WRANGLE AND ANALYZE 'WeRateGogs' DATA

Introduction

The purpose of this project is to put in practice what I learned in data wrangling data section from Udacity Data Analysis Nanodegree program.

The dataset that you will be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. WeRateDogs has over 4 million followers and has received international media coverage.

Datasets.

1. Enhanced Twitter Archive

   The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. The Twitter archive is great, but it only contains very basic tweet information. Additional gathering, then assessing and cleaning is required for analyses and visualizations.

2. Image Predictions File

   every image in the WeRateDogs Twitter archive through a neural network that can classify breeds of dogs via different algorithms. The results is a table consists of predictions algorithms (p1, p2, p3) with confidence ratio (p1_ conf, p2_ conf, p3_ conf) and whether or not the prediction is a breed of dog (p1_dog, p2_dog, p3_dog) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images).

3. Data via the Twitter API

   Retweet count and favorite count are two of the notable column omissions. This valuable data can be gathered from Twitter's API. Using twitter developer account and tweepy.

Project steps

My tasks in this project are as follows:

1. Gathering data.
2. Assessing data.
3. Cleaning data.
4. Storing
5. Analyzing, and visualizing your wrangled data.

Gathering data

Data is successfully gathered from three different sources. Each piece of data is imported into a separate pandas DataFrame at first.

1. Twitter Archive File (.csv)
   This was extracted programmatically by Udacity and provided as a csv file (twitter-archive-enhanced.csv). Using pandas I was able to read this file.
2. Image Predictions File (.tsv)
   This file (image_predictions.tsv) is hosted on Udacity's servers and should be downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv
3. Twitter API (.json)
   Query Twitter API for each tweet in the Twitter archive and save JSON in a text file. Then read the json file (tweet-json.txt)

After reading the three files, convert them into a three DataFrames.

Assessing data

Inspection our collected data sets from both the quality and tidiness per.

1. Visual assessing using pandas basic methods (sample , head , tail) to take a look at the contents of the three DataFrames.
2. Programmatical assessing, by using different methods (e.g. info, value_counts, describe, duplicated, groupby, query, loc, etc).

3. Finally, I document all issues to clean the data in terms of quality and tidiness.

Cleaning data

- First, create a copy of the three original DataFrames to manipulate the copies to avoid issues in the original files. Whenever I made a mistake, I could create another copy of the DataFrames and continue working.
- I have solved 11 quality issues as follows:
    1. df_Twitter_Archive DataFrame : convert [timestamp] to datetime64.
    2. df_Twitter_Archive DataFrame: convert [tweet_id] datatype to object.
    3. df_Twitter_Archive DataFrame: edit the missing and erroneous values of the [name] column.
    4. df_Twitter_Archive DataFrame: convert *None* in [name] column to NAN.
    5. df_Twitter_Archive DataFrame: drop the retweets rows then drop the retweets columns.
    6. df_Twitter_Archive DataFrame: drop missing values from [expanded_urls] column.
    7. df_Twitter_Archive DataFrame: correct denominators and numerator issues.
    8. image_prediction DataFrame: [p1,p2,p3] not descriptive column names.
    9. image_prediction DataFrame: merge the three columns [p1,p2,p3].
    10. image_prediction DataFrame: convert [tweet_id_image] datatype to object.
    11. image_prediction DataFrame: drop duplicated [jpg_url].
    12. Api_df DataFrame: convert [id] datatype to object.
    13. Api_df DataFrame: drop some columns.

- And 2 tidiness issues as follows:
    1. df_Twitter_Archive DataFrame: merge the four columns [dogoo, floofer, pupper, puppo].
    2. Merge the three DataFrame in one dataset.

Storing

1. Sort clean data in a new file.
2. Sort clean data in database.