Data Wrangling Project On WeRateDogs Twitter Data

Ayush Umrao

About

I completed this project as part of Udacity's Data Analyst Nanodegree. The project is based around the "WeRateDogs" Twitter page, a page which will kindly rate pictures and videos of dogs out of ten. Since dogs are all round fantastic creatures, all of WeRateDogs' ratings are above ten. They also tag each dog with a different category out of "doggo", "floofer", "pupper", or "puppo".

An archive of this Twitter data for WeRateDogs' tweets was provided for this project as a CSV file. Two more sources of data were also gathered as part of this project: predictions for which type of dog is present in each picture (carried out previously, not by myself, by being passed through an image classification algorithm) and additional tweet information acquired from Twitter.

I approached this project using the three steps of data wrangling: gather, assess, clean. In the gather phase, the image prediction data was downloaded using Python's Requests library. The additional Twitter information (i.e. retweet and favorite counts) was downloaded using the Twitter API. In the following assess step, I then inspected the generated data frames in order to find any quality or tidiness issues. The cleaning step subsequently involved implementing steps to fix the quality and tidiness issues that were previously identified.

Following the data wrangling process, and some exploration and analysis of the (now clean and tidy) data, was carried out, numerous interesting results were observed.

Wrangling Methods

I used general assessment techniques to identify quality and tidiness issues. These were .head() , .value_counts(), .sample() etc. I identified eight issues quality issues and two tidiness issues within this master dataset.

Quality of dataframes

archive

1. There are 181 retweets (**retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp**).
2. There are 78 reply tweets (**in_reply_to_status_id, in_reply_to_user_id**).
3. There are 2297 tweets with **expanded_urls** (links to the tweet), indicating 59 tweets with missing data.
    a. 56 of these tweets are replies or retweets.
    b. The remaining 3 tweets have not got the url within the **text** column. They are NOT in the **predictions** table, but even though they ARE in the **json_data**, there was NO image url in the JSON data.
4. The **timestamp** column is in string format.
5. There are 109 tweets with regular words in the **name** column that are NOT a valid name; these words are always the 3rd word in the tweet and are all lowercase; all valid names start with an uppercase letter.
6. There are 775 tweets with the dog **name** as "None". (Probably not worth looking at as there are too many to verify.) [_**This issue will not be cleaned**_]
7. Ignoring replies and retweets, there are 17 tweets with **rating_denominator** NOT equal to 10.
    a. 4 tweets have the correct rating within the text, and can be manually fixed:
        i. 740373189193256964: replace 9/11 with 14/10
        ii. 716439118184652801: replace 50/50 with 11/10
        iii. 682962037429899265: replace 7/11 with 10/10
        iv. 666287406224695296: replace 1/2 with 9/10
    b. 13 are about multiple dogs/pups, and can be dropped.
8. There are 28 tweets with **rating_numerator** >= 15. The max value is 1776, which does not make sense. When we only look at tweets with **rating_denominator** of 10, there are 12 tweets with **rating_numerator** >= 15. Going further, by ignoring the 7 retweets and replies (these are not “original” tweets as specified in the **Key Points**) we end up with 5 tweets with a **rating_numerator** >= 15.
9. There are only 4 types of values in the **source** column, and they can be simplified by using the display string portion just before the final "<\a>":
    a. Twitter for iPhone
    b. Vine - Make a Scene
    c. Twitter Web Client
    d. TweetDeck

predictions

1. There are 2075 image predictions, 281 less than the number of tweets in the archive, so will be classified as "missing data".

json_data

1. Several tweets (16 _during this run_) from the original archive table have been deleted since the archive was created (TweepErrors were reported).

Tidiness of dataframes

archive

1. There are 4 columns for dog stages (**doggo, floofer, pupper, puppo**). The 4 columns for one variable doesn't conform to the rules of "tidy data".
2. The **Key Points** indicates that we're only interested in "original tweets", no "retweets"; this data is stored in the columns **retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp**.
3. Reply tweets are also not "original tweets" either; this data is stored in the columns **in_reply_to_status_id, in_reply_to_user_id**.
4. When all **rating_denominator**s are the same (10) this column is no longer needed.
5. Columns with numerical data are located to the far right of the table, which makes it difficult to readily see the data that will be used for analyses.

predictions

1. The table does not follow the rules of "Tidy Data"; the columns **p1**, **p2** and **p3** contain the same type of data, predictions. The columns **p1_conf**, **p2_conf** and **p3_conf** all contain values for confidence level, and columns **p1_dog**, **p2_dog** and **p3_dog** all contain Boolean values indicating whether the prediction is in fact a type of dog. [_**This issue will not be cleaned**_]
2. The column **jpg_url** contains a link to an image file (JPG), but it's not the same link as the **expanded_urls** field in the archive dataset. It is actually the same link as the **media_url_https** field in the tweet's JSON data. [_**This issue will not be cleaned**_]
3. The dog breed prediction with the highest confidence level can be combined with the archive table as the twitter table contains information that is all about the dog in the tweet.

json_data

1. The json_data table should be combined with the archive table.

Cleaning Data

As all the quality and tidiness issues were related to archive dataframe, I created a copy of only this table and named it archive_clean. For each quality/tidiness issue, I performed the programmatic data cleaning process in 3 stages - Define, Code & Test. During the cleaning process, I converted the datatypes of source and newly created columns of archive_clean to category data type.

Storing Data

After the completion of the cleaning process, I stored the archive_clean DataFrame in twitter_archive_master.csv file.