wrangle_and_analyse_a_dataset

September 6, 2022

1 Wrangle and Analyse WeRateDogs Twitter Data

Udacity alx Data Analyst Nanodegree

Salami Suleiman, September 2022

1.1 Introduction

This project illustrates methods to gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it.

The dataset used in this notebook is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

1.2 Data Gathering

- In this section, we will gather three pieces of data for the data wrangling

[167]:

```
# load required libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tweepy
import json
import re

%matplotlib inline
```

[47]:

```
# load Twitter archive dataset

path = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/
  ↪59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv"
twt_archive = pd.read_csv(path)

twt_archive.head(1)
```

[47]:

```
             tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
0  892420643555336193                    NaN                  NaN

                   timestamp  \
0  2017-08-01 16:23:56 +0000

                                               source  \
0  <a href="http://twitter.com/download/iphone" r…

                                                 text  retweeted_status_id  \
0  This is Phineas. He's a mystical boy. Only eve…                  NaN

   retweeted_status_user_id retweeted_status_timestamp  \
0                       NaN                        NaN

                                        expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643…                13

   rating_denominator     name doggo floofer pupper puppo
0                  10  Phineas  None     None   None  None
```

[48]:

```
# load tweet image predictions dataset

path = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/
 ↪599fd2ad_image-predictions/image-predictions.tsv"
img_pred = pd.read_csv(path, sep = "\t")
img_pred.head(2)
```

[48]:

```
             tweet_id                                          jpg_url  \
0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg

   img_num                      p1   p1_conf  p1_dog                  p2  \
0        1  Welsh_springer_spaniel  0.465074    True              collie
1        1                 redbone  0.506826    True  miniature_pinscher

    p2_conf  p2_dog                   p3   p3_conf  p3_dog
0  0.156665    True    Shetland_sheepdog  0.061428    True
1  0.074192    True  Rhodesian_ridgeback  0.072010    True
```

[49]:

```
# load Additional data from the Twitter API from txt file

df_tweet = []
with open('tweet-json.txt') as f:
    for line in f:
        tweet = (json.loads(line))
        tweet_id = tweet['id']
        retweets_count = tweet['retweet_count']
        favorite_count = tweet['favorite_count']
        df_tweet.append({'tweet_id':tweet_id, 'retweets_count':retweets_count,␣
 ↪'favorite_count':favorite_count, })

twt_api = pd.DataFrame(df_tweet)
twt_api.head(3)
```

```
[49]:              tweet_id  retweets_count  favorite_count
      0  892420643555336193            8853           39467
      1  892177421306343426            6514           33819
      2  891815181378084864            4328           25461
```

1.3 Assessing the data

In this section, we perform visual and programatic assessment of the 3 datasets and outline our quality and tidiness issues .

We start with the visual assessments by looking at the data with pandas and excel

```
[53]: # visual assessment of Twitter archive dataset

      twt_archive.head()
```

```
[53]:              tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
      0  892420643555336193                    NaN                  NaN
      1  892177421306343426                    NaN                  NaN
      2  891815181378084864                    NaN                  NaN
      3  891689557279858688                    NaN                  NaN
      4  891327558926688256                    NaN                  NaN

                         timestamp  \
      0  2017-08-01 16:23:56 +0000
      1  2017-08-01 00:17:27 +0000
      2  2017-07-31 00:18:03 +0000
      3  2017-07-30 15:58:51 +0000
      4  2017-07-29 16:00:24 +0000

                                                    source  \
      0  <a href="http://twitter.com/download/iphone" r…
      1  <a href="http://twitter.com/download/iphone" r…
      2  <a href="http://twitter.com/download/iphone" r…
      3  <a href="http://twitter.com/download/iphone" r…
      4  <a href="http://twitter.com/download/iphone" r…

                                                      text  retweeted_status_id  \
      0  This is Phineas. He's a mystical boy. Only eve…                  NaN
      1  This is Tilly. She's just checking pup on you…                  NaN
```

```
2  This is Archie. He is a rare Norwegian Pouncin…                  NaN
3  This is Darla. She commenced a snooze mid meal…                  NaN
4  This is Franklin. He would like you to stop ca…                  NaN

   retweeted_status_user_id retweeted_status_timestamp  \
0                       NaN                        NaN
1                       NaN                        NaN
2                       NaN                        NaN
3                       NaN                        NaN
4                       NaN                        NaN

                                       expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643…                13
1  https://twitter.com/dog_rates/status/892177421…                13
2  https://twitter.com/dog_rates/status/891815181…                12
3  https://twitter.com/dog_rates/status/891689557…                13
4  https://twitter.com/dog_rates/status/891327558…                12

   rating_denominator      name doggo floofer pupper puppo
0                  10   Phineas  None    None   None  None
1                  10     Tilly  None    None   None  None
2                  10    Archie  None    None   None  None
3                  10     Darla  None    None   None  None
4                  10  Franklin  None    None   None  None
```

