



UNIVERSITÉ LUMIÈRE LYON 2

Master 2 SISE

Statistique et Informatique pour la Science des données

Projet RADAR

NLP Text Mining - Analyse régionale d'offres d'emploi

Auteurs :

Mohamed Habib BAH

Thibaud LECOMTE

Aya MECHERI

Rina RAZAFIMAHEFA

Encadrement :

Ricco RAKOTOMALALA

Année universitaire : 2025–2026

TABLE DES MATIÈRES

Introduction	4
0.1 Contexte général	4
0.2 Problématique	4
0.3 Objectifs du projet RADAR	5
0.4 Dimension régionale et enjeux territoriaux	5
0.5 Contributions et périmètre du projet	5
1 Cadre théorique et concepts mobilisés	7
1.1 Text Mining et traitement automatique du langage naturel	7
1.1.1 Spécificités des données textuelles non structurées	7
1.1.2 Principes du Text Mining	7
1.1.3 Notions fondamentales du NLP	8
1.2 Représentation vectorielle des textes	8
1.2.1 Modèles sac-de-mots	8
1.2.2 TF-IDF	8
1.2.3 Limites des approches statistiques classiques	8
1.3 Extraction d'information à partir des offres d'emploi	9
1.3.1 Extraction de compétences	9
1.3.2 Analyse thématique des descriptions	9
1.3.3 Enjeux liés au vocabulaire métier	9
1.4 Analyse géographique des données textuelles	9
1.4.1 Visualisation cartographique	9
2 Collecte et structuration des données	11
2.1 Sources de données	11
2.1.1 France Travail	11
2.1.2 Hellowork	11
2.1.3 Emploi Territorial	12
2.2 Méthodes de collecte	12
2.2.1 Collecte via API	12
2.2.2 Web scraping HTML	12
2.3 Nettoyage et normalisation des données	12
2.3.1 Traitement des champs textuels	12
2.3.2 Gestion des doublons et des valeurs manquantes	13
2.4 Modélisation et stockage	13
2.4.1 Schéma de la base de données	13
2.4.2 Entrepôt de données et choix technologiques	13

3 Méthodes de Text Mining et NLP mises en œuvre	15
3.1 Prétraitement et préparation des documents de compétence	15
3.1.1 Construction des documents synthétiques	15
3.2 Vectorisation par pondération TF-IDF	16
3.2.1 Représentation vectorielle du corpus	16
3.3 Analyse thématique	17
3.3.1 Algorithme des K-Means	17
3.3.2 Réduction de dimensionnalité (PCA)	17
3.4 Caractérisation métier des clusters	18
4 Application web et visualisation	19
4.1 Architecture globale de l'application	19
4.1.1 Framework et Philosophie de développement	19
4.2 Analyses statistiques et géographiques	19
4.2.1 Tableau de bord interactif (Plotly)	19
4.2.2 Cartographie régionale (Folium)	20
4.3 L'Intelligence artificielle au service de l'utilisateur	21
4.3.1 Analyse sémantique (Onglet Intelligence)	21
4.3.2 Aide à la candidature (Onglet Assistant)	21
5 Déploiement et reproductibilité	23
5.1 Conteneurisation avec Docker	23
5.1.1 Motivations du choix de Docker	23
5.1.2 Description de l'image (Dockerfile)	23
5.2 Gestion des dépendances et configuration	23
5.2.1 Dépendances Python (<code>requirements.txt</code>)	23
5.3 Installation et exécution	24
5.3.1 Lancement de l'application	24
5.4 Limites et contraintes techniques	24
6 Analyse métier et discussion	25
6.1 Résultats principaux et dynamiques territoriales	25
6.1.1 Volume global et pôles d'attractivité	25
6.1.2 Focus sur l'insertion : alternance et stages	25
6.2 Étude de cas : La région Auvergne-Rhône-Alpes (AuRA)	26
6.2.1 L'apport de la vue <i>Sunburst</i>	26
6.3 Discussion et limites	27
6.3.1 Limites des données	27
6.3.2 Limites méthodologiques	27
6.3.3 Apports métier du projet	27
6.3.4 Pour les candidats et demandeurs d'emploi	27
6.3.5 Pour les recruteurs et acteurs RH	28
6.3.6 Pour les acteurs publics et institutionnels	28

7 Conclusion et perspectives	29
7.1 Synthèse du projet	29
7.2 Considérations éthiques et limites techniques	29
7.2.1 Éthique et respect des plateformes	29
7.2.2 Utilisation responsable des LLM	29
7.3 Perspectives d'amélioration	29
7.3.1 Enrichissements supplémentaires	29
7.3.2 Gamification et engagement communautaire	30
7.3.3 Coach Carrière IA (LLM augmenté)	30
7.3.4 Architecture logicielle et Clean Code	30
7.4 Mot de la fin	31

INTRODUCTION

0.1 Contexte général

La généralisation des plateformes numériques de recrutement a profondément transformé la diffusion des offres d'emploi. Chaque jour, des milliers d'annonces sont publiées sur des sites spécialisés, des portails institutionnels ou des plateformes privées, constituant un volume massif de données textuelles hétérogènes. Ces offres contiennent des informations riches mais peu structurées concernant les compétences recherchées, les technologies utilisées, les niveaux d'expérience attendus, les types de contrats ou encore les localisations géographiques.

Dans les domaines de la data science et de l'intelligence artificielle, cette dynamique est particulièrement marquée. L'essor rapide de ces disciplines s'accompagne d'une diversification des intitulés de postes, des compétences techniques demandées et des contextes métiers. L'analyse manuelle de ces annonces devient rapidement impraticable à grande échelle, tant en raison du volume que de la variabilité linguistique des descriptions.

Dans ce contexte, les approches de Text Mining et de traitement automatique du langage naturel (*Natural Language Processing*, NLP) offrent des outils adaptés pour exploiter ces corpus textuels. Elles permettent d'extraire automatiquement de l'information pertinente, de structurer des données initialement non structurées et de dégager des tendances globales à partir de textes libres. Ces méthodes constituent aujourd'hui un levier central pour analyser le marché de l'emploi à partir de sources ouvertes issues du web.

0.2 Problématique

Si les offres d'emploi publiées en ligne représentent une source d'information précieuse sur l'état et "évolution du marché du travail, leur exploitation pose plusieurs défis. Les données sont dispersées sur de multiples sources, rédigées dans des formats variés et ne suivent pas de structure standardisée. De plus, les descriptions textuelles mêlent informations essentielles et éléments discursifs, rendant leur analyse directe complexe.

La problématique centrale de ce projet est donc la suivante : comment transformer un corpus massif et hétérogène d'offres d'emploi orientées data et intelligence artificielle en informations exploitables permettant d'analyser les tendances du marché, tant sur le plan thématique que géographique ? Il s'agit notamment de déterminer quelles compétences sont les plus demandées, comment elles se répartissent selon les régions, et quelles dynamiques territoriales émergent à partir de ces données textuelles.

Répondre à cette problématique nécessite la mise en œuvre conjointe de techniques de collecte automatisée, de structuration des données, de traitements NLP et de visualisation interactive, afin de proposer une analyse à la fois rigoureuse, reproductible et accessible.

0.3 Objectifs du projet RADAR

Le projet RADAR s'inscrit dans cette problématique en proposant une chaîne complète de traitement des offres d'emploi, depuis leur collecte jusqu'à leur analyse et leur restitution. Les objectifs principaux du projet sont les suivants :

- automatiser la collecte d'offres d'emploi orientées data et intelligence artificielle à partir de sources web hétérogènes, via des techniques de web scraping et l'utilisation d'API ;
- stocker ces données dans une base structurée de type entrepôt, facilitant leur interrogation et leur exploitation ultérieure ;
- appliquer des méthodes de Text Mining et de NLP afin d'extraire des compétences, thématiques et informations pertinentes à partir des descriptions textuelles ;
- analyser la répartition géographique des offres et des compétences afin de mettre en évidence des tendances régionales ;
- développer une application web interactive permettant l'exploration du corpus, l'ajout dynamique de nouvelles offres et la visualisation des résultats ;
- livrer une solution complète, reproductible et déployable via une image Docker.

Ainsi, RADAR vise à fournir un outil d'analyse du marché de l'emploi data et IA combinant rigueur méthodologique et accessibilité.

0.4 Dimension régionale et enjeux territoriaux

L'intégration d'une dimension géographique constitue un axe central du projet RADAR. Le marché de l'emploi ne se structure pas de manière homogène sur le territoire : certaines régions concentrent des pôles d'innovation, des écosystèmes technologiques ou des bassins d'emploi spécialisés, tandis que d'autres présentent des dynamiques différentes.

L'analyse régionale des offres d'emploi permet d'identifier des disparités territoriales en matière de compétences recherchées, de types de postes proposés ou de niveaux de spécialisation attendus. Ces informations sont pertinentes aussi bien pour les candidats, souhaitant orienter leur mobilité ou leur formation, que pour les décideurs publics ou privés, intéressés par l'attractivité et le développement économique des territoires.

