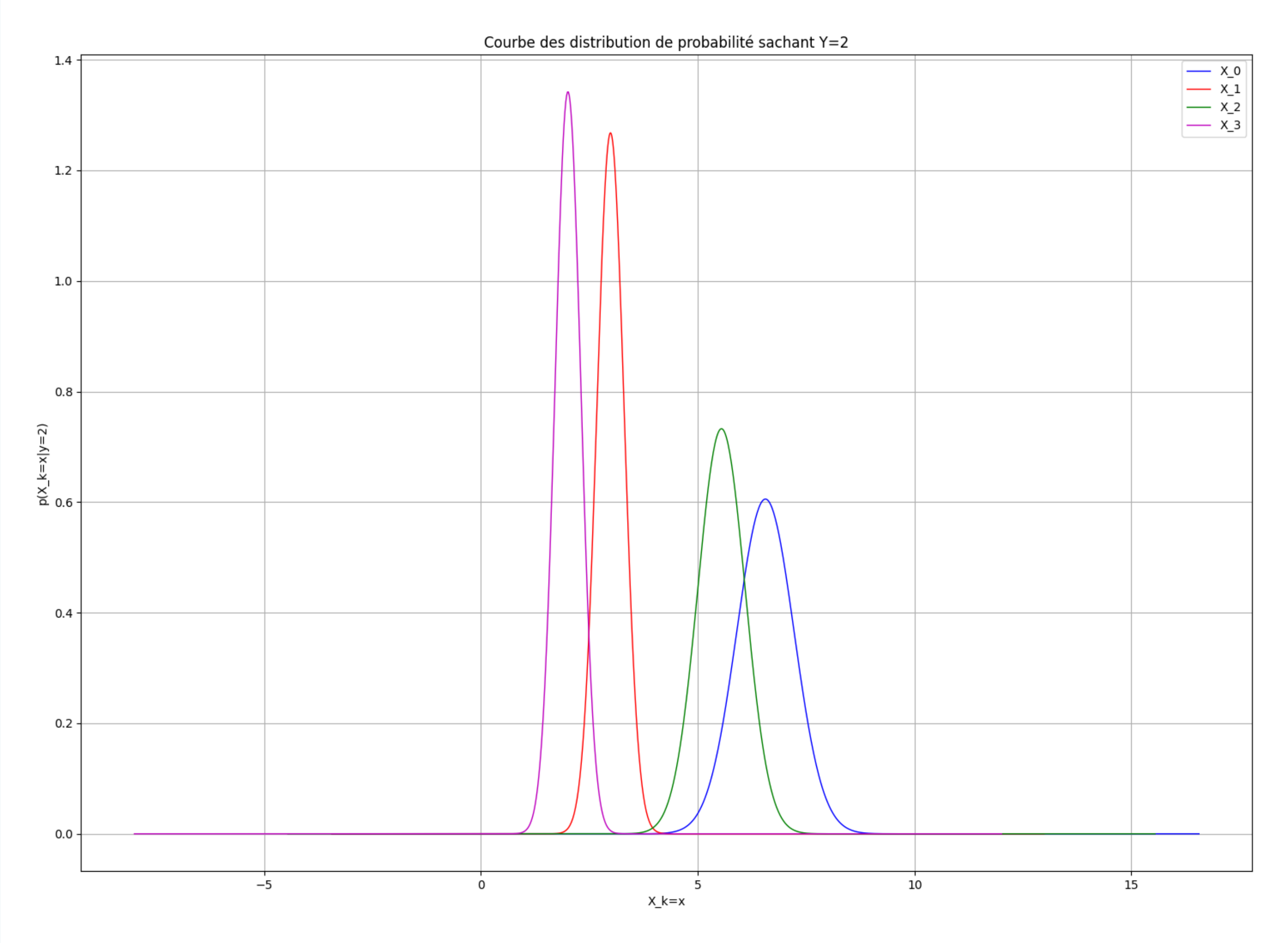
# ANALYSE DES FEATURES

- pic bleu et vert faible $\Rightarrow X_0$ et $X_2$ - ont moins d'influence sur la classe.
- Chevauchement fort entre bleu et vert et vert et rouge $\Rightarrow$ interdépendance entre $X_1$ et $X_2$ et $X_0$ et $X_2$



# ANALYSE DES FEATURES

- Pic bleu et vert faible $\Rightarrow X_0$ et $X_2$ - ont moins d'influence sur la classe.
- Chevauchement fort entre bleu et vert et rouge et magenta $\Rightarrow$ interdépendance entre $X_1$ et $X_3$ et entre $X_0$ et $X_2$



# FONCTIONS UTILISÉES

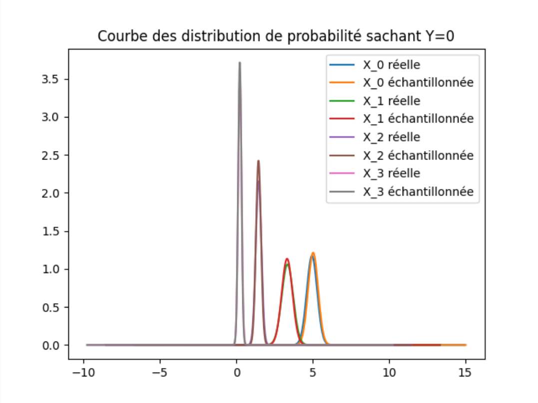
`plot_util.py` : modification de la fonction plot_vs afin de pouvoir comparer jusqu'à 4 fonctions.

