Wrangle and Analyze Data Report

Mohammed M. Huda

Version: 1.00
Date: May 24, 2020

INTRODUCTION

Data wrangling is the process of gathering raw data from a variety of sources, then cleaning and structuring this raw data into a desired format for better decision making in less time.

There are six iterative steps that make up the data wrangling process:

- Discovering: Understanding what is in your data and how you will analyze it
- Structuring: Organizing the data for better computation and analysis
- Cleaning: Improving the quality of the raw data received
- Enriching: Determining what new types of data can be derived from what has already been gathered or what other information would better inform our decision making about the current data
- Validating: Verifying data accuracy, consistency, quality, and security
- Publishing: Preparing the wrangled data for downstream use, including analytics

This project is intended to take data from a popular Twitter account @dog_rates (WeRateDogs), which rates photos of dogs and comments. We will download the archive presented in CSV format, the images in TSV format, then create a JSON file for further analysis and exploration.

This project will also make use of Python's Tweepy library. This library can be installed on Jupyter notebooks by visiting the Tweepy website and downloading the installation files. Link: http://docs.tweepy.org/en/latest/install.html

It is best to use the Tweepy library that's already installed on Udacity's servers, as it is consistently updated with the newest version.

GATHERING DATA

Required files:

WeRateDogs Twitter archive:
https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv

Image Predictions file:
https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

Required code for accessing the Tweepy library (requires a Twitter account and access to the API: http://developer.twitter.com/).

```
# NOTE:  For consumer_key, consumer_secret, access_token,
# access_secret, you have to obtain a developer account on Twitter,
# which will guide you through the process on obtaining such an
# account.  The keys are randomly generated and will consist of a
# string of alphanumeric characters.  This is required in order to
# generate the tweet_json.txt file

consumer_key = 'YOUR CONSUMER KEY'
consumer_secret = 'YOUR CONSUMER SECRET'
access_token = 'YOUR ACCESS TOKEN'
access_secret = 'YOUR ACCESS SECRET'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth,
                 wait_on_rate_limit = True,
                 wait_on_rate_limit_notify = True)
```

Gathering of the data should produce another required file, tweet_json.txt, that is 10 MB and consist of the following readout during our tests:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2331 entries, 0 to 2330
Data columns (total 3 columns):
tweet_id          2331 non-null int64
retweet_count     2331 non-null int64
```

```
favorite_count    2331 non-null int64
dtypes: int64(3)
memory usage: 54.7 KB
```

During our tests, we noticed that the Python cell creating the tweet_json.txt file timed out when the file reached 4.46 MB, then timed out again when the file reached 8.23 MB. It was best to let the cell run as is, without any interruptions.

ASSESSING THE DATA:

We used the following panda functions to perform our assessment:

- info()
- describe()
- value_counts()
- sample()

Observations:

1. The data types for many of the attributes in our dataset are incorrect. For example, in_reply_to_status_id is listed as a float, but it really isn't. And many of the values in in_reply_to_status_id contain "NaN" (Not a Number, which is really just a null value).
2. The following data types are also incorrect: timestamp, tweet_id, retweeted_status_id, retweeted_status_user_id
3. Sources are not readable, but users posted their tweets from the following sources: Twitter app for iPhone, Vine, Twitter Web Client, and Tweetdeck. Interesting things about this:
    a. Twitter now owns Tweetdeck
    b. Vine is now discontinued as of 2019
    c. Nobody posted their tweets using Twitter for Android
4. Names of dogs are "a", "actually", "an", "the", "this", "very", "quite"; clearly this is inaccurate
5. Missing dog types or dog breeds
6. There are only 181 retweets (retweeted_status_id, retweeted_status_userid, retweeted_status_timestamp). These will be removed since we're interested in original tweets.
7. Number of images (2075) is not consistent with number of tweets overall (2356).
8. p1, p2 and p3 contain values that are of FALSE, and those entries should be removed (since they're images of animals other than dogs)
9. Ratings should be between 1 - 10, but rating_numerator has values that are higher than 10. This is okay, since ratings higher than 10 are meant for humorous effect and not to be taken seriously.

CLEANING THE DATA:

Having three separate dataframes makes our data look cluttered and confusing. For our project, cleaning will require us to create copies of the following:

- Our original Twitter archive (i.e., the WeRateDogs Twitter archive)
- Image Predictions file
- The tweet_json.txt file

The main objective is to clean each file (twitter_archive, tweet_info, image_predictions) and combine the cleaned files into one new Twitter archive dataframe (new_twitter_archive).

In cleaning our original Twitter archive, we performed the following:

- Removed all instances of "retweeted_status" and "in_reply_to", since they're irrelevant to our objectives
- Combined four dog categories into one category, drop unnecessary columns
- Removed tags identifying the sources where the tweet messages originated, so that they are more readable when viewing the new Twitter archive dataframe
- Merged the clean twitter_archive and tweet_info files into the new_twitter_archive
- Changed the timestamp format and added columns for year, month, day, and time
- Merge the "tweet_id" attributes from twitter_archive and tweet_info
- Create a new column for the dog prediction summary in image_prediction
- Join the image_predictions_clean dataframe with the new_twitter_archive
- Clean "name" column by converting non-names to "none", then dropping them
- Remove underscore from predictions
- Convert confidence levels to a percentage
- Remove dogs labeled "Not a dog" from new_twitter_archive
- Replace p1, p2, and p3 with appropriate prediction varbinds
- Create a copy of the dataset that confirms dogs, and we'll call this dog_twitter_archive
- Create the official *.csv files: new_twitter_archive = twitter_archive_master.csv and dog_twitter_archive = twitter_archive_dogs.csv