En combinant *Text Mining* et visualisation géographique, le projet RADAR cherche à mettre en lumière ces enjeux territoriaux et à proposer une lecture synthétique et interactive des dynamiques régionales du marché de l'emploi data et IA.

0.5 Contributions et périmètre du projet

Le projet RADAR propose une contribution méthodologique et applicative à l'analyse du marché de l'emploi à partir de données textuelles issues du web. Il met en œuvre une architecture complète intégrant collecte automatisée, stockage structuré, traitements NLP, visualisation interactive et déploiement.

Le périmètre du projet se concentre sur des offres d'emploi orientées data science, machine learning et intelligence artificielle, extraites de sources identifiées. Les analyses portent principalement sur l'extraction de compétences, l'identification de thématiques et la répartition géographique des offres. En revanche, le projet ne vise pas à réaliser des prédictions salariales ni à entraîner des modèles de deep learning complexes, mais plutôt à fournir une analyse exploratoire et descriptive du corpus.

Enfin, RADAR s'inscrit dans une démarche pédagogique, visant à mobiliser de manière cohérente les concepts et outils abordés en programmation du Text Mining et du NLP, tout en répondant à une problématique concrète et actuelle.

CADRE THÉORIQUE ET CONCEPTS MOBILISÉS

1.1 Text Mining et traitement automatique du langage naturel

L'essor massif des données textuelles issues du web, des réseaux sociaux ou des plateformes professionnelles a conduit au développement de méthodes spécifiques destinées à exploiter ces informations non structurées. Le *Text Mining* et le traitement automatique du langage naturel (*Natural Language Processing*, NLP) constituent aujourd'hui des domaines centraux de la science des données lorsqu'il s'agit d'analyser, structurer et interpréter des corpus textuels de grande taille.

Dans le cadre du projet RADAR, ces approches sont mobilisées afin d'extraire des connaissances exploitables à partir des descriptions d'offres d'emploi, qui se présentent sous forme de textes libres, hétérogènes et fortement dépendants du vocabulaire métier.

1.1.1 Spécificités des données textuelles non structurées

Contrairement aux données numériques ou catégorielles, les données textuelles ne possèdent pas de structure explicite directement exploitable par des algorithmes classiques. Un texte est caractérisé par :

- une forte variabilité lexicale, incluant synonymes, abréviations et formulations implicites ;
- une dimension contextuelle, où le sens dépend de l'environnement linguistique ;
- une dimension bruitée, notamment dans les données issues du web (fautes, répétitions, formulations imprécises).

Les offres d'emploi illustrent parfaitement ces difficultés : deux annonces peuvent décrire des compétences similaires en utilisant des terminologies très différentes, tandis qu'un même terme peut recouvrir des réalités métiers distinctes selon le contexte.

1.1.2 Principes du Text Mining

Le *Text Mining* regroupe un ensemble de techniques visant à transformer un corpus textuel brut en données structurées permettant une analyse quantitative. Il s'appuie généralement sur les étapes suivantes :

1. Prétraitement linguistique (nettoyage, tokenisation, normalisation),
2. Représentation numérique des textes,
3. Extraction de motifs, thèmes ou entités pertinentes,
4. Analyse statistique ou algorithmique des résultats.

Dans RADAR, le *Text Mining* est utilisé pour :

- identifier les thématiques dominantes des offres,
- extraire les compétences techniques et transversales,
- comparer les contenus textuels selon des dimensions régionales ou sectorielles.

1.1.3 Notions fondamentales du NLP

Le **NLP** vise à permettre aux machines de traiter le langage humain de manière automatisée. Il repose sur des concepts linguistiques tels que :

- la **tokenisation**, qui consiste à découper un texte en unités élémentaires ;
- la **lemmatisation**, qui ramène les mots à leur forme canonique ;
- l'élimination des **mots outils** (*stop words*), peu informatifs sur le plan sémantique.

Ces opérations sont indispensables pour réduire la complexité du langage naturel et améliorer la qualité des représentations numériques utilisées par les algorithmes d'analyse.

1.2 Représentation vectorielle des textes

Afin d'appliquer des méthodes statistiques ou de machine learning aux textes, il est nécessaire de convertir ces derniers en représentations numériques. Cette étape constitue un élément central du *Text Mining*.

1.2.1 Modèles sac-de-mots

Le modèle du **sac-de-mots** (*Bag of Words*) repose sur une hypothèse simplificatrice : un document est représenté comme un ensemble non ordonné de termes, sans tenir compte de la syntaxe ni de l'ordre des mots. Chaque document est ainsi décrit par un vecteur dont les composantes correspondent aux fréquences des termes dans le corpus.

Bien que simple à mettre en œuvre, cette approche permet déjà de capturer des informations pertinentes sur le contenu des offres d'emploi, notamment la présence ou l'absence de compétences clés.

1.2.2 TF-IDF

La pondération **TF-IDF** (*Term Frequency - Inverse Document Frequency*) améliore le modèle sac-de-mots en attribuant un poids plus important aux termes spécifiques d'un document et un poids plus faible aux termes trop fréquents dans l'ensemble du corpus.

Cette méthode permet de :

- réduire l'impact des mots génériques,
- mettre en évidence les termes discriminants entre différentes offres,
- améliorer la qualité des analyses thématiques et comparatives.

Dans le projet RADAR, TF-IDF est utilisé comme base pour l'analyse thématique et la comparaison des descriptions d'offres entre régions.

1.2.3 Limites des approches statistiques classiques

Malgré leur efficacité et leur simplicité, les approches statistiques comme le sac-de-mots ou TF-IDF présentent plusieurs limites :

- absence de prise en compte du contexte sémantique,

- difficulté à gérer les synonymes et les expressions composées,
- sensibilité au vocabulaire spécifique des métiers.

Ces limites justifient une interprétation prudente des résultats et ouvrent la voie à des améliorations futures basées sur des modèles plus avancés.

1.3 Extraction d'information à partir des offres d'emploi

L'un des enjeux majeurs du projet RADAR réside dans l'extraction d'informations pertinentes à partir des descriptions textuelles des offres d'emploi.

1.3.1 Extraction de compétences

Les compétences constituent une information clé pour analyser le marché de l'emploi. Leur extraction repose sur :

- des listes de compétences prédéfinies,
- des méthodes de correspondance lexicale,
- des techniques statistiques visant à identifier les termes récurrents.

Cette extraction permet de comparer les besoins en compétences selon les régions ou les types de postes.

1.3.2 Analyse thématique des descriptions

L'analyse thématique vise à regrouper les offres selon des thèmes dominants (data science, intelligence artificielle, ingénierie logicielle, etc.). Elle repose sur la similarité entre les représentations vectorielles des textes et permet d'identifier des tendances structurelles dans le corpus.

1.3.3 Enjeux liés au vocabulaire métier

Le vocabulaire utilisé dans les offres d'emploi est fortement dépendant des pratiques professionnelles et des évolutions technologiques. Certains termes peuvent être polysémiques ou émergents, ce qui complexifie leur interprétation automatique. L'analyse textuelle doit donc être accompagnée d'une lecture métier afin de garantir la pertinence des résultats.

1.4 Analyse géographique des données textuelles

La dimension géographique constitue un axe central du projet RADAR, qui vise à mettre en relation les contenus textuels des offres avec leur localisation.

1.4.1 Visualisation cartographique

La visualisation cartographique facilite l'exploration des résultats et la compréhension des disparités territoriales. Elle permet de mettre en évidence :

- la concentration des offres par région,

- la spécialisation thématique de certains territoires,
- les dynamiques régionales du marché de l'emploi data et IA.

COLLECTE ET STRUCTURATION DES DONNÉES

Ce chapitre détaille la phase amont du projet RADAR : l'acquisition des données et leur organisation au sein d'un entrepôt structuré. La fiabilité des analyses NLP produites ultérieurement dépend directement de la qualité de cette étape de collecte multi-sources.

2.1 Sources de données

La pertinence de l'analyse RADAR repose sur la diversité de son corpus. Le projet cible trois plateformes majeures du marché de l'emploi en France, chacune apportant une perspective différente sur les métiers de la donnée.

2.1.1 France Travail

Théorie : L'accès aux données via une API (*Application Programming Interface*) garantit une extraction robuste et normée. Contrairement au scraping de pages web, l'API fournit des données typées où chaque attribut (salaire, localisation, code métier) est explicitement défini par le fournisseur.

Implémentation : La collecte est centralisée dans le module `scrapers/`. Le script utilise les identifiants client (Client ID et Secret) pour obtenir un jeton d'accès OAuth2. Les requêtes sont formulées pour cibler spécifiquement les domaines de l'informatique et de la data, récupérant des objets JSON qui sont ensuite convertis en dictionnaires Python.

Éclairage Métier : En tant que premier agrégateur national, France Travail offre une vision exhaustive des besoins des PME et des institutions publiques, constituant un socle de référence pour la comparaison régionale. Son API permet d'accéder à des métadonnées précises comme les codes ROME (ex : M1805 pour le développement informatique), ce qui facilite la segmentation ultérieure du marché.

