WeRateDogs - Twitter Data

The dataset that I have wrangled (and analyzed and visualized) is the tweet archive of Twitter user **@dog_rates**, also known as **WeRateDogs**. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "**they're good dogs Brent**." WeRateDogs has over 4 million followers and has received international media coverage.

My goal: wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information. Additional gathering, then assessing and cleaning is required for "*Wow!*"-worthy analyses and visualizations.

Gathering Data:

● The Twitter archive data was downloaded from udacity .

● The tweet_json.txt was downloaded from udacity due to some issues faced with the twitter API.

● The image_predictions.tsv was on Udacity's servers was downloaded using the request library . The url was provided as https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_imag e-predictions/image-predictions.tsv

Assessing Data:

Twitter Archive:

QUALITY

- Delete unwanted columns.
- Missing value in columns.
- Numerator having value as 0.
- Denominator having value other than 10.
- Timestamp should be date type.
- Names of dog that are unlikely.
- Float numerator rating values have been incorrectly entered in the column.
- Due to different denominator values comparison of rating cannot be adequate.

Tidiness:

- Dog Stages classification should be 1 column.

Twitter API:

Quality:

- Deleting unwanted columns.
- Null values in columns

Tidiness:

- Merging retweet_count and favorite_count with twitter archive.

Image Prediction:

Quality:

- Duplicate jpg_url present
- Only 2075 entries while in archive 2536 entries.

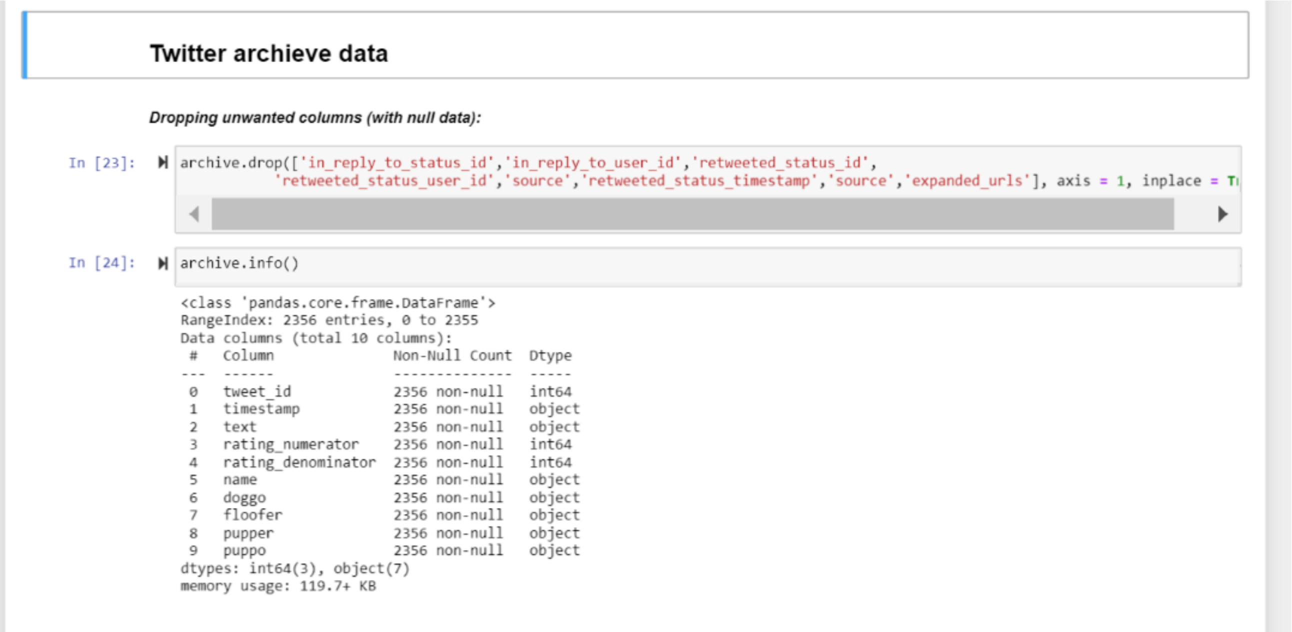
Tidiness:

- Column names should be more descriptive

Cleaning:

1) Twitter achieve data

Dropping unwanted columns with null data:



Twitter archieve data

Dropping unwanted columns (with null data):

In [23]:

