Data Wrangling Report

1. Gathering Data

The first dataset (df_1) is provided by Udacity in the form of a csv file (**twitter_archive_enhanced.csv**) which consists of 2356 basic tweet data from November, 2015 to August, 2017.

The second dataset (img_df) is created by reading in (image_predictions.tsv) which is hosted on Udacity's servers and downloaded programmatically using the **Requests** library and the following URL: **https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv**

The third dataset (status_df) is created by gathering data from Twitter API using the tweet IDs in the twitter_ archive_enhanced.csv file. I accessed the entire data for every tweet from Twitter API and stored every tweet's entire JSON data in a file called tweet_json.txt file. Only tweet_id, retweet_count, favorite_count and display_text_range data from the tweet_json.txt file were stored in status_df dataframe.

2. Assessing Data

df_1 renamed as df for ease of writing data analysis and visualization commands.

Data assessment was done both visually and programatically. After reading in the data, a number of functions were used to assess the data in df:

df.head() to visually assess the top 5 rows of the dataset
df.tail() to visually assess last 5 rows of the dataset
df.info() to get an idea on the number of rows and columns, datatypes and missing values
df.sample(5) to select 5 random rows for visual assessment
df.isnull().sum() to find out total missing values for each variable

A heatmap was also created using seaborn library to see the extent of missing values in each variable.

Quality issues identified:

- df

Missing data:

- many tweet_id(s) of df table are missing in img_df (image predictions). So drop the tweet_ids that are not present in img_df
- in_reply_to_status_id, in_reply_to_user_id,retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp have a lot of missing values.
- expanded_urls also has a few missing values.

Incorrect datatype:
Timestamp should be of timestamp datatype

Inconsistent data:

- unnecessary html tags in source column of df in place of utility name e.g. Twitter for iPhone
- text column of df contains untruncated text instead of displayable text

Invalid data:

- df contains retweets and hence duplicates which need to be deleted
- rating_denominator has values other than 10 present even though rating is always given out of 10
- rating_numerator has values less than 10 even though in most cases the ratings given are greater than 10 (because "**they're good dogs Brent**.")
- rating_numerator also has some very high values (e.g. 1176) which will be considered as outliers for this project.
- name column has values starting with lowercase characters (e.g. a, an, actually, by) which are incorrect names
- some rows have more than one dog stage

Tidiness issues identified:

- doggo, floofer, pupper and puppo columns in df should be merged into one column named "stage"
- img_df table should be merged with df on tweet_id
- status_df table should be merged with df on tweet_id

- img_df

Inconsistent data:

- Inconsistent pattern in values in p1, p2, p3 variables (first letter is in upper case and sometimes in lower case)

3. Cleaning Data

a. Created a backup copy of all the three dataframes (df_clean, imf_df_copy, status_df_copy).
b. For each quality/tidiness issue, the programmatic data cleaning process was performed in 3 stages - Define, Code & Test.
c. Kept only those records in df_clean table whose tweet_id exists in img_df table
d. Kept only those rows in df_clean that are original tweets and NOT retweets (i.e. rows where retweeted_status_id column is null)
e. Dropped retweeted_status_id, retweeted_status_user_id , retweeted_status_timestamp from df_clean because they have no values now.
f. Changed datatype of timestamp variable to datetime using to_datetime function of pandas
g. Dropped in_reply_to_status_id, in_reply_to_user_id from df_clean because they do not appear to be useful in the analysis
h. Stripped all the html tags from values in source variable. Kept only the text between the tags using BeautifulSoup.get_text(). Converted source to category datatype
i. values in rating_denominator should be 10 or multiple of 10 in df_clean. Kept only those records where rating_denominator was greater than 10. Value of 11 in rating_denominator was incorrectly parsed. Changed rating_denominator from 11 to 10 in cases applicable.
j. Deleted those rows where rating_numerator has outliers.
k. Extracted and reassigned correct values to rating_numerator where rating was in decimals.
l. Created a new feature called 'rating' which is calculated as rating_numerator divided by rating_denominator. Dropped rating_numerator and rating_denominator
m. Erroneous names in name column start with lowercase letter (e.g. a, by, the). These are replaced with 'None'. There is one row with name 'O' which was parsed incorrectly. The correct name was "O'Malley". 'O' was replaced to reflect the correct name.
n. Some rows have more than one dog stage

    One tweet_id has both doggo and puppo

Nine tweet_id(s) have values present in both doggo column and pupper column.

One tweet_id has both doggo and floofer

Assigned value ‘Multiple’ for stage variable for the rows that have more than one dog stage.

o. Values in p1, p2, p3 columns in img_df_copy were in inconsistent format which were changed to capitalize the first letter.
p. Created breed column on the basis of p1 and p1_conf. If p1_conf > 0.95 and p1_dog is True, then breed is the value contained in p1.
q. Created a function to find gender based on some words in text column.

4. Tidying the data

a. doggo, floofer, pupper and puppo columns in df_clean table were merged into one column named "stage". The data type of stage column was changed to category. Later doggo, floofer, pupper and puppo columns were dropped.
b. retweet_count, favorite_count,display_text_range columns from status_df table were joined with df_clean table on the basis of tweet_id by doing inner join.
c. Using pd.merge merged p1, p1_conf, p1_dog from img_df_copy with df_clean dataset on tweet_id

5. Storing the cleaned data

Stored the cleaned data in data_clean in a csv file named twitter_archive_master.csv