wrangle_act

April 7, 2019

1 Data wrangling

By Wessam Alhallak

1.1 Table of Contents

## Introduction
wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information. Additional gathering, then assessing and cleaning is required for "Wow!"-worthy analyses and visualizations.

## The Data ### Enhanced Twitter Archive The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. One column the archive does contain though: each tweet's text, which I used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced." Of the 5000+ tweets, I have filtered for tweets with ratings only (there are 2356).

1.1.1 Additional Data via the Twitter API

Back to the basic-ness of Twitter archives: retweet count and favorite count are two of the notable column omissions. Fortunately, this additional data can be gathered by anyone from Twitter's API. Well, "anyone" who has access to data for the 3000 most recent tweets, at least. But you, because you have the WeRateDogs Twitter archive and specifically the tweet IDs within it, can gather this data for all 5000+. And guess what? You're going to query Twitter's API to gather this valuable data.

1.1.2 Image Predictions File

One more cool thing: I ran every image in the WeRateDogs Twitter archive through a neural network that can classify breeds of dogs*. The results: a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images).

```
In [1]: #import packages
        import pandas as pd
        import requests
        import os as os
        import json
        import matplotlib.pyplot as plt
        import tweepy
        from timeit import default_timer as timer
        import ast
        import re
        import seaborn as sns
```

## Gathring Data

1.1.3 Reading The CSV from local file system

```
In [2]: df_csv=pd.read_csv("twitter-archive-enhanced.csv")
```

1.1.4 Test

```
In [3]: df_csv.head()
```

```
Out[3]:              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
```

1.1.5 Reading The img Data file

```
In [4]: imgpre_url="https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-pre
```

```
In [5]: imgpre_filename=imgpre_url.split("/")[-1]
```

```
In [6]: response=requests.get(imgpre_url)
```

1.1.6 Saving The img Data File

```
In [7]: with open(imgpre_filename,mode="wb") as file:
            file.write(response.content)
```

```
In [8]: df_img=pd.read_csv(imgpre_filename,sep="\t")
```

1.1.7 Test

```
In [9]: df_img.head()
```

```
Out[9]:              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-1Eu.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
```

1.2 Prapring the api creditial from json file

```
In [10]: #you need to write your api in the attached json file"

         if (0):
             credintials=pd.read_json("TweeterApiToken.json")
             credintials.head()

             consumer_key = credintials.tokens.APIkey
             consumer_secret = credintials.tokens.APISecretkey
             access_token = credintials.tokens.AccessToken
             access_secret = credintials.tokens.AccessTokenSecret

             auth=tweepy.OAuthHandler(consumer_key,consumer_secret)
             auth.set_access_token(access_token,access_secret)

             api=tweepy.API(auth,wait_on_rate_limit=True,wait_on_rate_limit_notify=True)
```

```
In [11]: # you need to change 0 to 1 if you want to start reading the tweets
         if (0) :
             start = timer()
             failer={}
             Readinglog=""
             n=0
             with open ("tweet_json.txt",mode="w") as file:
                 for tweetid in df_csv.tweet_id:
                     n=n+1
                     Readinglog+=(str(n)+" "+str(tweetid)+" ")
                     print ( n," : ",timer() )
                     try:
                         temp=api.get_status(tweetid, tweet_mode='extended')._json
                         Readinglog+="Success "
                         json.dump(temp,file)
                         file.writelines("\n")
                     except tweepy.TweepError as e:
                         Readinglog+="Faild "
                         failer[tweetid]=e
                         pass
                     #print(temp["favorite"])#favorite
                     Readinglog+="\n"
                     if n == 10000:
                         break
             end = timer()
             print ( end - start)

             start = timer()
             failer={}
             Readinglog1=""
             n=0
             with open ("tweet_json1.txt",mode="w") as file:
                 for tweetid in df_csv.tweet_id:
                     n=n+1
                     Readinglog1+=(str(n)+" "+str(tweetid)+" ")
                     print ( n," : ",timer() )
                     try:
                         temp=api.get_status(tweetid, tweet_mode='extended')._json
                         Readinglog1+="Success "
                         json.dump(temp,file)
                         file.writelines("\n")
                     except tweepy.TweepError as e:
                         Readinglog1+="Faild "
                         failer[tweetid]=e
                         pass
                     #print(temp["favorite"])#favorite
                     Readinglog1+="\n"
                     if n == 10:
                         break

             end = timer()
             print ( end - start)
```

```
In [12]: #https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object.h
         with open ("tweet_json1.txt",mode="r") as file:
             lines=file.readlines()
```

```
In [13]: json.loads(lines[3])
```

```
Out[13]: {'created_at': 'Sun Jul 30 15:58:51 +0000 2017',
          'id': 891689557279858688,
          'id_str': '891689557279858688',
          'full_text': 'This is Darla. She commenced a snooze mid meal. 13/10 happens to the be
          'truncated': False,
          'display_text_range': [0, 79],
          'entities': {'hashtags': [],
           'symbols': [],
           'user_mentions': [],
           'urls': [],
           'media': [{'id': 891689552724799489,
             'id_str': '891689552724799489',
             'indices': [80, 103],
             'media_url': 'http://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
             'media_url_https': 'https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
             'url': 'https://t.co/tD36da7qLQ',
             'display_url': 'pic.twitter.com/tD36da7qLQ',
             'expanded_url': 'https://twitter.com/dog_rates/status/891689557279858688/photo/1'
             'type': 'photo',
             'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
              'small': {'w': 510, 'h': 680, 'resize': 'fit'},
              'medium': {'w': 901, 'h': 1200, 'resize': 'fit'},
              'large': {'w': 1201, 'h': 1600, 'resize': 'fit'}}}]},
          'extended_entities': {'media': [{'id': 891689552724799489,
             'id_str': '891689552724799489',
             'indices': [80, 103],
             'media_url': 'http://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
             'media_url_https': 'https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
             'url': 'https://t.co/tD36da7qLQ',
             'display_url': 'pic.twitter.com/tD36da7qLQ',
             'expanded_url': 'https://twitter.com/dog_rates/status/891689557279858688/photo/1'
             'type': 'photo',
             'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
              'small': {'w': 510, 'h': 680, 'resize': 'fit'},
              'medium': {'w': 901, 'h': 1200, 'resize': 'fit'},
              'large': {'w': 1201, 'h': 1600, 'resize': 'fit'}}}]},
          'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for i
          'in_reply_to_status_id': None,
          'in_reply_to_status_id_str': None,
          'in_reply_to_user_id': None,
          'in_reply_to_user_id_str': None,
          'in_reply_to_screen_name': None,
          'user': {'id': 4196983835,
           'id_str': '4196983835',
           'name': 'WeRateDogs',
           'screen_name': 'dog_rates',
           'location': ' DM YOUR DOGS ',
           'description': 'Your Only Source For Professional Dog Ratings Instagram and Faceboo
```

```
 'url': 'https://t.co/N7sNNHAEXS',
 'entities': {'url': {'urls': [{'url': 'https://t.co/N7sNNHAEXS',
     'expanded_url': 'http://weratedogs.com',
     'display_url': 'weratedogs.com',
     'indices': [0, 23]}]},
  'description': {'urls': []}},
 'protected': False,
 'followers_count': 7896527,
 'friends_count': 12,
 'listed_count': 6059,
 'created_at': 'Sun Nov 15 21:41:29 +0000 2015',
 'favourites_count': 141345,
 'utc_offset': None,
 'time_zone': None,
 'geo_enabled': True,
 'verified': True,
 'statuses_count': 9960,
 'lang': 'en',
 'contributors_enabled': False,
 'is_translator': False,
 'is_translation_enabled': False,
 'profile_background_color': '000000',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg
 'profile_background_tile': False,
 'profile_image_url': 'http://pbs.twimg.com/profile_images/1110029608794161152/2SIlO
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/111002960879416115
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4196983835/1553486409'
 'profile_link_color': 'F5ABB5',
 'profile_sidebar_border_color': '000000',
 'profile_sidebar_fill_color': '000000',
 'profile_text_color': '000000',
 'profile_use_background_image': False,
 'has_extended_profile': False,
 'default_profile': False,
 'default_profile_image': False,
 'following': False,
 'follow_request_sent': False,
 'notifications': False,
 'translator_type': 'none'},
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 8375,
'favorite_count': 41040,
'favorited': False,
          'retweeted': False,
          'possibly_sensitive': False,
          'possibly_sensitive_appealable': False,
          'lang': 'en'}
```

```
In [14]: with open ("tweet_json.txt",mode="r") as file:
             lines=file.readlines()
```

```
In [15]: df_list=[]
         for line in lines:
             line=json.loads(line)
             df_list.append({
                 "tweet_id":line["id"],
                 "favorite_count":line["favorite_count"],
                 "retweet_count":line["retweet_count"],
                    })
```

```
In [16]: df_api=pd.DataFrame(df_list)
```

1.2.1 Test

```
In [17]: df_api.head()
```

```
Out[17]:    favorite_count  retweet_count            tweet_id
         0           37731           8221  892420643555336193
         1           32404           6077  892177421306343426
         2           24401           4022  891815181378084864
         3           41040           8376  891689557279858688
         4           39238           9079  891327558926688256
```

