wrangle_report

September 11, 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.

To gather all three pieces of data for this project and load them i used 3 different techniques of gathering in the notebook.the methods required to gather each data are different , the first method is to simply to download the csv file (we rate dogs twtter archive) , upload it in the notebook then read the csv file using pandas dataframe method read_csv.

The second method is more advanced , i downloaded programaticly from a website the tsv file using the requests library and i wrote the content in a file named '"image_predictions.tsv".

The third method was tricky , i needed to query the twitter API for each tweet's JSON data using python's Tweepy . but i also needed to get access to api keys which i could not do, instead i just dowloaded the "tweet_json.txt" file provided by udacity and extracted the data that i needed for my project.

In the assessement phase, i structred my assesing into visual and programmatic assessement , i began by taking a look of the three datasets using sample which generate random rows , i tried to find patterns between columns and table , after that i began looking and scrolling at rows values . this helped me to notice few qualtity and tidiness issues which i noted . Afterwards, i started programming assesement by using code espacially some lines of code that can help me look deeper in each table, methods like : info, value_counts, describe, with the help of code i noticed many quality issues, some of these are missing values, invalid values or inaccurate values. i made sure that i wrote down every single issue that i found to later address them in cleaning.

.In the cleaning phase , i began by dealing with missing values and retweets that we do not need them for our project , i made sure that each step was clear by defining the issue , address it then test it's no longer an issue. after dealing with incomplete data, i addressed two tidiness issues that break the rule of "each variable forms a column" , and "each type of observational unit forms a table". then i dealt with several invalid and inaccurate quality issues like extracting strings from text, changing datatype, renaming columns, etc... then finally i merged the two dataframes that i had to form one beautiful clean table ready to be stored and analysed.

In [ ]: