**Analyse et Extraction d'Informations des CV (Resumes Parsing )**

**Introduction**

Dans cette section, l'objectif principal était de développer des outils et des techniques pour extraire automatiquement des informations structurées à partir de CV au format PDF. Cette étape est cruciale pour automatiser le processus de traitement des candidatures et faciliter l'analyse des compétences des candidats par les recruteurs.

1. Méthodologie de Parsing des CV

Pour atteindre cet objectif, j'ai mis en place plusieurs fonctions Python spécialisées. Chaque fonction a été conçue pour traiter une catégorie spécifique d'informations à extraire :

* 1. Extraction du Texte d'un PDF :

La première étape consistait à extraire le texte brut à partir des fichiers PDF des CV. Cette fonction est essentielle car elle permet de convertir le contenu non structuré des CV en données textuelles manipulables.

* 1. Extraction des Numéros de Téléphone :

À l'aide d'une expression régulière adaptée, j'ai développé une méthode pour détecter et extraire les numéros de téléphone mentionnés dans les CV. Cela inclut la gestion de divers formats tels que les codes régionaux, les séparateurs et les indications internationales.

1.3 Extraction de l'Adresse Email :

Une autre fonction utilise une expression régulière pour identifier et extraire les adresses email des CV. Cette méthode garantit la récupération précise des informations de contact essentielles des candidats.

* 1. Extraction des Compétences :

En utilisant une liste prédéfinie de compétences chargée à partir d'un fichier CSV , j'ai développé un algorithme pour extraire les compétences spécifiques mentionnées dans les CV. L'approche inclut la sensibilité à la casse pour maximiser la précision de l'extraction.

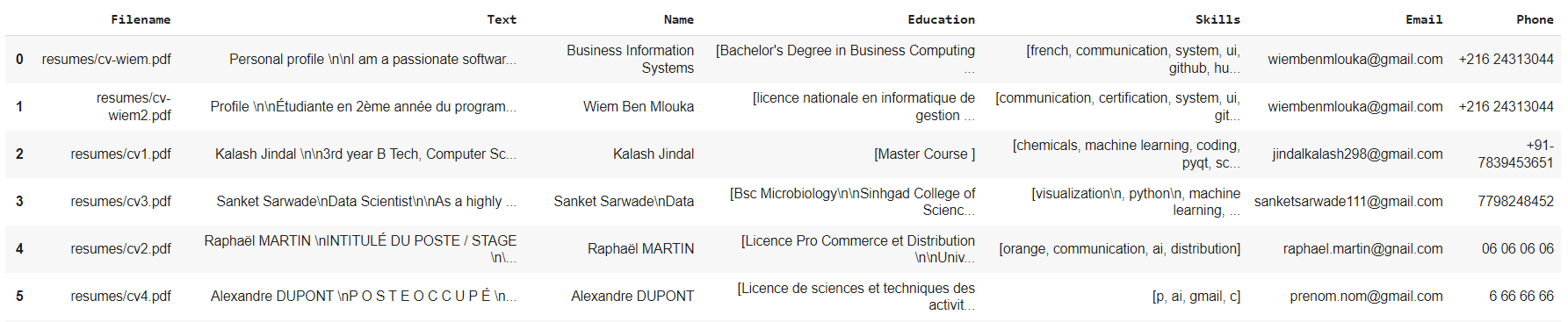
* 1. Extraction des Informations d'Éducation :

J'ai également mis en place une fonction pour extraire les informations sur l'éducation des candidats, en identifiant des termes clés tels que les diplômes universitaires et les niveaux d'éducation mentionnés dans les CV.

* 1. Extraction du Nom du Candidat :

Une fonction spécifique a été développée pour extraire le nom du candidat à partir du texte des CV, en utilisant des motifs spécifiques pour capturer les noms complets et les formats variés.

1. Traitement Batch des CV (Batch Processing)

****Les techniques mises en œuvre ont été évaluées à travers un ensemble diversifié de CV. Les résultats ont démontré une haute précision dans l'extraction des coordonnées, des compétences et des informations sur l'éducation. En utilisant un traitement par lot sur un ensemble de CV choisi aléatoirement, j’ai réussi à extraire de manière automatisée des informations cruciales. Cette approche a non seulement accéléré le processus d'analyse mais a aussi réduit les erreurs potentielles associées à une extraction manuelle.

