wrangle_act

September 14, 2020

1 Data Wrangle Project:

1.1 weratedogs twitter archive

-

1.1.1 Data wrangling, which consists of:

1. Gathering data
2. Assessing data
3. Cleaning data

-

1.1.2 Storing, analyzing, and visualizing wrangled data

-

1.1.3 Reporting on (the two reports will be on the same directory and saved as pdf file)

1. Data wrangling efforts.
2. Data analyses and visualizations

1.2 Packages imported

```
In [1]: import numpy as np
        import pandas as pd
        import os
        import requests
        import tweepy
        from tweepy import OAuthHandler
        import json
        from timeit import default_timer as timer
        import re
        import matplotlib.pyplot as plt
        import seaborn as sns
        % matplotlib inline
```

1.3 1. Gather Data:

Gathering data from diffrent resources: 1. Download file manually 2. Download file from the internet 3. Access twitter API

1. Download file (twitter_archive_enhanced.csv) and load it in pandas dataframe

In [2]:
```
#load the csv file in dataframe
twitter_archive_df = pd.read_csv('twitter-archive-enhanced.csv')
```

In [3]:
```
#test that the data loaded in the dataframe
twitter_archive_df.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
```

2. Download file (image_predictions.tsv) programmatically and load it in pandas dataframe

In [4]:
```
#save the url in string
url = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictio
#get the file name from the url
file_name = url.split('/')[-1]

#use requests package to get the file
response = requests.get(url)
if not os.path.isfile(file_name):
    with open(file_name, mode='wb') as file:
        file.write(response.content)

#Read the image predictions tsv file in pandas dataframe
image_predictions_df = pd.read_csv(file_name , sep="\t")
```

In [5]:
```
#test that the data loaded in the dataframe
image_predictions_df.head()
```

Out[5]:
```
             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
```

3. Query Twitter API to get more info about tweets in twitter-archive file

```
In [6]: #Twitter api credentials
        consumer_key = '52EfRzZgQFBiLCW2m0PWkZSyI'
        consumer_secret = 'NLPNCtUItJvNPXZ24hSA9ZMa4gDJBxPGP5rMmNmYXtAq6KQRjL'
        access_token = '283071900-cFu6ktz6X6PJHGDP9bFZ8LtDaUA0JGeMuQbvJIdD'
        access_secret = 'wiU870tObTGuBpVaBG886rHedLwL61lEalZ5xJIehrzpB'

        #Get authanticated access to api
        auth = OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_secret)

        #Conect to tweeter api using tweepy package
        api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)

        # Tweet IDs in twitter archive dataframe for which to gather more data via Twitter's API
        archive_tweet_ids = twitter_archive_df.tweet_id.values

        # Query Twitter's API for JSON data for each tweet ID in the Twitter archive

        #set variable to show which tweet success and which one fail
        count = 0

        #set dict variable to save tweets that failed to get from the api
        #which will be tweet that only in tweeter archived and not in the api
        tweet_query_fails_dict = {}

In [7]: # Save each tweet's returned JSON as a new line in a tweet_json.txt file
        if not os.path.isfile('tweet_json.txt'):
            with open('tweet_json.txt', 'w') as outfile:
                for tweet_id in archive_tweet_ids:
                    count += 1
                    print(str(count) + ": " + str(tweet_id))
                    try:
                        tweet = api.get_status(tweet_id, tweet_mode='extended')
                        print("Success")
                        json.dump(tweet._json, outfile)
                        outfile.write('\n')
                    except tweepy.TweepError as e:
                        print("Fail")
                        tweet_query_fails_dict[tweet_id] = e
                        pass

In [8]: print(tweet_query_fails_dict)
```

```
{}
```

In [9]:
```
#print tweet to check the content of each tweet to get the potential data needed
with open('tweet_json.txt', 'r') as file:
    for line in file:
        tweet_test = json.loads(line)
        print(tweet_test)
        break
```


```
{'created_at': 'Tue Aug 01 16:23:56 +0000 2017', 'id': 892420643555336193, 'id_str': '8924206435
```


In [10]:
```
#load the tweet json txt file in dataframe
tweet_json_list = []

with open('tweet_json.txt', 'r') as json_file:
    for line in json_file:
        tweet_content= json.loads(line)

        tweet_id = tweet_content['id']
        tweet_retweet_count = tweet_content['retweet_count']
        tweet_favorite_count = tweet_content['favorite_count']
        user_follower_count = tweet_content['user']['followers_count']
        tweet_device= tweet_content['source'].split(">")[-2][:-3]

        tweet_dict= {'tweet_id': tweet_id,
                     'retweet_count': tweet_retweet_count,
                     'favorite_count': tweet_favorite_count,
                     'followers_count': user_follower_count,
                     'source': tweet_device
        }
        tweet_json_list.append(tweet_dict)

twitter_api_df = pd.DataFrame(tweet_json_list)
```

In [11]:
```
twitter_api_df.head()
```

Out[11]:
```
   favorite_count  followers_count  retweet_count              source  \
0           35705          8855809           7548  Twitter for iPhone
1           30873          8855809           5593  Twitter for iPhone
2           23208          8855809           3705  Twitter for iPhone
3           39002          8855809           7731  Twitter for iPhone
4           37257          8855809           8332  Twitter for iPhone

             tweet_id
0  892420643555336193
1  892177421306343426
2  891815181378084864
```

```
3   891689557279858688
4   891327558926688256
```

1.4 2. Assess Data:

Assess Data visually and programatically

1. Assess Data visually by open files using excel program and show them here in the notebook

```
In [12]: twitter_archive_df
```

```
Out[12]:                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
        ...                  ...                    ...                  ...
        2326  666411507551481857                    NaN                  NaN
        2327  666407126856765440                    NaN                  NaN
        2328  666396247373291520                    NaN                  NaN
        2329  666373753744588802                    NaN                  NaN
        2330  666362758909284353                    NaN                  NaN
        2331  666353288456101888                    NaN                  NaN
```

2332	666345417576210432	NaN	NaN
2333	666337882303524864	NaN	NaN
2334	666293911632134144	NaN	NaN
2335	666287406224695296	NaN	NaN
2336	666273097616637952	NaN	NaN
2337	666268910803644416	NaN	NaN
2338	666104133288665088	NaN	NaN
2339	666102155909144576	NaN	NaN
2340	666099513787052032	NaN	NaN
2341	666094000022159362	NaN	NaN
2342	666082916733198337	NaN	NaN
2343	666073100786774016	NaN	NaN
2344	666071193221509120	NaN	NaN
2345	666063827256086533	NaN	NaN
2346	666058600524156928	NaN	NaN
2347	666057090499244032	NaN	NaN
2348	666055525042405380	NaN	NaN
2349	666051853826850816	NaN	NaN
2350	666050758794694657	NaN	NaN
2351	666049248165822465	NaN	NaN
2352	666044226329800704	NaN	NaN
2353	666033412701032449	NaN	NaN
2354	666029285002620928	NaN	NaN
2355	666020888022790149	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
...                         ...
2326  2015-11-17 00:24:19 +0000
2327  2015-11-17 00:06:54 +0000
2328  2015-11-16 23:23:41 +0000
2329  2015-11-16 21:54:18 +0000
2330  2015-11-16 21:10:36 +0000
2331  2015-11-16 20:32:58 +0000
2332  2015-11-16 20:01:42 +0000
2333  2015-11-16 19:31:45 +0000
2334  2015-11-16 16:37:02 +0000
2335  2015-11-16 16:11:11 +0000
2336  2015-11-16 15:14:19 +0000
2337  2015-11-16 14:57:41 +0000
2338  2015-11-16 04:02:55 +0000
2339  2015-11-16 03:55:04 +0000
2340  2015-11-16 03:44:34 +0000
2341  2015-11-16 03:22:39 +0000
2342  2015-11-16 02:38:37 +0000
2343  2015-11-16 01:59:36 +0000
2344  2015-11-16 01:52:02 +0000
2345  2015-11-16 01:22:45 +0000
2346  2015-11-16 01:01:59 +0000
2347  2015-11-16 00:55:59 +0000
2348  2015-11-16 00:49:46 +0000
2349  2015-11-16 00:35:11 +0000
2350  2015-11-16 00:30:50 +0000
2351  2015-11-16 00:24:50 +0000
2352  2015-11-16 00:04:52 +0000
2353  2015-11-15 23:21:54 +0000
2354  2015-11-15 23:05:30 +0000
2355  2015-11-15 22:32:08 +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...
...                                                 ...
2326   <a href="http://twitter.com/download/iphone" r...
2327   <a href="http://twitter.com/download/iphone" r...
2328   <a href="http://twitter.com/download/iphone" r...
2329   <a href="http://twitter.com/download/iphone" r...
2330   <a href="http://twitter.com/download/iphone" r...
2331   <a href="http://twitter.com/download/iphone" r...
2332   <a href="http://twitter.com/download/iphone" r...
2333   <a href="http://twitter.com/download/iphone" r...
2334   <a href="http://twitter.com/download/iphone" r...
2335   <a href="http://twitter.com/download/iphone" r...
2336   <a href="http://twitter.com/download/iphone" r...
2337   <a href="http://twitter.com/download/iphone" r...
2338   <a href="http://twitter.com/download/iphone" r...
2339   <a href="http://twitter.com/download/iphone" r...
2340   <a href="http://twitter.com/download/iphone" r...
2341   <a href="http://twitter.com/download/iphone" r...
2342   <a href="http://twitter.com/download/iphone" r...
2343   <a href="http://twitter.com/download/iphone" r...
2344   <a href="http://twitter.com/download/iphone" r...
2345   <a href="http://twitter.com/download/iphone" r...
2346   <a href="http://twitter.com/download/iphone" r...
2347   <a href="http://twitter.com/download/iphone" r...
2348   <a href="http://twitter.com/download/iphone" r...
2349   <a href="http://twitter.com/download/iphone" r...
```

```
2350  <a href="http://twitter.com/download/iphone" r...
2351  <a href="http://twitter.com/download/iphone" r...
2352  <a href="http://twitter.com/download/iphone" r...
2353  <a href="http://twitter.com/download/iphone" r...
2354  <a href="http://twitter.com/download/iphone" r...
2355  <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
...                                                 ...                  ...
2326  This is quite the dog. Gets really excited whe...                  NaN
2327  This is a southern Vesuvius bumblegruff. Can d...                  NaN
2328  Oh goodness. A super rare northeast Qdoba kang...                  NaN
2329  Those are sunglasses and a jean jacket. 11/10 ...                  NaN
2330  Unique dog here. Very small. Lives in containe...                  NaN
2331  Here we have a mixed Asiago from the Galápagos...                  NaN
2332  Look at this jokester thinking seat belt laws ...                  NaN
2333  This is an extremely rare horned Parthenon. No...                  NaN
2334  This is a funny dog. Weird toes. Won't come do...                  NaN
```