[54]:

```
# visual assessment of tweet image predictions dataset

img_pred.head()
```

[54]:

```
             tweet_id                                          jpg_url  \
0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2  666033412701032449  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3  666044226329800704  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4  666049248165822465  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg

   img_num                      p1   p1_conf  p1_dog                  p2  \
0        1  Welsh_springer_spaniel  0.465074    True              collie
1        1                 redbone  0.506826    True  miniature_pinscher
2        1         German_shepherd  0.596461    True            malinois
3        1     Rhodesian_ridgeback  0.408143    True             redbone
4        1      miniature_pinscher  0.560311    True          Rottweiler

    p2_conf  p2_dog                   p3   p3_conf  p3_dog
0  0.156665    True    Shetland_sheepdog  0.061428    True
1  0.074192    True  Rhodesian_ridgeback  0.072010    True
2  0.138584    True           bloodhound  0.116197    True
```

```
3  0.360687    True   miniature_pinscher  0.222752    True
4  0.243682    True             Doberman  0.154629    True
```

[55]:
```
# visual assessment of data from the Twitter API from txt file

twt_api.head()
```

[55]:
```
             tweet_id  retweets_count  favorite_count
0  892420643555336193            8853           39467
1  892177421306343426            6514           33819
2  891815181378084864            4328           25461
3  891689557279858688            8964           42908
4  891327558926688256            9774           41048
```

We begin our programtic assessment from here by using multiple approaches

[59]:
```
# assess the various data types associated with the variables

twt_archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   tweet_id                    2356 non-null   int64
 1   in_reply_to_status_id       78 non-null     float64
 2   in_reply_to_user_id         78 non-null     float64
 3   timestamp                   2356 non-null   object
 4   source                      2356 non-null   object
 5   text                        2356 non-null   object
 6   retweeted_status_id         181 non-null    float64
 7   retweeted_status_user_id    181 non-null    float64
 8   retweeted_status_timestamp  181 non-null    object
 9   expanded_urls               2297 non-null   object
 10  rating_numerator            2356 non-null   int64
 11  rating_denominator          2356 non-null   int64
 12  name                        2356 non-null   object
 13  doggo                       2356 non-null   object
 14  floofer                     2356 non-null   object
 15  pupper                      2356 non-null   object
 16  puppo                       2356 non-null   object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

[60]:
```
img_pred.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
```

```
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   tweet_id  2075 non-null   int64
 1   jpg_url   2075 non-null   object
 2   img_num   2075 non-null   int64
 3   p1        2075 non-null   object
 4   p1_conf   2075 non-null   float64
 5   p1_dog    2075 non-null   bool
 6   p2        2075 non-null   object
 7   p2_conf   2075 non-null   float64
 8   p2_dog    2075 non-null   bool
 9   p3        2075 non-null   object
 10  p3_conf   2075 non-null   float64
 11  p3_dog    2075 non-null   bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

[61]:
```
twt_api.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   tweet_id        2354 non-null   int64
 1   retweets_count  2354 non-null   int64
 2   favorite_count  2354 non-null   int64
dtypes: int64(3)
memory usage: 55.3 KB
```

[51]:
```
# check for duplicates

twt_archive.duplicated().sum()
```

[51]: 0

[43]:
```
img_pred.duplicated().sum()
```

[43]: 0

[44]:
```
twt_api.duplicated().sum()
```

[44]: 0

[63]:
```
# check for missing data

twt_archive.isna().sum()
```

[63]:

```
tweet_id                         0
in_reply_to_status_id         2278
in_reply_to_user_id           2278
timestamp                        0
source                           0
text                             0
retweeted_status_id           2175
retweeted_status_user_id      2175
retweeted_status_timestamp    2175
expanded_urls                   59
rating_numerator                 0
rating_denominator               0
name                             0
doggo                            0
floofer                          0
pupper                           0
puppo                            0
dtype: int64
```

[64]:

```
img_pred.isna().sum()
```

[64]:

```
tweet_id    0
jpg_url     0
img_num     0
p1          0
p1_conf     0
p1_dog      0
p2          0
p2_conf     0
p2_dog      0
p3          0
p3_conf     0
p3_dog      0
dtype: int64
```

[65]:

```
twt_api.isna().sum()
```

[65]:

```
tweet_id          0
retweets_count    0
favorite_count    0
dtype: int64
```

[56]:

```
# check summary stats on numeric variables

twt_archive.describe()
```

[56]:

```
           tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
count  2.356000e+03           7.800000e+01         7.800000e+01
mean   7.427716e+17           7.455079e+17         2.014171e+16
std    6.856705e+16           7.582492e+16         1.252797e+17
min    6.660209e+17           6.658147e+17         1.185634e+07
25%    6.783989e+17           6.757419e+17         3.086374e+08
50%    7.196279e+17           7.038708e+17         4.196984e+09
75%    7.993373e+17           8.257804e+17         4.196984e+09
max    8.924206e+17           8.862664e+17         8.405479e+17

       retweeted_status_id  retweeted_status_user_id  rating_numerator  \
count         1.810000e+02              1.810000e+02       2356.000000
mean          7.720400e+17              1.241698e+16         13.126486
std           6.236928e+16              9.599254e+16         45.876648
min           6.661041e+17              7.832140e+05          0.000000
25%           7.186315e+17              4.196984e+09         10.000000
50%           7.804657e+17              4.196984e+09         11.000000
75%           8.203146e+17              4.196984e+09         12.000000
max           8.874740e+17              7.874618e+17       1776.000000

       rating_denominator
count         2356.000000
mean            10.455433
std              6.745237
min              0.000000
25%             10.000000
50%             10.000000
75%             10.000000
max            170.000000
```

[58]:

```
img_pred.describe()
```

[58]:

```
           tweet_id      img_num      p1_conf       p2_conf       p3_conf
count  2.075000e+03  2075.000000  2075.000000  2.075000e+03  2.075000e+03
mean   7.384514e+17     1.203855     0.594548  1.345886e-01  6.032417e-02
std    6.785203e+16     0.561875     0.271174  1.006657e-01  5.090593e-02
min    6.660209e+17     1.000000     0.044333  1.011300e-08  1.740170e-10
25%    6.764835e+17     1.000000     0.364412  5.388625e-02  1.622240e-02
50%    7.119988e+17     1.000000     0.588230  1.181810e-01  4.944380e-02
75%    7.932034e+17     1.000000     0.843855  1.955655e-01  9.180755e-02
max    8.924206e+17     4.000000     1.000000  4.880140e-01  2.734190e-01
```

[57]:

```
twt_api.describe()
```

[57]:

```
           tweet_id  retweets_count  favorite_count
count  2.354000e+03     2354.000000     2354.000000
mean   7.426978e+17     3164.797366     8080.968564
```

```
std    6.852812e+16     5284.770364    11814.771334
min    6.660209e+17        0.000000        0.000000
25%    6.783975e+17      624.500000     1415.000000
50%    7.194596e+17     1473.500000     3603.500000
75%    7.993058e+17     3652.000000    10122.250000
max    8.924206e+17    79515.000000   132810.000000
```

1.3.1 Quality issues

`twitter_archive` table

- tweet_id is number not a string
- only keep original ratings (no retweets) that have images for analysis
- 'None' is used to represet missing data in name column and dog stage columns
- 'timestamp' should be formatted as a date
- 'expanded_urls', etc should be dropped from the data for the analysis
- numerator ratings should be formatted as floats
- incorrect dog names name column
- some ratings_numerator values have decimal
- some records have more than on dog stage

`image_predictions` table

- tweet_id is number not a string

`twitter_api_data` table

- tweet_id is number not a string

1.3.2 Tidiness issues

`twitter_archive` table

- the dog stages: doggo, floofer, pupper and puppo columns should be merged into one column

`image_predictions` table

- the image predictions table should be merged with the twitter archive

`twitter_api_data` table

- the twitter api table columns should be merged with the twitter archive

1.4 Cleaning the data

In this section, we perform data cleaning on the 3 datasets using the define-code-test framework.

We begin be making copies of the orignal data sets

[314]:

```
# Make copies of the original datasets
```

```
twt_archive_clean = twt_archive.copy()
img_pred_clean = img_pred.copy()
twt_api_clean = twt_api.copy()
```

- define: only keep original ratings (no retweets) that have images for analysis
- code:

[315]:

```
# filter out retweets using retweeted_status_user_id

twt_archive_clean = twt_archive_clean.query('retweeted_status_user_id.isnull()')
```

- test:

[316]:

```
# test

twt_archive_clean.retweeted_status_user_id.value_counts().sum()
```

[316]: 0

- define: drop 'expanded_urls' etc. column
- code:

[317]:

```
#drop columns

twt_archive_clean.drop(columns = ['expanded_urls', 'in_reply_to_status_id',␣
 ↪'in_reply_to_user_id', 'source',
                  'retweeted_status_id', 'retweeted_status_user_id',␣
 ↪'retweeted_status_timestamp'], inplace=True)
```

- test:

[251]:

```
# test

twt_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            2175 non-null   int64
 1   timestamp           2175 non-null   object
 2   text                2175 non-null   object
 3   rating_numerator    2175 non-null   int64
 4   rating_denominator  2175 non-null   int64
 5   name                2175 non-null   object
 6   doggo               2175 non-null   object
 7   floofer             2175 non-null   object
 8   pupper              2175 non-null   object
 9   puppo               2175 non-null   object
dtypes: int64(3), object(7)
memory usage: 186.9+ KB
```

- define: change tweet_id data type to string
- code:

[318]:
```
# convert tweet_id to a string

twt_archive_clean.tweet_id = twt_archive_clean.tweet_id.astype(str)
img_pred_clean.tweet_id = img_pred_clean.tweet_id.astype(str)
twt_api_clean.tweet_id = twt_api_clean.tweet_id.astype(str)
```

- test:

[319]:
```
# test

twt_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            2175 non-null   object
 1   timestamp           2175 non-null   object
 2   text                2175 non-null   object
 3   rating_numerator    2175 non-null   int64
 4   rating_denominator  2175 non-null   int64
 5   name                2175 non-null   object
 6   doggo               2175 non-null   object
 7   floofer             2175 non-null   object
 8   pupper              2175 non-null   object
 9   puppo               2175 non-null   object
dtypes: int64(2), object(8)
memory usage: 186.9+ KB
```

- define: change timestamp to datetime
- code:

[320]:
```
# convert timestamp to datetime

twt_archive_clean.timestamp = pd.to_datetime(twt_archive_clean.timestamp)
```

- test:

[255]:
```
# test

twt_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            2175 non-null   object
 1   timestamp           2175 non-null   datetime64[ns, UTC]
 2   text                2175 non-null   object
 3   rating_numerator    2175 non-null   int64
 4   rating_denominator  2175 non-null   int64
 5   name                2175 non-null   object
 6   doggo               2175 non-null   object
 7   floofer             2175 non-null   object
 8   pupper              2175 non-null   object
 9   puppo               2175 non-null   object
dtypes: datetime64[ns, UTC](1), int64(2), object(7)
memory usage: 186.9+ KB
```

- define: fix incorrect dog names and set to NA
- code:

[321]:

```
import warnings
warnings.filterwarnings('ignore') # disable warnings from computation

# remove all improper dog names and replace with NA

twt_archive_clean.name = twt_archive_clean.name.str.replace('^[a-z]', 'None' )
twt_archive_clean.loc[twt_archive_clean['name'] == 'None']= np.NaN
```

- test:

[322]:

```
# test

twt_archive_clean.name.value_counts()
```

[322]:

```
Lucy          11
Charlie       11
Cooper        10
Oliver        10
Tucker         9
              ..
Wishes         1
Rose           1
Theo           1
Fido           1
Christoper     1
Name: name, Length: 953, dtype: int64
```

[323]:
```
twt_archive_clean.name.isna().sum()
```

[323]:
```
735
```

- define: fix numerator ratings with decimals
- code:

[324]:
```
decimal_numerators = []
for i, text in twt_archive_clean['text'].iteritems():
    if bool(re.search('\d+\.\d+\/\d+', str(text))):
        decimal_numerators.append({twt_archive_clean['tweet_id'][i]:[i, text,␣
 ↪twt_archive_clean['rating_numerator'][i]]})

decimal_numerators
```

