WeRateDogs data wrangling report

My Data wrangling efforts were divided into 3 major steps below:

1. Gather
2. Assess
3. Clean

Gather

There were three different datasets to be gathered from different sources:

- **twitter-archive-enhanced.csv**: Used read_csv() function to import this file into a pandas dataframe named **df_1**
- **image-predictions.tsv**: Used requests library to access the file (code included). Later decided to use read_csv() to directly import the file into data frame **df_2,** to save time and code.
- **TwitterAPI**: Accessed twitter's REST API using tweepy library in python. This was done in following steps:

1. Created application in twitter at https://apps.twitter.com/ to get Consumer Key (API Key), Consumer Secret (API Secret), Access Token, Access Secret.
2. Saved this information in a .txt file and used with open() function to read the information line by line for OAuthHandler.
3. Converted to json object, saved this information in a tweet_json.txt file as a string, later used json.load() and created a data frame, **df_3** with the required information.

Assess

1. For every data frame, eyeballed the dataset in the jupyter notebook.
2. Used functions like info(), value_counts(), sample() to check overall issues.
3. Also accessed few columns and cells separately to go deeper into the issues.
4. Copied all the observed issues into a single cell and expanded on all issues noting possible rectifications

Following are the issues I found:

Quality Issues

df_1: twitter_archive

1. Dtypes for tweet_id, in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id
2. timestamp and retweeted_status_timestamp is a string
3. source is in html format and not text
4. Missing values in reply and retweet columns
5. expanded_urls have multiple urls for some entries and also have whitespace for some entries
6. Some of the tweet texts contain shortened urls

7. rating numerator ranged from 0 to 1000+, some were outliers, wrong text extractions and other needed transformation
8. rating_denominator ranged from 0 to 100+, fewer issues compared to numerator

df_2 : image-predictions.tsv

9. non-descriptive column names
10. p1, p2, p3 have inconsistent alphabetic casing and contained "_" instead of space
11. tweet_id of int64 dtype
12. needed a column conclusive of the breed

df_3: twitter API data

- favorite_count had 170 columns with value 0 which seems raises questions when followers close 7million and tweets are about cute dogs
    - these were retweets or replies which will be dropped with an inner merge to df_1

Tidiness Issues

- separate tables for retweets and replies

1. doggo, pupper, floofer, puppo
2. unwanted columns
3. all three datasets need to be merged
4. column order needs to be changed

Clean

Tackled every issue one by one. For every issue, divided cleaning process into following:

1. Define
2. Code
3. Check

df_1: twitter_archive

1. Changed dtypes for tweet_id using astype(str), ignored reply and retweet columns as they would be dropped later.
2. timestamp and retweeted_status_timestamp converted to datetime type using to_datetime() function
3. Values in source column extracted and shortened using str.split() and str.replace()
4. Dropped retweets rows (duplicates) and all the retweets and replies columns
    - Created separate data frames for replies and retweets, **df_4** and **df_5**
    - Later this allowed me to create the is_reply column in **df_1** to track if tweet was a reply
5. Removed expanded_urls column and created new url column
    - Used tweet_id and common subdomain for url
6. Removed shortened urls from texts using regex and contains() function
7. I tackled issue 7 and 8 together since they both were regarding ratings.
    - Used regex to extract the ratings from text, took care of rating with decimal
    - Ddtype changed to float.

- Removed interesting outliers related to Snoop Dogg the 420/10 Dogg and Uncle Sam's dog with 1776/10 (US independence year, the dog was wearing clothes with the US flag on 4th of July).
- 3 incorrect entries changed using a dictionary.
- Divided the denominator by 10 to get a divider, used this to transform multiples for packs/litters

df_2 : image-predictions.tsv

9. Changed column names using rename()
10. Used str.replace and str.title() in p1, p2, p3 to rectify inconsistent alphabetic casing and replace "_" with spaces
11. Used astype to change tweet_id to string type
12. Used apply() with a defined function to conclude the breed

Tidiness Issues

1. Created stage column, floofer column transformed to binary values
2. Drop() unwanted columns
3. Left merged df_2 and df_3 into df_1
4. Ordered the columns as needed