wrangle_report

September 12, 2022

0.1 Reporting: wragle_report

- Create a **300-600 word written report** called "wrangle_report.pdf" or "wrangle_report.html" that briefly describes your wrangling efforts. This is to be framed as an internal document.

Dans cette étape de data wrangling, nous avons mis en place tous les étapes du processus à savoir la collecte des données, l'évaluation des données et leurs nettoyage .

Pour la collecte des données nous avons utilisé trois méthodes de collecte à savoir l'importation d'un fichier plat à partir du répertoire du travail (csv, tsv...), l'utilisation de la librairie Python Requests pour le téléchargement d'une deuxième base et la base requise à partir de l'API twitter.

L'étape suivante est l'évaluation des données. Pour ceci on a utilisé le ń visual assessment ż et le ń programmatic assessement ż .Le ń visual assessment ż nous a permis de se familiariser avec les bases de données et d'identifier les différentes colonnes dedans. On a aussi pu comparé les tailles des différentes bases et savoir que certaines sont plus petites que d'autres ce qui implique que peut etre on n'a pas le meme niveau d'information pour tout l'échantillon d'étude.On a aussi remarqué que certaines colonnes présentent des valeurs manquantes ń NaN ż.On a aussi identifié un probléme d'ordre visuellement qui est le fait que les 4 dernieres colonnes de la base archive (doggo,floofer,pupper et puppo) forment tous une seule variable qui renseigne sur la phase du chien et donc ils doivent etre une seule variable On a par la suite procédé au ń programatic assessement ż pour investiguer davantage les problémes de qualité et d'ordre de nos bases.On a utilisé les commandes de programatic assessment tel que df.head pour voir le format de la base (df.sample également), df.info pour investiguer les types des différentes variabes et le nombre des éventuelles valeurs manquantes, df.columns pour voir les libellés des colonnes et si ces libellés sont significatifs,df.describe qui renseigne sur les caractéristiques statistiques des variables numériques ce qui aide a détecter s'il y'a des éventuelles valeurs abbérantes(la commande df.value_counts permet aussi de détecter des valeurs abbérantes surtout pour les variables catégoriques). On a pu identifier plusieurs problémes à nettoyer des les trois datasets à partir desquels on a choisi 8 problèmes de qualité et 2 problème d'ordre à corriger qui sont :

Problémes de qualité : 1.Timestamp pour le dataset archive est un string alors que ça devrait etre une date.

2.tweet_id dans le dataset archive est de type entier alors que les identifiants devrait être des chaines de caractères.

3.expanded_urls pour le dataset archive contient des valeurs manquantes qu'on a decidé d'éliminer car leur nombre n'était pas important par rapport à la taille de la base des données.

4.retweeted_status_timestamp n'est pas de type date

5.Le dénominateur dans le dataset archive est parfois différent de 10 donc on a ramené toutes les valeurs des dénominateurs à 10

6.Tweet Id est un identifiant qui doit etre un string pour le dataset predict_images

7.id est un identifiant et doit etre un string pour le dataset tweet_json

8.in_reply_to_user_id doit etre un string

Problèmes d'ordre 1.Les quatres dernières colonnes de archive doivent être une seule colonne.

2.L'identifiant du tweet est nommé id dans le tableau tweet_json alors qu'il est nommé tweet_id dans les autres datasets

Après avoir collecter et évaluer les bases des données, on a procédé au nettoyage sur 3 étapes (define,code et test).La plupart des corrections de type s'est fait par la commande df.astype. Pour le premier problème d'ordre cité ci-dessus, on a utlisé la commande pd.melt. L'inconvenient est que lorsqu'on a éliminé les observations aves des valeurs nulles pour la variable ńphase ż créé, la taille de la base des données résultante est devenue restreinte par rapport à celle de départ mais c'était une étape nécessaire car parmi les questions auxquelles on cherchait des réponses, c'était le nombre de favoris par phase.