

**École Nationale des Sciences Appliquées - Al Hoceima**

Département Informatique / Data Engineering

# **JobRecommander**

Système Intelligent d’Aide à l’Orientation  
et d’Analyse du Marché de l’Emploi

*Rapport Technique de Projet*

## **Équipe projet :**

KBIRI ALAOUI Abdelhafid  
Boukaich Mohammed Rida  
Sabbar Mohamed  
Lamadi Youssef

## **Encadré par :**

Pr. EL HADDADI Anass

Année Universitaire 2025-2026

# Table des matières

<b>1 Résumé Exécutif</b>	<b>2</b>
<b>2 Problématique et Objectifs</b>	<b>2</b>
<b>3 Architecture Technique</b>	<b>2</b>
<b>4 Méthodologie Algorithmique</b>	<b>3</b>
4.1 Construction du Corpus . . . . .	3
4.2 Vectorisation (TF-IDF) . . . . .	3
4.3 Calcul de Similarité . . . . .	3
<b>5 Expérience Utilisateur (Application Streamlit)</b>	<b>3</b>
5.1 Flux d'utilisation . . . . .	4
<b>6 Analyse Décisionnelle (PowerBI)</b>	<b>4</b>
6.1 Indicateurs Clés et Visuels . . . . .	4
<b>7 Conclusion et Perspectives</b>	<b>4</b>

# 1 Résumé Exécutif

Dans un écosystème de recrutement saturé d'informations, la rencontre entre l'offre et la demande est souvent inefficace. Ce projet, intitulé **JobRecommender**, propose une solution double : un moteur de recommandation personnalisé basé sur le Traitement du Langage Naturel (NLP) et une interface d'analyse décisionnelle (Business Intelligence).

En exploitant une base de données de plus de 550 000 offres, la solution permet d'aligner sémantiquement les compétences des candidats avec les opportunités du marché, tout en offrant une transparence sur les salaires et les tendances via des tableaux de bord interactifs.

## 2 Problématique et Objectifs

Le processus de recherche d'emploi souffre actuellement de deux maux majeurs :

1. **La surcharge cognitive** : Les candidats doivent filtrer manuellement des milliers d'offres non pertinentes sur des plateformes généralistes.
2. **L'opacité du marché** : Il existe un manque de clarté sur les compétences réelles demandées par les recruteurs et sur les grilles salariales pratiquées.

**L'objectif de JobRecommender** est d'automatiser le "matching" entre un profil (CV ou description textuelle) et une offre d'emploi. Contrairement à une recherche par mots-clés simple, notre approche repose sur une analyse de la similarité textuelle globale, garantissant des résultats plus pertinents contextuellement.

## 3 Architecture Technique

Le projet repose sur une architecture en pipeline ("Data Pipeline") composée de quatre phases distinctes, développée principalement en Python :

- **Ingestion & Nettoyage (Pandas/NumPy)** : Fusion de 10 fichiers sources (offres, entreprises, avantages) et stratégies d'imputation pour gérer les valeurs manquantes.
- **Moteur NLP (Scikit-learn)** : Transformation des descriptions textuelles en vecteurs mathématiques exploitables par la machine.
- **Interface Utilisateur (Streamlit)** : Développement d'un front-end web léger permettant l'interaction en temps réel avec le modèle.
- **Business Intelligence (PowerBI)** : Création d'un tableau de bord pour la visualisation macro-économique des données nettoyées.

## 4 Méthodologie Algorithmique

Le cœur du système repose sur une approche de **Filtrage par Contenu** (Content-Based Filtering).

### 4.1 Construction du Corpus

Chaque offre d'emploi est d'abord prétraitée pour former une entité sémantique unique. Nous concaténons les champs suivants : le titre du poste, la description détaillée, les compétences requises, le niveau d'expérience et le type de contrat. Cela crée une "signature textuelle" riche pour chaque offre.

### 4.2 Vectorisation (TF-IDF)

Le système utilise la méthode *Term Frequency-Inverse Document Frequency* (TF-IDF) pour convertir le texte en vecteurs numériques. Cette technique permet de pondérer l'importance des mots : un terme rare dans le corpus global mais fréquent dans une offre spécifique (ex : "Kubernetes") aura un poids plus élevé qu'un terme générique (ex : "équipe").

La configuration du modèle est la suivante :

- **Vocabulaire** : Limité aux 3 000 traits (features) les plus significatifs.
- **N-grams** : Utilisation d'unigrammes et de bigrammes (pour capturer des termes comme "Data Scientist").
- **Stop-words** : Élimination des mots de liaison anglais.

### 4.3 Calcul de Similarité

Pour recommander une offre, le système projette le profil du candidat dans le même espace vectoriel que les offres. Nous utilisons la **Similarité Cosinus** pour mesurer l'angle entre le vecteur "Candidat" ( $A$ ) et le vecteur "Offre" ( $B$ ) :

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

Un score proche de 1 indique une correspondance sémantique parfaite.

## 5 Expérience Utilisateur (Application Streamlit)

L'application web, développée avec Streamlit, offre une interface épurée et fonctionnelle qui remplace les processus de recherche traditionnels complexes.

## 5.1 Flux d'utilisation

1. **Saisie du Profil :** L'utilisateur entre une description libre (ex : "*Je suis un Data Engineer junior maîtrisant Python et AWS, cherchant du télétravail*").
2. **Filtrage Dynamique :** Des options latérales permettent de restreindre la recherche (Télétravail uniquement, Niveau Senior/Junior, Type de contrat).
3. **Résultats :** Le système retourne instantanément une liste classée des 10 meilleures opportunités. Chaque résultat affiche le titre, l'entreprise, et un score de pertinence (matching score) transparent pour l'utilisateur.

## 6 Analyse Décisionnelle (PowerBI)

En complément du moteur de recommandation, un module analytique offre une vue d'ensemble du marché du travail. Ce tableau de bord permet aux candidats de situer leurs attentes.

### 6.1 Indicateurs Clés et Visuels

Le dashboard présente une analyse structurée autour de plusieurs axes :

- **Vue Macro :** Affichage des volumes totaux d'offres et du salaire médian global du marché.
- **Géographie :** Carte interactive et graphiques montrant la répartition des offres par ville et la proportion de postes en télétravail versus sur site.
- **Compétences & Salaires :** Une analyse de corrélation mettant en évidence l'impact des compétences techniques (Nuage de mots) et des années d'expérience sur la rémunération proposée.

## 7 Conclusion et Perspectives

Le projet **JobRecommander** démontre la faisabilité et l'efficacité de l'application de l'Intelligence Artificielle au domaine du recrutement. En remplaçant la recherche par mots-clés stricte par une compréhension sémantique via TF-IDF, nous avons réussi à améliorer la pertinence des résultats pour les candidats.

Pour les évolutions futures, nous envisageons l'intégration de modèles de Deep Learning (type BERT ou Transformers) pour saisir des nuances contextuelles plus fines, ainsi que le développement d'une API permettant l'ingestion d'offres en temps réel pour maintenir la base de données à jour.