La figure ci-dessus illustre la structuration des données extraites pour faciliter une analyse ultérieure. Chaque CV est représenté dans un tableau de données (DataFrame), où chaque ligne correspond à un CV individuel avec des colonnes spécifiques telles que le nom du candidat, ses compétences, son éducation, etc. Cette organisation permet une comparaison facile et la recherche de critères spécifiques par les recruteurs.

1. Recommandation de CV

Pour recommander des CV pertinents basés sur des critères spécifiques tels que les compétences, nous avons utilisé la similarité cosinus calculée à l'aide de TfidfVectorizer de scikit-learn.

* 1. Vectorisation des Compétences

Les compétences extraites de chaque CV ont été vectorisées en utilisant TfidfVectorizer. Cela a permis de représenter chaque CV comme un vecteur numérique basé sur la fréquence des mots-clés importants.

* 1. Fonction de Recommandation de CV

La fonction recommend\_resumes commence par prendre un ou plusieurs critères de recherche comme input. Ces critères peuvent être des compétences spécifiques recherchées par un recruteur, telles que "communication", "machine learning", ou toute autre compétence pertinente pour le poste.

**Étape 1:** Vectorisation des Critères de Recherche

Avant de calculer la similarité cosinus, les critères de recherche sont d'abord transformés en vecteurs TF-IDF à l'aide du même TfidfVectorizer utilisé pour vectoriser les compétences extraites des CV. Cela permet de représenter les critères de manière numérique, en tenant compte de leur importance relative dans le contexte des compétences extraites des CV.

**Étape 2:** Calcul de la Similarité Cosinus

Une fois les critères de recherche vectorisés, la fonction calcule la similarité cosinus entre ce vecteur de critères et tous les vecteurs TF-IDF des CV stockés dans tfidf\_matrix. La similarité cosinus mesure la similitude entre deux vecteurs en calculant le cosinus de l'angle entre eux dans l'espace vectoriel.

**Étape 3:** Classement des CV par Similarité

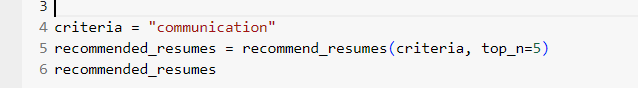
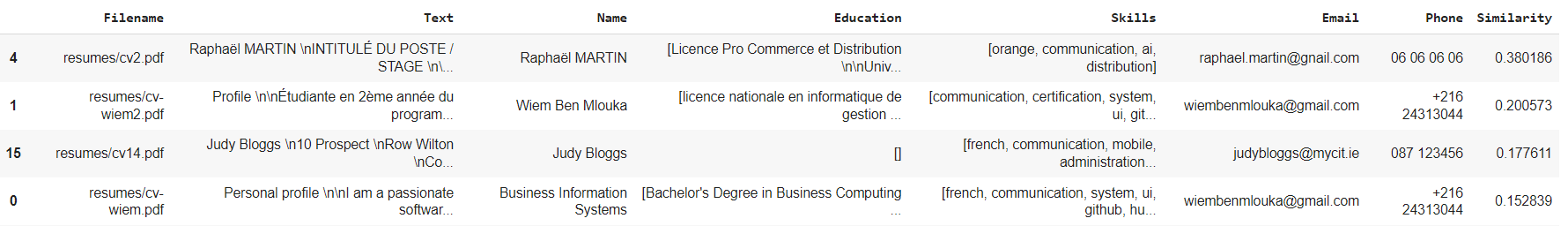
Après avoir calculé la similarité cosinus pour chaque CV, la fonction filtre les CV ayant une similarité positive (supérieure à 0) pour exclure les CV qui ne correspondent pas du tout aux critères de recherche.

Enfin, les CV sont triés par ordre décroissant de similarité cosinus pour recommander les CV les plus pertinents en fonction des critères de recherche spécifiés.

* 1. Exemple d'utilisation de la fonction de recommandation

L'exemple utilisé dans la figure 3 illustre bien l'efficacité de l'approche basée sur la similarité cosinus pour recommander des CV en fonction d'un critère de recherche spécifique, en l'occurrence "communication".

Les CV recommandés sont triés selon leur similarité cosinus par rapport au critère "communication". Cela signifie que les CV en tête de liste sont ceux qui présentent le contenu le plus proche du critère recherché. Par exemple :

****Raphaël MARTIN (cv2.pdf) : Ce CV est recommandé en premier avec une similarité de 0.380186. Il met en avant des compétences telles que "communication" et "distribution", qui correspondent bien au critère spécifié.

