Wrangle and Analyzing Data Report

STUDENT: ABDULLAH ABUSABBA

Gathering Data:

For this process, I’ve gathered both twitter archive and image prediction data programmatically whereas json file was provided by Udacity it can be retrieved via twitter API if granted by them.

Enhanced Twitter Archive:

The WeRateDogs Twitter archive provided by Udacity. This has simple tweet data for all 5000+ of their tweets, but not everything or all the tweets.

Image Predictions File:

The tweet image predictions, i.e., what breed of dog (or another object, animal, etc.) is present in each tweet according to a neural network. The file is hosted on Udacity's servers and it was downloaded programmatically.

Data via the Twitter API:

Each tweet's retweet count and favorite ("like") count at minimum, and any additional data you find interesting. Using the tweet IDs in the WeRateDogs Twitter archive, query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count. The file is provided by Udacity if the request for twitter PAI wasn’t granted.

Assessing Data:

Data quality assessment (DQA) is the process of scientifically and statistically evaluating data in order to determine whether they meet the quality required for projects or business processes and are of the right type and quantity to be able to actually support their intended use. It can be considered a set of guidelines and techniques that are used to describe data, given an application context, and to apply processes to assess and improve the quality of data.

Cleaning:

In the process of cleaning data we want to modify each data type to it’s proper type and making tidy and with the quality we wanted.

My observations and notes:

I had an idea that I want to implement for the variety of photos, some photos have more than one, so I wanted to extract the src attribute from the image tag at each tweet page to include them in the image URL column in twitter archive data frame. This will duplicate the tweet id but it would make it user friendly, every time I use beautiful soup library with requests I couldn't get what I want. After some investigation it turns out that this could be done via twitter API but I don't have the access.

The second obstacle I had was with the image prediction data, as I wanted to include all predictions (9 columns) then merge them all together. However, I wanted to maintain the unique Id's although I managed to have a data frame with 4 columns (tweet_id , algorithm , confidence , is_dog) the process was to drop duplicated confidence (after the melt of all 9 columns) then to loop over each confidence row in the original image predication data and get the tweet id for each tweet_id that match either p1_conf thru p3_conf. After reviewing Udacity project page they mentioned that p1_conf is how confident the algorithm is in its #1 prediction → 95% so I included p1 only to the final data.

Note:

not all Quality, Tidiness issues will be mentioned

Quality issues:

1. column (1,2,6,7,8) have missing data → twitter archive
2. timestamp is set to str Instead of datetime object→ twitter archive
3. most text start with 'this is {name}' Corresponding to name column but some names are incorrect (i.e. this,a,all,old)→twitter archive
4. source columns data is valid, but there are multiple correct ways of referring to the same thing we can split it and use the source of the Tweet (i.e for iPhone, Twitter Web Client)→ twitter archive
5. Numerator has incorrect values→ twitter archive
6. Denominator has incoreect values→ twitter archive
7. upon merging the data we need to drop some rows/column for the merge to complete
8. remove entries that are retweets→ twitter archive
9. replace None with NaN twitter archive
10. drop duplicates in jpg URL → image prediction
11. Missing data in more than 10 columns → json tweet

Tidiness issues:

1. inconsistent columns (doggo, floofer , pupper , puppo) can be in one single columns 'stage' → twitter archive
2. multiple variables are stored in one column 'expanded URLs '→ twitter archive
3. multiple variables are stored in one column 'display text range '→ json tweet
4. <algorithm columns should be melt () to three columns (prediction , algorithm , confidence)→ image prediction
5. combine data frames