```
2335  This is an Albanian 3 1/2 legged  Episcopalian...                   NaN
2336     Can take selfies 11/10 https://t.co/ws2AMaNwPW                   NaN
2337  Very concerned about fellow dog trapped in com...                   NaN
2338  Not familiar with this breed. No tail (weird)...                  NaN
2339  Oh my. Here you are seeing an Adobe Setter giv...                   NaN
2340  Can stand on stump for what seems like a while...                   NaN
2341  This appears to be a Mongolian Presbyterian mi...                   NaN
2342  Here we have a well-established sunblockerspan...                   NaN
2343  Let's hope this flight isn't Malaysian (lol). ...                   NaN
2344  Here we have a northern speckled Rhododendron...                  NaN
2345  This is the happiest dog you will ever see. Ve...                   NaN
2346  Here is the Rand Paul of retrievers folks! He'...                   NaN
2347  My oh my. This is a rare blond Canadian terrie...                   NaN
2348  Here is a Siberian heavily armored polar bear ...                   NaN
2349  This is an odd dog. Hard on the outside but lo...                   NaN
2350  This is a truly beautiful English Wilson Staff...                   NaN
2351  Here we have a 1949 1st generation vulpix. Enj...                   NaN
2352  This is a purebred Piers Morgan. Loves to Netf...                   NaN
2353  Here is a very happy pup. Big fan of well-main...                   NaN
2354  This is a western brown Mitsubishi terrier. Up...                   NaN
2355  Here we have a Japanese Irish Setter. Lost eye...                   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
...                         ...                          ...
2326                        NaN                          NaN
2327                        NaN                          NaN
2328                        NaN                          NaN
2329                        NaN                          NaN
2330                        NaN                          NaN
2331                        NaN                          NaN
2332                        NaN                          NaN
2333                        NaN                          NaN
2334                        NaN                          NaN
2335                        NaN                          NaN
2336                        NaN                          NaN
2337                        NaN                          NaN
2338                        NaN                          NaN
2339                        NaN                          NaN
2340                        NaN                          NaN
2341                        NaN                          NaN
2342                        NaN                          NaN
2343                        NaN                          NaN
2344                        NaN                          NaN
2345                        NaN                          NaN
2346                        NaN                          NaN
2347                        NaN                          NaN
2348                        NaN                          NaN
2349                        NaN                          NaN
2350                        NaN                          NaN
2351                        NaN                          NaN
2352                        NaN                          NaN
2353                        NaN                          NaN
2354                        NaN                          NaN
2355                        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
...	...	...
2326	https://twitter.com/dog_rates/status/666411507...	2
2327	https://twitter.com/dog_rates/status/666407126...	7
2328	https://twitter.com/dog_rates/status/666396247...	9
2329	https://twitter.com/dog_rates/status/666373753...	11
2330	https://twitter.com/dog_rates/status/666362758...	6
2331	https://twitter.com/dog_rates/status/666353288...	8
2332	https://twitter.com/dog_rates/status/666345417...	10
2333	https://twitter.com/dog_rates/status/666337882...	9
2334	https://twitter.com/dog_rates/status/666293911...	3
2335	https://twitter.com/dog_rates/status/666287406...	1
2336	https://twitter.com/dog_rates/status/666273097...	11
2337	https://twitter.com/dog_rates/status/666268910...	10
2338	https://twitter.com/dog_rates/status/666104133...	1
2339	https://twitter.com/dog_rates/status/666102155...	11
2340	https://twitter.com/dog_rates/status/666099513...	8
2341	https://twitter.com/dog_rates/status/666094000...	9
2342	https://twitter.com/dog_rates/status/666082916...	6
2343	https://twitter.com/dog_rates/status/666073100...	10
2344	https://twitter.com/dog_rates/status/666071193...	9
2345	https://twitter.com/dog_rates/status/666063827...	10
2346	https://twitter.com/dog_rates/status/666058600...	8
2347	https://twitter.com/dog_rates/status/666057090...	9
2348	https://twitter.com/dog_rates/status/666055525...	10
2349	https://twitter.com/dog_rates/status/666051853...	2
2350	https://twitter.com/dog_rates/status/666050758...	10
2351	https://twitter.com/dog_rates/status/666049248...	5
2352	https://twitter.com/dog_rates/status/666044226...	6

```
2353  https://twitter.com/dog_rates/status/666033412...                 9
2354  https://twitter.com/dog_rates/status/666029285...                 7
2355  https://twitter.com/dog_rates/status/666020888...                 8

      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
...                  ...       ...    ...     ...     ...    ...
2326                  10     quite   None    None    None   None
2327                  10         a   None    None    None   None
2328                  10      None   None    None    None   None
2329                  10      None   None    None    None   None
2330                  10      None   None    None    None   None
2331                  10      None   None    None    None   None
2332                  10      None   None    None    None   None
2333                  10        an   None    None    None   None
2334                  10         a   None    None    None   None
2335                   2        an   None    None    None   None
2336                  10      None   None    None    None   None
2337                  10      None   None    None    None   None
```

```
2338                    10      None   None    None    None   None
2339                    10      None   None    None    None   None
2340                    10      None   None    None    None   None
2341                    10      None   None    None    None   None
2342                    10      None   None    None    None   None
2343                    10      None   None    None    None   None
2344                    10      None   None    None    None   None
2345                    10       the   None    None    None   None
2346                    10       the   None    None    None   None
2347                    10         a   None    None    None   None
2348                    10         a   None    None    None   None
2349                    10        an   None    None    None   None
2350                    10         a   None    None    None   None
2351                    10      None   None    None    None   None
2352                    10         a   None    None    None   None
2353                    10         a   None    None    None   None
2354                    10         a   None    None    None   None
2355                    10      None   None    None    None   None

[2356 rows x 17 columns]
```

In [13]:
```
image_predictions_df
```

Out[13]:
```
                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
5     666050758794694657  https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg
6     666051853826850816  https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg
7     666055525042405380  https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg
8     666057090499244032  https://pbs.twimg.com/media/CT5PY90WoAAQGLo.jpg
9     666058600524156928  https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.jpg
10    666063827256086533  https://pbs.twimg.com/media/CT5Vg_wXIAAXfnj.jpg
11    666071193221509120  https://pbs.twimg.com/media/CT5cN_3WEAAlOoZ.jpg
12    666073100786774016  https://pbs.twimg.com/media/CT5d9DZXAAALcwe.jpg
13    666082916733198337  https://pbs.twimg.com/media/CT5m4VGWEAAtKc8.jpg
14    666094000022159362  https://pbs.twimg.com/media/CT5w9gUW4AAsBNN.jpg
15    666099513787052032  https://pbs.twimg.com/media/CT51-JJUEAA6hV8.jpg
16    666102155909144576  https://pbs.twimg.com/media/CT54YGiWUAEZnoK.jpg
17    666104133288665088  https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg
18    666268910803644416  https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg
19    666273097616637952  https://pbs.twimg.com/media/CT8T1mtUwAA3aqm.jpg
20    666287406224695296  https://pbs.twimg.com/media/CT8g3BpUEAAuFjg.jpg
21    666293911632134144  https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg
22    666337882303524864  https://pbs.twimg.com/media/CT9OwFIWEAMuRje.jpg
23    666345417576210432  https://pbs.twimg.com/media/CT9Vn7PWoAA_ZCM.jpg
```

```
24    666353288456101888    https://pbs.twimg.com/media/CT9cx0tUEAAhNN_.jpg
25    666362758909284353    https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg
26    666373753744588802    https://pbs.twimg.com/media/CT9vZEYWUAAlZ05.jpg
27    666396247373291520    https://pbs.twimg.com/media/CT-D2ZHWIAA3gK1.jpg
28    666407126856765440    https://pbs.twimg.com/media/CT-NvwmW4AAugGZ.jpg
29    666411507551481857    https://pbs.twimg.com/media/CT-RugiWIAELEaq.jpg
...                  ...                                                ...
2045  886366144734445568    https://pbs.twimg.com/media/DE0BTnQUwAApKEH.jpg
2046  886680336477933568    https://pbs.twimg.com/media/DE4fEDzWAAAyHMM.jpg
2047  886736880519319552    https://pbs.twimg.com/media/DE5Se8FXcAAJFx4.jpg
2048  886983233522544640    https://pbs.twimg.com/media/DE8yicJW0AAAvBJ.jpg
2049  887101392804085760    https://pbs.twimg.com/media/DE-eAq6UwAA-jaE.jpg
2050  887343217045368832  https://pbs.twimg.com/ext_tw_video_thumb/88734...
2051  887473957103951883    https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2052  887517139158093824  https://pbs.twimg.com/ext_tw_video_thumb/88751...
2053  887705289381826560    https://pbs.twimg.com/media/DFHDQBbXgAEqY7t.jpg
2054  888078434458587136    https://pbs.twimg.com/media/DFMWn56WsAAkA7B.jpg
2055  888202515573088257    https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2056  888554962724278272    https://pbs.twimg.com/media/DFTH_O-UQAACu20.jpg
2057  888804989199671297    https://pbs.twimg.com/media/DFWra-3VYAA2piG.jpg
2058  888917238123831296    https://pbs.twimg.com/media/DFYRgsOUQAARGhO.jpg
2059  889278841981685760  https://pbs.twimg.com/ext_tw_video_thumb/88927...
2060  889531135344209921    https://pbs.twimg.com/media/DFg_2PVW0AEHN3p.jpg
2061  889638837579907072    https://pbs.twimg.com/media/DFihzFfXsAYGDPR.jpg
2062  889665388333682689    https://pbs.twimg.com/media/DFi579UWsAAatzw.jpg
2063  889880896479866881    https://pbs.twimg.com/media/DFl99B1WsAITKsg.jpg
2064  890006608113172480    https://pbs.twimg.com/media/DFnwSY4WAAAMliS.jpg
2065  890240255349198849    https://pbs.twimg.com/media/DFrEyVuW0AAO3t9.jpg
2066  890609185150312448    https://pbs.twimg.com/media/DFwUU__XcAEpyXI.jpg
2067  890729181411237888    https://pbs.twimg.com/media/DFyBahAVwAAhUTd.jpg
2068  890971913173991426    https://pbs.twimg.com/media/DF1eOmZXUAALUcq.jpg
2069  891087950875897856    https://pbs.twimg.com/media/DF3HwyEWsAABqE6.jpg
2070  891327558926688256    https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg
2071  891689557279858688    https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg
2072  891815181378084864    https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg
2073  892177421306343426    https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg
2074  892420643555336193    https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg

      img_num                      p1   p1_conf  p1_dog  \
0           1  Welsh_springer_spaniel  0.465074    True
1           1                 redbone  0.506826    True
2           1         German_shepherd  0.596461    True
3           1     Rhodesian_ridgeback  0.408143    True
4           1      miniature_pinscher  0.560311    True
5           1    Bernese_mountain_dog  0.651137    True
6           1              box_turtle  0.933012   False
7           1                    chow  0.692517    True
8           1           shopping_cart  0.962465   False
```

9	1	miniature_poodle	0.201493	True
10	1	golden_retriever	0.775930	True
11	1	Gordon_setter	0.503672	True
12	1	Walker_hound	0.260857	True
13	1	pug	0.489814	True
14	1	bloodhound	0.195217	True
15	1	Lhasa	0.582330	True
16	1	English_setter	0.298617	True
17	1	hen	0.965932	False
18	1	desktop_computer	0.086502	False
19	1	Italian_greyhound	0.176053	True
20	1	Maltese_dog	0.857531	True
21	1	three-toed_sloth	0.914671	False
22	1	ox	0.416669	False
23	1	golden_retriever	0.858744	True
24	1	malamute	0.336874	True
25	1	guinea_pig	0.996496	False
26	1	soft-coated_wheaten_terrier	0.326467	True
27	1	Chihuahua	0.978108	True
28	1	black-and-tan_coonhound	0.529139	True
29	1	coho	0.404640	False
...	...	...	...	...
2045	1	French_bulldog	0.999201	True
2046	1	convertible	0.738995	False
2047	1	kuvasz	0.309706	True
2048	2	Chihuahua	0.793469	True
2049	1	Samoyed	0.733942	True
2050	1	Mexican_hairless	0.330741	True
2051	2	Pembroke	0.809197	True
2052	1	limousine	0.130432	False
2053	1	basset	0.821664	True
2054	1	French_bulldog	0.995026	True
2055	2	Pembroke	0.809197	True
2056	3	Siberian_husky	0.700377	True
2057	1	golden_retriever	0.469760	True
2058	1	golden_retriever	0.714719	True
2059	1	whippet	0.626152	True
2060	1	golden_retriever	0.953442	True
2061	1	French_bulldog	0.991650	True
2062	1	Pembroke	0.966327	True
2063	1	French_bulldog	0.377417	True
2064	1	Samoyed	0.957979	True
2065	1	Pembroke	0.511319	True
2066	1	Irish_terrier	0.487574	True
2067	2	Pomeranian	0.566142	True
2068	1	Appenzeller	0.341703	True
2069	1	Chesapeake_Bay_retriever	0.425595	True
2070	2	basset	0.555712	True