## Assessing

```
In [18]: df_csv.head(50)
```

```
Out[18]:               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
         5   891087950875897856                    NaN                  NaN
         6   890971913173991426                    NaN                  NaN
         7   890729181411237888                    NaN                  NaN
         8   890609185150312448                    NaN                  NaN
         9   890240255349198849                    NaN                  NaN
         10  890006608113172480                    NaN                  NaN
         11  889880896479866881                    NaN                  NaN
         12  889665388333682689                    NaN                  NaN
         13  889638837579907072                    NaN                  NaN
```

```
14  889531135344209921                    NaN                  NaN
15  889278841981685760                    NaN                  NaN
16  888917238123831296                    NaN                  NaN
17  888804989199671297                    NaN                  NaN
18  888554962724278272                    NaN                  NaN
19  888202515573088257                    NaN                  NaN
20  888078434458587136                    NaN                  NaN
21  887705289381826560                    NaN                  NaN
22  887517139158093824                    NaN                  NaN
23  887473957103951883                    NaN                  NaN
24  887343217045368832                    NaN                  NaN
25  887101392804085760                    NaN                  NaN
26  886983233522544640                    NaN                  NaN
27  886736880519319552                    NaN                  NaN
28  886680336477933568                    NaN                  NaN
29  886366144734445568                    NaN                  NaN
30  886267009285017600           8.862664e+17         2.281182e+09
31  886258384151887873                    NaN                  NaN
32  886054160059072513                    NaN                  NaN
33  885984800019947520                    NaN                  NaN
34  885528943205470208                    NaN                  NaN
35  885518971528720385                    NaN                  NaN
36  885311592912609280                    NaN                  NaN
37  885167619883638784                    NaN                  NaN
38  884925521741709313                    NaN                  NaN
39  884876753390489601                    NaN                  NaN
40  884562892145688576                    NaN                  NaN
41  884441805382717440                    NaN                  NaN
42  884247878851493888                    NaN                  NaN
43  884162670584377345                    NaN                  NaN
44  883838122936631299                    NaN                  NaN
45  883482846933004288                    NaN                  NaN
46  883360690899218434                    NaN                  NaN
47  883117836046086144                    NaN                  NaN
48  882992080364220416                    NaN                  NaN
49  882762694511734784                    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
5   2017-07-29 00:08:17 +0000
6   2017-07-28 16:27:12 +0000
7   2017-07-28 00:22:40 +0000
8   2017-07-27 16:25:51 +0000
9   2017-07-26 15:59:51 +0000
```

```
10  2017-07-26 00:31:25 +0000
11  2017-07-25 16:11:53 +0000
12  2017-07-25 01:55:32 +0000
13  2017-07-25 00:10:02 +0000
14  2017-07-24 17:02:04 +0000
15  2017-07-24 00:19:32 +0000
16  2017-07-23 00:22:39 +0000
17  2017-07-22 16:56:37 +0000
18  2017-07-22 00:23:06 +0000
19  2017-07-21 01:02:36 +0000
20  2017-07-20 16:49:33 +0000
21  2017-07-19 16:06:48 +0000
22  2017-07-19 03:39:09 +0000
23  2017-07-19 00:47:34 +0000
24  2017-07-18 16:08:03 +0000
25  2017-07-18 00:07:08 +0000
26  2017-07-17 16:17:36 +0000
27  2017-07-16 23:58:41 +0000
28  2017-07-16 20:14:00 +0000
29  2017-07-15 23:25:31 +0000
30  2017-07-15 16:51:35 +0000
31  2017-07-15 16:17:19 +0000
32  2017-07-15 02:45:48 +0000
33  2017-07-14 22:10:11 +0000
34  2017-07-13 15:58:47 +0000
35  2017-07-13 15:19:09 +0000
36  2017-07-13 01:35:06 +0000
37  2017-07-12 16:03:00 +0000
38  2017-07-12 00:01:00 +0000
39  2017-07-11 20:47:12 +0000
40  2017-07-11 00:00:02 +0000
41  2017-07-10 15:58:53 +0000
42  2017-07-10 03:08:17 +0000
43  2017-07-09 21:29:42 +0000
44  2017-07-09 00:00:04 +0000
45  2017-07-08 00:28:19 +0000
46  2017-07-07 16:22:55 +0000
47  2017-07-07 00:17:54 +0000
48  2017-07-06 15:58:11 +0000
49  2017-07-06 00:46:41 +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...
5   <a href="http://twitter.com/download/iphone" r...
```

```
6   <a href="http://twitter.com/download/iphone" r...
7   <a href="http://twitter.com/download/iphone" r...
8   <a href="http://twitter.com/download/iphone" r...
9   <a href="http://twitter.com/download/iphone" r...
10  <a href="http://twitter.com/download/iphone" r...
11  <a href="http://twitter.com/download/iphone" r...
12  <a href="http://twitter.com/download/iphone" r...
13  <a href="http://twitter.com/download/iphone" r...
14  <a href="http://twitter.com/download/iphone" r...
15  <a href="http://twitter.com/download/iphone" r...
16  <a href="http://twitter.com/download/iphone" r...
17  <a href="http://twitter.com/download/iphone" r...
18  <a href="http://twitter.com/download/iphone" r...
19  <a href="http://twitter.com/download/iphone" r...
20  <a href="http://twitter.com/download/iphone" r...
21  <a href="http://twitter.com/download/iphone" r...
22  <a href="http://twitter.com/download/iphone" r...
23  <a href="http://twitter.com/download/iphone" r...
24  <a href="http://twitter.com/download/iphone" r...
25  <a href="http://twitter.com/download/iphone" r...
26  <a href="http://twitter.com/download/iphone" r...
27  <a href="http://twitter.com/download/iphone" r...
28  <a href="http://twitter.com/download/iphone" r...
29  <a href="http://twitter.com/download/iphone" r...
30  <a href="http://twitter.com/download/iphone" r...
31  <a href="http://twitter.com/download/iphone" r...
32  <a href="http://twitter.com/download/iphone" r...
33  <a href="http://twitter.com/download/iphone" r...
34  <a href="http://twitter.com/download/iphone" r...
35  <a href="http://twitter.com/download/iphone" r...
36  <a href="http://twitter.com/download/iphone" r...
37  <a href="http://twitter.com/download/iphone" r...
38  <a href="http://twitter.com/download/iphone" r...
39  <a href="http://twitter.com/download/iphone" r...
40  <a href="http://twitter.com/download/iphone" r...
41  <a href="http://twitter.com/download/iphone" r...
42  <a href="http://twitter.com/download/iphone" r...
43  <a href="http://twitter.com/download/iphone" r...
44  <a href="http://twitter.com/download/iphone" r...
45  <a href="http://twitter.com/download/iphone" r...
46  <a href="http://twitter.com/download/iphone" r...
47  <a href="http://twitter.com/download/iphone" r...
48  <a href="http://twitter.com/download/iphone" r...
49  <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
5   Here we have a majestic great white breaching ...                  NaN
6   Meet Jax. He enjoys ice cream so much he gets ...                  NaN
7   When you watch your owner call another dog a g...                  NaN
8   This is Zoey. She doesn't want to be one of th...                  NaN
9   This is Cassie. She is a college pup. Studying...                  NaN
10  This is Koda. He is a South Australian decksha...                  NaN
11  This is Bruno. He is a service shark. Only get...                  NaN
12  Here's a puppo that seems to be on the fence a...                  NaN
13  This is Ted. He does his best. Sometimes that'...                  NaN
14  This is Stuart. He's sporting his favorite fan...                  NaN
15  This is Oliver. You're witnessing one of his m...                  NaN
16  This is Jim. He found a fren. Taught him how t...                  NaN
17  This is Zeke. He has a new stick. Very proud o...                  NaN
18  This is Ralphus. He's powering up. Attempting ...                  NaN
19  RT @dog_rates: This is Canela. She attempted s...         8.874740e+17
20  This is Gerald. He was just told he didn't get...                  NaN
21  This is Jeffrey. He has a monopoly on the pool...                  NaN
22  I've yet to rate a Venezuelan Hover Wiener. Th...                  NaN
23  This is Canela. She attempted some fancy porch...                  NaN
24  You may not have known you needed to see this ...                  NaN
25  This... is a Jubilant Antarctic House Bear. We...                  NaN
26  This is Maya. She's very shy. Rarely leaves he...                  NaN
27  This is Mingus. He's a wonderful father to his...                  NaN
28  This is Derek. He's late for a dog meeting. 13...                  NaN
29  This is Roscoe. Another pupper fallen victim t...                  NaN
30  @NonWhiteHat @MayhewMayhem omg hello tanner yo...                  NaN
31  This is Waffles. His doggles are pupside down...                  NaN
32  RT @Athletics: 12/10 #BATP https://t.co/WxwJmv...         8.860537e+17
33  Viewer discretion advised. This is Jimbo. He w...                  NaN
34  This is Maisey. She fell asleep mid-excavation...                  NaN
35  I have a new hero and his name is Howard. 14/1...                  NaN
36  RT @dog_rates: This is Lilly. She just paralle...         8.305833e+17
37  Here we have a corgi undercover as a malamute...                  NaN
38  This is Earl. He found a hat. Nervous about wh...                  NaN
39  This is Lola. It's her first time outside. Mus...                  NaN
40  This is Kevin. He's just so happy. 13/10 what ...                  NaN
41  I present to you, Pup in Hat. Pup in Hat is gr...                  NaN
42  OMG HE DIDN'T MEAN TO HE WAS JUST TRYING A LIT...                  NaN
43  Meet Yogi. He doesn't have any important dog m...                  NaN
44  This is Noah. He can't believe someone made th...                  NaN
45  This is Bella. She hopes her smile made you sm...                  NaN
46  Meet Grizzwald. He may be the floofiest floofe...                  NaN
47  Please only send dogs. We don't rate mechanics...                  NaN
48  This is Rusty. He wasn't ready for the first p...                  NaN
49  This is Gus. He's quite the cheeky pupper. Alr...                  NaN
```

```
    retweeted_status_user_id retweeted_status_timestamp  \
0                        NaN                        NaN
1                        NaN                        NaN
2                        NaN                        NaN
3                        NaN                        NaN
4                        NaN                        NaN
5                        NaN                        NaN
6                        NaN                        NaN
7                        NaN                        NaN
8                        NaN                        NaN
9                        NaN                        NaN
10                       NaN                        NaN
11                       NaN                        NaN
12                       NaN                        NaN
13                       NaN                        NaN
14                       NaN                        NaN
15                       NaN                        NaN
16                       NaN                        NaN
17                       NaN                        NaN
18                       NaN                        NaN
19              4.196984e+09  2017-07-19 00:47:34 +0000
20                       NaN                        NaN
21                       NaN                        NaN
22                       NaN                        NaN
23                       NaN                        NaN
24                       NaN                        NaN
25                       NaN                        NaN
26                       NaN                        NaN
27                       NaN                        NaN
28                       NaN                        NaN
29                       NaN                        NaN
30                       NaN                        NaN
31                       NaN                        NaN
32              1.960740e+07  2017-07-15 02:44:07 +0000
33                       NaN                        NaN
34                       NaN                        NaN
35                       NaN                        NaN
36              4.196984e+09  2017-02-12 01:04:29 +0000
37                       NaN                        NaN
38                       NaN                        NaN
39                       NaN                        NaN
40                       NaN                        NaN
41                       NaN                        NaN
42                       NaN                        NaN
43                       NaN                        NaN
44                       NaN                        NaN
45                       NaN                        NaN
```

```
46                       NaN                        NaN
47                       NaN                        NaN
48                       NaN                        NaN
49                       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
5   https://twitter.com/dog_rates/status/891087950...                13
6   https://gofundme.com/ydvmve-surgery-for-jax,ht...                13
7   https://twitter.com/dog_rates/status/890729181...                13
8   https://twitter.com/dog_rates/status/890609185...                13
9   https://twitter.com/dog_rates/status/890240255...                14
10  https://twitter.com/dog_rates/status/890006608...                13
11  https://twitter.com/dog_rates/status/889880896...                13
12  https://twitter.com/dog_rates/status/889665388...                13
13  https://twitter.com/dog_rates/status/889638837...                12
14  https://twitter.com/dog_rates/status/889531135...                13
15  https://twitter.com/dog_rates/status/889278841...                13
16  https://twitter.com/dog_rates/status/888917238...                12
17  https://twitter.com/dog_rates/status/888804989...                13
18  https://twitter.com/dog_rates/status/888554962...                13
19  https://twitter.com/dog_rates/status/887473957...                13
20  https://twitter.com/dog_rates/status/888078434...                12
21  https://twitter.com/dog_rates/status/887705289...                13
22  https://twitter.com/dog_rates/status/887517139...                14
23  https://twitter.com/dog_rates/status/887473957...                13
24  https://twitter.com/dog_rates/status/887343217...                13
25  https://twitter.com/dog_rates/status/887101392...                12
26  https://twitter.com/dog_rates/status/886983233...                13
27  https://www.gofundme.com/mingusneedsus,https:/...                13
28  https://twitter.com/dog_rates/status/886680336...                13
29  https://twitter.com/dog_rates/status/886366144...                12
30                                                NaN                12
31  https://twitter.com/dog_rates/status/886258384...                13
32  https://twitter.com/dog_rates/status/886053434...                12
33  https://twitter.com/dog_rates/status/885984800...                12
34  https://twitter.com/dog_rates/status/885528943...                13
35  https://twitter.com/4bonds2carbon/status/88551...                14
36  https://twitter.com/dog_rates/status/830583320...                13
37  https://twitter.com/dog_rates/status/885167619...                13
38  https://twitter.com/dog_rates/status/884925521...                12
39  https://twitter.com/dog_rates/status/884876753...                13
40  https://twitter.com/dog_rates/status/884562892...                13
41  https://twitter.com/dog_rates/status/884441805...                14
```

```
42  https://twitter.com/kaijohnson_19/status/88396...                13
43  https://twitter.com/dog_rates/status/884162670...                12
44  https://twitter.com/dog_rates/status/883838122...                12
45  https://twitter.com/dog_rates/status/883482846...                 5
46  https://twitter.com/dog_rates/status/883360690...                13
47  https://twitter.com/dog_rates/status/883117836...                13
48  https://twitter.com/dog_rates/status/882992080...                13
49  https://twitter.com/dog_rates/status/882762694...                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
5                   10       None   None     None    None   None
6                   10        Jax   None     None    None   None
7                   10       None   None     None    None   None
8                   10       Zoey   None     None    None   None
9                   10     Cassie  doggo     None    None   None
10                  10       Koda   None     None    None   None
11                  10      Bruno   None     None    None   None
12                  10       None   None     None    None  puppo
13                  10        Ted   None     None    None   None
14                  10     Stuart   None     None    None  puppo
15                  10     Oliver   None     None    None   None
16                  10        Jim   None     None    None   None
17                  10       Zeke   None     None    None   None
18                  10    Ralphus   None     None    None   None
19                  10     Canela   None     None    None   None
20                  10     Gerald   None     None    None   None
21                  10    Jeffrey   None     None    None   None
22                  10       such   None     None    None   None
23                  10     Canela   None     None    None   None
24                  10       None   None     None    None   None
25                  10       None   None     None    None   None
26                  10       Maya   None     None    None   None
27                  10     Mingus   None     None    None   None
28                  10      Derek   None     None    None   None
29                  10     Roscoe   None     None  pupper   None
30                  10       None   None     None    None   None
31                  10    Waffles   None     None    None   None
32                  10       None   None     None    None   None
33                  10      Jimbo   None     None    None   None
34                  10     Maisey   None     None    None   None
35                  10       None   None     None    None   None
36                  10      Lilly   None     None    None   None
37                  10       None   None     None    None   None
```

```
38                  10        Earl  None     None    None  None
39                  10        Lola  None     None    None  None
40                  10       Kevin  None     None    None  None
41                  10        None  None     None    None  None
42                  10        None  None     None    None  None
43                  10        Yogi doggo     None    None  None
44                  10        Noah  None     None    None  None
45                  10       Bella  None     None    None  None
46                  10   Grizzwald  None  floofer    None  None
47                  10        None  None     None    None  None
48                  10       Rusty  None     None    None  None
49                  10         Gus  None     None  pupper  None
```

```
In [19]: df_img.head()
```

```
Out[19]:            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
```

```
In [20]: df_img.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
```

```
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

```
In [21]: df_api.head()
```

```
Out[21]:    favorite_count  retweet_count            tweet_id
         0           37731           8221  892420643555336193
         1           32404           6077  892177421306343426
         2           24401           4022  891815181378084864
         3           41040           8376  891689557279858688
         4           39238           9079  891327558926688256
```

```
In [22]: df_csv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

```
In [23]: df_csv.doggo.unique()
```

```
Out[23]: array(['None', 'doggo'], dtype=object)
```

```
In [24]: df_csv.floofer.unique()
```

```
Out[24]: array(['None', 'floofer'], dtype=object)
```

```
In [25]: df_csv.puppo.unique()

Out[25]: array(['None', 'puppo'], dtype=object)

In [26]: df_csv.pupper.unique()

Out[26]: array(['None', 'pupper'], dtype=object)

In [27]: df_csv[df_csv.expanded_urls.isna()].info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 59 entries, 30 to 2298
Data columns (total 17 columns):
tweet_id                      59 non-null int64
in_reply_to_status_id         55 non-null float64
in_reply_to_user_id           55 non-null float64
timestamp                     59 non-null object
source                        59 non-null object
text                          59 non-null object
retweeted_status_id           1 non-null float64
retweeted_status_user_id      1 non-null float64
retweeted_status_timestamp    1 non-null object
expanded_urls                 0 non-null object
rating_numerator              59 non-null int64
rating_denominator            59 non-null int64
name                          59 non-null object
doggo                         59 non-null object
floofer                       59 non-null object
pupper                        59 non-null object
puppo                         59 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 8.3+ KB


In [28]: df_csv[df_csv.expanded_urls.isna()].head()

Out[28]:                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         30   886267009285017600           8.862664e+17         2.281182e+09
         55   881633300179243008           8.816070e+17         4.738443e+07
         64   879674319642796034           8.795538e+17         3.105441e+09
         113  870726314365509632           8.707262e+17         1.648776e+07
         148  863427515083354112           8.634256e+17         7.759620e+07

                             timestamp  \
         30   2017-07-15 16:51:35 +0000
         55   2017-07-02 21:58:53 +0000
         64   2017-06-27 12:14:36 +0000
         113  2017-06-02 19:38:25 +0000
         148  2017-05-13 16:15:35 +0000
```

```
                                                 source  \
30   <a href="http://twitter.com/download/iphone" r...
55   <a href="http://twitter.com/download/iphone" r...
64   <a href="http://twitter.com/download/iphone" r...
113  <a href="http://twitter.com/download/iphone" r...
148  <a href="http://twitter.com/download/iphone" r...

                                                   text  retweeted_status_id  \
30   @NonWhiteHat @MayhewMayhem omg hello tanner yo...                  NaN
55   @roushfenway These are good dogs but 17/10 is ...                  NaN
64                    @RealKentMurphy 14/10 confirmed                  NaN
113  @ComplicitOwl @ShopWeRateDogs &gt;10/10 is res...                  NaN
148  @Jack_Septic_Eye I'd need a few more pics to p...                  NaN

     retweeted_status_user_id retweeted_status_timestamp expanded_urls  \
30                        NaN                        NaN           NaN
55                        NaN                        NaN           NaN
64                        NaN                        NaN           NaN
113                       NaN                        NaN           NaN
148                       NaN                        NaN           NaN

     rating_numerator  rating_denominator  name doggo floofer pupper puppo
30                 12                  10  None  None    None   None  None
55                 17                  10  None  None    None   None  None
64                 14                  10  None  None    None   None  None
113                10                  10  None  None    None   None  None
148                12                  10  None  None    None   None  None
```

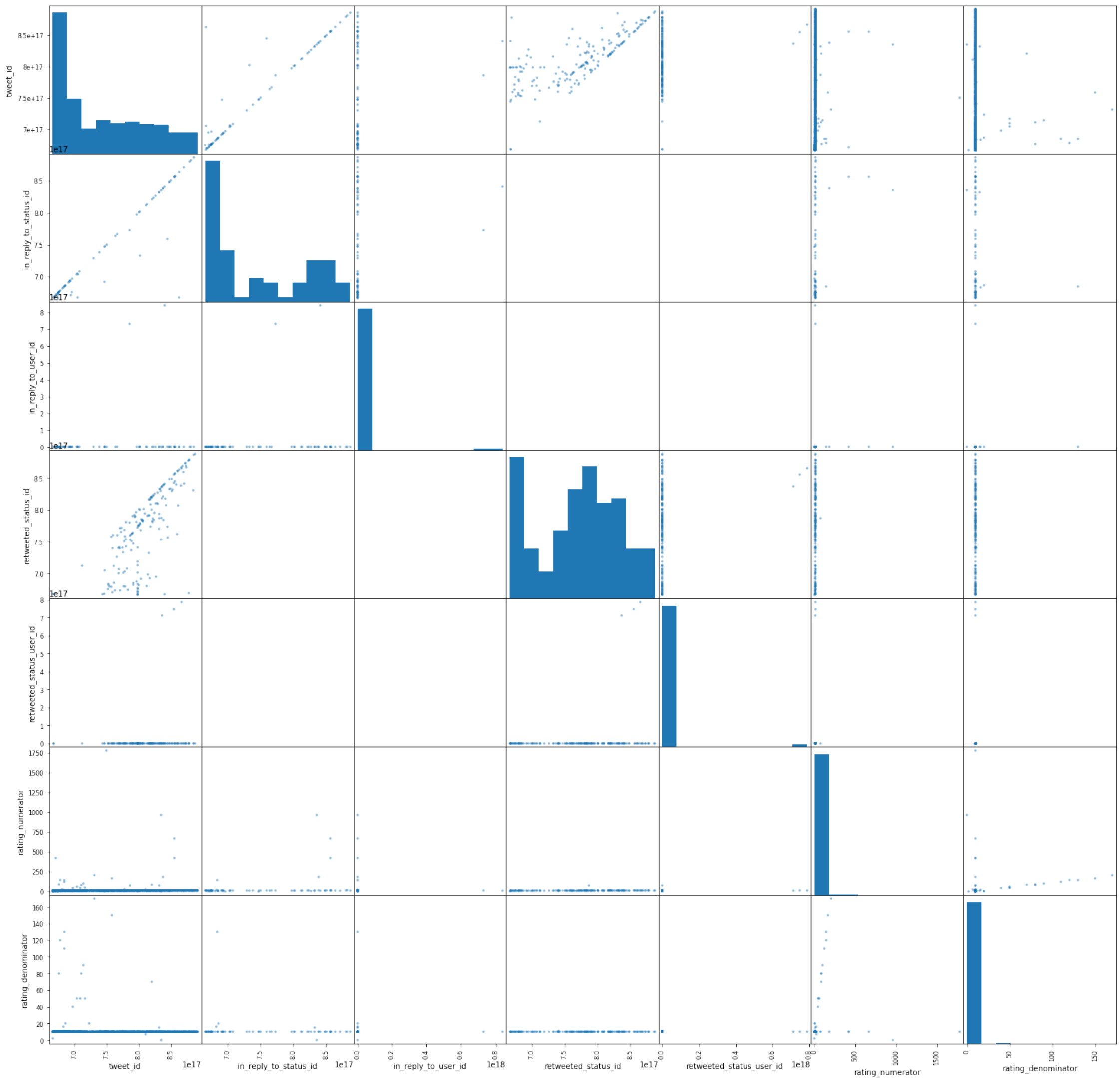
```
In [29]: df_csv[df_csv.tweet_id.duplicated()]
```

```
Out[29]: Empty DataFrame
         Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, tex
         Index: []
```

```
In [30]: pd.scatter_matrix(df_csv,figsize=(25,25))
```

```
C:\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: pandas.scatter_matrix is
  """Entry point for launching an IPython kernel.
```

```
Out[30]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000233E6E4A2B0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E6F40AC8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E6F76160>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E6F9C7F0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E6FC2E80>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E6FC2EB8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E701ABE0>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x00000233E704B2B0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E7073940>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E709DFD0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E70FB6A0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E7124D30>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E7154400>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E7179A58>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x00000233E71AF128>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E71D87B8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E71FFE48>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E722E518>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E7256BA8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E7287278>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E6E4AA58>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x00000233E72D6F60>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E7308630>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E732FCC0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E8330390>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E8357A20>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E83890F0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E83AE780>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x00000233E83D9E10>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E84094E0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E842FB70>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E8462240>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E84898D0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E84B2F60>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E84E4630>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x00000233E8508CC0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E853A390>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E8562A20>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E85950F0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E85BB780>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E85E4E10>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E86174E0>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x00000233E863CB70>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E866C240>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E86958D0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E86BCF60>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E86EB630>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E8719CC0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x00000233E8746390>]],
               dtype=object)
```



```
In [31]: df_img.head()

Out[31]:             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
```

```
In [32]: df_img.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

```
In [33]: df_img.p3.unique()
```

```
Out[33]: array(['Shetland_sheepdog', 'Rhodesian_ridgeback', 'bloodhound',
                'miniature_pinscher', 'Doberman', 'Greater_Swiss_Mountain_dog',
                'terrapin', 'fur_coat', 'golden_retriever',
                'soft-coated_wheaten_terrier', 'Labrador_retriever', 'Pekinese',
                'Ibizan_hound', 'French_bulldog', 'malinois', 'Dandie_Dinmont',
                'borzoi', 'partridge', 'bookcase', 'basenji', 'miniature_poodle',
                'great_grey_owl', 'groenendael', 'Eskimo_dog', 'hamster', 'briard',
                'papillon', 'flat-coated_retriever', 'gar', 'Chihuahua',
                'Shih-Tzu', 'Pomeranian', 'dingo', 'power_drill', 'Saluki',
                'Great_Pyrenees', 'West_Highland_white_terrier', 'collie',
                'toy_poodle', 'vizsla', 'acorn', 'giant_schnauzer', 'teddy',
                'common_iguana', 'wig', 'water_buffalo', 'coyote', 'seat_belt',
                'kelpie', 'space_heater', 'Brabancon_griffon', 'standard_poodle',
                'beagle', 'Irish_water_spaniel', 'bluetick', 'Weimaraner',
                'Chesapeake_Bay_retriever', 'toilet_tissue',
                'black-and-tan_coonhound', 'kuvasz', 'Christmas_stocking',
                'badger', 'hen', 'Staffordshire_bullterrier', 'Yorkshire_terrier',
                'Lakeland_terrier', 'weasel', 'ski_mask', 'cocker_spaniel',
                'Australian_terrier', 'lampshade', 'oscilloscope', 'ram', 'jeep',
                'ice_bear', 'African_grey', 'Great_Dane', 'curly-coated_retriever',
                'doormat', 'African_chameleon', 'schipperke', 'muzzle',
                'triceratops', 'Newfoundland', 'Band_Aid', 'wood_rabbit',
                'white_wolf', 'giant_panda', 'Welsh_springer_spaniel',
                'French_horn', 'toy_terrier', 'Pembroke', 'Cardigan', 'bassinet',
                'pug', 'Afghan_hound', 'American_Staffordshire_terrier', 'whippet',
                'English_setter', 'panpipe', 'crane', 'mouse', 'titi', 'Angora',
                'Boston_bull', 'silky_terrier', 'Japanese_spaniel', 'sandbar',
                'balance_beam', 'black-footed_ferret', 'miniature_schnauzer',
                'Blenheim_spaniel', 'bathtub', 'Saint_Bernard', 'redbone',
                'goldfish', 'Norfolk_terrier', 'llama', 'koala', 'pillow',
                'jersey', 'chow', 'minibus', 'malamute', 'bulletproof_vest',
                'beach_wagon', 'cairn', 'plunger', 'paper_towel', 'wing',
                'English_foxhound', 'Brittany_spaniel', 'bolete', 'ashcan',
                'box_turtle', 'guinea_pig', 'bison', 'bull_mastiff', 'racket',
                'cardoon', 'Tibetan_mastiff', 'window_screen', 'Irish_terrier',
                'agama', 'common_newt', 'car_wheel', 'gorilla', 'bagel', 'clumber',
                'Egyptian_cat', 'television', 'boxer', 'brown_bear', 'leafhopper',
                'German_shepherd', 'Border_collie', 'menu', 'wolf_spider',
                'bathing_cap', 'stinkhorn', 'drumstick', 'mask',
                'Scottish_deerhound', 'shower_curtain', 'Appenzeller',
                'plastic_bag', 'swimming_trunks', 'prairie_chicken', 'red_wolf',
                'Maltese_dog', 'snail', 'gibbon', 'Gordon_setter', 'black_swan',
                'beacon', 'wool', 'cowboy_boot', 'Rottweiler', 'poncho', 'swing',
                'Arctic_fox', 'bib', 'Italian_greyhound', 'steam_locomotive',
                'fountain', 'chickadee', 'abaya', 'Border_terrier', 'bubble',
                'chimpanzee', 'hammerhead', 'Norwegian_elkhound',
                'Norwich_terrier', 'Airedale', 'Siamese_cat', 'sea_cucumber',
                'seashore', 'nipple', 'moped', 'Arabian_camel', 'crayfish',
                'wallaby', 'wire-haired_fox_terrier', 'toilet_seat',
                'Old_English_sheepdog', 'pajama', 'Walker_hound', 'shovel',
                'bucket', 'Sealyham_terrier', 'Windsor_tie', 'Siberian_husky',
                'quill', 'Persian_cat', 'European_fire_salamander',
                'three-toed_sloth', 'swab', 'echidna', 'tennis_ball', 'Lhasa',
                'coral_reef', 'keeshond', 'mink', 'screw', 'basset', 'wreck',
                'kimono', 'German_short-haired_pointer', 'joystick', 'microwave',
                'Tibetan_terrier', 'Irish_wolfhound', 'Samoyed', 'loggerhead',
                'French_loaf', 'Irish_setter', 'komondor', 'purse', 'greenhouse',
                'broccoli', 'shopping_basket', 'macaque', 'squirrel_monkey',
                'green_lizard', 'parallel_bars', 'cloak', 'chest', 'sundial',
                'mosquito_net', 'bath_towel', 'cuirass', 'zebra', 'lumbermill',
                'wallet', 'feather_boa', 'English_springer', 'electric_fan',
                'hippopotamus', 'ox', 'quilt', 'assault_rifle', 'axolotl', 'pot',
                'toyshop', 'pizza', 'scuba_diver', 'beaver', 'Mexican_hairless',
                'cliff', 'loupe', 'wild_boar', 'jaguar', 'hog', 'polecat', 'lion',
                'EntleBucher', 'hand-held_computer', 'washbasin', 'whiptail',
                'rock_crab', 'hare', 'shoji', 'sombrero', 'bell_cote', 'rifle',
                'goose', 'pickup', 'sunglasses', 'limousine', 'bow_tie', 'pretzel',
                'marmot', 'ice_lolly', 'vacuum', 'dalmatian', 'prison',
                'shower_cap', 'sliding_door', 'dugong', 'otterhound', 'eel',
                'binder', 'bullfrog', 'soap_dispenser', 'sea_lion', 'carton',
                'brass', 'mitten', 'golfcart', 'cougar', 'warthog', 'umbrella',
                'neck_brace', 'cup', 'book_jacket', 'padlock', 'cab', 'chime',
                'Leonberg', 'viaduct', 'American_black_bear', 'tub', 'hand_blower',
                'king_penguin', 'rotisserie', 'bannister', 'passenger_car',
                'mongoose', 'dhole', 'consomme', 'valley', 'park_bench',
                'mushroom', 'barrow', 'parachute', 'desktop_computer', 'snorkel',
                'wok', 'affenpinscher', 'space_shuttle', 'rain_barrel',
                'ballplayer', 'mountain_tent', 'oxcart', 'buckeye', 'sunglass',
                'croquet_ball', 'refrigerator', 'snow_leopard', 'tripod',
                'rapeseed', 'tiger_cat', 'Bernese_mountain_dog', 'notebook',
                'maraca', 'pool_table', 'lakeside', 'theater_curtain', 'pier',
                'cheetah', 'mousetrap', 'pop_bottle', 'soccer_ball', 'wombat',
                'rhinoceros_beetle', 'paddlewheel', 'paintbrush', 'maze',
                'hatchet', 'chain', 'jigsaw_puzzle', 'switch',
                'Kerry_blue_terrier', 'barbell', 'convertible',
                'entertainment_center', 'file', 'guillotine', 'nail',
                'standard_schnauzer', 'bow', 'grocery_store', 'boathouse', 'conch',
                'Bouvier_des_Flandres', 'grey_fox', 'shopping_cart', 'meerkat',
                'grand_piano', 'envelope', 'screen', 'coffeepot', 'printer',
                'otter', 'restaurant', 'bonnet', 'crossword_puzzle', 'go-kart',
                'Sussex_spaniel', 'orangutan', 'canoe', 'barber_chair',
                'traffic_light', 'ibex', 'can_opener', 'Indian_elephant',
                'spatula', 'banana'], dtype=object)
```

```
In [34]: df_img.img_num.unique()
```

```
Out[34]: array([1, 4, 2, 3], dtype=int64)
```

```
In [35]: df_img[df_img.img_num==4].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 31 entries, 144 to 2040
Data columns (total 12 columns):
tweet_id    31 non-null int64
jpg_url     31 non-null object
img_num     31 non-null int64
p1          31 non-null object
p1_conf     31 non-null float64
p1_dog      31 non-null bool
p2          31 non-null object
p2_conf     31 non-null float64
p2_dog      31 non-null bool
p3          31 non-null object
p3_conf     31 non-null float64
```

```
p3_dog      31 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 2.5+ KB
```

```
In [36]: df_img[df_img.img_num==4].head()
```

```
Out[36]:                 tweet_id                                           jpg_url  \
         144   668623201287675904  https://pbs.twimg.com/media/CUdtP1xUYAIeBnE.jpg
         779   689905486972461056  https://pbs.twimg.com/media/CZMJYCRVAAE35Wk.jpg
         1024  710588934686908417  https://pbs.twimg.com/media/CdyE2x1W8AAe0TG.jpg
         1161  734787690684657664  https://pbs.twimg.com/media/CjJ9gQ1WgAAXQtJ.jpg
         1286  750868782890057730  https://pbs.twimg.com/media/CmufLLsXYAAsU0r.jpg

               img_num                p1  p1_conf  p1_dog                 p2  \
         144         4         Chihuahua  0.708163    True         Pomeranian
         779         4        Pomeranian  0.943331    True  Shetland_sheepdog
         1024        4          Pembroke  0.982004    True           Cardigan
         1161        4  golden_retriever  0.883991    True               chow
         1286        4        toy_poodle  0.912648    True   miniature_poodle

                p2_conf  p2_dog                   p3  p3_conf  p3_dog
         144   0.091372    True                 titi  0.067325   False
         779   0.023675    True                 chow  0.007165    True
         1024  0.008943    True             malamute  0.007550    True
         1161  0.023542    True   Labrador_retriever  0.016056    True
         1286  0.035059    True            seat_belt  0.026376   False
```

