

02-01

Recherche des plus proches voisins

**NOUS ÉCLAIRONS.
VOUS BRILLENZ.**

FORMATION CONTINUE
ET SERVICES AUX ENTREPRISES



Sommaire

1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. k-NN avec scikit-learn
6. Questions ouvertes
7. Lectures et références

Sommaire

1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. k-NN avec scikit-learn
6. Questions ouvertes
7. Lectures et références

Recherche de documents

- Commençons par un exemple: vous êtes présentement en train de lire un article sur le soccer ...

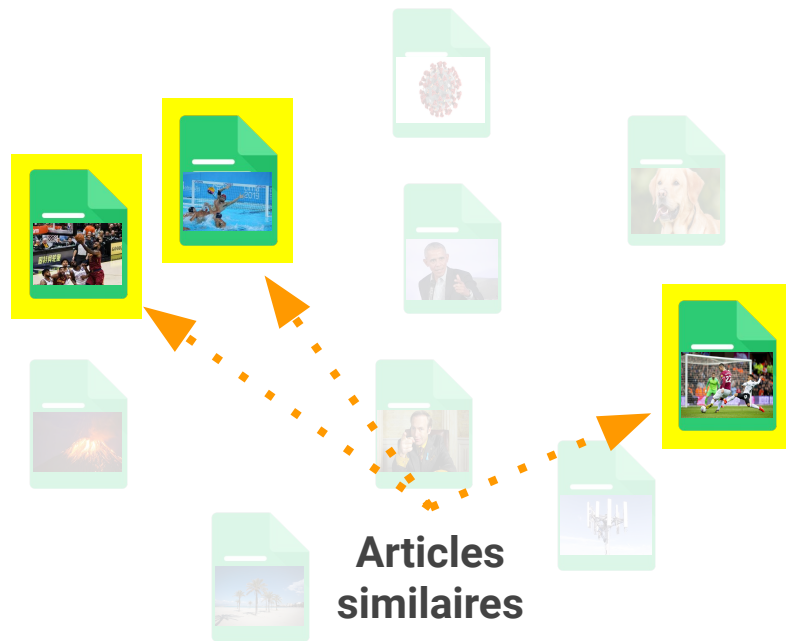


Vous souhaitez trouver des articles
similaires



Recherche de documents

- Commençons par un exemple: vous êtes présentement en train de lire un article sur le soccer ...

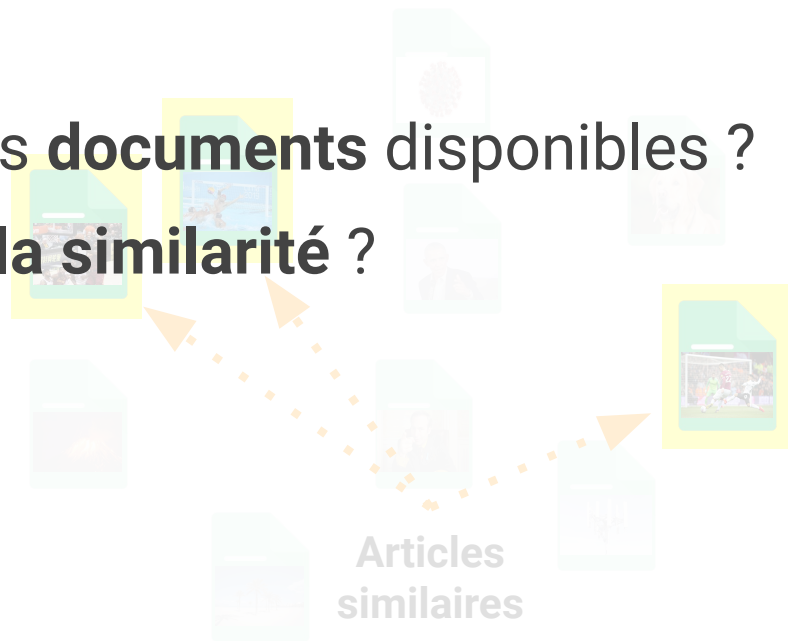
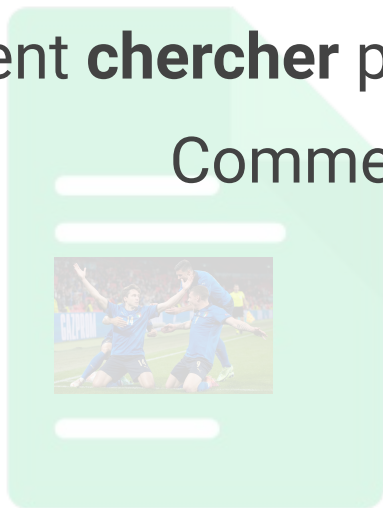


Recherche de documents

- Commençons par un exemple: vous êtes présentement en train de lire un article sur le soccer ...

