MACHINE LEARNING SUPERVISED ALGORITHMS

KNN & Native Bayes (NB)

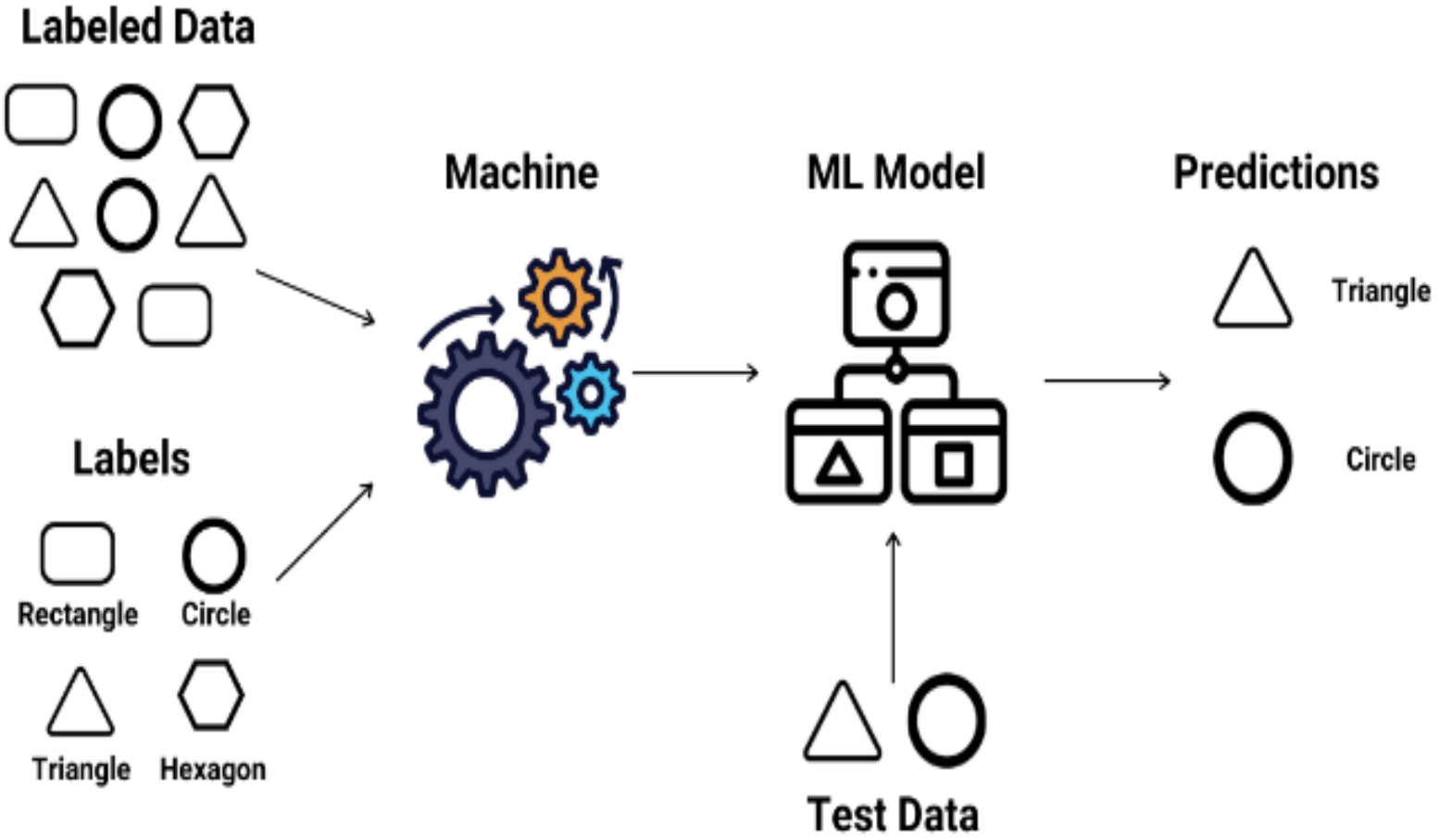
Introduction

- L'apprentissage supervisé est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés, au contraire de l'apprentissage non supervisé. C'est-à-dire que les prédictions sont réalisées à partir de données historiques.
- On distingue les problèmes de régression des problèmes de classement :

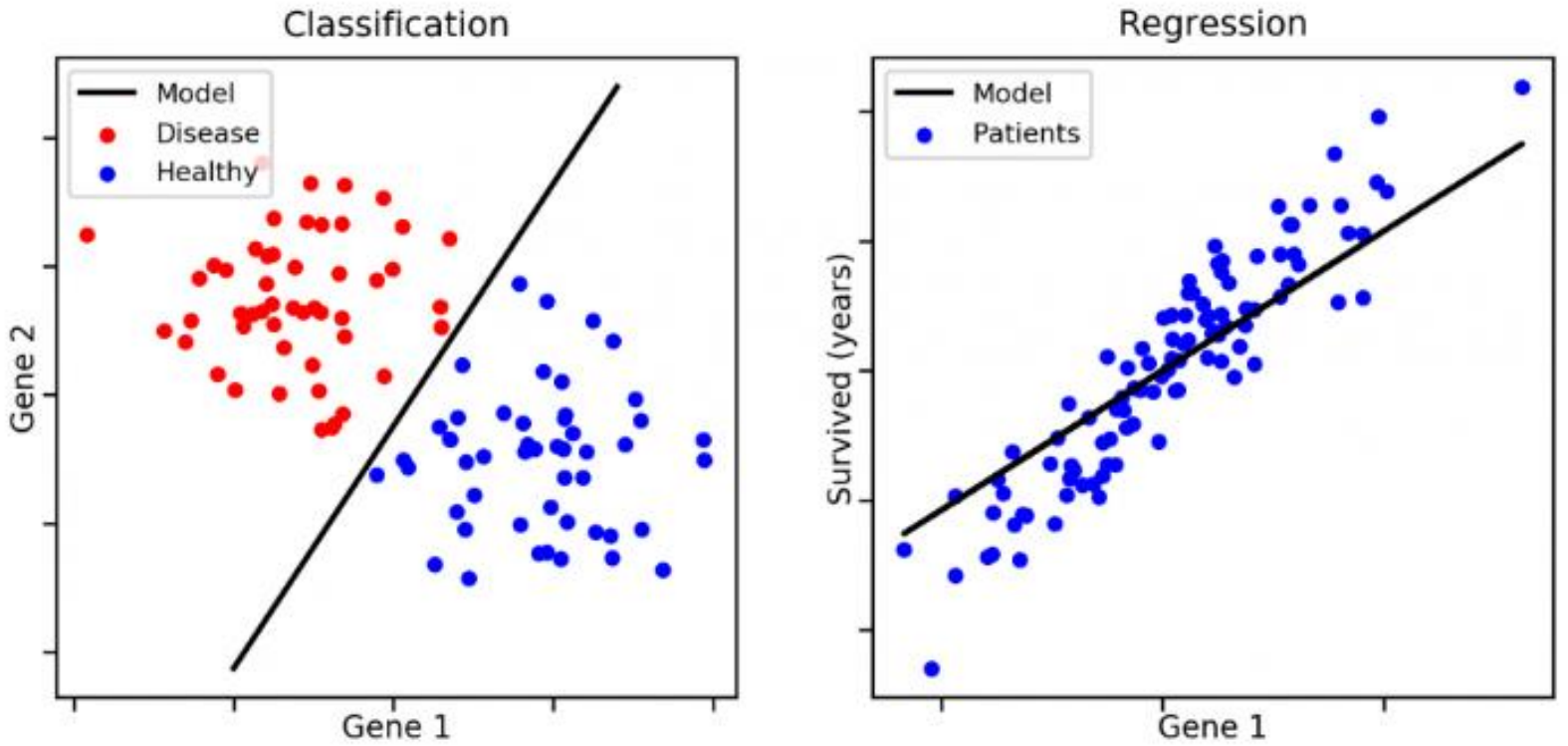
➢ *Prédiction d'une variable quantitative : problème de régression.*