2.1.2 Hellowork

Théorie : En l'absence d'interface de programmation publique, la technique du *Web Scraping* est employée. Elle consiste à simuler une navigation humaine pour analyser le code source HTML et en extraire les informations sémantiques.

Implémentation : Le script `scraper_hellowork.py` utilise la bibliothèque `BeautifulSoup`. Le processus parcourt les pages de résultats, identifie les balises `<div>` et `` contenant les titres et les descriptions, puis stocke ces informations brutes. Une gestion des en-têtes (*headers*) est mise en place pour simuler un navigateur réel.

Éclairage Métier : Hellowork est une source privilégiée pour le secteur privé et les startups. C'est sur cette plateforme que l'on observe la plus grande variété de "buzzwords" techniques et d'exigences en termes de frameworks récents.

2.1.3 Emploi Territorial

Théorie : Le secteur public, en particulier l'emploi public territorial, possède ses propres circuits de diffusion. L'extraction sur cette source nécessite un nettoyage plus poussé en raison de la présence de nombreuses mentions réglementaires dans les textes, avec une attention particulière à la séparation entre le descriptif du poste et ces mentions réglementaires.

Implémentation : Le scraper dédié à la plateforme Emploi Territorial extrait les annonces des collectivités locales. Le code traite les spécificités de mise en page de ce portail pour garantir que seul le contenu informatif lié aux missions et compétences est conservé pour l'analyse textuelle.

Éclairage Métier : L'inclusion de cette source permet de mettre en lumière les besoins en "Data" des collectivités locales, souvent liés à l'Open Data ou à la gestion des services urbains.

2.2 Méthodes de collecte

Le processus de collecte a été conçu pour être à la fois automatisé et itératif, permettant la mise à jour régulière de l'entrepôt de données.

2.2.1 Collecte via API

Le flux de données est géré par des requêtes paginées. L'implémentation assure le traitement des réponses par lots, ce qui permet de collecter plusieurs centaines d'offres en une seule exécution tout en respectant les limites de charge des serveurs distants.

2.2.2 Web scraping HTML

Pour les sources web non structurées, le script utilise des sélecteurs CSS précis pour extraire l'information. L'implémentation prévoit la récupération systématique de l'URL de l'offre, servant à la fois de source de vérification et d'identifiant pour la gestion des doublons.

2.3 Nettoyage et normalisation des données

Les données collectées présentent des niveaux de qualité hétérogènes selon leur source. Une phase de nettoyage et de normalisation est donc appliquée afin de transformer les données hétérogènes en un corpus homogène exploitable par des algorithmes de NLP.

2.3.1 Traitement des champs textuels

Les descriptions brutes contiennent souvent des scories techniques (balises HTML, entités de caractères, résidus d'encodage). L'implémentation utilise des expressions régulières pour normaliser le texte, supprimant par exemple les balises de mise en forme (``, ``, etc.) afin de ne conserver que la chaîne de caractères brute.

2.3.2 Gestion des doublons et des valeurs manquantes

Théorie : La multidiffusion des offres sur différentes plateformes peut engendrer des doublons qui biaiserait les analyses statistiques.

Implémentation : Un identifiant unique (UID) est calculé pour chaque annonce. Avant l'insertion dans l'entrepôt, une procédure de vérification de l'existence de l'UID est déclenchée. Si l'offre est déjà présente dans la base, l'insertion est ignorée, garantissant ainsi l'unicité des données analysées.

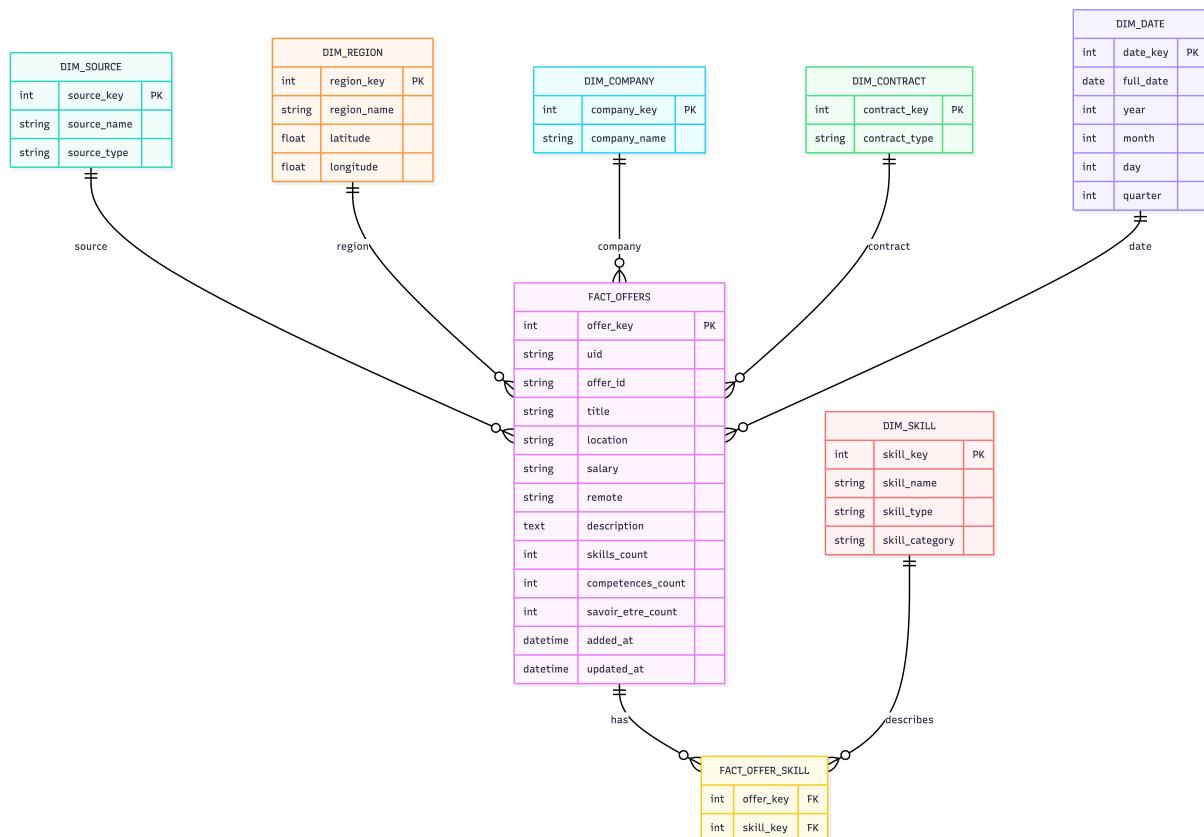
2.4 Modélisation et stockage

2.4.1 Schéma de la base de données

Le projet s'appuie sur une structure relationnelle simple mais efficace, centrée sur la table des offres et conçue pour supporter des requêtes analytiques complexes.

Implémentation : Le fichier `db_manager.py` définit la structure de la table `offers`. Celle-ci regroupe les attributs essentiels : titre, description, entreprise, localisation, source et région.

Le schéma logique de la base est présenté en Figure 2.1.



2.4.2 Entrepôt de données et choix technologiques

Théorie : Pour un projet de recherche, l'entrepôt doit concilier performance et portabilité.

Implémentation : Le choix technique s'est porté sur SQLite (`jobs.db`). Contrairement à un serveur de base de données traditionnel, SQLite stocke l'intégralité des données dans un fichier unique.

Éclairage Métier : Ce choix garantit la reproductibilité totale du projet. L'entrepôt peut être déplacé, archivé ou partagé sans nécessiter de configuration serveur complexe. Cette approche simplifie également le déploiement via le conteneur Docker, car la base de données est directement embarquée et prête à l'emploi pour l'application de visualisation.

MÉTHODES DE TEXT MINING ET NLP MISES EN ŒUVRE

Ce chapitre expose les traitements appliqués au corpus pour le transformer en un ensemble de données exploitables. L'enjeu est de passer d'une description d'offre non structurée à une représentation mathématique permettant le regroupement thématique des métiers.

3.1 Prétraitement et préparation des documents de compétence

Le projet RADAR privilégie une approche ciblée sur les entités sémantiques à forte valeur ajoutée : les compétences (*Hard Skills*) et le savoir-être (*Soft Skills*).

3.1.1 Construction des documents synthétiques

Rappel théorique : La réduction du bruit linguistique (mots de liaison, formules de politesse, jargon administratif) est une étape qui permet d'améliorer la précision des algorithmes de partitionnement. Reconstruire un "document synthétique" composé uniquement de mots-clés métier permet de focaliser l'analyse sur le contenu technique des offres.

Implémentation : La méthode `prepare_skill_documents` de la classe `OfferClusterer` crée des "documents de compétences". Au lieu de procéder à une tokenisation standard sur le texte brut, elle fusionne les listes de `competences` et de `savoir_être` préalablement extraites. Ces listes sont converties en une chaîne de caractères unique par offre, créant ainsi un "profil de compétences" purifié pour chaque annonce.

