wrangle_act

May 6, 2021

WeRateDogs Data Wrangling Project
This data wrangling project focus on fixing the data quality and tidiness issues using python.

```
In [1]: import pandas as pd
        import json
        import requests
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
```

1.Gather Data

1. twitter_archive: The WeRateDogs Twitter archive, which is provides by the Udacity Course and I use pd.read_csv() to import them into dataframe.
2. image_predictions: The tweet image predictions, i.e., what breed of dog (or other objects, animal, etc.) is present in each tweet according to a neural network. This file ('image_predictions.tsv') is hosted on Udacity's servers and downloaded programmatically using the requests library and the provided url.
3. tweet_data: Using the tweet IDs in the WeRateDogs Twitter archive, query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called 'tweet_json_1.txt' file. Each tweet's JSON data is written to its own line.

```
In [2]: # Read in 1st dataset
        twitter_archive = pd.read_csv('twitter-archive-enhanced.csv')

In [3]: # Read in 2nd dataset
        image_predictions = pd.read_csv('image_predictions.tsv', delimiter = '\t')

In [4]: response = requests.get('https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2
        with open('image_predictions.tsv', mode = 'wb') as file:
            file.write(response.content)

In [5]: image_predictions = pd.read_csv('image_predictions.tsv', delimiter = '\t')
```

In [6]:
```
# use tweepy to query Twitter's API and hide the API info here

#consumer_key = ''
#consumer_secret = ''
#access_token = ''
#access_secret = ''

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

#api = tweepy.API(auth, wait_on_rate_limit=True)
#df_1=pd.read_csv('twitter-archive-enhanced.csv')

#tweet_ids = df_1.tweet_id.values
#len(tweet_ids)


#count = 0
#fails_dict = {}
#start = timer()

#with open('tweet_json_1.txt', 'w') as outfile:
    #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)
```

In [7]:
```
tweets = []

tweet_json = open('tweet_json_1.txt', 'r')

for line in tweet_json:
    tweet = json.loads(line)
    tweets.append(tweet)

tweet_json.close()
```

```
In [8]: """
        We can see lots of information from 'tweet_json' above, but here I'm only focus on certa
        such as 'retweet_count', 'favorite_count', and of course 'id' for table merge.
        """

        tweet_data = pd.DataFrame()  # create a empty dataframe for map the tweet_data info

        tweet_data['id'] = list(map(lambda tweet: tweet['id'], tweets))

        tweet_data['retweet_count'] = list(map(lambda tweet: tweet['retweet_count'], tweets))

        tweet_data['favorite_count'] = list(map(lambda tweet: tweet['favorite_count'], tweets))

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

Out[9]:

	id	retweet_count	favorite_count
0	892420643555336193	8853	39467
1	892177421306343426	6514	33819
2	891815181378084864	4328	25461
3	891689557279858688	8964	42908
4	891327558926688256	9774	41048

2. Assess Data

Inspecting data set for two things: data quality issues and lack of tidiness
Quality Issues means content issues like missing, duplicate, or incorrect data
Untidy Data has specific structural issues

In addition, four dimensions of data quality assessment help me guide the thought process while assessing the data. For example, Completeness: are there any missing data in specific rows or columns? Validity: are there any records not correct due to any reason? Accuracy: are there any extreme data or unusual data? Consistency: are they keep the consistence of scale standard or data type?

twitter_archive

```
In [10]: twitter_archive.head(3)
```

Out[10]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	\
0	892420643555336193	NaN	NaN	
1	892177421306343426	NaN	NaN	
2	891815181378084864	NaN	NaN	

	timestamp	\
0	2017-08-01 16:23:56 +0000	
1	2017-08-01 00:17:27 +0000	
2	2017-07-31 00:18:03 +0000	

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

```
2  <a href="http://twitter.com/download/iphone" r...

                                                text  retweeted_status_id  \
0  This is Phineas. He's a mystical boy. Only eve...                   NaN
1  This is Tilly. She's just checking pup on you...                   NaN
2  This is Archie. He is a rare Norwegian Pouncin...                   NaN

   retweeted_status_user_id retweeted_status_timestamp  \
0                       NaN                         NaN
1                       NaN                         NaN
2                       NaN                         NaN

                                        expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643...                13
1  https://twitter.com/dog_rates/status/892177421...                13
2  https://twitter.com/dog_rates/status/891815181...                12

   rating_denominator     name doggo floofer pupper puppo
0                  10  Phineas  None    None   None  None
1                  10    Tilly  None    None   None  None
2                  10   Archie  None    None   None  None
```

In [11]:
```
twitter_archive.tail(3)
```

Out[11]:
```
                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
2353  666033412701032449                    NaN                  NaN
2354  666029285002620928                    NaN                  NaN
2355  666020888022790149                    NaN                  NaN

                      timestamp  \
2353  2015-11-15 23:21:54 +0000
2354  2015-11-15 23:05:30 +0000
2355  2015-11-15 22:32:08 +0000

                                               source  \
2353  <a href="http://twitter.com/download/iphone" r...
2354  <a href="http://twitter.com/download/iphone" r...
2355  <a href="http://twitter.com/download/iphone" r...

                                                 text  retweeted_status_id  \
2353  Here is a very happy pup. Big fan of well-main...                   NaN
2354  This is a western brown Mitsubishi terrier. Up...                   NaN
2355  Here we have a Japanese Irish Setter. Lost eye...                   NaN

      retweeted_status_user_id retweeted_status_timestamp  \
2353                       NaN                         NaN
2354                       NaN                         NaN
2355                       NaN                         NaN
```

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

      rating_denominator  name doggo floofer pupper puppo
2353                  10     a  None    None   None  None
2354                  10     a  None    None   None  None
2355                  10  None  None    None   None  None
```

```
In [12]: 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 [13]: twitter_archive.isnull().sum()
```

```
Out[13]: tweet_id                    0
         in_reply_to_status_id    2278
         in_reply_to_user_id      2278
         timestamp                   0
         source                      0
         text                        0
         retweeted_status_id      2175
         retweeted_status_user_id 2175
         retweeted_status_timestamp 2175
         expanded_urls               59
```

```
rating_numerator                0
rating_denominator              0
name                            0
doggo                           0
floofer                         0
pupper                          0
puppo                           0
dtype: int64
```

```
In [14]: twitter_archive.name.str.islower().sum()
```

```
Out[14]: 109
```

```
In [15]: twitter_archive.describe()
```

```
Out[15]:            tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
count  2.356000e+03           7.800000e+01         7.800000e+01
mean   7.427716e+17           7.455079e+17         2.014171e+16
std    6.856705e+16           7.582492e+16         1.252797e+17
min    6.660209e+17           6.658147e+17         1.185634e+07
25%    6.783989e+17           6.757419e+17         3.086374e+08
50%    7.196279e+17           7.038708e+17         4.196984e+09
75%    7.993373e+17           8.257804e+17         4.196984e+09
max    8.924206e+17           8.862664e+17         8.405479e+17

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

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

```
In [16]: twitter_archive.rating_denominator.value_counts()
```

```
Out[16]: 10     2333
         11        3
```

```
50         3
80         2
20         2
2          1
16         1
40         1
70         1
15         1
90         1
110        1
120        1
130        1
150        1
170        1
7          1
0          1
Name: rating_denominator, dtype: int64
```

10 is the standard denominator, whereas others could be an error.