[324]:
```
[{'883482846933004288': [45,
   'This is Bella. She hopes her smile made you smile. If not, she is also
offering you her favorite monkey. 13.5/10 https://t.co/qjrljjt948',
   5.0]},
 {'786709082849828864': [695,
   "This is Logan, the Chow who lived. He solemnly swears he's up to lots of
good. H*ckin magical af 9.75/10 https://t.co/yBO5wuqaPS",
   75.0]},
 {'778027034220126208': [763,
   "This is Sophie. She's a Jubilant Bush Pupper. Super h*ckin rare. Appears at
random just to smile at the locals. 11.27/10 would smile back
https://t.co/QFaUiIHxHq",
   27.0]}]
```

[327]:
```
# change values

twt_archive_clean.at[45,'rating_numerator'] = 13.5
twt_archive_clean.at[695,'rating_numerator'] = 9.75
twt_archive_clean.at[763,'rating_numerator'] = 11.27
```

- test:

[330]:
```
# test

decimal_numerators = []
for i, text in twt_archive_clean['text'].iteritems():
    if bool(re.search('\d+\.\d+\/\d+', str(text))):
        decimal_numerators.append({twt_archive_clean['tweet_id'][i]:[text,␣
 ↪twt_archive_clean['rating_numerator'][i]]})

decimal_numerators
```

[330]:

```
[{'883482846933004288': ['This is Bella. She hopes her smile made you smile. If
not, she is also offering you her favorite monkey. 13.5/10
https://t.co/qjrljjt948',
   13.5]},
 {'786709082849828864': ["This is Logan, the Chow who lived. He solemnly swears
he's up to lots of good. H*ckin magical af 9.75/10 https://t.co/yBO5wuqaPS",
   9.75]},
 {'778027034220126208': ["This is Sophie. She's a Jubilant Bush Pupper. Super
h*ckin rare. Appears at random just to smile at the locals. 11.27/10 would smile
back https://t.co/QFaUiIHxHq",
   11.27]}]
```

- define: change numerator and denominator ratings to float
- code:

[331]:

```
# convert to float datatype
twt_archive_clean[['rating_numerator', 'rating_denominator']] =␣
↪twt_archive_clean[['rating_numerator','rating_denominator']].astype(float)
```

- test:

[263]:

```
#test

twt_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            1440 non-null   object
 1   timestamp           1440 non-null   datetime64[ns, UTC]
 2   text                1440 non-null   object
 3   rating_numerator    1440 non-null   float64
 4   rating_denominator  1440 non-null   float64
 5   name                1440 non-null   object
 6   doggo               1440 non-null   object
 7   floofer             1440 non-null   object
 8   pupper              1440 non-null   object
 9   puppo               1440 non-null   object
dtypes: datetime64[ns, UTC](1), float64(2), object(7)
memory usage: 251.5+ KB
```

- define: Melt the doggo, floofer, pupper, puppo columns to a dog_stage column.
- code:

[333]:

```
twt_archive_clean = pd.melt(twt_archive_clean, id_vars=['tweet_id',␣
↪'timestamp', 'text', 'rating_numerator', 'rating_denominator', 'name'],
                           var_name='dog_stager', value_name='dog_stage')
twt_archive_clean = twt_archive_clean.drop('dog_stager', axis=1)
```

- test:

[265]:
```
# test

twt_archive_clean.head()
```

[265]:
```
             tweet_id                  timestamp  \
0  892420643555336193 2017-08-01 16:23:56+00:00
1  892177421306343426 2017-08-01 00:17:27+00:00
2  891815181378084864 2017-07-31 00:18:03+00:00
3  891689557279858688 2017-07-30 15:58:51+00:00
4  891327558926688256 2017-07-29 16:00:24+00:00

                                                text  rating_numerator  \
0  This is Phineas. He's a mystical boy. Only eve…              13.0
1  This is Tilly. She's just checking pup on you…             13.0
2  This is Archie. He is a rare Norwegian Pouncin…              12.0
3  This is Darla. She commenced a snooze mid meal…              13.0
4  This is Franklin. He would like you to stop ca…              12.0

   rating_denominator      name dog_stage
0                10.0   Phineas      None
1                10.0     Tilly      None
2                10.0    Archie      None
3                10.0     Darla      None
4                10.0  Franklin      None
```

[266]:
```
twt_archive_clean.dog_stage.value_counts()
```

[266]:
```
None       5561
pupper      133
doggo        45
puppo        16
floofer       5
Name: dog_stage, dtype: int64
```

- define: remove duplicated rows
- code:

