**Reformulation**

# **Présentation**

L’idée de ce projet est de retrouver le travail d’une personne à partir d’un extrait de son CV, et d'afficher les résultats que nous avons de manière graphique et visuelle.

# **Ressources**

Pour cela, nous avons à notre disposition un fichier JSON contenant les métadonnées suivantes :

* L’ID d’une personne, qui permettra de l’identifier (Id)
* L’extrait du CV associé à cette personne (description)
* Le genre de la personne (homme ou femme), avec “M” pour un homme et “F” pour une femme (gender)

Un second fichier CSV nous permettra d’associer le nom du travail avec un ID.

Enfin, un fichier CSV de résultats, qui nous permettra d’effectuer une comparaison finale entre les résultats que nous obtenons et ceux que nous sommes censés obtenir.

# 

# **Déroulé**

Ce projet devra être réalisé en **6 étapes** :

* La première étape consiste à déposer toutes les données sources sur HDFS sur une VM Hadoop en local. En parallèle, il est nécessaire de faire une première observation des données pour avoir une idée de ce que nous allons exploiter. Pour cela, nous utiliserons des outils de visualisation que nous avons vu en cours, tels que Tableau Software ou D3.js, pour faire cette première analyse.
* Dans un second temps, il faut récupérer ces données en local à partir de HDFS à l’aide d’un script.
* En troisième étape, il faut effectuer un chiffrement des données, avant de les envoyer sur une VM dans le cloud AWS. Il sera nécessaire de faire attention à la sécurité des données, pour leur envoi ainsi que pour leur stockage.
* La quatrième étape consiste à mettre en place un algorithme de machine learning pour nos données, afin de déterminer de la manière la plus exacte possible (résultats indépendants du genre de l’individu) à quel travail est associé une personne. Une première problématique pourrait se poser dans le cas où un métier n’a pas assez d’individus associés, ce qui ne permettrait pas d’avoir un modèle cohérent dans le cas de ce poste. Cet algorithme de machine learning utilisera une partie des données à notre disposition, l’autre partie nous servira lors de l’étape suivante.
* La cinquième étape nous mènera à utiliser une partie des données du fichier data.json à notre disposition, et qui n’a pas été utilisée pour la phase d’apprentissage de l’algorithme de machine learning, pour créer un fichier predict.csv. Une fois ce fichier créé, nous ferons tourner sur notre VM AWS le modèle mis en place à l’étape précédente afin de prédire les postes de chacune des personnes de ce fichier. Les résultats seront concaténés avec le fichier predict.csv et sauvegardés au format CSV sur le File System de la VM AWS.
* La dernière étape permettra de récupérer et de charger à l’aide d’un script le fichier de résultats sur une base MongoDB qui sera sur une VM ou en local. Enfin, nous utiliserons D3.js afin d’afficher nos résultats de manière graphique et visuelle.

**Première analyse des données**

Nous disposons de trois fichiers :

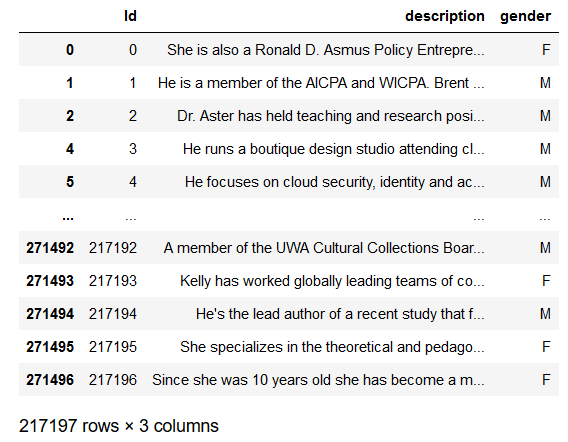
* ***categories\_string.csv***

Il s’agit du référentiel des métiers. Chaque ligne correspondra à une classe unique lors du processus de classification.



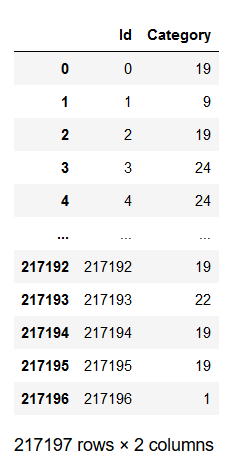
* ***data.json***

Le fichier de données à exploiter. Volumineux, il contient pour chaque ligne un ID, un sexe et une description de CV.



