PROJECT2 : WRANGLING AND ANALYZE DATA

WRANGLING DATA

I. Data Gathering

- Download twitter_archive_enhanced.cvs file and create a dataframe named **twitter_archive** with it
- Download image-predictions.tsv by using the **Requests** library
- Use the **Tweepy** library to query additional data (retweet_count and favorite_count) via the **Twitter API** , and write results in **tweet_json.txt** file
- Format content of tweet_json.txt file to JSON format, and create a dataframe named df with this file
- Extract ***id, retweet_count, favorite_count*** from ***df*** to create dataframe ***df_tweet***
- Create a dataframe named img_prediction by loading image-predictions.tsv

II. Assessing Data

- Check number of occurrences of each img_num value in table img_prediction table (with value_counts() method)
- Check type and number of entries for each column in img_prediction, df_tweet twitter_archive tables
- Display samples of rows of twitter_archive table
- Display rows of twitter_archive table where expanded_urls column is null
- Check values of in_reply_to_status_id column in twitter_archive table
- Check rows with duplicated values of tweet_id in twitter_archive table
- Check rows with duplicated values of name in twitter_archive table
- Extract some rows with specific values of name in twitter_archive table
- Check number of occurrences of values for source column in twitter_archive table
- Check values of name, rating_numerator, puppo, doggo, pupper, and floofer columns in twitter_archive table
- Check number of occurrences for values of puppo, doggo, pupper, and floofer columns in twitter_archive table
- Check duplicated id in df_tweet table
- Check duplicated columns in twitter_archive, img_prediction, df_tweet

Results of assessing data

1- Quality issues

a. tweeter_archive table - NaN (Null) values in in_reply_to_status_id, in_reply_to_user_id, retweeted_status_user_id, retweeted_status_id, retweeted_status_timestamp, expanded_urls columns

b. tweeter_archive table - Null values replaced by None in doggo, floofer, pupper, puppo; name column sometimes has value 'a'

c. tweeter_archive table - timestamp and retweeted_status_timestamp columns are object(string) type not datetime

d. tweeter_archive, df_tweet tables - some tweet_id has no image in img_prediction table

e. tweeter_archive table - in_reply_to_status_id, in_reply_to_user_id, retweeted_status_user_id, retweeted_status_id columns have float type instead of object; tweet_id column has int type, instead of object

e. img_prediction table - for image number 4, there is no properties(p4, p4_conf, p4_dog)

f. twitter_archive table - some tweet_id have dog image with multiple dog stages

g. img_prediction table - some tweet_id have not a dog img (p1_dog, p2_dog, p3_dog are False)

2. Tidiness issues

a. tweet_id column(which is in img_prediction table) is duplicated in df_tweet and twitter_archive tables

b. twitter_archive table - the dog_stage variable is hidden in column headers: doggo, floofer, pupper, puppo

III. Cleaning Data

1. Make a copy of each dataframe : twitter_archive, df_tweet, img_prediction
2. Address issue 1 : tweeter_archive: NaN (Null) values in in_reply_to_status_id, in_reply_to_user_id, retweeted_status_user_id, retweeted_status_id, retweeted_status_timestamp, expanded_urls columns
    a. Define : Drop rows where expanded_urls misses, replace all missing id by 0, replace missing retweeted_status_timestamp values by 2099-12-31 00:00:00 +0000
3. Address issue 2 : tweeter_archive: Null values replaced by None in doggo, floofer, pupper, puppo columns; name column sometimes has value a non-dog name like 'the', 'just', 'a'

   a. Define :
      - In doggo, floofer, pupper, puppo columns, replace None by 0
      - Delete all rows which have a lowercase value name(non-dog name) in name column

4. Address Issue 3 : tweeter_archive table: timestamp and retweeted_status_timestamp columns are object(string) type not datetime
   a. Define : Convert type of timestamp and retweeted_status_timestamp columns into datetime
5. Address Issue 4 : tweeter_archive, df_tweet tables: some tweet_id has no image in img_prediction
   a. Define :
      - Filter id (in df_tweet table) which exist in img_prediction table
      - Filter tweet_id (in twitter_archive table) which exist in img_prediction table
      - Delete rows and columns for retweet in twitter_archive_clean table
6. Address Issue 5 : tweet_id column(which is in img_prediction table) is duplicated in df_tweet and twitter_archive tables

a. Define : Rename id column to tweet_id in df_tweet table, merge all columns of twitter_archive and img_prediction tables to df_tweet table, joining on tweet_id

7. Address Issue 6 : in twitter_archive, some tweet_id have dog image with multiple dog stages
   a. Define :
      - Replace all values different from '0' by '1' in doggo, floofer, pupper, and puppo columns
      - Create multiple_stages column which contain list separated comma of dog stages for such tweet_id
8. Address Issue 7 : twitter_archive table: the dog_stage variable is hidden in column headers: doggo, floofer, pupper, puppo
   a. Define : Melt the *doggo, flooter, pupper, puppo* columns to a *dog_stage* column
9. Address Issue 8 : in tweeter_archive table, in_reply_to_status_id, in_reply_to_user_id, retweeted_status_user_id, retweeted_status_id columns have float type instead of object; tweet_id column has int type instead of object
   a. Define : Since retweeted_status_user_id, retweeted_status_id columns have been removed, change type of all in_reply_to_status_id, in_reply_to_user_id, tweet_id columns to object
10. Address Issue 9 : img_prediction table : for image number 4, there is no properties(p4, p4_conf, p4_dog)
    a. Define : Filter rows with img_num different from 4 in twitter_archive_master table

11. Address Issue 10 : img_prediction table: some tweet_id have not a dog img (p1_dog, p2_dog, p3_dog are False)
    a. Define : Filter rows with p1_dog, p2_dog and p3_dog are not False in same row