Cette approche permet aux recruteurs de trouver rapidement et efficacement les CV qui correspondent le mieux aux compétences spécifiques recherchées. En calculant la similarité cosinus, la fonction recommend\_resumes automatise le processus de tri et de recommandation des CV, facilitant ainsi la sélection des candidats les plus appropriés pour les postes disponibles.

**Conclusion**

La capacité à extraire de manière fiable et précise ces informations essentielles des CV est cruciale pour la phase suivante du projet, qui vise à recommander les CV les plus pertinents aux recruteurs. Ces données structurées alimentent les modèles d'intelligence artificielle et de deep learning, permettant une analyse avancée pour une prise de décision plus informée et efficace dans le processus de recrutement.

**Modèles de Machine Learning pour la Classification des CVs**

**Introduction**

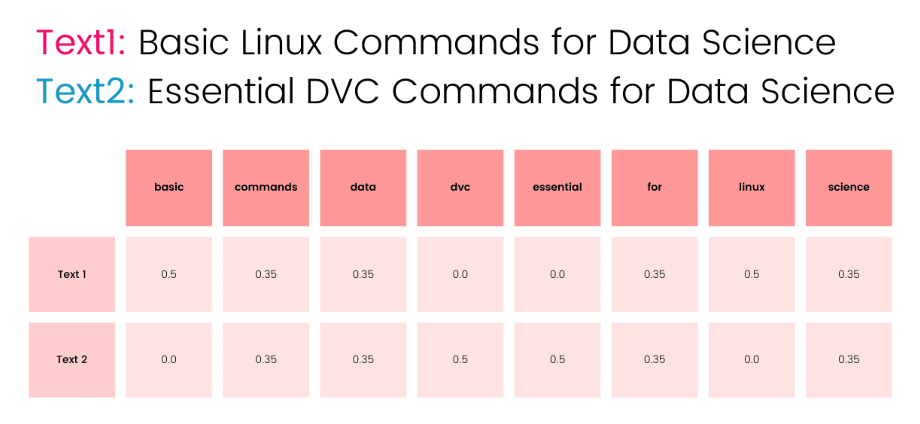
L'objectif de cette section est d'explorer divers modèles de machine learning pour classifier les CV en différentes catégories basées sur leur contenu. En utilisant des techniques de prétraitement de texte et en appliquant différents algorithmes de machine learning, je vise à identifier le modèle le plus efficace pour une classification précise des CV. La section couvre l'extraction des caractéristiques, l'entraînement des modèles, l'évaluation et la prédiction.

1. Encodage des Labels

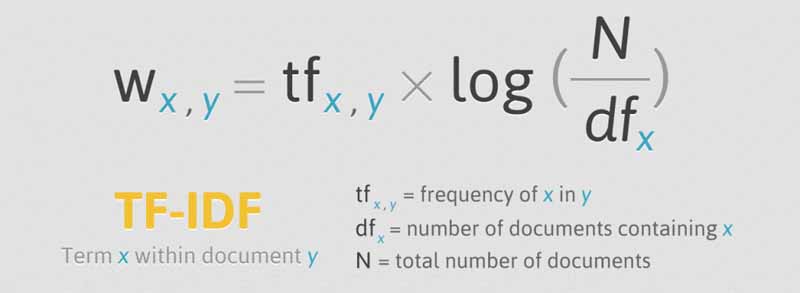
Les catégories des CVs sont encodées en valeurs numériques. Cela permet aux algorithmes de machine learning de traiter les catégories comme des valeurs discrètes.

 L'encodage des labels convertit les catégories textuelles en entiers, rendant les données compatibles avec les modèles de classification .(figure x)

1. Vectorisation des Textes

Nous utilisons la méthode TF-IDF (Term Frequency-Inverse Document Frequency) pour transformer les textes nettoyés en vecteurs de caractéristiques. TF-IDF aide à identifier les mots importants dans chaque document, en tenant compte de leur fréquence dans l'ensemble du corpus. La vectorisation convertit les textes en matrices numériques, prêtes à être utilisées par les algorithmes de machine learning.

**Figure :** Convertion des documents texte en une matrice TF-IDF avec TfidfVectorizer

TF-IDF (Term Frequency-Inverse Document Frequency) est une méthode utilisée pour évaluer l'importance d'un mot dans un document par rapport à un corpus de documents. La formule du TF-IDF est representé par la figure sur dessous :

Avec :

TF(x,y) (Term Frequency) : la fréquence du terme x dans le document y.

