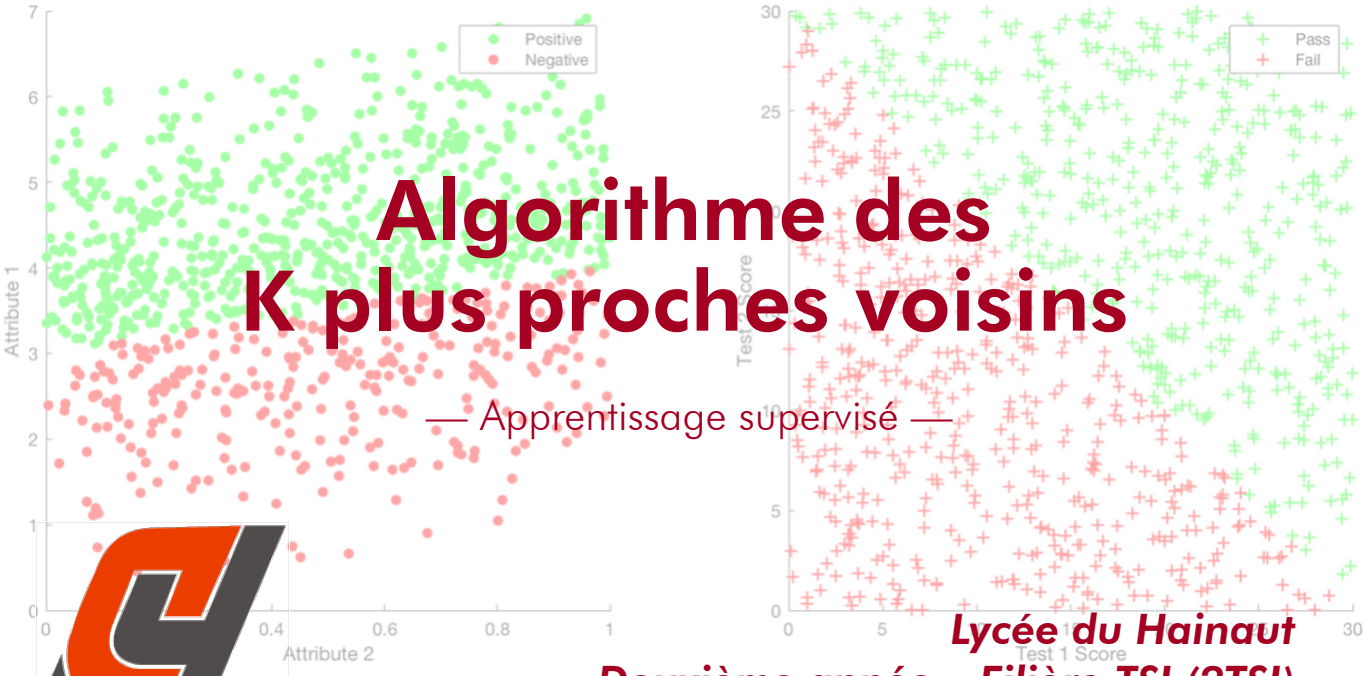
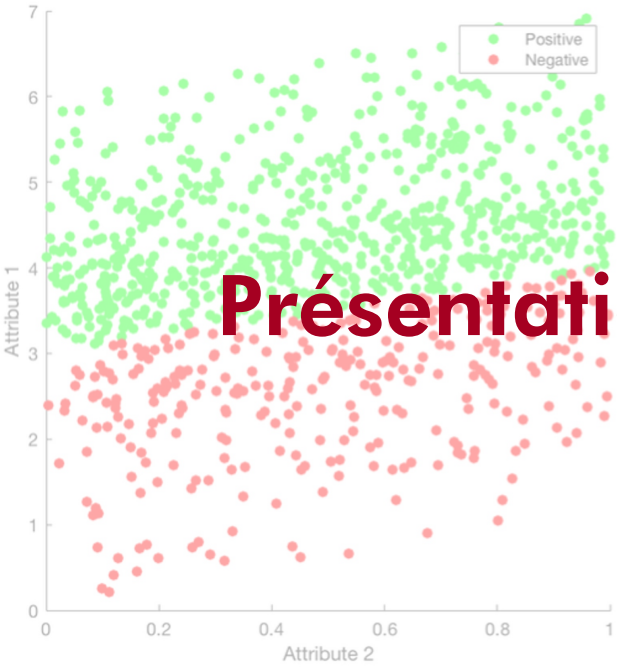