2071	1	paper_towel	0.170278	False
2072	1	Chihuahua	0.716012	True
2073	1	Chihuahua	0.323581	True
2074	1	orange	0.097049	False

	p2	p2_conf	p2_dog	p3 \
0	collie	0.156665	True	Shetland_sheepdog
1	miniature_pinscher	0.074192	True	Rhodesian_ridgeback
2	malinois	0.138584	True	bloodhound
3	redbone	0.360687	True	miniature_pinscher
4	Rottweiler	0.243682	True	Doberman
5	English_springer	0.263788	True	Greater_Swiss_Mountain_dog
6	mud_turtle	0.045885	False	terrapin
7	Tibetan_mastiff	0.058279	True	fur_coat
8	shopping_basket	0.014594	False	golden_retriever
9	komondor	0.192305	True	soft-coated_wheaten_terrier
10	Tibetan_mastiff	0.093718	True	Labrador_retriever
11	Yorkshire_terrier	0.174201	True	Pekinese
12	English_foxhound	0.175382	True	Ibizan_hound
13	bull_mastiff	0.404722	True	French_bulldog
14	German_shepherd	0.078260	True	malinois
15	Shih-Tzu	0.166192	True	Dandie_Dinmont
16	Newfoundland	0.149842	True	borzoi
17	cock	0.033919	False	partridge
18	desk	0.085547	False	bookcase
19	toy_terrier	0.111884	True	basenji
20	toy_poodle	0.063064	True	miniature_poodle
21	otter	0.015250	False	great_grey_owl
22	Newfoundland	0.278407	True	groenendael
23	Chesapeake_Bay_retriever	0.054787	True	Labrador_retriever
24	Siberian_husky	0.147655	True	Eskimo_dog
25	skunk	0.002402	False	hamster
26	Afghan_hound	0.259551	True	briard
27	toy_terrier	0.009397	True	papillon
28	bloodhound	0.244220	True	flat-coated_retriever
29	barracouta	0.271485	False	gar
...	...	...	...	...
2045	Chihuahua	0.000361	True	Boston_bull
2046	sports_car	0.139952	False	car_wheel
2047	Great_Pyrenees	0.186136	True	Dandie_Dinmont
2048	toy_terrier	0.143528	True	can_opener
2049	Eskimo_dog	0.035029	True	Staffordshire_bullterrier
2050	sea_lion	0.275645	False	Weimaraner
2051	Rhodesian_ridgeback	0.054950	True	beagle
2052	tow_truck	0.029175	False	shopping_cart
2053	redbone	0.087582	True	Weimaraner
2054	pug	0.000932	True	bull_mastiff
2055	Rhodesian_ridgeback	0.054950	True	beagle

2056	Eskimo_dog	0.166511	True	malamute
2057	Labrador_retriever	0.184172	True	English_setter
2058	Tibetan_mastiff	0.120184	True	Labrador_retriever
2059	borzoi	0.194742	True	Saluki
2060	Labrador_retriever	0.013834	True	redbone
2061	boxer	0.002129	True	Staffordshire_bullterrier
2062	Cardigan	0.027356	True	basenji
2063	Labrador_retriever	0.151317	True	muzzle
2064	Pomeranian	0.013884	True	chow
2065	Cardigan	0.451038	True	Chihuahua
2066	Irish_setter	0.193054	True	Chesapeake_Bay_retriever
2067	Eskimo_dog	0.178406	True	Pembroke
2068	Border_collie	0.199287	True	ice_lolly
2069	Irish_terrier	0.116317	True	Indian_elephant
2070	English_springer	0.225770	True	German_short-haired_pointer
2071	Labrador_retriever	0.168086	True	spatula
2072	malamute	0.078253	True	kelpie
2073	Pekinese	0.090647	True	papillon
2074	bagel	0.085851	False	banana

	p3_conf	p3_dog
0	0.061428	True
1	0.072010	True
2	0.116197	True
3	0.222752	True
4	0.154629	True
5	0.016199	True
6	0.017885	False
7	0.054449	False
8	0.007959	True
9	0.082086	True
10	0.072427	True
11	0.109454	True
12	0.097471	True
13	0.048960	True
14	0.075628	True
15	0.089688	True
16	0.133649	True
17	0.000052	False
18	0.079480	False
19	0.111152	True
20	0.025581	True
21	0.013207	False
22	0.102643	True
23	0.014241	True
24	0.093412	True
25	0.000461	False
26	0.206803	True

```
27    0.004577   True
28    0.173810   True
29    0.189945  False
...        ...    ...
2045  0.000076   True
2046  0.044173  False
2047  0.086346   True
2048  0.032253  False
2049  0.029705   True
2050  0.134203   True
2051  0.038915   True
2052  0.026321  False
2053  0.026236   True
2054  0.000903   True
2055  0.038915   True
2056  0.111411   True
2057  0.073482   True
2058  0.105506   True
2059  0.027351   True
2060  0.007958   True
2061  0.001498   True
2062  0.004633   True
2063  0.082981  False
2064  0.008167   True
2065  0.029248   True
2066  0.118184   True
2067  0.076507   True
2068  0.193548  False
2069  0.076902  False
2070  0.175219   True
2071  0.040836  False
2072  0.031379   True
2073  0.068957   True
2074  0.076110  False

[2075 rows x 12 columns]
```

In [14]: `twitter_api_df`

Out[14]:

```
      favorite_count  followers_count  retweet_count              source  \
0              35705          8855809           7548  Twitter for iPhone
1              30873          8855809           5593  Twitter for iPhone
2              23208          8855809           3705  Twitter for iPhone
3              39002          8855809           7731  Twitter for iPhone
4              37257          8855809           8332  Twitter for iPhone
5              18779          8855809           2794  Twitter for iPhone
6              10913          8855809           1813  Twitter for iPhone
7              60180          8855809          16893  Twitter for iPhone
```

8	25831	8855809	3848	Twitter for iPhone
9	29497	8855809	6566	Twitter for iPhone
10	28428	8855809	6559	Twitter for iPhone
11	25860	8855809	4465	Twitter for iPhone
12	44476	8855809	8955	Twitter for iPhone
13	25029	8855809	4003	Twitter for iPhone
14	14051	8855809	2022	Twitter for iPhone
15	23336	8855809	4775	Twitter for iPhone
16	26955	8855809	4019	Twitter for iPhone
17	23650	8855809	3789	Twitter for iPhone
18	18261	8855809	3101	Twitter for iPhone
19	20177	8855809	3109	Twitter for iPhone
20	28043	8855809	4835	Twitter for iPhone
21	42951	8855809	10559	Twitter for iPhone
22	63622	8855809	16115	Twitter for iPhone
23	31189	8855809	9398	Twitter for iPhone
24	28382	8855809	5346	Twitter for iPhone
25	32294	8855809	6855	Twitter for iPhone
26	11084	8855809	2866	Twitter for iPhone
27	20834	8855809	4010	Twitter for iPhone
28	19571	8855809	2836	Twitter for iPhone
29	111	8855809	4	Twitter for iPhone
...	...	...	...	...
2301	404	8855846	288	Twitter for iPhone
2302	98	8855846	31	Twitter for iPhone
2303	155	8855846	73	Twitter for iPhone
2304	171	8855846	79	Twitter for iPhone
2305	712	8855846	505	Twitter for iPhone
2306	194	8855846	64	Twitter for iPhone
2307	271	8855846	128	Twitter for iPhone
2308	180	8855846	82	Twitter for iPhone
2309	459	8855846	313	Twitter for iPhone
2310	133	8855846	58	Twitter for iPhone
2311	159	8855846	70	Twitter for iPhone
2312	95	8855846	32	Twitter for iPhone
2313	13436	8855846	5884	Twitter for iPhone
2314	70	8855846	11	Twitter for iPhone
2315	142	8855846	58	Twitter for iPhone
2316	153	8855846	66	Twitter for iPhone
2317	102	8855846	41	Twitter for iPhone
2318	292	8855846	142	Twitter for iPhone
2319	135	8855846	52	Twitter for iPhone
2320	444	8855846	193	Twitter for iPhone
2321	104	8855846	52	Twitter for iPhone
2322	265	8855846	122	Twitter for iPhone
2323	406	8855846	216	Twitter for iPhone
2324	1110	8855846	759	Twitter for iPhone
2325	123	8855846	51	Twitter for iPhone

2326	96	8855846	40	Twitter for iPhone
2327	266	8855846	126	Twitter for iPhone
2328	111	8855846	39	Twitter for iPhone
2329	120	8855846	41	Twitter for iPhone
2330	2380	8855846	456	Twitter for iPhone

	tweet_id
0	892420643555336193
1	892177421306343426
2	891815181378084864
3	891689557279858688
4	891327558926688256
5	891087950875897856
6	890971913173991426
7	890729181411237888
8	890609185150312448
9	890240255349198849
10	890006608113172480
11	889880896479866881
12	889665388333682689
13	889638837579907072
14	889531135344209921
15	889278841981685760
16	888917238123831296
17	888804989199671297
18	888554962724278272
19	888078434458587136
20	887705289381826560
21	887517139158093824
22	887473957103951883
23	887343217045368832
24	887101392804085760
25	886983233522544640
26	886736880519319552
27	886680336477933568
28	886366144734445568
29	886267009285017600
...	...
2301	666411507551481857
2302	666407126856765440
2303	666396247373291520
2304	666373753744588802
2305	666362758909284353
2306	666353288456101888
2307	666345417576210432
2308	666337882303524864
2309	666293911632134144
2310	666287406224695296

```
2311  666273097616637952
2312  666268910803644416
2313  666104133288665088
2314  666102155909144576
2315  666099513787052032
2316  666094000022159362
2317  666082916733198337
2318  666073100786774016
2319  666071193221509120
2320  666063827256086533
2321  666058600524156928
2322  666057090499244032
2323  666055525042405380
2324  666051853826850816
2325  666050758794694657
2326  666049248165822465
2327  666044226329800704
2328  666033412701032449
2329  666029285002620928
2330  666020888022790149

[2331 rows x 5 columns]
```

Assess Data Programatically

```
In [15]: twitter_archive_df.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 [16]: twitter_api_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2331 entries, 0 to 2330
Data columns (total 5 columns):
favorite_count     2331 non-null int64
followers_count    2331 non-null int64
retweet_count      2331 non-null int64
source             2331 non-null object
tweet_id           2331 non-null int64
dtypes: int64(4), object(1)
memory usage: 91.1+ KB
```

```
In [17]: image_predictions_df.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 [18]: twitter_archive_df.shape

Out[18]: (2356, 17)

In [19]: twitter_api_df.shape

Out[19]: (2331, 5)

In [20]: image_predictions_df.shape

Out[20]: (2075, 12)
```

In [21]: twitter_archive_df.name.value_counts()

```
Out[21]: None            745
         a                55
         Charlie          12
         Cooper           11
         Oliver           11
         Lucy             11
         Penny            10
         Lola             10
         Tucker           10
         Winston           9
         Bo                9
         Sadie             8
         the               8
         Daisy             7
         Bailey            7
         an                7
         Buddy             7
         Toby              7
         Stanley           6
         Scout             6
         Koda              6
         Rusty             6
         Jax               6
         Dave              6
         Leo               6
         Milo              6
         Bella             6
         Jack              6
         Oscar             6
         Oakley            5
                        ...
         Rooney            1
         Danny             1
         Katie             1
         Marlee            1
         Margo             1
         Fwed              1
         Enchilada         1
         Jockson           1
         Rudy              1
         Cleopatricia      1
         Brownie           1
         Tove              1
         Lizzie            1
         Nida              1
         Kramer            1
```

```
Trevith           1
Asher             1
Henry             1
Glacier           1
Dixie             1
Hamrick           1
Simba             1
Todo              1
Kloey             1
Eevee             1
Wesley            1
Jameson           1
Toffee            1
Amber             1
Ulysses           1
Name: name, Length: 957, dtype: int64
```

In [22]:
```
twitter_archive_df[twitter_archive_df.name == 'O']
```

Out[22]:
```
                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
775   776201521193218049                    NaN                  NaN

                      timestamp  \
775   2016-09-14 23:30:38 +0000

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

                                                   text  retweeted_status_id  \
775   This is O'Malley. That is how he sleeps. Doesn...                  NaN

     retweeted_status_user_id retweeted_status_timestamp  \
775                       NaN                        NaN

                                          expanded_urls  rating_numerator  \
775   https://twitter.com/dog_rates/status/776201521...                10

     rating_denominator name doggo floofer pupper puppo
775                  10    O  None    None   None  None
```

In [23]:
```
twitter_archive_df.expanded_urls.isnull().sum()
```

Out[23]:
```
59
```

In [24]:
```
twitter_archive_df.retweeted_status_id.notnull().sum()
```

Out[24]:
```
181
```

In [25]:
```
twitter_archive_df['rating_numerator'].value_counts()
```

Out[25]:
```
12      558
11      464
10      461
13      351
9       158
8       102
7        55
14       54
5        37
6        32
3        19
4        17
1         9
2         9
420       2
0         2
15        2
75        2
80        1
20        1
24        1
26        1
44        1
50        1
60        1
165       1
84        1
88        1
144       1
182       1
143       1
666       1
960       1
1776      1
17        1
27        1
45        1
99        1
121       1
204       1
Name: rating_numerator, dtype: int64
```