N : nombre totalede documents

DF(x) : nombre de documents qui contiennent x

1. Division des Données

Les données sont divisées en ensembles d'entraînement (training set :80%) et de test (test set : 20%). Cette division permet d'évaluer la performance des modèles sur des données qu'ils n'ont pas vues lors de l'entraînement.

En réservant une partie des données pour les tests, nous pouvons obtenir une estimation plus précise de la capacité de généralisation des modèles.

1. Définition et Entraînement des Modèles

plusieurs modèles de classification ont été testé incluant Naive Bayes, KNeighborsClassifier, Logistic Regression, RandomForestClassifier, SVC, et DecisionTreeClassifier. Chaque modèle est ensuite entraîné sur l'ensemble d'entraînement.

L'entraînement des modèles consiste à ajuster leurs paramètres internes pour minimiser l'erreur de prédiction sur l'ensemble d'entraînement.

1. Définition de la mesure de performance : Accuracy

L'accuracy (exactitude en français) est une mesure de performance couramment utilisée pour évaluer les modèles de classification. Elle représente le pourcentage de prédictions correctes faites par un modèle par rapport à l'ensemble de données de test.

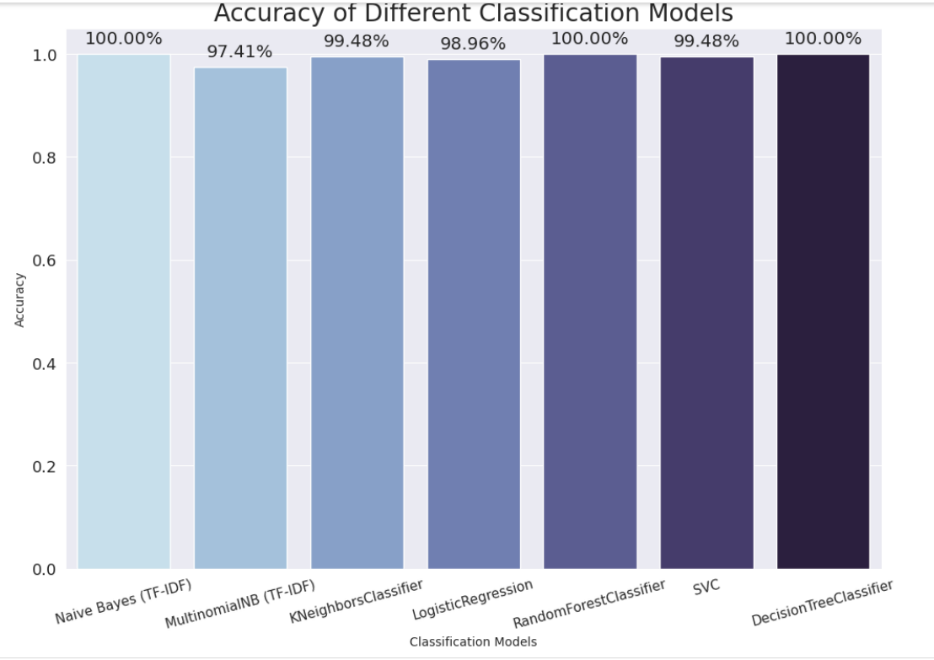
Mathématiquement, l'accuracy est calculée comme le ratio du nombre de prédictions correctes sur le nombre total de prédictions :

**Accuracy =** Nombre de predictions correctes / Nombre total de predictions

1. Visualisation des Résultats

Nous traçons un graphique pour visualiser les précisions des différents modèles. Cette visualisation aide à comparer les performances et à identifier le modèle le plus performant.