➢ *Prédiction d'une variable qualitative : problème de classification.*

Apprentissage Supervisé



La Classification et la Régression



- Dans l'apprentissage supervisé, on a deux types d'algorithmes :
  - Les algorithmes de **régression**, qui cherchent à prédire une valeur continue, une quantité (variable quantitative).
  - Les algorithmes de **classification**, qui cherchent à prédire une classe/catégorie (variable qualitative).

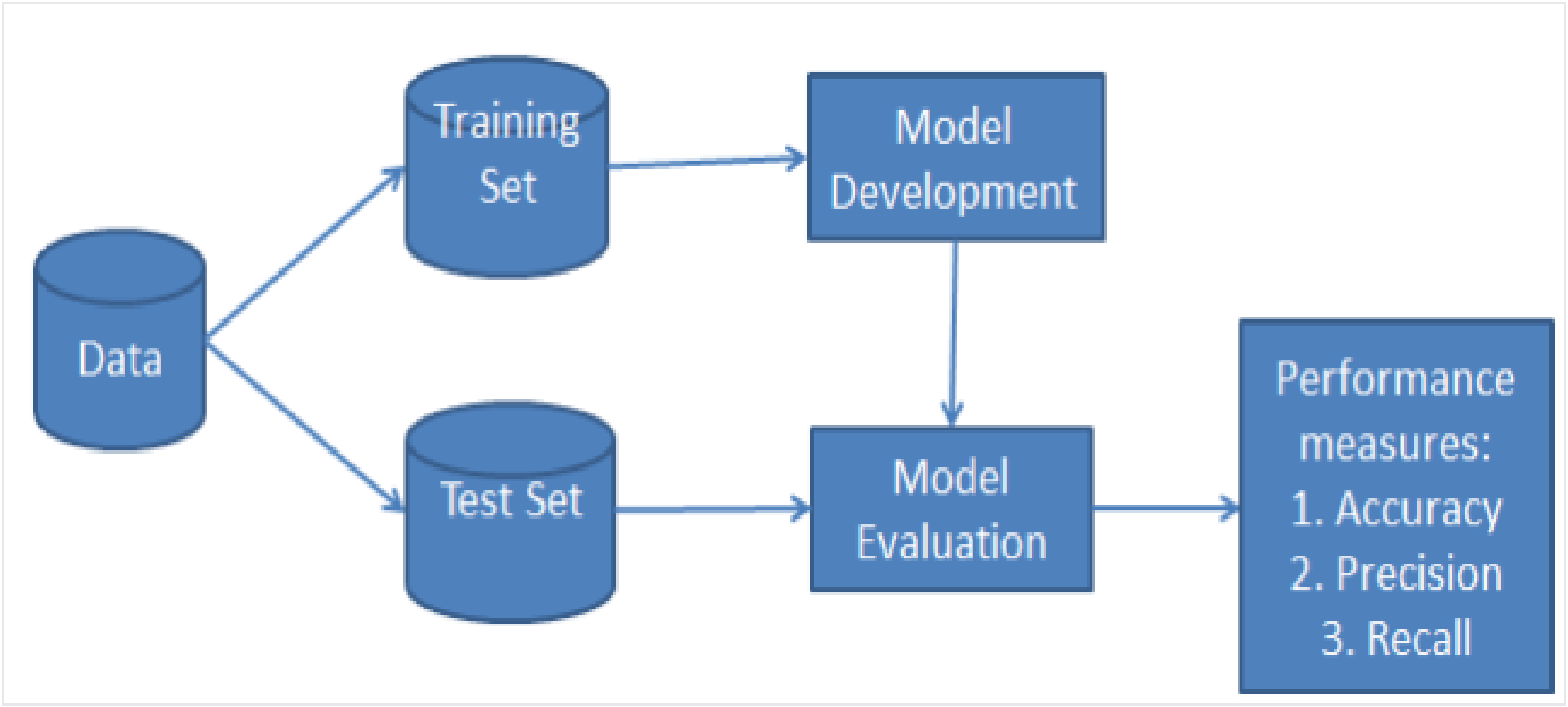
La Classification

- On parle d'un problème de classification quand la variable à prédire est *une variable discrète* (variable ne pouvant prendre qu'un nombre fini de valeurs – ex. 1 ou 2, malade ou pas malade) (une catégorie).
- *Exemples* :
  - En finance et dans le secteur bancaire pour la détection de la fraude par carte de crédit (fraude, pas fraude).
  - Détection de courrier électronique indésirable (spam, pas spam).
  - Dans le domaine du marketing utilisé pour l'analyse du sentiment de texte (heureux, pas heureux).
  - En médecine, pour prédire si un patient a une maladie particulière ou non.