Éclairage Métier : Cette stratégie garantit que le clustering se concentre exclusivement sur le profil de poste et les exigences techniques, neutralisant les disparités de style rédactionnel entre les différents recruteurs ou plateformes.

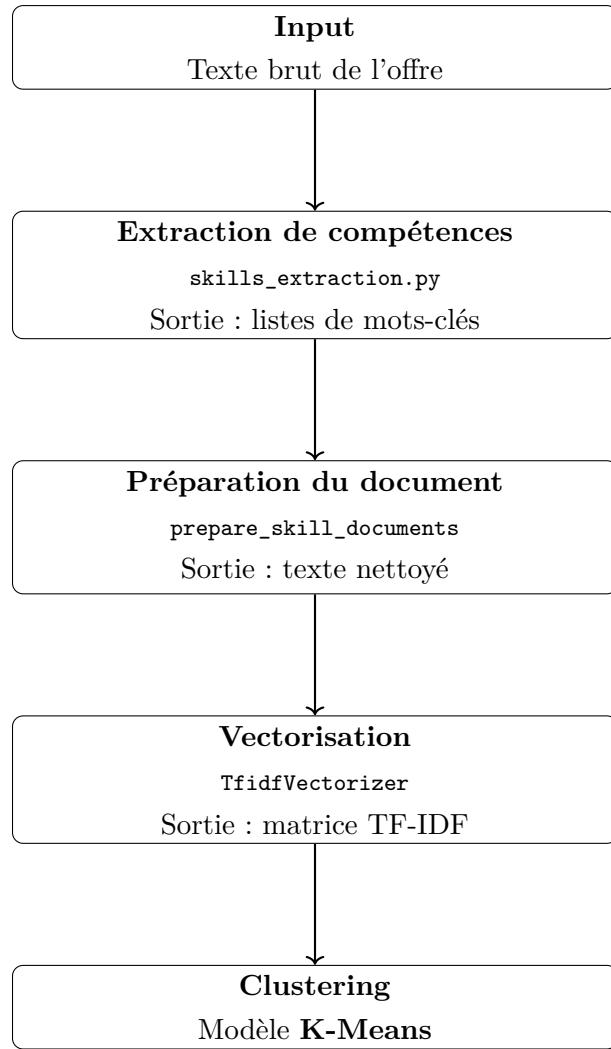


FIGURE 3.1 – Pipeline NLP de transformation des offres d’emploi en représentations vectorielles pour le clustering

3.2 Vectorisation par pondération TF-IDF

3.2.1 Représentation vectorielle du corpus

Rappel théorique : La méthode TF-IDF (*Term Frequency-Inverse Document Frequency*) transforme un texte en vecteur numérique en pondérant l’importance de chaque terme. Un mot rare et spécifique (ex. : "Kubernetes") obtient un poids supérieur à un mot omniprésent (ex. : "données").

Implémentation : Le projet utilise la classe `TfidfVectorizer` de `scikit-learn`. Comme les documents sont déjà pré-nettoyés par le module d’extraction, le vectoriseur transforme ces profils de compétences en une matrice numérique creuse (*sparse matrix*). Chaque colonne de cette matrice représente une dimension du vocabulaire métier du corpus.

3.3 Analyse thématique

L'objectif est de segmenter le marché de l'emploi en grandes familles de métiers de manière automatisée et non supervisée.

3.3.1 Algorithme des K-Means

Rappel théorique : L'algorithme des K-Means regroupe les vecteurs par proximité (distance euclidienne) autour de centres de gravité appelés "centroïdes".

Implémentation : Le choix s'est porté sur une initialisation du modèle à 6 clusters. Ce paramètre, bien qu'empirique, correspond à la segmentation métier observée sur le marché de la Data (Data Science, Engineering, Analysis, Management, Cloud, MLOps). Pour confirmer la pertinence statistique de ce découpage, le script calcule le score de silhouette, qui mesure la qualité de la séparation des groupes.

Éclairage Métier : Dans le cadre de ce projet, le score de silhouette obtenu est de **0.042**. Bien que statistiquement faible, ce score reflète la porosité réelle entre les métiers de la donnée. Des compétences comme "SQL", "Python" ou "gestion de projet" sont transversales à tous les clusters, rendant la séparation mathématique complexe malgré une cohérence métier évidente.

3.3.2 Réduction de dimensionnalité (PCA)

Rappel théorique : Une matrice TF-IDF possède trop de dimensions pour être représentée graphiquement. L'Analyse en Composantes Principales (PCA) permet de projeter ces données sur un plan en deux dimensions (2D) tout en conservant l'essentiel de l'information (variance).

Implémentation : La méthode `PCA(n_components=2)` est appliquée aux vecteurs TF-IDF. Elle génère les coordonnées (x, y) utilisées pour la visualisation graphique dans l'application web, rendant les clusters immédiatement interprétables.

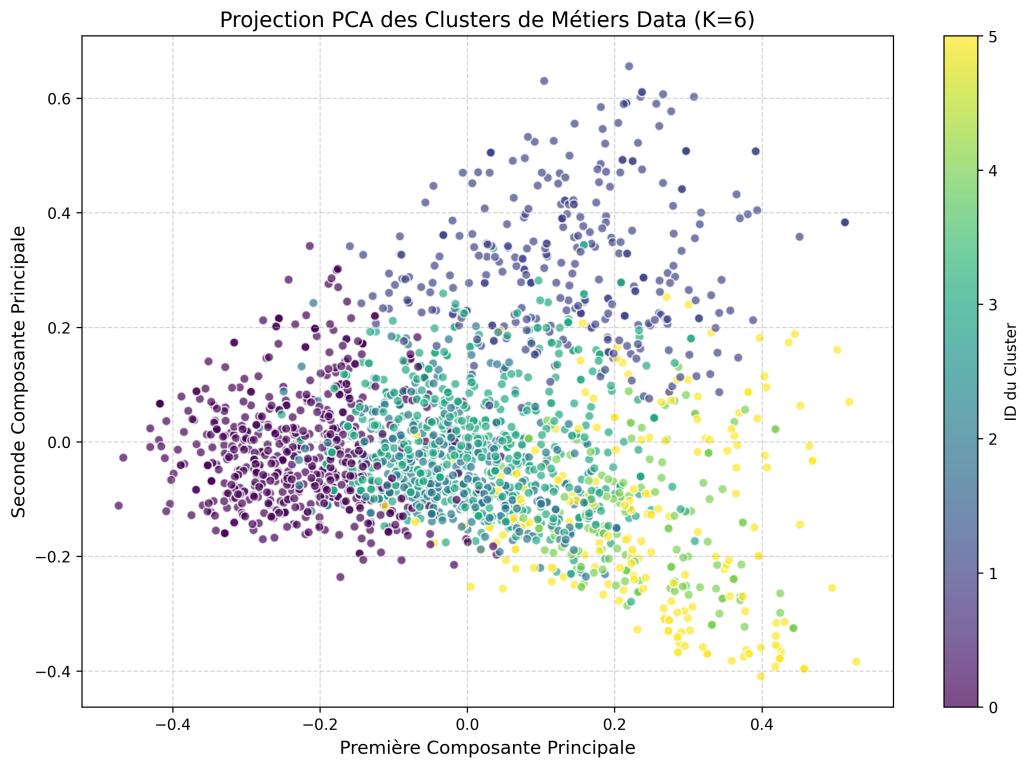


FIGURE 3.2 – Projection PCA des 2 500 offres d’emploi en deux dimensions.

3.4 Caractérisation métier des clusters

Éclairage Métier : L’analyse des termes dominants par cluster confirme la pertinence de la segmentation non supervisée. Les Figures 3.3 et 3.4 illustrent cette spécialisation :

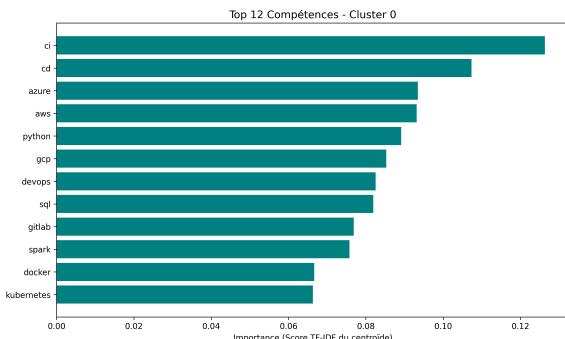


FIGURE 3.3 – Profil de compétences du Cluster 0 (Data Science).

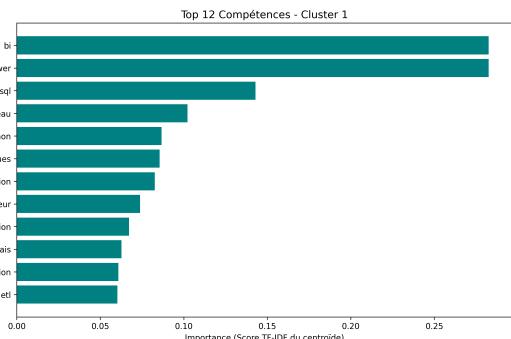


FIGURE 3.4 – Profil de compétences du Cluster 1 (Data Engineering).

