WeRateDogs Twitter Archive - Wrangle Report

Designed By: Ahmed Hashish
Submitted To: Udacity Data Analyst Nanodegree
Date: 27 February, 2021

Introduction:

Data wrangling is a core skill that everyone who works with data should be familiar with since so much of the world's data isn't clean.

In this project, we have used tweepy library to query Twitter's API for data included in the WeRateDogs Twitter archive. This data includes retweet count and favorite count. We have developed some code to create an API object that has been used to gather Twitter data. After querying each tweet ID, we have writed its JSON data to a tweet_json.txt file with each tweet's JSON data on its own line. Then we have read this file, line by line, to create a pandas DataFrame. After that we develop consecutive steps to assess and clean such data.

Key Points:

We had some Key points to keep in mind with wrangling this data:

- We only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
- In this project, it is required to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset although assessing and cleaning the entire dataset completely would require a lot of time.
- Cleaning includes merging individual pieces of data according to the rules of tidy data.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.
- We do not need to gather the tweets beyond August 1st, 2017. We can, but note that We won't be able to gather the image predictions for these tweets since we don't have access to the algorithm used.

Data Gathering:

We have used three different sources of data:

- **WeRateDogs has downloaded their Twitter archive and sent it to Udacity via email exclusively to be used in this project. This archive is downloaded manually from the Udacity server as** `twitter-archive-enhanced.csv`**. This file is provided as a starting point contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017.**
- `image_predictions.tsv` **file contains a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images). In which, every image was run in the WeRateDogs Twitter archive through a neural network that can classify breeds of dogs. That neural network is created previously as one of the projects in Udacity. This file will be downloaded programmatically using the Requests library.**
- `tweet_json.txt` **file was streamed using tweepy API. This file contains the** `favourite_count` **and** `retweet_count` **along with** `tweet_id` **and more.**

Finally, all three files are loaded into dataframes named respectively df_enhanced, df_pred, and df_updated.

Data Assessment and Clean:

We started with exploring our data to identify and fix several quality and tidiness issues and we found the following:

1- **Tidiness Issue**
In `df_enhanced` table, `text` column contains two variables tweet `text` and tweet `url` therefore, We have splitted them.

2- **Quality Issue**
In `df_enhanced` table, when a row refers to a reply tweet case the two columns `in_reply_to_status_id` and `in_reply_to_user_id` are taking non null values. By which, we have used this information to remove all reply tweets records and then we have removed the columns itselves because no need for them anymore. (i.e. 78 reply tweets have been removed)

3- **Quality Issue**
In `df_enhanced` table, when a row refers to retweet case the three columns `retweeted_status_id`, `retweeted_status_user_id`,
and `retweeted_status_timestamp` are taking non null values. By which, we have used this information to remove all retweets records and then we have removed the columns itselves because no need for them anymore. (i.e. 181 retweets have been removed)

4- **Quality Issue**
In `df_enhanced` table, the column `expanded_urls` presents the dog photos links mentioned with each tweet. Somtimes `expanded_urls` takes Null values. we found 59 records that dog photos aren't included inside. Therefore, we have removed such records. All of these records happened with retweets and reply tweets.

5- **Quality Issue**
In `df_enhanced` table, the name `O` have been replaced with `O'Malley` as mentioned within its tweet text

6- **Quality Issue**
With `name` column in `df_enhanced` table, some dog's names are `None` or starting with lowercase letter like (a, actually, all, an, by, getting, his, incredibly, infuriating, just, life, light, mad, my, not, officially, old, one, quite, space, such, the, this, unacceptable, very). These names are wrongly extracted from their tweets text or its name doesn't mentioned along within their tweet text. we found a pattern for some names which were wrongly extracted and have been fixed.

7- **Tidiness Issue**
In `df_enhanced` table, the columns `doggo`, `floofer`, `pupper`, and `puppo` have been merged and reduced to one column named `stage`.

8- **Quality Issue**
In `df_enhanced` table, there are only 4 different sources in the `source` column that we have replaced them with short words (i.e. `Twitter for iPhone`, `Vine - Make a Scene`, `Twitter Web Client`, and `TweetDeck`).

9- **Quality Issue**
In `df_enhanced` table, the `rating_denominator` column has 23 value which are not equal to the basic 10 denominator value. In addition to existing high values with `rating_numerator` column therefore, we have fixed that.

10-**Quality Issue**
In `df_pred` table, the column `jpg_url` presents the photo link which used in prediction process but if we get a deep look we found that the column `img_num` presents the photo number or order within its set of photos for each tweet (this set is listed in 'expanded_urls' column with `df_enhanced` table). so, we removed `jpg_url` column.

11-**Tidiness Issue**
In `df_updated` table, the two columns `id` and `id_str` in `df_updated` table are duplicated. We have removed one of them and rename the another `tweet_id`.

12-**Tidiness Issue**
In `df_updated` table, both `retweet_count` and `favorite_count` columns have been merged with `df_enhanced` table via `tweet_id`. (Note that each tweet in `df_enhanced` table is expected to has equivalent in `df_updated` table)

13-**Tidiness Issue**
`df_pred` table has been merged with `df_enhanced` table via `tweet_id` (Note: There are tweets in `df_enhanced` that don't have a prediction record in `df_pred`).

Finally, after finishing our assessment and cleaning, we got a clean table named `df_final`. By then, this table was saved to a new file named “twitter_archive_master.csv” file.