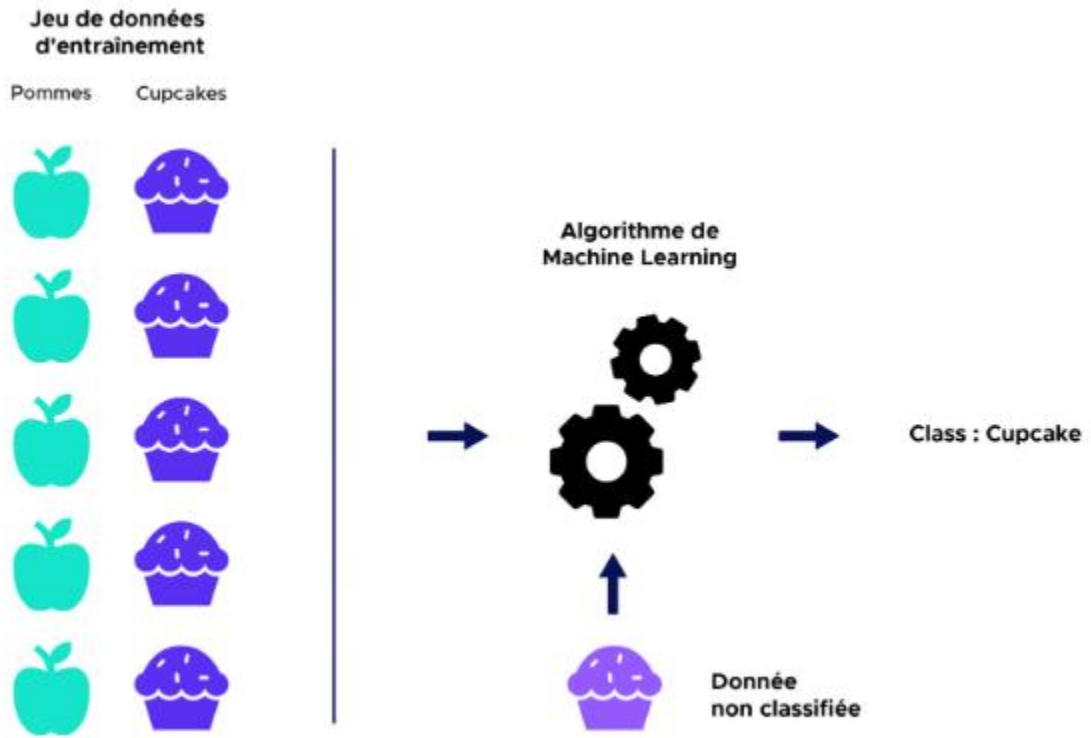
Les algorithmes de **classification**

- La classification supervisée est la catégorisation algorithmique d'objets.
- En se basant sur des modèles statistiques, l'algorithme développé doit prédire à quelle classe appartient la donnée.
- Cette classification peut compter deux dimensions (binaires) ou plus (multi-classes).
- Dans cette classification,
  - ✓ on connaît déjà le nombre de groupes qui existent dans la population;
  - ✓ on connaît le groupe auquel appartient chaque observation de la population;
  - ✓ on veut classer les observations dans les bons groupes à partir de différentes variables.
  - ✓ On peut ensuite utiliser une règle de classification pour prédire les groupes auxquels appartiennent de nouvelles observations.

Classification Workflow



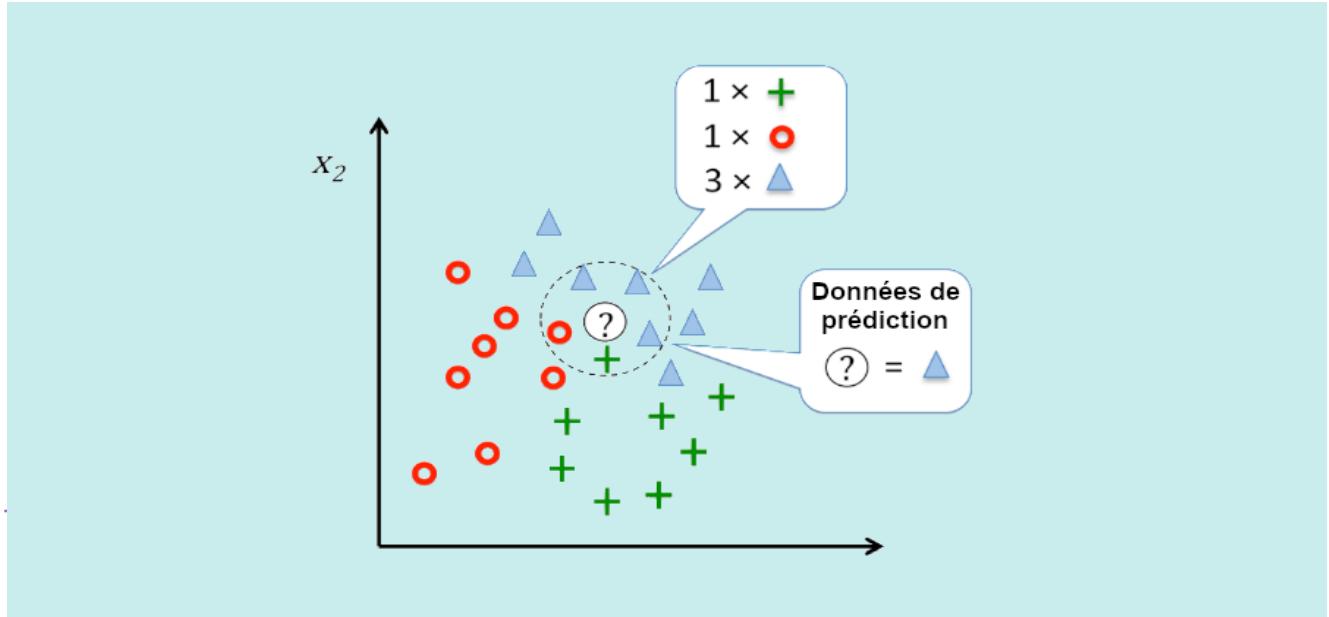
- Voici une illustration simplifiée :



- Plusieurs algorithmes d'apprentissage automatique supervisé qui traitent des problèmes de classification :
  - K-NN (K-Nearest Neighbors)
  - Le classificateur bayésien naïf (Naïve Bayes Classifier)
  - K-means (K-moyennes)
  - Machine à vecteurs de support (SVM)
  - Arbre de décision
  - …..

