Data Wrangling Report

# For internal users only

This report is intended to describe data wrangling process in a simplified manner and meant for internal use only. The entire process consists of three major steps: gathering, assessing and cleansing. Each step is further divided into sub-steps and they all are shown as following.

1. Gathering
   1. ***twitter-archive-enhanced.csv*** is given by default
   2. ***image-predication.tsv*** is downloaded using requests API and saved as csv file in same folder
   3. The details of each tweets are not give, since we have tweet ids from twitter-archive-enhanced file, all tweets can be queried with its id and downloaded. By using statuses\_lookup () API from tweepy, all query results are downloaded and stored into list *lst\_tweets* which contains multiple lines of json strings, then later it is saved to ***tweet\_json.txt*** in same folder. After that, *lst\_tweets* is imported into *df\_tweets*, and then *tweet\_id, favorite\_count and retweet\_count* are extracted into its corresponding data frames (*df\_retweet\_favorite\_count*)
   4. In the process of query, there are nine ids that failed, these are discarded.
   5. All original data frames are copied into *\_copy* ones for further processing

Note: Using *get\_status*() method to query tweets costs over 3 hours due to rate limit from twitter, therefore, I have turned to *statuses\_lookup* () which takes 100 ids on each query and each limit windows takes 300 queries, resulting in a very short running time of approximately 15 seconds.

1. Assessing
   1. By merging all three data frames:
      1. *df\_retweet\_favorite\_count\_copy*
      2. *df\_twitter\_archive\_enhanced\_copy*
      3. *df\_image\_predication\_copy*

*df\_twitter\_archive\_master* contains all raw data, detailed information is shown below:

<class 'pandas.core.frame.DataFrame'>

Int64Index: 2070 entries, 0 to 2069

Data columns (total 30 columns):

tweet\_id 2070 non-null int64

in\_reply\_to\_status\_id 23 non-null float64

in\_reply\_to\_user\_id 23 non-null float64

timestamp 2070 non-null object

source 2070 non-null object

text 2070 non-null object

retweeted\_status\_id 76 non-null float64

retweeted\_status\_user\_id 76 non-null float64

retweeted\_status\_timestamp 76 non-null object

expanded\_urls 2070 non-null object

rating\_numerator 2070 non-null int64

rating\_denominator 2070 non-null int64

name 2070 non-null object

doggo 2070 non-null object

floofer 2070 non-null object

pupper 2070 non-null object

puppo 2070 non-null object

jpg\_url 2070 non-null object

img\_num 2070 non-null int64

p1 2070 non-null object

p1\_conf 2070 non-null float64

p1\_dog 2070 non-null bool

p2 2070 non-null object

p2\_conf 2070 non-null float64

p2\_dog 2070 non-null bool

p3 2070 non-null object

p3\_conf 2070 non-null float64

p3\_dog 2070 non-null bool

favorite\_count 2070 non-null int64

retweet\_count 2070 non-null int64

dtypes: bool(3), float64(7), int64(6), object(14)

memory usage: 458.9+ KB

* 1. Issues found
     1. Quality Issues
        1. df\_twitter\_archive\_enhanced\_copy
           1. missing values in multiple columns
           2. source column has html instead of text
           3. duplicated rating info
           4. *retweeted\_status\_id, retweeted\_status\_user\_id,retweeted\_status\_timestamp* has missing values over 93% *in\_reply\_to\_status\_id,in\_reply\_to\_user\_id* have missing values over 97%
           5. *timestamp* has +0000 ending
           6. *name* column has characters 'a','actually','all','an','by','getting','his' and so on as names which don't seem to be valid
           7. name contains '*None*'
           8. name column has some with lower case
           9. *tweet\_id* should be string instead of int64
           10. *text* containsratings as float such as 12.5/10 which is not reflected by rating\_numerator column
        2. b. df\_image\_predication\_copy
           1. has less number of instances comparing with *df\_twitter\_archive\_enhanced\_copy*
           2. dog stage name columns contain *'None'*
           3. dog name column contains *'None'*
        3. c. df\_tweets\_copy
           1. missing values in multiple columns
           2. *created at* has +0000 extra string

* + 1. Tidiness Issues
       1. df\_twitter\_archive\_enhanced\_copy
          1. text column contains multiple variables such as rating values and url
          2. dog name/stage name values should be in a columns
       2. df\_image\_predication\_copy
          1. p1,p2,p3, p1\_conf,p2\_conf,p3\_conf are redundant only one column with most probable value needed
       3. dt\_tweet\_copy
          1. *id* and *id\_str* are duplicated
          2. only *retweet\_count* and *favorite\_count* are needed, can be combined into one table

1. Cleansing
   1. Converting tweet\_id from int64 to str
   2. Dropping columns with excessive number of missing values and urls for pictures since the attributes with numerous missing values will not represent any insights, also no more image analysis will be performed, those image urls can be removed too.
   3. Extract text from source column and strip the key word out such as “iphone”
   4. Using regex and lambda to convert ‘timestamp’ values to epoch time so that its values are in simple standard format
   5. *'rating\_numerator','rating\_denominator'* are removed and new column named rating is added from data extracted from text column and represented with float fomat
   6. Replace missing values in *'doggo','floofer','pupper','puppo'* with ‘0’ and replace with existing value with ‘1’
   7. Replace missing values in ‘name’ column with value ‘0’
   8. Creating new column named ‘breed’ from prediction columns *'p1\_dog','p2\_dog','p3\_dog'*  with following conditions:
      1. If *p1\_dog*  is true, then use value from *p1* as dog breed
      2. If *p2\_dog*  is true and If *p1\_dog*  is false, then use value from *p2* as dog breed
      3. If *p3\_dog*  is true and If *p2\_dog and p1\_dog* both are false, then use value from *p3* as dog breed
   9. Drop columns *'p1','p2','p3','p1\_conf','p1\_dog','p2\_dog','p3\_dog','p2\_conf','p3\_conf','img\_num'* from *df\_twitter\_archive\_master* since they will not be useful anymore.

This concludes the entire wrangling process and the cleaned dataset should contains no missing values, no ambiguities and meaningful data for further analysis