```
In [37]:  df_api.head()
```

```
Out[37]:    favorite_count  retweet_count            tweet_id
         0           37731           8221  892420643555336193
         1           32404           6077  892177421306343426
         2           24401           4022  891815181378084864
         3           41040           8376  891689557279858688
         4           39238           9079  891327558926688256
```

```
In [38]: df_api.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2339 entries, 0 to 2338
Data columns (total 3 columns):
favorite_count    2339 non-null int64
retweet_count     2339 non-null int64
tweet_id          2339 non-null int64
dtypes: int64(3)
memory usage: 54.9 KB
```

1.3 Assessing Results

1.3.1 Quality:

#### df_csv 1 - time stamp need to be changed to datatime instead of string 2 - dummy varialbs (dog type) need to be cleaned into int (boolean) 3 - in_reply_to_status_id should be changed to string type 4 - retweeted_status_user_id should be changed to string type 5 - rating_numerator ,rating_denominator changing into one float variable #### df_img: changing id's into string #### df_api

changing id's into string

1.3.2 Tideness

1 - Dog stages need to be combined into one column 2 - df_api , df_csv represent the same obeservations it would better to be merged together but we should be careful about the nullable and deference count between the raw 3 - merged , df_img represent the same obeservations it would better to be merged together but we should be careful about the nullable and deference count between the raw

## Cleaning Data

1.3.3 Quality

1.3.4 df_csv

First We will copy our dataframe to a new data frame to ensure the consisty of our work

```
In [39]: df_csv_clean=df_csv.copy()
```

Define chaging time into datatime type

Code

```
In [40]: df_csv_clean["timestamp"]=pd.to_datetime(df_csv_clean.timestamp)
```

Test

```
In [41]: df_csv_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null datetime64[ns]
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
```

```
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: datetime64[ns](1), float64(4), int64(3), object(9)
memory usage: 313.0+ KB
```

Define chaging tweet_id into string type

Code

```
In [42]: df_csv_clean["tweet_id"]=df_csv_clean["tweet_id"].astype(str)
         #df_csv_clean["retweeted_status_id"]=df_csv_clean["retweeted_status_id"].astype(str)
         #df_csv_clean["retweeted_status_user_id"]=df_csv_clean["retweeted_status_user_id"].as
         #df_csv_clean["in_reply_to_status_id"]=df_csv_clean["in_reply_to_status_id"].astype(s
         #df_csv_clean["in_reply_to_user_id"]=df_csv_clean["in_reply_to_user_id"].astype(str)
```

Test

```
In [43]: df_csv_clean.head()
```

```
Out[43]:              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                                             source  \
         0 2017-08-01 16:23:56  <a href="http://twitter.com/download/iphone" r...
         1 2017-08-01 00:17:27  <a href="http://twitter.com/download/iphone" r...
         2 2017-07-31 00:18:03  <a href="http://twitter.com/download/iphone" r...
         3 2017-07-30 15:58:51  <a href="http://twitter.com/download/iphone" r...
         4 2017-07-29 16:00:24  <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
```

Define Drop retweeted_status_id, retweeted_status_user_id,in_reply_to_status_id,in_reply_to_user_id

Code

```
In [44]: df_csv_clean.drop(["retweeted_status_id",
                             "retweeted_status_user_id",
                             "in_reply_to_status_id",
                             "in_reply_to_user_id",
                            "retweeted_status_timestamp"],
                           axis=1,inplace=True)
```

Test

```
In [45]: df_csv_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2356 non-null object
timestamp             2356 non-null datetime64[ns]
source                2356 non-null object
text                  2356 non-null object
expanded_urls         2297 non-null object
rating_numerator      2356 non-null int64
rating_denominator    2356 non-null int64
name                  2356 non-null object
doggo                 2356 non-null object
floofer               2356 non-null object
```

```
pupper                 2356 non-null object
puppo                  2356 non-null object
dtypes: datetime64[ns](1), int64(2), object(9)
memory usage: 221.0+ KB
```

Define The current pipeline captures incorrect values when rating numerators contain decimals and it will better to compine both numerator and deomerator in one column as it descripe one mesure

Code

```
In [46]: df_csv_clean[df_csv_clean.rating_numerator>10].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1455 entries, 0 to 2339
Data columns (total 12 columns):
tweet_id               1455 non-null object
timestamp              1455 non-null datetime64[ns]
source                 1455 non-null object
text                   1455 non-null object
expanded_urls          1412 non-null object
rating_numerator       1455 non-null int64
rating_denominator     1455 non-null int64
name                   1455 non-null object
doggo                  1455 non-null object
floofer                1455 non-null object
pupper                 1455 non-null object
puppo                  1455 non-null object
dtypes: datetime64[ns](1), int64(2), object(9)
memory usage: 147.8+ KB
```

```
In [47]: ratting=df_csv_clean.text.str.extract('((?:\d+\.)?\d+)\/(\d+)', expand=True)
         df_csv_clean.rating_numerator=ratting[0].astype(float)
         df_csv_clean.rating_denominator=ratting[1].astype(float)
         df_csv_clean=df_csv_clean[df_csv_clean["rating_denominator"]!=0]
```

Test

```
In [48]: df_csv_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2355 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id               2355 non-null object
timestamp              2355 non-null datetime64[ns]
source                 2355 non-null object
```

```
text                   2355 non-null object
expanded_urls          2297 non-null object
rating_numerator       2355 non-null float64
rating_denominator     2355 non-null float64
name                   2355 non-null object
doggo                  2355 non-null object
floofer                2355 non-null object
pupper                 2355 non-null object
puppo                  2355 non-null object
dtypes: datetime64[ns](1), float64(2), object(9)
memory usage: 239.2+ KB
```

```
In [49]: df_csv_clean.describe()
```

```
Out[49]:        rating_numerator  rating_denominator
         count       2355.000000         2355.000000
         mean          12.661584           10.459873
         std           41.485316            6.743226
         min            0.000000            2.000000
         25%           10.000000           10.000000
         50%           11.000000           10.000000
         75%           12.000000           10.000000
         max         1776.000000          170.000000
```

1.3.5 df_img

Copy df_img to a new dataframe

```
In [50]: df_img_clean=df_img.copy()
```

Define 1- chaging id into string

Code

```
In [51]: df_img_clean["tweet_id"]=df_img_clean["tweet_id"].astype(str)
```

Test

```
In [52]: df_img_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null object
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
```

```
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(1), object(5)
memory usage: 152.1+ KB
```

```
In [53]: df_img_clean.describe()
```

```
Out[53]:            img_num      p1_conf       p2_conf       p3_conf
         count  2075.000000  2075.000000  2.075000e+03  2.075000e+03
         mean      1.203855     0.594548  1.345886e-01  6.032417e-02
         std       0.561875     0.271174  1.006657e-01  5.090593e-02
         min       1.000000     0.044333  1.011300e-08  1.740170e-10
         25%       1.000000     0.364412  5.388625e-02  1.622240e-02
         50%       1.000000     0.588230  1.181810e-01  4.944380e-02
         75%       1.000000     0.843855  1.955655e-01  9.180755e-02
         max       4.000000     1.000000  4.880140e-01  2.734190e-01
```

1.3.6 df_api

Copy df_api into other cleaning Dataframe

```
In [54]: df_api_clean=df_api.copy()
```

Define chage id into string

Code

```
In [55]: df_api_clean["tweet_id"]=df_api_clean["tweet_id"].astype(str)
```

Test

```
In [56]: df_api_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2339 entries, 0 to 2338
Data columns (total 3 columns):
favorite_count    2339 non-null int64
retweet_count     2339 non-null int64
tweet_id          2339 non-null object
dtypes: int64(2), object(1)
memory usage: 54.9+ KB
```

```
In [57]: df_api.describe()

Out[57]:        favorite_count  retweet_count      tweet_id
         count     2339.000000    2339.000000  2.339000e+03
         mean      7887.823001    2899.079949  7.421620e+17
         std      12224.011478    4888.510848  6.828729e+16
         min          0.000000       0.000000  6.660209e+17
         25%       1366.500000     583.000000  6.783378e+17
         50%       3431.000000    1353.000000  7.186133e+17
         75%       9658.500000    3379.000000  7.986907e+17
         max     162830.000000   82862.000000  8.924206e+17
```

1.4 Cleaning

1.4.1 Tidiness

df_csv_clean

Define Dog stages need to be combined into one column

```
In [58]: df_csv_clean.loc[(df_csv_clean[['doggo', 'floofer', 'pupper', 'puppo']] != 'None'
                          ).sum(axis=1) > 1]

Out[58]:                 tweet_id           timestamp  \
         191   855851453814013952 2017-04-22 18:31:02
         200   854010172552949760 2017-04-17 16:34:26
         460   817777686764523521 2017-01-07 16:59:28
         531   808106460588765185 2016-12-12 00:29:28
         565   802265048156610565 2016-11-25 21:37:47
         575   801115127852503040 2016-11-22 17:28:25
         705   785639753186217984 2016-10-11 00:34:48
         733   781308096455073793 2016-09-29 01:42:20
         778   775898661951791106 2016-09-14 03:27:11
         822   770093767776997377 2016-08-29 03:00:36
         889   759793422261743616 2016-07-31 16:50:42
         956   751583847268179968 2016-07-09 01:08:47
         1063  741067306818797568 2016-06-10 00:39:48
         1113  733109485275860992 2016-05-19 01:38:16

                                                            source  \
         191   <a href="http://twitter.com/download/iphone" r...
         200   <a href="http://twitter.com/download/iphone" r...
         460   <a href="http://twitter.com/download/iphone" r...
         531   <a href="http://twitter.com/download/iphone" r...
         565   <a href="http://twitter.com/download/iphone" r...
         575   <a href="http://twitter.com/download/iphone" r...
         705   <a href="http://twitter.com/download/iphone" r...
         733   <a href="http://vine.co" rel="nofollow">Vine -...
         778   <a href="http://twitter.com/download/iphone" r...
```

```
822   <a href="http://twitter.com/download/iphone" r...
889   <a href="http://twitter.com/download/iphone" r...
956   <a href="http://twitter.com/download/iphone" r...
1063  <a href="http://twitter.com/download/iphone" r...
1113  <a href="http://twitter.com/download/iphone" r...

                                                   text  \
191   Here's a puppo participating in the #ScienceMa...
200   At first I thought this was a shy doggo, but i...
460   This is Dido. She's playing the lead role in "...
531   Here we have Burke (pupper) and Dexter (doggo)...
565   Like doggo, like pupper version 2. Both 11/10 ...
575   This is Bones. He's being haunted by another d...
705   This is Pinot. He's a sophisticated doggo. You...
733   Pupper butt 1, Doggo 0. Both 12/10 https://t.c...
778   RT @dog_rates: Like father (doggo), like son (...
822   RT @dog_rates: This is just downright precious...
889   Meet Maggie &amp; Lila. Maggie is the doggo, L...
956   Please stop sending it pictures that don't eve...
1063  This is just downright precious af. 12/10 for ...
1113  Like father (doggo), like son (pupper). Both 1...

