Table of Contents

1- Gathering the data

Importing Librairies

In [1]:

```
import pandas as pd
import numpy as py
import matplotlib.pyplot as plt
%matplotlib inline
import requests
import sys
import os
import tweepy
from tweepy import OAuthHandler
import json
from timeit import default_timer as timer
```

Importing enhanced twitter archive

In [2]:

```
t_archive = pd.read_csv(r'C:\Users\ElMehdi\Downloads\t_archive.csv')
t_archive.head()
```

Out[2]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	sc
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	href="http://twitter.com/download/iph
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://twitter.com/download/iph
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	href="http://twitter.com/download/iph

Downloading the tweet image prediction

In [3]:

```
t_archive.info()
```

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

In [4]:

```
#URL downloaded programatically
url = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/
response = requests.get(url)

with open('image-predictions.tsv', mode ='wb') as file:
    file.write(response.content)

#Read TSV file
image_prediction = pd.read_csv('image-predictions.tsv', sep='\t' )
image_prediction.head()
```

Out[4]:

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

Getting tweet data from twitter API

In [7]:

```
# Using the twitter API to get the json file for each tweet made by WeRateDogs.

# Query Twitter API for each tweet in the Twitter archive and save JSON in a text file
# These are hidden to comply with Twitter's API terms and conditions
consumer_key = 'HIDDEN'
consumer_secret = 'HIDDEN'
access_token = 'HIDDEN'
access_secret = 'HIDDEN'

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

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

# Tweet IDs for which to gather additional data via Twitter's API
tweet_id = df_1.tweet_id.values
len(tweet_id)

# Query Twitter's API for JSON data for each tweet ID in the Twitter archive
count = 0
fails_dict = {}
start = timer()
# Save each tweet's returned JSON as a new line in a .txt file
with open('tweet_json.txt', 'w') as outfile:
    # This loop will likely take 20-30 minutes to run because of Twitter's rate limit
    for tweet_id in 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")
            fails_dict[tweet_id] = e
            pass
end = timer()
print(end - start)
print(fails_dict)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-6f906a9b14f3> in <module>
     19 # Twitter API code was sent to this student from a Udacity instructo
r
     20 # Tweet IDs for which to gather additional data via Twitter's API
---> 21 tweet_id = df_1.tweet_id.values
     22 len(tweet_id)
     23

NameError: name 'df_1' is not defined
```

In [5]:

```
# converting the txt file to data list where each element (line) contains one place of twee
#download data from twitter API
tweet_status = pd.read_json(r'C:\Users\ElMehdi\Downloads\tweet-json\tweet-json', lines = Tr
```

2 - Assessing the data

In [6]:

```
t_archive
```

Out[6]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	href="http://twitter.com/downloa
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://twitter.com/downloa
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	href="http://twitter.com/downloa

In [7]:

```
image_prediction
```

Out[7]:

	tweet_id	jpg_url	img_num	
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_sp
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	Ger
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodes
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	mini
...	...	...	...	
2070	891327558926688256	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg	2	
2071	891689557279858688	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	1	
2072	891815181378084864	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	1	
2073	892177421306343426	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	1	
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	1	

2075 rows × 12 columns

In [8]:

```
tweet_status
```

Out[8]:

	created_at	id	id_str	full_text	truncated	display_
0	2017-08-01 16:23:56+00:00	892420643555336193	892420643555336192	This is Phineas. He's a mystical boy. Only eve...	False	
1	2017-08-01 00:17:27+00:00	892177421306343426	892177421306343424	This is Tilly. She's just checking pup on you....	False	
2	2017-07-31 00:18:03+00:00	891815181378084864	891815181378084864	This is Archie. He is a rare Norwegian Pouncin...	False	
3	2017-07-30 15:58:51+00:00	891689557279858688	891689557279858688	This is Darla. She commenced a snooze mid meal...	False	
4	2017-07-29 16:00:24+00:00	891327558926688256	891327558926688256	This is Franklin. He would like you to stop ca...	False	
...	...	...	...	...	...	
2349	2015-11-16 00:24:50+00:00	666049248165822465	666049248165822464	Here we have a 1949 1st generation vulpix. Enj...	False	
2350	2015-11-16 00:04:52+00:00	666044226329800704	666044226329800704	This is a purebred Piers Morgan. Loves to Netf...	False	
2351	2015-11-15 23:21:54+00:00	666033412701032449	666033412701032448	Here is a very happy pup. Big fan of well-main...	False	
2352	2015-11-15 23:05:30+00:00	666029285002620928	666029285002620928	This is a western brown Mitsubishi terrier. Up...	False	
2353	2015-11-15 22:32:08+00:00	666020888022790149	666020888022790144	Here we have a Japanese Irish Setter. Lost eye...	False	

2354 rows × 31 columns

In [9]:

```
t_archive.describe()
```

Out[9]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	retweeted_status_id	retweeted_s
count	2.356000e+03	7.800000e+01	7.800000e+01	1.810000e+02	
mean	7.427716e+17	7.455079e+17	2.014171e+16	7.720400e+17	
std	6.856705e+16	7.582492e+16	1.252797e+17	6.236928e+16	
min	6.660209e+17	6.658147e+17	1.185634e+07	6.661041e+17	
25%	6.783989e+17	6.757419e+17	3.086374e+08	7.186315e+17	
50%	7.196279e+17	7.038708e+17	4.196984e+09	7.804657e+17	
75%	7.993373e+17	8.257804e+17	4.196984e+09	8.203146e+17	
max	8.924206e+17	8.862664e+17	8.405479e+17	8.874740e+17	

In [10]:

```
t_archive.info()
```

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

In [11]:

```
t_archive[t_archive['rating_denominator'] == 0 ]
```

Out[11]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
313	835246439529840640	8.352460e+17	26259576.0	2017-02-24 21:54:03 +0000	href="http://twitte

In [12]:

```
t_archive['text'][t_archive['rating_denominator'] == 0 ]
```

Out[12]:

```
313    @jonnysun @Lin_Manuel ok jomny I know you're e...
Name: text, dtype: object
```

In [13]:

```
# delete this row.
t_archive[t_archive['rating_denominator'] < 10 ]
```

Out[13]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
313	835246439529840640	8.352460e+17	26259576.0	2017-02-24 21:54:03 +0000	href="http://twitt
516	810984652412424192	NaN	NaN	2016-12-19 23:06:23 +0000	href="http://twitt
2335	666287406224695296	NaN	NaN	2015-11-16 16:11:11 +0000	href="http://twitt

In [14]:

```
t_archive[t_archive['rating_denominator'] > 20 ]
```

Out[14]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
433	820690176645140481	NaN	NaN	2017-01-15 17:52:40 +0000	href="http://twitt
902	758467244762497024	NaN	NaN	2016-07-28 01:00:57 +0000	href="http://twitt
1120	731156023742988288	NaN	NaN	2016-05-13 16:15:54 +0000	href="http://twitt
1202	716439118184652801	NaN	NaN	2016-04-03 01:36:11 +0000	href="http://twitt
1228	713900603437621249	NaN	NaN	2016-03-27 01:29:02 +0000	href="http://twitt
1254	710658690886586372	NaN	NaN	2016-03-18 02:46:49 +0000	href="http://twitt
1274	709198395643068416	NaN	NaN	2016-03-14 02:04:08 +0000	href="http://twitt
1351	704054845121142784	NaN	NaN	2016-02-28 21:25:30 +0000	href="http://twitt
1433	697463031882764288	NaN	NaN	2016-02-10 16:51:59 +0000	href="http://twitt
1634	684225744407494656	6.842229e+17	4.196984e+09	2016-01-05 04:11:44 +0000	href="http://twitt
1635	684222868335505415	NaN	NaN	2016-01-05 04:00:18 +0000	href="http://twitt
1779	677716515794329600	NaN	NaN	2015-12-18 05:06:23 +0000	href="http://twitt
1843	675853064436391936	NaN	NaN	2015-12-13 01:41:41 +0000	href="http://twitt

In [15]:

```
t_archive[t_archive['rating_numerator'] == 0 ]
```

Out[15]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
315	835152434251116546	NaN	NaN	2017-02-24 15:40:31 +0000	href="http://twitt
1016	746906459439529985	7.468859e+17	4.196984e+09	2016-06-26 03:22:31 +0000	href="http://twitt

In [16]:

```
len(t_archive[t_archive['rating_numerator'] == 0 ])
```

Out[16]:

2

In [17]:

```
t_archive[t_archive['rating_numerator'] > 20]
```

Out[17]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
188	855862651834028034	8.558616e+17	1.943518e+08	2017-04-22 19:15:32 +0000	href="http://tv
189	855860136149123072	8.558585e+17	1.361572e+07	2017-04-22 19:05:32 +0000	href="http://tv
290	838150277551247360	8.381455e+17	2.195506e+07	2017-03-04 22:12:52 +0000	href="http://tv
313	835246439529840640	8.352460e+17	2.625958e+07	2017-02-24 21:54:03 +0000	href="http://tv
340	832215909146226688	NaN	NaN	2017-02-16 13:11:49 +0000	href="http://tv
433	820690176645140481	NaN	NaN	2017-01-15 17:52:40 +0000	href="http://tv
516	810984652412424192	NaN	NaN	2016-12-19 23:06:23 +0000	href="http://tv
695	786709082849828864	NaN	NaN	2016-10-13 23:23:56 +0000	href="http://tv
763	778027034220126208	NaN	NaN	2016-09-20 00:24:34 +0000	href="http://tv
902	758467244762497024	NaN	NaN	2016-07-28 01:00:57 +0000	href="http://tv
979	749981277374128128	NaN	NaN	2016-07-04 15:00:45 +0000	href="https://abc
1120	731156023742988288	NaN	NaN	2016-05-13 16:15:54 +0000	href="http://tv
1202	716439118184652801	NaN	NaN	2016-04-03 01:36:11 +0000	href="http://tv

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
1228	713900603437621249	NaN	NaN	2016-03-27 01:29:02 +0000	href="http://tv
1254	710658690886586372	NaN	NaN	2016-03-18 02:46:49 +0000	href="http://tv
1274	709198395643068416	NaN	NaN	2016-03-14 02:04:08 +0000	href="http://tv
1351	704054845121142784	NaN	NaN	2016-02-28 21:25:30 +0000	href="http://tv
1433	697463031882764288	NaN	NaN	2016-02-10 16:51:59 +0000	href="http://tv
1634	684225744407494656	6.842229e+17	4.196984e+09	2016-01-05 04:11:44 +0000	href="http://tv
1635	684222868335505415	NaN	NaN	2016-01-05 04:00:18 +0000	href="http://tv
1712	680494726643068929	NaN	NaN	2015-12-25 21:06:00 +0000	href="http://tv
1779	677716515794329600	NaN	NaN	2015-12-18 05:06:23 +0000	href="http://tv
1843	675853064436391936	NaN	NaN	2015-12-13 01:41:41 +0000	href="http://tv
2074	670842764863651840	NaN	NaN	2015-11-29 05:52:33 +0000	href="http://tv

In [18]:

```
len(t_archive[t_archive['rating_numerator'] > 20 ])
```

Out[18]:

24

In [19]:

```
image_prediction.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   tweet_id  2075 non-null   int64  
 1   jpg_url   2075 non-null   object 
 2   img_num   2075 non-null   int64  
 3   p1        2075 non-null   object 
 4   p1_conf   2075 non-null   float64
 5   p1_dog    2075 non-null   bool   
 6   p2        2075 non-null   object 
 7   p2_conf   2075 non-null   float64
 8   p2_dog    2075 non-null   bool   
 9   p3        2075 non-null   object 
 10  p3_conf   2075 non-null   float64
 11  p3_dog    2075 non-null   bool   
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

In [20]:

```
image_prediction.describe()
```

Out[20]:

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

In [21]:

```
tweet_status.describe()
```

Out[21]:

	id	id_str	in_reply_to_status_id	in_reply_to_status_id_str	in_reply_to_
count	2.354000e+03	2.354000e+03	7.800000e+01	7.800000e+01	7.8000
mean	7.426978e+17	7.426978e+17	7.455079e+17	7.455079e+17	2.0141
std	6.852812e+16	6.852812e+16	7.582492e+16	7.582492e+16	1.2527
min	6.660209e+17	6.660209e+17	6.658147e+17	6.658147e+17	1.1856
25%	6.783975e+17	6.783975e+17	6.757419e+17	6.757419e+17	3.0863
50%	7.194596e+17	7.194596e+17	7.038708e+17	7.038708e+17	4.1969
75%	7.993058e+17	7.993058e+17	8.257804e+17	8.257804e+17	4.1969
max	8.924206e+17	8.924206e+17	8.862664e+17	8.862664e+17	8.4054

In [22]:

```
[t_archive.duplicated() == False]
```

Out[22]:

```
[0       True
 1       True
 2       True
 3       True
 4       True
        ...
 2351    True
 2352    True
 2353    True
 2354    True
 2355    True
 Length: 2356, dtype: bool]
```

In [23]:

```
[image_prediction.duplicated() == False]
```

Out[23]:

```
[0       True
 1       True
 2       True
 3       True
 4       True
        ...
 2070    True
 2071    True
 2072    True
 2073    True
 2074    True
 Length: 2075, dtype: bool]
```

In [24]:

```
[tweet_status.id.duplicated() == False]
```

Out[24]:

```
[0       True
 1       True
 2       True
 3       True
 4       True
        ...
 2349    True
 2350    True
 2351    True
 2352    True
 2353    True
 Name: id, Length: 2354, dtype: bool]
```

In [25]:

```
image_prediction.p1.value_counts()
```

Out[25]:

```
golden_retriever      150
Labrador_retriever    100
Pembroke               89
Chihuahua              83
pug                    57
                     ...
convertible             1
china_cabinet           1
piggy_bank              1
long-horned_beetle      1
clog                    1
Name: p1, Length: 378, dtype: int64
```

In [26]:

```
rare_things= image_prediction.groupby('p1').filter(lambda x: len(x) < 3)
```

In [27]:

```
rare_things.sample(5)
```

Out[27]:

	tweet_id	jpg_url	img_num	
1335	758041019896193024	https://pbs.twimg.com/media/CoUaSKEXYAAYsAI.jpg	1	
1297	752309394570878976	https://pbs.twimg.com/ext_tw_video_thumb/67535...	1	
289	671163268581498880	https://pbs.twimg.com/media/CVBzbWsWsAEyNMA.jpg	1	African_
2022	881268444196462592	https://pbs.twimg.com/media/DDrk-f9WAAI-WQv.jpg	1	
664	682697186228989953	https://pbs.twimg.com/media/CXltdtaWYAIuX_V.jpg	1	

In [28]:

```
len(rare_things)
```

Out[28]:

```
271
```

In [29]:

```
len(image_prediction)
```

Out[29]:

```
2075
```

In [30]:

```
tweet_status['id'].sample(5)
```

Out[30]:

```
1847    675781562965868544
1654    683357973142474752
389     826240494070030336
2038    671544874165002241
1152    725458796924002305
Name: id, dtype: int64
```

In [31]:

```
t_archive[t_archive.tweet_id == 782722598790725632]
```

Out[31]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
725	782722598790725632	NaN	NaN	2016-10-02 23:23:04 +0000	href="http://twitte

In [32]:

```
tweet_status[tweet_status.id == 782722598790725632]
```

Out[32]:

	created_at	id	id_str	full_text	truncated	display_text
724	2016-10-02 23:23:04+00:00	782722598790725632	782722598790725632	This is Penny. She fought a bee and the bee wo...	False	

1 rows × 31 columns

In [33]:

```
# Merging the 3 data frames :
df1 = tweet_status.filter(['id','created_at'])
df1 = df1.rename(columns={'id': 'tweet_id'})
df2 = t_archive.filter(['tweet_id', 'timestamp'])

dfmerged = pd.merge(df1, df2, how='inner', on=['tweet_id'])
```

In [34]:

```
dfmerged.sample(5)
```

Out[34]:

	tweet_id	created_at	timestamp
1183	718613305783398402	2016-04-09 01:35:37+00:00	2016-04-09 01:35:37 +0000
793	773336787167145985	2016-09-07 01:47:12+00:00	2016-09-07 01:47:12 +0000
2293	667119796878725120	2015-11-18 23:18:48+00:00	2015-11-18 23:18:48 +0000
37	884925521741709313	2017-07-12 00:01:00+00:00	2017-07-12 00:01:00 +0000
255	843981021012017153	2017-03-21 00:22:10+00:00	2017-03-21 00:22:10 +0000

Assessment report

Tidiness Issues

T_archive

- Columns 'doggo', 'floofer', 'pupper', 'puppo' in t_archive should be a single column stage
- Change tweet_id to type int64 in order to merge with the other 2 tables

Tweet_status

- Join 'tweet_json' and 'image_prediction' to 't_archive'

Quality Issues

T_archive

- Delete columns that won't be used for analysis
- The datatype of "timestamp" is not correct.
- The standard for "rating_denominator" is 10 correct the same.
- The "rating_numerator" also has some incorrect values.
- The dog names format should be consistent. Make the first letter capital for all the names.

Image Prediction

- Drop duplicate values from jpg_url
- The column names such as p1,p2 are not decriptive.
- The prediction dog breeds involve both uppercase and lowercase for the first letter.

Tweet_status

- Delete columns that won't be used for analysis

3 - Data Cleaning

In [35]:

```
# Reading gathered files
t_archive = pd.read_csv(r'C:\Users\ElMehdi\Downloads\t_archive.csv')
image_prediction = pd.read_csv('image-predictions.tsv', sep='\t' )
tweet_status = pd.read_json(r'C:\Users\ElMehdi\Downloads\tweet-json\tweet-json', lines = Tr
```

In [36]:

```
# create a cleaned dataframe from the archive
cleaned_arch = t_archive.copy()
```

In [37]:

```
# test
cleaned_arch.info()
```

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

In [38]:

```
# let's exclude all the retweets
cleaned_arch = cleaned_arch[cleaned_arch['in_reply_to_status_id'].isnull()]
```

In [39]:

```
cleaned_arch.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2278 entries, 0 to 2355
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   tweet_id                    2278 non-null   int64  
 1   in_reply_to_status_id       0 non-null      float64
 2   in_reply_to_user_id         0 non-null      float64
 3   timestamp                   2278 non-null   object 
 4   source                      2278 non-null   object 
 5   text                        2278 non-null   object 
 6   retweeted_status_id         181 non-null    float64
 7   retweeted_status_user_id    181 non-null    float64
 8   retweeted_status_timestamp  181 non-null    object 
 9   expanded_urls               2274 non-null   object 
 10  rating_numerator            2278 non-null   int64  
 11  rating_denominator          2278 non-null   int64  
 12  name                        2278 non-null   object 
 13  doggo                       2278 non-null   object 
 14  floofer                     2278 non-null   object 
 15  pupper                      2278 non-null   object 
 16  puppo                       2278 non-null   object 
dtypes: float64(4), int64(3), object(10)
memory usage: 320.3+ KB
```

In [40]:

```
# Dropping unneded columns using drop function
cleaned_arch = cleaned_arch.drop(['in_reply_to_status_id', 'in_reply_to_user_id', 'retweete
```

In [41]:

```
cleaned_arch.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2278 entries, 0 to 2355
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   tweet_id            2278 non-null   int64 
 1   timestamp           2278 non-null   object
 2   source              2278 non-null   object
 3   text                2278 non-null   object
 4   expanded_urls       2274 non-null   object
 5   rating_numerator    2278 non-null   int64 
 6   rating_denominator  2278 non-null   int64 
 7   name                2278 non-null   object
 8   doggo               2278 non-null   object
 9   floofer             2278 non-null   object
 10  pupper              2278 non-null   object
 11  puppo               2278 non-null   object
dtypes: int64(3), object(9)
memory usage: 231.4+ KB
```

In [42]:

```
cleaned_arch = cleaned_arch.drop(['expanded_urls'], axis=1)
```

In [43]:

```
cleaned_arch.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2278 entries, 0 to 2355
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   tweet_id            2278 non-null   int64 
 1   timestamp           2278 non-null   object
 2   source              2278 non-null   object
 3   text                2278 non-null   object
 4   rating_numerator    2278 non-null   int64 
 5   rating_denominator  2278 non-null   int64 
 6   name                2278 non-null   object
 7   doggo               2278 non-null   object
 8   floofer             2278 non-null   object
 9   pupper              2278 non-null   object
 10  puppo               2278 non-null   object
dtypes: int64(3), object(8)
memory usage: 213.6+ KB
```

In [44]:

```
# using loc to correct the typo mistakes in specified cells

cleaned_arch.loc[cleaned_arch['rating_numerator']==50, ['rating_numerator']] = 10
cleaned_arch.loc[cleaned_arch['rating_denominator']==50, ['rating_denominator']] = 10

cleaned_arch.loc[cleaned_arch['rating_numerator']==88, ['rating_numerator']] = 11
cleaned_arch.loc[cleaned_arch['rating_denominator']==80, ['rating_denominator']] = 10

cleaned_arch.loc[cleaned_arch['rating_numerator']==80, ['rating_numerator']] = 10
cleaned_arch.loc[cleaned_arch['rating_denominator']==80, ['rating_denominator']] = 10

cleaned_arch.loc[cleaned_arch['rating_numerator']==44, ['rating_numerator']] = 11
cleaned_arch.loc[cleaned_arch['rating_denominator']==40, ['rating_denominator']] = 10
```

In [45]:

```
len(cleaned_arch[cleaned_arch['rating_numerator'] > 20 ])
```

Out[45]:

```
15
```

In [46]:

```
# let's exclude all extreme and zero values from the numertor and denominator

cleaned_arch = cleaned_arch[cleaned_arch['rating_numerator'] != 0 ]
cleaned_arch = cleaned_arch[cleaned_arch['rating_denominator'] >= 10 ]
cleaned_arch = cleaned_arch[cleaned_arch['rating_numerator'] <= 20 ]
cleaned_arch = cleaned_arch[cleaned_arch['rating_denominator'] < 20 ]
```

In [47]:

```
len(cleaned_arch[cleaned_arch['rating_numerator'] > 20 ])
```

Out[47]:

```
0
```

In [48]:

```
len(cleaned_arch[cleaned_arch['rating_denominator'] < 10 ])
```

Out[48]:

```
0
```

In [49]:

```
len(cleaned_arch[cleaned_arch['rating_denominator'] > 20 ])
```

Out[49]:

```
0
```

In [50]:

```
# merge the column doggo, floofer, pupper or poppo to column_class
cleaned_arch['column_class'] = cleaned_arch[['doggo', 'floofer', 'pupper', 'puppo']].max(ax
```

In [51]:

```
# drop the doggo, floofer, pupper and puppo columns
cleaned_arch.drop(['doggo', 'floofer', 'pupper', 'puppo'], axis=1, inplace=True)
```

In [52]:

```
cleaned_arch.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2260 entries, 0 to 2355
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   tweet_id            2260 non-null   int64 
 1   timestamp           2260 non-null   object
 2   source              2260 non-null   object
 3   text                2260 non-null   object
 4   rating_numerator    2260 non-null   int64 
 5   rating_denominator  2260 non-null   int64 
 6   name                2260 non-null   object
 7   column_class        2260 non-null   object
dtypes: int64(3), object(5)
memory usage: 158.9+ KB
```

In [53]:

```
cleaned_images = image_prediction.copy()
```

In [54]:

```
cleaned_images.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   tweet_id  2075 non-null   int64  
 1   jpg_url   2075 non-null   object 
 2   img_num   2075 non-null   int64  
 3   p1        2075 non-null   object 
 4   p1_conf   2075 non-null   float64
 5   p1_dog    2075 non-null   bool   
 6   p2        2075 non-null   object 
 7   p2_conf   2075 non-null   float64
 8   p2_dog    2075 non-null   bool   
 9   p3        2075 non-null   object 
 10  p3_conf   2075 non-null   float64
 11  p3_dog    2075 non-null   bool   
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

In [55]:

```
# renaming columns
cleaned_images= cleaned_images.rename(columns={'p1':'Breed_probability1', 'p2':'Breed_proba
```

In [56]:

```
cleaned_images.sample(5)
```

Out[56]:

	tweet_id	jpg_url	img_num	Breed_
1496	783391753726550016	https://pbs.twimg.com/media/Ct8qn8EWIAAk9zP.jpg	4	Norweg
1376	763183847194451968	https://pbs.twimg.com/media/CpdfpzKWYAAWSUi.jpg	1	min
1289	751251247299190784	https://pbs.twimg.com/ext_tw_video_thumb/75125...	1	V
1562	793500921481273345	https://pbs.twimg.com/media/CwMU34YWIAAz1nU.jpg	2	gol
561	677895101218201600	https://pbs.twimg.com/media/CWhd_7WWsAAaqWG.jpg	1	

In [57]:

```
# Exclude all rows with P1_confidence less than 0.5

cleaned_images = cleaned_images[cleaned_images['p1_conf'] > 0.5 ]
```

In [58]:

```
# filtering to select only needed columns

cleaned_images = cleaned_images.filter(['tweet_id','Breed_probability1','p1_conf'] )
```

In [59]:

```
cleaned_images.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1239 entries, 1 to 2072
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   tweet_id            1239 non-null   int64  
 1   Breed_probability1  1239 non-null   object 
 2   p1_conf             1239 non-null   float64
dtypes: float64(1), int64(1), object(1)
memory usage: 38.7+ KB
```

In [60]:

```
cleaned_status = tweet_status.copy()
```

In [61]:

```
cleaned_status.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   created_at                     2354 non-null   datetime64[ns, UTC]
 1   id                             2354 non-null   int64
 2   id_str                         2354 non-null   int64
 3   full_text                      2354 non-null   object
 4   truncated                      2354 non-null   bool
 5   display_text_range             2354 non-null   object
 6   entities                       2354 non-null   object
 7   extended_entities              2073 non-null   object
 8   source                         2354 non-null   object
 9   in_reply_to_status_id          78 non-null     float64
 10  in_reply_to_status_id_str      78 non-null     float64
 11  in_reply_to_user_id            78 non-null     float64
 12  in_reply_to_user_id_str        78 non-null     float64
 13  in_reply_to_screen_name        78 non-null     object
 14  user                           2354 non-null   object
 15  geo                            0 non-null      float64
 16  coordinates                    0 non-null      float64
 17  place                          1 non-null      object
 18  contributors                   0 non-null      float64
 19  is_quote_status                2354 non-null   bool
 20  retweet_count                  2354 non-null   int64
 21  favorite_count                 2354 non-null   int64
 22  favorited                      2354 non-null   bool
 23  retweeted                      2354 non-null   bool
 24  possibly_sensitive             2211 non-null   float64
 25  possibly_sensitive_appealable  2211 non-null   float64
 26  lang                           2354 non-null   object
 27  retweeted_status               179 non-null    object
 28  quoted_status_id               29 non-null     float64
 29  quoted_status_id_str           29 non-null     float64
 30  quoted_status                  28 non-null     object
dtypes: bool(4), datetime64[ns, UTC](1), float64(11), int64(4), object(11)
memory usage: 505.9+ KB
```

In [62]:

```
# rename the column id
cleaned_status = cleaned_status.rename(columns={'id':'tweet_id'})
```

In [63]:

```
cleaned_status.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   created_at                     2354 non-null   datetime64[ns, UTC]
 1   tweet_id                       2354 non-null   int64
 2   id_str                         2354 non-null   int64
 3   full_text                      2354 non-null   object
 4   truncated                      2354 non-null   bool
 5   display_text_range             2354 non-null   object
 6   entities                       2354 non-null   object
 7   extended_entities              2073 non-null   object
 8   source                         2354 non-null   object
 9   in_reply_to_status_id          78 non-null     float64
 10  in_reply_to_status_id_str      78 non-null     float64
 11  in_reply_to_user_id            78 non-null     float64
 12  in_reply_to_user_id_str        78 non-null     float64
 13  in_reply_to_screen_name        78 non-null     object
 14  user                           2354 non-null   object
 15  geo                            0 non-null      float64
 16  coordinates                    0 non-null      float64
 17  place                          1 non-null      object
 18  contributors                   0 non-null      float64
 19  is_quote_status                2354 non-null   bool
 20  retweet_count                  2354 non-null   int64
 21  favorite_count                 2354 non-null   int64
 22  favorited                      2354 non-null   bool
 23  retweeted                      2354 non-null   bool
 24  possibly_sensitive             2211 non-null   float64
 25  possibly_sensitive_appealable  2211 non-null   float64
 26  lang                           2354 non-null   object
 27  retweeted_status               179 non-null    object
 28  quoted_status_id               29 non-null     float64
 29  quoted_status_id_str           29 non-null     float64
 30  quoted_status                  28 non-null     object
dtypes: bool(4), datetime64[ns, UTC](1), float64(11), int64(4), object(11)
memory usage: 505.9+ KB
```

In [64]:

```
# filter/select needed columns
cleaned_status = cleaned_status.filter(['tweet_id','favorite_count','retweet_count', 'sourc

cleaned_status.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   tweet_id        2354 non-null   int64 
 1   favorite_count  2354 non-null   int64 
 2   retweet_count   2354 non-null   int64 
 3   source          2354 non-null   object
 4   user            2354 non-null   object
dtypes: int64(3), object(2)
memory usage: 92.1+ KB
```

Merging documents to form a working dataframe

In [65]:

```
tweet_df = pd.merge(cleaned_arch, cleaned_images, how='outer', on=['tweet_id'])
```

In [66]:

```
tweet_df = pd.merge(tweet_df, cleaned_status, how = 'outer', on=['tweet_id'])
```

In [67]:

```
tweet_df.sample(5)
```

Out[67]:

	tweet_id	timestamp	source_x	text	rating_
535	802247111496568832	2016-11-25 20:26:31 +0000	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: Everybody drop what you're doin...	
2091	668988183816871936	2015-11-24 03:03:06 +0000	<a href="http://twitter.com/download/iphone" r...	Honor to rate this dog. Lots of fur on him. Tw...	
2133	668268907921326080	2015-11-22 03:24:58 +0000	<a href="http://twitter.com/download/iphone" r...	Here we have an Azerbaijani Buttermilk named G...	
2273	731156023742988288	NaN	NaN	NaN	
19	888202515573088257	2017-07-21 01:02:36 +0000	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: This is Canela. She attempted s...	

In [68]:

```
tweet_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2356 entries, 0 to 2355
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   tweet_id            2356 non-null   int64  
 1   timestamp           2260 non-null   object 
 2   source_x            2260 non-null   object 
 3   text                2260 non-null   object 
 4   rating_numerator    2260 non-null   float64
 5   rating_denominator  2260 non-null   float64
 6   name                2260 non-null   object 
 7   column_class        2260 non-null   object 
 8   Breed_probability1  1239 non-null   object 
 9   p1_conf             1239 non-null   float64
 10  favorite_count      2354 non-null   float64
 11  retweet_count       2354 non-null   float64
 12  source_y            2354 non-null   object 
 13  user                2354 non-null   object 
dtypes: float64(5), int64(1), object(8)
memory usage: 276.1+ KB
```

In [69]:

```
# Saving df as csv
tweet_df.to_csv('twitter_archive_master.csv')
```

4 - Analyze and visualizing

In [70]:

```
# Most used twitter ressources
tweet_df['source_x'].value_counts()
```

Out[70]:

```
<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPho
ne</a>     2126
<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
91
<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
33
<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetD
eck</a>      10
Name: source_x, dtype: int64
```

In [71]:

```
tweet_df['rating_numerator'].value_counts().sort_index()
```

Out[71]:

```
1.0       5
2.0       9
3.0      19
4.0      15
5.0      35
6.0      32
7.0      53
8.0     102
9.0     155
10.0    457
11.0    453
12.0    544
13.0    331
14.0     49
15.0      1
Name: rating_numerator, dtype: int64
```

Analysis of rating of dogs

In [72]:

```
tweet_df['rating_numerator'][tweet_df['rating_numerator'] > 10].value_counts().sum()
```

Out[72]:

```
1378
```

In [73]:

```
# Analysis of retweet and favorite counts

print('%s\t%s' % ('Mean Retweet Count', round(tweet_df.retweet_count.mean())))
print('%s\t%s' % ('Mean Favorite Count', round(tweet_df.favorite_count.mean())))
```

```
Mean Retweet Count      3165
Mean Favorite Count     8081
```

In [74]:

```
# When the dog is rated greater than 10

print('%s\t%s' % ('Mean Retweet Count',
                  round(tweet_df.retweet_count[tweet_df.rating_numerator > 10].mean())))
print('%s\t%s' % ('Mean Favorite Count',
                  round(tweet_df.favorite_count[tweet_df.rating_numerator > 10].mean())))
```

```
Mean Retweet Count      4424
Mean Favorite Count     11435
```

In [75]:

```
# When the dog has a name

print('%s\t%s' % ('Mean Retweet Count',
                  round(tweet_df.retweet_count[tweet_df.name != 'None'].mean())))
print('%s\t%s' % ('Mean Favorite Count',
                  round(tweet_df.favorite_count[tweet_df.name != 'None'].mean())))
```

```
Mean Retweet Count      3012
Mean Favorite Count     8190
```

In [78]:

```
# Categorized on dog-class

print('Doggo')
print('%s\t%s' % ('Mean Retweet Count',
                  round(tweet_df.retweet_count[tweet_df.column_class  == 'doggo'].mean())))
print('%s\t%s' % ('Mean Favorite Count',
                  round(tweet_df.favorite_count[tweet_df.column_class  == 'doggo'].mean())))

print('Floofer')
print('%s\t%s' % ('Mean Retweet Count',
                  round(tweet_df.retweet_count[tweet_df.column_class  == 'floofer'].mean())
print('%s\t%s' % ('Mean Favorite Count',
                  round(tweet_df.favorite_count[tweet_df.column_class  == 'floofer'].mean()

print('Pupper')
print('%s\t%s' % ('Mean Retweet Count',
                  round(tweet_df.retweet_count[tweet_df.column_class  == 'pupper'].mean())))
print('%s\t%s' % ('Mean Favorite Count',
                  round(tweet_df.favorite_count[tweet_df.column_class  == 'pupper'].mean())

print('Puppo')
print('%s\t%s' % ('Mean Retweet Count',
                  round(tweet_df.retweet_count[tweet_df.column_class == 'puppo'].mean())))
print('%s\t%s' % ('Mean Favorite Count',
                  round(tweet_df.favorite_count[tweet_df.column_class  == 'puppo'].mean())))
```

```
Doggo
Mean Retweet Count      7800
Mean Favorite Count     16254
Floofer
Mean Retweet Count      4084
Mean Favorite Count     11675
Pupper
Mean Retweet Count      3023
Mean Favorite Count     6807
Puppo
Mean Retweet Count      6802
Mean Favorite Count     18799
```

In [79]:

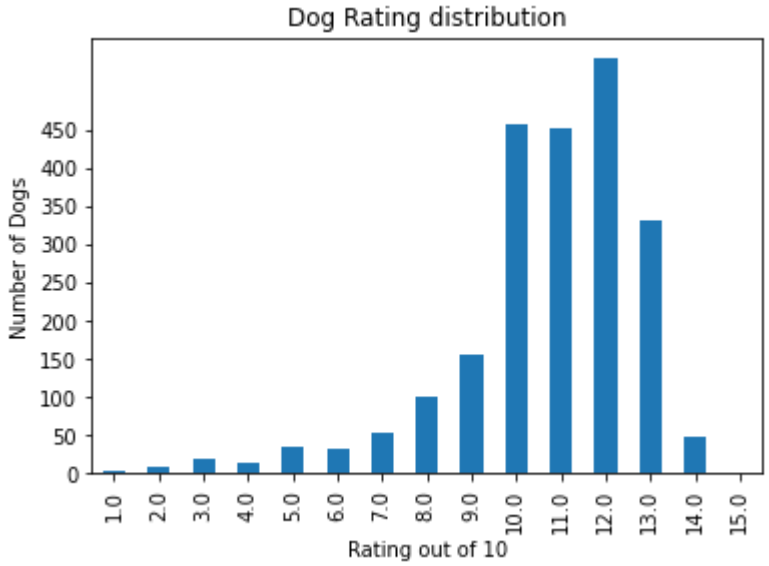
```
tweet_df.name.value_counts()
```

Out[79]:

```
None       658
a           54
Charlie     12
Oliver      11
Lucy        11
          ...
Sam          1
Rover        1
Banjo        1
Monkey       1
Willie       1
Name: name, Length: 954, dtype: int64
```

In [83]:

```
ax = tweet_df.rating_numerator.value_counts().sort_index().plot(kind='bar', title = 'Dog Ra
ax.set_xlabel("Rating out of 10")
ax.set_ylabel("Number of Dogs")
ax.set_yticks([0, 50, 100, 150, 200, 250, 300, 350, 400, 450])
plt.savefig('rating_dist')
```



In [86]:

```
tweet_df.name.value_counts()[1:7].plot(kind='barh', figsize=(11,5), title='Top 6 common dog
plt.savefig('dog_names')
```