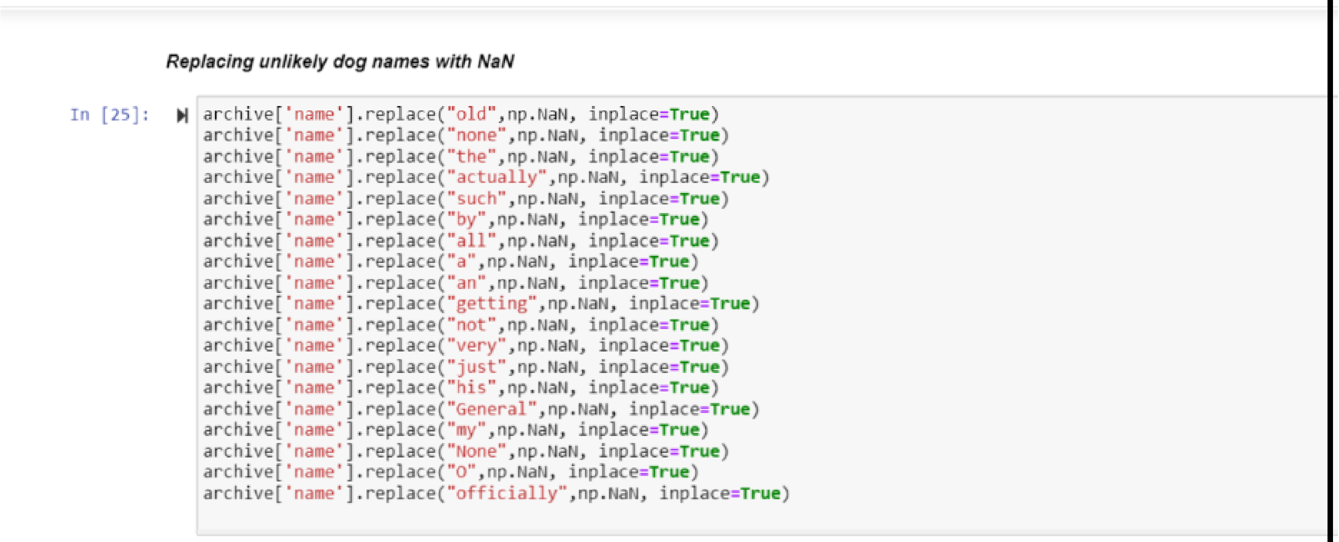
```
archive.drop(['in_reply_to_status_id','in_reply_to_user_id','retweeted_status_id',
          'retweeted_status_user_id','source','retweeted_status_timestamp','source','expanded_urls'], axis = 1, inplace = Tı
```

In [24]:

```
archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   tweet_id            2356 non-null   int64 
 1   timestamp           2356 non-null   object
 2   text                2356 non-null   object
 3   rating_numerator    2356 non-null   int64 
 4   rating_denominator  2356 non-null   int64 
 5   name                2356 non-null   object
 6   doggo               2356 non-null   object
 7   floofer             2356 non-null   object
 8   pupper              2356 non-null   object
 9   puppo               2356 non-null   object
dtypes: int64(3), object(7)
memory usage: 119.7+ KB
```

Replacing unlikely dog names :

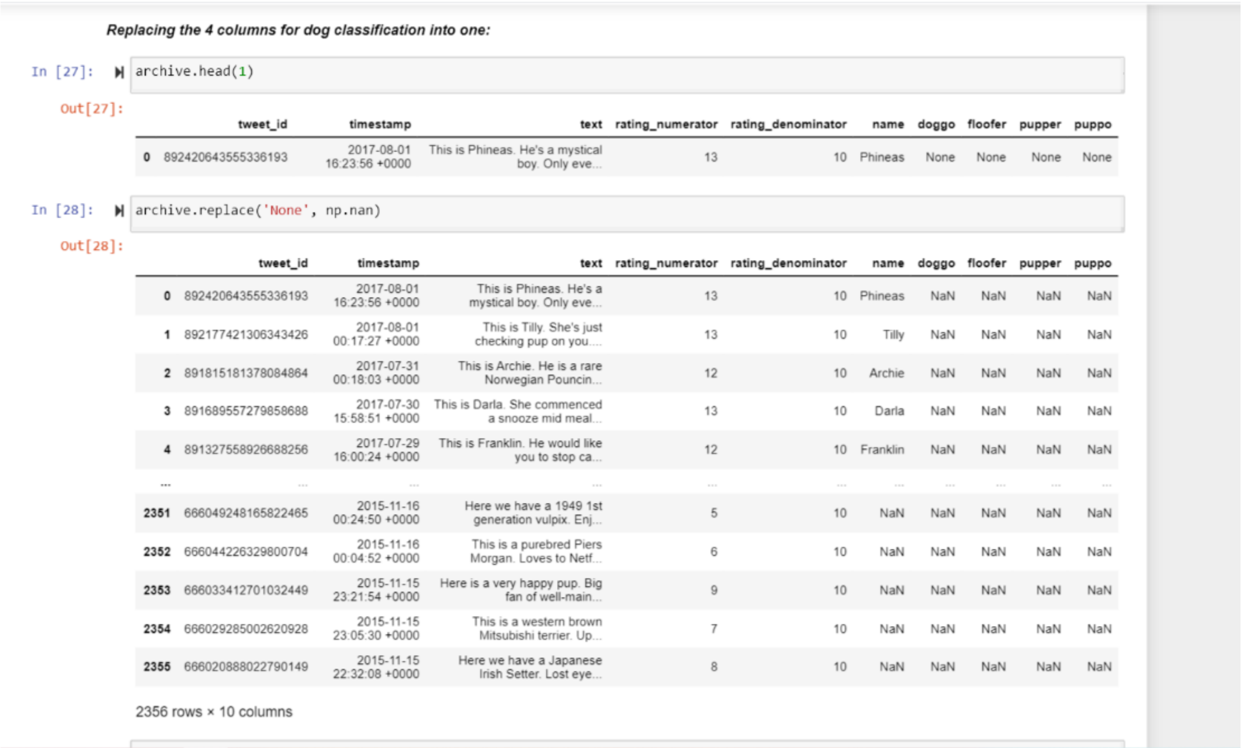


Replacing unlikely dog names with NaN

In [25]:

```
archive['name'].replace("old",np.NaN, inplace=True)
archive['name'].replace("none",np.NaN, inplace=True)
archive['name'].replace("the",np.NaN, inplace=True)
archive['name'].replace("actually",np.NaN, inplace=True)
archive['name'].replace("such",np.NaN, inplace=True)
archive['name'].replace("by",np.NaN, inplace=True)
archive['name'].replace("all",np.NaN, inplace=True)
archive['name'].replace("a",np.NaN, inplace=True)
archive['name'].replace("an",np.NaN, inplace=True)
archive['name'].replace("getting",np.NaN, inplace=True)
archive['name'].replace("not",np.NaN, inplace=True)
archive['name'].replace("very",np.NaN, inplace=True)
archive['name'].replace("just",np.NaN, inplace=True)
archive['name'].replace("his",np.NaN, inplace=True)
archive['name'].replace("General",np.NaN, inplace=True)
archive['name'].replace("my",np.NaN, inplace=True)
archive['name'].replace("None",np.NaN, inplace=True)
archive['name'].replace("O",np.NaN, inplace=True)
archive['name'].replace("officially",np.NaN, inplace=True)
```

Replacing the 4 columns of dog classification into one:



Replacing the 4 columns for dog classification into one:

In [27]:

```
archive.head(1)
```

Out[27]:

	tweet_id	timestamp	text	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	13	10	Phineas	None	None	None	None

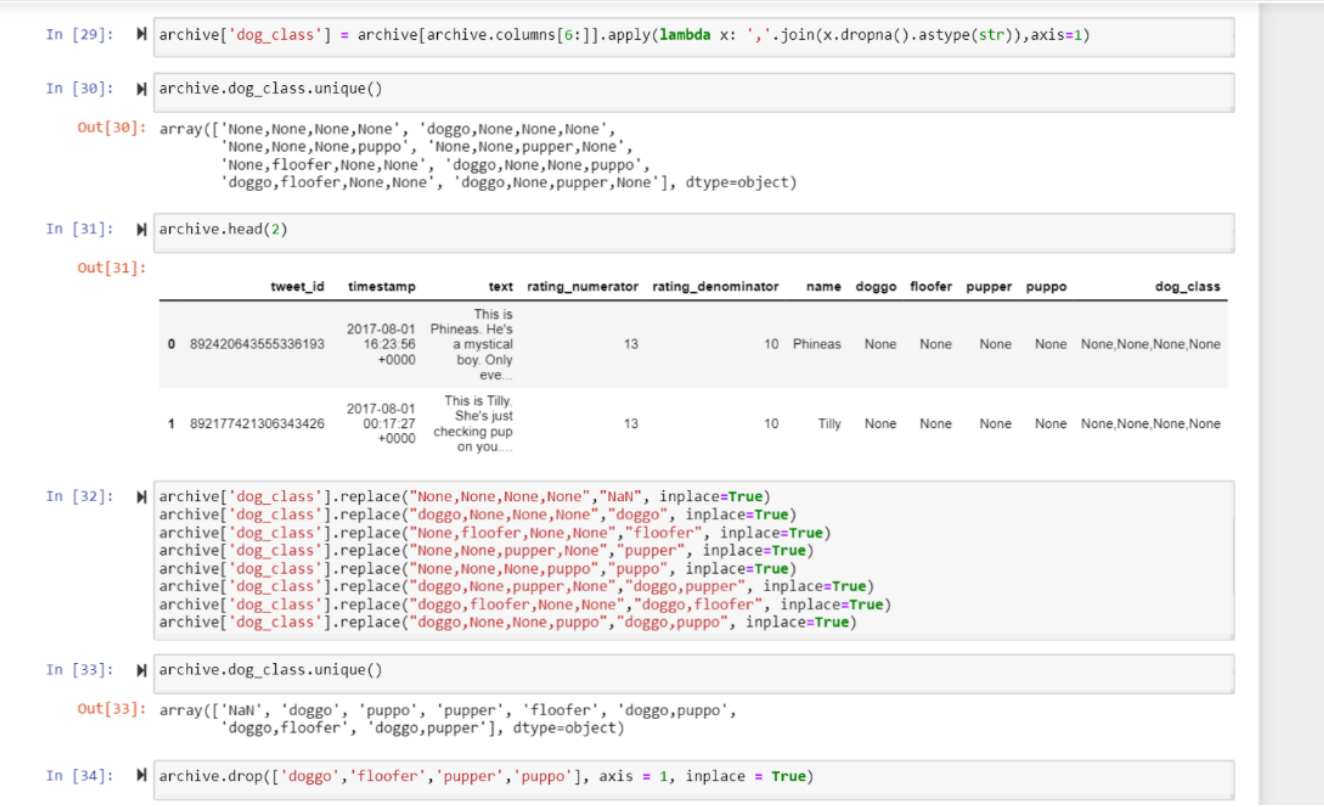
In [28]:

```
archive.replace('None', np.nan)
```

Out[28]:

	tweet_id	timestamp	text	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	13	10	Phineas	NaN	NaN	NaN	NaN
1	892177421306343426	2017-08-01 00:17:27 +0000	This is Tilly. She's just checking pup on you....	13	10	Tilly	NaN	NaN	NaN	NaN
2	891815181378084864	2017-07-31 00:18:03 +0000	This is Archie. He is a rare Norwegian Pouncin...	12	10	Archie	NaN	NaN	NaN	NaN
3	891689557279858688	2017-07-30 15:58:51 +0000	This is Darla. She commenced a snooze mid meal...	13	10	Darla	NaN	NaN	NaN	NaN
4	891327558926688256	2017-07-29 16:00:24 +0000	This is Franklin. He would like you to stop ca...	12	10	Franklin	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
2351	666049248165822465	2015-11-16 00:24:50 +0000	Here we have a 1949 1st generation vulpix. Enj...	5	10	NaN	NaN	NaN	NaN	NaN
2352	666044226329800704	2015-11-16 00:04:52 +0000	This is a purebred Piers Morgan. Loves to Netf...	6	10	NaN	NaN	NaN	NaN	NaN
2353	666033412701032449	2015-11-15 23:21:54 +0000	Here is a very happy pup. Big fan of well-main...	9	10	NaN	NaN	NaN	NaN	NaN
2354	666029285002620928	2015-11-15 23:05:30 +0000	This is a western brown Mitsubishi terrier. Up...	7	10	NaN	NaN	NaN	NaN	NaN
2355	666020888022790149	2015-11-15 22:32:08 +0000	Here we have a Japanese Irish Setter. Lost eye...	8	10	NaN	NaN	NaN	NaN	NaN

2356 rows × 10 columns



```
In [29]:  archive['dog_class'] = archive[archive.columns[6:]].apply(lambda x: ','.join(x.dropna().astype(str)),axis=1)
```

```
In [30]:  archive.dog_class.unique()

Out[30]:  array(['None,None,None,None', 'doggo,None,None,None',
                 'None,None,None,puppo', 'None,None,pupper,None',
                 'None,floofer,None,None', 'doggo,None,None,puppo',
                 'doggo,floofer,None,None', 'doggo,None,pupper,None'], dtype=object)
```

```
In [31]:  archive.head(2)

Out[31]:
```

	tweet_id	timestamp	text	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo	dog_class
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	13	10	Phineas	None	None	None	None	None,None,None,None
1	892177421306343426	2017-08-01 00:17:27 +0000	This is Tilly. She's just checking pup on you....	13	10	Tilly	None	None	None	None	None,None,None,None

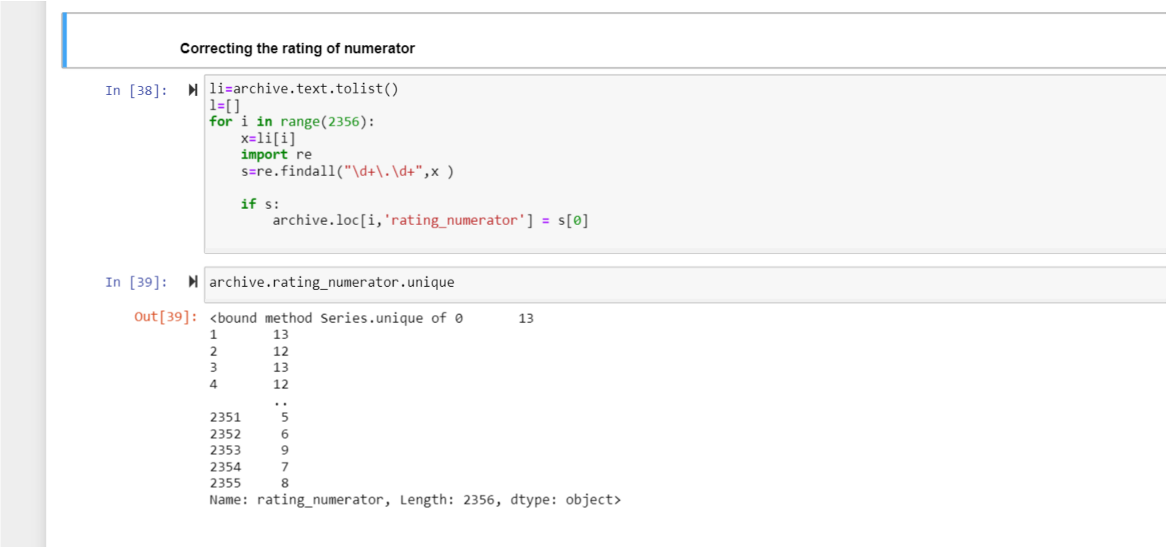
```
In [32]:  archive['dog_class'].replace("None,None,None,None","NaN", inplace=True)
          archive['dog_class'].replace("doggo,None,None,None","doggo", inplace=True)
          archive['dog_class'].replace("None,floofer,None,None","floofer", inplace=True)
          archive['dog_class'].replace("None,None,pupper,None","pupper", inplace=True)
          archive['dog_class'].replace("None,None,None,puppo","puppo", inplace=True)
          archive['dog_class'].replace("doggo,None,pupper,None","doggo,pupper", inplace=True)
          archive['dog_class'].replace("doggo,floofer,None,None","doggo,floofer", inplace=True)
          archive['dog_class'].replace("doggo,None,None,puppo","doggo,puppo", inplace=True)
```

```
In [33]:  archive.dog_class.unique()

Out[33]:  array(['NaN', 'doggo', 'puppo', 'pupper', 'floofer', 'doggo,puppo',
                 'doggo,floofer', 'doggo,pupper'], dtype=object)
```

```
In [34]:  archive.drop(['doggo','floofer','pupper','puppo'], axis = 1, inplace = True)
```

Correcting rating of numerator:



Correcting the rating of numerator

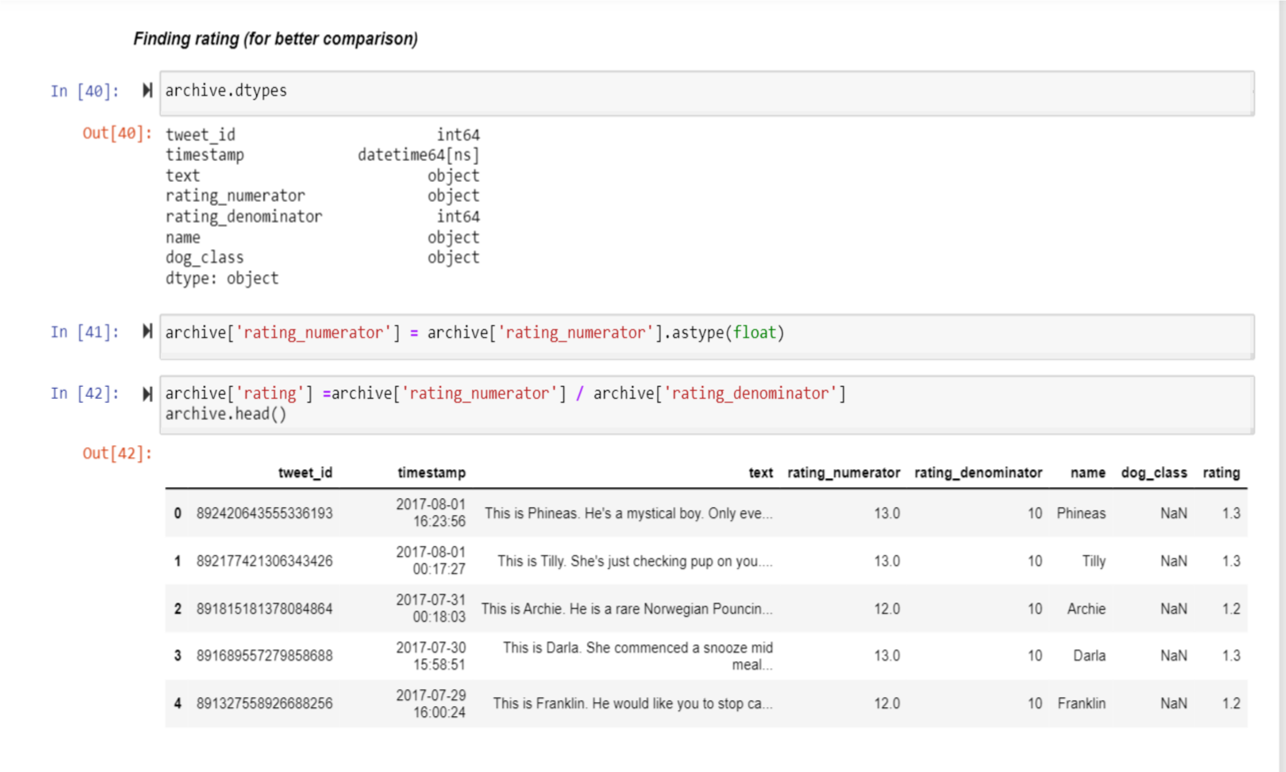
```
In [38]:  li=archive.text.tolist()
          l=[]
          for i in range(2356):
              x=li[i]
              import re
              s=re.findall("\d+\.\d+",x )

              if s:
                  archive.loc[i,'rating_numerator'] = s[0]
```

```
In [39]:  archive.rating_numerator.unique

Out[39]:  <bound method Series.unique of 0       13
          1       13
          2       12
          3       13
          4       12
                  ..
          2351     5
          2352     6
          2353     9
          2354     7
          2355     8
          Name: rating_numerator, Length: 2356, dtype: object>
```

Finding rating for better comparison:



Finding rating (for better comparison)

In [40]:
```
archive.dtypes
```

Out[40]:
```
tweet_id                       int64
timestamp             datetime64[ns]
text                          object
rating_numerator              object
rating_denominator             int64
name                          object
dog_class                     object
dtype: object
```

In [41]:
```
archive['rating_numerator'] = archive['rating_numerator'].astype(float)
```

In [42]:
```
archive['rating'] =archive['rating_numerator'] / archive['rating_denominator']
archive.head()
```

Out[42]:

	tweet_id	timestamp	text	rating_numerator	rating_denominator	name	dog_class	rating
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	13.0	10	Phineas	NaN	1.3
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	13.0	10	Tilly	NaN	1.3
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	12.0	10	Archie	NaN	1.2
3	891689557279858688	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	13.0	10	Darla	NaN	1.3
4	891327558926688256	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	12.0	10	Franklin	NaN	1.2

2)Image Prediction:

Dropping duplicate image url and changing column names:

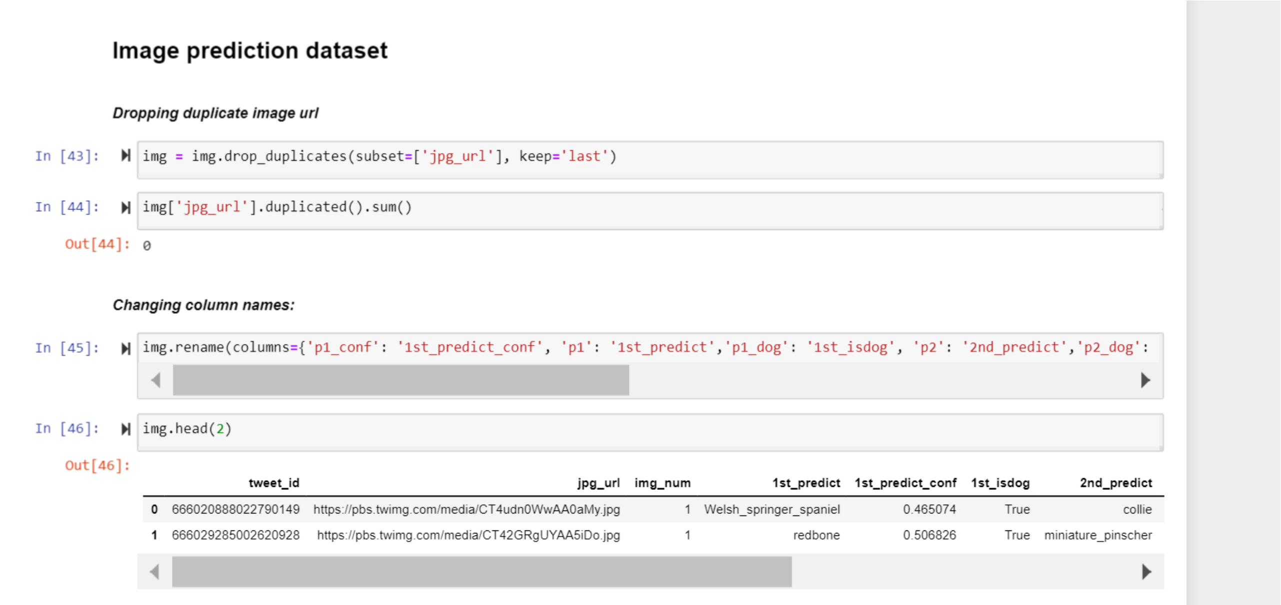


Image prediction dataset

Dropping duplicate image url

In [43]:
```
img = img.drop_duplicates(subset=['jpg_url'], keep='last')
```

In [44]:
```
img['jpg_url'].duplicated().sum()
```

Out[44]:
```
0
```

Changing column names:

In [45]:
```
img.rename(columns={'p1_conf': '1st_predict_conf', 'p1': '1st_predict','p1_dog': '1st_isdog', 'p2': '2nd_predict','p2_dog':
```

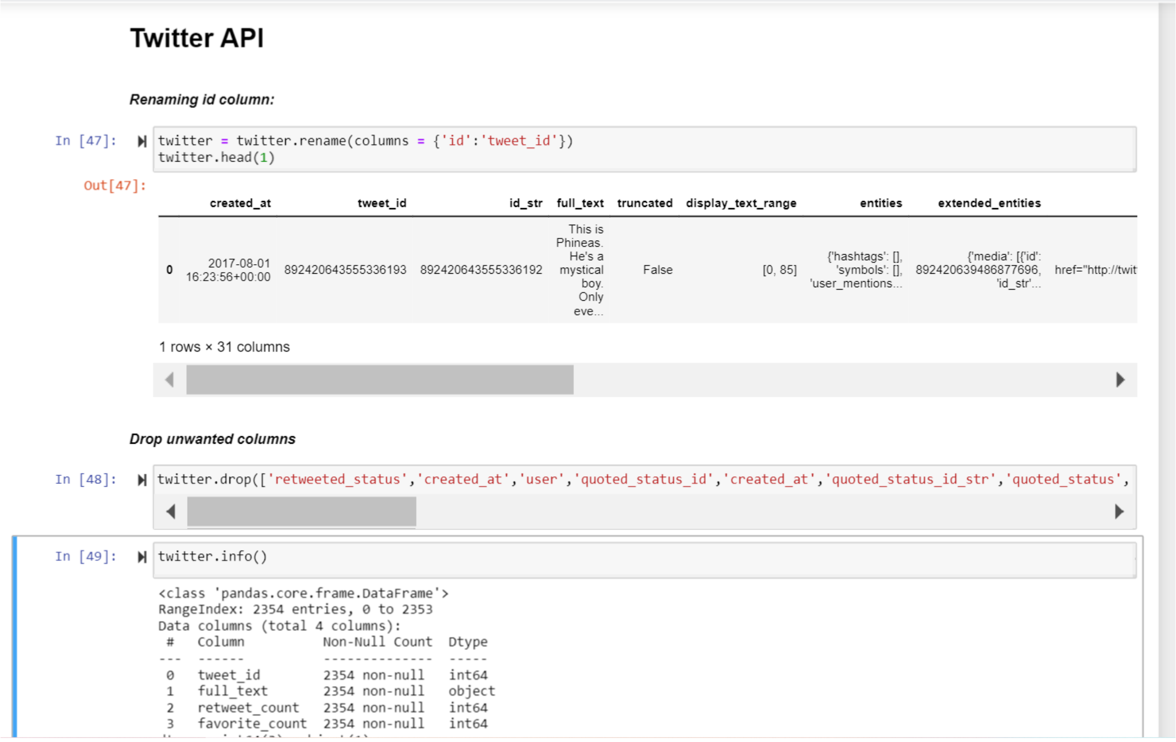
In [46]:
```
img.head(2)
```

Out[46]:

	tweet_id	jpg_url	img_num	1st_predict	1st_predict_conf	1st_isdog	2nd_predict
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True	miniature_pinscher

2) Twitter API

Renaming column and dropping unwanted columns:



Twitter API

Renaming id column:

In [47]:
```
twitter = twitter.rename(columns = {'id':'tweet_id'})
twitter.head(1)
```

