

Entrainement sujet B CCINP : Système de recommandation

19 mai 2025

L'objectif de cet exercice est d'écrire en Ocaml un système de recommandation basé sur l'algorithme des k-voisins. On dispose donc d'un tableau à deux dimensions qui contient pour chaque personne une information traduisant si elle a ou non aimé chacun des films (+ si elle l'a aimé, - si elle ne l'a pas aimé et 0 si elle ne l'a pas vu).

	007	Batman 1	Shrek 2	Toy Story 3	Star Wars 4	Twilight 5
Alice	+	-	0	+	0	-
Bob	-	0	+	-	+	+
Charles	+	+	+	+	-	-
Daisy	+	+	0	0	+	-
Everett	+	-	+	+	-	0

Pour mesurer la similarité entre deux personnes, on calculera le nombre de films sur lesquels leurs avis concordent moins celui où leurs avis divergent (dès lors que l'une des deux personnes n'a pas vu un film il ne compte pas dans la mesure de similarité).

Par exemple, le score entre Alice et Charles vaut $3 - 1 = 2$.

1. Calculer le score entre Alice et Bob.
2. Rappeler le principe de l'algorithme des k-voisins et expliquer comment on peut l'utiliser dans ce contexte.

On va implémenter le système de recommandation, pour cela un fichier `tp_knn_eleves` à compléter vous est fourni.

3. Ecrire une fonction `compute_score : int -> int -> int` qui prend en entrée deux entiers `i` et `j` et telle que `compute_score i j` renvoie le score entre la personne correspondant à la ligne `i` du tableau global `notes` et à celle correspondant à la ligne `j`.
4. Ecrire une fonction `compute_all_scores : void -> int array array` qui renvoie un tableau à deux dimensions tel que dans la case (i, j) on trouve la valeur `compute_score i j`.
5. Spécifier précisément la fonction `init` fournie dans le fichier.
6. Compléter la fonction `nearest_neighbors` pour qu'elle renvoie la liste des k plus proches voisins de `i` sous forme d'une liste de couples (j, k) où k est la similarité entre la personne d'indice `i` et celle d'indice `k`.
7. Ecrire une fonction `majorite : int list -> int` qui prend en entrée une liste contenant uniquement les valeurs 1, 0 et -1 et qui renvoie la valeur majoritaire parmi 1 et -1 sans tenir compte des 0.
8. Ecrire une fonction `pred : int -> int -> int` telle que `pred i j` renvoie 1 si le système de recommandation suggère à la personne d'indice `i` de regarder le film d'indice `j` et -1 sinon.