Malak Alshedokhi

In [1]:

```
#importing required libraries:
import pandas as pd
import requests
import tweepy
import json
from timeit import default_timer as timer
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import datetime
```

Gathering the data:

1) Twitter Archive

Loading the csv file twitter archive enhanced provided by Udacity

In [2]:

```
twitter_archive = pd.read_csv('twitter-archive-enhanced.csv')
```

Testing the data frame

In [3]:

```
twitter_archive.head(10)
```

Out[3]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	tim

0	892420643555336193	NaN	NaN	1
1	892177421306343426	NaN	NaN	2 0
2	891815181378084864	NaN	NaN	2 0
3	891689557279858688	NaN	NaN	2 1
4	891327558926688256	NaN	NaN	2 1
5	891087950875897856	NaN	NaN	2 0
6	890971913173991426	NaN	NaN	2 1
7	890729181411237888	NaN	NaN	2 0
8	890609185150312448	NaN	NaN	2 1
9	890240255349198849	NaN	NaN	2 1

2) Image Predictions

In [4]:

```
# Using requests library to retrieve the TSV file from Udacity server
url = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv"
response = requests.get(url)
with open('image_predictions.tsv', 'wb') as file:
    file.write(response.content)
# Saving the tsv file into a data frame
image_predictions = pd.read_csv('image_predictions.tsv', sep='\t')
#Testing the data frame
image_predictions.head(10)
```

Out[4]:

	tweet_id	jp
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aM
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iD
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMy
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-IE
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4
5	666050758794694657	https://pbs.twimg.com/media/CT5Jof1WUAEuVxI
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJas
7	666055525042405380	https://pbs.twimg.com/media/CT5N9tpXIAAifs
8	666057090499244032	https://pbs.twimg.com/media/CT5PY90WoAAQGL
9	666058600524156928	https://pbs.twimg.com/media/CT5Qw94XAAA_2d

3) Twitter count - using twitter API

In [6]:

```
# Authorization to use twitter API
consumer_key = 'Qa0gXiqTgYxLSkHYlhaOxiKSD'
consumer_secret = 'P6mCqni30MqmC9OtQd1m3htZLH9SnEJsi7WNDSVh6wKrDrnTn7'
access_token = '379208944-FvCCVlgYn9jswLpdFrDa6tHp2gJ1EEWMSSxqiQXk'
access_secret = 'YhTHgvlIG86FIEouEpeLy7UI3tjOAbYvQiG6IkVhtC3B7'


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

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

Used the file twitter_api.py

In [7]:

```python
tweet_ids = twitter_archive.tweet_id.values
len(tweet_ids)

# 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, 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)
```

```
1: 892420643555336193
Success
2: 892177421306343426
Success
3: 891815181378084864
Success
4: 891689557279858688
Success
5: 891327558926688256
Success
6: 891087950875897856
Success
7: 890971913173991426
Success
8: 890729181411237888

---------------------------------------------------------------------------
```

```
-------------------------
KeyboardInterrupt
Traceback (most recent call last)
<ipython-input-7-8d6fb7828632> in <module>
     13         print(str(count) + ": " + str(twee
t_id))
     14         try:
---> 15             tweet = api.get_status(tweet_i
d, tweet_mode='extended')
     16             print("Success")
     17             json.dump(tweet, outfile)

/anaconda3/lib/python3.7/site-packages/tweepy/bind
er.py in _call(*args, **kwargs)
    248             return method
    249         else:
--> 250             return method.execute()
    251
    252     # Set pagination mode

/anaconda3/lib/python3.7/site-packages/tweepy/bind
er.py in execute(self)
    188
timeout=self.api.timeout,
    189
auth=auth,
--> 190
proxies=self.api.proxy)
    191                 except Exception as e:
    192                     six.reraise(TweepError
, TweepError('Failed to send request: %s' % e),
sys.exc_info()[2])

/anaconda3/lib/python3.7/site-packages/requests/se
ssions.py in request(self, method, url, params, da
ta, headers, cookies, files, auth, timeout, allow_
redirects, proxies, hooks, stream, verify, cert, j
son)
    531         }
    532         send_kwargs.update(settings)
--> 533         resp = self.send(prep, **send_kwar
gs)
    534
    535         return resp

/anaconda3/lib/python3.7/site-packages/requests/se
ssions.py in send(self, request, **kwargs)
    644
    645         # Send the request
--> 646         r = adapter.send(request, **kwargs
)
    647 
    648         # Total elapsed time of the reques
t (approximately)

/anaconda3/lib/python3.7/site-packages/requests/ad
apters.py in send(self, request, stream, timeout,
verify, cert, proxies)
    447                     decode_content=False,
    448                     retries=self.max_retri
es,
--> 449                     timeout=timeout
    450                 )
    451 

/anaconda3/lib/python3.7/site-packages/urllib3/con
nectionpool.py in urlopen(self, method, url, body,
headers, retries, redirect, assert_same_host, time
out, pool_timeout, release_conn, chunked, body_pos
, **response_kw)
    598 
timeout=timeout_obj,
    599 
body=body, headers=headers,
--> 600 
chunked=chunked)
    601 
    602             # If we're going to release th
e connection in ``finally:``, then

/anaconda3/lib/python3.7/site-packages/urllib3/con
nectionpool.py in _make_request(self, conn, method
, url, timeout, chunked, **httplib_request_kw)
    341         # Trigger any extra validation we
need to do.
    342         try:
--> 343             self._validate_conn(conn)
    344         except (SocketTimeout,
BaseSSLError) as e:
    345             # Py2 raises this as a BaseSSL
Error, Py3 raises it as socket timeout.

/anaconda3/lib/python3.7/site-packages/urllib3/con
nectionpool.py in _validate_conn(self, conn)
    837         # Force connect early to allow us
to validate the connection.
```

```
    838                 if not getattr(conn, 'sock', None)
:  # AppEngine might not have  `.sock`
--> 839                     conn.connect()
    840 
    841                 if not conn.is_verified:

/anaconda3/lib/python3.7/site-packages/urllib3/con
nection.py in connect(self)
    342                     ca_cert_dir=self.ca_cert_dir,
    343                     server_hostname=server_hostnam
e,
--> 344                     ssl_context=context)
    345 
    346                 if self.assert_fingerprint:

/anaconda3/lib/python3.7/site-packages/urllib3/uti
l/ssl_.py in ssl_wrap_socket(sock, keyfile, certfi
le, cert_reqs, ca_certs, server_hostname, ssl_vers
ion, ciphers, ssl_context, ca_cert_dir)
    319         if ca_certs or ca_cert_dir:
    320             try:
--> 321                 context.load_verify_locations(
ca_certs, ca_cert_dir)
    322             except IOError as e:  # Platform-s
pecific: Python 2.7
    323                 raise SSLError(e)

/anaconda3/lib/python3.7/site-packages/urllib3/con
trib/pyopenssl.py in load_verify_locations(self, c
afile, capath, cadata)
    426             if capath is not None:
    427                 capath = capath.encode('utf-8'
)
--> 428             self._ctx.load_verify_locations(ca
file, capath)
    429             if cadata is not None:
    430                 self._ctx.load_verify_location
s(BytesIO(cadata))

/anaconda3/lib/python3.7/site-packages/OpenSSL/SSL
.py in load_verify_locations(self, cafile, capath)
    776 
    777             load_result = _lib.SSL_CTX_load_ve
rify_locations(
--> 778                 self._context, cafile, capath
    779             )
    780             if not load_result:
```

```
KeyboardInterrupt:
```

Reading JSON content as pandas dataframe

In [8]:

```
tweet_status = pd.read_json('tweet-json.txt', lines = True)
```

In [9]:

```
tweet_status.columns
```

Out[9]:

```
Index(['contributors', 'coordinates', 'created_at'
, 'display_text_range',
       'entities', 'extended_entities', 'favorite_
count', 'favorited',
       'full_text', 'geo', 'id', 'id_str', 'in_rep
ly_to_screen_name',
       'in_reply_to_status_id', 'in_reply_to_statu
s_id_str',
       'in_reply_to_user_id', 'in_reply_to_user_id
_str', 'is_quote_status',
       'lang', 'place', 'possibly_sensitive', 'pos
sibly_sensitive_appealable',
       'quoted_status', 'quoted_status_id', 'quote
d_status_id_str',
       'retweet_count', 'retweeted', 'retweeted_st
atus', 'source', 'truncated',
       'user'],
      dtype='object')
```

In [15]:

```
tweet_status.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
contributors                     0 non-null float6
4
coordinates                      0 non-null float6
4
created_at                       2354 non-null dat
etime64[ns]
display_text_range               2354 non-null obj
```

```
ect
entities                         2354 non-null obj
ect
extended_entities                2073 non-null obj
ect
favorite_count                   2354 non-null int
64
favorited                        2354 non-null boo
l
full_text                        2354 non-null obj
ect
geo                              0 non-null float6
4
id                               2354 non-null int
64
id_str                           2354 non-null int
64
in_reply_to_screen_name          78 non-null objec
t
in_reply_to_status_id            78 non-null float
64
in_reply_to_status_id_str        78 non-null float
64
in_reply_to_user_id              78 non-null float
64
in_reply_to_user_id_str          78 non-null float
64
is_quote_status                  2354 non-null boo
l
lang                             2354 non-null obj
ect
place                            1 non-null object
possibly_sensitive               2211 non-null flo
at64
possibly_sensitive_appealable    2211 non-null flo
at64
quoted_status                    28 non-null objec
t
quoted_status_id                 29 non-null float
64
quoted_status_id_str             29 non-null float
64
retweet_count                    2354 non-null int
64
retweeted                        2354 non-null boo
l
retweeted_status                 179 non-null obje
ct
```

```
source                             2354 non-null obj
ect
truncated                          2354 non-null boo
l
user                               2354 non-null obj
ect
dtypes: bool(4), datetime64[ns](1), float64(11), i
nt64(4), object(11)
memory usage: 505.8+ KB
```

Assessing :

1) Visually:

In [10]:

```
twitter_archive.head(10)
```

Out[10]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	tim
0	892420643555336193	NaN	NaN	2 1
1	892177421306343426	NaN	NaN	2 0
2	891815181378084864	NaN	NaN	2 0
3	891689557279858688	NaN	NaN	2 1
4	891327558926688256	NaN	NaN	2 1

5	891087950875897856	NaN	NaN	2 0
6	890971913173991426	NaN	NaN	2 1
7	890729181411237888	NaN	NaN	2 0
8	890609185150312448	NaN	NaN	2 1
9	890240255349198849	NaN	NaN	2 1

In [18]:

```
twitter_archive.tail(10)
```

Out[18]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id
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

In [11]:

```
image_predictions.head(10)
```

Out[11]:

	tweet_id	jp
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aM
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iD
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMy
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lE
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4
5	666050758794694657	https://pbs.twimg.com/media/CT5Jof1WUAEuVx
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJas
7	666055525042405380	https://pbs.twimg.com/media/CT5N9tpXIAAifs
8	666057090499244032	https://pbs.twimg.com/media/CT5PY90WoAAQGL
9	666058600524156928	https://pbs.twimg.com/media/CT5Qw94XAAA_2d

In [12]:

```
image_predictions.tail(10)
```

Out[12]:

	tweet_id	
2065	890240255349198849	https://pbs.twimg.com/media/DFrEyVuW0AA
2066	890609185150312448	https://pbs.twimg.com/media/DFwUU__XcAE
2067	890729181411237888	https://pbs.twimg.com/media/DFyBahAVwAA
2068	890971913173991426	https://pbs.twimg.com/media/DF1eOmZXUAA
2069	891087950875897856	https://pbs.twimg.com/media/DF3HwyEWsAA
2070	891327558926688256	https://pbs.twimg.com/media/DF6hr6BUMAA
2071	891689557279858688	https://pbs.twimg.com/media/DF_q7IAWsAE
2072	891815181378084864	https://pbs.twimg.com/media/DGBdLU1WsAA
2073	892177421306343426	https://pbs.twimg.com/media/DGGmoV4XsAA
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAA

In [13]:

