Étape n°1 : Zone d'Aterrisage On lit le fichier et à chaque Une fois la ligne On passe le Tweet dans la 'ZoneAterrisage(chaine)' ligne on en extrait le transformé en Entrée : Nom du fichier fonction dictionnaire contenu dans dictionnaire on extrait le Json ('str') 'SuprCaracterSpe(chaine)' celle ci avec 'loads()' Tweet de celui-ci Étape n°2 : DataFrame Principal On fusionne les DataFra On lit le fichier et à chaque Une fois la ligne transformé en 'RecupDonneesP(chaine)' représentant chaque ligne ligne on en extrait le Entrée : Nom du fichier Json dictionnaire on Transforme le former notre DataFrmae pi dictionnaire contenu dans dictionnaire en DataFrame ("DFP") dans lequel on va ('str') celle ci avec 'loads()' en local la base de donr Étape n°3 : Opérations d'Analyse Méthode: Récupération des Tweets du DataFrame (pincipal) → Analyse → DataFrame (d'analyse) Les Données d'Analyse seront liées aux données du DataFrame principal par leur index (le numéro de leur ligne tout simplement) On utilise en suite le principe On sépare le tweet en une liste de d'expression régulière pour détecter 'Hashtags(chaine)' chaînes de caractères pour Enfin on renvoie la l le début et la fin du Hashtag et on Entrée : Un Tweet ('str') différencier plus aisément les mots des Hashtags du Tv l'ajoute à la liste des Hashtags du des Hashtags tweet Une fois la liste des Hashtags du Tweet 'DicoHashtags(DF)' On parcourt le DataFrame et récupéré on la parcourt et on la compare Entrée : DataFrame contenant les Tweets de la on applique la fonction avec le dictionnaire des hashtags qui base de donnée (on sélectionne la colonne 'Hashtags()' à chaque tweet contient en clé les Hashtags et en valeur "TweetText" du DataFrame principal) leur nombre d'apparition.

Étape n°4 : Complément de la base de donnée (donc du DataFrame principal 'DFP')

La même **méthode** d'analyse est en suite utilisée pour analyser les @, les

sentiments et les topics.

L'enjeux est ici de compléter notre base de données incomplète. En effet notre base de données comporte pour chaque tweet un champ ID mais aucun "Pseudo" ou "Nom d'Utilisateur". C'est pourquoi nous allons aiouter à

On obtient le Tweet sans caractère spéciaux, on l'écrit donc dans le fichier Json 'Zone d'aterrissage'

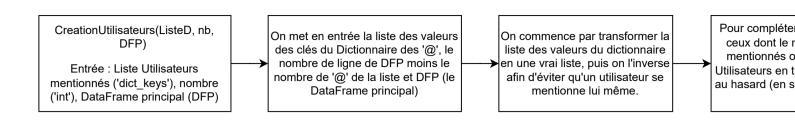
ame pour incipal stocker iées

iste /eet

> Si le Hashtag n'est pas dans le dictionnaire on l'ajoute. Sinon on ajoute 1 à sa valeur (qui modélise le compteur du nombre de mention du Hashtag)

notre DataFrame principal ce nouveau champ "Utilisateurs".

La création d'Utilisateurs est essentielle au bon déroulement de certaines opérations d'Analyses, notamment pour 'Les utilisateurs mentionnés par un utilisateur spécifique'. Il est donc impératif que tous les utilisateurs mentionnés dans les tweets soient présent au sein de la base de données, donc dans le champ "Utilisateurs" du DataFrame DFP.



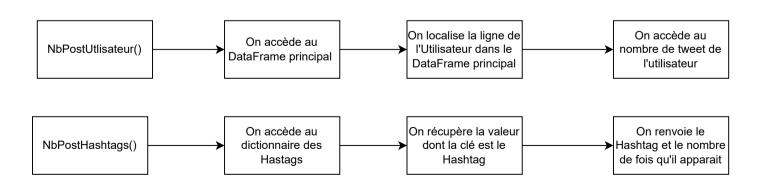
Étape n°5 : Stockage des données d'Analyse

Par soucis de rapidité d'exécution nous avons opter pour la méthode de parcourir le DataFrame contenant tous les tweets et d'effectuer au fur et à mesure les opérations d'analyses sur ceux-ci pour en même temps les réunir dans un même dictionnaire et faciliter la création d'un DataFrame réunissant toutes les analyses.



Étape n°6 : Fonctionnalités de Inpoda

Maintenant que l'on dispose de toutes les analyses de la base de données stockées dans des structures adaptées, on peut se préoccuper des fonctionnalités de Inpoda



r les Utilisateurs en plus de nom est connu car ils sont n crée de toute pièce des irant des lettres et un chiffre 'appuyant sur les intervalles d'Unicode).

A chaque utilisateur créé on ajoute son nom précédé d'un '@' dans la liste des Utilisateurs. Puis on crée un DataFrame contenant comme nom de colonne "Utilisateurs", comme valeur pour chaque ligne le nom d'un Utilisateur.

Enfin on fusionne ce DataFrame d'utlisateurs d'une seule colonne avec le DataFrame principal ('DFP')