Wrangling Report

WeRateDog Data Wrangling and Analysis

Abstract

Documentation of the steps used to wrangle and analyze data from WeRateDog Twitter account based on the methodology we learnt in the data wrangling course

Kareem Zaghawa

Data Wrangling

Data Gathering :

There are 3 sources of data that we need to gather data from and each of different nature :

- Enhanced Twitter Archive, which was provided to us by instructors and loading that into a pandas dataframe is easy
- Image prediction files, which a link of it was provided and by using requests library, it was downloaded and loaded in dataframe
- Some missing information in the tweets given in the Enhanced Twitter Archive, need to be fetched, so querying Twitter API was essential to download the necessary information to gather all data.

Data Assessing:

Upon investigating the data there were quality issues and tidy issues that needed to be fixed.

Quality Issues for:

DataFrame 1 (given dataset):

- Some of the tweets are retweets
- Some tweets are replies to another tweets
- Some tweets no longer exist
- Some values of numerators are way high like (420,666,1970,...)
- Some values of denominators are more than 10
- Some tweets' URLs are missing
- Some columns are not needed like ('retweeted_timestamp', 'source')
- Some dogs are missing their dog stages
- Some dogs names are None instead of NaN
- Some dog names are not added or aren't valid like (a, one,just,actually, etc..)

DataFrame 2 (data downloaded using Twitter API)

- Some columns aren't needed like (created at, lang, place, coordinates, source,..)

Image Predictions

- The guesses needs investigating to extract dog breeds and merge it into the master data frame

Tidiness issues

- Column headers are values, not variable names (dog 'stage') in DataFrame1
- The two dataframe (enhanced twitter archive, queried data from twitter API) need to be merged.
- The dog breed in image prediction files should be merged with the two dataframes.
- Different names for columns for the same variables in the dataframes (tweet_id in df1, id in df2)
- Timestamp can be broken into 3 columns (hour, day ,month) to analyze when did the account got the most interactions.

Data Cleaning:

- By using retweeted_status_id and in_reply_to_status_id columns in dataframe1, we can eliminate the retweets and eliminating quote tweets as well which are tweets retweeted with a quote in dataframe2
- treated the rating numerators by extracting the rating from the text using regular expression and correcting the whole column
- treated the rating denominator by using the assign function of pandas library
- Gathering dog stages using regular expression from text
- Treating invalid dog names by checking for the different patterns the name is usually written in tweets using regex and using simple code to add the 4 columns into one fixing the tidiness issue
- Breaking Timestamp columns into 3 columns: day, month, week