                                          expanded_urls  rating_numerator  \
191   https://twitter.com/dog_rates/status/855851453...              13.0
200   https://twitter.com/dog_rates/status/854010172...              11.0
460   https://twitter.com/dog_rates/status/817777686...              13.0
531   https://twitter.com/dog_rates/status/808106460...              12.0
565   https://twitter.com/dog_rates/status/802265048...              11.0
575   https://twitter.com/dog_rates/status/801115127...              12.0
705   https://twitter.com/dog_rates/status/785639753...              10.0
733                       https://vine.co/v/5rgu2Law2ut              12.0
778   https://twitter.com/dog_rates/status/733109485...              12.0
822   https://twitter.com/dog_rates/status/741067306...              12.0
889   https://twitter.com/dog_rates/status/759793422...              12.0
956   https://twitter.com/dog_rates/status/751583847...               5.0
1063  https://twitter.com/dog_rates/status/741067306...              12.0
1113  https://twitter.com/dog_rates/status/733109485...              12.0

      rating_denominator    name  doggo  floofer  pupper  puppo
191                 10.0    None  doggo     None    None  puppo
200                 10.0    None  doggo  floofer    None   None
460                 10.0    Dido  doggo     None  pupper   None
531                 10.0    None  doggo     None  pupper   None
565                 10.0    None  doggo     None  pupper   None
575                 10.0   Bones  doggo     None  pupper   None
705                 10.0   Pinot  doggo     None  pupper   None
733                 10.0    None  doggo     None  pupper   None
778                 10.0    None  doggo     None  pupper   None
```

```
822                 10.0    just  doggo     None  pupper   None
889                 10.0  Maggie  doggo     None  pupper   None
956                 10.0    None  doggo     None  pupper   None
1063                10.0    just  doggo     None  pupper   None
1113                10.0    None  doggo     None  pupper   None
```

Code

```
In [59]: df_csv_clean.doggo.replace("None","",inplace=True)
         df_csv_clean.floofer.replace("None","",inplace=True)
         df_csv_clean.pupper.replace("None","",inplace=True)
         df_csv_clean.puppo.replace("None","",inplace=True)
```

```
In [60]: df_csv_clean['stage'] = df_csv_clean.doggo + df_csv_clean.floofer + df_csv_clean.puppe
         df_csv_clean.loc[df_csv_clean.stage == 'doggopupper', 'stage'] = 'doggo,pupper'
         df_csv_clean.loc[df_csv_clean.stage == 'doggopuppo', 'stage'] = 'doggo,puppo'
         df_csv_clean.loc[df_csv_clean.stage == 'doggofloofer', 'stage'] = 'doggo,floofer'
         df_csv_clean.stage.replace("","None",inplace=True)
```

Test

```
In [61]: df_csv_clean.stage.value_counts()
```

```
Out[61]: None             1975
         pupper            245
         doggo              83
         puppo              29
         doggo,pupper       12
         floofer             9
         doggo,puppo         1
         doggo,floofer       1
         Name: stage, dtype: int64
```

Define Drop the dummy variabls

Code

```
In [62]: df_csv_clean.drop(['doggo', 'floofer', 'pupper', 'puppo'],axis=1,inplace=True)
```

Test

```
In [63]: df_csv_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2355 entries, 0 to 2355
Data columns (total 9 columns):
tweet_id              2355 non-null object
timestamp             2355 non-null datetime64[ns]
source                2355 non-null object
```

```
text                    2355 non-null object
expanded_urls           2297 non-null object
rating_numerator        2355 non-null float64
rating_denominator      2355 non-null float64
name                    2355 non-null object
stage                   2355 non-null object
dtypes: datetime64[ns](1), float64(2), object(6)
memory usage: 184.0+ KB
```

### df_api , df_csv #### Define ##
represent the same obeservations it would better to be merged together but we should be careful about the nullable and deference count between the raw

Code We will Use merge for two Dataframe

```
In [64]: merged_1=df_csv_clean.merge(df_api_clean,how="inner",on="tweet_id")
```

Test

```
In [65]: merged_1.head()
```

```
Out[65]:              tweet_id           timestamp  \
         0  892420643555336193 2017-08-01 16:23:56
         1  892177421306343426 2017-08-01 00:17:27
         2  891815181378084864 2017-07-31 00:18:03
         3  891689557279858688 2017-07-30 15:58:51
         4  891327558926688256 2017-07-29 16:00:24

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

                                                 expanded_urls  rating_numerator  \
         0  https://twitter.com/dog_rates/status/892420643...              13.0
         1  https://twitter.com/dog_rates/status/892177421...              13.0
         2  https://twitter.com/dog_rates/status/891815181...              12.0
         3  https://twitter.com/dog_rates/status/891689557...              13.0
```

```
4  https://twitter.com/dog_rates/status/891327558...             12.0

   rating_denominator      name stage  favorite_count  retweet_count
0                10.0   Phineas  None           37731           8221
1                10.0     Tilly  None           32404           6077
2                10.0    Archie  None           24401           4022
3                10.0     Darla  None           41040           8376
4                10.0  Franklin  None           39238           9079
```

```
In [66]: merged_1.describe()
```

```
Out[66]:        rating_numerator  rating_denominator  favorite_count  retweet_count
         count       2338.000000         2338.000000     2338.000000    2338.000000
         mean          12.663828           10.462789     7890.263045    2900.286997
         std           41.635748            6.767591    12226.056769    4889.207987
         min            0.000000            2.000000        0.000000       0.000000
         25%           10.000000           10.000000     1365.250000     583.500000
         50%           11.000000           10.000000     3436.500000    1356.500000
         75%           12.000000           10.000000     9658.750000    3379.500000
         max         1776.000000          170.000000   162830.000000   82862.000000
```

```
In [67]: merged_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2338 entries, 0 to 2337
Data columns (total 11 columns):
tweet_id              2338 non-null object
timestamp             2338 non-null datetime64[ns]
source                2338 non-null object
text                  2338 non-null object
expanded_urls         2280 non-null object
rating_numerator      2338 non-null float64
rating_denominator    2338 non-null float64
name                  2338 non-null object
stage                 2338 non-null object
favorite_count        2338 non-null int64
retweet_count         2338 non-null int64
dtypes: datetime64[ns](1), float64(2), int64(2), object(6)
memory usage: 219.2+ KB
```

1.4.2 df_img , merged_1

Define df_img represent the same obeservations it would better to be merged together but we should be careful about the nullable and deference count between the raw

Code

```
In [68]: merged_2=merged_1.merge(df_img_clean,how="inner",on="tweet_id")
```

Test

```
In [69]: merged_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2066 entries, 0 to 2065
Data columns (total 22 columns):
tweet_id              2066 non-null object
timestamp             2066 non-null datetime64[ns]
source                2066 non-null object
text                  2066 non-null object
expanded_urls         2066 non-null object
rating_numerator      2066 non-null float64
rating_denominator    2066 non-null float64
name                  2066 non-null object
stage                 2066 non-null object
favorite_count        2066 non-null int64
retweet_count         2066 non-null int64
jpg_url               2066 non-null object
img_num               2066 non-null int64
p1                    2066 non-null object
p1_conf               2066 non-null float64
p1_dog                2066 non-null bool
p2                    2066 non-null object
p2_conf               2066 non-null float64
p2_dog                2066 non-null bool
p3                    2066 non-null object
p3_conf               2066 non-null float64
p3_dog                2066 non-null bool
dtypes: bool(3), datetime64[ns](1), float64(5), int64(3), object(10)
memory usage: 328.9+ KB
```

```
In [70]: merged_2.describe()
```

```
Out[70]:        rating_numerator  rating_denominator  favorite_count  retweet_count  \
         count       2066.000000         2066.000000     2066.000000    2066.000000
         mean          12.221578           10.513553     8332.854792    2772.818490
         std           40.742699            7.192615    12562.080796    4830.478374
         min            0.000000            2.000000        0.000000      11.000000
         25%           10.000000           10.000000     1584.500000     591.000000
         50%           11.000000           10.000000     3664.500000    1305.000000
         75%           12.000000           10.000000    10408.500000    3199.500000
         max         1776.000000          170.000000   162830.000000   82862.000000

                   img_num      p1_conf       p2_conf       p3_conf
         count  2066.000000  2066.000000  2.066000e+03  2.066000e+03
         mean      1.203291     0.594568  1.346716e-01  6.034151e-02
         std       0.562172     0.271062  1.007233e-01  5.094272e-02
```

```
min       1.000000    0.044333  1.011300e-08  1.740170e-10
25%       1.000000    0.364254  5.387868e-02  1.621080e-02
50%       1.000000    0.588030  1.184015e-01  4.939645e-02
75%       1.000000    0.843883  1.955693e-01  9.208967e-02
max       4.000000    1.000000  4.880140e-01  2.734190e-01
```

In [71]:
```
merged_2.head()
```

Out[71]:
```
             tweet_id           timestamp  \
0  892420643555336193 2017-08-01 16:23:56
1  892177421306343426 2017-08-01 00:17:27
2  891815181378084864 2017-07-31 00:18:03
3  891689557279858688 2017-07-30 15:58:51
4  891327558926688256 2017-07-29 16:00:24

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

                                       expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643...              13.0
1  https://twitter.com/dog_rates/status/892177421...              13.0
2  https://twitter.com/dog_rates/status/891815181...              12.0
3  https://twitter.com/dog_rates/status/891689557...              13.0
4  https://twitter.com/dog_rates/status/891327558...              12.0

   rating_denominator      name stage  favorite_count  ...    img_num  \
0                10.0   Phineas  None           37731  ...          1
1                10.0     Tilly  None           32404  ...          1
2                10.0    Archie  None           24401  ...          1
3                10.0     Darla  None           41040  ...          1
4                10.0  Franklin  None           39238  ...          2

            p1   p1_conf  p1_dog                  p2   p2_conf  p2_dog  \
0       orange  0.097049   False               bagel  0.085851   False
1    Chihuahua  0.323581    True            Pekinese  0.090647    True
2    Chihuahua  0.716012    True            malamute  0.078253    True
3  paper_towel  0.170278   False  Labrador_retriever  0.168086    True
```

```
4       basset  0.555712   True    English_springer  0.225770   True

                           p3   p3_conf p3_dog
0                      banana  0.076110  False
1                    papillon  0.068957   True
2                      kelpie  0.031379   True
3                     spatula  0.040836  False
4  German_short-haired_pointer  0.175219   True

[5 rows x 22 columns]
```

```
In [72]: df=merged_2.copy()
```

## Saving Data ### saving as csv in local file system