```
In [17]: twitter_archive.rating_numerator.value_counts()
```

```
Out[17]: 12      558
         11      464
         10      461
         13      351
         9       158
         8       102
         7        55
         14       54
         5        37
         6        32
         3        19
         4        17
         1         9
         2         9
         420       2
         0         2
         15        2
         75        2
         80        1
         20        1
         24        1
         26        1
         44        1
         50        1
         60        1
         165       1
```

```
84        1
88        1
144       1
182       1
143       1
666       1
960       1
1776      1
17        1
27        1
45        1
99        1
121       1
204       1
Name: rating_numerator, dtype: int64
```

Extreme values could be errors - 1776,960

```
In [18]: twitter_archive.source.value_counts()
```

```
Out[18]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
         <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
         <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
         <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
         Name: source, dtype: int64
```

image predictions

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

```
Out[19]:              tweet_id                                          jpg_url  \
         0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
         1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
         2  666033412701032449  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
         3  666044226329800704  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
         4  666049248165822465  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg

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

             p2_conf  p2_dog                   p3   p3_conf  p3_dog
         0  0.156665    True    Shetland_sheepdog  0.061428    True
         1  0.074192    True  Rhodesian_ridgeback  0.072010    True
         2  0.138584    True           bloodhound  0.116197    True
         3  0.360687    True   miniature_pinscher  0.222752    True
         4  0.243682    True             Doberman  0.154629    True
```

```
In [20]: image_predictions.tail(3)
```

Out[20]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
2072	891815181378084864	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	1	Chihuahua	0.716012	True	malamute	0.078253	True	kelpie	0.031379	True
2073	892177421306343426	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	1	Chihuahua	0.323581	True	Pekinese	0.090647	True	papillon	0.068957	True
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	1	orange	0.097049	False	bagel	0.085851	False	banana	0.076110	False

```
In [21]: 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 [22]: image_predictions.p1.value_counts()
```

Out[22]:

golden_retriever	150
Labrador_retriever	100
Pembroke	89
Chihuahua	83
pug	57
chow	44
Samoyed	43
toy_poodle	39
Pomeranian	38

malamute	30
cocker_spaniel	30
French_bulldog	26
miniature_pinscher	23
Chesapeake_Bay_retriever	23
seat_belt	22
German_shepherd	20
Siberian_husky	20
Staffordshire_bullterrier	20
web_site	19
Cardigan	19
teddy	18
Eskimo_dog	18
Shetland_sheepdog	18
Maltese_dog	18
beagle	18
Shih-Tzu	17
Rottweiler	17
Lakeland_terrier	17
Italian_greyhound	16
kuvasz	16
...	
bearskin	1
groenendael	1
sunglasses	1
electric_fan	1
pencil_box	1
polecat	1
binoculars	1
peacock	1
syringe	1
four-poster	1
beaver	1
cheeseburger	1
otter	1
cliff	1
Madagascar_cat	1
king_penguin	1
piggy_bank	1
bighorn	1
robin	1
wild_boar	1
microwave	1
grey_fox	1
candle	1
guenon	1
marmot	1
crash_helmet	1

```
canoe                          1
silky_terrier                  1
beach_wagon                    1
Egyptian_cat                   1
Name: p1, Length: 378, dtype: int64
```

```
In [23]: image_predictions.describe()
```

Out[23]:

	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

tweet_data

```
In [24]: tweet_data.head(3)
```

Out[24]:

	id	retweet_count	favorite_count
0	892420643555336193	8853	39467
1	892177421306343426	6514	33819
2	891815181378084864	4328	25461

```
In [25]: tweet_data.tail()
```

Out[25]:

	id	retweet_count	favorite_count
2349	666049248165822465	41	111
2350	666044226329800704	147	311
2351	666033412701032449	47	128
2352	666029285002620928	48	132
2353	666020888022790149	532	2535

```
In [26]: tweet_data=tweet_data.drop_duplicates()

In [27]: tweet_data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2354 entries, 0 to 2353
Data columns (total 3 columns):
id                2354 non-null int64
retweet_count     2354 non-null int64
favorite_count    2354 non-null int64
dtypes: int64(3)
memory usage: 73.6 KB
```

Cleaning Plan Summary

Tidiness Issues:

1.Columns 'doggo', 'floofer', 'pupper', 'puppo' in twitter_archive should belong to one column -- stage

2.The tweet_data table and image predictions table need to merge into the twitter_archive table.

Quality Issues :

1.Some columns have huge amount of missing values, for example, "in_reply_to_status_id", "in_reply_to_user_id", "retweeted_status_id", "in_reply_to_user_id", "retweeted_status_id", "retweeted_status_user_id", "retweeted_status_timestamp". I perfer to delete those columns directly,since not needed.

2.The varaible "expanded_urls" also has few missing values. Any ratings without images should not be considered.

3.The datatype of "timestamp" is incorrect.

4.Change the long url links to certain words.

5.The standard for "rating_denominator" is 10, but it includes some other numbers.

6.The "rating_numerator" also has some incorrect values.

7.Remove all invalid dog names

8.Change the column names for better readability in twitter_archive_clean and image_predictions_clean.

9.Capitalize the first letter of first prediction.

3. Clean Data

In [28]:
```
#making copies for cleaning
twitter_archive_clean=twitter_archive.copy()
image_predictions_clean=image_predictions.copy()
tweet_data_clean=tweet_data.copy()
```

twitter_archive table

Tidiness Issue 1:

Define: Create a new variable – 'stage' to show the four dog stages, drop the four columns, and fill the empty with NaN

Code:

In [29]:
```
twitter_archive_clean[twitter_archive_clean['doggo'] == 'None'].head()
```

Out[29]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000
4	891327558926688256	NaN	NaN	