Comment **chercher** parmi tous les **documents** disponibles ?

Comment **mesurer la similarité** ?

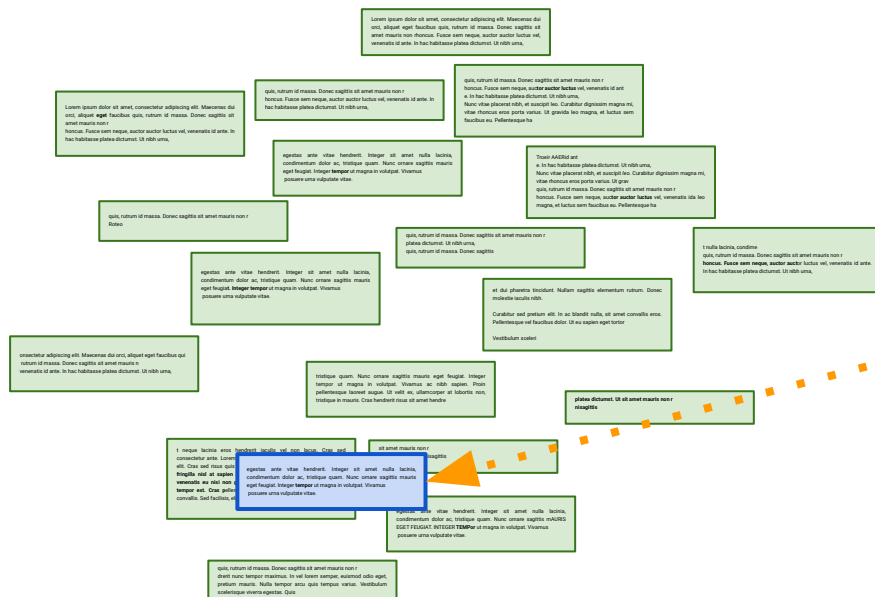


Sommaire

1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. k-NN avec scikit-learn
6. Questions ouvertes
7. Lectures et références

Recherche 1-NN

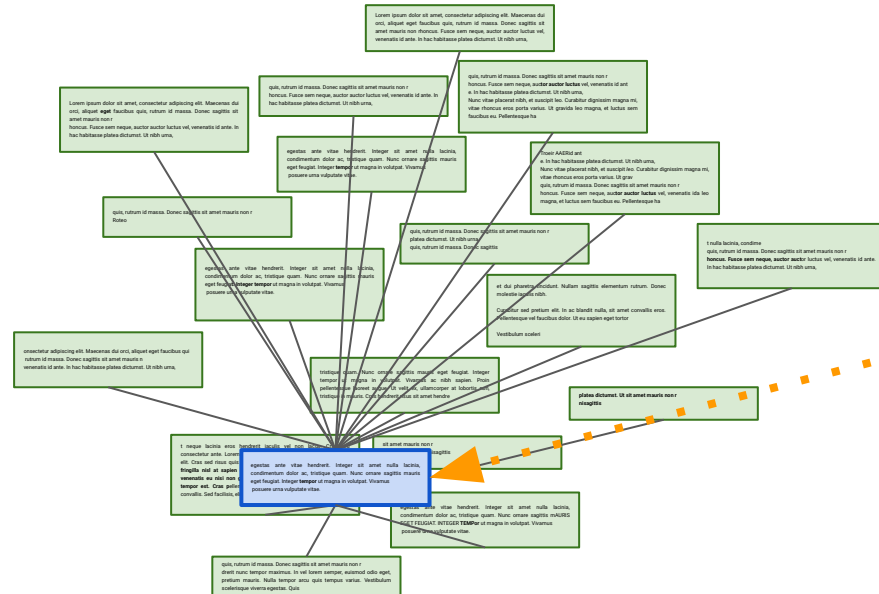
- Ci-dessous, est représenté l'espace de tous les documents organisés par **similarité du texte**