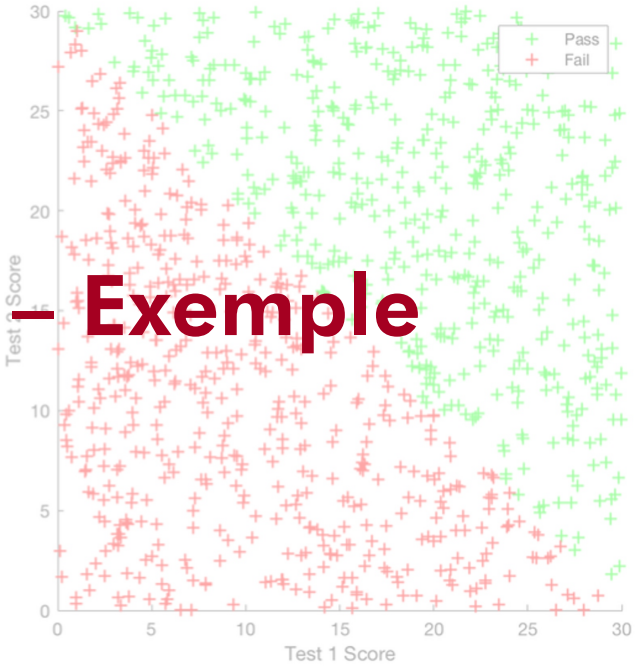
Algorithme des K plus proches voisins

— Apprentissage supervisé —

Lycée du Hainaut
Deuxième année – Filière TSI (2TSI)
2022-2023

Plan





Présentation – Exemple

Généralités

L'algorithme des K plus proches voisins, *K nearest neighbors* (K-nn), est un algorithme de classification en apprentissage supervisé [WIKI1]

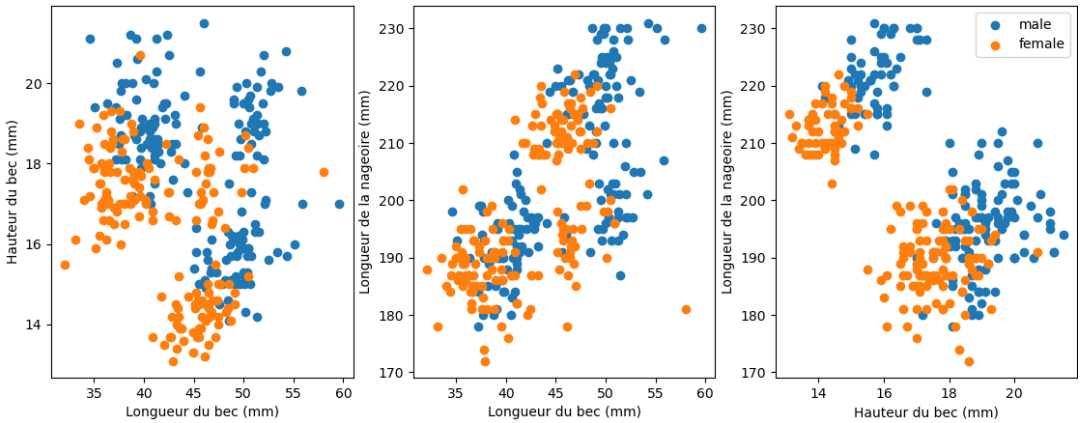
- **À partir d'un ensemble de N observations étiquetées, à chaque nouvelle observation (nouvel échantillon) une étiquette (parmi celles existantes) lui est associée en considérant que c'est la réponse majoritaire (la plus rencontrée) parmi les K plus proches voisins (au sens de la distance) de la nouvelle observation.**
- **Ce classement peut être facilement être adapté à un problème de régression.**

Apprentissage supervisé, rappel

- **Dans ce type d'apprentissage (appelé analyse discriminante), les catégories sont déjà définies et étiquetées (labellisées) à partir d'exemples d'une phase d'apprentissage.**
- **Quand une donnée non labellisée est présentée, son analyse consiste à prédire comment la regrouper :**
  1. Par une **classification** (ou classement), c'est-à-dire associer la donnée à une catégorie (cela s'appelle aussi un « partitionnement ») et donc lui attribuer une étiquette dans un ensemble discret ;
  2. Par une **régression**, c'est à dire lui attribuer une étiquette qui est une valeur numérique dans un intervalle continu ; comme par exemple, lui attribuer la moyenne de différents paramètres.
- **Nous pouvons alors noter que le classement peut être vu comme un cas particulier de régression où les valeurs à prédire sont discrètes (nbre fini de valeurs dans l'intervalle).**

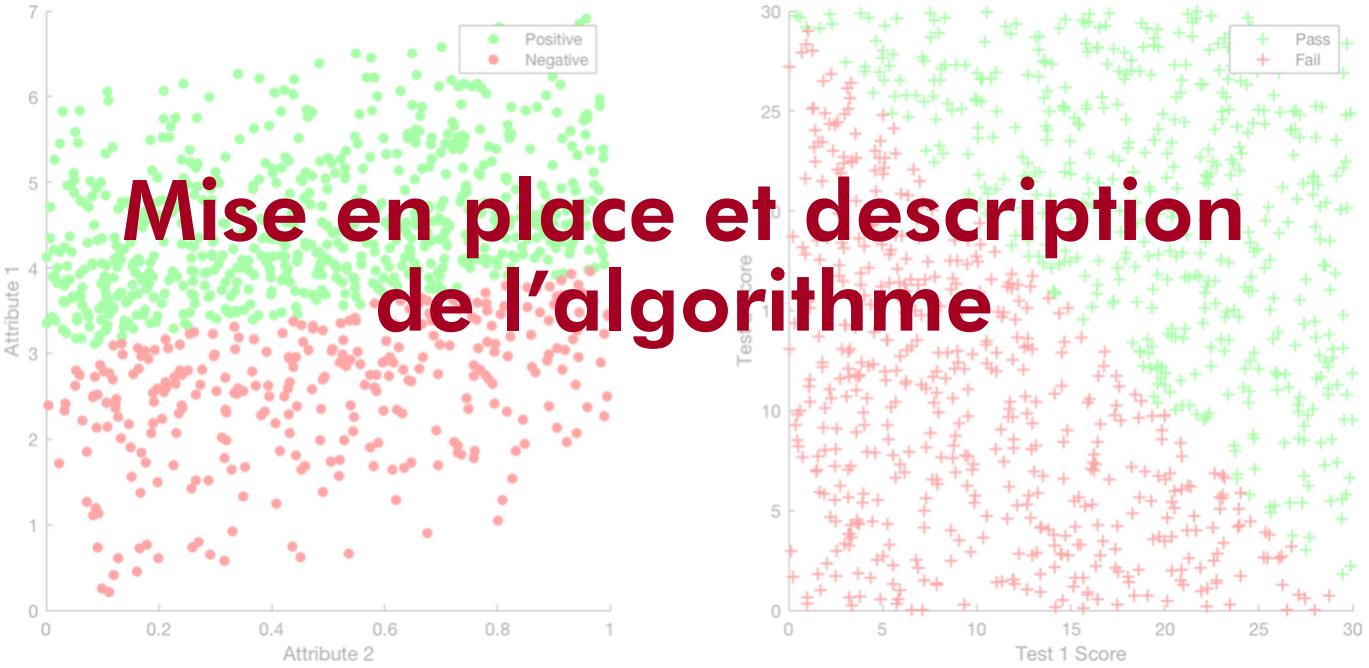
Contexte

- Dans une base de données répertoriant les différents types de pingouins [AZENCOTT2021]. Pour chaque pingouin, elle comprend l'espèce, l'île d'origine, la longueur du bec, la hauteur du bec, la longueur de la nageoire, la masse, le sexe et l'année de naissance.
- La figure suivante représente la répartition des pingouins par sexe en suivant les caractéristiques dimensionnelles du bec et de la nageoire.



Problématique de la classification binaire

- Comment déterminer le sexe d'un pingouin connaissant la longueur et la hauteur du bec, ainsi que la longueur de sa nageoire. Cette classification à deux états est dite binaire.



Mise en place et description de l'algorithme

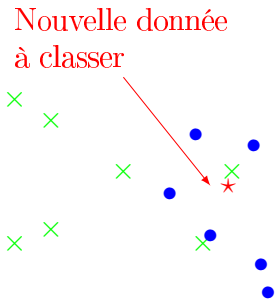
Pseudocode

```
1  Début algorithme "K plus proches voisins"
2    # Données
3    Entrée - X : Matrice des n-uplets d'observation, taille (N × (p+1));
4    # La dernière colonne (+1) est celle de l'étiquette
5    Entrée - K : nombre de voisins à considérer (hyper-paramètre K)
6    Entrée - Xs : observations dont on cherche l'étiquette ; taille (1×p)
7    Sortie (résultat) - E : étiquette de l'observation Xs
8    pour i ← 1..N faire
9       Calculer la distance dsi entre Xs et Xi
10   E ← Étiquette majoritaire parmi les K plus petites distances dsi
11 fin algorithme "K plus proches voisins"
```

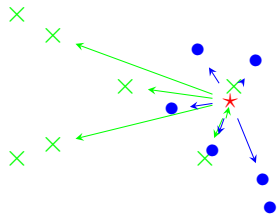
Comme pour l'algorithme des K-moyennes, l'algorithme des K plus proches voisins nécessite de calculer une « distance ». Nous nous limitons à la distance euclidienne définie dans le cours idoine. Et ici aussi, il est nécessaire de normaliser les données qui sont à traiter par l'algorithme.

Évolution de l'algorithme sur un exemple

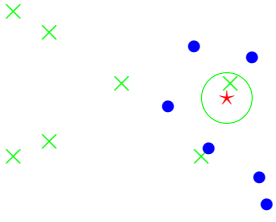
Illustration pour des observations à attribuer à un groupe parmi deux



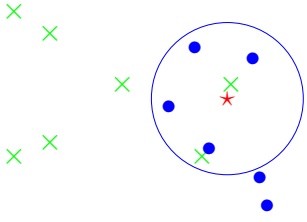
Observations initiales



Calcul des distances d_{SI}



Classement pour $K = 1$



Classement pour $K = 6$

Organisation des données

- **Dans le cas d'un apprentissage supervisé, il est possible de contrôler la qualité du choix des hyper-paramètres (valeur de K pour cet algorithme). Pour cela, il faut séparer les données d'entrée en deux parties :**
  - des **données « d'apprentissage »** servant à utiliser l'algorithme avec de futures données ;
  - des **données de « test »** permettant de valider l'algorithme car leur étiquette est connue a priori.
- **De manière générale, le découpage « apprentissage-test » s'effectue selon des proportions dont l'ordre de grandeur est 80%–20% et 65%–35% test.**
- **Le découpage doit être effectué de manière aléatoire sur l'ensemble des données d'observation.**
- **Avec la bibliothèque `Scikit-Learn`, la fonction `train_test_split()` du module `sklearn.model_selection` est utilisée.**
  - X est la matrice des observations de taille `N` × `(p+1)` et y la matrice des étiquettes ;
  - Le code suivant permet de réaliser le découpage.

```
1 from sklearn import model_selection # fonction extraite, sans préfixe [Wiki2]
2 (X_entrainement,X_test,y_entrainement,y_test) = model_selection.train_test_split(X,
3 y, test_size=0.3, random_state=25)
```

  - L'option `test_size=0.3` signifie que 70% des données sont conservées pour l'apprentissage et les 30% restants pour les tests. L'option `random_state` permet de générer un découpage aléatoire.