```
4  2017-07-29 16:00:24 +0000

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

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

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

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

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

In [30]:
```
# use a for loop to replace all the 'None' before cat
stage = ['doggo','pupper', 'floofer', 'puppo' ]
for i in stage:
        twitter_archive_clean[i] = twitter_archive_clean[i].replace('None', '')
```

In [31]:
```
# use cat to combine
twitter_archive_clean['stage'] = twitter_archive_clean.doggo.str.cat(twitter_archive_cl

# drop the four old colomns
twitter_archive_clean = twitter_archive_clean.drop(['doggo','floofer','pupper','puppo']
```

```
# use np.nan to fill the empty
twitter_archive_clean['stage'] = twitter_archive_clean['stage'].replace('', np.nan)
```

Test

In [32]:
```
twitter_archive_clean.sample(5)
```

Out[32]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	expanded_urls	rating_numerator
1754	678798276842360832	NaN	NaN	2015-12-21 04:44:55 +0000	<a href="http://twitter.com/download/iphone" r...	This is Linda. She fucking hates trees. 7/10 h...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/678798276...	7
351	831322785565769729	NaN	NaN	2017-02-14 02:02:51 +0000	<a href="http://twitter.com/download/iphone" r...	This is Pete. He has no eyes. Needs a guide do...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/831322785...	12
715	783839966405230592	NaN	NaN	2016-10-06 01:23:05 +0000	<a href="http://twitter.com/download/iphone" r...	This is Riley. His owner put a donut pillow ar...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/783839966...	13
845	766423258543644672	NaN	NaN	2016-08-18 23:55:18 +0000	<a href="http://twitter.com/download/iphone" r...	This is Shadoe. Her tongue flies out of her mo...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/766423258...	9
1592	686394059078897668	NaN	NaN	2016-01-11 03:47:50 +0000	<a href="http://vine.co" rel="nofollow">Vine -...	This pup's having a nightmare that he forgot t...	NaN	NaN	NaN	https://vine.co/v/iMqBebnOvav	12

```
      rating_denominator    name  stage
1754                  10   Linda    NaN
351                   10    Pete  doggo
715                   10   Riley    NaN
845                   10  Shadoe    NaN
1592                  10    None    NaN
```

```
In [33]: print(twitter_archive_clean.shape)
         print(twitter_archive.shape)
```

```
(2356, 14)
(2356, 17)
```

Tidness Issue 2:

Define : Merge the tweet_data and image_predictions data into the twitter_archive using inner join.

Code:

```
In [34]: twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 14 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
stage                         380 non-null object
dtypes: float64(4), int64(3), object(7)
memory usage: 257.8+ KB
```

```
In [35]: tweet_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2354 entries, 0 to 2353
Data columns (total 3 columns):
id                2354 non-null int64
```

```
retweet_count     2354 non-null int64
favorite_count    2354 non-null int64
dtypes: int64(3)
memory usage: 73.6 KB
```

```
In [36]: 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 [37]: # rename the 'id' from `tweet_data` in preparing for table join
         tweet_data_clean.rename(columns={'id': 'tweet_id'}, inplace = True)

         # check the number of foreign key in two tables
         print(twitter_archive_clean.tweet_id.count())
         print(tweet_data_clean.tweet_id.count())
         print(image_predictions_clean.tweet_id.count())
```

```
2356
2354
2075
```

```
In [38]: # join three tables on 'tweet_id' and use inner join method
         twitter_archive_clean = pd.merge(twitter_archive_clean, image_predictions_clean, on='tw
         twitter_archive_clean = pd.merge(twitter_archive_clean, tweet_data_clean, on='tweet_id'
```

Test

```
In [39]: twitter_archive_clean.tweet_id.count()
```

```
Out[39]: 2073
```

```
In [40]: # drop the duplicates due to join
         twitter_archive_clean = twitter_archive_clean.drop_duplicates()

In [41]: print(twitter_archive_clean.tweet_id.count())
         print(tweet_data_clean.tweet_id.count())
         print(image_predictions_clean.tweet_id.count())

2073
2354
2075


In [42]: twitter_archive_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2072
Data columns (total 27 columns):
tweet_id                      2073 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2073 non-null object
source                        2073 non-null object
text                          2073 non-null object
retweeted_status_id           79 non-null float64
retweeted_status_user_id      79 non-null float64
retweeted_status_timestamp    79 non-null object
expanded_urls                 2073 non-null object
rating_numerator              2073 non-null int64
rating_denominator            2073 non-null int64
name                          2073 non-null object
stage                         320 non-null object
jpg_url                       2073 non-null object
img_num                       2073 non-null int64
p1                            2073 non-null object
p1_conf                       2073 non-null float64
p1_dog                        2073 non-null bool
p2                            2073 non-null object
p2_conf                       2073 non-null float64
p2_dog                        2073 non-null bool
p3                            2073 non-null object
p3_conf                       2073 non-null float64
p3_dog                        2073 non-null bool
retweet_count                 2073 non-null int64
favorite_count                2073 non-null int64
dtypes: bool(3), float64(7), int64(6), object(11)
memory usage: 411.0+ KB
```

Quality Issue 1

Define : Remove all the rows that have non-empty retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp and unnecessary columns directly ('retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', 'in_reply_to_status_id', 'in_reply_to_user_id', 'in_reply_to_user_id)

code

```
In [43]: #twitter_archive_clean['retweeted_status_id'].value_counts()
         #twitter_archive_clean['retweeted_status_id'].sample(10)
         #twitter_archive_clean.head()
         twitter_archive_clean = twitter_archive_clean[twitter_archive_clean['retweeted_status_i
         twitter_archive_clean.shape
         #twitter_archive_clean = twitter_archive_clean[(twitter_archive_clean['in_reply_to_stat

Out[43]: (1994, 27)

In [44]: # drop the columns unnecessary and contain huge amount of missing data
         columns_drop = ['retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_ti
                         'in_reply_to_user_id', 'in_reply_to_user_id']

         twitter_archive_clean = twitter_archive_clean.drop(columns_drop, axis = 1)
```

Test

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

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1994 entries, 0 to 2072
Data columns (total 22 columns):
tweet_id              1994 non-null int64
timestamp             1994 non-null object
source                1994 non-null object
text                  1994 non-null object
expanded_urls         1994 non-null object
rating_numerator      1994 non-null int64
rating_denominator    1994 non-null int64
name                  1994 non-null object
stage                 306 non-null object
jpg_url               1994 non-null object
img_num               1994 non-null int64
p1                    1994 non-null object
p1_conf               1994 non-null float64
p1_dog                1994 non-null bool
p2                    1994 non-null object
p2_conf               1994 non-null float64
p2_dog                1994 non-null bool
p3                    1994 non-null object
p3_conf               1994 non-null float64
p3_dog                1994 non-null bool
retweet_count         1994 non-null int64
```

```
favorite_count        1994 non-null int64
dtypes: bool(3), float64(3), int64(6), object(10)
memory usage: 317.4+ KB
```

Quality Issue 2
Define : Remove the records with no images information ('expanded_urls' is NaN)
Code

```
In [46]: twitter_archive_clean = twitter_archive_clean.dropna(subset = ['expanded_urls'])
```

test

```
In [47]: twitter_archive_clean.expanded_urls.isnull().sum()

Out[47]: 0
```

Quality Issue 3
Define : Change the datatype of 'timestamp' to datetime
Code

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

Test

```
In [49]: twitter_archive_clean['timestamp'].head()

Out[49]: 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
         Name: timestamp, dtype: datetime64[ns]
```

Quality Issue 4
Define : Optimize the source content by 'Twitter for iphone', 'Vine - Make a Scene', 'Twitter Web Client', and 'TweetDeck'.
Code

```
In [50]: twitter_archive_clean['source'].value_counts()

         # Four types of source: Twitter for iphone / Vine - Make a Scene / Twitter Web Client /

Out[50]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
         <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
         <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
         Name: source, dtype: int64
```

```
In [51]:
          twitter_archive_clean['source'] = twitter_archive_clean['source'].replace('<a href="htt
                                                                                    'Twitter for
          twitter_archive_clean['source'] =  twitter_archive_clean['source'].replace('<a href="ht
                                                                                    'Vine - Make
          twitter_archive_clean['source'] =  twitter_archive_clean['source'].replace('<a href="ht
                                                                                     'Twitter Web
          twitter_archive_clean['source'] =  twitter_archive_clean['source'].replace('<a href="ht
                                                                                     'TweetDeck')
```

Test

```
In [52]: twitter_archive_clean['source'].value_counts()

Out[52]: Twitter for iphone    1955
         Twitter Web Client      28
         TweetDeck               11
         Name: source, dtype: int64
```

Quality Issue 5

Define : 10 is the default value of 'rating_denominator', then correct the wrong values based on the corresponding text information.

Code

```
In [53]: twitter_archive_clean.rating_denominator.value_counts()

Out[53]: 10     1976
         50        3
         80        2
         11        2
         170       1
         150       1
         130       1
         120       1
         110       1
         90        1
         70        1
         40        1
         20        1
         7         1
         2         1
         Name: rating_denominator, dtype: int64

In [54]:  #filter the wrong rating_denominator values

          df1 = twitter_archive_clean[twitter_archive_clean['rating_denominator'] != 10]

          df1[['tweet_id','text','rating_numerator','rating_denominator']]
```

Out[54]:

	tweet_id	text	\
345	820690176645140481	The floofs have been released I repeat the flo...	
415	810984652412424192	Meet Sam. She smiles 24/7 & secretly aspir...	
734	758467244762497024	Why does this never happen at my front door...	
876	740373189193256964	After so many requests, this is Bretagne. She ...	
924	731156023742988288	Say hello to this unbelievably well behaved sq...	
967	722974582966214656	Happy 4/20 from the squad! 13/10 for all https...	
1001	716439118184652801	This is Bluebert. He just saw that both #Final...	
1022	713900603437621249	Happy Saturday here's 9 puppers on a bench. 99...	
1047	710658690886586372	Here's a brigade of puppers. All look very pre...	
1065	709198395643068416	From left to right:\nCletus, Jerome, Alejandro...	
1131	704054845121142784	Here is a whole flock of puppers. 60/50 I'll ...	
1207	697463031882764288	Happy Wednesday here's a bucket of pups. 44/40...	
1379	684225744407494656	Two sneaky puppers were not initially seen, mo...	
1380	684222868335505415	Someone help the girl is being mugged. Several...	
1405	682962037429899265	This is Darrel. He just robbed a 7/11 and is i...	
1512	677716515794329600	IT'S PUPPERGEDDON. Total of 144/120 ...I think...	
1571	675853064436391936	Here we have an entire platoon of puppers. Tot...	
2052	666287406224695296	This is an Albanian 3 1/2 legged Episcopalian...	

	rating_numerator	rating_denominator
345	84	70
415	24	7
734	165	150
876	9	11
924	204	170
967	4	20
1001	50	50
1022	99	90
1047	80	80
1065	45	50
1131	60	50
1207	44	40
1379	143	130
1380	121	110
1405	7	11
1512	144	120
1571	88	80
2052	1	2

In [55]:
```
# tweet_id : 740373189193256964,722974582966214656,716439118184652801,682962037429899265
```

```
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 740373189193256964, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 722974582966214656, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 716439118184652801, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 682962037429899265, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 666287406224695296, ['ratin
```

Test

```
In [56]: twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 740373189193256964]
```

```
Out[56]:              tweet_id           timestamp              source  \
876  740373189193256964 2016-06-08 02:41:38  Twitter for iphone

                                                    text  \
876  After so many requests, this is Bretagne. She ...

                                           expanded_urls  rating_numerator  \
876  https://twitter.com/dog_rates/status/740373189...                 9

     rating_denominator  name stage  \
876                  10  None   NaN

                                              jpg_url       ...        \
876  https://pbs.twimg.com/media/CkZVdJ6WYAAXZ5A.jpg       ...

      p1_conf p1_dog      p2   p2_conf p2_dog                 p3   p3_conf  \
876  0.807644   True  kuvasz  0.101286   True  Labrador_retriever  0.023785

    p3_dog  retweet_count  favorite_count
876   True           9220           20648

[1 rows x 22 columns]
```

Quality Issue 6
Define : Correct the 'rating_numerator' values from the text information
Code

```
In [57]: # tweet_id : 740373189193256964,722974582966214656,716439118184652801,682962037429899265
```

```
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 740373189193256964, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 722974582966214656, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 716439118184652801, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 682962037429899265, ['ratin
twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 666287406224695296, ['ratin
```

Test

```
In [58]: twitter_archive_clean.loc[twitter_archive_clean.tweet_id == 740373189193256964]
```

```
Out[58]:              tweet_id           timestamp              source  \
876  740373189193256964 2016-06-08 02:41:38  Twitter for iphone

                                                    text  \
876  After so many requests, this is Bretagne. She ...

                                           expanded_urls  rating_numerator  \
876  https://twitter.com/dog_rates/status/740373189...                14
```

```
     rating_denominator  name stage  \
876                  10  None   NaN

                                              jpg_url       ...        \
876  https://pbs.twimg.com/media/CkZVdJ6WYAAXZ5A.jpg       ...

      p1_conf p1_dog      p2   p2_conf p2_dog                  p3   p3_conf  \
876  0.807644   True  kuvasz  0.101286   True  Labrador_retriever  0.023785

    p3_dog  retweet_count  favorite_count
876   True           9220           20648

[1 rows x 22 columns]
```

```
In [59]: twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1994 entries, 0 to 2072
Data columns (total 22 columns):
tweet_id              1994 non-null int64
timestamp             1994 non-null datetime64[ns]
source                1994 non-null object
text                  1994 non-null object
expanded_urls         1994 non-null object
rating_numerator      1994 non-null int64
rating_denominator    1994 non-null int64
name                  1994 non-null object
stage                 306 non-null object
jpg_url               1994 non-null object
img_num               1994 non-null int64
p1                    1994 non-null object
p1_conf               1994 non-null float64
p1_dog                1994 non-null bool
p2                    1994 non-null object
p2_conf               1994 non-null float64
p2_dog                1994 non-null bool
p3                    1994 non-null object
p3_conf               1994 non-null float64
p3_dog                1994 non-null bool
retweet_count         1994 non-null int64
favorite_count        1994 non-null int64
dtypes: bool(3), datetime64[ns](1), float64(3), int64(6), object(9)
memory usage: 317.4+ KB
```

Quality Issue 7
Define : Remove all invalid dog names
Code

```
In [60]: twitter_archive_clean[twitter_archive_clean.name.str.islower()].name.value_counts()
```

```
Out[60]: a               55
         the              7
         an               6
         one              4
         very             4
         quite            3
         just             3
         getting          2
         not              1
         incredibly       1
         my               1
         infuriating      1
         light            1
         all              1
         actually         1
         his              1
         officially       1
         such             1
         unacceptable     1
         by               1
         this             1
         space            1
         Name: name, dtype: int64
```

```
In [61]: mask = twitter_archive_clean.name.str.contains('^[a-z]', regex = True)
         twitter_archive_clean[mask].name.value_counts().sort_index()
```

```
Out[61]: a               55
         actually         1
         all              1
         an               6
         by               1
         getting          2
         his              1
         incredibly       1
         infuriating      1
         just             3
         light            1
         my               1
         not              1
         officially       1
         one              4
         quite            3
         space            1
         such             1
         the              7
```

```
this             1
unacceptable     1
very             4
Name: name, dtype: int64
```

```
In [62]: remove_names=twitter_archive_clean[twitter_archive_clean.name.str.islower()]['name'].un
```

```
In [63]: len(remove_names)
```

```
Out[63]: 22
```

```
In [64]: twitter_archive_clean=twitter_archive_clean[~twitter_archive_clean['name'].isin(remove_
```

```
In [65]: twitter_archive_clean.shape
```

```
Out[65]: (1896, 22)
```

```
In [66]: twitter_archive_clean['name'].value_counts().head(10)
```

```
Out[66]: None       546
         Charlie     11
         Lucy        10
         Oliver      10
         Cooper      10
         Penny        9
         Tucker       9
         Sadie        8
         Winston      8
         Toby         7
         Name: name, dtype: int64
```

Test

```
In [67]: twitter_archive_clean['name'].value_counts().head(10)
```

```
Out[67]: None       546
         Charlie     11
         Lucy        10
         Oliver      10
         Cooper      10
         Penny        9
         Tucker       9
         Sadie        8
         Winston      8
         Toby         7
         Name: name, dtype: int64
```

Quality Issue 8

Define : Change the column names for better readability in twitter_archive_clean.

```
In [68]: twitter_archive_clean.head()
```

Out[68]:

	tweet_id	timestamp	source	text	expanded_urls	rating_numerator	rating_denominator	name	stage
0	892420643555336193	2017-08-01 16:23:56	Twitter for iphone	This is Phineas. He's a mystical boy. Only eve...	https://twitter.com/dog_rates/status/892420643...	13	10	Phineas	NaN
1	892177421306343426	2017-08-01 00:17:27	Twitter for iphone	This is Tilly. She's just checking pup on you...	https://twitter.com/dog_rates/status/892177421...	13	10	Tilly	NaN
2	891815181378084864	2017-07-31 00:18:03	Twitter for iphone	This is Archie. He is a rare Norwegian Pouncin...	https://twitter.com/dog_rates/status/891815181...	12	10	Archie	NaN
3	891689557279858688	2017-07-30 15:58:51	Twitter for iphone	This is Darla. She commenced a snooze mid meal...	https://twitter.com/dog_rates/status/891689557...	13	10	Darla	NaN
4	891327558926688256	2017-07-29 16:00:24	Twitter for iphone	This is Franklin. He would like you to stop ca...	https://twitter.com/dog_rates/status/891327558...	12	10	Franklin	NaN

	jpg_url	...	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog	retweet_count	favorite_count
0	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	...	0.097049	False	bagel	0.085851	False	banana	0.076110	False	8853	39467
1	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	...	0.323581	True	Pekinese	0.090647	True	papillon	0.068957	True	6514	33819
2	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	...	0.716012	True	malamute	0.078253	True	kelpie	0.031379	True	4328	25461
3	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	...	0.170278	False	Labrador_retriever	0.168086	True	spatula				
4	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg	...	0.555712	True	English_springer	0.225770	True	German_short-haired_pointer				

```
3  0.040836  False           8964           42908
4  0.175219   True           9774           41048

[5 rows x 22 columns]
```

Code:

```
In [69]: # change the column names
         twitter_archive_clean.rename(columns={'p1':'first_prediction', 'p1_conf': 'first_confid
                                               'p2': 'second_prediction', 'p2_conf': 'second_confide
                                               'p3': 'third_prediction', 'p3_conf': 'third_confidenc
         image_predictions_clean.rename(columns={'p1':'first_prediction', 'p1_conf': 'first_conf
                                               'p2': 'second_prediction', 'p2_conf': 'second_confide
                                               'p3': 'third_prediction', 'p3_conf': 'third_confidenc
```

Test

```
In [70]: twitter_archive_clean.head()
```

```
Out[70]:             tweet_id           timestamp              source  \
         0  892420643555336193 2017-08-01 16:23:56  Twitter for iphone
         1  892177421306343426 2017-08-01 00:17:27  Twitter for iphone
         2  891815181378084864 2017-07-31 00:18:03  Twitter for iphone
         3  891689557279858688 2017-07-30 15:58:51  Twitter for iphone
         4  891327558926688256 2017-07-29 16:00:24  Twitter for iphone

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

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

            rating_denominator      name stage  \
         0                  10   Phineas   NaN
         1                  10     Tilly   NaN
         2                  10    Archie   NaN
         3                  10     Darla   NaN
         4                  10  Franklin   NaN

                                                 jpg_url       ...        \
         0  https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg       ...
```

```
1  https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg       ...
2  https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg       ...
3  https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg       ...
4  https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg       ...

   first_confidence first_dog   second_prediction  second_confidence  \
0          0.097049     False               bagel           0.085851
1          0.323581      True            Pekinese           0.090647
2          0.716012      True            malamute           0.078253
3          0.170278     False  Labrador_retriever           0.168086
4          0.555712      True    English_springer           0.225770

  second_dog             third_prediction  third_confidence third_dog  \
0      False                       banana          0.076110     False
1       True                     papillon          0.068957      True
2       True                       kelpie          0.031379      True
3       True                      spatula          0.040836     False
4       True  German_short-haired_pointer          0.175219      True

   retweet_count  favorite_count
0           8853           39467
1           6514           33819
2           4328           25461
3           8964           42908
4           9774           41048

[5 rows x 22 columns]
```

Quality Issue 10

Define : Capitalize the first letter of first prediction (I could do that for all the predictions, but I decide to only apply to the first prediction since this variable is the important one).

Code

```
In [71]: twitter_archive_clean['first_prediction'] = twitter_archive_clean.first_prediction.str.
```

Test

```
In [72]: twitter_archive_clean.head()
```

```
Out[72]:             tweet_id           timestamp              source  \
0  892420643555336193 2017-08-01 16:23:56  Twitter for iphone
1  892177421306343426 2017-08-01 00:17:27  Twitter for iphone
2  891815181378084864 2017-07-31 00:18:03  Twitter for iphone
3  891689557279858688 2017-07-30 15:58:51  Twitter for iphone
4  891327558926688256 2017-07-29 16:00:24  Twitter for iphone

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

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

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

   rating_denominator      name stage  \
0                  10   Phineas   NaN
1                  10     Tilly   NaN
2                  10    Archie   NaN
3                  10     Darla   NaN
4                  10  Franklin   NaN

                                         jpg_url       ...        \
0  https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg       ...
1  https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg       ...
2  https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg       ...
3  https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg       ...
4  https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg       ...

   first_confidence first_dog   second_prediction  second_confidence  \
0          0.097049     False               bagel           0.085851
1          0.323581      True            Pekinese           0.090647
2          0.716012      True            malamute           0.078253
3          0.170278     False  Labrador_retriever           0.168086
4          0.555712      True    English_springer           0.225770

  second_dog             third_prediction  third_confidence third_dog  \
0      False                       banana          0.076110     False
1       True                     papillon          0.068957      True
2       True                       kelpie          0.031379      True
3       True                      spatula          0.040836     False
4       True  German_short-haired_pointer          0.175219      True

   retweet_count  favorite_count
0           8853           39467
1           6514           33819
2           4328           25461
3           8964           42908
4           9774           41048

[5 rows x 22 columns]
```

```
In [73]: twitter_archive_clean['first_prediction'].value_counts().head()

Out[73]: Golden_retriever      135
         Labrador_retriever     92
         Pembroke               84
         Chihuahua              77
         Pug                    51
         Name: first_prediction, dtype: int64
```

Final Tests

```
In [74]: twitter_archive_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1896 entries, 0 to 2072
Data columns (total 22 columns):
tweet_id              1896 non-null int64
timestamp             1896 non-null datetime64[ns]
source                1896 non-null object
text                  1896 non-null object
expanded_urls         1896 non-null object
rating_numerator      1896 non-null int64
rating_denominator    1896 non-null int64
name                  1896 non-null object
stage                 294 non-null object
jpg_url               1896 non-null object
img_num               1896 non-null int64
first_prediction      1896 non-null object
first_confidence      1896 non-null float64
first_dog             1896 non-null bool
second_prediction     1896 non-null object
second_confidence     1896 non-null float64
second_dog            1896 non-null bool
third_prediction      1896 non-null object
third_confidence      1896 non-null float64
third_dog             1896 non-null bool
retweet_count         1896 non-null int64
favorite_count        1896 non-null int64
dtypes: bool(3), datetime64[ns](1), float64(3), int64(6), object(9)
memory usage: 301.8+ KB


In [75]: tweet_data_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2354 entries, 0 to 2353
Data columns (total 3 columns):
tweet_id          2354 non-null int64
retweet_count     2354 non-null int64
```

```
favorite_count    2354 non-null int64
dtypes: int64(3)
memory usage: 73.6 KB
```

Store

```
In [76]: twitter_archive_clean.to_csv('twitter_archive_master.csv', encoding='utf-8')

In [77]: #image_predictions_clean.to_csv('image_predictions_master.csv',encoding='utf-8')

In [78]: #tweet_data_clean.to_csv('tweet_data_master.csv',encoding='utf-8')
```

Analyse and Visualize:

```
In [79]: twitter_archive_clean.head()
```

Out[79]:

	tweet_id	timestamp	source	text	expanded_urls	rating_numerator	rating_denominator	name	stage	jpg_url	...
0	892420643555336193	2017-08-01 16:23:56	Twitter for iphone	This is Phineas. He's a mystical boy. Only eve...	https://twitter.com/dog_rates/status/892420643...	13	10	Phineas	NaN	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	...
1	892177421306343426	2017-08-01 00:17:27	Twitter for iphone	This is Tilly. She's just checking pup on you...	https://twitter.com/dog_rates/status/892177421...	13	10	Tilly	NaN	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	...
2	891815181378084864	2017-07-31 00:18:03	Twitter for iphone	This is Archie. He is a rare Norwegian Pouncin...	https://twitter.com/dog_rates/status/891815181...	12	10	Archie	NaN	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	...
3	891689557279858688	2017-07-30 15:58:51	Twitter for iphone	This is Darla. She commenced a snooze mid meal...	https://twitter.com/dog_rates/status/891689557...	13	10	Darla	NaN		
4	891327558926688256	2017-07-29 16:00:24	Twitter for iphone	This is Franklin. He would like you to stop ca...	https://twitter.com/dog_rates/status/891327558...	12	10	Franklin	NaN		

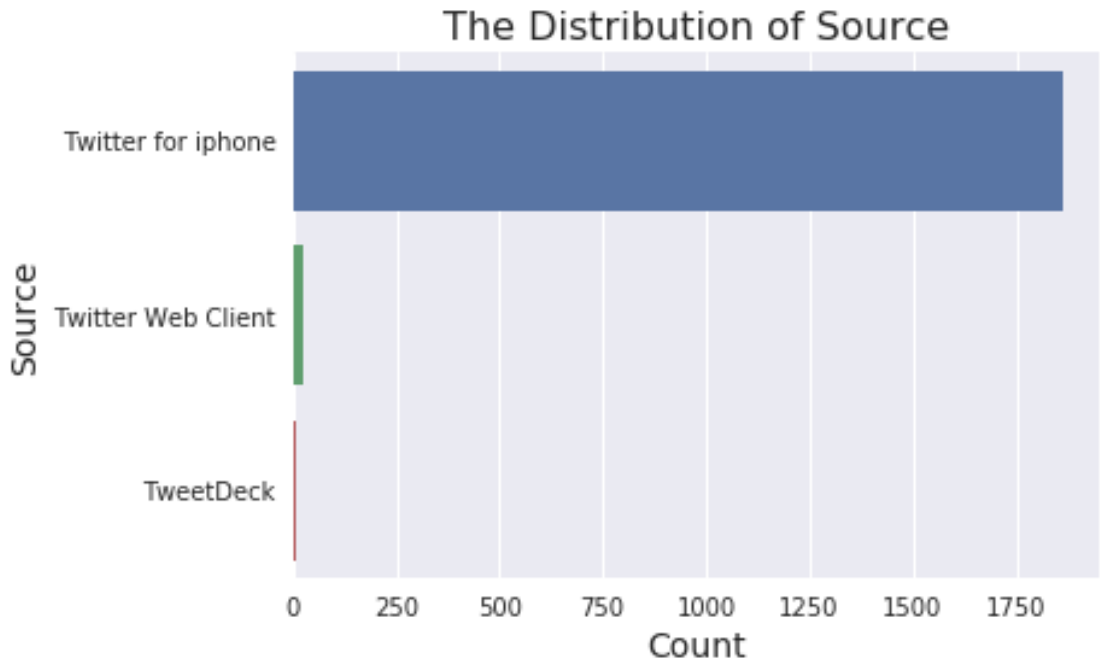
```
3  https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg       ...
4  https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg       ...

   first_confidence first_dog   second_prediction  second_confidence  \
0          0.097049     False               bagel           0.085851
1          0.323581      True            Pekinese           0.090647
2          0.716012      True            malamute           0.078253
3          0.170278     False  Labrador_retriever           0.168086
4          0.555712      True    English_springer           0.225770

  second_dog             third_prediction  third_confidence third_dog  \
0      False                       banana          0.076110     False
1       True                     papillon          0.068957      True
2       True                       kelpie          0.031379      True
3       True                      spatula          0.040836     False
4       True  German_short-haired_pointer          0.175219      True

   retweet_count  favorite_count
0           8853           39467
1           6514           33819
2           4328           25461
3           8964           42908
4           9774           41048

[5 rows x 22 columns]
```

Distribution of Source

```
In [80]: sorted_source = twitter_archive_clean['source'].value_counts().index
         print(twitter_archive_clean['source'].value_counts())
         sns.set(style="darkgrid")
         sns.countplot(data = twitter_archive_clean, y = 'source', order = sorted_source)
         plt.xticks(rotation = 360)
         plt.xlabel('Count', fontsize=14)
         plt.ylabel('Source', fontsize=14)
         plt.title('The Distribution of Source',fontsize=16)
```

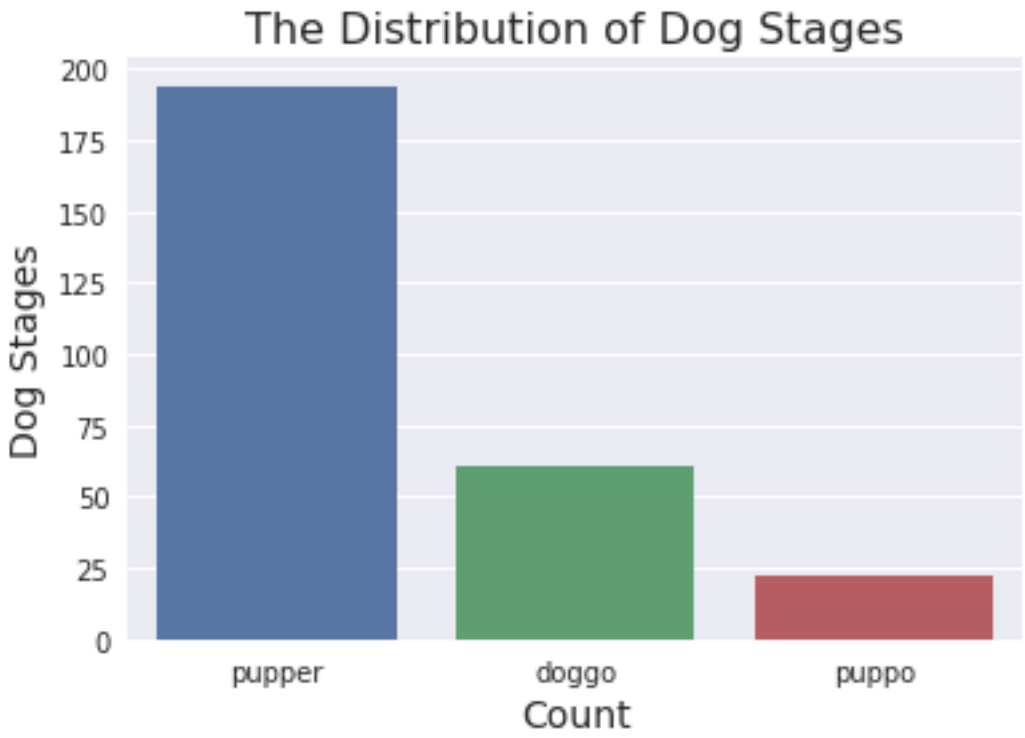
```
Twitter for iphone    1861
Twitter Web Client      25
TweetDeck               10
Name: source, dtype: int64
```

```
Out[80]: Text(0.5,1,'The Distribution of Source')
```



This plot above shows the distribution of source. We can see that the dominate source of tweets is from iPhone twitter app, which is more than 95Distribution of Dog stages

```
In [81]: sorted_stage = twitter_archive_clean['stage'].value_counts().head(3).index
         sns.set(style="darkgrid")
         sns.countplot(data = twitter_archive_clean, x = 'stage', order = sorted_stage, orient =
         plt.xticks(rotation = 360)
         plt.xlabel('Count', fontsize=14)
         plt.ylabel('Dog Stages', fontsize=14)
         plt.title('The Distribution of Dog Stages',fontsize=16)

