Data wrangling

(1) Import libraries which I used:

Importing libraries:

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

(2) Data gathering:

1-tweet-json.txt

2-image-prediction.tsv

3-twitter-archive-enhanced-2.csv

4-create data frame twitter_data from tweet-json.txt

Gathering:

In [2]:
```
tweet_json=[]
with open('tweet-json.txt') as file:
    for line in file:
        tweet_json.append(json.loads(line))
```

In [3]:
```
url="https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv"
image=requests.get(url)
with open ('image-predictions.tsv',mode='wb')as file:
    file.write(image.content)
```

In [4]:
```
image_predictions=pd.read_csv('image-predictions.tsv',sep='\t')
twitter_archive=pd.read_csv('twitter-archive-enhanced-2.csv')
```

In [5]:
```
tweet_json[0]
```

Out[5]:
```
{'created_at': 'Tue Aug 01 16:23:56 +0000 2017',
 'id': 892420643555336193,
 'id_str': '892420643555336193',
 'full_text': "This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU",
 'truncated': False,
 'display_text_range': [0, 85],
 'entities': {'hashtags': [],
  'symbols': [],
  'user_mentions': [],
  'urls': [],
  'media': [{'id': 892420639486877696,
```

Create a new dataframe with coulmns which i will use.

In [6]:
```
twitter_data=pd.DataFrame(tweet_json,columns=['id','retweet_count','favorite_count'])
twitter_data=twitter_data.rename(columns={'id':'tweet_id'})
twitter_data.head()
```

Out[6]:

	tweet_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

In [7]:
```
twitter_data.to_csv('twitter_data.csv',index=False)
```

In [8]:
```
twitter_data=pd.read_csv('twitter_data.csv')
twitter_data
```

Out[8]:

	tweet_id	retweet_count	favorite_count
0	892420643555336193	8853	39467
1	892177421306343426	6514	33819
2	891815181378084864	4328	25461
3	891689557279858688	8964	42908

(3) Data assessing:

1-using describe(): to know the summation of each column ,if there missing value or not And some statistics operations like(mean,max,min,….)

2-using info(): to know columns names ,datatypes and the non null count

3- using duplicated():to know if there any duplicated value

4- image_predictions: to see the dataframe

5- sum(image_predictions.p1_conf>image_predictions.p2_conf) : to know how many attributies that p1_conf bigger than p2_conf

6- sum(image_predictions.p1_conf>image_predictions.p3_conf): to know that p1_conf has the biggest conf

7-using value_counts() :to know the number of iteration for each value in column

Assessing:

Using describe(): to know the summation of each column ,if there missing value or not , And some statistics operations like(mean,max,min,....) ¶

In [9]:
```
twitter_data.describe()
```

Out[9]:

	tweet_id	retweet_count	favorite_count
count	2.354000e+03	2354.000000	2354.000000
mean	7.426978e+17	3164.797366	8080.968564
std	6.852812e+16	5284.770364	11814.771334
min	6.660209e+17	0.000000	0.000000
25%	6.783975e+17	624.500000	1415.000000
50%	7.194596e+17	1473.500000	3603.500000
75%	7.993058e+17	3652.000000	10122.250000
max	8.924206e+17	79515.000000	132810.000000

Using info(): to know columns names ,datatypes and the non null count.

In [10]:
```
twitter_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   tweet_id        2354 non-null   int64
 1   retweet_count   2354 non-null   int64
 2   favorite_count  2354 non-null   int64
dtypes: int64(3)
memory usage: 55.3 KB
```

In [11]:
```
sum(twitter_data.tweet_id.duplicated())
```

Out[11]: 0

In [12]:
```
#to show the dataframe:
image_predictions
```

Out[12]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True	miniature_pinscher	0.074192	
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461	True	malinois	0.138584	
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.408143	True	redbone	0.360687	
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311	True	Rottweiler	0.243682	
...	...	...	...	...	...	...	...	...	...
2070	891327558926688256	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg	2	basset	0.555712	True	English_springer	0.225770	
2071	891689557279858688	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	1	paper_towel	0.170278	False	Labrador_retriever	0.168086	
2072	891815181378084864	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	1	Chihuahua	0.716012	True	malamute	0.078253	
2073	892177421306343426	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	1	Chihuahua	0.323581	True	Pekinese	0.090647	
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	1	orange	0.097049	False	bagel	0.085851	

2075 rows × 12 columns

we want to know the better conf ALG from the three ALG ... it's (p1_conf).

