Quelque soit les données sur lesquelles nous travaillons, il est nécéssaire de faire un prétraitement(après collecte) afin que ces données puisse être utilisable par d'autres prgrammes(par exemple pour entrainer des modèles IA).
Ainsi le chapitre 2 nous parle de quelques techniques les plus utilisées pour faire un prétraitement des données.
Nous précision que les données qui ont été utilisées dans ce cas sont des données tabulaires mais même avec les types de données les images, audios, etc, nous aurons un processus de prétraitement.
Par contre nous pouvons souligner que le processus prétraitement des données dépend du programme qui va utiliser les données par la suite ou du type de données avec lesquelles nous travaillons.
Par exemple en IA pour utiliser la technique de **transfer learning** faudra adapter nos données avec le type de données du modèle de base à utiliser.
De même que si on travaille avec des images la partie prétraitement seraient peut-être de redimensionner les images, supprimer les images floues, faire de l'augmentation, etc et des fonctions différentes de celles utilisées dans ce livre seront utilisées.

Comme dans le premier livre on a eu à faire des implémentations sur l'environnement cloud **Databricks,** cette fois-ci nous avons utilisé l'outil de contenerisation **Docker** pour faire nos implémentations sur Jupyter notebook.

Pour le premier lancement de Docker, il n'y avait pas une image de PySpark et elle a été buildée et chargée par la suite.

Tout d'abord nous avons commencé par créer un **Schema Spark** pour définir la structure du DataFrame qui peut être obtenu en appelant la fonction **printSchema**().
La création de ce Shecma se fait grâce au type de donnée **StructType** qui est une liste de champ représentant une ligne. Il définit principalement le nom de la colonne, son type et un boolean pour spécifier si la colonne peut avoir une valeur null ou pas.
Ensuite nous avons crée notre DataFrame grâce à la fonction **createDataFrame** spark en utlisant notre schema.

Ainsi nous avons commencé par l'utlisation des fonctions les plus basique c'est à dire remplacer ou supprimer les données manquantes(avec une

valeur **null)** grâce aux fonctions **fillna** et **na.drop()** comme le montre notre notebook de la cellule **1** à la cellule **18.**

L'application des fonction de selection pemettant d'obtenir un sous-ensemble de notre jeu de données a été faite également dans le notebook. Faut noter que nous avons un jeu de données différent de celui utilisé dans le livre ce qui a fait les opérations ont été appliquées sur des colonnes différentes.
Par exemple nous avons pu faire une opération de filtre sur les numéro téléphone ay le chiffre **1** comme indicateur(cellule 50)

D'autres fonctions ont été utilisées aussi comme les aggrégation, collect, etc mais également l'utilisation des fonctions UDF qui sont définies par l'utilisateur et qui peuvent être utilisées sur chaque ligne de notre DataFrame(normal Python UDF) ou sur des blocks(Pandas UDF).
Donc nous pouvons dire que dans le monde du Big Data où nous aurons affaire avec de larges volumes de données, il serait préférable d'utiliser les Pandas UDF pour une meilleure performance.