Out[81]: Text(0.5,1,'The Distribution of Dog Stages')
```



Similarly, I check the distribution of dog stages. It shows that 'pupper' (a small doggo, usually younger) is the most popular dog stage, followed by 'doggo' and 'puppo'. It could be due to the young and unmatured dog is usually cuter than the adult dog. It should also be noticed that there's huge amount missing data in dog stages, thus the distribution may not reflect the truth.

Classification of Dogs Result Analysis

```
In [82]: image_predictions_clean.head()
```

Out[82]:

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

	img_num	first_prediction	first_confidence	first_dog
0	1	Welsh_springer_spaniel	0.465074	True
1	1	redbone	0.506826	True
2	1	German_shepherd	0.596461	True
3	1	Rhodesian_ridgeback	0.408143	True
4	1	miniature_pinscher	0.560311	True

	second_prediction	second_confidence	second_dog	third_prediction
0	collie	0.156665	True	Shetland_sheepdog

```
1  miniature_pinscher           0.074192        True  Rhodesian_ridgeback
2            malinois           0.138584        True            bloodhound
3             redbone           0.360687        True   miniature_pinscher
4          Rottweiler           0.243682        True              Doberman

   third_confidence  third_dog
0          0.061428       True
1          0.072010       True
2          0.116197       True
3          0.222752       True
4          0.154629       True
```

The 'image_predictions' table stores the result of a classification of dog breeds through a neural network. I am curious about the how this model works? What's the accuracy of this model? Therefore, I analyze and visualize the results in below.

```
In [83]: image_predictions_clean['first_prediction'].value_counts().head(10)
```

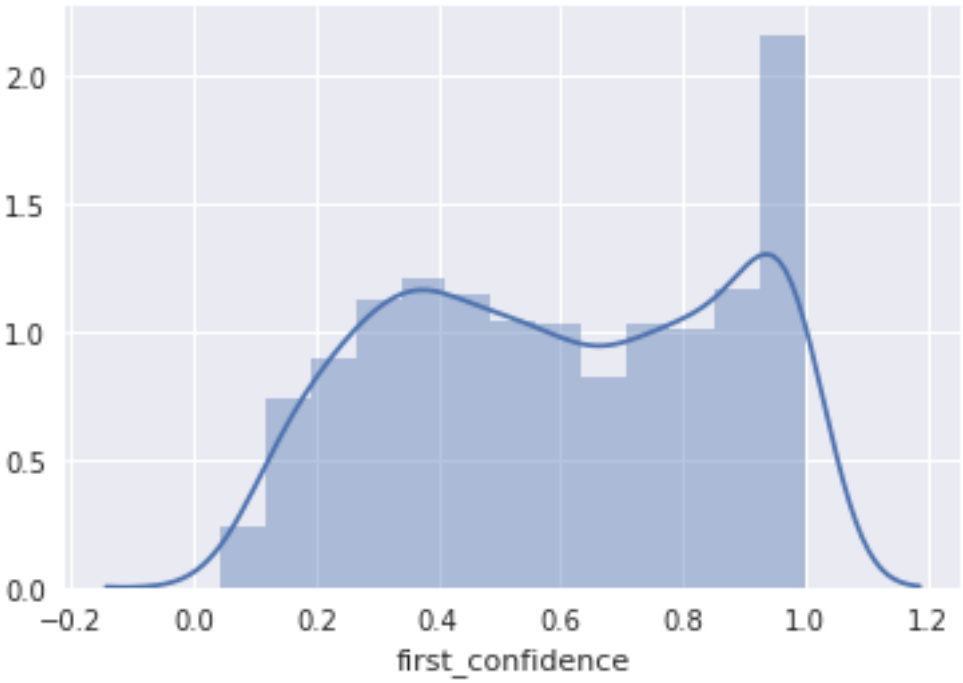
```
Out[83]: golden_retriever      150
         Labrador_retriever    100
         Pembroke               89
         Chihuahua              83
         pug                    57
         chow                   44
         Samoyed                43
         toy_poodle             39
         Pomeranian             38
         malamute               30
         Name: first_prediction, dtype: int64
```

```
In [84]: twitter_archive_clean['first_prediction'].value_counts().head(10)
```

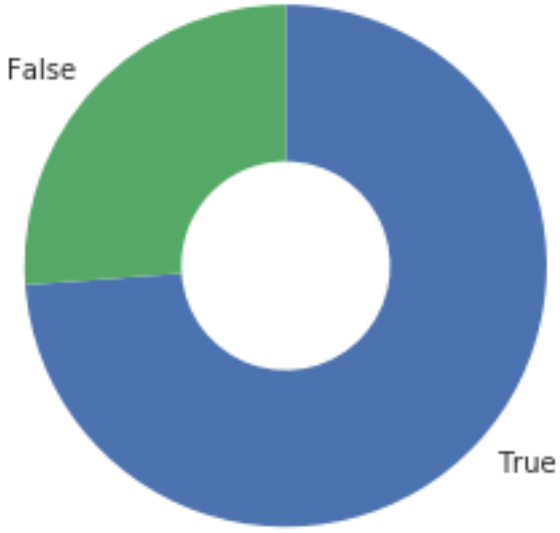
```
Out[84]: Golden_retriever      135
         Labrador_retriever     92
         Pembroke               84
         Chihuahua              77
         Pug                    51
         Chow                   38
         Samoyed                38
         Pomeranian             36
         Toy_poodle             34
         Malamute               28
         Name: first_prediction, dtype: int64
```

These breeds above are the top 10 dog breeds this model predicted. Golden retriever and Labrador retriever are top 2 and both over 100 predictions. It could be because those two are most common breeds in U.S. We have more image data on those breeds, and thus trained a better result.

```
In [85]: sns.set(style='darkgrid')
         ax = sns.distplot(image_predictions_clean['first_confidence'])
```




```
In [86]: sns.set(style='darkgrid')
         sorted_p1 = image_predictions_clean['first_dog'].value_counts()
         plt.pie(sorted_p1, labels = sorted_p1.index, startangle = 90, counterclock = False,
                wedgeprops = {'width': 0.6})
         plt.axis('square')

Out[86]: (-1.1086866580463621,
          1.1197796704326126,
          -1.1183657634231952,
          1.1101005650557796)
```



The first plot above shows the prediction success rate of whether or not first prediction is a breed of dog. The pie chart indicates almost 2/3 situations the predictions are correct, even though this result is not good enough for a deep learning model. The second plot shows how confident the algorithm is in its first prediction. We can see 100% is the most cases, however the amounts of 0.1 to 0.8 dominate the entire distribution. That also could suggest that the model is not good enough.

Reference

Tweepy documentation: https://media.readthedocs.org/pdf/tweepy/latest/tweepy.pdf

WeRateDogs Twitter: https://twitter.com/dog_rates?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr

In [ ]: