Wrangle Report

In this report I will describe briefly the steps I did in order to build my analysis from gathering, assessing, and finally cleaning it properly.

Gathering data

There are three types of data in this project:

1- twitter-archive-enhanced which was in my local computer and in csv format.

For me this is the easiest one to gather data from, it only requires one line of code and pandas function to read it.

2- image-predictions which was on the internet and in tsv format, so for this file I did two steps to gather it.

- Downloading it using requests library, using the url given in the classroom I could send a request to the server and receive the response for that url, then after that I wrote its content into a file and store it in my local computer.
- After downloading it and having the file on my computer now I can read it using the same pandas function read_csv, but with an argument sep that defines the kind of separators used in the file.

Hint: I did not use the sep argument for reading the csv file as it is the default value for read_csv function.

3- api this part is the hardest way and also it is divided into two steps.

- First one is using tweepy library for gathering data using tweeter api and storing it in a json file.
- Secondly reading this json file into a dataframe and for this I created an empty list, read the json file line by line and converting it into a dictionary, appending that dictionary to the empty list, and finally I got a list of dictionaries which I could easily convert it into a dataframe.

OF Course after each step I checked if the code succeeded or not by different ways.

For example using head method to see the data frame

Check if the response succeeded or not from the number 200.

See the names of files stored in the same directory to see if the file was stored or failed.

Assessing Data

For this part I try to find anything wrong with data whether using visual assessment or programmatic and take notes for anything I recognize.

I will describe what I got for each data frame

1- archive_df

a. visual assessment

- Source column data is not written well to be able to read it properly (Quality)

- There are many None value in classifiactions columns. (Quality)

- Also there are four columns for the classifiactions.(Tidiness)

b. programmatic

- timestamp column is object type not datetime -→ using info method(Quality)

- ratings for both numerator and denominator have values that do not specify the description -→ using .value_counts and describe(Quality)

- name column had weird names and None -→ using value_counts(Quality)

- dogs classifications better to be of categorical type instead of strings.(Quality)

2- image_predictions_df

a. visual assessment

- I can see that there are some images that are not dogs, so I will have to exclude them from my analysis.

b. programmatic

- Also sometimes the same image some predictions defines it as dogs and some not and so I need to filter them also -→ using value_counts() (Quality)

- Dog types in p1, p2, and p3 better to be of categorical type instead of strings → .info() (Quality)

3- api_df

a. visual assessment

- id and id_str have the same importance (Tidiness)

b. programmatic

- created_at is if string type instead of datetime →.info() (Quality)

- many columns have no importance into our analysis and need to be dropped

Cleaning

First of all before doing any cleaning process I save a copy of each data frame into a new one so that whenever I want to see the original data or do anything like start the whole project from the beginning I can do it.

After having a copy of the each data frame I started by filtering my data frame so that I end having only the data and variables that I will need in my analysis.

1- archive_df_clean

- Dropping any retweets or replies as we are only interested in original tweets in our analysis.
- Dropping any unnecessary columns that I will not need it in my analysis.

2-image_predictions_df

- Dropping any images that none of them were dogs based on p1_dog, p2_dog, and p3_dog columns.
- Filtering for the highest conf of predictions and taking only rows that the highest values of it is recognized as dogs.

3-api_df_clean

- Choosing only the most interesting columns used in my analysis

Finally for choosing only the common tweets between all the three data frames so that all the filters is applied to all of them and store it into a new data frame called combined_df.

Tidiness

1- For classification columns:
1- Replace None values with empty strings.
2-Concate the four columns into one new column called stage with adding "-
" between two stages if they exist.
3-Create a function to fix the name.
4-I named the empty string at the last Not Classified to use in plots if I needed it.

Hint: I did not use melt as I saw this way is easier and will take the same effort from me as melt and may also less.

Hint: I did another tidiness work like:
1- summarize the three output of the predictions into the most important one that I will use
2- Also at the end I divided the combined_df, but I did it in the steps where I saw it was more suitable as there are many steps in cleaning that combine multi issues and can be solved together in the same step.

Quality

1- Fixing the source as I wanted to use it to see the device or method the highest percentage of clients use.

- I used regex to extract the content of the link tag, then replaced the whole tag with it.

2-Ratings do not satisfies the project description

- Extracted the text, rating_numerator, and rating_denominator that do not satisfies the criteria which for me the denominator equals 10 and the numerator between 6 and 15
- Opened the csv file in excel as I saw it is easier to look to the text there and tried to figure out the problem caused that problem what I found is most of them were due to that the rating I given for a group of dogs not individual one, so I assumed that all of them have denominator of 10 and calculated the number of group to use it to calculate the numerator. There were also some other issues like whole wrong number, but they were very small.
- After making sure that now all the denominators are equal 10 I saw if there are any other rating numerator that have numbers do not satisfies the criteria "not in range" and investigate each one separately in the same way as previous.

3-Converting created_at columns into datetime format to be used when needed and also to see the most time clients interact or tweets.

4-Changing source to categorical

5-Changing p1, p2, p3, type , and stage into categorical instead of strings.

6-Fixing weird names by first finding the weird names then for each group of them build a new pattern and use regex to exclude them

Hint: name column will not be needed in my analysis, so it does not have any importance and I could drop it if I want.