Classification Automatique de CV par Apprentissage Automatique

GORRY Ayoub
Ayoub.gorry@etu.uae.ac.ma

Encader par: Pr.KHAMJANE Aziz
akhamjane@uae.ac.ma

École Nationale Des Sciences Appliquées d'Al Hoceima

Résumé

L'idée principale de ce projet est de développer un système capable de classifier automatiquement des CV dans différents domaines professionnels en utilisant le traitement automatique du langage naturel (NLP) et l'apprentissage automatique.

I. Introduction

Recruter et gérer les candidatures est un défi de taille pour les entreprises et les professionnels des ressources humaines. Trier les CV manuellement prend beaucoup de temps et peut être influencé par des biais humains. Pour répondre à ces enjeux, notre projet propose une solution innovante, avec pour objectifs :

- Classer automatiquement les CV en fonction de leur domaine professionnel.
- Accélérer le processus de tri initial des candidatures.
- Offrir une analyse objective et claire des profils des candidats.

II. Études Précédentes

Les études précédentes dans le domaine de la classification automatique des CV ont mis en évidence plusieurs défis et approches innovantes. Les recherches ont souligné la complexité du traitement des CV en tant que données non structurées, où chaque document présente des mises en page et des formats uniques. Les travaux antérieurs, notamment ceux de Shanuhalli et Noran Mohamed[i], ont exploré l'utilisation des technologies de traitement du langage naturel (NLP) pour rationaliser le processus de criblage des candidats. Ces études ont développé des approches permettant de collecter, stocker et analyser électroniquement de grandes quantités de CV, en utilisant des techniques d'apprentissage automatique et de reconnaissance optique de caractères (OCR) pour extraire et catégoriser automatiquement les informations professionnelles. L'objectif principal était de réduire le temps et les ressources consacrés au tri manuel des candidatures, en proposant des solutions automatisées capables de gérer la diversité et la complexité des documents de candidature.

III. Matériel et Méthodologie

À partir des recherches mentionnées ci-dessus, nous avons collecté des connaissances et des informations liées que nous avons utilisées pour notre travail.

1. Architecture Globale du Système:

La **figure 1** présenté illustre le processus de classification des CV à l'aide de techniques de traitement automatique du langage naturel et de l'apprentissage automatique. Le processus commence par l'utilisateur, qui peut soit télécharger un fichier, soit entrer du texte directement. Ce texte est ensuite soumis à un prétraitement pour nettoyer et préparer les données. Une fois le texte prétraité, il est converti en vecteurs numériques en utilisant la méthode TF-IDF (Term Frequency-Inverse Document Frequency). Ces vecteurs sont alors analysés par un modèle d'apprentissage automatique pré-entraîné qui prédit la catégorie du CV. Enfin, le résultat de la prédiction est renvoyé à l'utilisateur.