In [26]:
```python
twitter_archive_df['rating_denominator'].value_counts()
```

Out[26]:
```
10     2333
11        3
50        3
80        2
```

```
20        2
2         1
16        1
40        1
70        1
15        1
90        1
110       1
120       1
130       1
150       1
170       1
7         1
0         1
Name: rating_denominator, dtype: int64
```

In [27]:
```
twitter_archive_df['rating_numerator'].describe()
```

Out[27]:
```
count    2356.000000
mean       13.126486
std        45.876648
min         0.000000
25%        10.000000
50%        11.000000
75%        12.000000
max      1776.000000
Name: rating_numerator, dtype: float64
```

In [28]:
```
twitter_archive_df['rating_denominator'].describe()
```

Out[28]:
```
count    2356.000000
mean       10.455433
std         6.745237
min         0.000000
25%        10.000000
50%        10.000000
75%        10.000000
max       170.000000
Name: rating_denominator, dtype: float64
```

In [29]:
```
twitter_archive_df['rating_numerator'].dtype
```

Out[29]:
```
dtype('int64')
```

In [30]:
```
image_predictions_df.shape
```

Out[30]:
```
(2075, 12)
```

In [31]:
```
image_predictions_df['p1'].value_counts()
```

```
Out[31]: golden_retriever             150
         Labrador_retriever           100
         Pembroke                      89
         Chihuahua                     83
         pug                           57
         chow                          44
         Samoyed                       43
         toy_poodle                    39
         Pomeranian                    38
         cocker_spaniel                30
         malamute                      30
         French_bulldog                26
         miniature_pinscher            23
         Chesapeake_Bay_retriever      23
         seat_belt                     22
         Staffordshire_bullterrier     20
         Siberian_husky                20
         German_shepherd               20
         web_site                      19
         Cardigan                      19
         beagle                        18
         Eskimo_dog                    18
         Shetland_sheepdog             18
         teddy                         18
         Maltese_dog                   18
         Rottweiler                    17
         Shih-Tzu                      17
         Lakeland_terrier              17
         Italian_greyhound             16
         kuvasz                        16
                                     ...
         basketball                     1
         microwave                      1
         bearskin                       1
         three-toed_sloth               1
         toilet_seat                    1
         dhole                          1
         electric_fan                   1
         four-poster                    1
         shopping_basket                1
         dining_table                   1
         convertible                    1
         walking_stick                  1
         rotisserie                     1
         quilt                          1
         crash_helmet                   1
         swab                           1
         hay                            1
```

```
zebra                         1
espresso                      1
pole                          1
groenendael                   1
bib                           1
shield                        1
sulphur-crested_cockatoo      1
American_black_bear           1
peacock                       1
microphone                    1
loupe                         1
wooden_spoon                  1
clumber                       1
Name: p1, Length: 378, dtype: int64
```

In [32]:
```
image_predictions_df['p2'].value_counts()
```

Out[32]:
```
Labrador_retriever                104
golden_retriever                   92
Cardigan                           73
Chihuahua                          44
Pomeranian                         42
Chesapeake_Bay_retriever           41
French_bulldog                     41
toy_poodle                         37
cocker_spaniel                     34
miniature_poodle                   33
Siberian_husky                     33
beagle                             28
collie                             27
Eskimo_dog                         27
Pembroke                           27
kuvasz                             26
Italian_greyhound                  22
Pekinese                           21
American_Staffordshire_terrier     21
chow                               20
Samoyed                            20
malinois                           20
miniature_pinscher                 20
toy_terrier                        20
Boston_bull                        19
Norwegian_elkhound                 19
Staffordshire_bullterrier          18
Irish_terrier                      17
pug                                17
Shih-Tzu                           16
                                 ...
```

```
breakwater                         1
sweatshirt                         1
hair_slide                         1
toucan                             1
cockroach                          1
necklace                           1
volcano                            1
neck_brace                         1
assault_rifle                      1
accordion                          1
shovel                             1
table_lamp                         1
crate                              1
crutch                             1
hotdog                             1
crib                               1
white_wolf                         1
stingray                           1
home_theater                       1
komondor                           1
bathing_cap                        1
Japanese_spaniel                   1
confectionery                      1
desk                               1
hatchet                            1
shower_curtain                     1
mailbox                            1
bow                                1
laptop                             1
cloak                              1
Name: p2, Length: 405, dtype: int64
```

```
In [33]: image_predictions_df['p3'].value_counts()
```

```
Out[33]: Labrador_retriever                79
         Chihuahua                         58
         golden_retriever                  48
         Eskimo_dog                        38
         kelpie                            35
         kuvasz                            34
         Staffordshire_bullterrier         32
         chow                              32
         cocker_spaniel                    31
         beagle                            31
         Pekinese                          29
         toy_poodle                        29
         Pomeranian                        29
         Pembroke                          27
```

```
Chesapeake_Bay_retriever          27
Great_Pyrenees                    27
malamute                          26
French_bulldog                    26
American_Staffordshire_terrier    24
Cardigan                          23
pug                               23
basenji                           21
toy_terrier                       20
bull_mastiff                      20
Siberian_husky                    19
Boston_bull                       17
Shetland_sheepdog                 17
Lakeland_terrier                  16
boxer                             16
doormat                           16
                                  ..
wallet                             1
cab                                1
swimming_trunks                    1
barber_chair                       1
bulletproof_vest                   1
Kerry_blue_terrier                 1
cardoon                            1
padlock                            1
bannister                          1
affenpinscher                      1
pop_bottle                         1
partridge                          1
plastic_bag                        1
traffic_light                      1
jaguar                             1
chimpanzee                         1
crossword_puzzle                   1
plunger                            1
broccoli                           1
bib                                1
Windsor_tie                        1
grand_piano                        1
balance_beam                       1
moped                              1
barbell                            1
space_shuttle                      1
chime                              1
sunglass                           1
coral_reef                         1
loggerhead                         1
Name: p3, Length: 408, dtype: int64
```

In [34]:
```
image_predictions_df['p1'].value_counts()
```

Out[34]:
```
golden_retriever              150
Labrador_retriever            100
Pembroke                       89
Chihuahua                      83
pug                            57
chow                           44
Samoyed                        43
toy_poodle                     39
Pomeranian                     38
cocker_spaniel                 30
malamute                       30
French_bulldog                 26
miniature_pinscher             23
Chesapeake_Bay_retriever       23
seat_belt                      22
Staffordshire_bullterrier      20
Siberian_husky                 20
German_shepherd                20
web_site                       19
Cardigan                       19
beagle                         18
Eskimo_dog                     18
Shetland_sheepdog              18
teddy                          18
Maltese_dog                    18
Rottweiler                     17
Shih-Tzu                       17
Lakeland_terrier               17
Italian_greyhound              16
kuvasz                         16
                             ...
basketball                      1
microwave                       1
bearskin                        1
three-toed_sloth                1
toilet_seat                     1
dhole                           1
electric_fan                    1
four-poster                     1
shopping_basket                 1
dining_table                    1
convertible                     1
walking_stick                   1
rotisserie                      1
quilt                           1
crash_helmet                    1
```

```
swab                          1
hay                           1
zebra                         1
espresso                      1
pole                          1
groenendael                   1
bib                           1
shield                        1
sulphur-crested_cockatoo      1
American_black_bear           1
peacock                       1
microphone                    1
loupe                         1
wooden_spoon                  1
clumber                       1
Name: p1, Length: 378, dtype: int64
```

1.4.1 Quality Issues:

1. should remove tweets in twitter_archive_df and not in twitter_api_df

- twitter_archive_df

2. tweet_id of type int64 to be easily used
3. timestamp of type object
4. Many None value in the columns name, doggo, floofer pupper and puppo column
5. Inaccurate names in name column like 'a' and 'an'
6. Inaccurate names 'O' in name column
7. 59 null values for the expanded_urls column
8. Data must contain original tweets no retweets and reply (no tweets has retweeted_status_id, in_reply_to_status_id, in_reply_to_user_id)
9. Some Columns not needed in the analysis like "in_reply_to_user_id", "retweeted_status_id", "retweeted_status_user_id", "retweeted_status_timestamp"
10. numenator column of type int64 and has inaccurate data
11. dominator column has inaccurate data
12. Source column has hard to read values

- image_predictions_df

13. tweet_id of type int64
14. it has 2075 records which need to be reflected in the other df
15. Inconsistent lower case for some of the predicted bread in the predected column (p1,p2,p3)

- twitter_api_df

16. tweet_id of type int64

1.4.2 Tidiness Issues:

1. In twitter_archive_df there are four columns (doggo, floofer pupper and puppo) which are values related to one variable

2. In image_predictions_df the column names (p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf, p3_dog) not descriptive and need to be merged to three column because each 3 is values to one variable

3. The two dataframes twitter_archive_df and twitter_api_df should merged in one dataframe because it is related to one observation

1.5 3. Clean Data:

Clean Data by using the points found in assess data phase using define, code, test procedure

- Make clean copy for the original dataframe

```
In [35]: twitter_archive_df_clean = twitter_archive_df.copy()
```

```
In [36]: image_predictions_df_clean = image_predictions_df.copy()
```

```
In [37]: twitter_api_df_clean = twitter_api_df.copy()
```

1. should remove tweets in twitter_archive_df and not in twitter_api_df

1.5.1 Define

- Drop records in twitter_archive_df and not in twitter_api_df because it is missing tweets which can be deleted

1.5.2 Code

```
In [38]: twitter_archive_df_clean.shape

Out[38]: (2356, 17)
```

```
In [39]: twitter_api_df.shape

Out[39]: (2331, 5)
```

```
In [40]: extra_twitter_archive_df_clean = twitter_archive_df_clean[~twitter_archive_df_clean['tw
```

```
In [41]: extra_twitter_archive_df_clean.shape

Out[41]: (25, 17)
```

```
In [42]: twitter_archive_df_clean = twitter_archive_df_clean.drop(extra_twitter_archive_df_clean
```

1.5.3 Test

```
In [43]: twitter_archive_df_clean.shape
```

```
Out[43]: (2331, 17)
```

2. tweet_id of type int64

1.5.4 Define

- convert tweet_id column type from int64 to object -string-

1.5.5 Code

```
In [44]: twitter_archive_df_clean.tweet_id.dtypes
```

```
Out[44]: dtype('int64')
```

```
In [45]: twitter_archive_df_clean.tweet_id = twitter_archive_df_clean.tweet_id.astype(str)
```

1.5.6 Test

```
In [46]: twitter_archive_df_clean.dtypes
```

```
Out[46]: tweet_id                       object
         in_reply_to_status_id         float64
         in_reply_to_user_id           float64
         timestamp                      object
         source                         object
         text                           object
         retweeted_status_id           float64
         retweeted_status_user_id      float64
         retweeted_status_timestamp     object
         expanded_urls                  object
         rating_numerator                int64
         rating_denominator              int64
         name                           object
         doggo                          object
         floofer                        object
         pupper                         object
         puppo                          object
         dtype: object
```

3. timestamp of type object

1.5.7 Define

- convert timestamp column datatype from object to datetime

1.5.8 Code

```
In [47]: twitter_archive_df_clean.dtypes
```

```
Out[47]: tweet_id                      object
         in_reply_to_status_id        float64
         in_reply_to_user_id          float64
         timestamp                     object
         source                        object
         text                          object
         retweeted_status_id          float64
         retweeted_status_user_id     float64
         retweeted_status_timestamp    object
         expanded_urls                 object
         rating_numerator               int64
         rating_denominator             int64
         name                          object
         doggo                         object
         floofer                       object
         pupper                        object
         puppo                         object
         dtype: object
```

```
In [48]: twitter_archive_df_clean.timestamp = pd.to_datetime(twitter_archive_df_clean.timestamp)
```

1.5.9 Test

```
In [49]: twitter_archive_df_clean.dtypes
```

```
Out[49]: tweet_id                                  object
         in_reply_to_status_id                    float64
         in_reply_to_user_id                      float64
         timestamp                         datetime64[ns]
         source                                    object
         text                                      object
         retweeted_status_id                      float64
         retweeted_status_user_id                 float64
         retweeted_status_timestamp                object
         expanded_urls                             object
         rating_numerator                           int64
         rating_denominator                         int64
         name                                      object
         doggo                                     object
         floofer                                   object
         pupper                                    object
         puppo                                     object
         dtype: object
```

4. Many None value in the columns name, doggo, floofer pupper and puppo column

1.5.10 Define

- convert "None value for the column name to np.nan
- convert "None" value for doggo, floofer pupper and puppo to '' to be able to merge in other point in tidness