1 – K-NN (K-Nearest Neighbors)

- Le K-nearest neighbors (*ou algorithmes des plus proches voisins*) peut être utilisé à la fois comme **algorithme de régression ou de classification**. Mais c'est souvent dans cette deuxième hypothèse qu'il est utilisé.
- L'idée est alors de classifier les variables d'un jeu de données en **analysant les similitudes entre elles**.
- Pour cela, le KNN utilise un graphique et calcule la distance entre les différents points.
- Ceux qui sont les plus proches sont enregistrés dans la même catégorie.



Algorithme KNN

- **Étape 1 :** Sélectionnez le nombre K de voisins
- **Étape 2 :** Calculez la distance Du point non classifié aux autres points.

$$\sum_{i=1}^{n} |x_i - y_i|$$

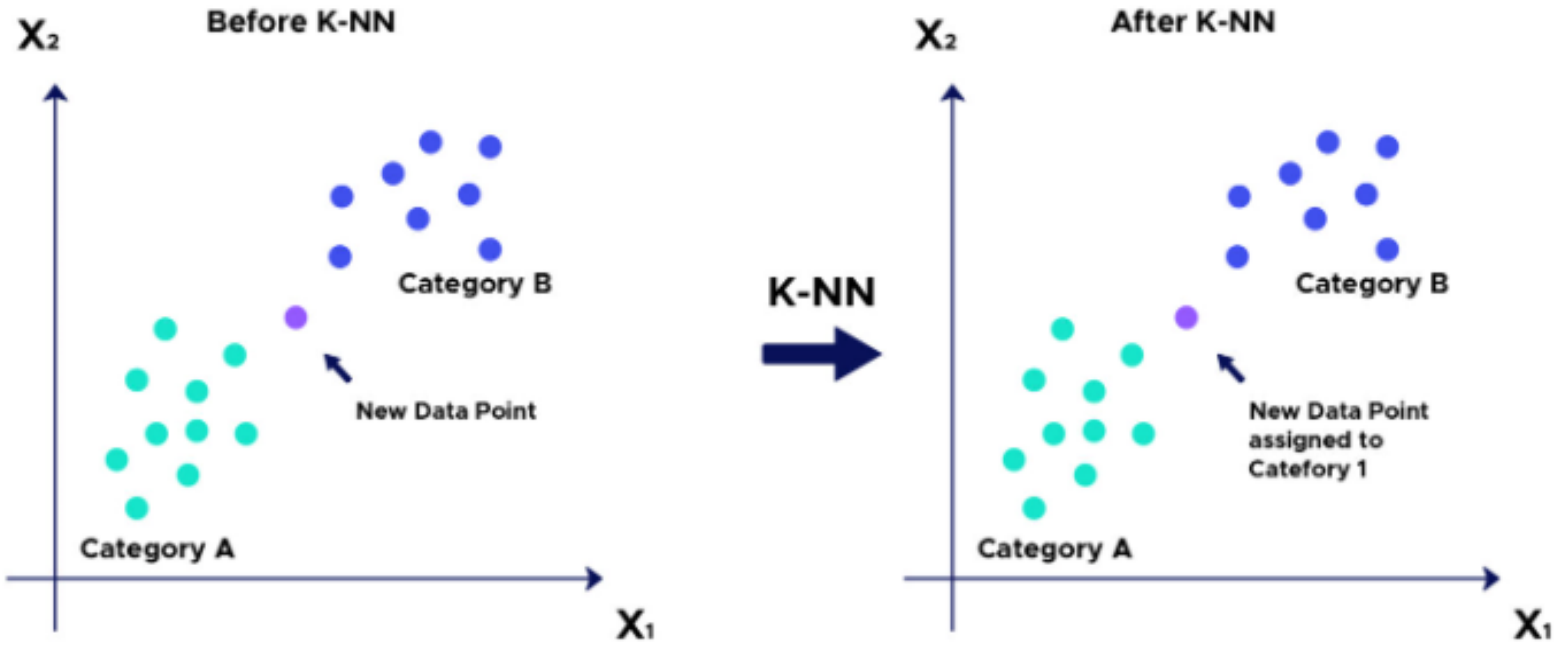
Manhattan

$$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Euclidienne

- **Étape 3** : Prenez les K voisins les plus proches selon la distance calculée.
- **Étape 4** : Parmi ces K voisins, comptez le nombre de points appartenant à chaque catégorie.
- **Étape 5** : Attribuez le nouveau point à la catégorie la plus présente parmis ces K voisins.
- **Étape 6** : Notre modèle est prêt

- **Étape 6** : Notre modèle est prêt :