```
In [73]: df.to_csv("twitter_archive_master.csv",index_label="tweetid")
```

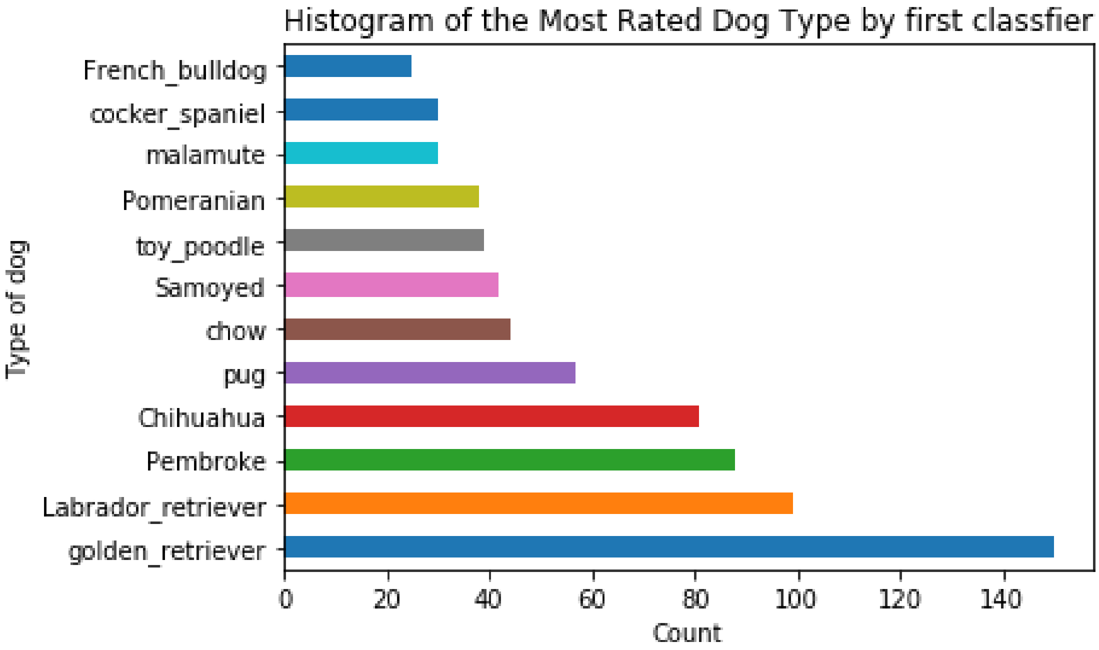
## Analysing and Visulisation ### Brief insights
Using the factors (Catagories) Stage , and name

```
In [74]: df.groupby("stage").mean()
```

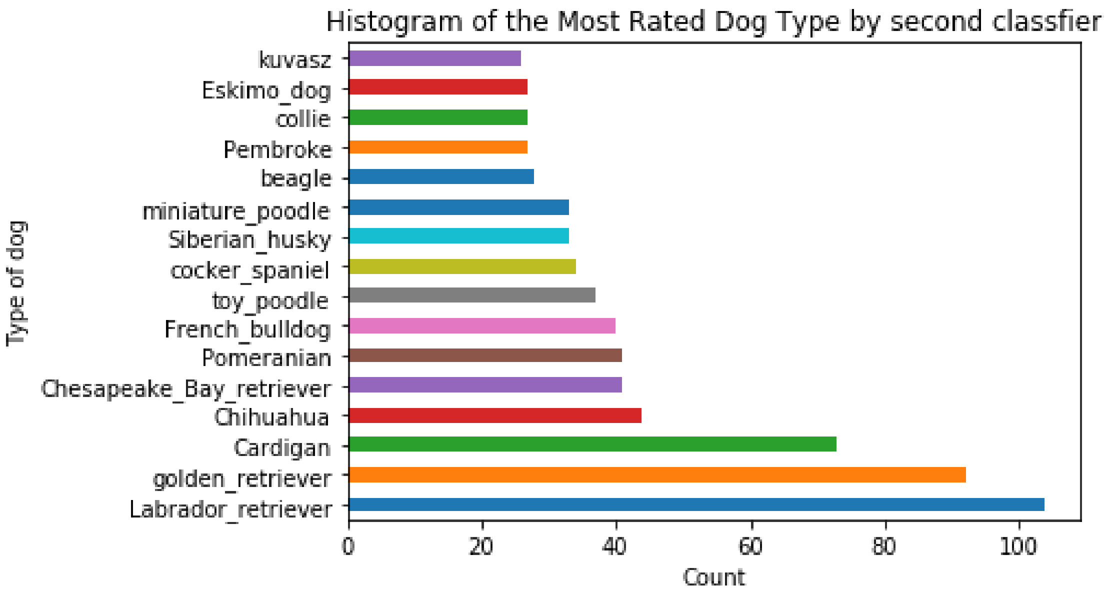
```
Out[74]:               rating_numerator  rating_denominator  favorite_count  \
         stage
         None                 12.427799           10.608023     7913.383954
         doggo                11.970149           10.000000    18727.641791
         doggo,floofer        11.000000           10.000000    16429.000000
         doggo,pupper         11.181818           10.000000    10563.636364
         doggo,puppo          13.000000           10.000000    45876.000000
         floofer              12.000000           10.000000    12563.000000
         pupper               10.683744           10.000000     6683.094787
         puppo                12.000000           10.000000    20673.565217

                       retweet_count   img_num   p1_conf    p1_dog   p2_conf  \
         stage
         None            2562.620630  1.191404  0.591132  0.738682  0.134372
         doggo           7708.029851  1.313433  0.581910  0.761194  0.159575
         doggo,floofer   3237.000000  1.000000  0.354733  1.000000  0.177538
         doggo,pupper    5149.909091  1.090909  0.844620  0.909091  0.076483
         doggo,puppo    18117.000000  1.000000  0.321676  1.000000  0.115138
         floofer         4638.428571  1.142857  0.609204  1.000000  0.155319
         pupper          2312.976303  1.251185  0.602090  0.701422  0.130868
         puppo           6170.521739  1.434783  0.721344  0.826087  0.140264

                         p2_dog   p3_conf    p3_dog
         stage
         None          0.746132  0.061148  0.722636
         doggo         0.776119  0.060745  0.701493
         doggo,floofer 1.000000  0.131706  1.000000
```

```
doggo,pupper   0.818182  0.026425  0.636364
doggo,puppo    1.000000  0.096100  1.000000
floofer        1.000000  0.047997  0.857143
pupper         0.725118  0.057050  0.696682
puppo          0.913043  0.043497  0.956522
```

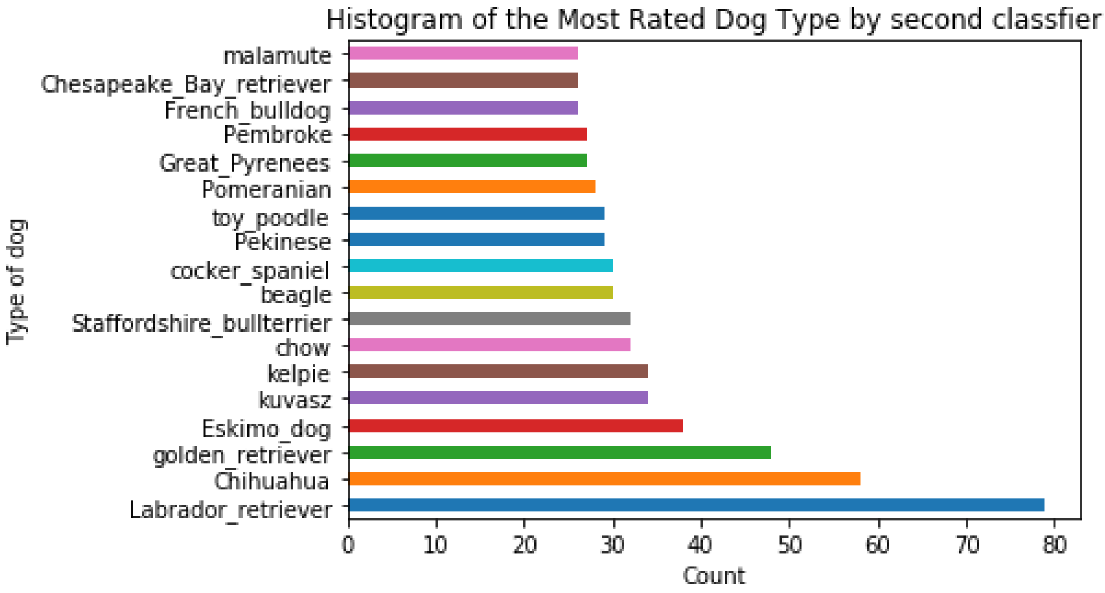
```
In [75]: df_p1 = df.groupby('p1').filter(lambda x: len(x) >= 25)

         df_p1['p1'].value_counts().plot(kind = 'barh')
         plt.title('Histogram of the Most Rated Dog Type by first classfier')
         plt.xlabel('Count')
         plt.ylabel('Type of dog')

         fig = plt.gcf()
         fig.savefig('output1.png',bbox_inches='tight');
```



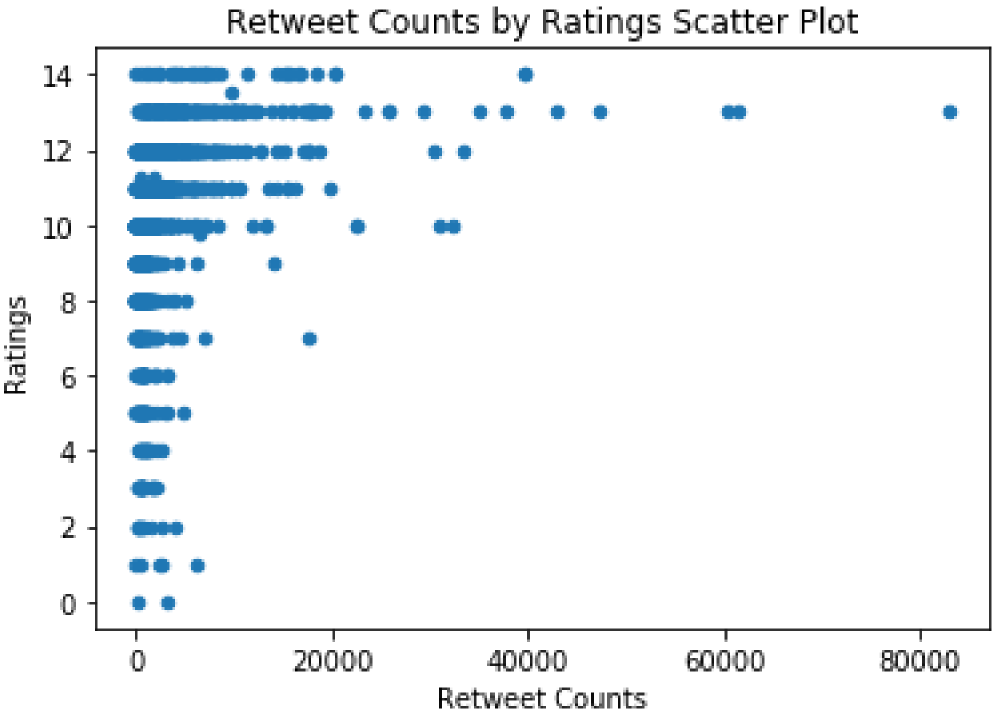
```
In [76]: df_p2 = df.groupby('p2').filter(lambda x: len(x) >= 25)

         df_p2['p2'].value_counts().plot(kind = 'barh')
         plt.title('Histogram of the Most Rated Dog Type by second classfier')
         plt.xlabel('Count')
         plt.ylabel('Type of dog')