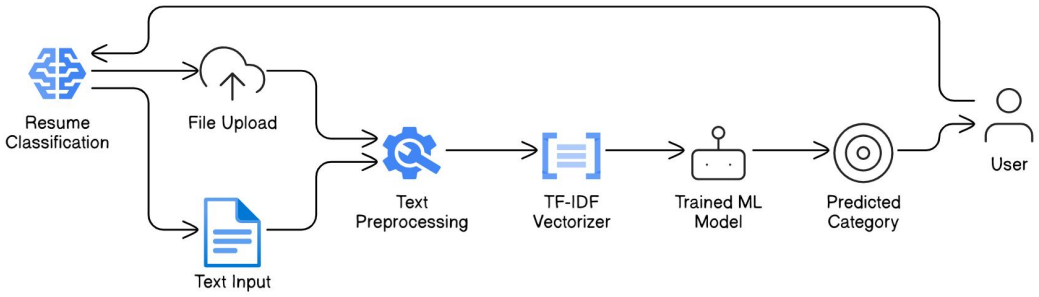

Figure 1: Architecture Globale du Système

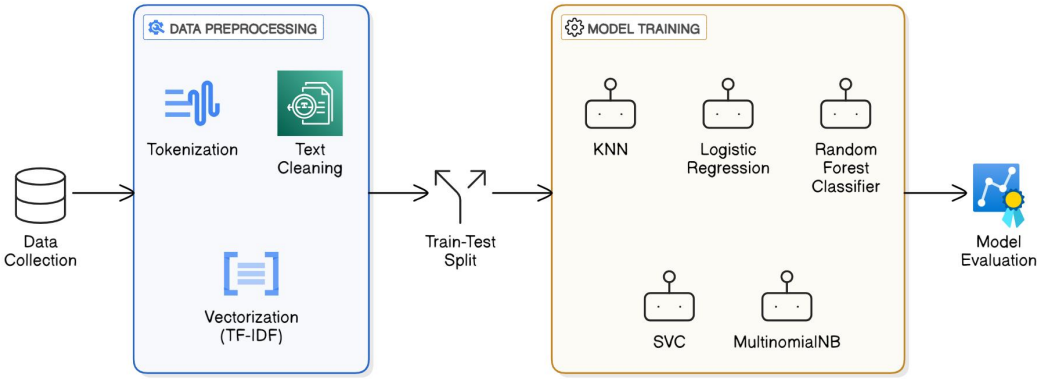

Figure 2: Construction de modèle

2. Architecture du Projet:

La **figure 2** illustre les étapes du processus de classification de texte en utilisant des techniques d'apprentissage automatique. Ce processus commence par la collecte des données textuelles, suivie d'un prétraitement comprenant le nettoyage du texte,la tokenisation et la vectorisation à l'aide de la méthode Term Frequency-Inverse Document Frequency (TF-IDF). Les données ainsi prétraitées sont ensuite divisées en deux ensembles : un ensemble d'entraînement et un ensemble de test. Plusieurs algorithmes d'apprentissage automatique, tels que KNN (K-Nearest Neighbors), la régression logistique, Random Forest Classifier, SVC (Support Vector Classifier) et MultinomialNB (Multinomial Naive Bayes), sont entraînés en utilisant l'ensemble d'entraînement. Enfin, les modèles entraînés sont évalués et comparés en termes de performance en utilisant l'ensemble de test afin de sélectionner le modèle le plus performant pour la tâche de classification.

3. Collecte de données :

Le dataset utilisé dans cette étude provient du dépôt Kaggle "Resumes Images Dataset", qui comprend une collection variée de 2 400 échantillons de CV dans différents formats, tels que *PDF*, *DOCX* et des fichiers image. Ce dataset couvre 43 catégories de métiers distinctes, allant de rôles techniques comme le développement logiciel et la science des données à des postes non techniques tels que les ressources humaines et le marketing. Les CV ont été collectés à partir de plusieurs sources, garantissant une diversité dans la mise en forme, la structure du contenu et les niveaux d'expérience professionnelle. Les images de CV ont été spécifiquement collectées depuis Google, Bing et LiveCareer, et dans le cadre de cette étude, **nous nous sommes focalisés uniquement sur les images provenant de Bing.**

Chaque CV a été étiqueté manuellement en fonction de sa catégorie professionnelle principale, créant ainsi un dataset pour l'apprentissage supervisé. Le dataset présente une répartition relativement équilibrée entre les catégories, avec environ 55 à 60 échantillons par classification de métier. Pour garantir la qualité des données, tous les CV ont été pré-sélectionnés pour leur complétude et leur pertinence, et les informations personnelles identifiables (IPI) ont été correctement anonymisées. Les images, principalement au format *PNG* et *JPEG*, ont constitué le cœur de notre analyse et ont

nécessité des approches spécifiques de prétraitement pour l'extraction de texte.
Ce dataset est open source, permettant une utilisation libre dans des projets de recherche et de développement.