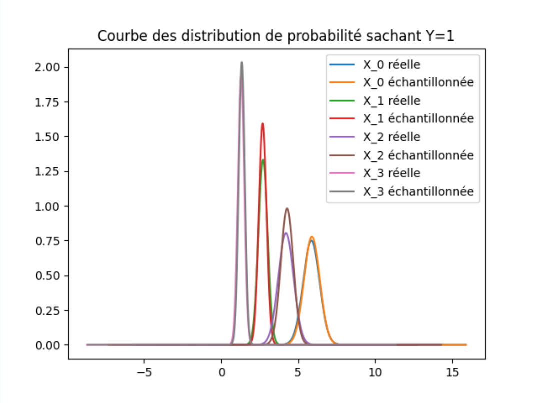
`feature_analyse_plot.py` : affichage pour chaque classe les courbes des normal PDF de chaque données.

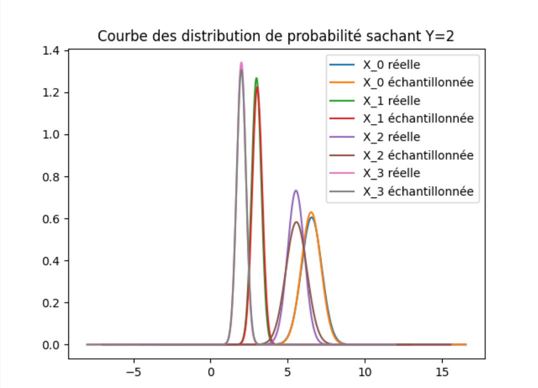
# SAMPLING

# SAMPLING

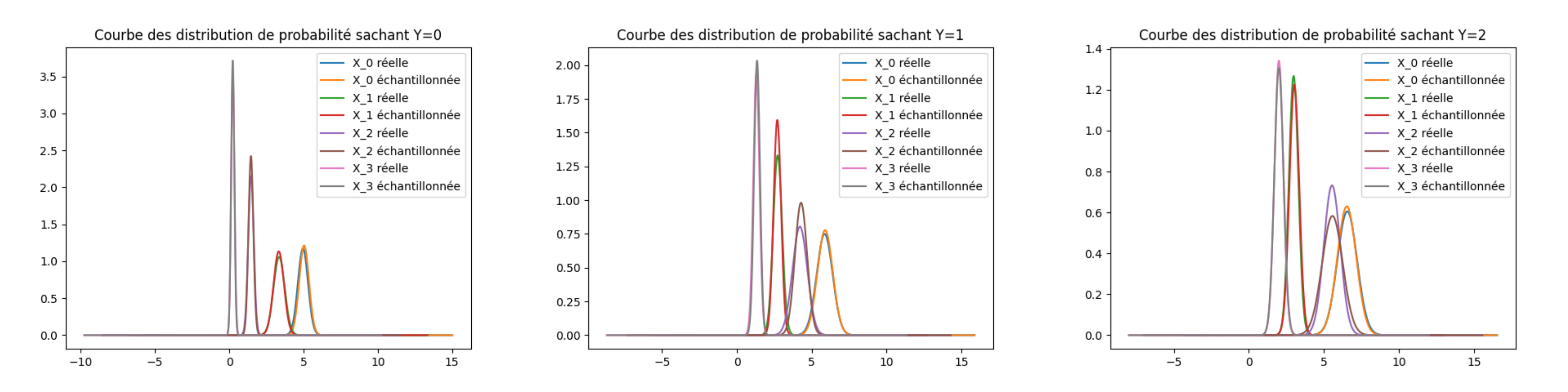
- Une fois que les paramètres des classes sont obtenus en supposant l'indépendance des variables, on échantillone de nouvelles données afin de comparer les résultats obtenus avec les données d'origine.
- L'échantillonage est fait dans le fichier `sampling.py`.
- On fait 50 échantillons pour chaque classe, à partir des paramètres des distributions obtenus dans la section précédente.
- On obtient les résultats suivants (la moyenne et l'écart-type sont donnés pour chaque classe et chaque variable) :



# GRAPHS PAR CLASSE ($Y \in \{0, 1, 2\}$)



# RÉSULTATS

## COMPARAISON AVEC SKLEARN

# NAIVE BAYES

## NOTRE NAIVE BAYES

- Precision: 0.976
- Recall: 0.974
- Accuracy: 0.977
- F1 score: 0.975

## SKLEARN NAIVE BAYES

- Precision: 0.976
- Recall: 0.974
- Accuracy: 0.977
- F1 score: 0.975

# LOGISTIC REGRESSION

## NOTRE LOGISTIC REGRESSION

- Precision: 0.850
- Recall: 0.846
- Accuracy: 0.866
- F1 score: 0.848

## SKLEARN LOGISTIC REGRESSION

- Precision: 0.976
- Recall: 0.974
- Accuracy: 0.977
- F1 score: 0.975

# CONCLUSION