Le Cluster 0, dominé par des outils de modélisation (profil « Data Science / Analytics »), se distingue nettement du Cluster 1, orienté vers la gestion des infrastructures et des flux de données (profil « Data Engineering / Infrastructure »).

APPLICATION WEB ET VISUALISATION

Ce chapitre présente l'interface "DataJobs Analytics", conçue pour rendre les analyses NLP et géographiques accessibles via une interface interactive. L'enjeu est de transformer une base de données complexe en un outil d'aide à la décision fluide et intuitif.

4.1 Architecture globale de l'application

4.1.1 Framework et Philosophie de développement

Rappel théorique : Le choix d'un framework "low-code" comme **Streamlit** permet de coupler l'analyse de données en Python à une interface web sans la complexité d'un développement front-end lourd. Cela garantit une réactivité maximale entre le traitement des données et leur affichage.

Implémentation : L'application adopte une structure multipages organisée par domaines fonctionnels (Analytics, Assistant, Exploration). Une attention particulière a été portée à l'esthétique via l'injection de CSS personnalisé (méthode `inject_premium_css`). L'usage de composants visuels modernes — tels que des cartes d'indicateurs (KPIs) avec dégradés, une barre de navigation stylisée et un thème sombre — permet de proposer un outil professionnel et immersif.

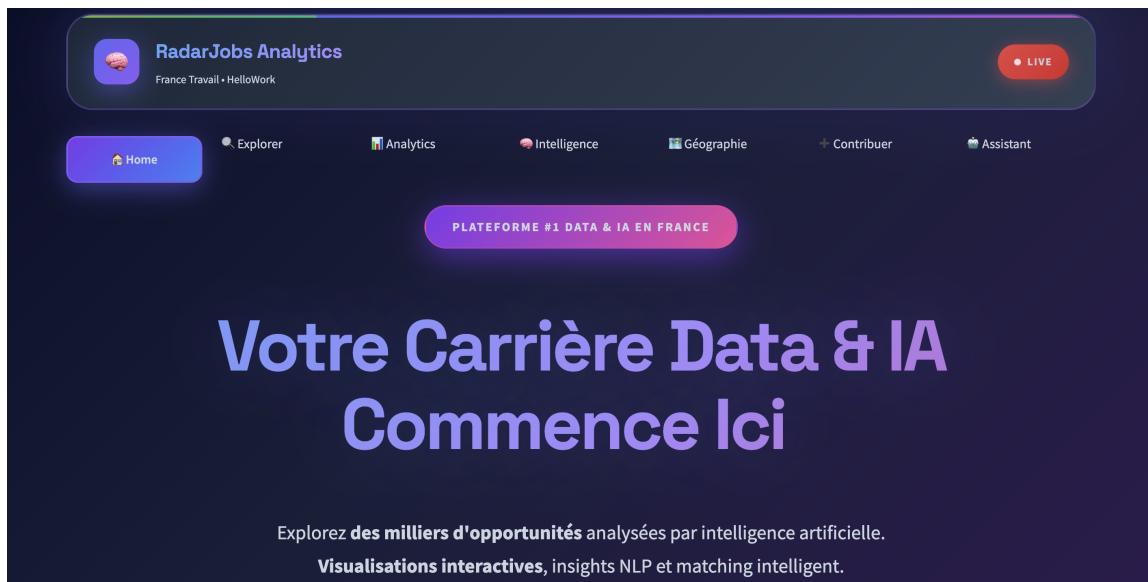


FIGURE 4.1 – Interface d'accueil du projet RADAR : intégration d'un design personnalisé via CSS et navigation modulaire.

4.2 Analyses statistiques et géographiques

4.2.1 Tableau de bord interactif (Plotly)

Implémentation : L'onglet "Analytics" exploite la bibliothèque **Plotly** pour générer des graphiques dynamiques. Contrairement aux images statiques du rapport, l'application permet à

l'utilisateur de filtrer les données en temps réel (par type de contrat ou par région) et d'explorer les compétences via des histogrammes interactifs.

- **Extraction structurée** : salaire, type de contrat et localisation.
- **Évaluation métier** : détection automatique du niveau d'expérience requis et des responsabilités de management.
- **Analyse des compétences** : identification des points clés pour aider le candidat à adapter son profil.

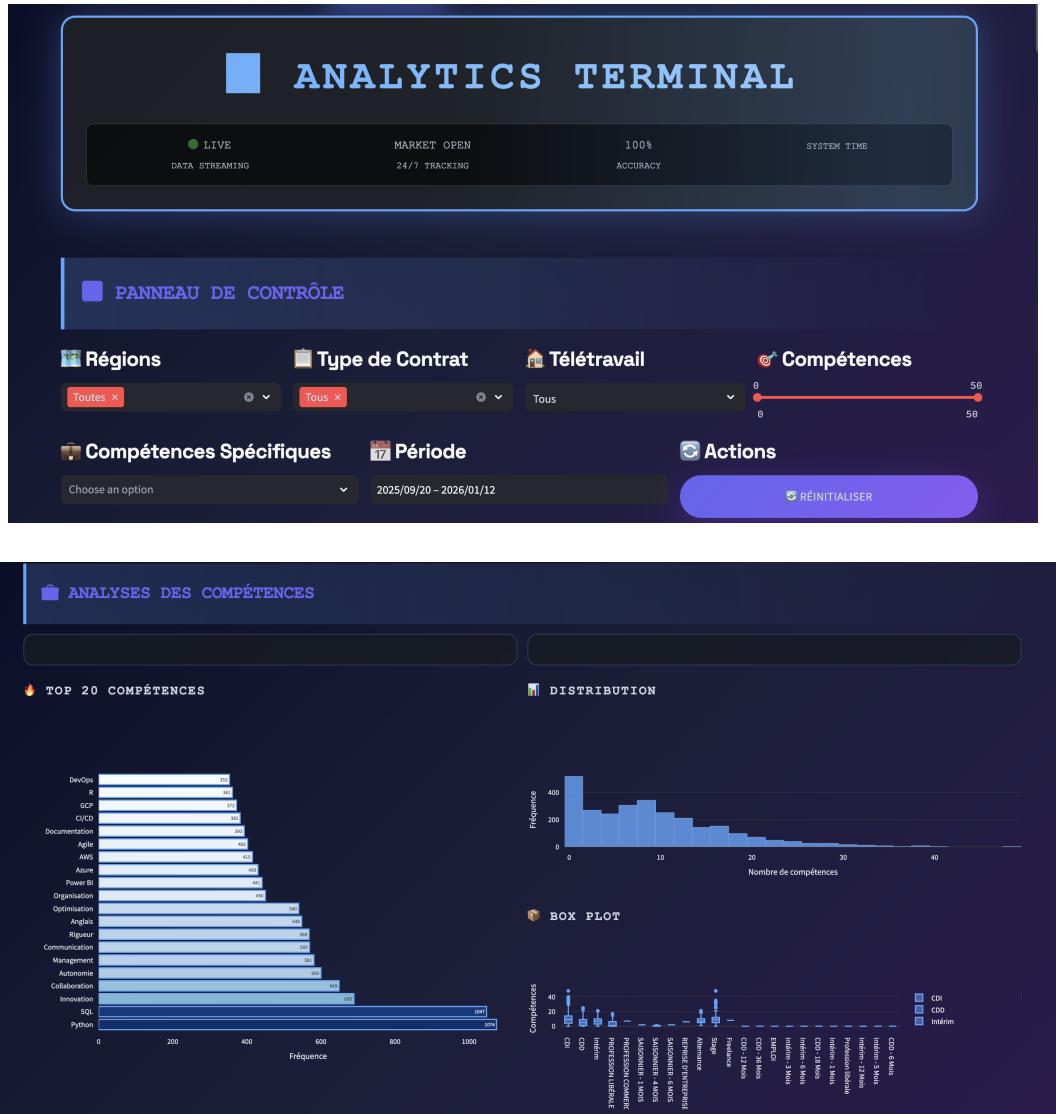


FIGURE 4.2 – Tableau de bord Analytics : (Haut) Filtres dynamiques et analyse des Hard Skills. (Bas) Répartition des types de contrats sur le marché.

4.2.2 Cartographie régionale (Folium)

Implémentation : La dimension territoriale est matérialisée par des cartes de France générées avec **Folium**. En s'appuyant sur la vue SQL `v_offers_by_region`, l'application affiche la densité des offres par région sous forme de carte choroplète, permettant d'identifier immédiatement les bassins d'emploi les plus dynamiques.

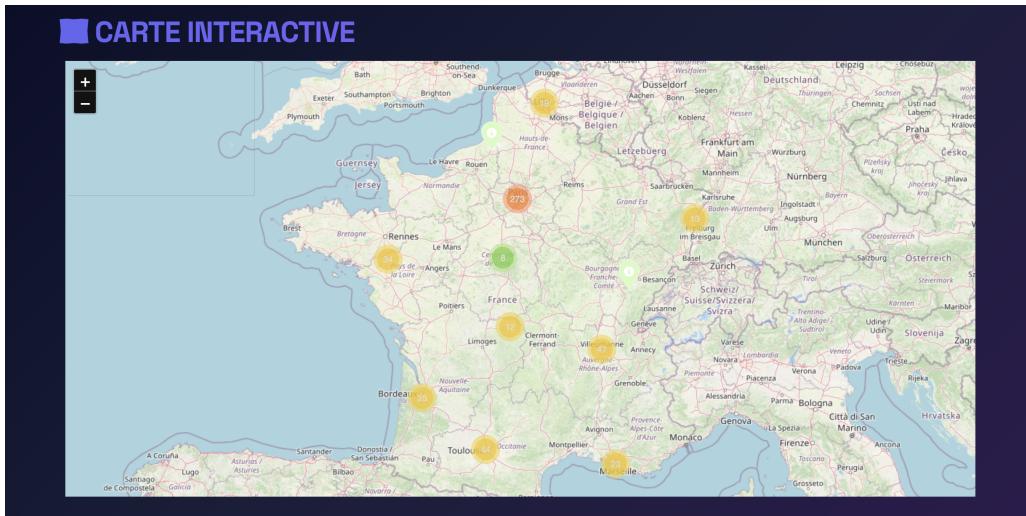


FIGURE 4.3 – Répartition géographique des opportunités : carte choroplète interactive identifiant les principaux pôles d'activité Data.

4.3 L'Intelligence artificielle au service de l'utilisateur

Cette section détaille l'intégration du modèle de langage **Mistral-Large** via la bibliothèque `pydantic_ai`, déclinée en deux outils complémentaires.

4.3.1 Analyse sémantique (Onglet Intelligence)

Implémentation : Ce module permet de transformer une offre d'emploi non structurée en une fiche technique exploitable. L'agent IA analyse le texte brut pour extraire des entités précises :

- **Données contractuelles** : salaire, localisation et type de contrat.
- **Évaluation métier** : détection automatique du niveau d'expérience requis (Junior, Senior, etc.) et des responsabilités de management.
- **Profil de poste** : identification des *Hard Skills* et *Soft Skills* réellement attendus.

4.3.2 Aide à la candidature (Onglet Assistant)

Implémentation : Ce module exploite les capacités génératives du LLM pour assister l'utilisateur dans sa recherche. En s'appuyant sur les informations extraites de l'annonce, l'agent génère une **lettre de motivation personnalisée**.

Éclairage Métier : L'enjeu est ici de proposer un argumentaire cohérent avec les compétences identifiées par l'IA dans la phase d'analyse, permettant au candidat de gagner en efficacité tout en personnalisant chaque candidature.

Jean MARTIN Paris, France jean.martin.datascience@gmail.com | +33 6 12 34 56 78
12 January 2026
À l'attention du Service Recrutement LIDL 92 - CHATENAY MALABRY
Objet : Candidature au poste de Data Scientist (H/F)
Madame, Monsieur,
Fort de cinq années d'expérience en data science appliquée à des enjeux business concrets, je me permets de vous soumettre ma candidature pour le poste de Data Scientist (H/F) au sein de LIDL. Votre recherche d'un profil alliant rigueur technique, curiosité opérationnelle et capacité à traduire des données en solutions impactantes résonne particulièrement avec mon parcours. Mon expertise en modélisation prédictive et optimisation de processus, couplée à une maîtrise avancée de Python et SQL, me permet d'envisager une contribution immédiate à vos projets data au sein de la #TeamLidl.
Chez NovaData Consulting, où j'exerce en tant que Data Scientist depuis 2021, j'ai conçu et déployé des modèles de machine learning pour des acteurs du retail et de la finance, des secteurs où l'optimisation des coûts et la personnalisation des offres sont critiques – des enjeux que je sais centraux pour LIDL. Par exemple, j'ai développé un modèle de prédiction des ruptures de stock pour un client de la grande distribution, réduisant les pertes de 15 % grâce à une analyse fine des historiques de ventes et des données logistiques (Python, scikit-learn, SQL). J'ai également mis en production un système de scoring client pour une banque, en utilisant des techniques de classification supervisée (XGBoost, pandas) et en assurant son suivi via des dashboards Looker – une compétence que je pourrais transposer à vos outils internes. Auparavant, chez Insight Analytics, j'ai participé à des projets de NLP pour analyser des avis clients, une expérience qui m'a permis de développer une sensibilité aux cas d'usage concrets de l'IA, comme ceux que vous mentionnez dans votre offre (LLMs, IA générative). Mon approche combine rigueur statistique (Master 2 Data Science & IA à Paris-Saclay) et pragmatisme opérationnel, avec une attention particulière portée à la collaboration avec les métiers – une dimension clé de votre poste.
Ce qui m'attire particulièrement chez LIDL, c'est votre ambition de réinventer le retail par la data, en plaçant l'IA au service de l'efficacité opérationnelle. Votre projet d'automatisation des processus via Python, SQL et Looker rejoint directement mes réalisations chez NovaData, où j'ai automatisé des rapports d'analyse pour un client retail en utilisant des scripts Python et des requêtes SQL optimisées. Votre insistance sur la curiosité et l'innovation correspond à ma démarche : par exemple, j'ai récemment exploré les API de modèles de langage (LLMs) pour un prototype de chatbot interne, une piste que je serais ravi d'approfondir chez LIDL pour des cas d'usage comme l'assistance aux équipes en magasin. Enfin, votre culture d'entreprise, axée sur l'impact concret et le travail d'équipe, est en phase avec ma manière de fonctionner : chez Insight Analytics, j'ai collaboré avec des chefs de produit pour traduire leurs besoins en solutions data, une expérience que je mettrai à profit pour comprendre et répondre aux enjeux des services métiers de LIDL.
Je serais honoré de pouvoir échanger avec vous sur la manière dont mon profil pourrait s'intégrer à vos projets. Disponible pour un entretien à votre convenance, je reste à votre disposition pour toute information complémentaire.
Veuillez agréer, Madame, Monsieur, l'expression de mes salutations distinguées.

Jean MARTIN

FIGURE 4.4 – Interface de l'Assistant : génération d'une lettre de motivation grâce à l'analyse sémantique de l'annonce.

DÉPLOIEMENT ET REPRODUCTIBILITÉ

Ce chapitre présente les choix techniques effectués pour garantir la portabilité du projet RADAR. L'enjeu est de permettre à n'importe quel utilisateur ou administrateur système de déployer l'intégralité de la chaîne (base de données, moteur NLP et interface web) sans conflit de dépendances.

5.1 Conteneurisation avec Docker

5.1.1 Motivations du choix de Docker

Rappel théorique : La conteneurisation permet d'encapsuler une application et ses dépendances dans une unité isolée (le conteneur). Contrairement à une machine virtuelle, Docker partage le noyau du système hôte, ce qui le rend léger et rapide.

Implémentation : Le projet utilise une image basée sur `python:3.11-slim`. Ce choix permet de réduire la taille de l'image finale tout en conservant les bibliothèques nécessaires aux calculs scientifiques (NumPy, Scikit-learn).

5.1.2 Description de l'image (Dockerfile)

L'image Docker est construite de manière à automatiser chaque étape :

- Installation des dépendances système (compilateurs pour certaines bibliothèques de calcul).
- Copie des sources du projet et de la base de données SQLite embarquée.
- Exposition du port 8501, utilisé par défaut par Streamlit.

5.2 Gestion des dépendances et configuration

5.2.1 Dépendances Python (`requirements.txt`)

La reproductibilité repose sur un fichier `requirements.txt` strict. On y retrouve notamment :

- **Streamlit** et **Plotly** pour l'interface et la visualisation.
- **Scikit-learn** pour le moteur de clustering et la vectorisation TF-IDF.
- **Pydantic-AI** pour la gestion des interactions structurées avec Mistral AI.

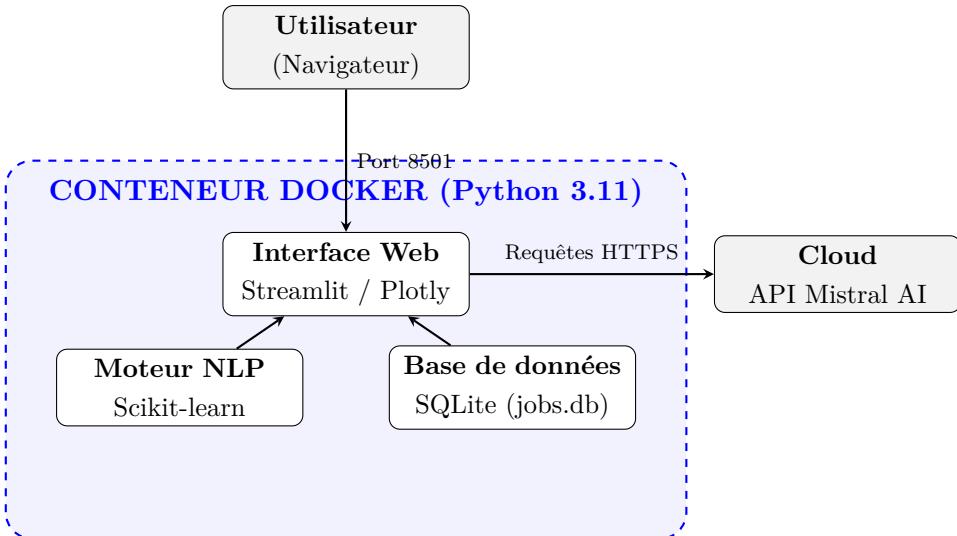


FIGURE 5.1 – Architecture de déploiement : isolation des services et communication externe.

5.3 Installation et exécution

5.3.1 Lancement de l'application

Implémentation : Grâce à Docker, le déploiement se résume à deux commandes :

1. `docker build -t radar-app .` (Construction de l'image)
2. `docker run -p 8501:8501 -env-file .env radar-app` (Lancement)

5.4 Limites et contraintes techniques

Éclairage Métier : Le choix de SQLite comme moteur de base de données facilite la portabilité car la base est un simple fichier inclus dans le conteneur. Toutefois, pour une mise en production à très grande échelle (plusieurs millions d'offres), il serait pertinent de migrer vers un serveur de base de données dédié comme PostgreSQL, ce que l'architecture actuelle permettrait avec des modifications mineures dans le module `db_manager.py`.

ANALYSE MÉTIER ET DISCUSSION

Ce chapitre propose une interprétation des résultats obtenus à travers la chaîne de traitement RADAR. Il s'agit de confronter les données extraites à la réalité du marché de l'emploi Data en France et d'évaluer l'apport de l'outil pour les différents acteurs.

6.1 Résultats principaux et dynamiques territoriales

L'analyse spatiale couplée à la nature des contrats révèle des disparités importantes selon les régions.

6.1.1 Volume global et pôles d'attractivité

Sans surprise, l'Île-de-France domine le marché en volume, suivie immédiatement par la région Auvergne-Rhône-Alpes. Ces deux pôles concentrent la majorité des offres collectées, confirmant la centralisation des activités de haute technologie autour des métropoles de Paris et Lyon.

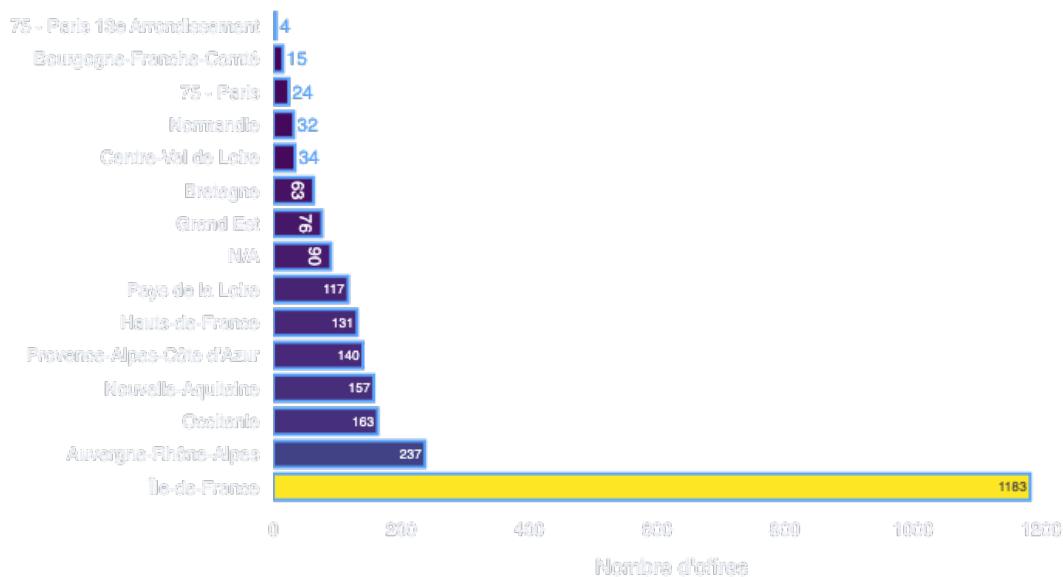


FIGURE 6.1 – Classement des régions par volume d'offres : prédominance des pôles franciliens et rhônalpins.

6.1.2 Focus sur l'insertion : alternance et stages

Les *heatmaps* générées par l'application permettent de visualiser spécifiquement les opportunités pour les profils juniors :

- **Alternance** : on observe un maillage territorial plus équilibré, avec un dynamisme marqué dans les régions à fort tissu industriel (AuRA, Hauts-de-France).

- **Stages** : la concentration est massivement parisienne, s'expliquant par la présence des sièges sociaux des grands groupes et des centres de R&D majeurs.

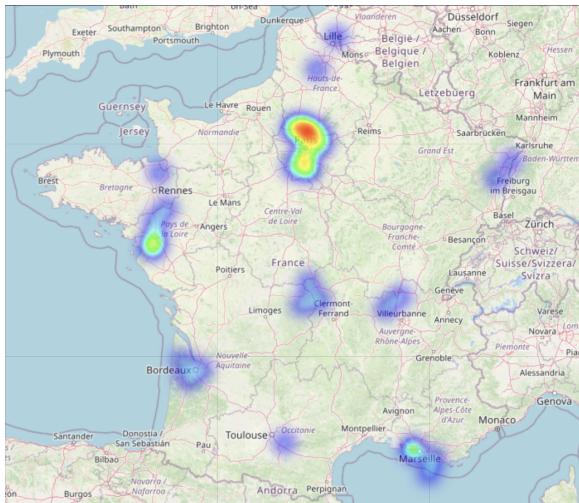


FIGURE 6.2 – Densité des offres en alternance.

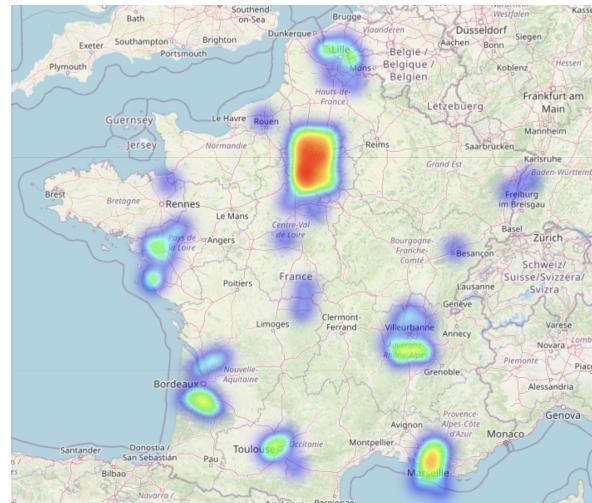


FIGURE 6.3 – Densité des offres de stage.

6.2 Étude de cas : La région Auvergne-Rhône-Alpes (AuRA)

Le projet RADAR permet une vue hiérarchique 360° (*Sunburst*) qui offre une granularité inédite sur le bassin d'emploi local, particulièrement pertinent pour les étudiants de l'Université Lyon 2.

6.2.1 L'apport de la vue *Sunburst*

L'analyse de la région AuRA montre un écosystème Data complet. La vue hiérarchique permet de décomposer les besoins par sous-secteurs et par clusters. On observe que Lyon agit comme un hub centralisateur, tandis que des pôles comme Grenoble ou Clermont-Ferrand présentent des profils de compétences plus ciblés sur le *Manufacturing* et la R&D.

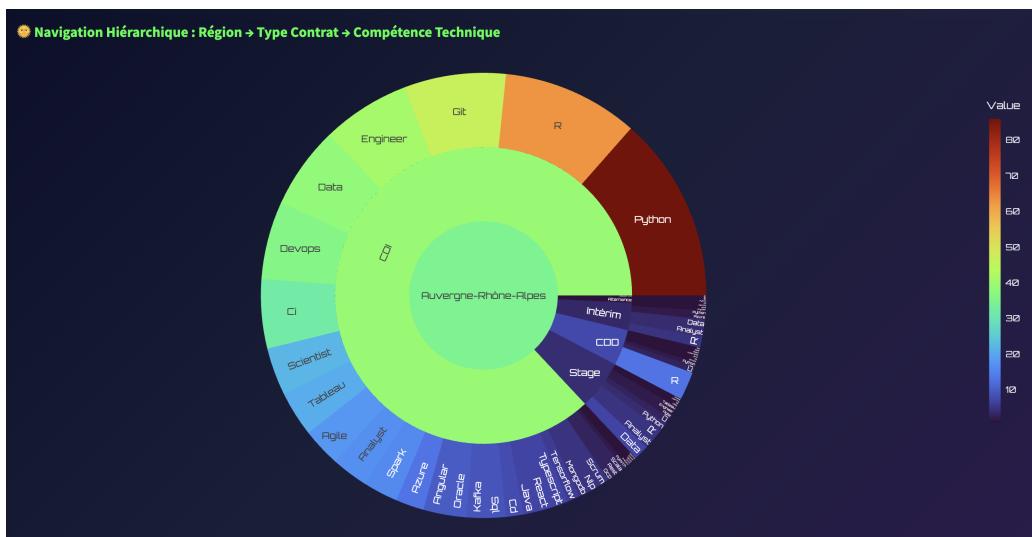


FIGURE 6.4 – Vue hiérarchique 360° des métiers de la donnée en région Auvergne-Rhône-Alpes.