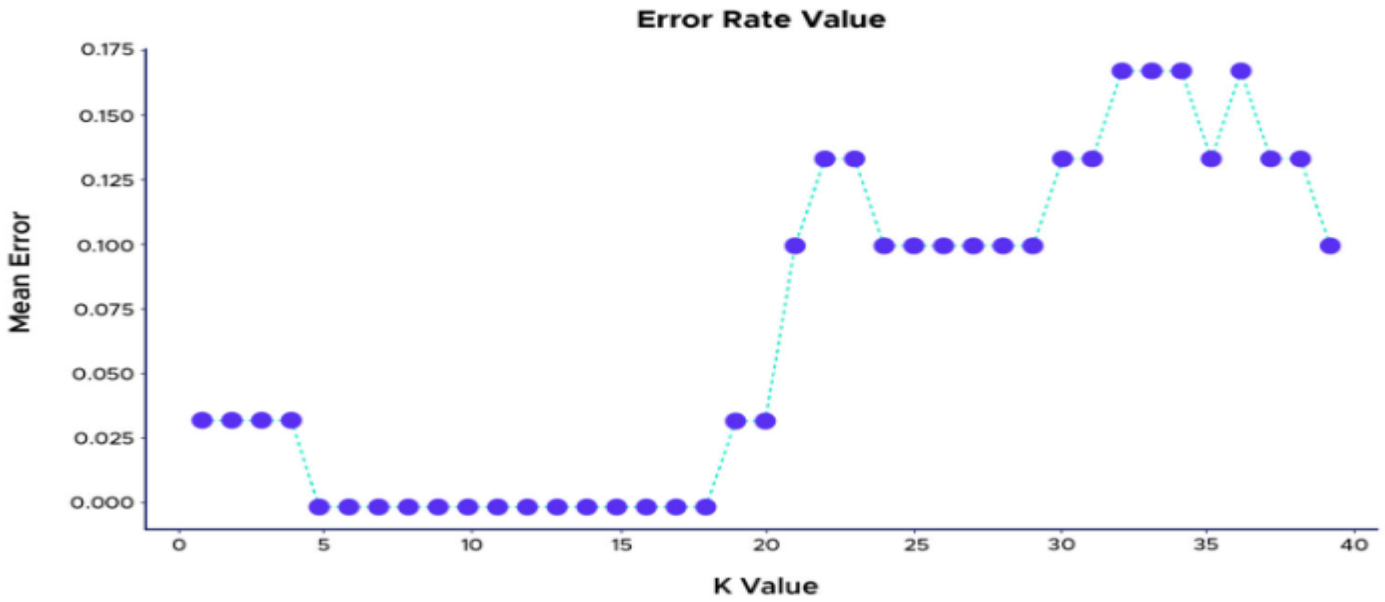
Apprendre l'algorithme KNN

KNN : Exemple d'utilisation

- Nous pouvons à présent nous intéresser à un exemple d'utilisation de l'algorithme des K plus proches voisins. Grâce à la librairie Scikit-Learn, nous pouvons importer la fonction **KKNeighborsClassifier** que nous utiliserons sur le jeu de donnée IRIS.

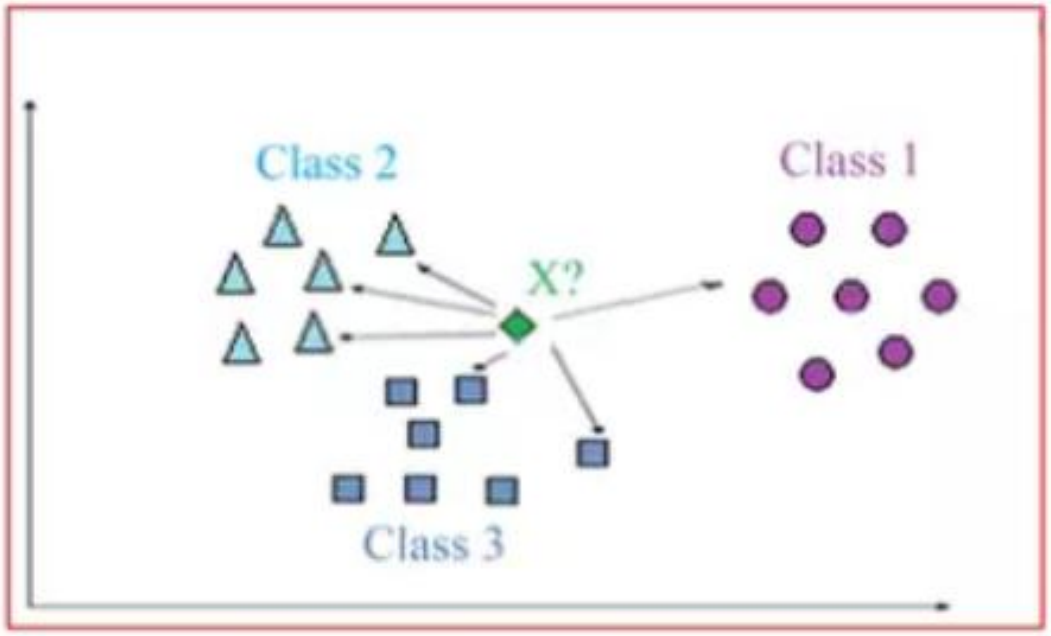
	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

- Grâce à l'algorithme KNN, nous obtenons un excellent taux de bonne classification des plantes proches des 100%. On peut également s'intéresser à un moyen de **choisir le K pour lequel la classification sera la meilleure.** Une façon de le trouver consiste à tracer le graphique de la valeur K et le taux d'erreur correspondant pour l'ensemble de données :



Le meilleur taux de prédiction est obtenu pour un K entre 5 et 18. Au-dessus de cette valeur, on peut observer un phénomène nommé Overfitting ou Sur apprentissage.

- Voici un exemple de méthode KNN avec trois voisins.
- Le but étant d'identifier à quelle classe appartient le point X.



Quelques Applications du KNN

- Il peut être utilisé dans des **technologies comme l'OCR (Optical Character Recognizer)**, qui tente de détecter l'écriture manuscrite, les images et même les vidéos.
- Il peut être utilisé dans le domaine des **notations de crédit**. Il essaie de faire correspondre les caractéristiques d'un individu avec le groupe de personnes existantes afin de lui attribuer la cote de crédit. Il se verra attribuer **la même note que celle accordée aux personnes correspondant à ses caractéristiques**.
- Il est utilisé pour **prédire si la banque doit accorder un prêt à un particulier**. Il tentera d'évaluer si l'individu donné correspond aux critères des personnes qui avaient précédemment fait défaut ou ne fera pas défaut à son prêt.

KNN : Avantages & Inconvénients

- *Avantages :*
  - L'algorithme est **simple et facile à mettre en œuvre**.
    Il n'est pas nécessaire de créer un modèle, de régler plusieurs paramètres ou de formuler des hypothèses supplémentaires.
  - L'algorithme est polyvalent. Il peut être utilisé pour la **classification** ou la **régression**.
- *Inconvénients :*
  - L'algorithme devient beaucoup plus lent à mesure que le nombre d'observation et de variables indépendantes augmente.

KNN Scikit Learn Libraries

- L'algorithme KNN est implémenté dans scikit-learn dans les classes *sklearn.neighbors.KNeighborsClassifier* .
- Parameters :
  - **n_neighbors :** *int, default=5* Number of neighbors to use by default for kneighbors queries.
  - **Weights :**{*'uniform', 'distance'*}, *callable or None, default='uniform'* Weight function used in prediction.
  - **Algorithm :** {*'auto', 'ball_tree', 'kd_tree', 'brute'*}, *default='auto'*
  - leaf_size : *int, default=30* Power parameter for the Minkowski metric.
  - **Metric :** *str or callable, default= 'minkowski'* Metric to use for distance computation
  - **metric_params :** *dict, default=None* Additional keyword arguments for the metric function.
  - **n_jobs :** *int, default = None* The number of parallel jobs to run for neighbors search.

KNN Classifier : Exemple d'application

- Un classifieur KNN pour classer les patientes souffrant d'un cancer du sein.
- A voir :

https://www.kaggle.com/code/prashant111/knn-classifier-tutorial

2 – Le classificateur bayésien naïf (Naïve Bayes Classifier)

- Cet algorithme reprend le théorème de Bayes et les probabilités conditionnelles.
- Il repose sur les jeux de données étiquetés, et les associe à d'autres données non étiquetées pour les classifier.
- Le classificateur bayésien naïf est principalement utilisé dans le traitement du langage naturel. Autrement dit, c'est ce qui permet aux machines de comprendre plus facilement le langage humain.
- Un exemple d'utilisation du Naive Bayes est celui du filtre anti-spam.

- Le *Naive Bayes Classifier* se base sur le **théorème de Bayes**. Ce dernier est un classique de la théorie des probabilités.
- Ce théorème est fondé sur les **probabilités conditionnelles**.

Probabilités conditionnelles : Quelle est la probabilité qu'un événement se produise ***sachant qu'un autre événement s'est déjà produit.***

- Le terme P(A|B) se lit : la probabilité que l'événement A se réalise sachant que l'événement B s'est déjà réalisé.
- L'intérêt d'utiliser cette méthode est de trouver la probabilité d'une classe, ou d'une étiquette, en fonction de certains paramètres.

- On appelle ceci la probabilité postérieure, qui peut être calculée avec la formule suivante : $$P(X|Y) = \frac{P(X) * P(Y|X)}{P(Y)}$$

où :

- Y est l'ensemble des paramètres et X est l'ensemble des observations
- P(X|Y) est la probabilité postérieure d'une classe et est donné par :

$$P(X|Y) = P(X \cap Y)/P(Y)$$

- P(X) est la probabilité antérieure d'une classe
- P(Y|X) est la probabilité d'un indicateur, connaissant la classe. Elle est donnée par :

$$P(Y|X) = P(X \cap Y)/P(X)$$

- P(Y) est la probabilité antérieure d'une variable dépendante.

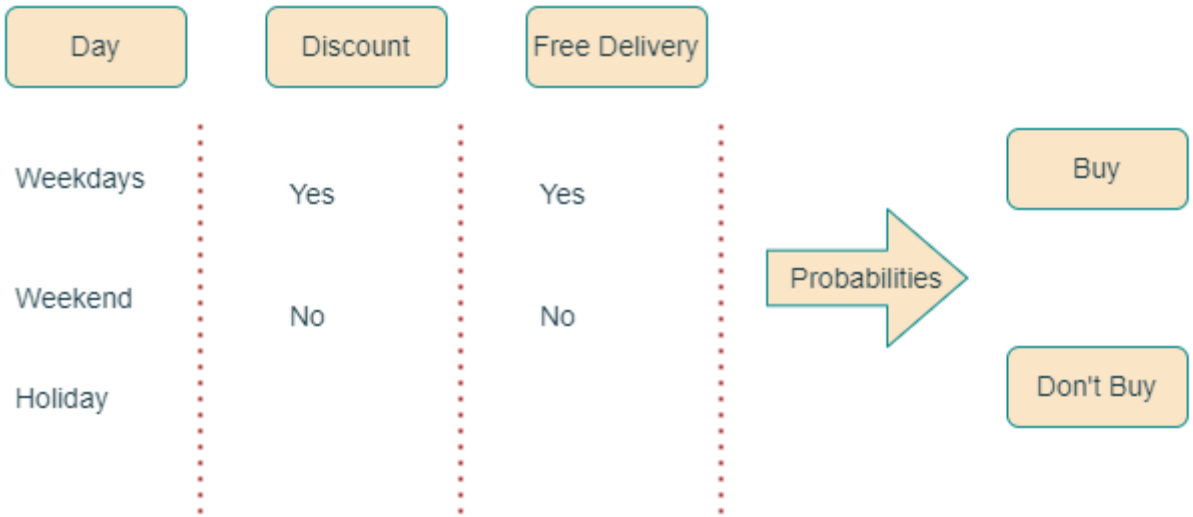
- Le modèle Naïve Bayésien est généralement utilisé pour la classification de documents, le filtrage des spam et les prédictions.
- Il existe différentes versions du modèle Naïve Bayésien, comme :
  - la Gaussian Naïve Bayes,
  - Bernoulli Naïve Bayes,
  - et Multinomial Naïve Bayes.

Exemple Naive Bayes Classifier

- Prenons un exemple de shopping pour comprendre le fonctionnement du Bayes Naive Classifier.
- Dans cet ensemble de données, il existe un petit ensemble de données échantillon de 30 lignes pour cet exemple.

Day	Discount	Free Delivery	Purchase
Weekday	Yes	Yes	Yes
Weekday	Yes	Yes	Yes
Weekday	No	No	No
Holiday	Yes	Yes	Yes
Weekend	Yes	Yes	Yes
Holiday	No	No	No
Weekend	Yes	No	Yes
Weekday	Yes	Yes	Yes
Weekend	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Holiday	No	Yes	Yes
Holiday	No	No	No
Weekend	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Weekday	Yes	Yes	Yes
Holiday	No	Yes	Yes
Weekday	Yes	No	Yes
Weekend	No	No	Yes
Weekend	No	Yes	Yes
Weekday	Yes	Yes	Yes
Weekend	Yes	Yes	No
Holiday	No	Yes	Yes
Weekday	Yes	Yes	Yes
Holiday	No	No	No
Weekday	No	Yes	No
Weekday	Yes	Yes	Yes
Weekday	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Weekend	Yes	Yes	Yes

- Le problème est de prédire si une personne achètera un produit sur une combinaison spécifique de jour, de remise et de livraison gratuite en utilisant le théorème naïf de Bayes.



- **Étape 1)** Nous créerons des tableaux de fréquence pour chaque attribut en utilisant les types d'entrée mentionnés dans l'ensemble de données, tels que les jours, la remise et la livraison gratuite.

Frequency Table		Buy	
		Yes	No
Discount	Yes	19	1
	No	5	5

Frequency Table		Buy	
		Yes	No
Day	Weekday	9	2
	Weekend	7	1
	Holiday	8	3

Frequency Table		Buy	
		Yes	No
Free Delivery	Yes	21	2
	No	3	4

GURU99.com

- Soit l'événement « Acheter » noté « A » et les variables indépendantes, à savoir « Remise », « Livraison gratuite » et « Jour », notées « B ». Nous utiliserons ces événements et variables pour appliquer le théorème de Bayes.
- **Étape 2)** Calculons maintenant les tables de vraisemblance une par une.

Frequency Table		Buy (A)		
		Yes	No	
Day (B)	Weekday	9	2	11
	Weekend	7	1	8
	Holiday	8	3	11
		24	6	

Row Sum

Column Sum

Frequency Table		Buy (A)		
		Yes	No	
Day (B)	Weekday	9/24	2/6	11/30
	Weekend	7/24	1/6	8/30
	Holiday	8/24	3/6	11/30
		24/30	6/30	

- Sur la base de ce tableau de vraisemblance, nous calculerons les probabilités conditionnelles comme ci-dessous.

```
P(A) = P(No Buy) = 6/30 = 0.2
P(B) = P(Weekday) = 11/30 = 0.37
P(B/A) = P(Weekday / No Buy) = 2/6 = 0.33
```

- Et trouvez P(A/B) en utilisant le théorème de Bayes,

```
P(A/B)
= P(No Buy / Weekday)
= P(Weekday / No Buy) * P(No Buy) / P(Weekday)
= (2/6 * 6/30) / (11/30)
= 0.1818
```

- De même, si A est Acheter, alors

```
= P(Buy / Weekday)
= P(Weekday / Buy) * P(Buy) / P(Weekday)
= (9/24 * 24/30) / (11/30)
= 0.8181
```

- **Remarque:** Comme le P(Acheter | Weekday) est supérieur à P(No Buy | Weekday), nous pouvons conclure qu'un client achètera très probablement le produit un jour de semaine.

- **Étape 3)** De même, nous pouvons calculer la probabilité d'occurrence d'un événement sur la base des trois variables. Nous allons maintenant calculer les tableaux de vraisemblance pour les trois variables en utilisant les tableaux de fréquence ci-dessus.

Frequency Table		Buy		
		Yes	No	
Discount	Yes	19/24	1/6	20/30
	No	5/24	5/6	10/30
		24/30	6/30	

Frequency Table		Buy		
		Yes	No	
Free Delivery	Yes	21/24	2/6	23/30
	No	3/24	4/6	7/30
		24/30	6/30	

- Désormais, à l'aide de ces trois tableaux de probabilité, nous allons calculer si un client est susceptible d'effectuer un achat en fonction d'une combinaison spécifique de « Jour », « Remise » et « Livraison gratuite ».
- Ici, prenons une combinaison de ces facteurs :
  - Jour = Jour férié
  - Remise = Oui
  - Livraison gratuite = Oui
- **Quand, A = Acheter**
- Calculez la probabilité conditionnelle d'achat sur la combinaison suivante de jour, de remise et de livraison gratuite.
- Où B est :
  - Jour = Jour férié
  - Remise = Oui
  - Livraison gratuite = Oui
- Et A = Acheter
- Par conséquent,

- Par conséquent,

```
= P(A/B)
= P(Buy / Discount=Yes, Day=Holiday, Free Delivery=Yes)
= ( P(Discount=(Yes/Buy)) * P(Free Delivery=(Yes/Buy)) * P(Day=(Holiday/Buy)) * P(Buy) )
/ ( P(Discount=Yes) * P(Free Delivery=Yes) * P(Day=Holiday) )
= (19/24 * 21/24 * 8/24 * 24/30) / (20/30 * 23/30 * 11/30)
= 0.986
```

- **Quand, A = Pas d'achat**
- De même, calculez la probabilité conditionnelle d'achat sur la combinaison suivante de jour, de remise et de livraison gratuite.

Où B est :

- Jour = Jour férié
- Remise = Oui
- Livraison gratuite = Oui

- Et A = Pas d'achat
- Par conséquent,

- Par conséquent,

```
= P(A/B)
= P(No Buy / Discount=Yes, Day=Holiday, Free Delivery=Yes)
= ( P(Discount=(Yes/No Buy)) * P(Free Delivery=(Yes/No Buy)) * P(Day=(Holiday/No Buy)) *
P(No Buy) )
/ ( P(Discount=Yes) * P(Free Delivery=Yes) * P(Day=Holiday) )
= (1/6 * 2/6 * 3/6 * 6/30) / (20/30 * 23/30 * 11/30)
= 0.027
```

- **Étape 4)** Par conséquent,
- Probabilité d'achat = 0.986
- Probabilité de non achat = 0.027
- Enfin, nous avons des probabilités conditionnelles d'acheter ce jour-là.
- Généralisons maintenant ces probabilités pour obtenir la vraisemblance des événements.
  - Somme des probabilités = 0.986 + 0.027 = 1.013
  - Probabilité d'achat = 0.986 / 1.013 = 97.33 %
  - Probabilité de aucun achat = 0.027 / 1.013 = 2.67 %
- Notez que, puisque 97.33 % est supérieur à 2.67 %.
- Nous pouvons conclure que le client moyen achètera un jour férié avec une remise et une livraison gratuite.

Avantages et limitations

- *Avantages*
  - le *Naive Bayes Classifier* **est très rapide pour la classification** : en effet les calculs de probabilités ne sont pas très coûteux.
  - La classification est possible même avec **un petit jeu de données**
- *Inconvénients*
  - l'algorithme *Naive Bayes Classifier* suppose **l'indépendance des variables** : C'est une **hypothèse forte** et qui est violée dans la majorité des cas réels.

NB Scikit Learn Librairies

- L'algorithme Native Bayes est implémenté dans scikit-learn dans les classes *sklearn. naive_bayes_*
- Pour créer un modèle NB Gaussian :
  - *from sklearn.naive_bayes import GaussianNB*
- Pour créer un modèle NB Bernoulli :
  - *from sklearn.naive_bayes import BernoulliNB*
- Pour créer un modèle NB Multinomial :
  - *from sklearn.naive_bayes import MultinomialNB*
- Pour créer un modèle NB Categorical :
  - *from sklearn.naive_bayes import CategoricalNB*
- Pour créer un modèle NB Compliment :
  - *from sklearn.naive_bayes import ComplimentNB*

Native Bayes Classifer : Exemple d'application

- Fraud Detection with Naive Bayes Classifier
- A voir :

https://www.kaggle.com/code/lovedeepsaini/fraud-detection-with-naive-bayes-classifier

Mesure d'évaluation pour les modèles de classification

- Une fois que un modèle a été déterminé et implanté, il est important d'établir la qualité de ce modèle.
- Pour cela, diverses mesures d'évaluation peuvent être utilisées et choisies soigneusement, puisque le choix de la mesure peut influencer la manière dont la performance est évaluée et interprétée.
- L'une des manières les plus répandues pour mesurer la performance d'un modèle de classification est la matrice de confusion.
- Cette dernière correspond à un résumé tabulaire du nombre de prédictions correctes et non correctes, faites par le modèle.
- Dans cette matrice, chaque ligne correspond à une classe réelle et chaque colonne correspond à une classe estimée.

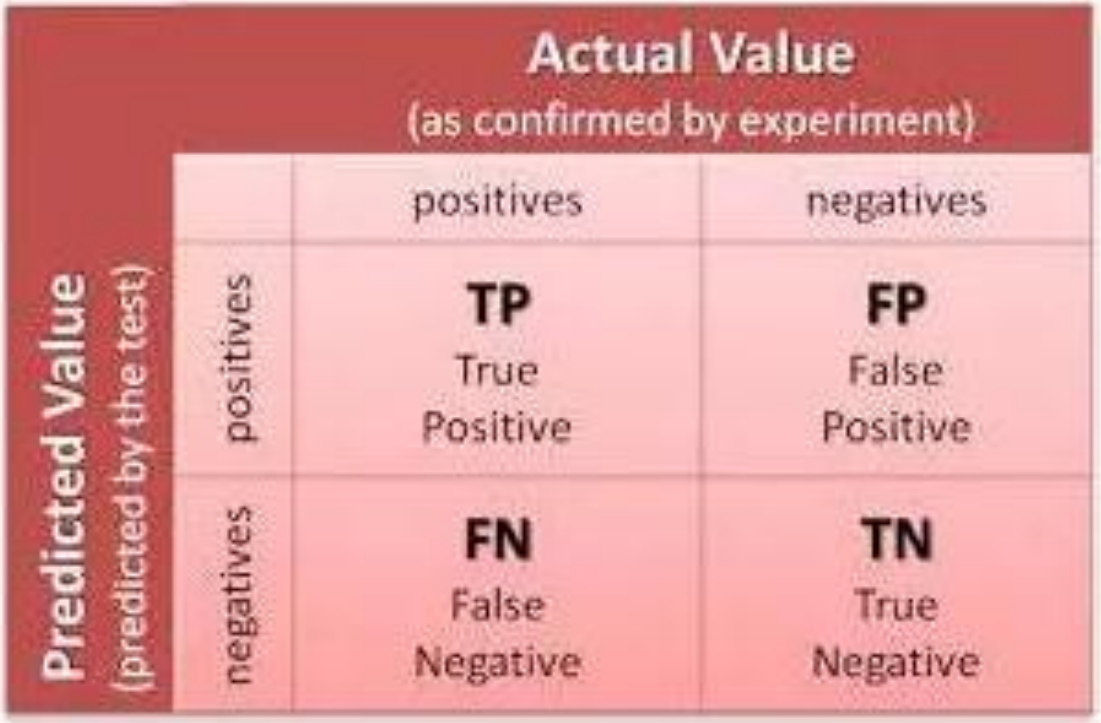
Matrice de Confusion

Elle inclut les valeurs suivantes :

- Vrais positifs (ou True Positive, TP) soit lorsque la classe réelle et la classe estimée sont toutes les deux positives
- Vrais négatifs (ou True Negative , TN) soit lorsque la classe réelle et la classe estimée sont toutes les deux négatives
- Faux positifs (ou False Positive, FP) soit lorsque la classe réelle est négative mais que la classe estimée est positive. On appelle ceci une erreur de Type 1.
- Faux négatifs (ou False Negative, FN) soit lorsque la classe réelle est positive mais que la classe estimée est négative. On appelle ceci une erreur de Type 2.

Matrice de Confusion

- Dans le cas d'une classification binaire, la matrice de confusion sera une matrice de 2 par 2, avec quatre valeurs, comme dans la photo suivante :



Confusion matrix for a binary classification

Dans ce cas, la variable à estimer peut prendre deux valeurs : Positive ou Négative. Les colonnes représentent la classe réelle de la valeur explicative, tant dis que les rangées correspondent à la classe estimée.

Mesures de classification

- Une fois que la matrice de confusion a été établit, elle peut être utilisée pour des mesures plus approfondies afin d'obtenir une meilleure évaluation de la qualité du modèle.
- Parmi les mesures de classification, on trouve :
  - l' Accuracy,
  - la Précision,
  - le Rappel,
  - la Spécificité
  - et le Score F1.

Références

- https://fr.linedata.com/apprentissage-supervise-et-classification
- https://www.guru99.com/fr/naive-bayes-classifiers.html
- https://www.datacamp.com/tutorial/naive-bayes-scikit-learn
- https://scikit-learn.org/stable/modules/naive_bayes.html#out-of-core-naive-bayes-model-fitting
- https://machinelearninggeek.com/naive-bayes-classification-using-scikit-learn/