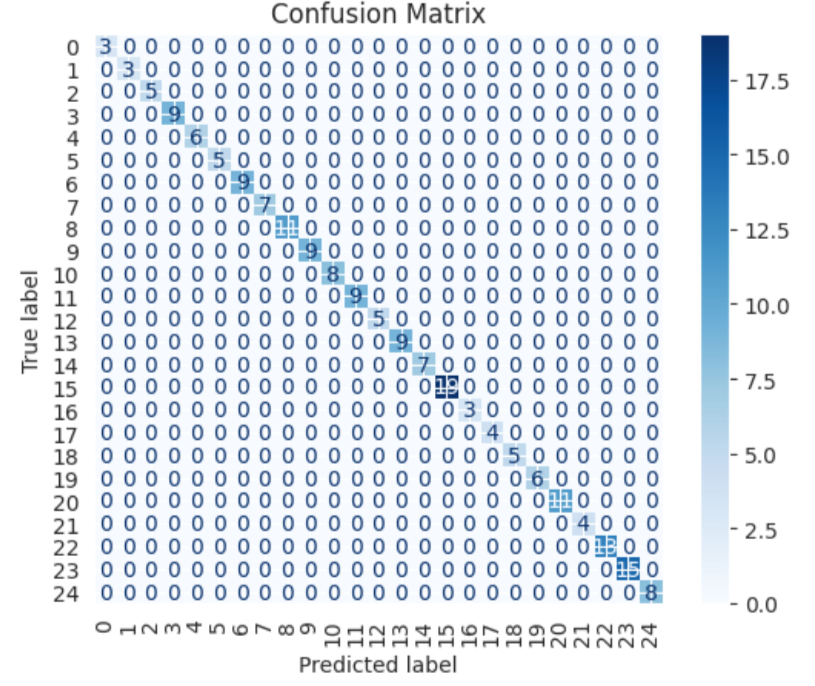
La figure suivante montre un graphique en barres représentant l'exactitude (accuracy) de différents modèles de classification



Ces résultats indiquent que les modèles Naive Bayes (TF-IDF), RandomForestClassifier, et DecisionTreeClassifier ont atteint une exactitude de 100%, tandis que MultinomialNB (TF-IDF) a obtenu l'exactitude la plus basse à 97.41%. Les autres modèles se situent entre ces deux extrêmes, avec KNeighborsClassifier et SVC à 99.48%, et LogisticRegression à 98.96%.

1. Prédiction et Évaluation

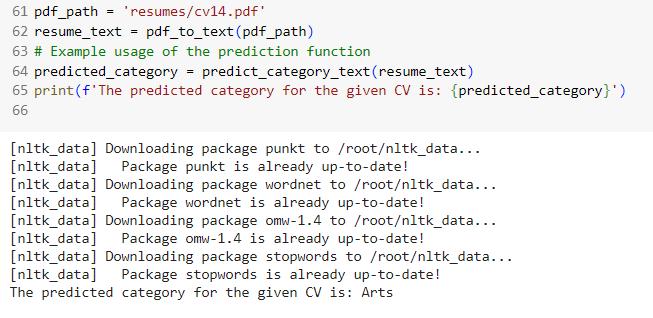
Nous utilisons les modèles entraînés pour prédire les catégories des CVs et visualiser les résultats à l'aide de matrices de confusion. Cela permet de vérifier les performances des modèles sur des données de test et d'identifier les erreurs de classification. Les matrices de confusion fournissent une vue détaillée des performances des modèles en montrant le nombre de prédictions correctes et incorrectes pour chaque catégorie.



Pour les modèles ayant une précision de 100%, il est nécessaire de comparer les prédictions des catégories sur un ensemble de CV qui n'appartient ni à l'ensemble d'entraînement ni à l'ensemble de test. Cela permettra de vérifier la capacité de généralisation des modèles sur des données inédites et d'éviter le surapprentissage. Les résultats de cette comparaison orienteront ou préciseront le choix des modèles que j'adopterai pour mon système.

(a compléter : justification du modele choisit)

1. Exemple d’utilisation

Imaginons que nous ayons un nouveau CV et que nous souhaitions prédire sa catégorie à l'aide du modèle Naive Bayes. Nous procédons comme suit :

Pour interpréter les résultats obtenus pour le CV numéro 14 (cv14.pdf), qui a été prédit comme appartenant à la catégorie "Arts" par le modèle Naive Bayes (TF-IDF), cela signifie que selon les caractéristiques extraites de ce CV et comparées aux modèles précédemment entraînés, le modèle a estimé que ce CV correspond le mieux à cette catégorie spécifique parmi toutes les catégories disponibles dans le jeu de données. Cette prédiction est basée sur la similarité des termes et des structures présents dans le CV avec ceux des autres CVs dans la catégorie "Arts".

Ce processus montre comment utiliser un modèle de classification pour prédire la catégorie d'un nouveau CV, démontrant l'application pratique des modèles entraînés.

**Conclusion**

Cette section démontre comment différents modèles de machine learning peuvent être utilisés pour classifier automatiquement les CVs en catégories. Cette approche améliore l'efficacité et la précision du processus de recrutement, en permettant un tri initial automatisé des CVs. Les modèles les plus précis peuvent être utilisés en production pour automatiser le tri initial des CVs, réduisant ainsi la charge de travail des recruteurs et augmentant la rapidité du processus de sélection des candidats.