         fig = plt.gcf()
         fig.savefig('output2.png',bbox_inches='tight');
```



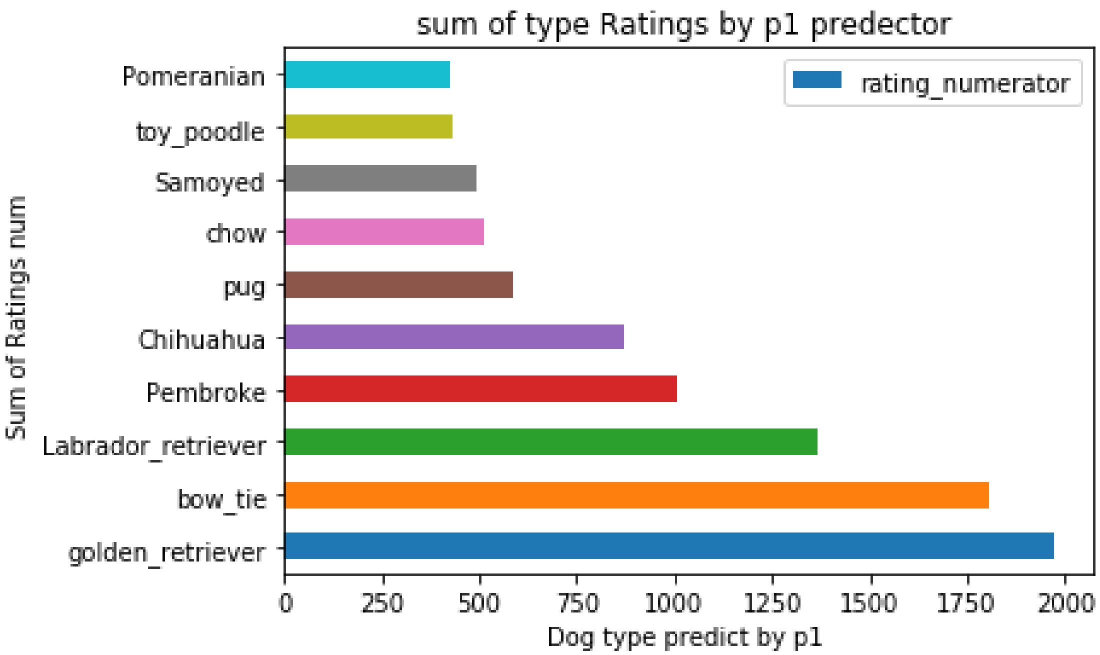
```
In [77]: df_p3 = df.groupby('p3').filter(lambda x: len(x) >= 25)

         df_p3['p3'].value_counts().plot(kind = 'barh')
         plt.title('Histogram of the Most Rated Dog Type by second classfier')
         plt.xlabel('Count')
         plt.ylabel('Type of dog')

         fig = plt.gcf()
         fig.savefig('output3.png',bbox_inches='tight');
```



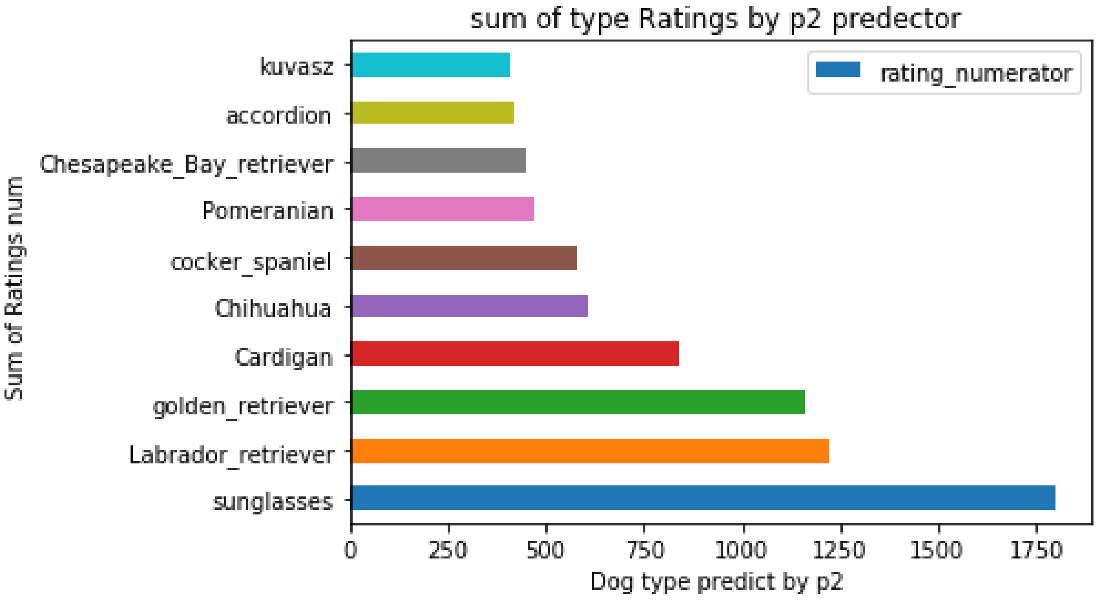
```
In [78]: df[df.rating_numerator<15].plot(x='retweet_count', y='rating_numerator', kind='scatte
         plt.xlabel('Retweet Counts')
         plt.ylabel('Ratings')
         plt.title('Retweet Counts by Ratings Scatter Plot')

         fig = plt.gcf()
         fig.savefig('output4.png',bbox_inches='tight');
```



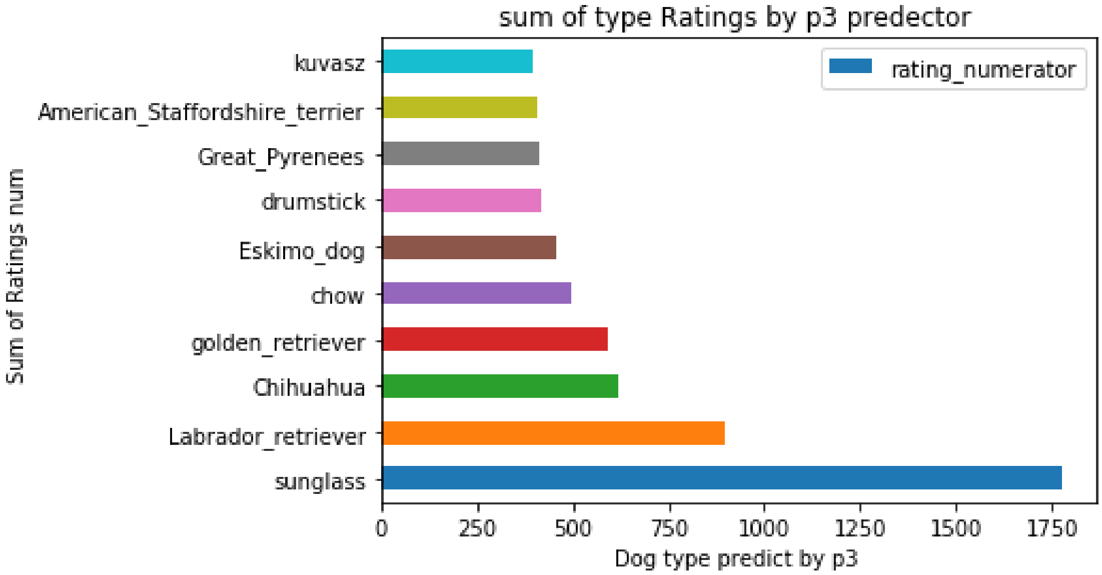
```
In [79]: df.groupby('p1').sum().sort_values("rating_numerator",ascending=0).head(10).plot(y='ra
         plt.xlabel('Dog type predict by p1 ')
         plt.ylabel('Sum of Ratings num ')
         plt.title('sum of type Ratings by p1 predector ')
         fig = plt.gcf()
         fig.savefig('output5.png',bbox_inches='tight');
```



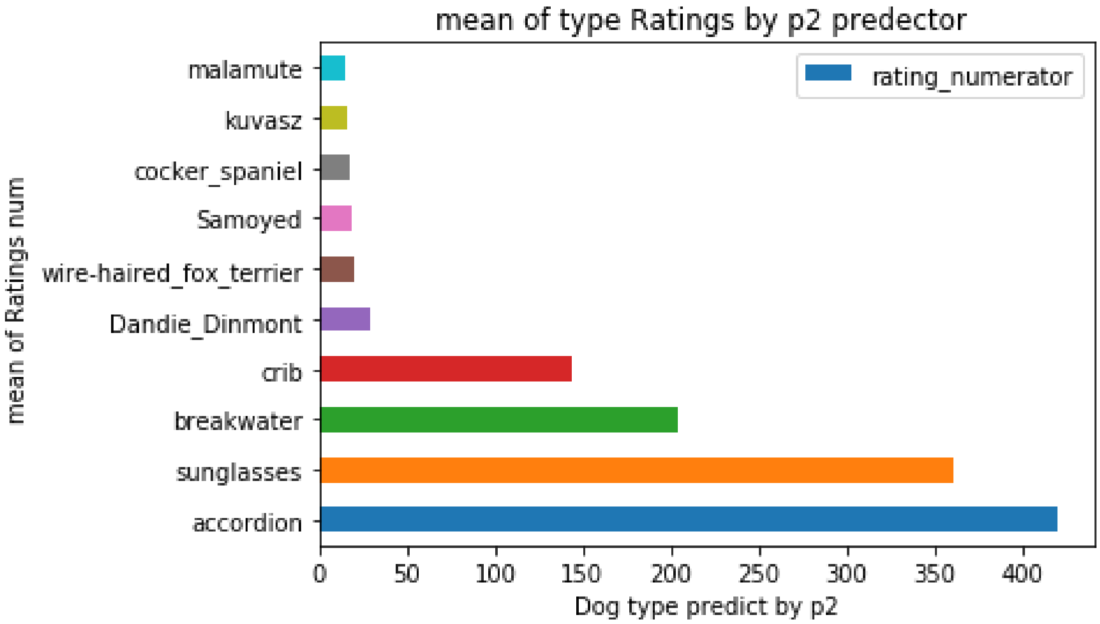
```
In [80]: df.groupby('p2').sum().sort_values("rating_numerator",ascending=0).head(10).plot(y='ra
         plt.xlabel('Dog type predict by p2 ')
         plt.ylabel('Sum of Ratings num ')
         plt.title('sum of type Ratings by p2 predector ')
         fig = plt.gcf()
         fig.savefig('output6.png',bbox_inches='tight');
```



```
In [81]: df.groupby('p3').sum().sort_values("rating_numerator",ascending=0).head(10).plot(y='ra
         plt.xlabel('Dog type predict by p3 ')
         plt.ylabel('Sum of Ratings num ')
         plt.title('sum of type Ratings by p3 predector ')
         fig = plt.gcf()
         fig.savefig('output7.png',bbox_inches='tight');
```



```
In [82]: df.groupby('p2').mean().sort_values("rating_numerator",ascending=0).head(10).plot(y='r
         plt.xlabel('Dog type predict by p2')
         plt.ylabel('mean of Ratings num ')
         plt.title('mean of type Ratings by p2 predector ')
         fig = plt.gcf()
         fig.savefig('output8.png',bbox_inches='tight');
```


mean of type Ratings by p2 predector
rating_numerator
malamute
kuvasz
cocker_spaniel
Samoyed
wire-haired_fox_terrier
Dandie_Dinmont
crib
breakwater
sunglasses
accordion
mean of Ratings num
0
50
100
150
200
250
300
350
400
Dog type predict by p2