Data Wrangling Report

By Amr Elsayed Elaref

January 13

As a Udacity student in the Data Analyst Nanodegree. I have an assignment for wrangling data in order to analyze and build useful insights. This report illustrates the different wrangling steps for a Twitter account called “WeRateDogs”.

Data Gathering

This step contains collecting data from different sources. In this project, there were three sources for the data to deal with:

1. "Twitter-archive-enhanced.csv" file which was manually downloaded from Udacity server to my working directory, then imported to my working environment as a DataFrame using Pandas library function “pd.read_csv”.

2. “Image_prediction.tsv” file which has been hosted on a webpage and downloaded programmatically using Requests library function “requests.get(the webpage url)”, then using Pandas library function “pd.read_csv” to read this file as DataFrame. This file contains, i.e., what breed of dog is present in each tweet according to a neural network.

3. “Tweet_json.txt” file which was gathered from Twitter API via the Tweepy library by querying the API to obtain extra information pertinent to tweets’ ids in the first file, e.g retweets counts and favorite counts. Then using Pandas library function “pd.read_csv” to read this file as DataFrame.

Data Assessment

This step contains investigating the gathered datasets in order to find quality and tidiness issues.

- **Visual Assessment**: each piece of gathered data was displayed in the Jupyter Notebook for visual assessment purposes. Once displayed, data was additionally be assessed in an external application, e.g. Excel, VS Code.

- **Programmatic Assessment**: Pandas' functions and/or methods are used to assess the data.

1. **Quality issues**: there are four dimensions for dealing with quality issues:

   i. Completeness: dealing with missing data.

   - Missing values of name column
   - Missing values of rating columns

   ii. Validity**:** we have the records, but they're not valid, i.e.*, they don't conform to a defined schema*.

   - Records that represent retweets or replies
   - Records with no image

   iii. Accuracy: inaccurate data is wrong data that is valid.

   - Wrong names extracted from text
   - Wrong ratings extracted from text

   iv. Consistency: inconsistent data is both valid and accurate, but there are multiple correct ways of referring to the same thing.

   - Tweet_id column was repeated in the three tables
   - Some columns needed to be dropped as they were not useful in forward analysis
   - None values instead of NaN in different columns

2. Tidiness issues:
    - The three DataFrames should be in one table
    - doggo', 'floofer', 'pupper' and 'puppo' are column headers instead of values
    - The image prediction table was too complicated it should be shortened into only two columns: breed and confidence
    - The columns in the twitter_archive_master should be ordered as important columns shows first

Data Cleaning

All issues identified in the assess step are successfully cleaned (if possible) using Python and pandas, and include the cleaning tasks required to satisfy the Project Motivation

Firstly, I created copies of the original dataframes to apply the cleaning codes on it so I have the original dataframes without any changes in order that I could Refer to it for reference.

Then, started the cleaning step in that order:

- Remove retweets and replies from data, then drop these columns
- Combine doggo, floofer, pupper and puppo columns into one columns called dog_stage then drop these four columns
- Merge the archive table with api table then drop the id column
- Remove tweets with no image by merging archive and image_predictions tables

  Note: in the clean data set, there were no records with favorite count equals zero although I didn't drop them. So, it seemed that those tweets are without images so they dropped while merging with the image prediction table.

- extract the clean source from source column using Regular Expression (REGEX) as the source column was not clear and difficult to read
- Replace all "None" value in the master table with "Nan" in order to guarantee consistency of all data

- Extract the correct name from the tweet full text using REGEX
- Correct the wrong datatype of columns
- correct the errors of “rating numerator” and “rating denominator” columns using some REGEX and make all values of rating denominator equal to 10
- Drop rows that don't contain retweet and favorite count
- Shorten the “image predictions” table into two columns: "breed" and "confidence" and then drop the remaining columns from the master table
- Finally, Rearrange the columns of the “twitter archive master” table so important columns show first

Storing Data

Finally, I stored the clean DataFrame using Pandas library in a CSV file named “twitter_archive_master.csv”.