[334]:
```
twt_archive_clean.duplicated().sum()
```

[334]:
```
7060
```

- test:

[335]:

```
twt_archive_clean.shape
```

[335]:

```
(8700, 7)
```

[336]:

```
# test
twt_archive_clean.drop_duplicates(inplace=True)
twt_archive_clean.shape
```

[336]:

```
(1640, 7)
```

- define: convert dog_stage to category
- code:

[337]:

```
# convert to category datatype
twt_archive_clean.dog_stage = twt_archive_clean.dog_stage.astype('category')
```

- test:

[272]:

```
twt_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1640 entries, 0 to 7430
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            1639 non-null   object
 1   timestamp           1639 non-null   datetime64[ns, UTC]
 2   text                1639 non-null   object
 3   rating_numerator    1639 non-null   float64
 4   rating_denominator  1639 non-null   float64
 5   name                1639 non-null   object
 6   dog_stage           1639 non-null   category
dtypes: category(1), datetime64[ns, UTC](1), float64(2), object(3)
memory usage: 91.5+ KB
```

- define: merge image prediction and twitter api datasets to twitter archive
- code:

[339]:

```
twt_archive_clean = pd.merge(left=twt_archive_clean, right=img_pred_clean,␣
 ↪how='left', on='tweet_id')
twt_archive_clean = pd.merge(left=twt_archive_clean, right=twt_api_clean,␣
 ↪how='left', on='tweet_id')
```

- test:

[340]:

```
# test

twt_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1640 entries, 0 to 1639
Data columns (total 20 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            1639 non-null   object
 1   timestamp           1639 non-null   datetime64[ns, UTC]
 2   text                1639 non-null   object
 3   rating_numerator    1639 non-null   float64
 4   rating_denominator  1639 non-null   float64
 5   name                1639 non-null   object
 6   dog_stage           1639 non-null   category
 7   jpg_url             1583 non-null   object
 8   img_num             1583 non-null   float64
 9   p1                  1583 non-null   object
 10  p1_conf             1583 non-null   float64
 11  p1_dog              1583 non-null   object
 12  p2                  1583 non-null   object
 13  p2_conf             1583 non-null   float64
 14  p2_dog              1583 non-null   object
 15  p3                  1583 non-null   object
 16  p3_conf             1583 non-null   float64
 17  p3_dog              1583 non-null   object
 18  retweets_count      1639 non-null   float64
 19  favorite_count      1639 non-null   float64
dtypes: category(1), datetime64[ns, UTC](1), float64(8), object(10)
memory usage: 258.1+ KB
```

- define: remove missing values
- code

[341]:

```
twt_archive_clean.isna().sum()
```

[341]:

```
tweet_id               1
timestamp              1
text                   1
rating_numerator       1
rating_denominator     1
name                   1
dog_stage              1
jpg_url               57
img_num               57
p1                    57
p1_conf               57
p1_dog                57
p2                    57
p2_conf               57
```

```
p2_dog                57
p3                    57
p3_conf               57
p3_dog                57
retweets_count         1
favorite_count         1
dtype: int64
```

[342]:

```
twt_archive_clean.dropna(axis = 0, inplace=True)
```

- test

[343]:

```
# test

twt_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1583 entries, 0 to 1639
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   tweet_id             1583 non-null   object
 1   timestamp            1583 non-null   datetime64[ns, UTC]
 2   text                 1583 non-null   object
 3   rating_numerator     1583 non-null   float64
 4   rating_denominator   1583 non-null   float64
 5   name                 1583 non-null   object
 6   dog_stage            1583 non-null   category
 7   jpg_url              1583 non-null   object
 8   img_num              1583 non-null   float64
 9   p1                   1583 non-null   object
 10  p1_conf              1583 non-null   float64
 11  p1_dog               1583 non-null   object
 12  p2                   1583 non-null   object
 13  p2_conf              1583 non-null   float64
 14  p2_dog               1583 non-null   object
 15  p3                   1583 non-null   object
 16  p3_conf              1583 non-null   float64
 17  p3_dog               1583 non-null   object
 18  retweets_count       1583 non-null   float64
 19  favorite_count       1583 non-null   float64
dtypes: category(1), datetime64[ns, UTC](1), float64(8), object(10)
memory usage: 249.1+ KB
```

1.5 Save cleaned data

[344]:

```
twt_archive_clean.to_csv('twitter_archive_master.csv', index=False)
```

[ ]:

[ ]: