Data wrangling report

By Fares Ahmed ELsayed

Data Gathering:

This is the first step in the project and here we are collecting the data from multiple sources so we can first clean then do our assessments.

There were 3 data files needed and each one was acquired by a different way and here is the summary:

1| ‘twitter-archive-enhanced.csv’ this file was downloaded manually then read by pandas as a df 2| 'image-predictions.tsv' however this file was downloaded with code and loaded to pandas making sure the separator is a tab ‘ \t ’
3| 'tweet_json.txt' and this one was extracted with twitter’s API programmatically too.

Data Assessing:

Here we were doing visual assessing and programmatic assessing to check for quality and tidiness issues without caring much about how they will be cleaned.

Checked for duplicates and NaN values as well as column names and if the data types are correct.

And these were the issues found:

Quality

archive_df

1| change time stamp to datetime format.

2| change tweet id to str.

3| Remove retweets, replies and tweets with no image.

4| dog names are some times "a", "an", "the" and "None"

5| wrong numerators

6| denominator sometimes != 10

7| the type of dog (doggo puppy etc) should be categoral

8| removing unneeded columns like (in_reply_to_status_id, in_reply_to_user_id etc)

9| expanded_urls sometimes have duplicated urls

10| ratings should be float both deno and numer

images_df

1| change tweet id to str.

2| confidence level 2 and 3 is so low that it can be dropped (75 percent of level 2 is bellow 0.2)

3| remove retweets and replies

4| rename confidence columns to more expressive names.

api_df_c

1| change id to tweet_id.

2| change tweet id to str.

3| removing unneeded columns like (in_reply_to_status_id, in_reply_to_user_id etc).

Tidiness

archive_df

1| doggo, floofer, pupper, puppo are variables represented as columns

2| sometimes more than one dog is there and hence 2 dog stages are there

images_df

1| merge this df with archive_df

api_df_c

1| merge this df with archive_df

Data Cleaning:

Here is where the magic happens, we start off with making copies of each df to make sure we have a restore point incase we messed up something.

Some cleaning was done manually if it is a one off however most of the issues were abundant in all of the records so programmatic cleaning isn't only more accurate but it is also much much faster.

Each issue is carried on its own in 3 stages: Define, Code then Test.

We first define what is the quality or tidiness issue then we clean it with code be it manually cell by cell or programmatically, and lastly, we test our df to check if the changes took place in the intended way.

Then I felt like there should only be 2 tables: one for the
image prediction and the other for the tweet info

But later I found out I needed a master df and that's why I created one.

The above steps aren't so linear as it might appear since most of the time while I was cleaning a certain issue, I would find another one that needs to be assessed so iteration is key here.

Lastly, the files were saved as clean data while maintain the old files still incase they were needed.