```
tweet_status.head(10)
```

Out[13]:

	contributors	coordinates	created_at	display_text_range	
0	NaN	NaN	2017-08-01 16:23:56	[0, 85]	{'hash 'sym 'user_me
1	NaN	NaN	2017-08-01 00:17:27	[0, 138]	{'hash 'sym 'user_me
2	NaN	NaN	2017-07-31 00:18:03	[0, 121]	{'hash 'sym 'user_me

3	NaN	NaN	2017-07-30 15:58:51	[0, 79]	{'hash 'sym 'user_me
4	NaN	NaN	2017-07-29 16:00:24	[0, 138]	{'ha 'Bar 'ind
5	NaN	NaN	2017-07-29 00:08:17	[0, 138]	{'ha 'Bar 'ind
6	NaN	NaN	2017-07-28 16:27:12	[0, 140]	{'hash 'sym 'user_me
7	NaN	NaN	2017-07-28 00:22:40	[0, 118]	{'hash 'sym 'user_me
8	NaN	NaN	2017-07-27 16:25:51	[0, 122]	{'ha 'Bar 'ind
9	NaN	NaN	2017-07-26 15:59:51	[0, 133]	{'hash 'sym 'user_me

10 rows × 31 columns

In [22]:

```
tweet_status.tail(10)
```

Out[22]:

	contributors	coordinates	created_at	display_text_range	
2344	NaN	NaN	2015-11-16	[0, 135]	{'h ';

			01:01:59		'user_
2345	NaN	NaN	2015-11- 16 00:55:59	[0, 124]	{'h 's 'user_
2346	NaN	NaN	2015-11- 16 00:49:46	[0, 140]	{'h 's 'user_
2347	NaN	NaN	2015-11- 16 00:35:11	[0, 138]	{'h 's 'user_
2348	NaN	NaN	2015-11- 16 00:30:50	[0, 140]	{'h 's 'user_
2349	NaN	NaN	2015-11- 16 00:24:50	[0, 120]	{'h 's 'user_
2350	NaN	NaN	2015-11- 16 00:04:52	[0, 137]	{'h 's 'user_
2351	NaN	NaN	2015-11- 15 23:21:54	[0, 130]	{'h 's 'user_
2352	NaN	NaN	2015-11- 15 23:05:30	[0, 139]	{'h 's 'user_

2353	NaN	NaN	2015-11-15 22:32:08	[0, 131]	{'h 's 'user_

10 rows × 31 columns

2) Programmatically:

In [14]:

```
twitter_archive.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 [15]:

```
twitter_archive.describe()
```

Out[15]:

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

In [16]:

```
image_predictions.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 [17]:

```
image_predictions.describe()
```

Out[17]:

	tweet_id	img_num	p1_conf	p2_conf	p
count	2.075000e+03	2075.000000	2075.000000	2.075000e+03	2.0750
mean	7.384514e+17	1.203855	0.594548	1.345886e-01	6.032
std	6.785203e+16	0.561875	0.271174	1.006657e-01	5.090
min	6.660209e+17	1.000000	0.044333	1.011300e-08	1.740
25%	6.764835e+17	1.000000	0.364412	5.388625e-02	1.622
50%	7.119988e+17	1.000000	0.588230	1.181810e-01	4.944
75%	7.932034e+17	1.000000	0.843855	1.955655e-01	9.180
max	8.924206e+17	4.000000	1.000000	4.880140e-01	2.734

In [23]:

```
tweet_status.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
contributors                     0 non-null float6
4
coordinates                      0 non-null float6
4
created_at                       2354 non-null dat
etime64[ns]
display_text_range               2354 non-null obj
ect
entities                         2354 non-null obj
ect
extended_entities                2073 non-null obj
ect
favorite_count                   2354 non-null int
64
favorited                        2354 non-null boo
l
full_text                        2354 non-null obj
ect
geo                              0 non-null float6
```

```
4
id                             2354 non-null int
64
id_str                         2354 non-null int
64
in_reply_to_screen_name          78 non-null objec
t
in_reply_to_status_id            78 non-null float
64
in_reply_to_status_id_str        78 non-null float
64
in_reply_to_user_id              78 non-null float
64
in_reply_to_user_id_str          78 non-null float
64
is_quote_status                2354 non-null boo
l
lang                           2354 non-null obj
ect
place                          1 non-null object
possibly_sensitive             2211 non-null flo
at64
possibly_sensitive_appealable  2211 non-null flo
at64
quoted_status                    28 non-null objec
t
quoted_status_id                 29 non-null float
64
quoted_status_id_str             29 non-null float
64
retweet_count                  2354 non-null int
64
retweeted                      2354 non-null boo
l
retweeted_status               179 non-null obje
ct
source                         2354 non-null obj
ect
truncated                      2354 non-null boo
l
user                           2354 non-null obj
ect
dtypes: bool(4), datetime64[ns](1), float64(11), i
nt64(4), object(11)
memory usage: 505.8+ KB
```

In [18]:

```
tweet_status.describe()
```

Out[18]:

	contributors	coordinates	favorite_count	geo	id
count	0.0	0.0	2354.000000	0.0	2.354000e+03
mean	NaN	NaN	8080.968564	NaN	7.426978e+17
std	NaN	NaN	11814.771334	NaN	6.852812e+16
min	NaN	NaN	0.000000	NaN	6.660209e+17
25%	NaN	NaN	1415.000000	NaN	6.783975e+17
50%	NaN	NaN	3603.500000	NaN	7.194596e+17
75%	NaN	NaN	10122.250000	NaN	7.993058e+17
max	NaN	NaN	132810.000000	NaN	8.924206e+17

Observations:

Quality:

twitter archive dataframe:

1) The colums (in_reply_to_status_id , in_reply_to_user_id, retweeted_status_id , retweeted_status_user_id, retweeted_status_timestamp) have a lot of missing values/

2) All IDs data type should be String not integer.

3) In column name, some hase None value

4) Time stamp column hase (+0000) in its values, that made it lengthy and messy

5) In column name, some names are not accurate, like "a" , "an".

image prediction dataframe:

6) column names are not informative and descriptive.

7) tweet_id should be string data type

tweet_status dataframe:

8) Id columns are integers not strings

Tidiness:

1) The nature of the data source provided us with 3 data frames, while we might merge them to get one clean table.

2) The columns (doggo , floofer , pupper , puppo) could be as values of one column named type.

Cleaning :

Making copies of the data frames

In [19]:

```
archive_clean = twitter_archive.copy()
```

In [20]:

```
tweet_status_clean1 = tweet_status.copy()
```

In [21]:

```
image_predictions_clean = image_predictions.copy()
```

Testing

In [22]:

```
archive_clean
```

Out[22]:

	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

2356 rows × 17 columns

In [23]:

```
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null 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
```

Testing

In [26]:

```
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null 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
```

Many columns have a lot of missing values, I will delete them

In [36]:

```
archive_clean = archive_clean.drop(['in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp'], axis=1)
```

Testing

In [37]:

```
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2278 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2278 non-null int64
timestamp             2278 non-null object
source                2278 non-null object
text                  2278 non-null object
expanded_urls         2274 non-null object
rating_numerator      2278 non-null int64
rating_denominator    2278 non-null int64
name                  2278 non-null object
doggo                 2278 non-null object
floofer               2278 non-null object
pupper                2278 non-null object
puppo                 2278 non-null object
dtypes: int64(3), object(9)
memory usage: 231.4+ KB
```

The word None is better to represent a vlaue than Nan. Will do replacing

In [30]:

```
archive_clean =  archive_clean.replace( np.nan ,'None')
```

Testing

In [31]:

```
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         2356 non-null object
in_reply_to_user_id           2356 non-null object
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           2356 non-null object
retweeted_status_user_id      2356 non-null object
retweeted_status_timestamp    2356 non-null object
expanded_urls                 2356 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: int64(3), object(14)
memory usage: 313.0+ KB
```

In [32]:

```
archive_clean.head()
```

Out[32]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	tim
0	892420643555336193	None	None	2 1
1	892177421306343426	None	None	2 0
2	891815181378084864	None	None	2 0
3	891689557279858688	None	None	2 1
4	891327558926688256	None	None	2 1

Add a new column called Dog_Type to define whether it is "doggo,floofer , pupper ,puppo".

if it is not defined it will be Nan

Then will drop the 4 colums.

Adding the new column

In [33]:

```
archive_clean.loc[archive_clean['doggo'] == 'doggo', 'Dog_Type'] = 'doggo'
archive_clean.loc[archive_clean['floofer'] == 'floofer', 'Dog_Type'] = 'floofer'
archive_clean.loc[archive_clean['pupper'] == 'pupper', 'Dog_Type'] = 'pupper'
archive_clean.loc[archive_clean['puppo'] == 'puppo', 'Dog_Type'] = 'puppo'
```

Dropping the 4 columns

In [34]:

```
archive_clean = archive_clean.drop(['doggo', 'floofer', 'pupper', 'puppo'], axis = 1)
```

Testing

In [35]:

```
archive_clean.head(10)
```

Out[35]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	tim
0	892420643555336193	None	None	2 1
1	892177421306343426	None	None	2 (
2	891815181378084864	None	None	2 (

3	891689557279858688	None	None	2 1
4	891327558926688256	None	None	2 1
5	891087950875897856	None	None	2 (
6	890971913173991426	None	None	2 1
7	890729181411237888	None	None	2 (
8	890609185150312448	None	None	2 1
9	890240255349198849	None	None	2 1

Renaming columns in image predictions data frame to have more informative headings

In [36]:

```
image_predictions_clean = image_predictions_clean.rename(columns={'p1':'Breed_probability_1', 'p2':'Breed_probability_2', 'p3':'Breed_probability_3', 'p1_conf': 'probability_1_conf','p2_conf': 'probability_2_conf', 'p3_conf': 'probability_3_conf', 'p1_dog': 'probability_1_dog', 'p2_dog': 'probability_2_dog', 'p3_dog': 'probability_3_dog'})
```

In [37]:

```
image_predictions_clean.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
Breed_probability_1     2075 non-null object
probability_1_conf      2075 non-null float64
probability_1_dog       2075 non-null bool
Breed_probability_2     2075 non-null object
probability_2_conf      2075 non-null float64
probability_2_dog       2075 non-null bool
Breed_probability_3     2075 non-null object
probability_3_conf      2075 non-null float64
probability_3_dog       2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

ID column is not String data type, so it will be changed.

In [38]:

```
image_predictions_clean.tweet_id = image_predictions_clean.twe
et_id.astype(str)
archive_clean.tweet_id = archive_clean.tweet_id.astype(str)
```

tweet_status_clean.tweet_id = tweet_status_clean.tweet_id.astype(str)

In [39]:

```
tweet_status_clean1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
contributors                      0 non-null float6
4
coordinates                       0 non-null float6
```

```
4
created_at                       2354 non-null dat
etime64[ns]
display_text_range               2354 non-null obj
ect
entities                         2354 non-null obj
ect
extended_entities                2073 non-null obj
ect
favorite_count                   2354 non-null int
64
favorited                        2354 non-null boo
l
full_text                        2354 non-null obj
ect
geo                              0 non-null float6
4
id                               2354 non-null int
64
id_str                           2354 non-null int
64
in_reply_to_screen_name          78 non-null objec
t
in_reply_to_status_id            78 non-null float
64
in_reply_to_status_id_str        78 non-null float
64
in_reply_to_user_id              78 non-null float
64
in_reply_to_user_id_str          78 non-null float
64
is_quote_status                  2354 non-null boo
l
lang                             2354 non-null obj
ect
place                            1 non-null object
possibly_sensitive               2211 non-null flo
at64
possibly_sensitive_appealable    2211 non-null flo
at64
quoted_status                    28 non-null objec
t
quoted_status_id                 29 non-null float
64
quoted_status_id_str             29 non-null float
64
retweet_count                    2354 non-null int
64
```

```
retweeted                         2354 non-null boo
l
retweeted_status                  179 non-null obje
ct
source                            2354 non-null obj
ect
truncated                         2354 non-null boo
l
user                              2354 non-null obj
ect
dtypes: bool(4), datetime64[ns](1), float64(11), i
nt64(4), object(11)
memory usage: 505.8+ KB
```

Testing

In [40]:

```
image_predictions_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id                2075 non-null object
jpg_url                 2075 non-null object
img_num                 2075 non-null int64
Breed_probability_1     2075 non-null object
probability_1_conf      2075 non-null float64
probability_1_dog       2075 non-null bool
Breed_probability_2     2075 non-null object
probability_2_conf      2075 non-null float64
probability_2_dog       2075 non-null bool
Breed_probability_3     2075 non-null object
probability_3_conf      2075 non-null float64
probability_3_dog       2075 non-null bool
dtypes: bool(3), float64(3), int64(1), object(5)
memory usage: 152.1+ KB
```

In [41]:

```
tweet_status_clean1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
contributors                      0 non-null float6
```

```
4
coordinates                         0 non-null float6
4
created_at                          2354 non-null dat
etime64[ns]
display_text_range                  2354 non-null obj
ect
entities                            2354 non-null obj
ect
extended_entities                   2073 non-null obj
ect
favorite_count                      2354 non-null int
64
favorited                           2354 non-null boo
l
full_text                           2354 non-null obj
ect
geo                                 0 non-null float6
4
id                                  2354 non-null int
64
id_str                              2354 non-null int
64
in_reply_to_screen_name             78 non-null objec
t
in_reply_to_status_id               78 non-null float
64
in_reply_to_status_id_str           78 non-null float
64
in_reply_to_user_id                 78 non-null float
64
in_reply_to_user_id_str             78 non-null float
64
is_quote_status                     2354 non-null boo
l
lang                                2354 non-null obj
ect
place                               1 non-null object
possibly_sensitive                  2211 non-null flo
at64
possibly_sensitive_appealable       2211 non-null flo
at64
quoted_status                       28 non-null objec
t
quoted_status_id                    29 non-null float
64
quoted_status_id_str                29 non-null float
64
```

```
retweet_count                    2354 non-null int
64
retweeted                        2354 non-null boo
l
retweeted_status                 179 non-null obje
ct
source                           2354 non-null obj
ect
truncated                        2354 non-null boo
l
user                             2354 non-null obj
ect
dtypes: bool(4), datetime64[ns](1), float64(11), i
nt64(4), object(11)
memory usage: 505.8+ KB
```

Rename the column id to be tweet_id then change the data type to be string

In [42]:

```
tweet_status_clean1 = tweet_status_clean1.rename(columns={'id'
:'tweet_id'})
tweet_status_clean1.info()
tweet_status_clean1.tweet_id = tweet_status_clean1.tweet_id.as
type(str)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
contributors                     0 non-null float6
4
coordinates                      0 non-null float6
4
created_at                       2354 non-null dat
etime64[ns]
display_text_range               2354 non-null obj
ect
entities                         2354 non-null obj
ect
extended_entities                2073 non-null obj
ect
favorite_count                   2354 non-null int
64
favorited                        2354 non-null boo
l
```

```
full_text                        2354 non-null obj
ect
geo                              0 non-null float6
4
tweet_id                         2354 non-null int
64
id_str                           2354 non-null int
64
in_reply_to_screen_name          78 non-null objec
t
in_reply_to_status_id            78 non-null float
64
in_reply_to_status_id_str        78 non-null float
64
in_reply_to_user_id              78 non-null float
64
in_reply_to_user_id_str          78 non-null float
64
is_quote_status                  2354 non-null boo
l
lang                             2354 non-null obj
ect
place                            1 non-null object
possibly_sensitive               2211 non-null flo
at64
possibly_sensitive_appealable    2211 non-null flo
at64
quoted_status                    28 non-null objec
t
quoted_status_id                 29 non-null float
64
quoted_status_id_str             29 non-null float
64
retweet_count                    2354 non-null int
64
retweeted                        2354 non-null boo
l
retweeted_status                 179 non-null obje
ct
source                           2354 non-null obj
ect
truncated                        2354 non-null boo
l
user                             2354 non-null obj
ect
dtypes: bool(4), datetime64[ns](1), float64(11), i
nt64(4), object(11)
memory usage: 505.8+ KB
```

Filtering to show only the needed columns

In [44]:

```
tweet_status_clean1 = tweet_status_clean1.filter(['tweet_id','favorite_count','retweet_count', 'source', 'user'] )
tweet_status_clean1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 5 columns):
tweet_id             2354 non-null object
favorite_count       2354 non-null int64
retweet_count        2354 non-null int64
source               2354 non-null object
user                 2354 non-null object
dtypes: int64(2), object(3)
memory usage: 92.0+ KB
```

In [45]:

```
tweeter_df = pd.merge(archive_clean, image_predictions_clean, how='outer', on=['tweet_id'])
```

In [46]:

```
tweeter_df = pd.merge(tweeter_df, tweet_status_clean1, how = 'outer', on=['tweet_id'])
```

In [47]:

```
tweeter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2356 entries, 0 to 2355
Data columns (total 29 columns):
tweet_id                      2356 non-null object
in_reply_to_status_id         2356 non-null object
in_reply_to_user_id           2356 non-null object
timestamp                     2356 non-null object
source_x                      2356 non-null object
text                          2356 non-null object
retweeted_status_id           2356 non-null object
retweeted_status_user_id      2356 non-null object
retweeted_status_timestamp    2356 non-null object
expanded_urls                 2356 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
Dog_Type                      380 non-null object
jpg_url                       2075 non-null object
img_num                       2075 non-null float6
4
Breed_probability_1           2075 non-null object
probability_1_conf            2075 non-null float6
4
probability_1_dog             2075 non-null object
Breed_probability_2           2075 non-null object
probability_2_conf            2075 non-null float6
4
probability_2_dog             2075 non-null object
Breed_probability_3           2075 non-null object
probability_3_conf            2075 non-null float6
4
probability_3_dog             2075 non-null object
favorite_count                2354 non-null float6
4
retweet_count                 2354 non-null float6
4
source_y                      2354 non-null object
user                          2354 non-null object
dtypes: float64(6), int64(2), object(21)
memory usage: 552.2+ KB
```

In [49]:

```
tweeter_df.sample(5)
```

Out[49]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id
161	860563773140209665	None	None
203	853299958564483072	None	None
918	756651752796094464	None	None
1809	676864501615042560	None	None
504	813051746834595840	None	None

5 rows × 29 columns

In column name, some names are not accurate, like "a" , "an" and "the"

In [72]:

```
tweeter_df.columns
```

Out[72]:

```
Index(['tweet_id', 'in_reply_to_status_id', 'in_re
ply_to_user_id', 'timestamp',
       'source_x', 'text', 'retweeted_status_id',
'retweeted_status_user_id',
       'retweeted_status_timestamp', 'expanded_url
s', 'rating_numerator',
       'rating_denominator', 'name', 'Dog_Type', '
jpg_url', 'img_num',
       'Breed_probability_1', 'probability_1_conf'
, 'probability_1_dog',
       'Breed_probability_2', 'probability_2_conf'
, 'probability_2_dog',
       'Breed_probability_3', 'probability_3_conf'
, 'probability_3_dog',
       'favorite_count', 'retweet_count', 'source_
y', 'user'],
      dtype='object')
```

In [73]:

```
xx= tweeter_df[tweeter_df['name'] == 'a'].index
```

In [74]:

```
tweeter_df.drop(xx,axis = 0, inplace=True)
```

In [75]:

```
tweeter_df[tweeter_df['name'] == 'a'].index
```

Out[75]:

```
Int64Index([], dtype='int64')
```

In [78]:

```
anx = tweeter_df[tweeter_df['name'] == 'an'].index
```

In [80]:

```
tweeter_df.drop(anx,axis = 0, inplace=True)
```

In [84]:

```
tweeter_df[tweeter_df['name'] == 'an'].index
```

Out[84]:

```
Int64Index([], dtype='int64')
```

In [81]:

```
thex = tweeter_df[tweeter_df['name'] == 'the'].index
```

In [82]:

```
tweeter_df.drop(thex,axis = 0, inplace=True)
```

In [83]:

```
tweeter_df[tweeter_df['name'] == 'the'].index
```

Out[83]:

```
Int64Index([], dtype='int64')
```

splitting the timestamp column

In [56]:

```
twitter_archive['date'], twitter_archive['Time'] = twitter_arc
hive['timestamp'].str.split(expand=True).loc[:,0:1]
```

In [57]:

```
twitter_archive['Time'].head()
```

Out[57]:

```
0    1
1    1
2    1
3    1
4    1
Name: Time, dtype: int64
```

In [58]:

```
twitter_archive['timestamp'].str.split(expand=True).loc[:,0:1]
```

Out[58]:

	0	1
0	2017-08-01	16:23:56
1	2017-08-01	00:17:27
2	2017-07-31	00:18:03
3	2017-07-30	15:58:51
4	2017-07-29	16:00:24
5	2017-07-29	00:08:17
6	2017-07-28	16:27:12
7	2017-07-28	00:22:40
8	2017-07-27	16:25:51
9	2017-07-26	15:59:51
10	2017-07-26	00:31:25
11	2017-07-25	16:11:53
12	2017-07-25	01:55:32
13	2017-07-25	00:10:02
14	2017-07-24	17:02:04
15	2017-07-24	00:19:32
16	2017-07-23	00:22:39
17	2017-07-22	16:56:37
18	2017-07-22	00:23:06
19	2017-07-21	01:02:36
20	2017-07-20	16:49:33
21	2017-07-19	16:06:48
22	2017-07-19	03:39:09
23	2017-07-19	00:47:34
24	2017-07-18	16:08:03
25	2017-07-18	00:07:08
26	2017-07-17	16:17:36
27	2017-07-16	23:58:41

28	2017-07-16	20:14:00
29	2017-07-15	23:25:31
...	...	...
2318	2015-11-17	03:16:00
2319	2015-11-17	02:46:43
2320	2015-11-17	02:06:42
2321	2015-11-17	02:00:15
2322	2015-11-17	01:40:41
2323	2015-11-17	01:30:57
2324	2015-11-17	01:02:40
2325	2015-11-17	00:53:15
2326	2015-11-17	00:24:19
2328	2015-11-16	23:23:41
2329	2015-11-16	21:54:18
2330	2015-11-16	21:10:36
2331	2015-11-16	20:32:58
2332	2015-11-16	20:01:42
2333	2015-11-16	19:31:45
2335	2015-11-16	16:11:11
2336	2015-11-16	15:14:19
2337	2015-11-16	14:57:41
2338	2015-11-16	04:02:55
2339	2015-11-16	03:55:04
2340	2015-11-16	03:44:34
2341	2015-11-16	03:22:39
2342	2015-11-16	02:38:37
2343	2015-11-16	01:59:36
2344	2015-11-16	01:52:02
2345	2015-11-16	01:22:45
2346	2015-11-16	01:01:59
2349	2015-11-16	00:25:11

2349	2015-11-16	00:35:11
2351	2015-11-16	00:24:50
2355	2015-11-15	22:32:08

2301 rows × 2 columns

In [85]:

```
tweeter_df
```

Out[85]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id
0	892420643555336193	None	None
1	892177421306343426	None	None
2	891815181378084864	None	None
3	891689557279858688	None	None
4	891327558926688256	None	None
5	891087950875897856	None	None
6	890971913173991426	None	None

7	890729181411237888	None	None
8	890609185150312448	None	None
9	890240255349198849	None	None
10	890006608113172480	None	None
11	889880896479866881	None	None
12	889665388333682689	None	None
13	889638837579907072	None	None
14	889531135344209921	None	None
15	889278841981685760	None	None
16	888917238123831296	None	None

17	888804989199671297	None	None
18	888554962724278272	None	None
19	888202515573088257	None	None
20	888078434458587136	None	None
21	887705289381826560	None	None
22	887517139158093824	None	None
23	887473957103951883	None	None
24	887343217045368832	None	None
25	887101392804085760	None	None
26	886983233522544640	None	None

27	886736880519319552	None	None
28	886680336477933568	None	None
29	886366144734445568	None	None
...	...	...	...
2312	666776908487630848	None	None
2313	666739327293083650	None	None
2315	666691418707132416	None	None
2316	666649482315059201	None	None
2317	666644823164719104	None	None
2318	666454714377183233	None	None

2319	666447344410484738	None	None
2320	666437273139982337	None	None
2321	666435652385423360	None	None
2322	666430724426358785	None	None
2323	666428276349472768	None	None
2324	666421158376562688	None	None
2325	666418789513326592	None	None
2326	666411507551481857	None	None
2328	666396247373291520	None	None
2329	666373753744588802	None	None

2330	666362758909284353	None	None
2331	666353288456101888	None	None
2332	666345417576210432	None	None
2336	666273097616637952	None	None
2337	666268910803644416	None	None
2338	666104133288665088	None	None
2339	666102155909144576	None	None
2340	666099513787052032	None	None
2341	666094000022159362	None	None
2342	666082916733198337	None	None

2343	666073100786774016	None	None
2344	666071193221509120	None	None
2351	666049248165822465	None	None
2355	666020888022790149	None	None

I wanted to clean out the column source to list out the device that has been used for tweeting. That might provide insights.

I have used the code that was used in the below reference, and fixed to work on my code.

Reference :
https://static1.squarespace.com/static/55bfa8e4e4b007976149574e/t/5b870d81
(https://static1.squarespace.com/static/55bfa8e4e4b007976149574e/t/5b870d8

In [141]:

```
# Text replacements
source_txt = {'<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>': 'Twitter for iPhone',
                '<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>': 'Vine - Make a Scene',
                '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>': 'Twitter Web Client',
                '<a href="https://about.twitter.com/products/tweetdeck"rel="nofollow">TweetDeck</a>': 'TweetDeck'}
# Apply function
def text_source(tweeter_df):
    if tweeter_df['source_x'] in source_txt.keys():
       abbrev = source_txt[tweeter_df['source_x']]
       return abbrev
    else:
       return tweeter_df['source_x']

tweeter_df['source_x'] = tweeter_df.apply(text_source, axis=1)
```

In [142]:

```
tweeter_df.source_x.value_counts()
```

Out[142]:

```
Twitter for iPhone
2155
Vine - Make a Scene
90
Twitter Web Client
30
<a href="https://about.twitter.com/products/tweetd
eck" rel="nofollow">TweetDeck</a>      11
Name: source_x, dtype: int64
```

In [130]:

```
tweeter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2286 entries, 0 to 2355
Data columns (total 29 columns):
tweet_id                      2286 non-null object
in_reply_to_status_id         2286 non-null object
in_reply_to_user_id           2286 non-null object
timestamp                     2286 non-null object
source_x                      2286 non-null object
text                          2286 non-null object
retweeted_status_id           2286 non-null object
retweeted_status_user_id      2286 non-null object
retweeted_status_timestamp    2286 non-null object
expanded_urls                 2286 non-null object
rating_numerator              2286 non-null int64
rating_denominator            2286 non-null int64
name                          2286 non-null object
Dog_Type                      374 non-null object
jpg_url                       2006 non-null object
img_num                       2006 non-null float6
4
Breed_probability_1           2006 non-null object
probability_1_conf            2006 non-null float6
4
probability_1_dog             2006 non-null object
Breed_probability_2           2006 non-null object
probability_2_conf            2006 non-null float6
4
probability_2_dog             2006 non-null object
Breed_probability_3           2006 non-null object
probability_3_conf            2006 non-null float6
4
probability_3_dog             2006 non-null object
favorite_count                2284 non-null float6
4
retweet_count                 2284 non-null float6
4
source_y                      2284 non-null object
user                          2284 non-null object
dtypes: float64(6), int64(2), object(21)
memory usage: 535.8+ KB
```

Dropping uneeded columns

In [132]:

```
tweeter_df = tweeter_df.drop([ 'retweeted_status_user_id', 'so
urce_y'], axis = 1)
```

In [133]:

```
tweeter_df.info()
```

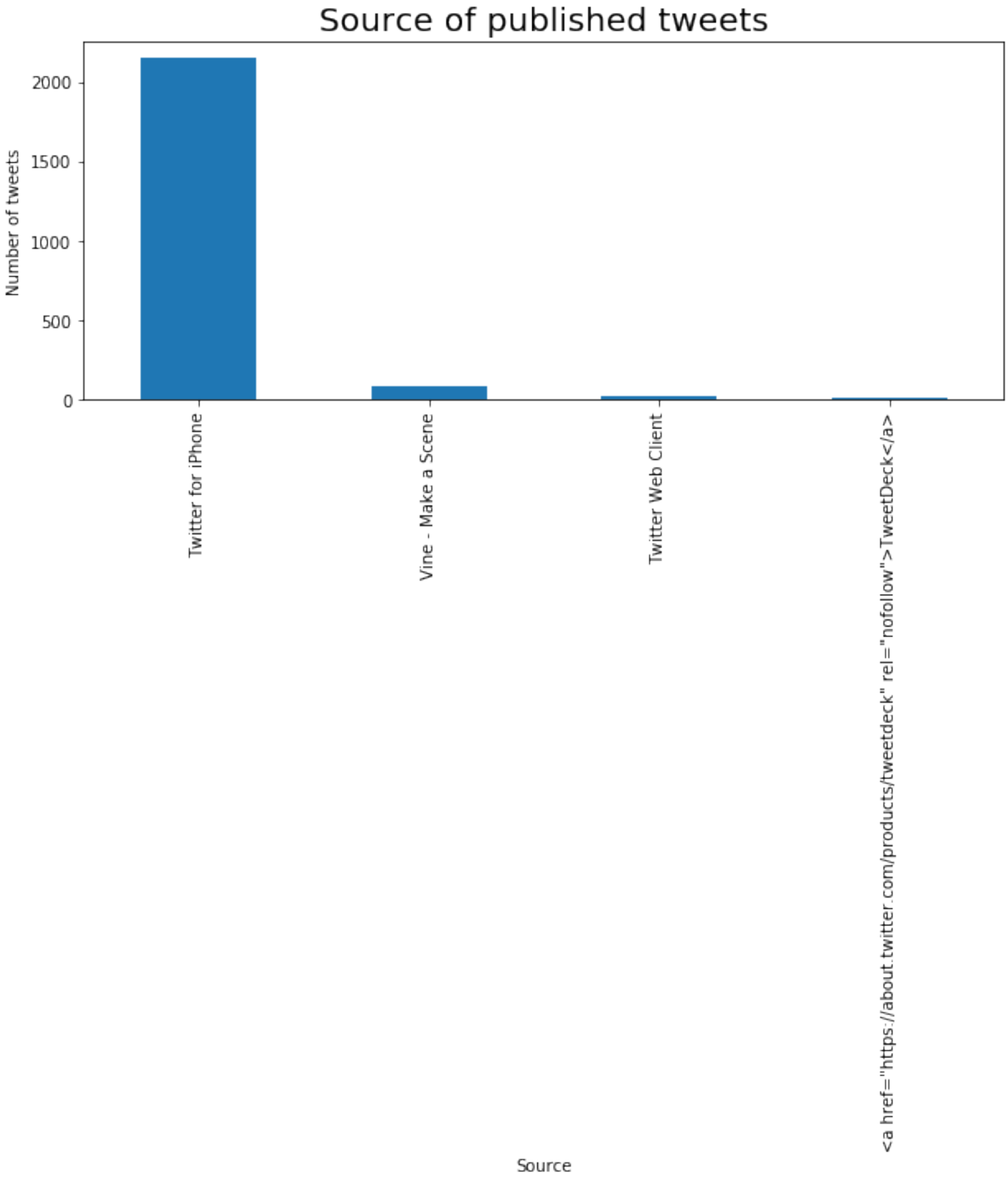
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2286 entries, 0 to 2355
Data columns (total 27 columns):
tweet_id                      2286 non-null object
in_reply_to_status_id         2286 non-null object
in_reply_to_user_id           2286 non-null object
timestamp                     2286 non-null object
source_x                      2286 non-null object
text                          2286 non-null object
retweeted_status_id           2286 non-null object
retweeted_status_timestamp    2286 non-null object
expanded_urls                 2286 non-null object
rating_numerator              2286 non-null int64
rating_denominator            2286 non-null int64
name                          2286 non-null object
Dog_Type                      374 non-null object
jpg_url                       2006 non-null object
img_num                       2006 non-null float6
4
Breed_probability_1           2006 non-null object
probability_1_conf            2006 non-null float6
4
probability_1_dog             2006 non-null object
Breed_probability_2           2006 non-null object
probability_2_conf            2006 non-null float6
4
probability_2_dog             2006 non-null object
Breed_probability_3           2006 non-null object
probability_3_conf            2006 non-null float6
4
probability_3_dog             2006 non-null object
favorite_count                2284 non-null float6
4
retweet_count                 2284 non-null float6
4
user                          2284 non-null object
dtypes: float64(6), int64(2), object(19)
memory usage: 500.1+ KB
```

Analysis:

Insights:

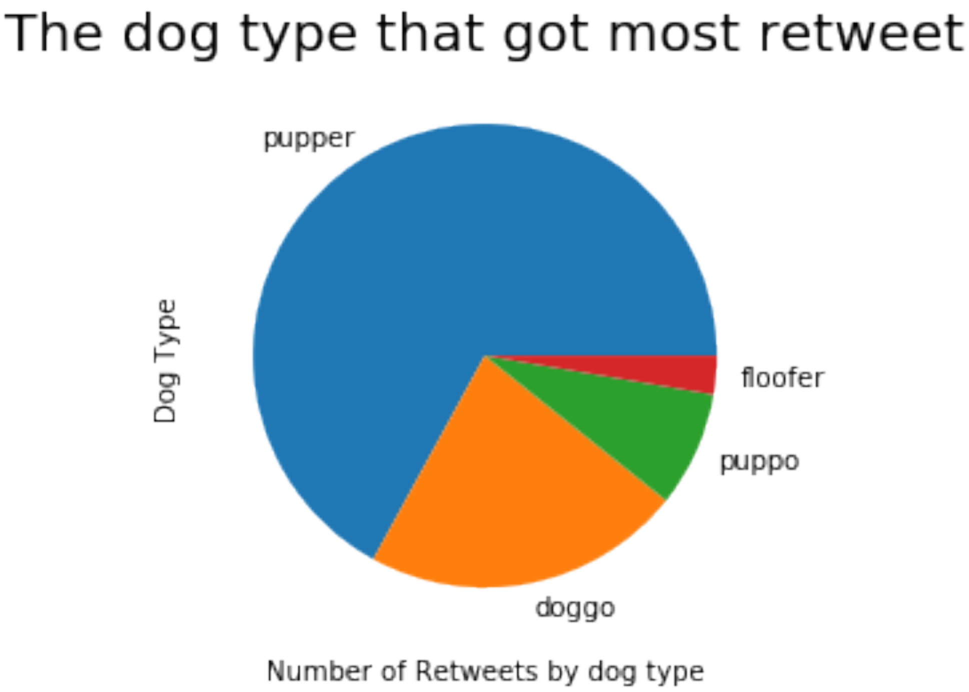
In [140]:

```
# Plot to show the type of dogs that got the most retweet
fig = plt.figure(figsize=(10,4))
tweeter_df['source_x'].value_counts().plot(kind='bar')
plt.title("Source of published tweets",fontsize=20)
plt.ylabel("Number of tweets")
plt.xlabel("Source");
```



In [143]:

```
# Plot to show the type of dogs that got the most retweet
fig = plt.figure(figsize=(15,4))
tweeter_df.groupby('Dog_Type').count()['retweet_count'].sort_values(ascending=False).nlargest(4).plot(kind='pie')
plt.title("The dog type that got most retweet",fontsize=20)
plt.ylabel("Dog Type")
plt.xlabel("Number of Retweets by dog type");
```



1) The dog type pupper got the most retweets

In [144]:

```
tweeter_df.describe()
```

Out[144]:

	rating_numerator	rating_denominator	img_num	probability
count	2286.000000	2286.000000	2006.000000	2006
mean	13.226159	10.455381	1.206381	0
std	46.556335	6.794867	0.563458	0
min	0.000000	0.000000	1.000000	0
25%	10.000000	10.000000	1.000000	0
50%	11.000000	10.000000	1.000000	0
75%	12.000000	10.000000	1.000000	0
max	1776.000000	170.000000	4.000000	1

2) The maximum retweets number that the account got is 81489

3) The minimum retweets number that the account got is only 1

4) Most tweets were published using twitter for iPhone. That gives an insight about the accounts are most probably personal accounts.

Storing the data

In [145]:

```
tweeter_df.to_csv('twitter_archive_master.csv')
```

The End