4. Transformation des données:

L'un des principaux défis rencontrés dans cette étude a été de transformer les images de CV en un fichier CSV structuré et exploitable pour le traitement ultérieur. Cette transformation s'est avérée complexe en raison de la qualité variable des images et des divers formats utilisés. Pour surmonter cet obstacle, un processus rigoureux a été mis en place, composé de plusieurs étapes clés décrites ci-dessous :

Validation des Images:

La première étape a consisté à filtrer les images pour s'assurer qu'elles répondent à des critères spécifiques de validité. Parmi les 2 852 fichiers analysés, 2 724 ont été jugés valides et 128 ont été écartés en raison de leur non-conformité. Les critères de validation incluaient une taille minimale de 100x100 pixels, des formats acceptés tels que *JPG*, *PNG* ou *PDF*, et une taille maximale de 10 MB. Ce processus de tri a permis de garantir une base de données cohérente et exploitable pour les étapes ultérieures.

Prétraitement de l'Image:

Afin d'optimiser la qualité des images pour la reconnaissance optique des caractères (OCR), plusieurs techniques de prétraitement ont été appliquées. Les images ont été converties en noir et blanc pour simplifier l'analyse, le bruit visuel a été réduit, et le contraste a été amélioré. Ces étapes étaient essentielles pour maximiser la précision de l'extraction de texte, notamment pour des CV de qualité variable.

Extraction de Texte:

Pour convertir les images en texte lisible, l'outil **OCR Pytesseract** a été utilisé. Cette phase a permis d'extraire le contenu textuel des CV valides. Les textes trop courts ou non significatifs ont ensuite été filtrés, garantissant ainsi la pertinence des données extraites.

Organisation des Données

Les informations extraites ont été organisées sous forme de DataFrame comprenant deux colonnes principales : la catégorie du CV et le texte extrait. Ce format structuré a facilité la création d'un fichier *CSV* prêt à être utilisé pour des analyses de données ou des modèles de machine learning.

Métriques:

Plusieurs métriques ont été calculées pour évaluer l'efficacité du processus :
Nombre total de fichiers traités : 2 852.
Fichiers valides : 2 724.
Fichiers invalides : 128.

Résultat Final:

Le processus de transformation a abouti à la création d'un fichier *CSV* structuré contenant les CV transformés en texte. Ce fichier est prêt à être utilisé dans des analyses avancées de données ou pour l'entraînement de modèles de machine learning, offrant ainsi une base solide pour des études futures.

5. Nettoyage des données:

Étapes Clés de Nettoyage des Données
Un processus structuré de nettoyage des données a été mis en place pour préparer le dataset à l'analyse. Les étapes clés, ainsi que des statistiques détaillées, sont présentées ci-dessous :

Chargement des Données:

Les données ont été chargées à partir du fichier CSV situé dans le répertoire `dataset_resumes.csv`. Le dataset contient deux colonnes principales : Category, représentant la catégorie professionnelle du CV, et Resume, qui contient le texte associé.

Validation du Dataset:

Une analyse initiale a été effectuée pour valider l'intégrité et la structure des données :

- Nombre total de lignes : 2 852.
- Nombre de colonnes : 2 (Category, Resume).
- Nombre total de catégories uniques : 43.

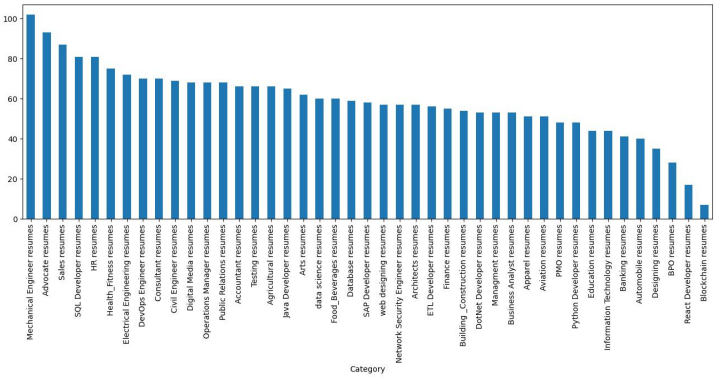

Figure 3: Nombre de CV par catégorie professionnelle

Distribution des Catégories

Les catégories présentent une distribution globale équilibrée, bien que des variations soient observées :
Top 3 des Catégories les plus représentées :

- Data Science resumes : 120 CV.
- Mechanical Engineer resumes : 105 CV.
- Civil Engineer resumes : 89 CV.

Catégories les moins représentées :

- Blockchain resumes : 9 CV.
- Designing resumes : 39 CV.
- Banking resumes : 43 CV.

La majorité des catégories contiennent entre 50 et 90 CV, bien que quelques extrêmes soient notés, ce qui pourrait influencer les résultats lors des analyses ou des modèles de machine learning.

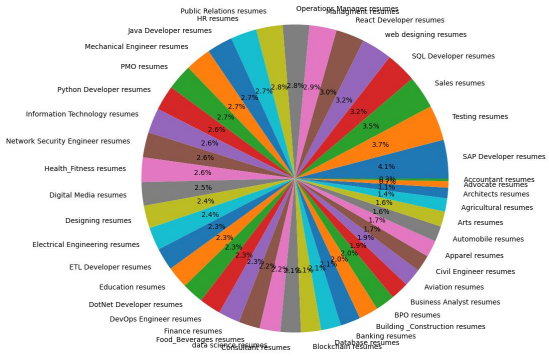

Figure 4: Nombre de CV par catégorie professionnelle

Préparation pour le Nettoyage:

Pour garantir la qualité des données textuelles, des outils et bibliothèques spécifiques ont été utilisés :

- **nltk :** Utilisé pour supprimer les stopwords et effectuer une analyse linguistique.
- **sklearn :** Employé pour les tâches de prétraitement comme la vectorisation des données textuelles.
- **re :** Utilisé pour nettoyer le texte en supprimant les caractères spéciaux et les données non pertinentes.

Résultat Final:

Le dataset nettoyé a été transformé en un fichier prêt pour les analyses avancées grâce à l'utilisation de la technique de TF-IDF (Term Frequency-Inverse Document Frequency) pour la vectorisation des données textuelles. Cette méthode a permis de pondérer les termes les plus significatifs, facilitant ainsi leur utilisation dans des tâches d'apprentissage automatique. Les données sont désormais optimisées pour une exploration approfondie des tendances professionnelles et l'entraînement de modèles prédictifs.

6. Approche de classification des CV :

Après la transformation et le nettoyage des données, l'étape suivante consiste à préparer celles-ci pour l'entraînement des modèles. Ce processus inclut la conversion des données textuelles en représentations numériques et l'utilisation de la méthode de séparation *train-test* pour évaluer les performances du modèle. Voici un résumé de la méthodologie :

Vectorisation TF-IDF:

La vectorisation TF-IDF (Term Frequency-Inverse Document Frequency) est une méthode couramment utilisée pour transformer du texte en données numériques exploitables par les modèles de machine learning. Elle permet de pondérer l'importance des mots dans un document en fonction de leur fréquence dans le document et leur rareté dans l'ensemble des documents.

Formules mathématiques :

TF (Term Frequency) : Représente la fréquence d'apparition d'un mot t dans un document d.

$$TF(t,d) = \frac{\text{Nombre d'occurrences de t dans d}}{\text{Nombre total de mots dans d}}$$

IDF (Inverse Document Frequency) : Mesure la rareté d'un mot t dans l'ensemble des documents D.

$$IDF(t,D) = \log\left(\frac{N}{1 + \text{Nombre de documents contenant t}}\right)$$

où N est le nombre total de documents. Le +1 dans le dénominateur est ajouté pour éviter la division par zéro.

TF-IDF : Combinaison des deux mesures pour obtenir le poids final d'un mot t dans un document d.

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

Application :

Grâce à la vectorisation TF-IDF, les mots fréquemment utilisés dans un document, mais rares dans l'ensemble des documents, obtiennent un poids plus élevé, tandis que les mots très communs (comme les stopwords) reçoivent un poids faible. Cette méthode est essentielle pour transformer les données textuelles en une représentation matricielle adaptée à l'entraînement de modèles de machine learning.

Train-Test Split:

Le train-test split consiste à diviser le dataset en deux sous-ensembles :

- **Ensemble d'entraînement (Train set) :** Utilisé pour entraîner le modèle.

- **Ensemble de test (Test set) :** Utilisé pour évaluer la performance du modèle.

Un ratio de 80/20 ou 70/30 est souvent utilisé, avec 80% des données pour l'entraînement et 20% pour le test.

7. **Algorithmes:**

- ***K-Nearest Neighbors (KNN)***
  - **Définition :**

Le K-Nearest Neighbors (KNN) est un algorithme d'apprentissage supervisé utilisé pour la classification et la régression. Il classe un point de données en fonction des classes de ses **K** voisins les plus proches. La mesure de la distance la plus courante utilisée est la distance euclidienne.

- **Formulation mathématique :**

Distance Euclidienne (pour la classification) :

$$d(x, x_i) = \sqrt{\sum_{j-1}^{n} (x_j - x_{ij})^2}$$

Où x est le point de données à classer et x_i sont les voisins les plus proches.

- **Fonction de coût :**

Le KNN ne possède pas de fonction de coût explicite, car il s'agit d'une méthode non paramétrique qui effectue des prédictions en fonction des voisins sans ajuster de paramètres.

- **Domaine du problème :**

- Classification supervisée (classification de données en fonction des voisins proches).
- Recommandations de produits, systèmes de reconnaissance d'images, etc.

- ***Logistic Regression***
  - **Définition :**

La régression logistique est un modèle statistique utilisé pour prédire la probabilité qu'un événement appartienne à une certaine classe. Il est couramment utilisé pour la classification binaire (par exemple, "oui" ou "non").

- **Formulation mathématique :**

La régression logistique utilise la fonction sigmoïde pour prédire une probabilité p :

$$p(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)}}$$

Où $\beta_0, \beta_1, ..., \beta_n$ sont les coefficients du modèle.

- **Fonction de coût :**

La fonction de coût est la **log-vraisemblance négative** ou **entropie croisée** :

$$\mathcal{L}(\beta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(p^{(i)}) + (1 - y^{(i)}) \log(1 - p^{(i)}) \right]$$

Où $y^{(i)}$ est la vérité terrain, $p^{(i)}$ la probabilité prédite, et m le nombre d'exemples.

- **Domaine du problème :**

- Classification binaire ou multiclasse (par exemple, maladie ou non).
- Analyse de risques, marketing, détection de fraude, etc.

- ***Random Forest:***
  - **Définition :**

Le Random Forest est un ensemble d'arbres de décision utilisés pour la classification et la régression. Chaque arbre est construit en utilisant une sous-partie aléatoire des données et des caractéristiques. L'algorithme fait ensuite une prédiction en agrégeant les prédictions de tous les arbres (par exemple, par vote majoritaire pour la classification).

- **Formulation mathématique :**

Le modèle est une moyenne ou un vote des prédictions de plusieurs arbres $T_1, T_2, ..., T_m$:

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} T_i(x)$$

Où m est le nombre d'arbres dans la forêt et $T_i(x)$ est la prédiction du i-ème arbre.

- **Fonction de coût :**

La fonction de coût pour la classification est généralement l'entropie ou le critère de Gini utilisé dans les arbres de décision.

- **Domaine du problème :**

- Classification et régression.

- Problèmes complexes avec des relations non linéaires (biomédecine, analyse des risques financiers, etc.).