Article requête
(query article)

Recherche 1-NN

- On calcul de la **distance** vers tous les documents

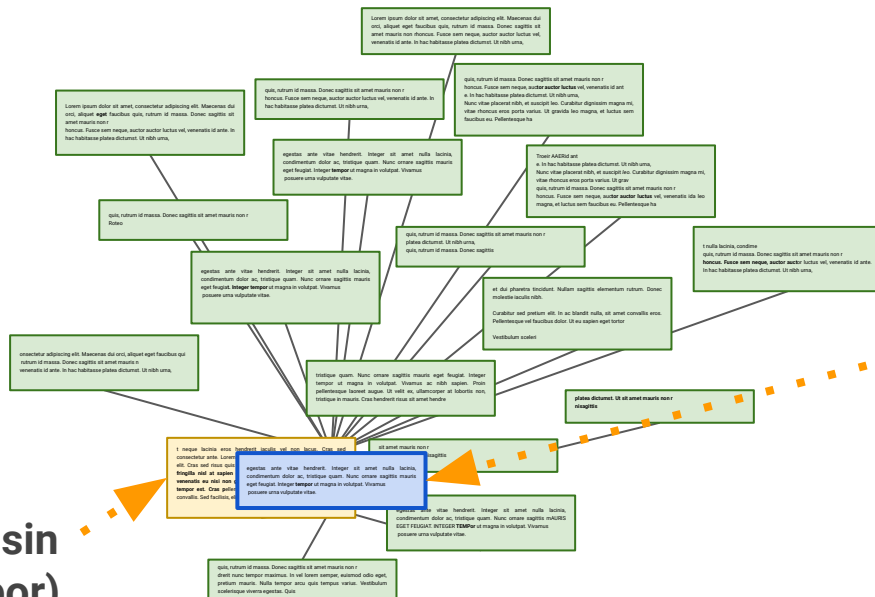


Article requête
(query article)

Recherche 1-NN

■ Identification du plus proche voisin

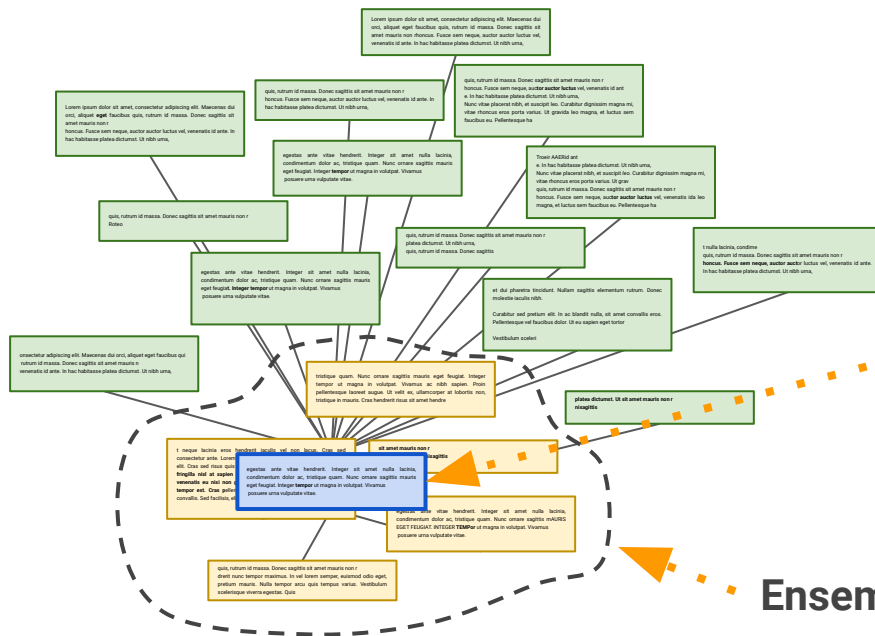
Plus proche voisin
(nearest neighbor)



Article requête
(query article)

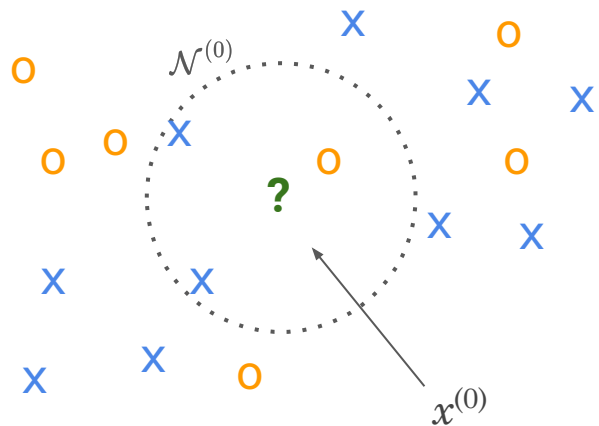
Recherche k-NN

■ Identification d'un ensemble de plus proches voisins

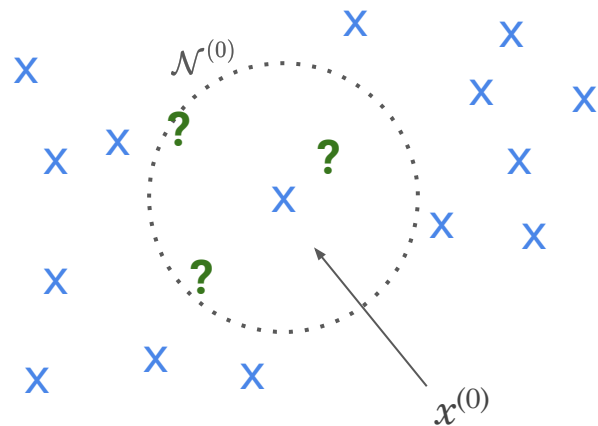


k-NN: comparaison supervisé / non supervisé

Apprentissage supervisé



Apprentissage non supervisé





Sommaire

1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. k-NN avec scikit-learn
6. Questions ouvertes
7. Lectures et références

Algorithme 1-NN: notations

■ Entrée:

- Document "query": x_q 
- Corpus de documents: x_1, x_2, \dots, x_M 

■ Sortie:

- Document le plus similaire: x^{NN} 

- Formellement, nous cherchons le document x_i ayant la distance minimale avec x_q

$$x^{NN} = \min_{x_i} \text{distance}(x_q, x_i)$$

Algorithme 1-NN: pseudo-code

Initialiser $\text{Dist2NN} = \infty$, $x^{\text{NN}} = \emptyset$

Itérer sur tous les documents x_1, x_2, \dots, x_M :

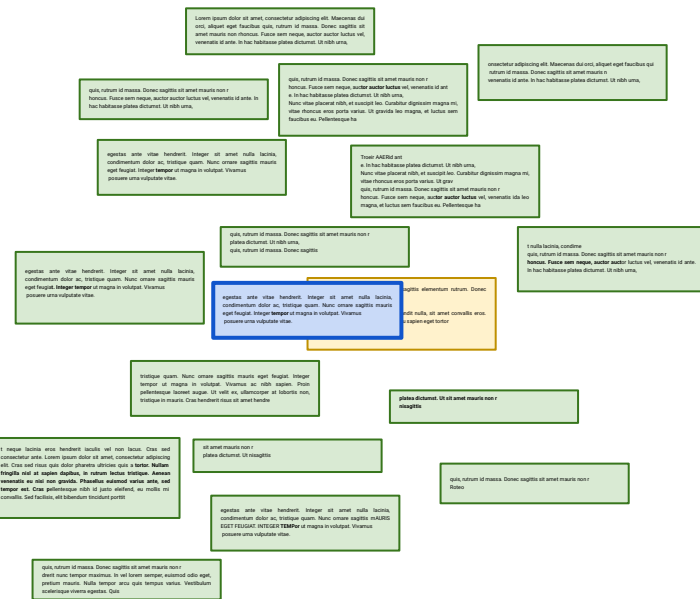
Calculer la distance δ entre x_q et x_i

Si $\delta < \text{Dist2NN}$

$x^{\text{NN}} = x_i$

$\text{Dist2NN} = \delta$

Retourner le document le plus similaire x^{NN}
(document du corpus le plus proche de x_q)





Sommaire


1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. k-NN avec scikit-learn
6. Questions ouvertes
7. Lectures et références

Algorithme k-NN

■ Entrée:

- Document "query": x_q 
- Corpus de documents: x_1, x_2, \dots, x_M 

■ Sortie:

- Liste des k documents les plus similaires: $X^{NN} = \{x^{NN1}, x^{NN2}, \dots, x^{NNk}\}$ 

- Formellement, nous cherchons tous les documents x_i tels que

$$\forall x_i \notin X^{NN}, \text{distance}(x_q, x_i) \geq \max_{x_i^{NNj}, j=1, \dots, k} \text{distance}(x_q, x^{NNj})$$

Algorithme k-NN

Initialiser $\text{Dist2kNN} = \text{sort}(\delta_1, \delta_2, \dots, \delta_k)$

Itérer sur tous les documents de $k+1$ à M : x_{k+1}, \dots, x_M :

Calculer la distance δ entre x_q et x_i

Si $\delta < \text{Dist2kNN}[k]$

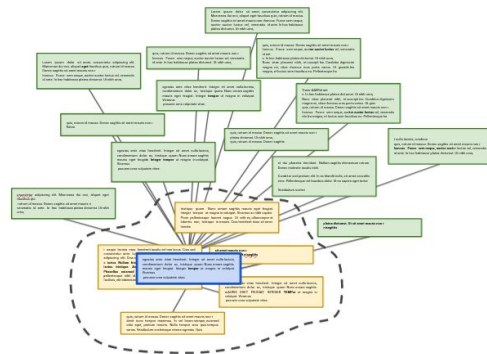
Trouver j tel que $\delta > \text{Dist2kNN}[j-1]$ et $\delta < \text{Dist2kNN}[j]$

Retirer le document le plus loin et déplacer les autres

$\text{Dist2kNN}[j+1:k] = \text{Dist2kNN}[j:k-1]$

$\text{Dist2kNN}[j] = \delta$

Retourne les k documents les plus similaires



**Trier les k premiers
documents selon leur
distance au document
"query"**

Algorithme k-NN

Initialiser $\text{Dist2kNN} = \text{sort}(\delta_1, \delta_2, \dots, \delta_k)$

Itérer sur tous les documents de $k+1$ à M : x_{k+1}, \dots, x_M :

Calculer la distance δ entre x_q et x_i

Si $\delta < \text{Dist2kNN}[k]$

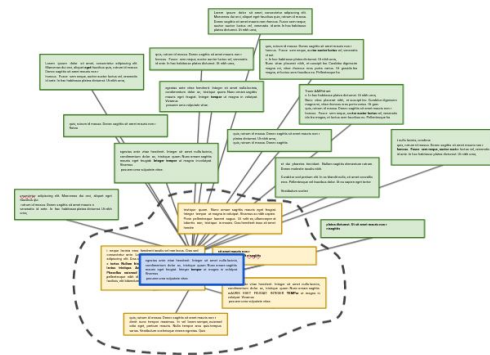
Trouver j tel que $\delta > \text{Dist2kNN}[j-1]$ et $\delta < \text{Dist2kNN}[j]$

Retirer le document le plus loin et déplacer les autres

$\text{Dist2kNN}[j+1:k] = \text{Dist2kNN}[j:k-1]$

$\text{Dist2kNN}[j] = \delta$

Retourne les k documents les plus similaires



Insertion du document

Sommaire

1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. **k-NN avec scikit-learn**
6. Questions ouvertes
7. Lectures et références

NearestNeighbors (scikit-learn 0.24.2)

```
class sklearn.neighbors.NearestNeighbors(*, n_neighbors=5, radius=1.0,  
algorithm='auto', leaf_size=30, metric='minkowski', p=2, metric_params=None,  
n_jobs=None)
```

Nombre de voisins

Rayon de recherche


Algorithme de recherche

Métrique de distance

Sommaire

1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. k-NN avec scikit-learn
6. Questions ouvertes
7. Lectures et références

Questions ouvertes

- Comment représenter les documents ?  $\rightarrow x_q$
- Quelle mesure utiliser pour calculer la distance entre les documents ? $\delta = \mathbf{distance}(x_i, x_q)$

Sommaire

1. Recherche de documents
2. Recherche des plus proches voisins
3. Algorithme 1-NN (exhaustif / brute-force)
4. Algorithme k-NN (exhaustif / brute-force)
5. k-NN avec scikit-learn
6. Questions ouvertes
7. Lectures et références

Lectures et références

[1] Machine Learning: Clustering and Retrieval - Emily Fox & Carlos Guestrin - University of Washington