Analyzing the tweet archive of Twitter user @dog_rates, also known as WeRateDogs

Introduction:

The dataset is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that makes a humerous comment about a dog and provides a rating. The goal here is to wrangle the data and extract interesting insight from the data.

Goals:

- Gather the data
- Assess the dirtiness and messiness issues of data
- Clean the data
- Insight and visualization

In [257]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import requests
import io
data=pd.read_csv('twitter-archive-enhanced.csv')
images_raw=requests.get('https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv')

%matplotlib inline
```

Gathering

In [258]:
```
images = pd.read_csv(io.StringIO(images_raw.content.decode('utf-8')),sep='\t')
#images.text
```

In [259]:
```
images.head()
```

Out[259]:

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

In [260]:
```
text_file_path = 'tweet-json_copy.txt'
df_tweet_likes = pd.read_json(text_file_path, lines = True)
```

In [261]:
```
columns_of_interest=['id', 'retweet_count', 'favorite_count']
df_new=df_tweet_likes[columns_of_interest]
df_new.rename(columns={'id':'tweet_id'}, inplace=True)
```

In [262]:
```
df_new.describe()
```

Out[262]:

	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

In [263]:
```
data.head()
```

Out[263]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	href="http://twitter.co
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://twitter.co
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	href="http://twitter.co
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	href="http://twitter.co
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	href="http://twitter.co

Assessing

Check for any misspelling in the dog stages

In [264]:
```
data.doggo.value_counts()
```

Out[264]:
```
None     2259
doggo      97
Name: doggo, dtype: int64
```

In [265]:
```
data.floofer.value_counts()
```

Out[265]:
```
None       2346
floofer      10
Name: floofer, dtype: int64
```

In [266]:
```
data.pupper.value_counts()
```

Out[266]:
```
None      2099
pupper     257
Name: pupper, dtype: int64
```

In [267]:
```
data.puppo.value_counts()
```

Out[267]:
```
None     2326
puppo      30
Name: puppo, dtype: int64
```

In [268]:
```
sum(data.timestamp.isnull())
```

Out[268]:
```
0
```

In [269]:
```
data.describe()
```

Out[269]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	retweeted_status_id	retweeted_st
count	2.356000e+03	7.800000e+01	7.800000e+01	1.810000e+02	1
mean	7.427716e+17	7.455079e+17	2.014171e+16	7.720400e+17	1
std	6.856705e+16	7.582492e+16	1.252797e+17	6.236928e+16	9
min	6.660209e+17	6.658147e+17	1.185634e+07	6.661041e+17	7
25%	6.783989e+17	6.757419e+17	3.086374e+08	7.186315e+17	4
50%	7.196279e+17	7.038708e+17	4.196984e+09	7.804657e+17	4
75%	7.993373e+17	8.257804e+17	4.196984e+09	8.203146e+17	4
max	8.924206e+17	8.862664e+17	8.405479e+17	8.874740e+17	7

In [270]:
```
data.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 [271]:
```
len(data[data['rating_numerator']>0])*100/len(data)
```

Out[271]:
```
99.9151103565365
```

In [272]:
```
sum(images.p1.isnull())
```

Out[272]:
```
0
```

In [273]:
```
data.rating_numerator.sort_values().head(10)
```

Out[273]:
```
315     0
1016    0
2335    1
2261    1
2338    1
605     1
1446    1
1869    1
2091    1
2038    1
Name: rating_numerator, dtype: int64
```

In [274]:
```
data.rating_numerator.sort_values().tail(40)
```

Out[274]:
```
199       14
101       14
214       14
924       14
1053      14
209       14
369       14
395       14
78        14
76        14
866       14
83        14
291       15
285       15
55        17
1663      20
516       24
1712      26
763       27
1433      44
1274      45
1202      50
1351      60
340       75
695       75
1254      80
433       84
1843      88
1228      99
1635     121
1634     143
1779     144
902      165
290      182
1120     204
2074     420
188      420
189      666
313      960
979     1776
Name: rating_numerator, dtype: int64
```

In [275]:
```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 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: 55.3 KB
```

In [276]:
```
sum(data.duplicated())  # indicates no duplicates
```

Out[276]: 0

In [277]:
```
sum(data.tweet_id.duplicated())
```

Out[277]: 0

Tidiness issues:

- Duggo,floofer,pupper, and puppo are values of one varible dog-stage and need to be categorical.
- The three tables need to join as the breed and retweet and favorite counts belong to tweets.

Quality issues:

- Missing dog_stages values
- We only want original rating that is no retweets
- The numerator ratings 0 and more than 15 are not valid as most of the values fall between 1 and 15.
- 99% of denominator is equal to 10. A small percetage is not equal to 10.
- Some columns do not provide useful information for analyses, so remove them.
- One column should only represent as rating.
- Type of dog_stage is not categorical while there are only a few categories.
- Two categories of dog-stages in some tweets.
- Time-stamp of tweets needs to be converted to datetime
- Change tweet_id to an object datatype
- Column names in the images table is descriptive

Cleaning:

Tidiness:

- Correct dog staging using
- Merge three tables using left join.

Define

Correct dog staging using pd.melt that is to melt dog stages into 1 column: 'dog_stage'

Code

In [278]:
```
data_clean=data.copy()
```

In [279]:
```
data_clean['dog_stage']=data_clean.doggo.replace('None','')+data_clean.floofer
.replace('None','')+data_clean.pupper.replace('None','')+data_clean.puppo.repl
ace('None','')
data_clean=data_clean.drop(['doggo','floofer','pupper','puppo'],axis=1)
```

Test

In [280]:
```
data_clean['dog_stage'].value_counts()
```

Out[280]:
```
                1976
pupper           245
doggo             83
puppo             29
doggopupper       12
floofer            9
doggopuppo         1
doggofloofer       1
Name: dog_stage, dtype: int64
```

In [281]:
```
data_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
dog_stage                     2356 non-null object
dtypes: float64(4), int64(3), object(7)
memory usage: 257.8+ KB
```

Define

We only want original rating that is no retweets

Code

In [282]:
```
data_clean = data_clean[pd.isnull(data_clean.retweeted_status_id)]
```

In [283]:
```
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 14 columns):
tweet_id                      2175 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2175 non-null object
source                        2175 non-null object
text                          2175 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null object
expanded_urls                 2117 non-null object
rating_numerator              2175 non-null int64
rating_denominator            2175 non-null int64
name                          2175 non-null object
dog_stage                     2175 non-null object
dtypes: float64(4), int64(3), object(7)
memory usage: 254.9+ KB
```

Define

Define merge three tables using pd.merge

Code

In [284]:
```
# check duplicated columns
```

In [285]:
```
all_columns = pd.Series(list(data) + list(df_new))
all_columns[all_columns.duplicated()]
```

Out[285]:
```
17    tweet_id
dtype: object
```

In [286]:
```
all_columns = pd.Series(list(data) + list(images))
all_columns[all_columns.duplicated()]
```

Out[286]:
```
17    tweet_id
dtype: object
```

tweet_id is the only duplicated column. We use 'tweet_id' to join tables.

In [287]:
```
data_clean = pd.merge(data_clean, images,
                                on=['tweet_id'], how='left')
```

In [288]:
```
data_clean = pd.merge(data_clean, df_new,
                                on=['tweet_id'])
```

Test

data_clean.info()

Quality:

Define

Replace empty cells with NaNs in the dog-stages column.

Code

In [289]:
```
data_clean['dog_stage']=data_clean['dog_stage'].replace('',np.nan)
```

Test

In [290]:
```
data_clean.dog_stage.value_counts()
```

Out[290]:
```
pupper          224
doggo            75
puppo            24
doggopupper      10
floofer           9
doggopuppo        1
doggofloofer      1
Name: dog_stage, dtype: int64
```

In [291]:
```
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2174
Data columns (total 27 columns):
tweet_id                      2175 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2175 non-null object
source                        2175 non-null object
text                          2175 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null object
expanded_urls                 2117 non-null object
rating_numerator              2175 non-null int64
rating_denominator            2175 non-null int64
name                          2175 non-null object
dog_stage                     344 non-null object
jpg_url                       1994 non-null object
img_num                       1994 non-null float64
p1                            1994 non-null object
p1_conf                       1994 non-null float64
p1_dog                        1994 non-null object
p2                            1994 non-null object
p2_conf                       1994 non-null float64
p2_dog                        1994 non-null object
p3                            1994 non-null object
p3_conf                       1994 non-null float64
p3_dog                        1994 non-null object
retweet_count                 2175 non-null int64
favorite_count                2175 non-null int64
dtypes: float64(8), int64(5), object(14)
memory usage: 475.8+ KB
```

Define

Remove out of range values in the 'rating_numerator' to keep only values that are between 1 and 15.

Code

In [292]:
```
mask=(data_clean['rating_numerator']>15) | (data_clean['rating_numerator']==0)
```

In [293]:
```
data_clean.loc[mask,'rating_numerator']=np.nan
```

Test

In [294]:
```
data_clean['rating_numerator'].sort_values()
```

Out[294]:
```
2154    1.0
1267    1.0
1690    1.0
2157    1.0
1761    1.0
       ...
1484    NaN
1533    NaN
1600    NaN
1664    NaN
1895    NaN
Name: rating_numerator, Length: 2175, dtype: float64
```

In [295]:
```
data_clean['rating_numerator'].describe()
```

Out[295]:
```
count    2148.000000
mean       10.615922
std         2.190309
min         1.000000
25%        10.000000
50%        11.000000
75%        12.000000
max        15.000000
Name: rating_numerator, dtype: float64
```

Define

Make all values in the 'rating_denominator' equal to 10.

Code

In [296]:
```
data_clean['rating_denominator']=10
```

Test

In [297]:
```
data_clean['rating_denominator'].unique()
```

Out[297]:
```
array([10], dtype=int64)
```

Define

Remove columns that you do not need for analyses to make the data cleaner (and/or have many missing values). We only keep the first prediction for dog images.

In [298]:
```
data_clean=data_clean.drop(['in_reply_to_status_id','in_reply_to_user_id','ret
weeted_status_id','retweeted_status_user_id','source','retweeted_status_timest
amp', 'expanded_urls','p2', 'p2_conf', 'p2_dog', 'p3', 'p3_conf',
       'p3_dog','jpg_url', 'img_num'],axis=1)
```

Test

In [299]:
```
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2174
Data columns (total 12 columns):
tweet_id              2175 non-null int64
timestamp             2175 non-null object
text                  2175 non-null object
rating_numerator      2148 non-null float64
rating_denominator    2175 non-null int64
name                  2175 non-null object
dog_stage             344 non-null object
p1                    1994 non-null object
p1_conf               1994 non-null float64
p1_dog                1994 non-null object
retweet_count         2175 non-null int64
favorite_count        2175 non-null int64
dtypes: float64(2), int64(4), object(6)
memory usage: 220.9+ KB
```

Define

Present one column as rating by devidng numerator by denominator and remove the two columns representing denominator and numerator.

In [300]:
```
data_clean['rating']=data_clean['rating_numerator']/data_clean['rating_denomin
ator']
data_clean=data_clean.drop(['rating_numerator','rating_denominator'],axis=1)
```

Test

In [301]:
```
data_clean.rating.sort_values()
```

Out[301]:
```
2154    0.1
1267    0.1
1690    0.1
2157    0.1
1761    0.1
       ...
1484    NaN
1533    NaN
1600    NaN
1664    NaN
1895    NaN
Name: rating, Length: 2175, dtype: float64
```

In [302]:
```
data_clean.rating.sort_values()
```

Out[302]:
```
2154    0.1
1267    0.1
1690    0.1
2157    0.1
1761    0.1
       ...
1484    NaN
1533    NaN
1600    NaN
1664    NaN
1895    NaN
Name: rating, Length: 2175, dtype: float64
```

Define

Make the dog-stage (currently string) as a categorical type.

In [303]:
```
# Code
data_clean.dog_stage=data_clean.dog_stage.astype('category')
# Test
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2174
Data columns (total 11 columns):
tweet_id          2175 non-null int64
timestamp         2175 non-null object
text              2175 non-null object
name              2175 non-null object
dog_stage         344 non-null category
p1                1994 non-null object
p1_conf           1994 non-null float64
p1_dog            1994 non-null object
retweet_count     2175 non-null int64
favorite_count    2175 non-null int64
rating            2148 non-null float64
dtypes: category(1), float64(2), int64(3), object(5)
memory usage: 189.4+ KB
```

Define

Manually check and correct the entries with more than one dog-stage defined. We manually check and correct each.

In [304]:
```
df=data_clean[data_clean['dog_stage']=='doggopupper']
for i in range(len(df)):
    print(df.text.iloc[i])
```

```
This is Dido. She's playing the lead role in "Pupper Stops to Catch Snow Befo
re Resuming Shadow Box with Dried Apple." 13/10 (IG: didodoggo) https://t.co/
m7isZrOBX7
Here we have Burke (pupper) and Dexter (doggo). Pupper wants to be exactly li
ke doggo. Both 12/10 would pet at same time https://t.co/ANBpEYHaho
Like doggo, like pupper version 2. Both 11/10 https://t.co/9IxWAXFqze
This is Bones. He's being haunted by another doggo of roughly the same size.
12/10 deep breaths pupper everything's fine https://t.co/55Dqe0SJNj
This is Pinot. He's a sophisticated doggo. You can tell by the hat. Also poin
tier than your average pupper. Still 10/10 would pet cautiously https://t.co/
f2wmLZTPHd
Pupper butt 1, Doggo 0. Both 12/10 https://t.co/WQvcPEpH2u
Meet Maggie &amp;amp; Lila. Maggie is the doggo, Lila is the pupper. They are sis
ters. Both 12/10 would pet at the same time https://t.co/MYwR4DQKll
Please stop sending it pictures that don't even have a doggo or pupper in the
m. Churlish af. 5/10 neat couch tho https://t.co/u2c9c7qSg8
This is just downright precious af. 12/10 for both pupper and doggo https://
t.co/o5J479bZUC
Like father (doggo), like son (pupper). Both 12/10 https://t.co/pG2inLaOda
```

In [305]:
```
mask=data_clean.tweet_id == df.tweet_id.iloc[0]
data_clean.loc[mask,'dog_stage']='pupper'
```

In [306]:
```
mask=data_clean.tweet_id == df.tweet_id.iloc[4]
data_clean.loc[mask,'dog_stage']='doggo'
```

In [307]:
```
mask=data_clean.tweet_id == df.tweet_id.iloc[9]
data_clean.loc[mask,'dog_stage']=np.nan
```

In [308]:
```
data_clean
```

Out[308]:

	tweet_id	timestamp	text	name	dog_stage	p1	
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	Phineas	NaN	orange	0
1	892177421306343426	2017-08-01 00:17:27 +0000	This is Tilly. She's just checking pup on you....	Tilly	NaN	Chihuahua	0
2	891815181378084864	2017-07-31 00:18:03 +0000	This is Archie. He is a rare Norwegian Pouncin...	Archie	NaN	Chihuahua	0
3	891689557279858688	2017-07-30 15:58:51 +0000	This is Darla. She commenced a snooze mid meal...	Darla	NaN	paper_towel	0
4	891327558926688256	2017-07-29 16:00:24 +0000	This is Franklin. He would like you to stop ca...	Franklin	NaN	basset	0
...	...	...	...	...	...	...	
2170	666049248165822465	2015-11-16 00:24:50 +0000	Here we have a 1949 1st generation vulpix. Enj...	None	NaN	miniature_pinscher	0
2171	666044226329800704	2015-11-16 00:04:52 +0000	This is a purebred Piers Morgan. Loves to Netf...	a	NaN	Rhodesian_ridgeback	0
2172	666033412701032449	2015-11-15 23:21:54 +0000	Here is a very happy pup. Big fan of well-main...	a	NaN	German_shepherd	0
2173	666029285002620928	2015-11-15 23:05:30 +0000	This is a western brown Mitsubishi terrier. Up...	a	NaN	redbone	0
2174	666020888022790149	2015-11-15 22:32:08 +0000	Here we have a Japanese Irish Setter. Lost eye...	None	NaN	Welsh_springer_spaniel	0

2175 rows × 11 columns

In [309]:
```
# Remove other rows that represent more than one dog. Since there are not many
of them, we just remove them from further analyses.
indices_to_remove=list(range(len(df)))
exclude=[0,4,9]
indices_to_remove=[ind for ind in indices_to_remove if ind not in exclude]
for ind in indices_to_remove:
    remove_rows=data_clean[data_clean.tweet_id == df.tweet_id.iloc[ind]].index
    data_clean.drop(remove_rows,inplace=True)
```

In [310]:
```
data_clean.head()
```

Out[310]:

	tweet_id	timestamp	text	name	dog_stage	p1	p1_conf	p1_c
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	Phineas	NaN	orange	0.097049	Fa
1	892177421306343426	2017-08-01 00:17:27 +0000	This is Tilly. She's just checking pup on you....	Tilly	NaN	Chihuahua	0.323581	T
2	891815181378084864	2017-07-31 00:18:03 +0000	This is Archie. He is a rare Norwegian Pouncin...	Archie	NaN	Chihuahua	0.716012	T
3	891689557279858688	2017-07-30 15:58:51 +0000	This is Darla. She commenced a snooze mid meal...	Darla	NaN	paper_towel	0.170278	Fa
4	891327558926688256	2017-07-29 16:00:24 +0000	This is Franklin. He would like you to stop ca...	Franklin	NaN	basset	0.555712	T

In [311]:
```
df=data_clean[data_clean['dog_stage']=='doggopupper']
df
```

Out[311]:

	tweet_id	timestamp	text	name	dog_stage	p1	p1_conf	p1_dog	retweet_count	favorite_cou