1.5.11 Code

In [50]:
```
twitter_archive_df_clean.name.value_counts()
```

Out[50]:
```
None          736
a              55
Charlie        11
Oliver         11
Cooper         11
Penny          10
Lucy           10
Tucker         10
Lola           10
Bo              9
Winston         9
the             8
Sadie           8
an              7
Toby            7
Buddy           7
Bailey          7
Daisy           7
Stanley         6
Koda            6
Jax             6
Dave            6
Bella           6
Jack            6
Leo             6
Oscar           6
Milo            6
Scout           6
Rusty           6
Louis           5
             ...
Kara            1
Danny           1
Fwed            1
Katie           1
Marlee          1
Margo           1
Jockson         1
Enchilada       1
```

```
Rudy              1
Cleopatricia      1
Brownie           1
Tove              1
Hamrick           1
Lizzie            1
Nida              1
Kramer            1
Trevith           1
Asher             1
Henry             1
Glacier           1
Dixie             1
Eevee             1
Simba             1
Todo              1
Kloey             1
Wesley            1
Jameson           1
Toffee            1
Amber             1
Rinna             1
Name: name, Length: 955, dtype: int64
```

```
In [51]: twitter_archive_df_clean['name'] = twitter_archive_df_clean['name'].replace("None", np.
```

```
In [52]: twitter_archive_df_clean['doggo'].value_counts()
```

```
Out[52]: None     2237
         doggo      94
         Name: doggo, dtype: int64
```

```
In [53]: twitter_archive_df_clean['floofer'].value_counts()
```

```
Out[53]: None       2321
         floofer      10
         Name: floofer, dtype: int64
```

```
In [54]: twitter_archive_df_clean['pupper'].value_counts()
```

```
Out[54]: None      2076
         pupper     255
         Name: pupper, dtype: int64
```

```
In [55]: twitter_archive_df_clean['puppo'].value_counts()
```

```
Out[55]: None     2301
         puppo      30
         Name: puppo, dtype: int64
```

```
In [56]: twitter_archive_df_clean['doggo'] = twitter_archive_df_clean['doggo'].replace('None', '
         twitter_archive_df_clean['floofer'] = twitter_archive_df_clean['floofer'].replace('None
         twitter_archive_df_clean['pupper'] = twitter_archive_df_clean['pupper'].replace('None',
         twitter_archive_df_clean['puppo'] = twitter_archive_df_clean['puppo'].replace('None', '
```

1.5.12 Test

```
In [57]: twitter_archive_df_clean.name.value_counts()

Out[57]: a                55
         Cooper           11
         Charlie          11
         Oliver           11
         Penny            10
         Lucy             10
         Lola             10
         Tucker           10
         Bo                9
         Winston           9
         Sadie             8
         the               8
         Buddy             7
         Toby              7
         Daisy             7
         an                7
         Bailey            7
         Leo               6
         Rusty             6
         Bella             6
         Koda              6
         Jax               6
         Oscar             6
         Jack              6
         Milo              6
         Stanley           6
         Dave              6
         Scout             6
         Oakley            5
         George            5
                          ..
         Danny             1
         Kara              1
         Fwed              1
         Jockson           1
         Katie             1
         Marlee            1
         Margo             1
         Baron             1
```

```
Cleopatricia     1
Brownie          1
Tove             1
Amber            1
Willow           1
Eevee            1
Lizzie           1
Nida             1
Kramer           1
Trevith          1
Asher            1
Henry            1
Glacier          1
Dixie            1
Wesley           1
Simba            1
Todo             1
Kloey            1
Jameson          1
Arnold           1
Toffee           1
Beemo            1
Name: name, Length: 954, dtype: int64
```

```
In [58]: twitter_archive_df_clean['doggo'].value_counts()
```

```
Out[58]:          2237
         doggo      94
         Name: doggo, dtype: int64
```

```
In [59]: twitter_archive_df_clean['floofer'].value_counts()
```

```
Out[59]:            2321
         floofer      10
         Name: floofer, dtype: int64
```

```
In [60]: twitter_archive_df_clean['puppo'].value_counts()
```

```
Out[60]:          2301
         puppo      30
         Name: puppo, dtype: int64
```

```
In [61]: twitter_archive_df_clean['pupper'].value_counts()
```

```
Out[61]:           2076
         pupper     255
         Name: pupper, dtype: int64
```

5. Inaccurate names in name column like 'a' and 'an'

1.5.13 Define

- Convert the in accurate value name 'a' and 'an' by an accurate name extracted from the text column

1.5.14 Code

```
In [62]: twitter_archive_df_clean.name.value_counts()
```

```
Out[62]: a                 55
         Cooper            11
         Charlie           11
         Oliver            11
         Penny             10
         Lucy              10
         Lola              10
         Tucker            10
         Bo                 9
         Winston            9
         Sadie              8
         the                8
         Buddy              7
         Toby               7
         Daisy              7
         an                 7
         Bailey             7
         Leo                6
         Rusty              6
         Bella              6
         Koda               6
         Jax                6
         Oscar              6
         Jack               6
         Milo               6
         Stanley            6
         Dave               6
         Scout              6
         Oakley             5
         George             5
                           ..
         Danny              1
         Kara               1
         Fwed               1
         Jockson            1
         Katie              1
         Marlee             1
         Margo              1
         Baron              1
         Cleopatricia       1
```

```
Brownie            1
Tove               1
Amber              1
Willow             1
Eevee              1
Lizzie             1
Nida               1
Kramer             1
Trevith            1
Asher              1
Henry              1
Glacier            1
Dixie              1
Wesley             1
Simba              1
Todo               1
Kloey              1
Jameson            1
Arnold             1
Toffee             1
Beemo              1
Name: name, Length: 954, dtype: int64
```

```
In [63]: name_reg_expression = re.compile('(?:name(?:d)?)\s{1}(?:is\s)?([A-Za-z]+)')
```

```
In [64]: for index, row in twitter_archive_df_clean.iterrows():
             if row['name'] == 'a' or row['name'] == 'an':
                 try:
                     text_value = row['text']
                     name_value=re.findall(name_reg_expression,text_value)[0]
                     twitter_archive_df_clean.loc[index,'name'] = twitter_archive_df_clean.loc[i
                     twitter_archive_df_clean.loc[index,'name'] = twitter_archive_df_clean.loc[i
                 except IndexError:
                     twitter_archive_df_clean.loc[index,'name'] = np.nan
```

1.5.15 Test

```
In [65]: twitter_archive_df_clean.name.value_counts()
```

```
Out[65]: Oliver            11
         Charlie           11
         Cooper            11
         Penny             10
         Lola              10
         Lucy              10
         Tucker            10
         Winston            9
         Bo                 9
         Sadie              8
```

```
the              8
Daisy            7
Toby             7
Buddy            7
Bailey           7
Scout            6
Koda             6
Rusty            6
Stanley          6
Leo              6
Jack             6
Milo             6
Oscar            6
Dave             6
Bella            6
Jax              6
Louis            5
Sunny            5
Chester          5
Gus              5
                ..
Glenn            1
Jockson          1
Kara             1
Baron            1
Dante            1
Katie            1
Marlee           1
Margo            1
Mosby            1
Rudy             1
Cleopatricia     1
Brownie          1
Willow           1
Nida             1
Kramer           1
Trevith          1
Asher            1
Henry            1
Glacier          1
Dixie            1
Arnold           1
Simba            1
Todo             1
Kloey            1
Jacob            1
Rooney           1
Toffee           1
```

```
Amber             1
Tove              1
Lizzie            1
Name: name, Length: 970, dtype: int64
```

6. Inaccurate names 'O' in name column

1.5.16 Define

- Correct the value 'O' in name column

1.5.17 Code

```
In [66]: twitter_archive_df_clean[twitter_archive_df_clean.name == 'O']
```

```
Out[66]:               tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         775  776201521193218049                    NaN                  NaN

                        timestamp                                             source  \
         775 2016-09-14 23:30:38  <a href="http://twitter.com/download/iphone" r...

                                                           text  retweeted_status_id  \
         775  This is O'Malley. That is how he sleeps. Doesn...                  NaN

              retweeted_status_user_id retweeted_status_timestamp  \
         775                       NaN                        NaN

                                                  expanded_urls  rating_numerator  \
         775  https://twitter.com/dog_rates/status/776201521...                10

              rating_denominator name doggo floofer pupper puppo
         775                  10    O
```

```
In [67]: twitter_archive_df_clean['name'] = twitter_archive_df_clean['name'].replace('O', "O'Mal
```

1.5.18 Test

```
In [68]: twitter_archive_df_clean[twitter_archive_df_clean.name == 'O']
```

```
Out[68]: Empty DataFrame
         Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text
         Index: []
```

7. 59 null values for the expanded_urls column

1.5.19 Define

- Drop rows that has null value on the column expanded_url

1.5.20 Code

```
In [69]: twitter_archive_df_clean.expanded_urls.isnull().sum()
```

```
Out[69]: 59
```

```
In [70]: twitter_archive_df_clean = twitter_archive_df_clean[twitter_archive_df_clean['expanded_
```

1.5.21 Test

```
In [71]: twitter_archive_df_clean.expanded_urls.isnull().sum()
```

```
Out[71]: 0
```

8. Data must contain original tweets no retweets and reply (no tweets has retweeted_status_id, in_reply_to_status_id, in_reply_to_user_id)

1.5.22 Define

- Drop rows that has value in retweeted_status_id , in_reply_to_user_id and in_reply_to_status_id

1.5.23 Code

```
In [72]: twitter_archive_df_clean.retweeted_status_id.notnull().sum()
```

```
Out[72]: 162
```

```
In [73]: twitter_archive_df_clean.in_reply_to_user_id.notnull().sum()
```

```
Out[73]: 23
```

```
In [74]: twitter_archive_df_clean.in_reply_to_status_id.notnull().sum()
```

```
Out[74]: 23
```

```
In [75]: twitter_archive_df_clean.shape
```

```
Out[75]: (2272, 17)
```

```
In [76]: twitter_archive_df_clean= twitter_archive_df_clean[~twitter_archive_df_clean.retweeted_
         twitter_archive_df_clean= twitter_archive_df_clean[~twitter_archive_df_clean.in_reply_t
```

1.5.24 Test

```
In [77]: twitter_archive_df_clean.retweeted_status_id.notnull().sum()
```

```
Out[77]: 0
```

```
In [78]: twitter_archive_df_clean.in_reply_to_user_id.notnull().sum()
```

```
Out[78]: 0
```

In [79]: twitter_archive_df_clean.in_reply_to_status_id.notnull().sum()

```
Out[79]: 0
```

```
In [80]: twitter_archive_df_clean.shape
```

```
Out[80]: (2087, 17)
```

9. Some Columns not needed in the analysis like "in_reply_to_user_id", "retweeted_status_id", "retweeted_status_user_id", "retweeted_status_timestamp"

1.5.25 Define

- Drop column that don't have any value no after deleted retweet and reply which are ["in_reply_to_status_id", "in_reply_to_user_id", "retweeted_status_id", "retweeted_status_user_id", "retweeted_status_timestamp"]

1.5.26 Code

```
In [81]: twitter_archive_df_clean.columns.values
```

```
Out[81]: array(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id',
                'timestamp', 'source', 'text', 'retweeted_status_id',
                'retweeted_status_user_id', 'retweeted_status_timestamp',
                'expanded_urls', 'rating_numerator', 'rating_denominator', 'name',
                'doggo', 'floofer', 'pupper', 'puppo'], dtype=object)
```

```
In [82]: twitter_archive_df_clean = twitter_archive_df_clean.drop(["in_reply_to_status_id", "in_
```

1.5.27 Test

```
In [83]: twitter_archive_df_clean.columns.values
```

```
Out[83]: array(['tweet_id', 'timestamp', 'source', 'text', 'expanded_urls',
                'rating_numerator', 'rating_denominator', 'name', 'doggo',
                'floofer', 'pupper', 'puppo'], dtype=object)
```

10. Numerator column of type int64

1.5.28 Define

- convert numerator type to float and re-extract it from the text value as there is some inaccurate data