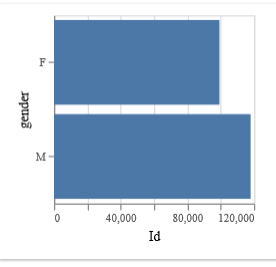
* ***label.csv***

Ce fichier est un résultat type. Cette référence nous permettra d’ajuster les modèles que nous allons utiliser afin d’obtenir un taux de confiance maximal pour nos classifications.



**Donnée particulière : *gender***

Le but étant d’obtenir des classifications visant les métiers, la colonne d’information “gender” semble superflue.



Cependant, une étude plus approfondie sur les données pourrait déterminer les corps de métier les plus fréquemment rencontrés, cette donnée va rester en retrait jusqu’à la finalisation du projet principal.

L’étape suivante à prendre en compte est le formatage des données. Les données de description étant des phrases avec des références, nous allons effectuer plusieurs opérations afin de rendre les modèles fonctionnels.



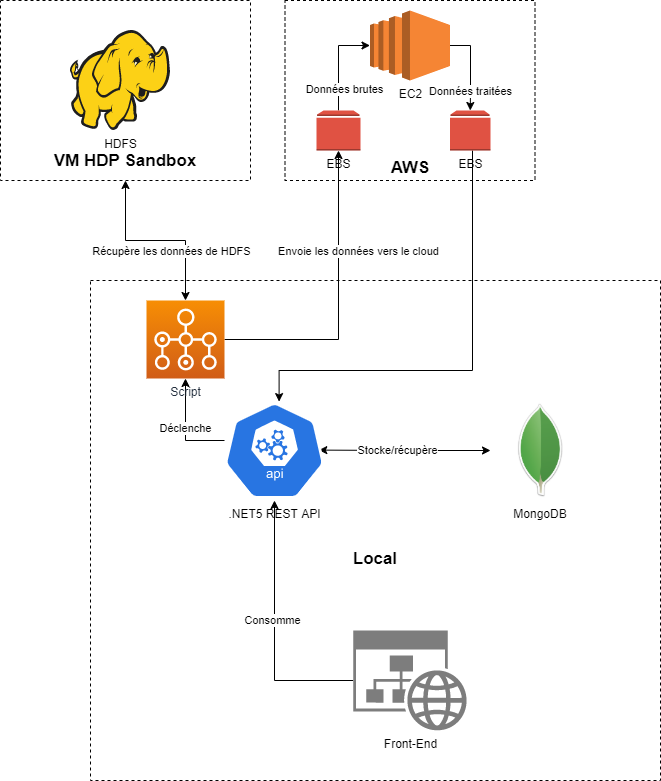
En plus d’enlever le bruit présent au sein des phrases et de les mettre en minuscule, nous allons tester deux solutions : le stemming et la lemmatisation.

Ces solutions combinées à la tokenisation vont révéler les stopwords (dans ce cas, des mots parasites comme “she”, “he”, “is”), nous permettant de les enlever afin d’effectuer une classification sur les mots, une vérification via un test “Bag of words” sera effectuée par la suite pour assurer qu’aucun stopword n’a été manqué.

Enfin, plusieurs modèles seront testés afin d’observer celui s’approchant le plus du résultat attendu. Des modèles SVM (SGDClassifier par exemple) ou de régression logistique (LogisticRegression ou tree par exemple) seront étudiés.

# 

# **Diagramme d’architecture**



A partir de notre compréhension du sujet, nous avons pu proposer une première architecture technique. Celle-ci a été conçue afin de permettre de proposer à l’utilisateur une interface unifiée via un portail web. L’utilisateur pourra donc à partir du portail web :

* Modifier la configuration des scripts (dossier à récupérer sur hdfs, IP de l’instance EC2...)
* Lancer l’exécution des scripts de transfert de données
* Lancer la récupération des données depuis le cloud vers la base de donnée
* Consulter les analyses de données via un dashboard

Ce choix est fait dans l’ambition de fournir une solution tout en offrant une expérience optimale pour l’utilisateur. Ce qui correspond à l’attente d’un livrable final ressemblant à un produit.

**Gestion du temps et livrables**

Afin d’atteindre les objectifs du projet et tenir les délais, nous avons mis en place un outil de suivi sous la forme d’un trello. Nous avons une liste par deadline :

* Rapport de reformulation du 25/01/2021
* Soutenance intermédiaire du 02/02/2021
* Soutenance finale du 20/05/2021
* Livraison finale du 23/05/2021

Sous ces listes, nous avons des cartes correspondant aux différentes activités à avoir réalisé d’ici là. Afin de ne pas glisser, nous mettons des dates limites à ces activités dans un ordre chronologique défini par les besoins du projet. Nous prévoyons aussi une pré-deadline qui détermine quand les travaux doivent être finalisés, ce qui nous permet d’inclure une marge en cas de dépassement.