In [312]:
```
indices_to_remove
```

Out[312]:
```
[1, 2, 3, 5, 6, 7, 8]
```

In [313]:
```
data_clean[data_clean['dog_stage']=='doggopuppo'].text.iloc[0]
```

Out[313]:
```
"Here's a puppo participating in the #ScienceMarch. Cleverly disguising her o
wn doggo agenda. 13/10 would keep the planet habitable for https://t.co/cMhq1
6isel"
```

In [314]:
```
mask=data_clean['dog_stage']=='doggopuppo'
data_clean.loc[mask,'dog_stage']='puppo'
```

In [315]:
```
data_clean[data_clean['dog_stage']=='doggofloofer'].text.iloc[0]
```

Out[315]:
```
"At first I thought this was a shy doggo, but it's actually a Rare Canadian F
loofer Owl. Amateurs would confuse the two. 11/10 only send dogs https://t.c
o/TXdT3tmuYk"
```

In [316]:
```
mask=data_clean['dog_stage']=='doggofloofer'
data_clean.loc[mask,'dog_stage']='floofer'
```

Test

In [317]:
```
data_clean.dog_stage.value_counts()
```

Out[317]:
```
pupper          225
doggo            76
puppo            25
floofer          10
doggopuppo        0
doggopupper       0
doggofloofer      0
Name: dog_stage, dtype: int64
```

Define

Correct the format of timestamp. Remove '+0000' from the end and change it to datetime format

In [318]:
```
# Code
data_clean.timestamp=pd.to_datetime(data_clean.timestamp.str.strip('+0000'))
```

In [319]:
```
# Test
data_clean.timestamp
```

Out[319]:
```
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
               ...
2170   2015-11-16 00:24:50
2171   2015-11-16 00:04:52
2172   2015-11-15 23:21:54
2173   2015-11-15 23:05:30
2174   2015-11-15 22:32:08
Name: timestamp, Length: 2168, dtype: datetime64[ns]
```

In [320]:
```
# Define and code
# change the tweet id type to string using to_string
data_clean.tweet_id=data_clean.tweet_id.to_string()
```

In [325]:
```
# Test
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2168 entries, 0 to 2174
Data columns (total 11 columns):
tweet_id          2168 non-null object
timestamp         2168 non-null datetime64[ns]
text              2168 non-null object
name              2168 non-null object
dog_stage         336 non-null category
prediction        1988 non-null object
confidence        1988 non-null float64
p1_dog            1988 non-null object
retweet_count     2168 non-null int64
favorite_count    2168 non-null int64
rating            2141 non-null float64
dtypes: category(1), datetime64[ns](1), float64(2), int64(2), object(5)
memory usage: 188.8+ KB
```

In [328]:
```
# Define and code
# change the tweet id type to string using to_string
data_clean = data_clean.rename(columns={'p1':'prediction','p1_conf':'confidenc
e','p1_dog':'is_dog'})
```

In [329]:
```
data_clean
```

Out[329]:

	tweet_id	timestamp	text	name	dog_stage	prediction
0	0 892420643555336193\n1 8921774213...	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	Phineas	NaN	orange
1	0 892420643555336193\n1 8921774213...	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	Tilly	NaN	Chihuahua
2	0 892420643555336193\n1 8921774213...	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	Archie	NaN	Chihuahua
3	0 892420643555336193\n1 8921774213...	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	Darla	NaN	paper_towel
4	0 892420643555336193\n1 8921774213...	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	Franklin	NaN	basset
...	...	...	...	...	...	...
2170	0 892420643555336193\n1 8921774213...	2015-11-16 00:24:50	Here we have a 1949 1st generation vulpix. Enj...	None	NaN	miniature_pinscher
2171	0 892420643555336193\n1 8921774213...	2015-11-16 00:04:52	This is a purebred Piers Morgan. Loves to Netf...	a	NaN	Rhodesian_ridgeback
2172	0 892420643555336193\n1 8921774213...	2015-11-15 23:21:54	Here is a very happy pup. Big fan of well-main...	a	NaN	German_shepherd
2173	0 892420643555336193\n1 8921774213...	2015-11-15 23:05:30	This is a western brown Mitsubishi terrier. Up...	a	NaN	redbone
2174	0 892420643555336193\n1 8921774213...	2015-11-15 22:32:08	Here we have a Japanese Irish Setter. Lost eye...	None	NaN	Welsh_springer_spaniel

2168 rows × 11 columns

In [322]:
```
sum(data_clean['p1'].isnull())
```

Out[322]: 180

Note the ratings are not available for all 2345 tweets. Same is true for p1 (dog breed classifier) which has 280 null values.

Saving it to a Master dataframe

In [330]:
```
data_clean.to_csv('twitter_archive_master.csv',index=False)
```

See the Act_report for Insight and visualization