Wrangle WeRateDogs Data

Gather

Data was successfully gathered from three different sources:

1. The WeRateDogs Twitter archive given by Udacity
2. The tweet image predictions downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv
3. Additional data beyond the data included in the WeRateDogs Twitter archive using Tweepy to query Twitter's API

The code for (3) is saved in the `wrangle_act.html` and is only executed once to store each tweet's entire set of JSON data in a file called 'tweet_json.txt' file. Each tweet's JSON data was written to its own line. Only 2,345 tweets were saved because the other tweets were already deleted. Additional data such as `quote_count` and `reply_count` would have been included but they were only available with the Premium and Enterprise tier products.

Each source was imported in a separate pandas DataFrame namely: (1) archive, (2) image, and (3) twitter.

Assess

The data was assessed visually first, then programmatically for quality and tidiness.

Visual

Each piece of gathered data was displayed in the Jupyter Notebook for visual assessment purposes. Then, they were additionally assessed in Excel.

The quality issues spotted during initial visual assessment are the following:

- Source column is in HTML-formatted string, not a normal string
  The tags are included. This was further assessed using `pd.value_counts()`.
- Replies and retweets are included (78 and 181, respectively)
  These were spotted when the column `in_reply_to_status_id` and `in_reply_to_user_id` contained values.
- Missing URLs in some rows
  These rows were blank.
- Numerator ratings have suspicious outliers
  Unusually high values such as 143 and 1776 were spotted. This was confirmed by using `pd.value_counts()`.
- Denominator ratings have values not equal to 10
  Other values such as 50 and 80 were spotted. This was confirmed by using `pd.value_counts()`.
- Missing dog names (745 with None)
  A lot of dogs had no names. When checking against the tweet text, there were really no names.
- Incorrect dog names (55 with 'a', 8 with 'the', and 7 with 'an')

Some dogs had names such as ‘a’, ‘the’ and ‘an’. When checking against the tweet text, there were really no names. These may be caused by certain keywords used to extract the name.

- Missing dog stages (None in dog stage columns)
  Most dogs had None in the four dog stage columns.
- Not all tweets are dog ratings
  Checking the text of some tweets did not produce a rating. Some were random numbers such as 420 and 1776. These may be caused by certain keywords used to extract the numerical ratings.
- Predicted dog breed is separated by underscore
  All dog breeds had underscores.

Programmatic

The rest of the quality issues can be spotted programmatically using pandas’ functions and methods.

- Tweet ID columns for archive and image are int64, not strings
  Using pd.info() revealed that this is an int64.
- Timestamp column is a Python object, not a datetime
  Using pd.dtype revealed that this is an object (or a string).
- Missing images (only 2075 counts out of possible 2356)
  Calling the DataFrame showed only 2075 rows out of the possible 2356 from the archive. Not all tweets from the archive had images.
- Missing tweets (2345 out of possible 2356)
  Calling the DataFrame showed only 2345 rows out of the possible 2356 from the archive. The 11 missing tweets were deleted.

The two tidiness issues were identified using a combination of visual and programmatic assessment. These were based on the rules of tidy data. It will be easier to extract needed variables when the data is tidy because it provides a standard way of structuring a dataset.

Clean

The define, code, and test steps were used in cleaning process. The details are shown in the wrangle_act notebook. First, copies of the DataFrames were created prior to cleaning. The process can be divided into four major categories: Excess Data, Missing Data, Tidiness and Quality.

Excess Data

One of the key points when data wrangling for the project was only wanting original ratings (no retweets) that have images. Not all tweets are ratings, and some are retweets and replies. Before doing any cleaning, it makes sense to remove the tweets which are retweets and replies first to reduce the amount of data to be cleaned. Then, those columns which contained values for them will be deleted. Out of the original 2356 rows from archive, only 2097 rows remained.

Missing Data Part A

For missing URLs, it should be remembered that tweet_id is part of the tweet URL after “status/” (https://twitter.com/dog_rates/status/[tweet_id]. Thus, filling the missing values needed the http string plus the tweet id.

Missing and incorrect dog names are a bit tricky to fill. Individual tweets should be checked visually if the name is somewhere in the string. However, this will require a lot of time and according to the key points, not necessary to practice and demonstrate skills in data wrangling. The incorrect dog names were significantly smaller in value, so they can be checked individually. No names were found in those tweets and they can be safely assigned a value of None.

Tidiness Part A

The issue of one variable in four dog stage columns was the trickiest to clean. There was a discussion by David Venturi in Slack on the appropriateness of using `melt` in combining multiple columns to just one value column (as opposed to the variable column and value column). The main issue was dropping the correct three duplicate rows with four 'None' doggo, floofer, pupper, and puppo values. The issue was exacerbated when there were some rows with multiple stages. It took a significant time to code and test but unfortunately, the `pd.melt()` solution could not be obtained.

Believing that there are multiple ways to solve a problem yielded a workaround to combine the four columns into a single column containing the variable. First, a new column was created which added all the four columns. Then, the final column would then be determined based on the sum of the previous four columns. For two dog stages, a value of 'Multiple' will be assigned. The code is clearly shown in the notebook.

Quality

Twitter suggests using a string rather than a large integer to store tweet ID. It also makes sense for them to be string types since their numbers don't mean anything in terms of value.

The timestamp column should be converted to a datetime object to do some time series analysis. Fortunately, the string can easily be converted to datetime without doing any modifications.

The source column in HTML formatted string can be left as is but would be much nicer to convert into the normal string value. Then, it was converted into a categorical value.

Numerator ratings having suspicious outliers are also tricky to clean. There are some ratings which are based on some random integers in the twitter text (ex: 24/7). It would be time consuming to individually check the tweets to confirm if the rating is correct. Thus, it will be assumed that ratings 14 and below will be assumed correct. Only the outliers will be cleaned. Fortunately, there are only 18 outliers, so they can be visually checked. Ratings with decimals will be rounded to the nearest integer (11.26 would yield to 11) while fractions will be simplified (44/40 to 11/10). For ratings which cannot be identified (1776), a value of 10 will be used.

Denominator ratings will get the same treatment for numerators. All remaining denominators will be set to 10. Another option would be to delete the column since it will have no bearing in the analysis.

Dog breeds with underscores can be easily changed to have a space and to have title format.

Tidiness Part B

The reason why merging the three tables is one of the last steps is that cleaning individual tables is easier than cleaning a larger table. The three tables will be combined into a single DataFrame, joining on the Tweet ID which was previously converted to string type. Only 1971 rows from the image table were merged. Some of the images were retweets or replies which were previously deleted.

Missing Data Part B

This last issue was not previously identified during the assessment part but was identified when the tables were merged. Going back to the key point of original tweets that have images, the tweet IDs that have no images can be safely deleted. Getting the image and running the algorithm of those tweets is beyond the scope of this project and can be explored in the future.

Store

The final DataFrame contains 1971 rows and 22 columns with the correct data types. The master dataset is then stored in a CSV file called 'twitter_archive_master.csv'.

It was completely satisfying to wrangle the data and make it ready for analysis.