Data Wrangling, Analysis and Visualisazion using Twitter API for WeRateDogs Dataset - Wrangling Report

By Michaela C

Wrangling is about gathering the data, accessing it and cleaning the unwanted data also fixing some errors in data. In this WeRateDogs Dataset I have done some wrangling as a part of the Udacity curriculum. WeRateDogs twitter account posts pictures and video's of dogs. It also give ratings and also describe dogs in its own fashion.

Step 1: Gathering the data

In this step, I have gathered data from Udacity "twitter_archive_enhanced.csv" through manual download and 'image_predictions.tsv' through programmatically. Also have gathered data from twitter API. It was the first time ever I extracted data using API. As a whole I loaded these 3 files to 3 separate dataframes.

To gather data from twitter API, I had to sign up to twitter account and had to create an app to get the credentials. Once credentials were acquired, code had to be written to gather data of twitter id's (use tweet_info=api.get_status(tweet_id,tweet_mode='extended')) that were present "twitter_archive_enhanced.csv". After gather data from twitter API I have deleted my credential and commented out all the code used for gathering.

In points:

1.1 Gather data from file on hand Use pd.read_csv() to read data from existing file twitter-archive-enhanced.csv and save it as df1.

1.2 Download file using requests library and URL Download file image_prediction.tsv programmatically from the Internet and store data in df2.

1.3 Gather data from twitter API using Python's Tweepy library and store data Get retweet_count and favorite_count from twitter API for records with tweet_id from df1. Save data as text file tweet_json.txt, then read the file and store data in df3.

Step 2: Accessing data

In this step, accessing data was made both visually and programatically. First I looked at the dataframes informations such as name of the variables, its lengths, if some data are null or duplicated and at the type of the variables. Then I look to the variables value_counts() and I tried to identify what kind of ratings exist, what kinds of breeds exist, how look the photo with the breed which was identified with the lowest confidence. I identified some errors in gathered data.

I separated messy and dirty data issues:

2.1 Messy data

Here we look at the structural issue.

1. In df1: Dog category should be 1 column instead of 4. Rearrange columns in the table
2. Merge df1 with df3 (retweet and favorite count) and with df2(images and breed predictions) - so we will have all data in one place for analysis, even if we loose some data as a whole.

2.2 Dirty data

Here we look at the quality issues

1. We only want ratings with images. Missing values in df2: 2075 ids have images, but in df1 we have 2356.
2. Missing values in df3: 2075 ids have images, but in df1 we have 2356.
3. Too many columns in df1, only some are needed for this project. Missing data for reply_to_user_id/retweeted_status_user_id (NaN)
4. In df1, rows with null expanded_urls are actually not tweets with dog pictures. They are typically replies from "WeRateDogs" to the followers.So we have actually 2297 valid tweets.
5. In df1 some ratings are wrong. F.e. https://twitter.com/dog_rates/status/666287406224695296 should be 9/10 instead of 1/2. Special ratings have typically photos with lot of dogs/puppies on the photo. Ratings with low numerator (<7<7) are usually just for fun and a tweet has no picture of dogs in it or it is a rating with last digits in float.Mistakes are made by some additional numbers in the text (except of ratings).
6. Tweets with numerator=0 are not valid.
7. In df1, erroneous datatype for timestamp. It is object (string) instead of datetime.
8. In df1, missing data represented as 'None' in columns 'name', 'doggo', 'floofer', 'pupper','puppo'.
9. In df1, some dog names are not correct.
10. Some of the tweets are just
11. In df2, 543 predictions are not dogs and so with variables p1_dog&p2_dog&p3_dog equal False, f.e. tweet_id 707693576495472641 (I would guess it is a collie). It needs human insight to identify breeds in these cases - which is not a part of this project.
12. In df3, erroneous type for favorite and retweet counts. It is object (string) but it should be integer, since we need it later for rating.
13. Merge df1 with df3 (retweet and favorite count) and with df2(images and breed predictions) - so we will have all data in one place for analysis, even if we loose some data as a whole.

Step 3: Cleaning data

3.1 Copy *df1*, *df2*, *df3* as *df1_clean*, *df2_clean*, *df3_clean*.

3.2 These issues were handled:

1. Issue
Too many columns in df1 which are not needed
Define
Remove abundant columns

2. Issue
Some expanded_urls are null. These are not tweets with a new picture of a dog.
Define
Drop all rows with missing expanded_urls.

3. Issue
Instead of one column for the variable dog_stage we have 4 columns for each possible values. Some of the dogs have are without the type, so we change these values to be recognised as NaN.
Define
Make a new column "dog_stage": Replace all "None" values with empty space, than join the strings in all 4 columns - doggo, floofer, pupper, puppo. After this, just remove all of these 4 columns.

4. Issue
df3 can be part of df1.
Define
Join df3 table to df1 table, joining on tweet_id.

5. Issue
Wrong format of timestamp in df1_clean.
Define
Convert timestamp to datetime format.

6. Issue
retweet_count and favorite_count should be in number format
Define
Convert retweet_count and favorite_count to float.

7. Issue
In df1, some ratings are wrong. If numerator is equal to 0, tweet is no valid. Tweet with id 666287406224695296 should be 9/10 instead of 1/2 and 716439118184652801 rating is 11/10.
Define
Drop tweets with numerator equal to 0. Correct tweet_ids 666287406224695296 and 716439118184652801 ratings. Make a special dataframe containing tweets with special ratings.

8. Issue
Some of the dogs are not identified as dogs in df2.
Define
Drop data which aren't identified as dogs, because they cannot tell us the breed of the dog.

9. Issue
We need a dog breed and image to so much tweets as possible. Merge df2 with df1.
Define
First drop columns which are not needed for our analysis in df2 - img_num, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf, p3_dog. Then merge df2 to df1.

Step 4: Store Data

Store the clean DataFrame df1_clean in a CSV file named twitter_archive_master.csv and df2_clean in additional file 'twitter_image_predictions.csv'.