- ***Support Vector Machine (SVC):***
  - **Définition :**

Les machines à vecteurs de support (SVC) sont des modèles d'apprentissage supervisé qui cherchent à séparer les classes avec un hyperplan optimal dans un espace à plus grande dimension. Les SVC sont particulièrement efficaces pour les problèmes de classification binaire.

- **Formulation mathématique :**

L'objectif est de maximiser la marge entre les classes en trouvant un hyperplan $w \cdot x + b = 0$, tout en minimisant l'erreur :

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

sous les contraintes :

$$y_i(w \cdot x_i + b) \geq 1, \quad \forall i$$

Où w est le vecteur normal à l'hyperplan, x_i sont les points d'entraînement, et y_i les étiquettes de classe.

- **Fonction de coût :**

La fonction de coût est une combinaison de la marge et des erreurs de classification, généralement formulée comme une fonction hinge :

$$\mathcal{L}(w,b) = \sum_{i-1}^{m} \max(0, 1 - y_i(w \cdot x_i + b))$$

- **Domaine du problème :**

- Classification binaire avec des marges larges.
- Vision par ordinateur, analyse de texte, détection d'anomalies.

- ***Multinomial Naive Bayes (MultinomialNB):***
  - **Définition** :

Le Naive Bayes multinomial est un classificateur probabiliste basé sur le théorème de Bayes, adapté pour des données de comptage (par exemple, le nombre d'occurrences d'un mot dans un texte). Il est largement utilisé pour la classification de texte.

- **Formulation mathématique :**

L'estimation de la probabilité $P(y \mid X)$ est effectuée selon la formule de Bayes :

$$P(y|X) = \frac{P(y)\prod_{i-1}^{n} P(x_i|y)}{P(X)}$$

Où $P(y)$ est la probabilité a priori de la classe y, et $P(x_i \mid y)$ est la probabilité d'observer x_i dans la classe y.

- **Fonction de coût :**

La fonction de coût est basée sur la log-vraisemblance :

$$\mathcal{L}(\theta) = -\sum_{i-1}^{m} \log P(y_i|X_i)$$

- **Domaine du problème :**

- Classification de texte (par exemple, spam vs. non-spam).
- Analyse de sentiments, filtrage de contenu, reconnaissance de documents.

8. Modèle sélectionné:

L'accuracy (ou précision) est une mesure de performance d'un modèle de classification. Elle indique la proportion de prédictions correctes par rapport au nombre total de prédictions effectuées. En d'autres termes, elle mesure la capacité du modèle à classer correctement les instances par rapport à l'ensemble des instances testées.

Formule mathématique :

$$\text{Accuracy} = \frac{\text{Nombre de predictions correctes}}{\text{Nombre totale de predictions}}$$

Résultats d'Accuracy pour les Modèles :

- **Logistic Regression** : un score de 0.879, ce qui indique que le modèle a correctement classé environ 87.9 % des instances.
- **Random Forest Classifier** : un score de 0.853, signifiant que ce modèle a correctement classé environ 85.3 % des instances.
- **K-Nearest Neighbors (KNN)** : un score de 0.844, ce qui signifie que le modèle a correctement classé environ 84.4 % des instances.

- **Support Vector Classifier (SVC)** : un score de 0.842, ce qui montre que le modèle a correctement classé environ 84.2 % des instances.
- **Multinomial Naive Bayes** : un score de 0.634, ce qui signifie que le modèle a correctement classé environ 63.4 % des instances.

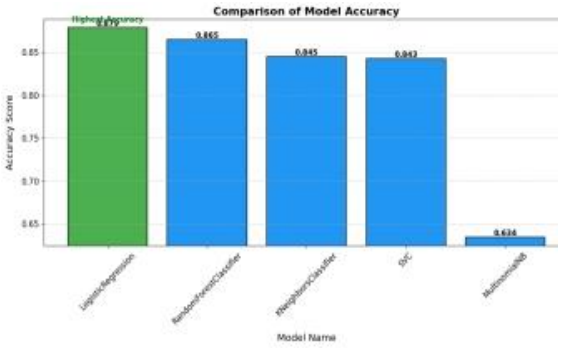

Figure 4: Comparaison de l'Accuracy des Modèles de Classification

Interprétation :

