DOG RATES DATA WRANGLING

In this project we will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. Finally, we will implement some analyses and visualizations using Python and its libraries.

DATA GATHERING

We gathered data from three different sources; manually downloaded a csv file 'twitter_archive_enhanced.csv', loaded a TSV file programmatiacally using requests library and finally extracted a json file using Twitter API to extract WeRateDogs tweets. At the end of different gathering method we imported data to a dataframe.

Downloading CSV file manually

- Enhanced Twitter Archive:
The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. One column the archive does contain though: each tweet's text, used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced." Of the 5000+ tweets, Only tweets with ratings are listed (there are 2356). We stored those columns in a data frame called « archive »

Loading TSV file programmatically

The tweet image predictions, i.e., what breed of dog (or other object, animal, etc.) are present in each tweet according to a neural network. This file (image_predictions.tsv) is hosted on Udacity's servers and was downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv and stored in a data frame called « predictions »

LOADING json file using Twitter api

We gathered each tweet's retweet count and favorite ("like") count. Using the tweet IDs in the WeRateDogs Twitter archive, We queried the Twitter API for each tweet's

JSON data using Python's Tweepy library and stored 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. Then we read this .txt file line by line into a pandas Data Frame called tweet_df with tweet ID, retweet count, and favorite count.

ASSESSING DATA

We opened 'twitter_archive_enhanced.csv' in a spread sheet to operate a visual assessment and found that 'source' column contains useless HTML tags and name column contains inappropriate names that needed to be fixed. We've also noticed that Columns 'doggo', 'floofer', 'pupper', 'puppo' need be combined in one categorical variable

Then we used python library pandas to operate a programmatic assessment with methods like describe(), info(), duplicated()…etc we found missing data and inappropriate datatype in several columns,

The detailed list of tasks to be performed in the cleaning process is below.

1) **Archive dataframe**

Quality issues

- 'tweet_id' should be str
- 'timestamp' should be datetime
- 'source'contains useless characters
- 'source' should be categorical
- 'text' should be string
- retweets need to be removed
- 'rating_denominator'should be 10
- 'rating_numerator'should be greater than 10
- Inappropriate name need to be fixed

Tidiness issues

Several columns with too many NaN

- in_reply_to_status_id
- in_reply_to_user_id
- retweeted_status_id
- retweeted_status_user_id
- retweeted_status_timestamp
- expanded_urls

Columns 'doggo', 'floofer', 'pupper', 'puppo' must be combined in one categorical variable

2) Predictions dataframe

Quality issues

- 'tweet_id' must be str
- 'img_num' must be str

Tidiness issues

- Merge dataframe with archive and tweet_df to obtain twitter_archive_master.csv

CLEANING DATA

We've made a copy of our dataframes to perform the cleaning process without damaging original dataframes.
For automation and time saving we've cleaned the data programmatically. For every quality or tidiness issue we've defined the problem, coded it using python methods and finally tested the result. Finally, we've merged our cleaned data in one called 'twitter_archive_master.csv'