Nous revenons sur l'exemple des pingouins

- **Le code suivant permet d'appliquer l'algorithme des *K* plus proches voisins et de le tester sur l'échantillon de test (`X_entrainement`) d'étiquettes (`y_entrainement`).**

```
1 from sklearn.neighbors import KNeighborsClassifier # fonction extraite, sans préfixe [Wiki2]
2 knnclass = KNeighborsClassifier(n_neighbors=7) # pour 7 voisins
3 knnclass.fit(X_entrainement, y_entrainement) # normalisation des données
4 y_test_pred = knnclass.predict(X_test) # attribution pour une nouvelle donnée
```

Analyse

- **La ligne 4 du code permet de prévoir le sexe des pingouins de l'échantillon de test, et de mettre la prédiction dans le tableau `y_test_pred`.**
- **Puisque le sexe des pingouins de l'échantillon test est connu, il est possible considérer s'il y a erreur de prédiction ou non dans la « matrice de confusion » [WIKI3].**

Validation de l'algorithme (2/3)

Matrice de confusion

- **En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité d'un système de classification.**
  - À chaque ligne correspond à une classe réelle et la classe estimée à chaque colonne.
  - À la cellule de la ligne L à la colonne C (L-C) contient le nombre d'éléments de la classe réelle L qui ont été estimés comme appartenant à la classe C.

		Classe estimée (par le classificateur)	
		F (0)	**M (1)**
Classe réelle (sexe avéré du pingouin)	**F**	Nbre de vraies **F** (`True`)	Nbre de fausses **F** (`False`)
	M	Nbre de faux **M** (`False`)	Nbre de vrais **M** (`True`)

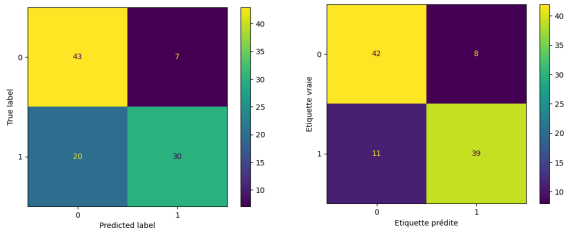
  - Remarque : **Attention** ! Suivant l'auteur les axes de la matrice peuvent être échangés.
- **La matrice s'interprète de cette manière :**
  - la case L0-C0 indique le nombre de femelles de l'échantillon test prédit correctement ;
  - la case L1-C1 correspond au nombre de femelle de l'échantillon test prédit comme mâle ;
  - la case L1-C1 correspond au nombre de mâles de l'échantillon test prédit correctement ;
  - la case L1-C0 correspond au nombre de mâles de l'échantillon test prédit comme femelle.
- **Intérêt de la matrice de confusion : elle montre si un système de classification parvient à classifier correctement (matrice diagonale, 0 hors diagonale)**

Validation de l'algorithme (3/3)

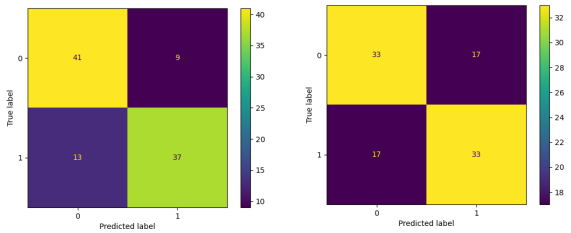
Illustrations issues de tracés Python pour l'exemple des pingouins

