Algorithme des k plus proches voisins

Activité – Première NSI

① Classer sans programmer

Exercice 1 *Deviner la catégorie*

Un site de vente en ligne souhaite classer automatiquement ses produits en deux catégories : **produits légers** (enveloppes, clés USB, accessoires) et **produits lourds** (livres, appareils). Chaque produit est décrit par deux caractéristiques : son **poids** (en g) et son **prix** (en euros).

Voici les données d'entraînement :

Produit	Poids (g)	Prix (euros)	Catégorie
A	50	5	Léger
B	120	15	Léger
C	800	25	Lourd
D	30	3	Léger
E	950	40	Lourd
F	500	20	Lourd
G	80	8	Léger
H	700	35	Lourd

Un nouveau produit **X** a un poids de **150 g** et un prix de **12 euros**.

1. Sans calcul, en observant les données, dans quelle catégorie classeriez-vous le produit X ? Justifier.

   .....

   .....

2. Quel critère avez-vous utilisé intuitivement ?

   .....

Remarque 1. L'idée de classer un nouvel élément en regardant les éléments qui lui **ressemblent le plus** est le principe fondamental de l'algorithme des k plus proches voisins (k-NN).

② Mesurer la ressemblance : la distance

Définition 1 **(Distance euclidienne)** La **distance euclidienne** entre deux points $P_1(x_1 ; y_1)$ et $P_2(x_2 ; y_2)$ du plan est :

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Exercice 2 *Calculer les distances*

En reprenant les données de l'exercice 1, calculer la distance euclidienne entre le produit X(150 ; 12) et chacun des produits de la base :

Produit	Calcul	Distance
A(50 ; 5)	$\sqrt{(150-50)^2+(12-5)^2} = \sqrt{10000+49}$	$\approx$
B(120 ; 15)		$\approx$
C(800 ; 25)		$\approx$
D(30 ; 3)		$\approx$
E(950 ; 40)		$\approx$
F(500 ; 20)		$\approx$
G(80 ; 8)		$\approx$
H(700 ; 35)		$\approx$

③ L'ALGORITHME DES k PLUS PROCHES VOISINS

Propriété 1 **(Principe de l'algorithme k-NN)** Pour classer un nouvel élément :

1. calculer la distance entre cet élément et chaque élément de la base d'entraînement ;
2. sélectionner les k éléments les plus proches ;
3. la catégorie majoritaire parmi ces k voisins est attribuée au nouvel élément.

Exercice 3 *Appliquer l'algorithme*

En utilisant les distances calculées à l'exercice 2 :

1. Classer les produits du plus proche au plus éloigné de X :

   .....

   .....

2. **Avec** $k = 3$ **:** quels sont les trois plus proches voisins ? Quelle est la catégorie majoritaire ?

Voisin	Distance	Catégorie

   Catégorie attribuée à X : .....

3. **Avec** $k = 5$ **:** quels sont les cinq plus proches voisins ? La conclusion change-t-elle ?

Voisin	Distance	Catégorie

Catégorie attribuée à X : .....

④ L'importance du choix de k

Exercice 4 *Réflexion sur k*

1. Pourquoi est-il déconseillé de choisir un k pair lorsqu'on a deux catégories ?

   .....

   .....

2. Que se passe-t-il si $k = 1$? Quel est le risque ?

   .....

   .....

3. Que se passe-t-il si k est égal au nombre total de données d'entraînement ?

   .....

   .....

4. En pratique, on teste plusieurs valeurs de k et on choisit celle qui minimise les erreurs sur un jeu de test. Pourquoi ne peut-on pas évaluer la qualité de l'algorithme sur les données d'entraînement ?

   .....

   .....

⑤ Normaliser les données

Exercice 5 *Un problème d'échelle*

Reprenons les données de l'exercice 2. Les poids vont de 30 à 950 g et les prix de 3 à 40 euros.

1. Lorsque l'on calcule $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, quelle caractéristique (poids ou prix) domine le calcul ? Pourquoi ?

