**Manuel d’exploitation**

1. **Mise en place du transfert des données de HDFS vers le cloud**

*Pré-requis :*

Afin de garantir l’automatisation du transfert, le système doit être configuré de telle manière que le transfert ssh sans demande de mot de passe soit possible entre la machine local et la VM HDP 3.0. Sinon, le système demandera à l’utilisateur de rentrer son mot de passe lors de le la connexion à la machine virtuelle. Pour mettre en place l’automatisation on suivra ce [tutoriel de référence](https://alvinalexander.com/linux-unix/how-use-scp-without-password-backups-copy/)

Après avoir suivi le tutoriel vous devrez avoir sous le répertoire home de l’utilisateur un dossier .ssh contenant un fichier authorized\_keys contenant la clé publique de la machine locale. Côté local on aura la clé privée dans le dossier .ssh du répertoire home de l’utilisateur.

Sur la machine virtuelle on crée aussi un répertoire data sous le répertoire home qui contiendra les données récupérées de hdfs.

Côté instance ec-2, on se connecte avec l’utilisateur ec2-user et la clé ssh, ce qui est la solution préconisée par AWS. On créera un répertoire data sous le répertoire home de l’utilisateur afin de stocker les données provenant de hdfs.

Côté hdfs. On crée un répertoire rawdata et un répertoire data sous /user/<username> pour lesquels l’utilisateur choisi pour le lancement du script aura les droits en lecture et en écriture.

Le début de la chaine de traitement se passe sous /user/<username>/rawdata où les données sont stockées en vrac.

Le transfert va déplacer et partitionner par timestamp technique ces données, afin de ne s’assurer qu’une même donnée ne soit pas traitée deux fois.

Le transfert des données vers le cloud se lance avec le script /Scripts/ScpTransfer.sh

Ce script prend deux paramètres : le DNS de l’instance ec2, ainsi que l’utilisateur de la VM cloudera avec lequel on souhaite se connecter.

Si tout a bien été configuré alors le script se déroule sans problèmes. On peut mettre en place une tache CRON pour qu’il s’exécute de manière régulière et parfaitement automatique. Le script assure que les données ne soient pas traitées deux fois.

1. **Mise en place de l’API et Mongodb**

La première étape est de mettre en place la base de donnée Mongo, [la documentation pour ce faire est disponible ici](https://docs.mongodb.com/guides/server/install/)

Ensuite il faut mettre en place l’environnement d’exécution de l’API. Pour ce projet, comme nous restons en local, on reste dans une config de dev. La manière la plus simple est de télécharger Visual Studio 2019 (L’édition community est gratuite) avec le module de développement ASP .Net Core. La version du framework utilisée est .Net 5.

On ouvre la solution (fichier .sln), et on se concentre sur le projet API.

Dans le fichier appsettings.json, renseigner les configs suivantes :

* BucketName : Nom du bucket dans lequel sont stocké les données prédites par le système de Machine Learning
* MongoConnectionString : Chaîne de connexion à la base Mongo, par défaut on garde la valeur "mongodb://localhost:27017"
* MongoDataBaseName : Nom de la base de donnée Mongo à renseigner avec le nom de la base donnée par l’utilisateur à la création

Configurer l’API comme projet de démarage et lancer le projet.

1. **Modèle de prédiction**

Les 2 fichiers .pickle présents dans le dossier Machine\_Learning contiennent des vecteurs de modèles déjà entraînés pour TF-IDF, pour vectoriser nos mots, ou celui de la descente de gradient que nous utilisons pour notre prédiction.

Le fichier “Main.py” est le fichier Python qui s’occupe de lire les dossiers et les fichiers JSON qui sont poussés par l’utilisateur via l’API sur EC2 et de les envoyer dans notre modèle de prédiction, puis une fois la prédiction effectuée, d’envoyer la prédiction sur un bucket S3. Tout le traitement qui est effectué sur le texte, avec les fonctions utilisées, se trouve sur le fichier “Functions.py”, avec le prétraitement des données et bien sûr la prédiction avec la vectorisation de TF-IDF et le modèle en descente de gradient.

Toutes les informations concernant l’algorithme de prédiction sont détaillées sur le document *Machine\_Learning.docx*

1. **Mise en place du site web**

Le contenu du site web est situé dans le répertoire ***“www”***. Le contenu de ce dossier est un projet NodeJS. Il est donc requis pour son bon fonctionnement de disposer de NodeJS pour installer les dépendances de ce projet. Vous pouvez trouver une version d’installation adaptée à votre machine ici : <https://nodejs.org/en/download/> . La version 14.15.4 de NodeJS a été utilisée pour initialiser ce projet.

Une fois le répertoire cloné, se placer dans le répertoire ***“www”*** dans une invite de commandes et taper la commande ***“npm install”***, cette commande installera les dépendances associées au projet (les dépendances utilisées sont consultables dans le fichier *package.json*).



Certaines dépendances chargées par le fichier *index.html* sont appelées par lien hypertexte. Si une connexion internet n’est pas disponible dans l’environnement cible, ces fichiers de dépendances peuvent être téléchargés via ces liens et placés localement. Ce procédé nécessite le changement des liens hypertexte par le chemin de ces fichiers.

Des exemples de fichiers de dépendances comme *d3.js* sont disponibles en guise d’exemple dans le répertoire ***src***.