1.5.29 Code

```
In [84]: twitter_archive_df_clean['rating_numerator'].describe()
```

```
Out[84]: count    2087.000000
         mean       12.191184
         std        40.461531
         min         0.000000
         25%        10.000000
         50%        11.000000
         75%        12.000000
         max      1776.000000
         Name: rating_numerator, dtype: float64
```

```
In [85]: twitter_archive_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2087 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2087 non-null object
timestamp             2087 non-null datetime64[ns]
source                2087 non-null object
text                  2087 non-null object
expanded_urls         2087 non-null object
rating_numerator      2087 non-null int64
rating_denominator    2087 non-null int64
name                  1447 non-null object
doggo                 2087 non-null object
floofer               2087 non-null object
pupper                2087 non-null object
puppo                 2087 non-null object
dtypes: datetime64[ns](1), int64(2), object(9)
memory usage: 212.0+ KB
```

```
In [86]: numenator_reg_expression = '(\d+\.?\d?\d?)\/\d{1,3}'
```

```
In [87]: twitter_archive_df_clean['rating_numerator'] = twitter_archive_df_clean.text.str.extrac
```

1.5.30 Test

```
In [88]: twitter_archive_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2087 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2087 non-null object
timestamp             2087 non-null datetime64[ns]
source                2087 non-null object
text                  2087 non-null object
expanded_urls         2087 non-null object
rating_numerator      2087 non-null float64
rating_denominator    2087 non-null int64
```

```
name                    1447 non-null object
doggo                   2087 non-null object
floofer                 2087 non-null object
pupper                  2087 non-null object
puppo                   2087 non-null object
dtypes: datetime64[ns](1), float64(1), int64(1), object(9)
memory usage: 212.0+ KB
```

11. Denominator column has inaccurate data

1.5.31 Define

- Correct rating_denominator values (denominator should equal 10)

1.5.32 Code

In [89]:
```
twitter_archive_df_clean['rating_denominator'].value_counts()
```

Out[89]:
```
10     2070
50        3
11        2
80        2
7         1
170       1
150       1
120       1
110       1
90        1
70        1
40        1
20        1
2         1
Name: rating_denominator, dtype: int64
```

In [90]:
```
twitter_archive_df_clean.loc[twitter_archive_df_clean['rating_denominator'] !=10, 'rati
twitter_archive_df_clean.loc[twitter_archive_df_clean['rating_denominator'] !=10, 'rati
```

1.5.33 Test

In [91]:
```
twitter_archive_df_clean.rating_denominator.value_counts()
```

Out[91]:
```
10    2087
Name: rating_denominator, dtype: int64
```

12. Source column has hard to read values

1.5.34 Define

- Drop source column as we already have same column in twitter_api_df which will merge later to this df

1.5.35 Code

```
In [92]: twitter_archive_df_clean.columns.values
```

```
Out[92]: array(['tweet_id', 'timestamp', 'source', 'text', 'expanded_urls',
                'rating_numerator', 'rating_denominator', 'name', 'doggo',
                'floofer', 'pupper', 'puppo'], dtype=object)
```

```
In [93]: twitter_archive_df_clean = twitter_archive_df_clean.drop(["source"], axis=1)
```

1.5.36 Test

```
In [94]: twitter_archive_df_clean.columns.values
```

```
Out[94]: array(['tweet_id', 'timestamp', 'text', 'expanded_urls',
                'rating_numerator', 'rating_denominator', 'name', 'doggo',
                'floofer', 'pupper', 'puppo'], dtype=object)
```

- image_predictions_df

13. tweet_id of type int64

1.5.37 Define

- convert tweet_id column type from int64 to object -string-

1.5.38 Code

```
In [95]: image_predictions_df_clean.dtypes
```

```
Out[95]: tweet_id      int64
         jpg_url      object
         img_num       int64
         p1           object
         p1_conf     float64
         p1_dog         bool
         p2           object
         p2_conf     float64
         p2_dog         bool
         p3           object
         p3_conf     float64
         p3_dog         bool
         dtype: object
```

```
In [96]: image_predictions_df_clean.tweet_id = image_predictions_df_clean.tweet_id.astype(str)
```

1.5.39 Test

```
In [97]: image_predictions_df_clean.dtypes
```

```
Out[97]: tweet_id      object
         jpg_url       object
         img_num        int64
         p1            object
         p1_conf      float64
         p1_dog          bool
         p2            object
         p2_conf      float64
         p2_dog          bool
         p3            object
         p3_conf      float64
         p3_dog          bool
         dtype: object
```

14. it has 2075 records which need to be reflected in the other df

1.5.40 Define

- Drop rows from twitter_archive_df with tweet_id that not in image_predictions_df and rows from image_predictions_df that not in twitter_archive_df

1.5.41 Code

```
In [98]: twitter_archive_df_clean.shape
```

```
Out[98]: (2087, 11)
```

```
In [99]: image_predictions_df_clean.shape
```

```
Out[99]: (2075, 12)
```

```
In [100]: extra_twitter_archive_df_clean = twitter_archive_df_clean[ ~ twitter_archive_df_clean[
```

```
In [101]: extra_twitter_archive_df_clean.shape
```

```
Out[101]: (123, 11)
```

```
In [102]: extra_image_predictions_df_clean = image_predictions_df_clean[ ~ image_predictions_df_
```

```
In [103]: extra_image_predictions_df_clean.shape
```

```
Out[103]: (111, 12)
```

```
In [104]: twitter_archive_df_clean = twitter_archive_df_clean.drop(extra_twitter_archive_df_clea
```

```
In [105]: image_predictions_df_clean = image_predictions_df_clean.drop(extra_image_predictions_d
```

1.5.42 Test

```
In [106]: twitter_archive_df_clean.shape
```

```
Out[106]: (1964, 11)
```

```
In [107]: image_predictions_df_clean.shape
```

```
Out[107]: (1964, 12)
```

15. Inconsistent lower case for some of the predicted bread in the predected column (p1,p2,p3)

1.5.43 Define

- Change values for p1 , p2, p3 column to be capitalized

1.5.44 Code

```
In [108]: image_predictions_df_clean.head()
```

```
Out[108]:              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 [109]: image_predictions_df_clean['p1'] = image_predictions_df_clean['p1'].str.capitalize()
          image_predictions_df_clean['p2'] = image_predictions_df_clean['p2'].str.capitalize()
          image_predictions_df_clean['p3'] = image_predictions_df_clean['p3'].str.capitalize()
```

1.5.45 Test

```
In [110]: image_predictions_df_clean.head()
```

```
Out[110]:              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
```

- twitter_api_df

16. tweet_id of type int64

1.5.46 Define

- convert tweet_id column type from int64 to object -string-

1.5.47 Code

```
In [111]: twitter_api_df_clean.dtypes
          twitter_api_df_clean.tweet_id = twitter_api_df_clean.tweet_id.astype(str)
```

1.5.48 Test

```
In [112]: twitter_api_df_clean.dtypes
```

```
Out[112]: favorite_count      int64
          followers_count     int64
          retweet_count       int64
          source             object
          tweet_id           object
          dtype: object
```

1.5.49 Tidiness Issues:

1. In twitter_archive_df there are four columns (doggo, floofer pupper and puppo) which are values related to one variable

1.5.50 Define

- Combine the four columns (doggo, floofer pupper and puppo) in one column and named it dog_bread

1.5.51 Code

In [113]:
```
twitter_archive_df_clean.columns.values
```

Out[113]:
```
array(['tweet_id', 'timestamp', 'text', 'expanded_urls',
       'rating_numerator', 'rating_denominator', 'name', 'doggo',
       'floofer', 'pupper', 'puppo'], dtype=object)
```

In [114]:
```
twitter_archive_df_clean['doggo'].value_counts()
```

Out[114]:
```
         1892
doggo      72
Name: doggo, dtype: int64
```

In [115]:
```
twitter_archive_df_clean['floofer'].value_counts()
```

Out[115]:
```
           1956
floofer       8
Name: floofer, dtype: int64
```

In [116]:
```
twitter_archive_df_clean['pupper'].value_counts()
```

Out[116]:
```
          1755
pupper     209
Name: pupper, dtype: int64
```

In [117]:
```
twitter_archive_df_clean['puppo'].value_counts()
```

Out[117]:
```
         1941
puppo      23
Name: puppo, dtype: int64
```

In [118]:
```
twitter_archive_df_clean['dog_bread'] = twitter_archive_df_clean['doggo'] + twitter_ar
twitter_archive_df_clean['dog_bread'].value_counts()
```

Out[118]:
```
                1662
pupper           201
doggo             62
puppo             22
doggopupper        8
floofer            7
doggopuppo         1
doggofloofer       1
Name: dog_bread, dtype: int64
```

In [119]:
```
twitter_archive_df_clean['dog_bread'] = twitter_archive_df_clean['dog_bread'].replace(
```

```
In [120]: twitter_archive_df_clean['dog_bread'].value_counts()

Out[120]: pupper          201
          doggo            62
          puppo            22
          doggopupper       8
          floofer           7
          doggopuppo        1
          doggofloofer      1
          Name: dog_bread, dtype: int64

In [121]: twitter_archive_df_clean[twitter_archive_df_clean['dog_bread'] =="doggopupper"]="mixed
          twitter_archive_df_clean[twitter_archive_df_clean['dog_bread'] =="doggopuppo"]="mixed
          twitter_archive_df_clean[twitter_archive_df_clean['dog_bread'] =="doggofloofer"]="mixe

In [122]: twitter_archive_df_clean = twitter_archive_df_clean.drop(['doggo','puppo','pupper','fl
```

1.5.52 Test

```
In [123]: twitter_archive_df_clean['dog_bread'].value_counts()

Out[123]: pupper         201
          doggo           62
          puppo           22
          mixed bread     10
          floofer          7
          Name: dog_bread, dtype: int64

In [124]: twitter_archive_df_clean.columns.values

Out[124]: array(['tweet_id', 'timestamp', 'text', 'expanded_urls',
                 'rating_numerator', 'rating_denominator', 'name', 'dog_bread'], dtype=object)
```

2. In image_predictions_df the column names (p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf, p3_dog) not descriptive and need to be merged to three column because each 3 is values to one variable

1.5.53 Define

- Rename the 9 column ('tweet_id', 'jpg_url', 'img_num','propability_1', 'confidence_1', 'dog_1','propability_2', 'confidence_2', 'dog_2','propability_3', 'confidence_3', 'dog_3') to be more describtive and merge them nto 3 columns 'propability', 'confidence', 'dog'

1.5.54 Code

```
In [125]: image_predictions_df_clean.columns.values

Out[125]: array(['tweet_id', 'jpg_url', 'img_num', 'p1', 'p1_conf', 'p1_dog', 'p2',
                 'p2_conf', 'p2_dog', 'p3', 'p3_conf', 'p3_dog'], dtype=object)
```

```
In [126]: image_predictions_df_clean.shape
```

```
Out[126]: (1964, 12)
```

```
In [127]: columns_name_list = ['tweet_id', 'jpg_url', 'img_num','propability_1', 'confidence_1',
          image_predictions_df_clean.columns = columns_name_list

          image_predictions_df_clean = pd.wide_to_long(image_predictions_df_clean, stubnames=['p
              i=['tweet_id', 'jpg_url', 'img_num'], j='propability_stage', sep="_").reset_index(
```

1.5.55 Test

```
In [128]: image_predictions_df_clean.columns.values
```

```
Out[128]: array(['tweet_id', 'jpg_url', 'img_num', 'propability_stage',
                 'propability', 'confidence', 'dog'], dtype=object)
```

```
In [129]: image_predictions_df_clean.shape
```

```
Out[129]: (5892, 7)
```

3. The two dataframes twitter_archive_df and twitter_api_df should merged in one dataframe because it is related to one observation

1.5.56 Define

- Merge twitter_archive_df_clean and twitter_api_df_clean into twitter_archive_master_clean df because the two related to the same observation

1.5.57 Code

```
In [130]: twitter_api_df_clean.shape
```

```
Out[130]: (2331, 5)
```

```
In [131]: twitter_archive_df_clean.shape
```

```
Out[131]: (1964, 8)
```

```
In [132]: twitter_archive_master_clean = pd.merge(twitter_archive_df_clean, twitter_api_df_clean
```

```
In [133]: twitter_archive_master_clean.columns.values
```

```
Out[133]: array(['tweet_id', 'timestamp', 'text', 'expanded_urls',
                 'rating_numerator', 'rating_denominator', 'name', 'dog_bread',
                 'favorite_count', 'followers_count', 'retweet_count', 'source'], dtype=object)
```

```
In [134]: twitter_archive_master_clean.shape
```

```
Out[134]: (1954, 12)
```

```
In [135]: twitter_archive_master_df_clean = twitter_archive_master_clean.reset_index()
```