6.3 Discussion et limites

Malgré la pertinence des résultats observés, le projet RADAR fait face à plusieurs contraintes qui nuancent l'interprétation des données.

6.3.1 Limites des données

- **Bruit et redondance :** Malgré les filtres mis en place, la multidiffusion des offres sur différentes plateformes peut engendrer des doublons partiels (textes légèrement modifiés). Cela peut artificiellement gonfler le poids de certaines compétences dans les statistiques.
- **Qualité de la source brute :** La qualité de l'analyse dépend de la précision rédactionnelle du recruteur. Une offre trop succincte ou mal structurée limite la capacité d'extraction du module NLP et peut conduire à un classement erroné dans les clusters.
- **Fraîcheur des données :** Le marché de la data évolue très vite. Les données collectées sont une « photographie » à un instant T ; elles ne permettent pas encore d'analyser des tendances historiques sur plusieurs années sans un processus de collecte sur le long terme.

6.3.2 Limites méthodologiques

- **Porosité des clusters :** Le score de silhouette de 0,042 témoigne d'une difficulté mathématique à séparer strictement les métiers. La réalité du terrain montre que les frontières entre un « Data Scientist » et un « Data Engineer » sont souvent floues dans les descriptions d'offres.
- **Approche par dictionnaire :** L'extraction de compétences repose en partie sur des listes prédéfinies. Cette méthode peut passer à côté de technologies émergentes très récentes qui ne figureraient pas encore dans notre référentiel technique.
- **Complexité sémantique :** L'IA (Mistral) aide à comprendre le contexte, mais l'analyse reste sensible à l'ironie, aux formulations indirectes ou aux exigences implicites qui ne sont pas explicitement formulées dans le texte brut de l'annonce.

6.3.3 Apports métier du projet

Le projet RADAR ne se contente pas de collecter des données ; il propose une véritable chaîne de valeur transformant l'information brute en connaissance actionnable.

6.3.4 Pour les candidats et demandeurs d'emploi

- **Aide à la décision stratégique :** Grâce aux visualisations géographiques et thématiques, le candidat peut identifier les régions où son profil est le plus en adéquation avec la demande, optimisant ainsi sa mobilité géographique.
- **Gain de productivité :** L'assistant de carrière (Mistral AI) automatise la rédaction de documents de candidature. En s'appuyant sur les compétences réellement extraites de l'offre, il garantit une lettre de motivation ciblée, augmentant les chances de franchir l'étape des algorithmes de filtrage (ATS).

- **Acculturation au marché** : La vue hiérarchique 360° permet aux profils juniors de comprendre la structure des métiers de la Data et de découvrir des technologies connexes auxquelles ils n'auraient pas pensé.

6.3.5 Pour les recruteurs et acteurs RH

- **Veille concurrentielle (*Benchmarking*)** : Les entreprises peuvent utiliser RADAR pour analyser les pratiques de recrutement de leurs concurrents : quelles compétences demandent-ils ? Quels types de contrats privilégient-ils ?
- **Standardisation sémantique** : L'outil permet de traduire des descriptions d'offres souvent hétérogènes en un dictionnaire de compétences standardisé, facilitant ainsi la comparaison entre différents postes.

6.3.6 Pour les acteurs publics et institutionnels

- **Observatoire territorial** : Pour des structures comme l'Université Lyon 2 ou les agences de développement économique, RADAR offre une photographie en temps réel des besoins en compétences sur le territoire Auvergne-Rhône-Alpes, permettant d'ajuster les offres de formation aux besoins du marché local.

En somme, RADAR agit comme un pont sémantique entre l'offre et la demande, utilisant le NLP pour réduire l'**asymétrie d'information** sur le marché de l'emploi.

CONCLUSION ET PERSPECTIVES

7.1 Synthèse du projet

Le projet RADAR a permis de concevoir et de déployer une solution complète d'analyse du marché de l'emploi Data et IA. En partant d'un corpus de plus de 2 500 offres d'emploi brutes, nous avons réussi à :

- **structurer l'information** en passant d'un texte libre à une base de données SQL et un dictionnaire de compétences standardisé.
- **cartographier le marché** en identifiant 6 clusters métiers cohérents et visualiser leur répartition sur le territoire français.
- **démocratiser l'IA** en proposant une interface "Premium" intégrant des agents conversationnels (Mistral AI) pour assister concrètement les candidats dans leur recherche d'emploi.

7.2 Considérations éthiques et limites techniques

7.2.1 Éthique et respect des plateformes

Le projet RADAR s'est heurté à des dispositifs anti-scraping sur des plateformes comme l'APEC ou Indeed. Conformément à une déontologie numérique responsable, nous avons choisi de ne pas forcer le contournement de ces protections. L'intégrité du projet repose sur le respect des *Terms of Service* des sites et sur l'utilisation prioritaire d'API officielles lorsque celles-ci sont disponibles.

7.2.2 Utilisation responsable des LLM

L'utilisation de modèles comme Mistral-Large soulève notamment des questions de souveraineté des données et de biais :

- **Confidentialité** : Les données d'offres étant publiques, le risque est faible, mais une extension aux CV personnels nécessiterait une anonymisation préalable stricte.
- **Biais** : Nous sommes conscients que les LLM peuvent reproduire des biais présents dans les descriptions d'offres (stéréotypes de genre ou de formation). Un axe de travail futur serait l'implémentation de filtres de "débiaisage" en sortie de l'assistant.

7.3 Perspectives d'amélioration

Bien que fonctionnel et déployable, le système RADAR constitue un socle qui peut être enrichi selon plusieurs axes.

7.3.1 Enrichissements supplémentaires

Pour transformer RADAR en un véritable observatoire socio-économique, nous envisageons d'intégrer des sources de données complémentaires :

- **Données démographiques (Insee)** : L'hybridation avec les bases de l'Insee permettrait de mettre en corrélation le volume d'offres avec la densité de population active ou le coût de la vie par région. Cela permettrait de calculer un « indice de tension » ou un ratio d'opportunités par habitant, offrant une lecture plus fine de l'attractivité réelle des territoires.
- **Cartographie de l'offre de formation** : L'intégration des centres de formation (universités, écoles d'ingénieurs, bootcamps) spécialisés en Data et IA permettrait de visualiser l'adéquation entre le "flux" de diplômés et les besoins des entreprises locales.

Éclairage Métier : Cette double perspective permettrait d'identifier des « zones de déséquilibre » où la demande des entreprises est forte mais l'offre de formation ou la densité de talents est insuffisante, guidant ainsi les politiques publiques de formation.

7.3.2 Gamification et engagement communautaire

Une piste majeure consiste à transformer RADAR en une plateforme collaborative via un système de gamification :

- **Système d'XP** : L'application intègre une logique de points d'expérience (XP). L'idée est d'inciter les utilisateurs à alimenter la base de données via l'onglet "Contribuer". Chaque nouvelle offre valide soumise rapporterait des XP.
- **Économie de fonctionnalités** : Ces points serviraient de "crédit" pour débloquer des services avancés dans l'onglet Assistant. Par exemple, un utilisateur débuterait avec 100 XP (permettant une première analyse/lettre), mais devrait contribuer pour accéder à de nouveaux jetons de génération.

7.3.3 Coach Carrière IA (LLM augmenté)

Au-delà de la lettre de motivation, l'assistant pourrait évoluer vers un véritable Coach Carrière :

- **Analyse d'écart (*Gap Analysis*)** : En comparant le CV de l'utilisateur avec le cluster visé, l'IA pourrait recommander spécifiquement quels *Hard Skills* ou *Soft Skills* acquérir pour monter en compétence.
- **Optimisation de profil** : Conseils personnalisés pour améliorer l'impact sémantique d'un CV face aux outils de recrutement automatisés.

7.3.4 Architecture logicielle et Clean Code

Pour assurer la pérennité de RADAR, une refactorisation complète en mode package est envisagée :

- **Séparation stricte des responsabilités** : Externalisation complète des requêtes de scraping, des dictionnaires de *stop words* et des configurations sémantiques.
- **Scalabilité métier et géographique** : Cette architecture permettrait d'étendre RADAR à d'autres pays (gestion multi-langues) ou à d'autres secteurs d'activité que la Data sans modifier le cœur du moteur.

7.4 Mot de la fin

Le projet RADAR s'inscrit pleinement dans les enjeux actuels de la science des données. Au-delà de l'aspect technique, il offre une réponse concrète à la complexité croissante du recrutement dans la Data. En rendant l'information plus transparente et accessible, RADAR se positionne non seulement comme un outil d'analyse, mais aussi comme un véritable facilitateur de carrière pour les futurs experts en science des données.