Data Wrangling Report

Gather

The project starts with two available datasets:

1. twitter-archive-enhanced.csv
2. image-predictions.tsv

The dataset #1 is available through download on Udacity and dataset #2 is obtained from Udacity server but using the Python package **requests**.

Dataset #1 contains the data about the tweet made by @dog_rates.

Dataset #2 contains the predictions made through Machine Learning algorithm to determine if the picture on the tweet is a dog, and what is the breed.

Also, to improve the dataset #1, it is used the the Twitter API through Python package **tweepy**, this API is used in this project to retrieve more information about each tweet.

Assess

The assessment is performed in two ways, visually and programmatically, the visual assessment is performed with Pandas methods like *head() and sample(),* in this case the visual assessment look for quality and tidiness issues that can be easily detected.

By the programmatic assessment, it uses pandas methods such as *info(), describe() and value_counts.* Also this enables to detect quality and tidiness issues.

Summarizing all findings:

From **Visual Assessment** the following issues are identified:

Quality

df_twitter table (tweets):

- NaN values
  for in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp,
- timestamp, text and expanded_url can have a better descriptive name
- source containing all HTML a tag
- doggo / floffer / pupper / popper - "None" as text
- name with values such as None or a

df_img_pred table (predictions from pictures):

- the prediction columns can have a better descriptive name than p1, p1_conf, p1_dog or jpg_url

df_tweets table (extra data obtained through API)

- NaN values for favorite_count and retweet_count

Tidiness

df_twitter table (tweets):

- doggo / floffer / pupper / popper (all can be in just one column dogtionary or dog_stage)
- once this table relates to **tweets** information about in_reply or retweet can be moved to another table (different observation units)
- rating_numerator and rating_denominator can be converted to a single column rating

df_img_pred table (predictions from pictures):

- None

df_tweets table (extra data obtained through API)

- None

Once it all relates to tweets, these three tables can be on the same table (tidiness, same observational unit)

From **Programmatic Assessment** the following issues are identified:

Quality

df_twitter table (tweets):

- timestamp and retweeted_status_timestamp not in date format
- rating_denominator it is normally **10**, there are unusual values
- rating_numerator it is normmally **little above 10**, but there values less than 10 or really higher
- from the **2355** entries, many of the columns have empty values those derived from retweeted (just 181 non-empty), those derived from in_reply (78 non-empty), those derived from Twitter API and those from picture inference (2075 non-empty)
- wrong name such as **a**, **an**, **the**, **this**
- expanded_urls with multiple values

Tidiness

Regarding the last point above and the **visual assessment**, retweeted and in_reply cases must be removed from the tweets dataset once this is a data analysis of tweets and not retweets

Clean

Finally, the Clean step is performed in three steps:

1. Define
2. Code
3. Test

In summary, it aims to address all issues evidenced on the Assess phase. In this data wrangling project all the Clean phase accomplished all major points raised on the Assess phase.

More details can be evidenced on the files **wrangle_act.ipynb** and **wrangle_act.html**.