-

1.5.58 Storing wrangled data

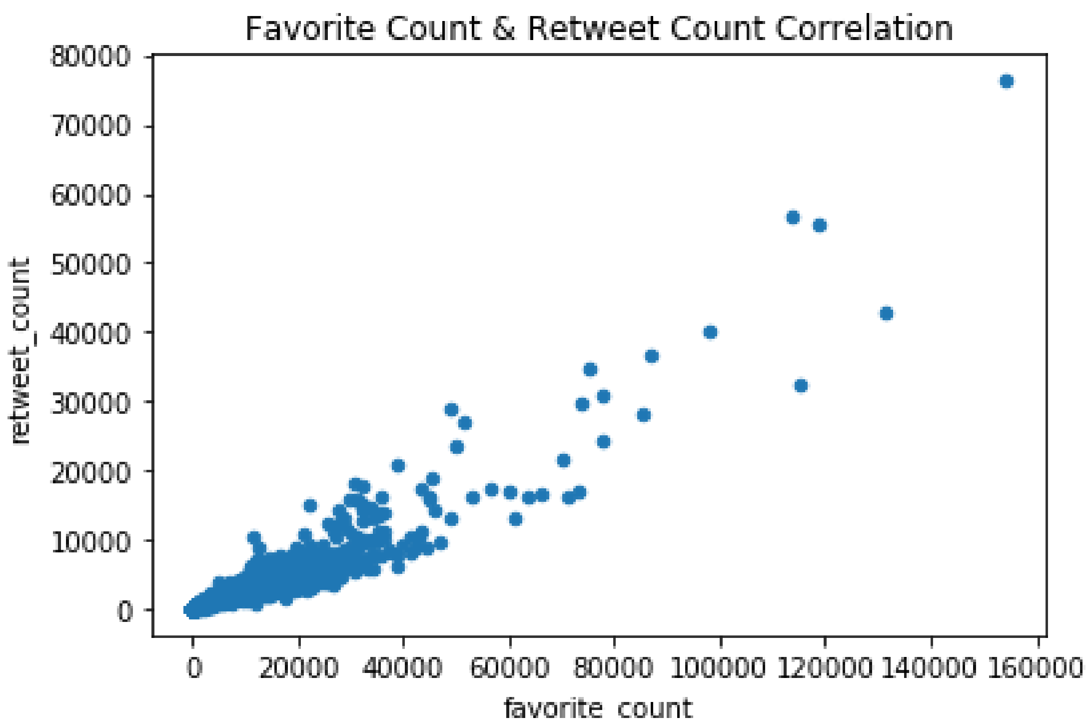
In [136]:
```
twitter_archive_master_df_clean.to_csv("twitter_archive_master.csv", index=False)
image_predictions_df_clean.to_csv("image_predictions_master.csv", index=False)
```

-

1.5.59 Analyzing and Visualizing wrangled data

1. check the coorelation between no of retweet and no of favorites

In [137]:
```
twitter_archive_master_df_clean.plot.scatter("favorite_count", "retweet_count", title=
```



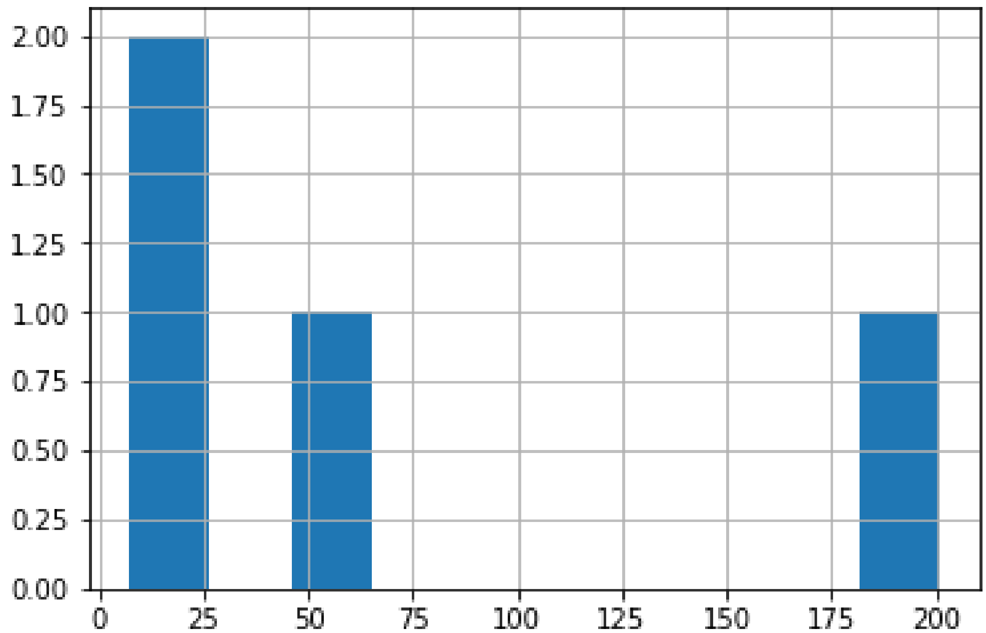
--- from graph above, we can notice that there is apositive co-relation between retweets and favorites

In [166]:
```
twitter_archive_master_df_clean['dog_bread'].value_counts()
```

Out[166]:
```
pupper     201
doggo       62
puppo       22
floofer      7
Name: dog_bread, dtype: int64
```

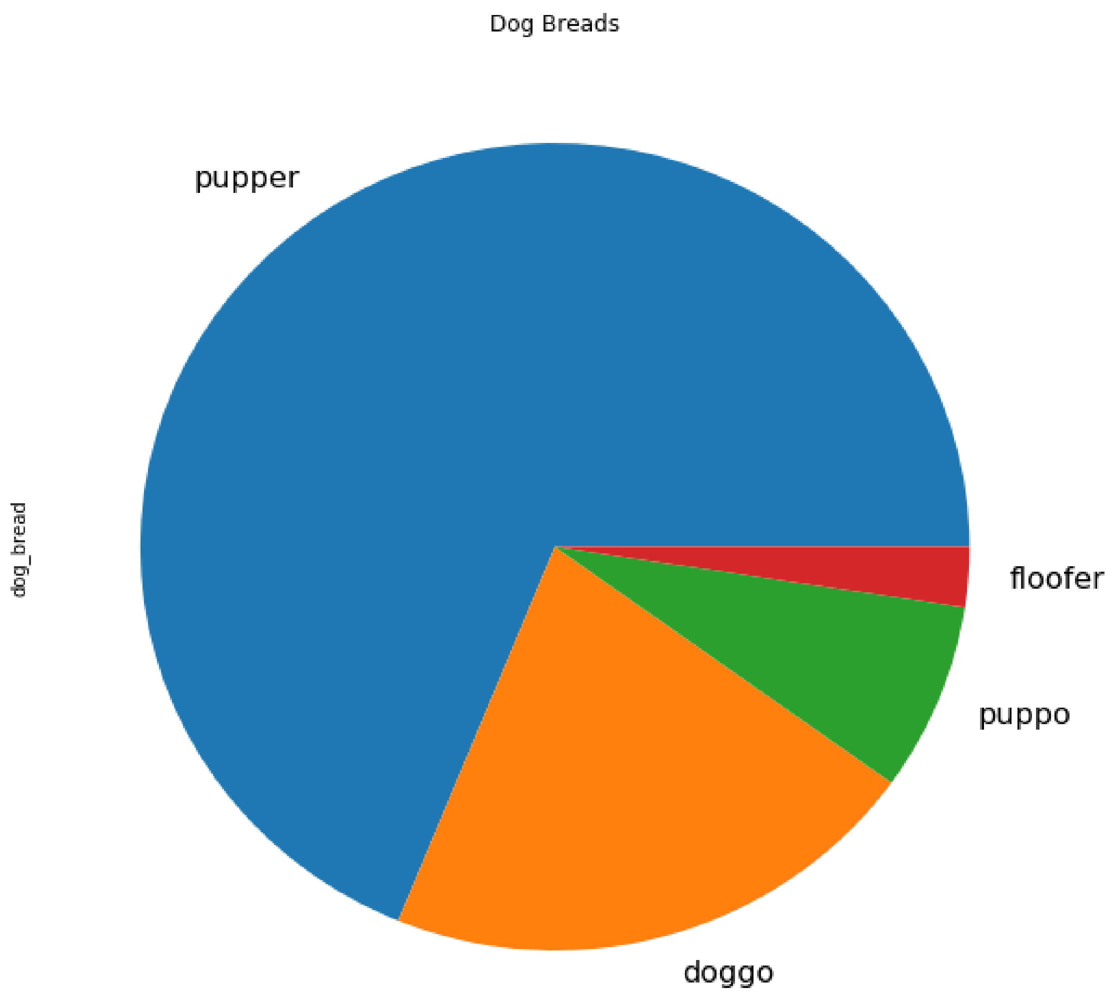
```
In [157]: twitter_archive_master_df_clean['dog_bread'].value_counts().hist()

Out[157]: <matplotlib.axes._subplots.AxesSubplot at 0x7fee373efbe0>
```




```
In [147]: twitter_archive_master_df_clean['dog_bread'].value_counts().plot(title="Dog Breads",fo

Out[147]: <matplotlib.axes._subplots.AxesSubplot at 0x7fee377bd908>
```



- From figure above we can noticed that the most popular dog bread is pupper

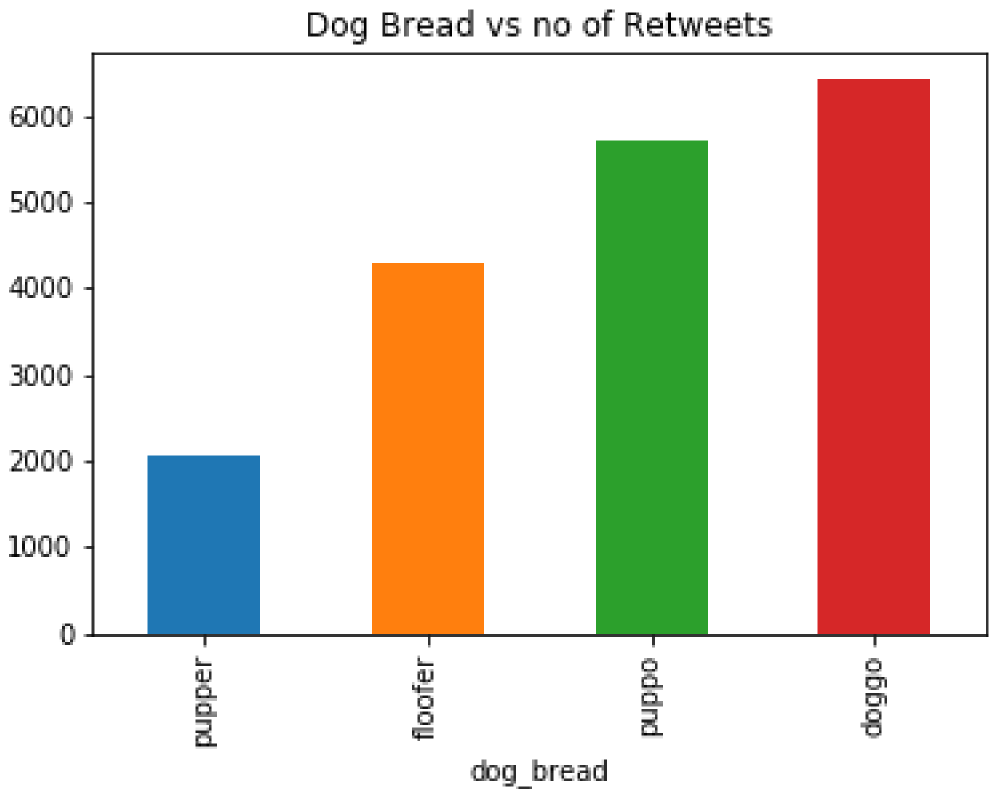
```
In [169]: twitter_archive_master_df_clean.groupby(['dog_bread']).retweet_count.mean().sort_value
```

```
Out[169]: dog_bread
          pupper     2069.611940
          floofer    4273.857143
          puppo      5717.409091
          doggo      6412.564516
          Name: retweet_count, dtype: float64
```

```
In [170]: dog_retweet_mean = twitter_archive_master_df_clean.groupby(['dog_bread']).retweet_coun
```

```
In [171]: dog_retweet_mean.plot(kind='bar', title="Dog Bread vs no of Retweets")
```

```
Out[171]: <matplotlib.axes._subplots.AxesSubplot at 0x7fee36aa2d30>
```