- **La bibliothèque `Scikit-Learn` propose le module `metrics` qui permet d'utiliser plusieurs indicateurs de performances d'algorithme d'apprentissage supervisé.**
- **Le code suivant permet de tracer ces matrices de confusion illustrées au-dessous**

```
1 from sklearn import metrics # fonction extraite, sans préfixe [Wiki2]
2 metrics.plot_confusion_matrix(knnclass, X_test, y_test)
```

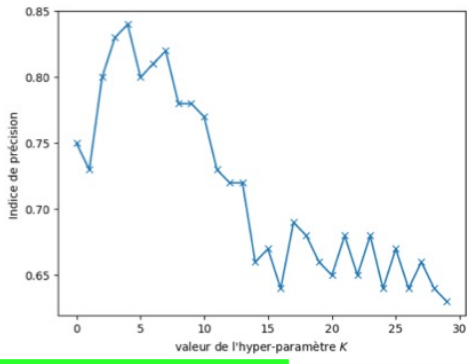


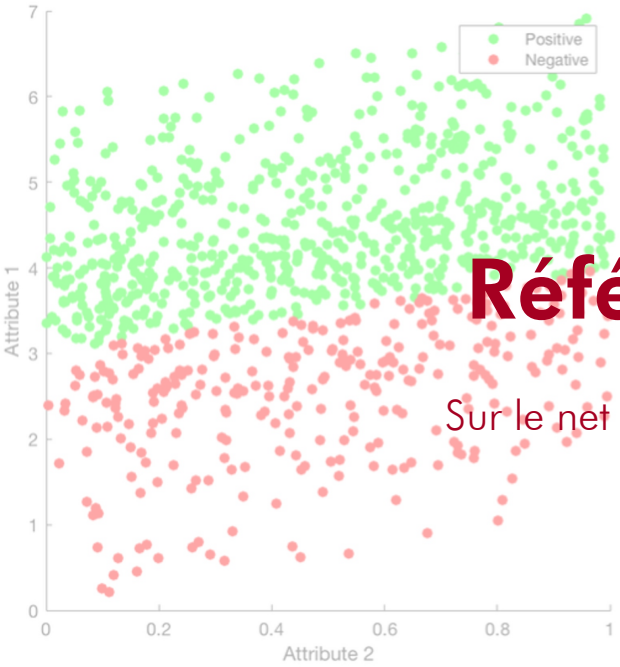
(a) $K = 2$ (b) $K = 7$

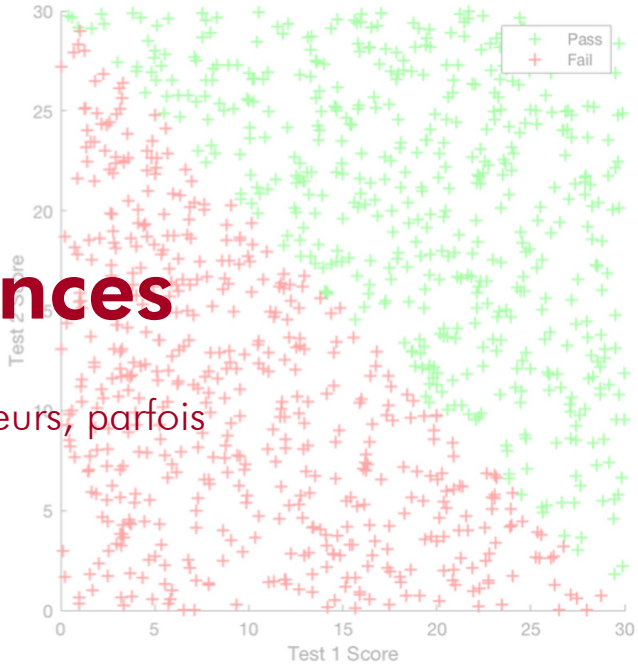


(c) $K = 10$ (d) $K = 15$

- **Il est aussi possible de calculer la précision des prédictions de l'algorithme (somme des termes diagonaux de la matrice de confusion divisée par la somme des prédictions) et ainsi tracer l'évolution de la précision en fonction du paramètre K.**