Out[47]:

	created_at	tweet_id	id_str	full_text	truncated	display_text_range	entities	extended_entities	
0	2017-08-01 16:23:56+00:00	892420643555336193	892420643555336192	This is Phineas. He's a mystical boy. Only eve...	False	[0, 85]	{'hashtags': [], 'symbols': [], 'user_mentions...	{'media': [{'id': 892420639486877696, 'id_str'...	href="http://twit

1 rows × 31 columns

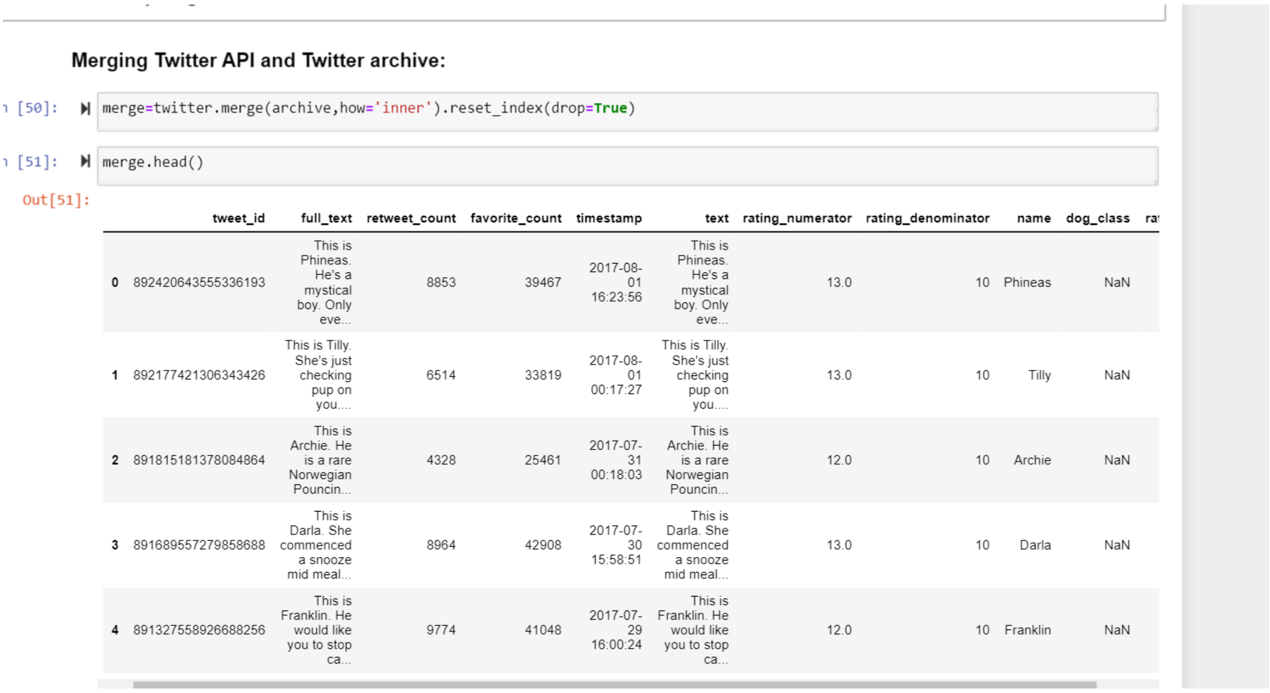
Drop unwanted columns

In [48]:
```
twitter.drop(['retweeted_status','created_at','user','quoted_status_id','created_at','quoted_status_id_str','quoted_status',
```

In [49]:
```
twitter.info()
```

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

Merging twitter API and archieve dataset:



Merging Twitter API and Twitter archive:

In [50]:
```
merge=twitter.merge(archive,how='inner').reset_index(drop=True)
```

In [51]:
```
merge.head()
```

Out[51]:

	tweet_id	full_text	retweet_count	favorite_count	timestamp	text	rating_numerator	rating_denominator	name	dog_class	ra
0	892420643555336193	This is Phineas. He's a mystical boy. Only eve...	8853	39467	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	13.0	10	Phineas	NaN	
1	892177421306343426	This is Tilly. She's just checking pup on you....	6514	33819	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	13.0	10	Tilly	NaN	
2	891815181378084864	This is Archie. He is a rare Norwegian Pouncin...	4328	25461	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	12.0	10	Archie	NaN	
3	891689557279858688	This is Darla. She commenced a snooze mid meal...	8964	42908	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	13.0	10	Darla	NaN	
4	891327558926688256	This is Franklin. He would like you to stop ca...	9774	41048	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	12.0	10	Franklin	NaN	

The dataset has been cleaned and the resultant dataset is tidy and of high quality.