TP Avancé : Indexation et Recherche de Données avec Elasticsearch

Objectifs du TP :

L'objectif de ce TP est de maitriser ces compétences :

- Importer un fichier CSV dans Elasticsearch et afficher son mapping.

- Concevoir des requêtes avancées avec filtres, agrégations et scoring.

- Explorer et personnaliser les analyseurs de texte en fonction d'un scénario guidé.

- Évaluer la pertinence et la performance du moteur de recherche en utilisant rappel et précision.

Chaque étudiant travaillera sur un fichier CSV de son choix, contenant des données exploitables pour une recherche d'information.

⚠️ Les sujets doivent être différents entre les étudiants pour garantir un travail personnalisé.

Partie 1 : Importation des Données et Exploration du Mapping

1.1 - Choix et Importation d'un Fichier CSV

Chaque étudiant doit :

- Sélectionner un fichier CSV pertinent.

- Vérifier son contenu et s'assurer qu'il contient au moins 5 colonnes exploitables.

- Importer les données dans Elasticsearch.

1.2 - Affichage et Analyse du Mapping

Après l'importation, chaque étudiant doit afficher le mapping de l'index créé :

GET mon_index/_mapping

Travail à faire :

- Expliquer le mapping généré automatiquement par Elasticsearch.

- Identifier les champs mal typés et proposer un mapping plus adapté.

- Modifier le mapping si nécessaire.

Partie 2 : Requêtes Avancées

2.1 - Requêtes avec Filtres

Travail à faire :

- Rechercher tous les documents contenant un mot spécifique.

- Filtrer les résultats selon une plage de valeurs.

- Appliquer un filtre booléen combinant plusieurs conditions ('should', 'must')

2.2 - Agrégations

- Calculer le nombre total de documents.

- Effectuer une agrégation par catégories.

- Trouver la valeur moyenne, minimum et maximum d'un champ numérique.

2.3 - Recherche Full-Text et Boosting

- Effectuer une recherche `match` sur un champ textuel.

- Comparer une requête `match` et `match_phrase`.

- Booster certains champs :

Dans Elasticsearch, le **boosting** permet de **donner plus de poids** à certains champs lorsqu'on effectue une recherche, influençant ainsi le **score de pertinence** (`_score`) des documents retournés. Plusieurs manières de faire, un exemple est le facteur de pondération :

- En ajoutant un facteur de pondération **(^) :**
  Exemple : GET mon_index/_search { "query": { "multi_match": { "query": "machine learning", "fields": ["titre^3", "description^1"] } } }

  Ici, le champ **`titre`** a un **facteur de boost de 3**, donc les documents où "machine learning" est présent dans le titre auront **plus de score**.
  Le champ **`description`** a un **boost normal (1)**, il est donc moins prioritaire.

D'autres solutions sont possibles (utiliser boost dans une requête match…..), documentez-vous pour choisir une possibilité pour booster votre requête.

Partie 3 : Analyseurs et Personnalisation

3.1 - Tester les Analyseurs Existant :

- Il s'agit de tester plusieurs analyseurs intégrés et comparer leurs résultats.

L'API `_analyze` permet de voir **comment Elasticsearch découpe et transforme** un texte selon un analyseur donné.

```
GET _analyze
{
  "text": "L'intelligence artificielle transforme le monde.",
  "analyzer": "standard"
}
```

Elasticsearch fournit plusieurs analyseurs intégrés: keyword, whitespace, simple, stop, custom_french.. Testez quelques uns (2 au moins).

3.2 - Création d'un Analyseur Personnalisé

- Créer un analyseur personnalisé adapté à votre dataset.

- Tester cet analyseur sur votre dataset.

- Comparer les résultats avec ceux obtenus avant.

- Justifier vos choix.

Partie 4 : Évaluation du Scoring et Performance du SRI

4.1 - Comprendre le Score de Pertinence

L'objectif est d'observer le `_score` des documents retournés.

4.2 - Calcul du Rappel et de la Précision

Formules :

Précision = Nb de documents pertinents retournés / Nb total de documents retournés

Rappel = Nb de documents pertinents retournés / Nb total de documents pertinents existants

Travail à faire :

- Calculer la précision et le rappel avant et après optimisation des requêtes.

- Proposer une amélioration pour optimiser ces valeurs.

Rendus Attendus

Chaque étudiant doit fournir :

- Un rapport écrit expliquant son travail.

- Copier coller chaque requête, afficher en dessous la capture d'écran de l'exécution de chaque requête.

- Expliquer quand cela est demandé le résultat obtenu.