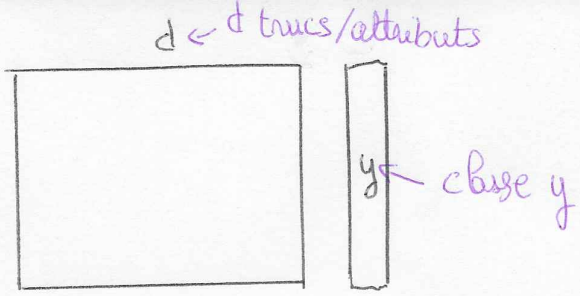
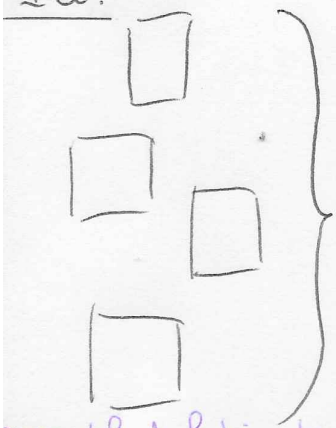
Ce dont on a l'habitude :



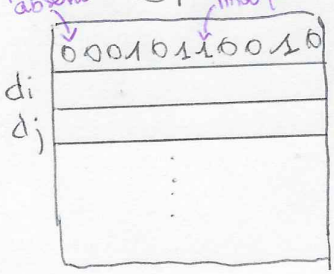
Ici :



dictionnaire

⇒ Tous les mots présents dans le corpus de texte ?
↳ ⚠ bcp de "the", "and", "be", "have", etc.
⇒ à virer ! ⇒ NLTK (Natural Language Toolkit)
⇒ Pour créer le dico, utiliser CountVectorizer dans scikit-learn.

Structures de données :



Mesurer la similarité ?

⚠ distance euclidienne : bcp de zéros, tout va être similaire
→ plutôt produit scalaire :

$$\frac{d_i \cdot d_j}{\|d_i\| \, \|d_j\|}$$ → pour normaliser (en fait c'est la formule des cosinus).

Pour le pb de stockage de tous ces zéros : tables de hachage → sparse matrix
⇒ Représentation sous forme de sac de mots (bag of words model)

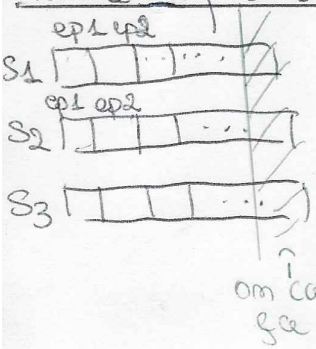
Outils : → pour les algos de machine learning : scikit-learn

→ TF-IDF (Term frequency - inverse document frequency) : pr évaluer l'importance d'un terme contenu dans un document relativement à un corpus.

$$TF\text{-}IDF(t_i, d_j) = tf(t_i, d_j) \times \log\left(\frac{1+N}{1+df(t_i)}\right)$$

$tf(t_i, d_j)$: nb de fois que t_i apparaît ds d_j
N : nb de docs ds le corpus
$df(t_i)$: nb de docs qui contiennent t_i

Modèle prédictif :



→ prédire si un épisode appartient à une saison/série
- on peut faire la même chose avec les saisons
- permet de faire de la recommandation sur le contenu