

**Rapport de projet du module**

**Big Data & DataViz**

**Membres du projet :**

* BENMECHICHE Khaled
* AMEZIANE Liticia
* HANY Mohamed Amine

2022-2023

**Introduction:**

Le Big Data est composé de jeux de données complexes, provenant essentiellement de nouvelles sources. Ces ensembles de données sont si volumineux qu’un logiciel de traitement de données traditionnel ne peut tout simplement pas les gérer. Mais ces énormes volumes de données peuvent être utilisés pour résoudre des problèmes que vous n’auriez jamais pu résoudre auparavant.

Dans ce petit rapport nous allons résumer l’ensemble de taches effectué pour la réalisation de notre projet, qui consiste à créer un pipeline Big Data (c.à.d un **pipeline** de données englobe une série d'actions qui débute avec l'ingestion de l'ensemble des données brutes issues de n'importe quelle source, pour les transformer rapidement en données prêtes à être exploitées.), qui sera donc capable de réaliser des traitements automatiques de données à partir de divers sources de données. Les données seront représenter par des visuelles grâce à des fonctions de visualisations divers tout cela dans le but d’aboutir à une connaissance ou bien une information utile et aussi d’améliorer, d’optimiser ou bien de proposer des solutions à certaines problématiques remarquer lors de la visualisation et l’analyse des données.

**Feuille de route de notre travail :**

-**En premier**, nous disposant de trois principales sources de données sur lesquelles on a travaillé et fait nos traitements ainsi que nos visualisations, tous ayant en commun la géolocalisation (Paris et région Ile de France), ces données sont **de format CSV**, ayant des informations sur (La pollution, la circulation et les accidents) sur quatre années 2018, 2019, 2020 et 2021, qu’on a télécharger via les liens https://www.data.gouv.fr/fr/ et https://opendata.paris.fr/pages/home/

-SPECIFICATIONS DES FICHIERS DE DONNĖES :

* Les fichiers pollution contiennent des colonnes représentant l’id pollution, l’année, le mois, la qualité d’air et le niveau de pollution (1 pour bonne, 2 pour moyenne, 3 pour mauvaise, 4 pour dégradée et 5 pour extrêmement mauvaise).
* Les fichiers circulation contenant des colonnes représentant l’id circulation, l’année, le mois, Indice fréquentation tous types de transport, Vitesse moyenne, Volume trafic BP, Vitesse moyenne BP, FREQ\_VELIB, FREQ\_METRO, FREQ\_RERA\_RERB, FREQ\_BUS\_PARIS.
* Les fichiers accident : contenants les informations de colonnes représenter comme suit :
* La rubrique LIEUX  
  Num\_Acc  
  Identifiant de l’accident identique à celui du fichier "rubrique CARACTERISTIQUES" repris dans l’accident.  
  catr  
  Catégorie de route :  
  1 – Autoroute  
  2 – Route nationale  
  3 – Route Départementale  
  4 – Voie Communales  
  5 – Hors réseau public  
  6 – Parc de stationnement ouvert à la circulation publique  
  7 – Routes de métropole urbaine  
  9 – autre  
  voie  
  Numéro de la route.  
  V1  
  Indice numérique du numéro de route (exemple : 2 bis, 3 ter etc.).
* V2  
  Lettre indice alphanumérique de la route.  
  circ  
  Régime de circulation :  
  -1 – Non renseigné  
  1 – A sens unique  
  2 – Bidirectionnelle  
  3 – A chaussées séparées  
  4 – Avec voies d’affectation variable  
  nbv  
  Nombre total de voies de circulation.  
  vosp  
  Signale l’existence d’une voie réservée, indépendamment du fait que l’accident ait lieu ou non sur  
  cette voie.  
  -1 – Non renseigné  
  0 – Sans objet  
  1 – Piste cyclable  
  2 – Bande cyclable  
  3 – Voie réservée  
  prof  
  Profil en long décrit la déclivité de la route à l'endroit de l'accident :  
  -1 – Non renseigné  
  1 – Plat  
  2 – Pente  
  3 – Sommet de côte  
  4 – Bas de côte  
  pr  
  Numéro du PR de rattachement (numéro de la borne amont). La valeur -1 signifie que le PR n’est pas  
  renseigné.  
  pr1  
  Distance en mètres au PR (par rapport à la borne amont). La valeur -1 signifie que le PR n’est pas  
  renseigné.  
  plan  
  Tracé en plan :  
  -1 – Non renseigné  
  1 – Partie rectiligne  
  2 – En courbe à gauche  
  3 – En courbe à droite  
  4 – En « S »  
  lartpc  
  Largeur du terre-plein central (TPC) s'il existe (en m).  
  larrout  
  Largeur de la chaussée affectée à la circulation des véhicules ne sont pas compris les bandes d'arrêt  
  d'urgence, les TPC et les places de stationnement (en m).surf  
  Etat de la surface :  
  -1 – Non renseigné  
  1 – Normale  
  2 – Mouillée  
  3 – Flaques  
  4 – Inondée  
  5 – Enneigée  
  6 – Boue  
  7 – Verglacée  
  8 – Corps gras – huile  
  9 – Autre  
  infra  
  Aménagement - Infrastructure :  
  -1 – Non renseigné  
  0 – Aucun  
  1 – Souterrain - tunnel  
  2 – Pont - autopont  
  3 – Bretelle d’échangeur ou de raccordement  
  4 – Voie ferrée  
  5 – Carrefour aménagé  
  6 – Zone piétonne  
  7 – Zone de péage  
  8 – Chantier  
  9 – Autres  
  situ  
  Situation de l’accident :  
  -1 – Non renseigné  
  0 – Aucun  
  1 – Sur chaussée  
  2 – Sur bande d’arrêt d’urgence  
  3 – Sur accotement  
  4 – Sur trottoir  
  5 – Sur piste cyclable  
  6 – Sur autre voie spéciale  
  8 – Autres  
  vma  
  Vitesse maximale autorisée sur le lieu et au moment de l’accident.

-**En second**, on s’est consacré au moyen d’envoyer ces données brutes dans notre pipeline, qui a été effectuer en batch avec le HDFS, donc on a créé un container dans le docker qui met en relation Hadoop et Spark, puis on la lancer à l’aide de la commande **docker-compose up –d** et puis la commande docker exec –it namenode bash qui sert à nous connecter au namenode de notre cluster, enfin après avoir lancer notre container on a copier nos données CSV dans le namenode a l’aide de la commande **cp** et dans des répertoires HDFS avec la commande **hdfs dfs –put**.

-**Ensuite**, on a commencé à effectuer nos différents traitements pyspark dans le jupyter notebook. On a pu voir des traitements comme la création de dataframes à partir des fichiers CSV pollution, circulation et accident, le calcul de la moyenne du max, de l’ajoute et la sélection de colonnes dans les différents dataframes , le calcul des statistiques de ces dataframes, et la création de fichiers globaux avec les différentes colonnes de chaque dataframe,…ect

-**Après**, avoir terminé nos différents traitements sur les fichiers CSV, on s’est lancé sur la visualisation de nos données, du coup on a convertie nos dataframes pyspark en dataframes pandas à l’aide de la fonction **ToPandas() ,** puis on a importé quelques librairies ([Matplotlib](https://matplotlib.org/" \t "_blank)et [Seaborn](https://seaborn.pydata.org/)) qui nous ont permis d’écrire nos bout de codes de visualisation.

-**Puis**, une fois nos données traitées on a mis en connection notre jupyter notebook avec Mongodb qui est l'une des bases de données NoSql les plus connues, en suivant ces quelques étapes :

* Installation MongoDB sur notre machine, on a démarrez le serveur MongoDB en entrant la commande ci-dessous à l'invite de commande : mongod
* Dans le shell mongo on a lancé la commande mongo.
* Puis un show dbs pour voir les BDDs existantes, use madata pour crée une base de données, createCollection() pour la création d’un collection dans la base de donnée.
* Puis on a utilisé le module python PyMongo pour connecter le notebook Jupyter à l'hôte local MongoDB. On écrivant le code !pip installer pymongo dans la cellule NoteBook.
* Ensuite on a importez le module PyMongo avec : importer pymongo

de pymongo importer MongoClient.

* Enfin on a connectez la base de données MongoDB à l'aide de la ligne de code : client= MongoClient ("localhost ", 27017) pour faire quelque traitements sur le shell mongo.

**-Enfin,** pour réaliser notre interface on a utilisé la techno **Power BI** qui transforme nos sources de données en informations visuelles immersives et interactives.