L'analyse des résultats d'accuracy des différents modèles montre que la **régression logistique est le modèle le plus performant avec un score de 0.879**, indiquant qu'il a correctement classé environ 87.9 % des instances. Ce modèle semble donc le mieux adapté à cette tâche de classification. Le Random Forest Classifier (0.853), le K-Nearest Neighbors (KNN) (0.844) et le Support Vector Classifier (SVC) (0.842) ont des performances relativement proches, avec une précision respectivement de 85.3 %, 84.4 % et 84.2 %. Ces modèles, bien qu'efficaces, sont légèrement moins performants que la régression logistique. Enfin, le Multinomial Naive Bayes obtient un score de 0.634, ce qui suggère qu'il n'est pas aussi adapté à ce problème de classification que les autres modèles. Cette différence de performance pourrait être attribuée à des caractéristiques particulières des données, comme leur complexité ou leur structure, qui ne s'adaptent pas bien aux hypothèses sous-jacentes du modèle Naive Bayes.

9. Déploiement de modèles dans le Web:

Le déploiement de modèles dans des environnements web permet de rendre les modèles de machine learning accessibles via une interface web interactive. **Streamlit** est utilisé pour créer rapidement des applications web en Python, sans nécessiter de compétences en développement front-end. Il permet d'intégrer facilement les modèles entraînés avec **Scikit-learn**, une bibliothèque Python dédiée au machine learning. Une fois le modèle entraîné, **Pickle** est utilisé pour le sérialiser et le sauvegarder sous forme de fichier, permettant ainsi de le charger rapidement à chaque interaction avec l'application, sans avoir à le réentraîner. Ce processus permet d'offrir des prédictions en temps réel et une expérience utilisateur fluide.

IV. Limitation:

Les limitations du modèle se déclinent en plusieurs catégories. D'un point de vue des données, des biais d'échantillonnage peuvent survenir si certaines catégories de CV sont surreprésentées ou sous-représentées, impactant la performance du modèle sur ces catégories. De plus, la qualité et la variabilité des données, notamment en raison des différents formats (PDF, DOCX, images), compliquent le processus d'extraction et de prétraitement du texte. Les limitations techniques incluent un prétraitement textuel imparfait et des contraintes liées à la vectorisation TF-IDF, qui peut ne pas capturer parfaitement la complexité du vocabulaire des CV. En outre, certains modèles peuvent rencontrer des difficultés de performance et de généralisation, particulièrement si les données sont déséquilibrées ou si les nouvelles catégories de CV ne sont pas bien représentées. Enfin, les limitations contextuelles et linguistiques incluent la dépendance à la langue anglaise, ce qui restreint l'applicabilité du modèle pour des CV rédigés dans d'autres langues. Le modèle peut également avoir du mal à s'adapter aux évolutions rapides des métiers et à saisir les nuances professionnelles, ce qui affecte son efficacité dans des secteurs spécifiques.

V. Conclusion :

En conclusion, ce projet de classification des CVs a permis de démontrer l'efficacité des modèles de machine learning dans l'automatisation de l'analyse de documents professionnels. Cependant, plusieurs limitations, liées à la qualité des données, aux contraintes techniques et à l'adaptabilité aux contextes spécifiques, ont été identifiées.

Malgré les performances satisfaisantes des modèles, des améliorations peuvent être apportées, notamment dans le prétraitement des données et la gestion des évolutions des métiers. En surmontant ces défis, le modèle pourrait devenir un outil puissant et fiable pour l'analyse de CV, avec des applications pratiques dans des domaines tels que le recrutement automatisé et la gestion des talents.

References :

Git Hub `noran-mohamed/Resume-Classification-Dataset`[Link]

Liens:

Git Hub pour le reprtoire du projet.
link pour le site web d'application.