Références

Sur le net et ailleurs, parfois

Documents

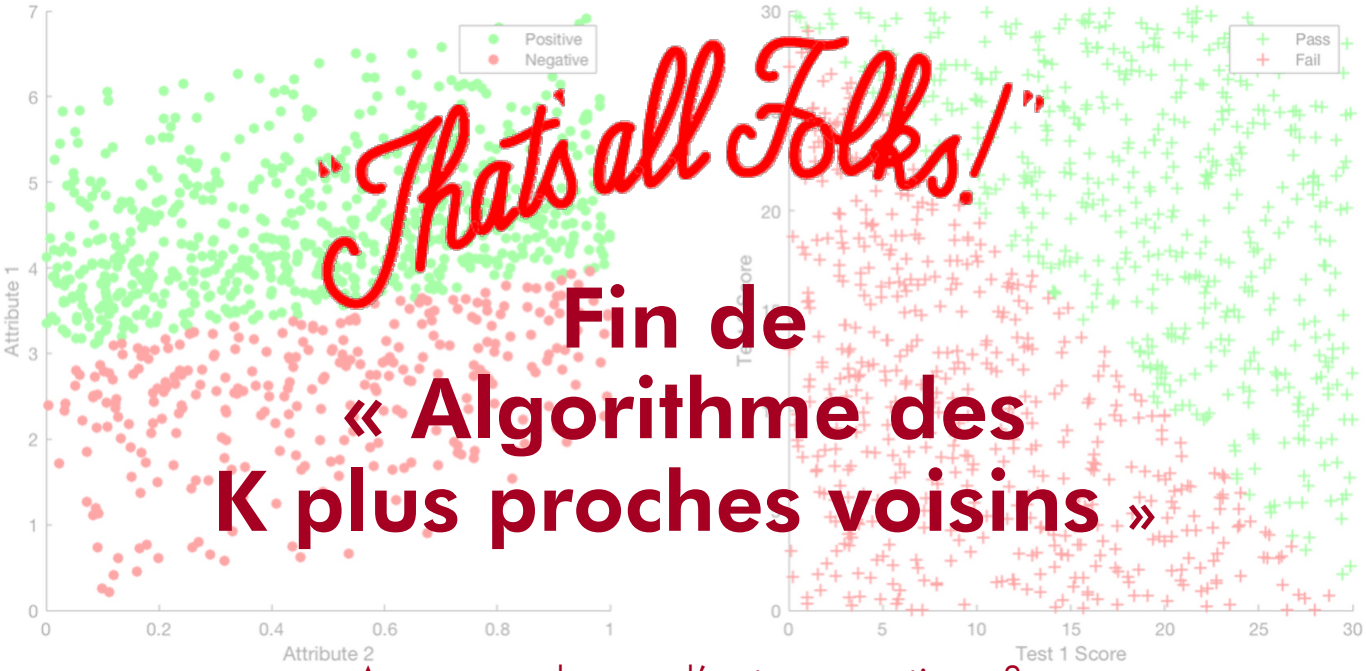
- [AZENCOTT2021] Intelligence Artificielle, Chloé-Agathe Azencott, stage LIESSE2021, Mines Paristech.

Wikipédia

- [WIKI1] k plus proches voisins (méthode des), https://fr.wikipedia.org/wiki/Méthode_des_k_plus_proches_voisins.
- [WIKI2] Scikit-Learn, https://scikit-learn.org.
- [WIKI3] Matrice de confusion, https://fr.wikipedia.org/wiki/Matrice_de_confusion

Documentation de certaines fonctions employées

- Fonction train_test_split, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- Fonction KNeighborsClassifier, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- Fonction confusion_matrix, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html#sklearn.metrics.confusion_matrix



"That's all Folks!"

Fin de « Algorithme des K plus proches voisins »

Avez-vous des ou d'autres questions ?