- from the two graphs above , we can noticed that the dog bread that got the higher retweets is duggo

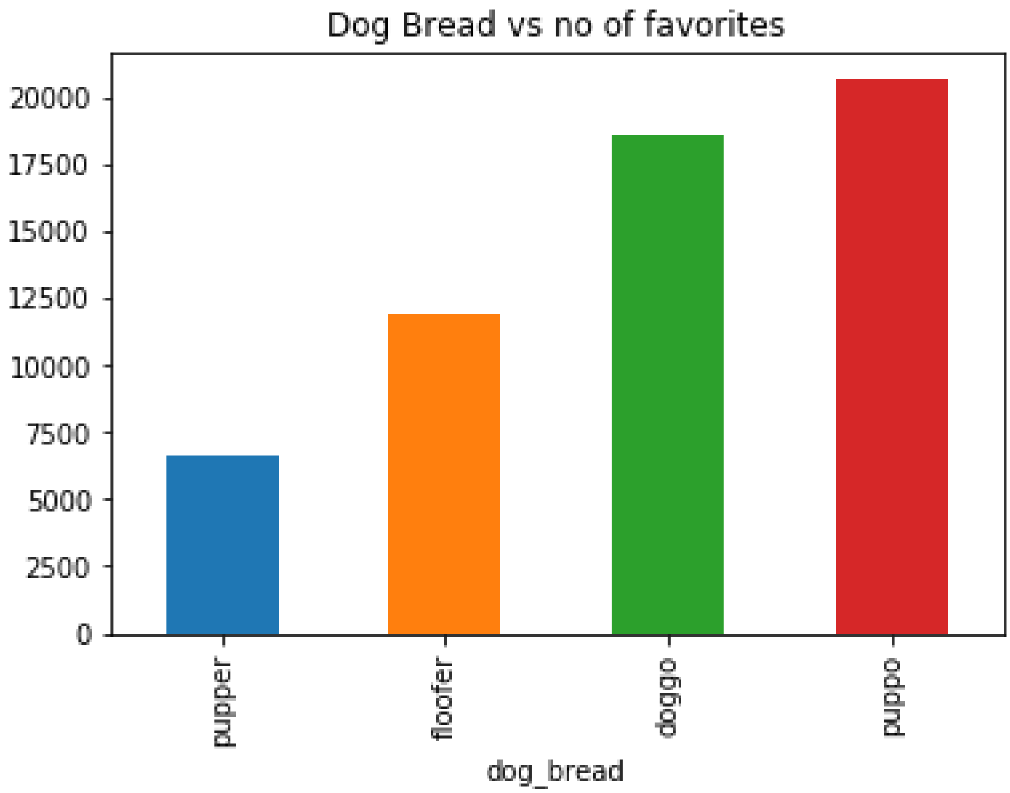
In [172]:
```
twitter_archive_master_df_clean.groupby(['dog_bread']).favorite_count.mean().sort_valu
```

Out[172]:
```
dog_bread
pupper      6625.855721
floofer    11886.857143
doggo      18632.870968
puppo      20680.000000
Name: favorite_count, dtype: float64
```

In [173]:
```
dog_favorite_mean = twitter_archive_master_df_clean.groupby(['dog_bread']).favorite_co
```

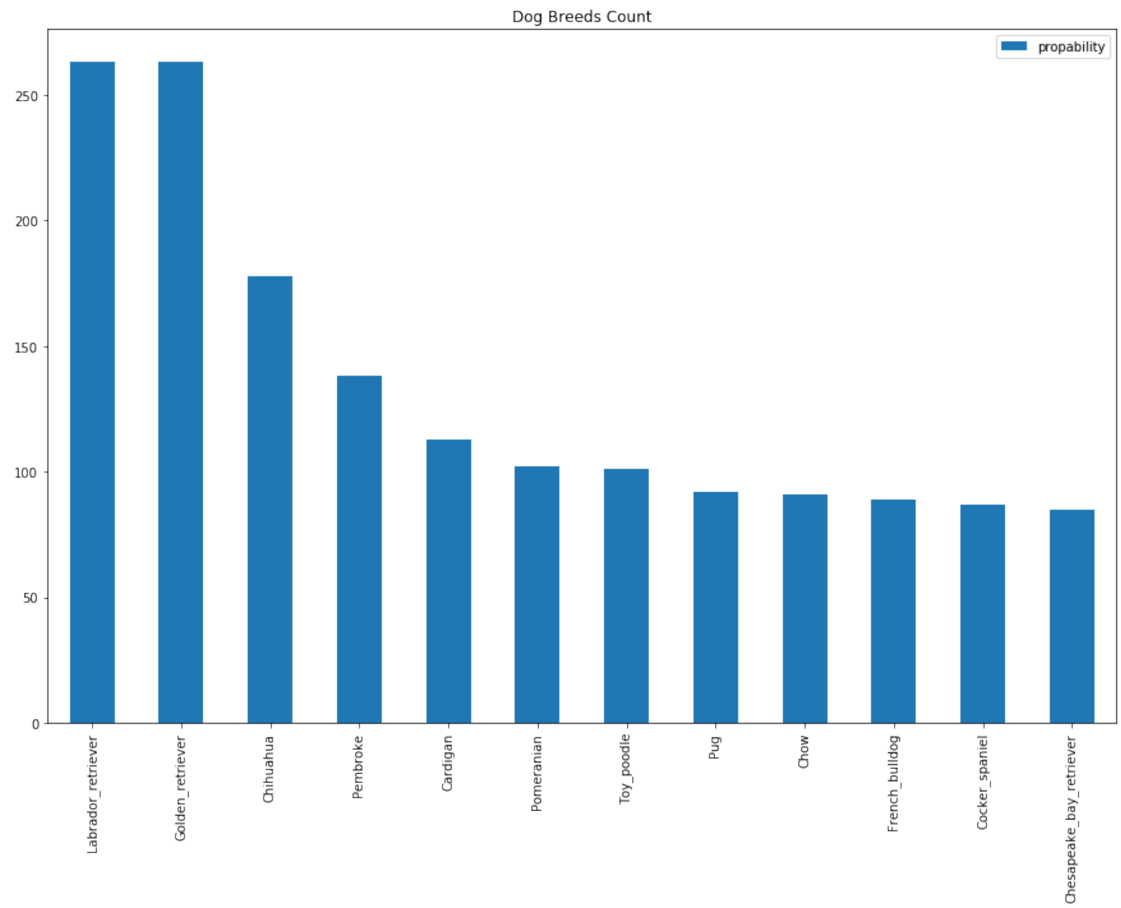
In [174]:
```
dog_favorite_mean.plot(kind='bar', title="Dog Bread vs no of favorites")
```

Out[174]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fee36768320>
```



- from the two graphs above , we can noticed that the dog bread that got the higher favorites is puppo

```
In [163]: pd.DataFrame(image_predictions_df_clean["propability"].value_counts()).nlargest(12, "p
```



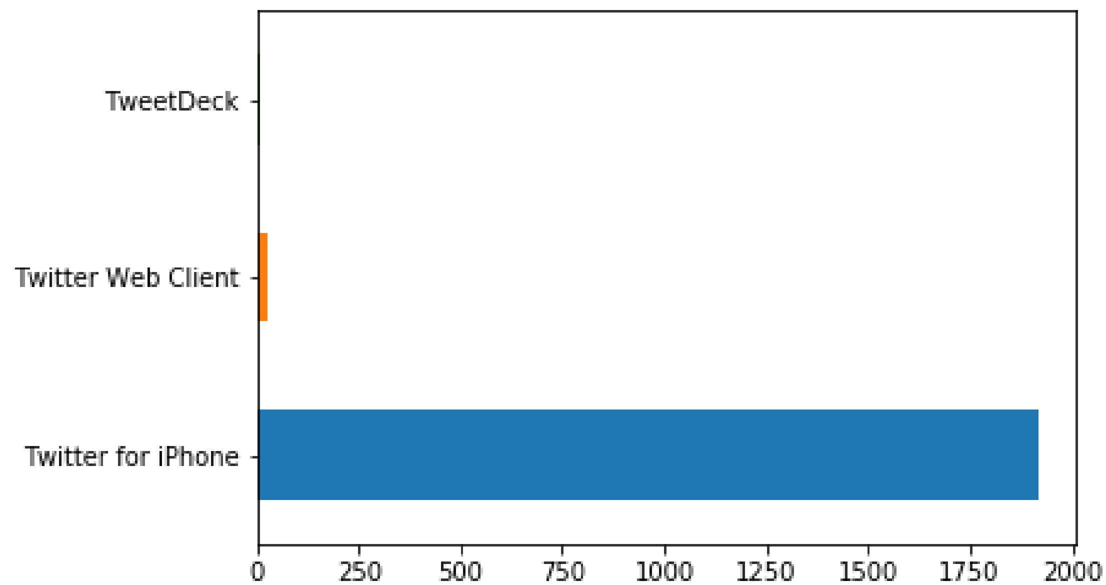
- from the above graph we can noticed that the dog bread related to highest propability is labrador retriever

In [165]:
```
twitter_archive_master_df_clean.source.value_counts()
```

Out[165]:
```
Twitter for iPhone    1916
Twitter Web Client      28
TweetDeck               10
Name: source, dtype: int64
```

In [164]:
```
twitter_archive_master_df_clean.source.value_counts().plot(kind='barh')
```

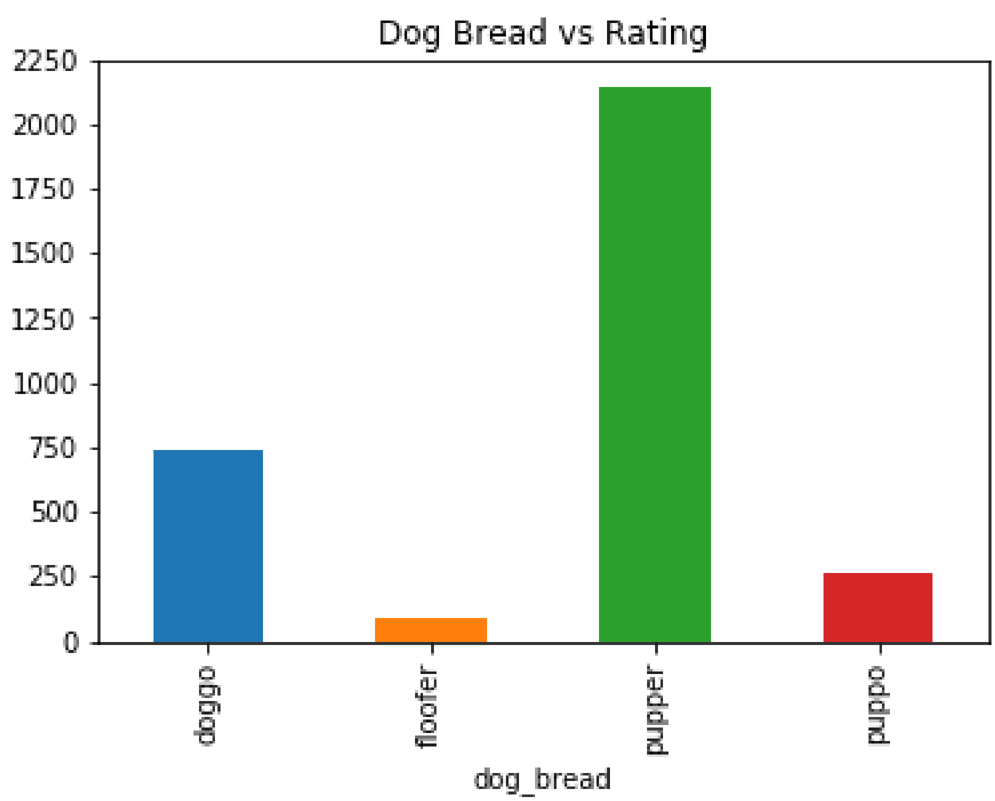
Out[164]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fee363c24e0>
```



- from the above graph and values , we noticed that the most of the user use iphone to tweet

```
In [191]: twitter_archive_master_df_clean.groupby(['dog_bread']).rating_numerator.sum().plot(kin
```

```
Out[191]: <matplotlib.axes._subplots.AxesSubplot at 0x7fee36c05c50>
```



- From the above graph we can noticed that the Dog Bread that got the heighest no of rating is pupper

```
In [ ]:
```