Wrangling Effort

By: Mahmoud Elshahawy

January 2021

This report demonstrates the data wrangling steps performed on the data obtained from the twitter account "WeRateDogs" as a requirement for the second assignment in the "Data Analysis Professional" Nanodegree.

Data Gathering

In this step, tweets data are collected from three main sources namely:

1. The file "twitter-archive-enhanced.csv" is downloaded to my local PC then uploaded to the directory that contains the jupyter notebook "wrangle_act.ipynb" and finally read to a dataframe using the pandas function "pd.read_csv".
2. The second file "image-predictions.tsv" is a tab-separated values file downloaded from the URL https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv using the "requests.get" function then it is read to a dataframe using the pandas function "pd.read_csv".
3. As I do not have a twitter developer account, the third file "tweet-json.txt" is downloaded and read line by line using the "json.loads" function and saved into a list which in turn is converted to a daraframe using the pandas "pd.DataFrame" function.

Data Assessment and Cleaning

In this step, the three dataframes are evaluated visually and programmatically for quality and tidiness issues:

- Visual assessment is performed via the spreadsheets application :Excel.
- Programmatic assessment is performed using jupyter notebook.

- Some issues are required by the rubric such as excluding retweets, replies and tweets without images.
- Other issues are suggested to facilitate the analysis and visualization process.
- Before cleaning, a copy of each dataframe is created and all cleaning steps are carried out on the copies.

Quality Issues

Archive dataframe

1. Removing the entries containing missing values in expanded_urls because those are tweets without photos.
2. Replacing the string 'None' in doggo, floofer, pupper and puppo columns with empty record " ".
3. Replacing the empty record " " with NaN in dog_stage (the new column that resulted from combining doggo, floofer, pupper and puppo columns).
4. Adding hyphen to 'doggopupper', 'doggofloofer' and 'doggopuppo'.
5. Selecting and removing tweets without photos based on the tweet_id in the image predictions dataframe.
6. Dropping the retweets and replies and removing their corresponding columns.
7. Removing invalid names ('a' & 'an') from the column 'name' and extracting the correct names from the tweets text if applicable then replacing 'None' with 'Nan'.
8. Dividing the numerator by the dogs count for the rows whose rating_numerator is greater than or equal to 40.
9. For the rows whose denominator is less than or greater than 10, replacing their numerator and denominator with the correct values from the tweet text.
10. Converting the type of the tweet_id column from integer into string.
11. Converting the type of the timestamp column from string into datetime.

Image predictions dataframe

1. Dropping the retweets and replies after doing the same step for the archive dataframe.
2. Converting the type of the tweet_id column from integer into string.

3. Changing the type of img_num and prediction_level columns to string.

Tidiness Issues

Archive dataframe

1. Concatenating the 4 columns of doggo, floofer, pupper and puppo into one column called 'dog_stage' and removing the old columns.

Image predictions dataframe

1. Column headers p1, p2, p3 are values, not variable names so, this issue is properly addressed using pd.wide_to_long function after renaming these columns.

API dataframe

1. The API dataframe is not considered an independent observational unit so, it is merged with the archive dataframe based on the tweet ID.

Output

Two dataframes:

1. archive_api_clean_df (1971 rows and 11 columns) saved as twitter_archive_master.csv
2. image_predictions_clean_df (5913 rows and 7 columns) saved as twitter_image_predictions.csv