   .....

   .....

2. Cela signifie que le prix a très peu d'influence sur la classification. Est-ce souhaitable ?

   .....

3. Pour résoudre ce problème, on peut **normaliser** les données. La formule de normalisation min-max est :

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Normaliser le poids et le prix du produit A(50 ; 5) sachant que : $\text{poids}_{\text{min}} = 30$, $\text{poids}_{\text{max}} = 950$, $\text{prix}_{\text{min}} = 3$, $\text{prix}_{\text{max}} = 40$.

Poids normalisé : .....

Prix normalisé : .....

4. Après normalisation, toutes les valeurs sont comprises entre ..... et .....

⑥ Données réelles : communes françaises (INSEE)

Remarque 2. Jusqu'ici nous avons travaillé sur des données fictives. En pratique, on charge un **fichier de données** (souvent au format CSV) produit par une institution publique : l'INSEE, l'État via `data.gouv.fr`, etc.

Exercice 6 *Lire un fichier CSV*

L'INSEE publie pour chaque commune française sa population, sa superficie et une **catégorie de densité** (de 1 : densément peuplée à 4 : très peu peuplée). Le fichier `communes-insee.csv` est un extrait de 252 communes au format CSV standard français :

— séparateur point-virgule `;`

— encodage UTF-8

— première ligne = en-tête

— colonnes : `code_insee ; nom_commune ; departement ; population ; superficie_km2 ; densite_hab_km2 ; grille_densite`

Voici les premières lignes du fichier :

code	nom	pop	superficie	densite	cat.
01004	Ambérieu-en-Bugey	15934	24.55	649	2
13055	Marseille	886040	237.93	3724	1
69123	Lyon	519127	47.97	10821	1
75056	Paris	2103778	105.36	19967	1

1. En Python, le module `csv` lit chaque ligne d'un fichier comme du **texte**. Pourquoi est-il **indispensable** de convertir certaines colonnes avec `int(...)` ou `float(...)` avant de calculer une distance euclidienne ?

   .....

   .....

2. Quelles colonnes du fichier doit-on convertir en nombres pour appliquer k-NN ?

   .....

3. Le fichier d'origine de l'INSEE contient environ 35 000 lignes (toutes les communes de France). Notre extrait n'en contient que 252, équilibré à environ 60 par catégorie. Quel **biais** cet équilibrage artificiel introduit-il pour l'entraînement du k-NN ?

   .....

   .....

Exercice 7 *Choix des caractéristiques et normalisation*

On souhaite prédire la **catégorie de densité** d'une nouvelle commune à partir de deux caractéristiques : sa **population** (de 11 à 2 103 778 habitants dans l'extrait) et sa **densité** (de 2 à 19 967 hab/km²).

1. Sans normalisation, quelle caractéristique va dominer le calcul de distance euclidienne ? Justifier en comparant les ordres de grandeur.

   .....

   .....

2. Pour la commune de Paris (population ≈ 2,1 M, densité ≈ 20 000 hab/km²), donner la valeur après normalisation min-max :

   Population normalisée : ..... Densité normalisée : .....

3. Pour un village (population = 420, densité = 35 hab/km²), donner les valeurs normalisées :

   Population normalisée : ..... Densité normalisée : .....

4. En quoi cela rend-il l'algorithme plus juste pour classer une commune intermédiaire (ex. Bordeaux, ≈ 260 000 habitants, densité ≈ 5 400 hab/km²) ?

   .....

   .....

⑦ Synthèse

Exercice 8 *Bilan*

Compléter les phrases :

1. L'algorithme k-NN est un algorithme d'apprentissage .....
2. Pour classer un nouvel élément, il cherche les k éléments les plus .... dans la base d'entraînement et attribue la catégorie .....
3. La « ressemblance » entre deux éléments se mesure par une .....
4. Le choix de k influence le résultat. Un k trop petit risque .....et un k trop grand risque .....
5. La .....des données est nécessaire lorsque les caractéristiques n'ont pas la même échelle.