In [13]:
```
sum(image_predictions.p1_conf>image_predictions.p2_conf)
```

Out[13]: 2075

In [14]:
```
sum(image_predictions.p1_conf>image_predictions.p3_conf)
```

Out[14]: 2075

In [16]:
```
image_predictions.loc[(image_predictions.p1_dog==False)&(image_predictions.p2_dog==False)&(image_predictions.p3_dog==False)]
#so there are some noisy photos
```

Out[16]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg	1	box_turtle	0.933012	False	mud_turtle	0.045885	False
17	666104133288665088	https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg	1	hen	0.965932	False	cock	0.033919	False
18	666268910803644416	https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg	1	desktop_computer	0.086502	False	desk	0.085547	False
21	666293911632134144	https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg	1	three-toed_sloth	0.914671	False	otter	0.015250	False
25	666362758909284353	https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg	1	guinea_pig	0.996496	False	skunk	0.002402	False
...	...	...	...	...	...	...	...	...	...
2021	880935762899988482	https://pbs.twimg.com/media/DDm2Z5aXUAEDS2u.jpg	1	street_sign	0.251801	False	umbrella	0.115123	False
2022	881268444196462592	https://pbs.twimg.com/media/DDrk-f9WAAI-WQv.jpg	1	tusker	0.473303	False	Indian_elephant	0.245646	False
2046	886680336477933568	https://pbs.twimg.com/media/DE4fEDzWAAAyHMM.jpg	1	convertible	0.738995	False	sports_car	0.139952	False
2052	887517139158093824	https://pbs.twimg.com/ext_tw_video_thumb/88751...	1	limousine	0.130432	False	tow_truck	0.029175	False
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	1	orange	0.097049	False	bagel	0.085851	False

324 rows × 12 columns

In [17]:
```
image_predictions.describe()
```

Out[17]:

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

In [18]:
```
image_predictions.info()
```

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

In [19]:
```
sum(image_predictions.tweet_id.duplicated())
```

Out[19]: 0

In [20]:
```
twitter_archive
```

Out[20]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweet
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	NaN	

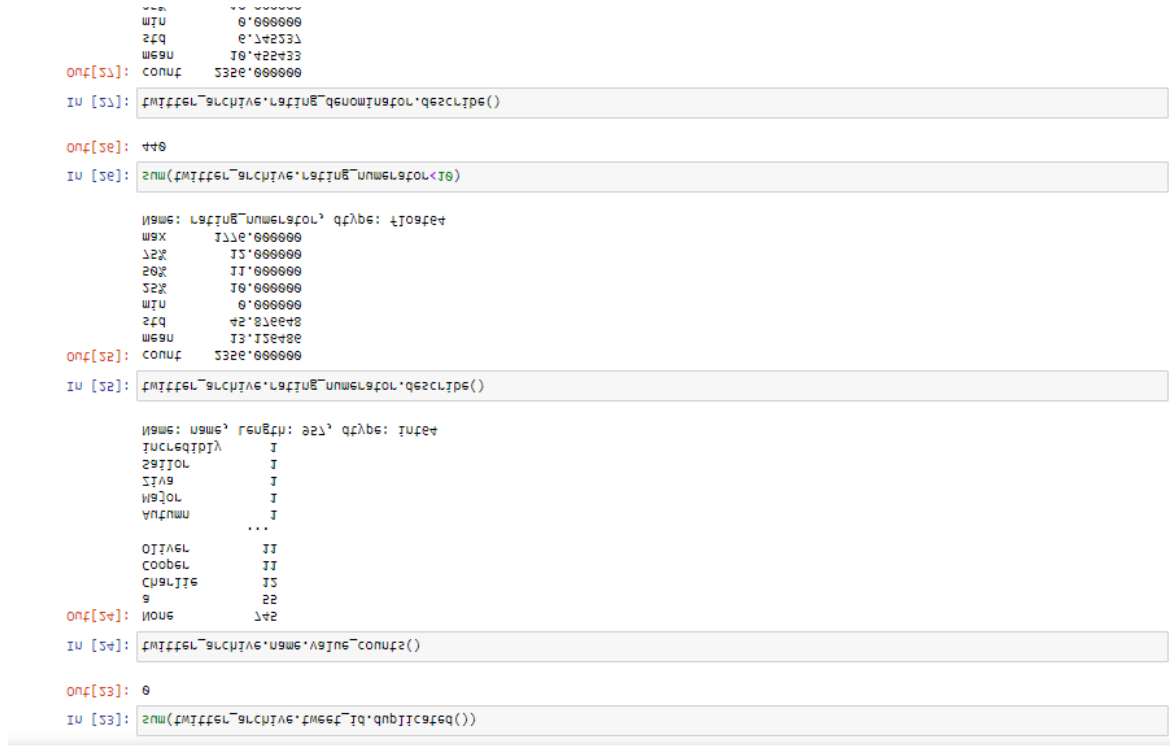
In [21]:
```
twitter_archive.describe()
```

Out[21]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	retweeted_status_id	retweeted_status_user_id	rating_numerator	rating_denominator
count	2.356000e+03	7.800000e+01	7.800000e+01	1.810000e+02	1.810000e+02	2356.000000	2356.000000
mean	7.427716e+17	7.455079e+17	2.014171e+16	7.720400e+17	1.241698e+16	13.126486	10.455433
std	6.856705e+16	7.582492e+16	1.252797e+17	6.236928e+16	9.599254e+16	45.876648	6.745237
min	6.660209e+17	6.658147e+17	1.185634e+07	6.661041e+17	7.832140e+05	0.000000	0.000000
25%	6.783989e+17	6.757419e+17	3.086374e+08	7.186315e+17	4.196984e+09	10.000000	10.000000
50%	7.196279e+17	7.038708e+17	4.196984e+09	7.804657e+17	4.196984e+09	11.000000	10.000000
75%	7.993373e+17	8.257804e+17	4.196984e+09	8.203146e+17	4.196984e+09	12.000000	10.000000
max	8.924206e+17	8.862664e+17	8.405479e+17	8.874740e+17	7.874618e+17	1776.000000	170.000000

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

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



(4) Data cleaning:

First take a copy from all datasets and start cleaning **Tidiness:**

(1)collect all dogs stage in one column

Tidiness:

(1)collect all dogs stage in one column

In [31]:
```
twitter_archive_clean=pd.melt(twitter_archive_clean,id_vars=['tweet_id','in_reply_to_status_id','in_reply_to_user_id','timestamp'
                                                             'retweeted_status_id','retweeted_status_user_id','retweeted_status_timestamp','e
                                                             'rating_numerator','rating_denominator','name'],var_name='dog_stage',value_name=
twitter_archive_clean = twitter_archive_clean.drop('dog_stage', 1)
twitter_archive_clean = twitter_archive_clean.sort_values('dogs_stage').drop_duplicates(subset='tweet_id', keep='last')
```

Test

In [32]:
```
twitter_archive_clean
```

Out[32]:

id	retweeted_status_user_id	retweeted_status_timestamp	expanded_urls	rating_numerator	rating_denominator	name	dogs_stage
17	4.296832e+09	2015-11-20 03:43:06 +0000	https://twitter.com/dogratingrating/status/667...	12	10	None	None
17	4.296832e+09	2015-11-20 03:41:59 +0000	https://twitter.com/dogratingrating/status/667...	5	10	None	None

(2)collect all data in one dataset

(2)collect all data in one dataset

In [37]:
```
twitter_archive_clean=pd.merge(twitter_archive_clean,image_predictions_clean,on='tweet_id',how='left')
twitter_archive_clean=pd.merge(twitter_archive_clean,twitter_data_clean,on='tweet_id',how='left')
```

Test

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2356 entries, 0 to 2355
Data columns (total 27 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   tweet_id                    2356 non-null   int64
 1   in_reply_to_status_id       78 non-null     float64
 2   in_reply_to_user_id         78 non-null     float64
 3   timestamp                   2356 non-null   object
 4   source                      2356 non-null   object
 5   text                        2356 non-null   object
 6   retweeted_status_id         181 non-null    float64
 7   retweeted_status_user_id    181 non-null    float64
 8   retweeted_status_timestamp  181 non-null    object
 9   expanded_urls               2297 non-null   object
 10  rating_numerator            2356 non-null   int64
 11  rating_denominator          2356 non-null   int64
 12  name                        2356 non-null   object
 13  dogs_stage                  2356 non-null   object
 14  jpg_url                     2075 non-null   object
 15  img_num                     2075 non-null   float64
 16  p1                          2075 non-null   object
 17  p1_conf                     2075 non-null   float64
 18  p1_dog                      2075 non-null   object
 19  p2                          2075 non-null   object
 20  p2_conf                     2075 non-null   float64
 21  p2_dog                      2075 non-null   object
 22  p3                          2075 non-null   object
 23  p3_conf                     2075 non-null   float64
 24  p3_dog                      2075 non-null   object
 25  retweet_count               2354 non-null   float64
 26  favorite_count              2354 non-null   float64
dtypes: float64(10), int64(3), object(14)
memory usage: 515.4+ KB
```

Quality:

#(1)invaild timestamp datatype (str)instead of(datetime)

#(2)invaild tweet_id datatype (int)instead of(str)

#(3)invalid p1_dog,p2_dog,p3_dog (str)instead of(bool)

#(4)invalid rating_numerator,rating_denominator (int)instead of(float)

#(5)drop unimportant cols and have many missing values

#(6)remove rows with missing images

#(7)correct rating_denominator must be all values =10

#(8)correct rating_numerator must be all values <10

#(9)some names upper and some lower ..make them all upper

#(10)collect all dogs_type in one column

#(11)collect the best confs in one column

#(12)remove the noisy data from names,dogs_stage

(1)

(1)invaild timestamp datatype (str)instead of(datetime) & remove mark(+)

In [41]:
```
twitter_archive_clean['timestamp']
```

Out[41]:
```
0       2015-11-20 03:51:52 +0000
1       2015-11-20 03:51:47 +0000
2       2015-11-20 03:44:31 +0000
3       2015-11-20 03:35:20 +0000
4       2015-11-20 03:25:43 +0000
                   ...
2351    2017-01-29 02:44:34 +0000
2352    2017-01-27 23:16:13 +0000
2353    2017-01-27 17:04:02 +0000
2354    2016-11-30 01:31:12 +0000
2355    2017-05-03 03:17:27 +0000
Name: timestamp, Length: 2356, dtype: object
```

In [42]:
```
twitter_archive_clean['time']=twitter_archive_clean.timestamp.str.split('+',1)
```

In [43]:
```
twitter_archive_clean['time']
```

Out[43]:
```
0       [2015-11-20 03:51:52 , 0000]
1       [2015-11-20 03:51:47 , 0000]
2       [2015-11-20 03:44:31 , 0000]
3       [2015-11-20 03:35:20 , 0000]
4       [2015-11-20 03:25:43 , 0000]
                    ...
2351    [2017-01-29 02:44:34 , 0000]
2352    [2017-01-27 23:16:13 , 0000]
2353    [2017-01-27 17:04:02 , 0000]
2354    [2016-11-30 01:31:12 , 0000]
2355    [2017-05-03 03:17:27 , 0000]
Name: time, Length: 2356, dtype: object
```

In [44]:
```
for i in range(2355):
    twitter_archive_clean['time&date']=twitter_archive_clean['time'][i][0]
```

In [45]:
```
twitter_archive_clean['time&date']
```

Out[45]:
```
0       2016-11-30 01:31:12
1       2016-11-30 01:31:12
2       2016-11-30 01:31:12
3       2016-11-30 01:31:12
4       2016-11-30 01:31:12
```

Test

In [46]:
```
twitter_archive_clean['time&date'] = pd.to_datetime(twitter_archive_clean['time&date'])
twitter_archive_clean['time&date']
```

Out[46]:
```
0      2016-11-30 01:31:12
1      2016-11-30 01:31:12
2      2016-11-30 01:31:12
3      2016-11-30 01:31:12
4      2016-11-30 01:31:12
               ...
2351   2016-11-30 01:31:12
2352   2016-11-30 01:31:12
2353   2016-11-30 01:31:12
2354   2016-11-30 01:31:12
2355   2016-11-30 01:31:12
Name: time&date, Length: 2356, dtype: datetime64[ns]
```

In [47]:
```
twitter_archive_clean=twitter_archive_clean.drop('timestamp',axis=1)
twitter_archive_clean=twitter_archive_clean.drop('time',axis=1)
```

In [48]:
```
twitter_archive_clean.head()
```

Out[48]:

expanded_urls	rating_numerator	...	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog	retweet_count	favorite_count	time&date
ingrating/status/667...	12	...	False	vizsla	0.000081	True	collie	0.000069	True	37.0	0.0	2016-11-30 01:31:12
ingrating/status/667...	5	...	False	dishwasher	0.000201	False	oscilloscope	0.000142	False	34.0	0.0	2016-11-30 01:31:12
s/status/667549055...	1	...	False	spotlight	0.007737	False	lampshade	0.001901	False	2454.0	6138.0	2016-11-30 01:31:12
s/status/667546741...	9	...	True	miniature_poodle	0.202225	True	teddy	0.004047	False	138.0	355.0	2016-11-30 01:31:12
s/status/667544320...	10	...	True	Pembroke	0.312958	True	Chihuahua	0.071960	True	568.0	917.0	2016-11-30

(2,3,4,5)

(2)invaild tweet_id datatype (int)instead of(str)

(3)invalid p1_dog,p2_dog,p3_dog (str)instead of(bool)

(4)invalid rating_numerator,rating_denominator (int)instead of(float)

In [50]:
```
twitter_archive_clean.tweet_id=twitter_archive_clean.tweet_id.astype(str)
twitter_archive_clean.rating_numerator = twitter_archive_clean.rating_numerator.astype(float)
twitter_archive_clean.rating_denominator=twitter_archive_clean.rating_denominator.astype(float)
twitter_archive_clean.p1_dog=twitter_archive_clean.p1_dog.astype(bool)
twitter_archive_clean.p2_dog=twitter_archive_clean.p2_dog.astype(bool)
twitter_archive_clean.p3_dog=twitter_archive_clean.p3_dog.astype(bool)
```

Test ¶

In [51]:
```
twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2356 entries, 0 to 2355
Data columns (total 27 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   tweet_id                    2356 non-null   object 
 1   in_reply_to_status_id       78 non-null     float64
 2   in_reply_to_user_id         78 non-null     float64
 3   source                      2356 non-null   object 
 4   text                        2356 non-null   object 
 5   retweeted_status_id         181 non-null    float64
 6   retweeted_status_user_id    181 non-null    float64
 7   retweeted_status_timestamp  181 non-null    object 
 8   expanded_urls               2297 non-null   object 
 9   rating_numerator            2356 non-null   float64
 10  rating_denominator          2356 non-null   float64
 11  name                        2356 non-null   object 
 12  dogs_stage                  2356 non-null   object 
 13  jpg_url                     2075 non-null   object 
 14  img_num                     2075 non-null   float64
 15  p1                          2075 non-null   object 
 16  p1_conf                     2075 non-null   float64
 17  p1_dog                      2356 non-null   bool
```

(5)drop unimportant cols and have many missing values

In [52]:
```
twitter_archive_clean=twitter_archive_clean.drop('in_reply_to_status_id',axis=1)
twitter_archive_clean=twitter_archive_clean.drop('in_reply_to_user_id',axis=1)
twitter_archive_clean=twitter_archive_clean.drop('retweeted_status_id',axis=1)
twitter_archive_clean=twitter_archive_clean.drop('retweeted_status_user_id',axis=1)
twitter_archive_clean=twitter_archive_clean.drop('retweeted_status_timestamp',axis=1)
```

Test

In [53]:
```
twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2356 entries, 0 to 2355
Data columns (total 22 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   tweet_id            2356 non-null   object 
 1   source              2356 non-null   object 
 2   text                2356 non-null   object 
 3   expanded_urls       2297 non-null   object 
 4   rating_numerator    2356 non-null   float64
 5   rating_denominator  2356 non-null   float64
 6   name                2356 non-null   object 
 7   dogs_stage          2356 non-null   object 
 8   jpg_url             2075 non-null   object 
 9   img_num             2075 non-null   float64
 10  p1                  2075 non-null   object 
 11  p1_conf             2075 non-null   float64
 12  p1_dog              2356 non-null   bool
 13  p2                  2075 non-null   object 
 14  p2_conf             2075 non-null   float64
 15  p2_dog              2356 non-null   bool
 16  p3                  2075 non-null   object 
 17  p3_conf             2075 non-null   float64
 18  p3_dog              2356 non-null   bool
 19  retweet_count       2354 non-null   float64
 20  favorite_count      2354 non-null   float64
 21  time&date           2356 non-null   datetime64[ns]
dtypes: bool(3), datetime64[ns](1), float64(8), object(10)
memory usage: 455.0+ KB
```

(7,8,9,10,11)

(7)correct rating_denominator must be all values =10

```
In [59]: k=list(twitter_archive_clean.rating_denominator)
         for i in range (2075):
             twitter_archive_clean.rating_denominator=twitter_archive_clean.rating_denominator.replace(k[i],10)
```

Test

```
In [60]: twitter_archive_clean.rating_denominator.value_counts()

Out[60]: 10.0    2075
         Name: rating_denominator, dtype: int64
```

```
In [61]: twitter_archive_clean.rating_denominator.describe()

Out[61]: count    2075.0
         mean       10.0
         std         0.0
         min        10.0
         25%        10.0
         50%        10.0
         75%        10.0
         max        10.0
         Name: rating_denominator, dtype: float64
```

```
In [62]: twitter_archive_clean.rating_numerator.value_counts()

Out[62]: 12.0     474
         10.0     429
         11.0     413
         13.0     284
         9.0      151
         8.0       95
         7.0       52
         14.0      40
         5.0       34
         6.0       32
         3.0       19
         4.0       16
         2.0        9
         1.0        5
```

(8)correct rating_numerator must be all values <10

```
In [65]: l=list(twitter_archive_clean.rating_numerator)
         for i in range(2075):
             if(l[i]<10):
                 twitter_archive_clean.rating_numerator=twitter_archive_clean.rating_numerator.replace(l[i],l[i]+10)
```

Test

```
In [66]: twitter_archive_clean.rating_numerator.value_counts()

Out[66]: 12.0     483
         10.0     431
         11.0     418
         13.0     303
         19.0     151
         18.0      95
         14.0      56
         17.0      52
         15.0      35
         16.0      32
         80.0       1
         44.0       1
         60.0       1
         50.0       1
         143.0      1
         75.0       1
         144.0      1
         88.0       1
         24.0       1
         84.0       1
         27.0       1
         121.0      1
         1776.0     1
         204.0      1
         420.0      1
         45.0       1
         165.0      1
         99.0       1
         26.0       1
         Name: rating_numerator, dtype: int64
```

(9)some names upper and some lower ..make them all upper

In [72]:
```
twitter_archive_clean.p1=twitter_archive_clean.p1.str.title()
twitter_archive_clean.p2=twitter_archive_clean.p2.str.title()
twitter_archive_clean.p3=twitter_archive_clean.p3.str.title()
```

Test

In [73]:
```
twitter_archive_clean
```

Out[73]:

	jpg_url	img_num	...	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog	retweet_count	favorite_count	time&date	rating
	/CUOb_gUUkAACXdS.jpg	1.0	...	Vizsla	0.000081	True	Collie	0.000069	True	37.0	0.0	2016-11-30 01:31:12	12.0
	ı/CUObvUJVEAAnYPF.jpg	1.0	...	Dishwasher	0.000201	False	Oscilloscope	0.000142	False	34.0	0.0	2016-11-30 01:31:12	15.0
	CUOcVCwWsAERUKY.jpg	1.0	...	Spotlight	0.007737	False	Lampshade	0.001901	False	2454.0	6138.0	2016-11-30 01:31:12	11.0
	'CUOaOWXWcAA0_Jy.jpg	1.0	...	Miniature_Poodle	0.202225	True	Teddy	0.004047	False	138.0	355.0	2016-11-30 01:31:12	19.0
	CUOYBbbWIAAXQGU.jpg	1.0	...	Pembroke	0.312958	True	Chihuahua	0.071960	True	568.0	917.0	2016-11-30 01:31:12	10.0

In [76]:
```
#### (10)collect all dogs_type in one column
#### (11)collect the best confs in one column

#create 2 new list to save the best values in them
dog_type=[]
conf=[]
#make lists of 3 p1 coulmns to be easier to use their values
q=list(twitter_archive_clean.p1_dog)
w=list(twitter_archive_clean.p1)
z=list(twitter_archive_clean.p1_conf)
#make lists of 3 p2 coulmns to be easier to use their values
e=list(twitter_archive_clean.p2_dog)
r=list(twitter_archive_clean.p2)
t=list(twitter_archive_clean.p2_conf)
#make lists of 3 p3 coulmns to be easier to use their values
y=list(twitter_archive_clean.p3_dog)
u=list(twitter_archive_clean.p3)
p=list(twitter_archive_clean.p3_conf)
#as we know the p1 is the better algorithm the p2 then p3
#so when p1 is true ,then this is a type of dog and we will save its value in list dog_type and its conf in list conf
#if it false we will do to the second ALG p2 and if it true we will save its value in list dog_type its conf in list conf
#if it false we will do to the third ALG p2 and if it true we will save its value in list dog_type its conf in list conf
#if false we will loop again untill finishing the range(1751)
for i in range(1751):
    if(q[i]==True):
        dog_type.append(w[i])
        conf.append(z[i])
    elif(e[i]==True):
        dog_type.append(r[i])
        conf.append(t[i])
    elif(y[i]==True):
        dog_type.append(u[i])
        conf.append(p[i])
#in the end we will create columns dogs_type ,conf to save the values which in lists
twitter_archive_clean['dogs_type']=dog_type
twitter_archive_clean['conf']=conf
twitter_archive_clean
```

drop unimportant columns after creating dogs_type and conf

In [77]:
```
twitter_archive_clean=twitter_archive_clean.drop(['p1','p1_conf','p1_dog','p2','p2_conf','p2_dog','p3','p3_conf','p3_dog'],1)
```

Test

In [78]:
```
twitter_archive_clean.name.value_counts()
```

Out[78]:
```
None            439
a                46
Tucker           10
Cooper           10
Lucy             10
               ...
Brandy            1
Tiger             1
Gunner            1
Alexanderson      1
Hermione          1
Name: name, Length: 852, dtype: int64
```

(12),(13)

(12)remove the noisy data from names

In [79]:
```
name_index=twitter_archive_clean.loc[(twitter_archive_clean.name=='None')].index
twitter_archive_clean=twitter_archive_clean.drop(name_index)
```

Test

In [80]:
```
twitter_archive_clean.name.value_counts()
```

Out[80]:
```
a               46
Cooper          10
Lucy            10
Tucker          10
Charlie         10
                ..
Brandy           1
Tiger            1
Gunner           1
Alexanderson     1
Hermione         1
Name: name, Length: 851, dtype: int64
```

(13)remove the noisy data from dgos_stage

In [81]:
```
dogs_index=twitter_archive_clean.loc[(twitter_archive_clean.dogs_stage=='None')].index
twitter_archive_clean=twitter_archive_clean.drop(dogs_index)
```

Test

In [82]:
```
twitter_archive_clean.dogs_stage.value_counts()
```

Out[82]:
```
pupper     109
doggo       36
puppo       16
floofer      5
Name: dogs_stage, dtype: int64
```