Data Wrangle Report

By Zeyad Remaih

January 2021

Introduction:

The purpose of this report is to illustrates the main steps involved in the data wrangling of twitter account – WeRateDogs.

Data Gathering:

The data for this project consist on three different dataset that were obtained as following:

- **Twitter archive file(df_ta):** the twitter_archive_enhanced.csv was provided by Udacity and downloaded manually.
- **The tweet image predictions(image_prediction)**, i.e., what breed of is 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 URL information
- **Twitter API & JSON(df_json):** by using the tweet IDs in the WeRateDogs Twitter archive, I 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. I read this .txt file line by line into a pandas dataframe with tweet ID, favorite count, retweet count, followers count, friends count, source, retweeted status and url.

Assessing Data:

In this step I started exploring the three dataframes we have:

Visually:
By printing the first lines in the dataframes separately in Jupiter Notebook

Programmatically:
by using different methods like (info, value_counts, duplicated, groupby, etc..)

Then I listed the quality and tidiness issues.

Cleaning Data:

First I started by taking copy from each dataframe so I can work on the copied dataframes rather than the original data frames.

-**1st dataframe** : df_ta_clean
-**2nd dataframe** : image_prediction_clean
-**3rd dataframe** : df_json_clean

Quality

Here are the issues that I worked on:

- **Twitter Archive file (df_ta_clean):**
    - Keeping the original ratings with no retweets and that have images
    - Delete columns that won't be needed
    - based on tweet_id except for the last occurrence.
    - Separate timestamp into three different columns
    - Rating Numerator correction as it had some wrong values that I needed to change manually and programmatically.
    - Rating denominator correction that should be 10 but there was some other extreme and wrong values , I corrected them manually and programmatically by checking the texts and images for these
    - Correcting the name column ('a','an', …), I created a pattern to extract the possible names from the text

- **Image prediction file (df_json_clean):**
    - Creating 1 column for image prediction and 1 column for confidence level
    - Dropping duplicating images
    - Drop unwanted columns

- **Tweet JSON file (df_json_clean):**
    - Keeping only the original tweets
    - Changing (tweet_id) column type to column name to (int64)

Tidiness

- Twitter Archive file (df_ta_clean):
    - Merging Twitter Archive and Image prediction to make columns part of one dataset

- Tweet JSON file (df_json_clean):
    - Erroneous datatypes (doggo, floofer, pupper and puppo columns)
        - Melt the doggo, floofer, pupper and puppo columns to dogs and dogs_stage column. Then drop dogs. Sort by dogs_stage in order to then drop duplicated

Storing, Analyzing and visualization

- Save master dataset to a "**twitter_archive_master.csv**" file.
- The master dataset is analyzed using pandas in the Jupyter Notebook and at least three (3) separate insights are produced.
- Four (1) labeled visualization is produced in the Jupyter Notebook using Python's plotting libraries.

1. **Insight 1 and visulization** - Golden retriever is the most common dog in the dataset
2. **Insight 2 and Visualization** - Tweets rate per Year
3. **Insight 3 and visualization** Retweet counts and Favorite counts are correlated
4. **Insight 4 and visualization** - Displaying the most common dog names