wrangle_act

November 9, 2018

In [100]:
```
import pandas as pd
import requests
import tweepy
import json
import matplotlib.pyplot as plt
%matplotlib inline
```

1 Gathering

In [2]:
```
#loading the data from the CSV file
df=pd.read_csv("twitter-archive-enhanced.csv")
```

In [3]:
```
#downloading the Images from the given URL.
url='https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions
response=requests.get(url)
with open(url.split('/')[-1],mode='wb') as file:
    file.write(response.content)
```

In [133]:
```
##loading the data from the TSV file
image_prd=pd.read_csv('image-predictions.tsv',sep='\t')
```

In [135]:
```
#Enter your secrete keys below!
consumer_key = 'CjQhIe8CwC7mFjWkY5l6q7NEb'
consumer_secret = 'bQzu4gMKldgwcHVmTMQQxULmnNjzvADSCxk1M5LC9FDfut0UgO'
access_token = '430048282-d5JzMDwkcoZYpQaKy1qOGFqyzzjkqaQdp8MMBVkh'
access_secret = 'r0Og2p1FOzxkU9aoghxt30R4a4dI1qlGNGReE6gQEssaY'

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

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

In [19]:
```
from timeit import default_timer as timer

tweet_tmr_lst=[]
excp_lst=[]
with open('tweet_json.txt','w',encoding='utf-8') as jsnfile:
```

```
            for tweetid in df['tweet_id']:
                try:
                    start = timer()
                    tweet=api.get_status(tweetid,tweet_mode='extended')
                    json.dump(tweet._json,jsnfile)
                    jsnfile.write('\n')
                    end = timer()
                    tweet_tmr_lst.append({tweetid:(end-start)})
                except:
                    excp_lst.append(tweetid)
                    continue
```

```
Rate limit reached. Sleeping for: 734
Rate limit reached. Sleeping for: 734
```

```
In [136]: #Read the json data.
          tweet_data=[]
          with open('tweet_json.txt','r') as tweetf:
              for line in tweetf:
                  tweetj=json.loads(line)
                  tweet_data.append(tweetj)
```

```
In [137]: #creating dataframe
          df_tweet=pd.DataFrame()
```

```
In [7]: df_tweet['tweet_id']=[tweet['id']for tweet in tweet_data]
        df_tweet['retweet_count']=[tweet['retweet_count']for tweet in tweet_data]
        df_tweet['favourite_count']=[tweet['favorite_count']for tweet in tweet_data]
```

2 Assessing

```
In [8]: df.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 [9]: df_tweet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2342 entries, 0 to 2341
Data columns (total 3 columns):
tweet_id           2342 non-null int64
retweet_count      2342 non-null int64
favourite_count    2342 non-null int64
dtypes: int64(3)
memory usage: 55.0 KB
```

```
In [10]: image_prd.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 [11]: df.sample(5)
```

```
Out[11]:                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         631   794205286408003585                    NaN                  NaN
         625   795400264262053889                    NaN                  NaN
         518   810657578271330305                    NaN                  NaN
```

```
1980  672975131468300288                    NaN                   NaN
1368  702539513671897089                    NaN                   NaN

                     timestamp  \
631   2016-11-03 15:51:10 +0000
625   2016-11-06 22:59:35 +0000
518   2016-12-19 01:26:42 +0000
1980  2015-12-05 03:05:49 +0000
1368  2016-02-24 17:04:07 +0000

                                                 source  \
631   <a href="http://twitter.com/download/iphone" r...
625   <a href="http://twitter.com/download/iphone" r...
518   <a href="http://twitter.com/download/iphone" r...
1980  <a href="http://twitter.com/download/iphone" r...
1368  <a href="http://twitter.com/download/iphone" r...

                                                   text  retweeted_status_id  \
631   This is Laika. She was a space pupper. The fir...                  NaN
625   This is Brody. He's trying to make the same fa...                  NaN
518   This is Pavlov. His floatation device has fail...                  NaN
1980  This is Chuckles. He is one skeptical pupper. ...                  NaN
1368  This is a Wild Tuscan Poofwiggle. Careful not ...                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
631                        NaN                        NaN
625                        NaN                        NaN
518                        NaN                        NaN
1980                       NaN                        NaN
1368                       NaN                        NaN

                                          expanded_urls  rating_numerator  \
631   https://twitter.com/dog_rates/status/794205286...                14
625   https://twitter.com/dog_rates/status/795400264...                12
518   https://twitter.com/dog_rates/status/810657578...                11
1980  https://twitter.com/dog_rates/status/672975131...                10
1368  https://twitter.com/dog_rates/status/702539513...                12

      rating_denominator      name doggo floofer  pupper puppo
631                   10     Laika  None    None  pupper  None
625                   10     Brody  None    None    None  None
518                   10    Pavlov  None    None    None  None
1980                  10  Chuckles  None    None  pupper  None
1368                  10         a  None    None    None  None
```

In [12]:
```
df_tweet.sample(5)
```

Out[12]:
```
                tweet_id  retweet_count  favourite_count
747   778408200802557953           4780            14629
```

```
1727  679475951516934144            686            2206
1183  717009362452090881           1043            3437
1524  689835978131935233            812            2288
679   787397959788929025           3134           11675
```

```
In [13]: image_prd.sample(5)
```

```
Out[13]:                 tweet_id                                          jpg_url  \
1800  831911600680497154    https://pbs.twimg.com/media/C4uLLGuUoAAkIHm.jpg
638   681281657291280384    https://pbs.twimg.com/media/CXRmDfWWMAADCdc.jpg
1234  746369468511756288    https://pbs.twimg.com/media/ClujESVXEAA4uH8.jpg
682   683834909291606017  https://pbs.twimg.com/ext_tw_video_thumb/68383...
1449  776201521193218049    https://pbs.twimg.com/media/CsWfKadWEAAtmlS.jpg

      img_num              p1   p1_conf  p1_dog                      p2  \
1800        4       bloodhound  0.777562    True               Great_Dane
638         1    Saint_Bernard  0.998830    True                 Pekinese
1234        1  German_shepherd  0.622957    True                 malinois
682         1      Maltese_dog  0.738449    True               toy_poodle
1449        1       Rottweiler  0.502228    True  black-and-tan_coonhound

       p2_conf  p2_dog             p3   p3_conf  p3_dog
1800  0.047418    True        Leonberg  0.017943    True
638   0.000391    True  Great_Pyrenees  0.000224    True
1234  0.338884    True         wallaby  0.024161   False
682   0.102992    True          Samoyed  0.023247    True
1449  0.154594    True      bloodhound  0.135176    True
```

```
In [14]: sum(df.tweet_id.duplicated())
```

```
Out[14]: 0
```

```
In [15]: sum(df_tweet.tweet_id.duplicated())
```

```
Out[15]: 0
```

```
In [16]: sum(image_prd.tweet_id.duplicated())
```

```
Out[16]: 0
```

```
In [17]: df.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
```

```
In [18]: df.query('rating_numerator== 960')
```

Out[18]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id
313	835246439529840640	8.352460e+17	26259576.0	2017-02-24 21:54:03 +0000	<a href="http://twitter.com/download/iphone" r...	@jonnysun @Lin_Manuel ok jomny I know you're e...	NaN

```
     retweeted_status_user_id retweeted_status_timestamp expanded_urls  \
313                      NaN                        NaN           NaN

     rating_numerator  rating_denominator  name doggo floofer pupper puppo
313               960                   0  None  None    None   None  None
```

```
In [19]: image_prd.query('tweet_id==835246439529840640')
```

```
Out[19]: Empty DataFrame
         Columns: [tweet_id, jpg_url, img_num, p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_
         Index: []
```

2.0.1 Quality issues:

Twitter archive table:

1-After visiting the dog_rates page on twitter, i saw that the tweet with id '835246439529840640' that have rating_numerator = 960 was not a tweet it was a reply and the rating was 13/10.

3- In_reply_to_status_id and in_reply_to_user_id columns dont make sense in our analysis since we have only 78 out of 2356 are non-null values and the tweet_id is the key.

4- timestamp is float datatype.

5- The source column is not clear since its URL.

Image predictions table:

1- The predictions are proportions.

2- Columns name are not clear.

3- Prediction names are in different cases (lower and upper)

2.0.2 Tidiness issues:

1- retweet_count and the favourite_count dont make sense if they are alone in a table.

2- Doggo, floofer, pupper, and puppo are 4 different columns in the archive table.

3 Cleaning

```
In [20]: #copy the dirty data into new one to start cleaning it.
         archive_clean=df.copy()
         tweet_clean=df_tweet.copy()
         img_clean=image_prd.copy()
```

```
In [21]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

3.0.1 Define

1-Drop the row with tweet_id '835246439529840640' that have rating_numerator = 960 from the archive table with index 313.

3.0.2 Code

```
In [22]: archive_clean.drop(archive_clean.index[313], inplace=True)
```

3.0.3 Test

```
In [23]: archive_clean.query('rating_numerator== 960')
```

```
Out[23]: Empty DataFrame
         Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text
         Index: []
```

```
In [24]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2355 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2355 non-null int64
in_reply_to_status_id         77 non-null float64
in_reply_to_user_id           77 non-null float64
timestamp                     2355 non-null object
source                        2355 non-null object
text                          2355 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              2355 non-null int64
rating_denominator            2355 non-null int64
name                          2355 non-null object
doggo                         2355 non-null object
floofer                       2355 non-null object
pupper                        2355 non-null object
puppo                         2355 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 331.2+ KB
```

3.0.4 Define

2- Since we are analysing the tweets, so there is no need for the retweets. Non null values in the retweet columns retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp should be dropped from the archive table.

3.0.5 Code

```
In [25]: retweet=archive_clean.query('retweeted_status_id!="NaN"' or 'retweeted_status_user_id!=
```

```
In [26]: len(retweet)
```

```
Out[26]: 181
```

```
In [27]: archive_clean=archive_clean.drop(retweet)
```

```
In [28]: archive_clean.drop(['retweeted_status_id','retweeted_status_user_id','retweeted_status_
```

3.0.6 Test

```
In [29]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2174 entries, 0 to 2355
Data columns (total 14 columns):
tweet_id                 2174 non-null int64
in_reply_to_status_id    77 non-null float64
in_reply_to_user_id      77 non-null float64
timestamp                2174 non-null object
source                   2174 non-null object
text                     2174 non-null object
expanded_urls            2117 non-null object
rating_numerator         2174 non-null int64
rating_denominator       2174 non-null int64
name                     2174 non-null object
doggo                    2174 non-null object
floofer                  2174 non-null object
pupper                   2174 non-null object
puppo                    2174 non-null object
dtypes: float64(2), int64(3), object(9)
memory usage: 254.8+ KB
```

3.0.7 Define

3- Drop In_reply_to_status_id and in_reply_to_user_id columns from the archive table.

3.0.8 Code

```
In [30]: archive_clean.drop(['in_reply_to_status_id' , 'in_reply_to_user_id'], axis=1, inplace=T
```

3.0.9 Test

```
In [31]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2174 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2174 non-null int64
timestamp             2174 non-null object
source                2174 non-null object
text                  2174 non-null object
expanded_urls         2117 non-null object
rating_numerator      2174 non-null int64
rating_denominator    2174 non-null int64
name                  2174 non-null object
doggo                 2174 non-null object
floofer               2174 non-null object
pupper                2174 non-null object
puppo                 2174 non-null object
dtypes: int64(3), object(9)
```

```
memory usage: 220.8+ KB
```

3.0.10 Define

4- Change the timestamp from float datatype into datetime datatype

3.0.11 Code

```
In [32]: archive_clean.timestamp = pd.to_datetime(archive_clean.timestamp, yearfirst = True)
```

3.0.12 Test

```
In [33]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2174 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2174 non-null int64
timestamp             2174 non-null datetime64[ns]
source                2174 non-null object
text                  2174 non-null object
expanded_urls         2117 non-null object
rating_numerator      2174 non-null int64
rating_denominator    2174 non-null int64
name                  2174 non-null object
doggo                 2174 non-null object
floofer               2174 non-null object
pupper                2174 non-null object
puppo                 2174 non-null object
dtypes: datetime64[ns](1), int64(3), object(8)
memory usage: 220.8+ KB
```

3.0.13 Define

5- Extract from the source url Twitter for iPhone, Vine - Make a Scene, Twitter Web Client and TweetDeck and replace them instead of the URL.

3.0.14 Code

```
In [34]: archive_clean.source.value_counts()
```

```
Out[34]: <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
```

```
In [35]: archive_clean.source=archive_clean['source'].replace('<a href="http://twitter.com/downl
         archive_clean.source=archive_clean['source'].replace('<a href="http://vine.co" rel="nof
         archive_clean.source=archive_clean['source'].replace('<a href="http://twitter.com" rel=
         archive_clean.source=archive_clean['source'].replace('<a href="https://about.twitter.co
```

3.0.15 Test

```
In [36]: archive_clean.source.value_counts()

Out[36]: Twitter for iPhone     2041
         Vine - Make a Scene      91
         Twitter Web Client       31
         TweetDeck                11
         Name: source, dtype: int64

In [37]: archive_clean.head(5)

Out[37]:              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 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
```

3.0.16 Define

6- change the predictions from proportions to percentages in the image prediction table

3.0.17 Code

```
In [38]: img_clean.head(1)
```

```
Out[38]:              tweet_id                                          jpg_url  \
         0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg

            img_num                     p1   p1_conf  p1_dog      p2   p2_conf  \
         0        1  Welsh_springer_spaniel  0.465074    True  collie  0.156665

            p2_dog                 p3   p3_conf  p3_dog
         0    True  Shetland_sheepdog  0.061428    True
```

```
In [39]: img_clean.p1_conf=img_clean['p1_conf'].apply(lambda x: x*100)
         img_clean.p2_conf=img_clean['p2_conf'].apply(lambda x: x*100)
         img_clean.p3_conf=img_clean['p3_conf'].apply(lambda x: x*100)
```

3.0.18 Test

```
In [40]: img_clean.head(1)
```

```
Out[40]:              tweet_id                                          jpg_url  \
         0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg

            img_num                     p1  p1_conf  p1_dog      p2  p2_conf  p2_dog  \
         0        1  Welsh_springer_spaniel  46.5074    True  collie  15.6665    True

                        p3  p3_conf  p3_dog
         0  Shetland_sheepdog  6.14285    True
```

3.0.19 Define

7- Rename the columns in the image prediction table to be more clear

3.0.20 Code

```
In [41]: img_clean=img_clean.rename(columns={'jpg_url': 'img_url', 'p1':'prediction_1', 'p1_conf
                           'p2':'prediction_2', 'p2_conf':'conf_percentage_2','p2_dog':'breed_pr
                           'p3':'prediction_3', 'p3_conf':'conf_percentage_3','p3_dog':'breed_pr
                                  )
```

3.0.21 Test

```
In [42]: img_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id              2075 non-null int64
img_url               2075 non-null object
```

```
img_num               2075 non-null int64
prediction_1          2075 non-null object
conf_percentage_1     2075 non-null float64
breed_prediction_1    2075 non-null bool
prediction_2          2075 non-null object
conf_percentage_2     2075 non-null float64
breed_prediction_2    2075 non-null bool
prediction_3          2075 non-null object
conf_percentage_3     2075 non-null float64
breed_prediction_3    2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

3.0.22 Define

8- Make all the prediction names of the same letter case. prediction_1, prediction_2 and prediction_3 must be with upper case.

3.0.23 Code

```
In [43]: img_clean.head()
```

Out[43]:

	tweet_id	img_url	img_num	prediction_1	conf_percentage_1	breed_prediction_1	prediction_2	conf_percentage_2	breed_prediction_2	prediction_3	conf_percentage_3	breed_prediction_3
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	46.5074	True	collie	15.66650	True	Shetland_sheepdog	6.14285	True
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	50.6826	True	miniature_pinscher	7.41917	True	Rhodesian_ridgeback	7.20100	True
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	59.6461	True	malinois	13.85840	True	bloodhound	11.61970	True
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	40.8143	True	redbone	36.06870	True			
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	56.0311	True	Rottweiler	24.36820	True			

```
3   miniature_pinscher           22.27520                True
4              Doberman          15.46290                True
```

```
In [44]: img_clean['prediction_1']=img_clean['prediction_1'].str.capitalize()
         img_clean['prediction_2']=img_clean['prediction_2'].str.capitalize()
         img_clean['prediction_3']=img_clean['prediction_3'].str.capitalize()
```

3.0.24 Test

```
In [45]: img_clean.sample(5)
```

Out[45]:

	tweet_id	img_url	\
1030	711306686208872448	https://pbs.twimg.com/media/Cd8RplOWOAAN1kU.jpg	
959	705591895322394625	https://pbs.twimg.com/media/CcrEFQdUcAA7CJf.jpg	
732	686730991906516992	https://pbs.twimg.com/media/CYfCMdFWAAA44YA.jpg	
1382	765395769549590528	https://pbs.twimg.com/media/Cp87YOjXYAQyjuV.jpg	
651	682003177596559360	https://pbs.twimg.com/media/CXb2RcDUsAEnkJb.jpg	

	img_num	prediction_1	conf_percentage_1	breed_prediction_1	\
1030	1	Leatherback_turtle	28.0835	False	
959	1	Basenji	87.7207	True	
732	1	Tibetan_mastiff	33.8812	True	
1382	1	Pembroke	50.9491	True	
651	1	Triceratops	24.9872	False	

	prediction_2	conf_percentage_2	breed_prediction_2	\
1030	Loggerhead	12.32900	False	
959	Italian_greyhound	4.78542	True	
732	Newfoundland	18.09250	True	
1382	Cardigan	33.04010	True	
651	Chimpanzee	6.09293	False	

	prediction_3	conf_percentage_3	breed_prediction_3
1030	Dandie_dinmont	8.67925	True
959	Miniature_pinscher	3.56381	True
732	Golden_retriever	18.00230	True
1382	Shetland_sheepdog	3.88749	True
651	Mask	5.02210	False

4 Tidiness issues:

4.0.1 Define:

1- Merge both tweet_clean table and the archive_clean into one table

4.0.2 Code:

```
In [46]: archive_clean=archive_clean.merge(tweet_clean, on="tweet_id", how = 'inner')
```

```
In [47]: archive_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2173 entries, 0 to 2172
Data columns (total 14 columns):
tweet_id              2173 non-null int64
timestamp             2173 non-null datetime64[ns]
source                2173 non-null object
text                  2173 non-null object
expanded_urls         2116 non-null object
rating_numerator      2173 non-null int64
rating_denominator    2173 non-null int64
name                  2173 non-null object
doggo                 2173 non-null object
floofer               2173 non-null object
pupper                2173 non-null object
puppo                 2173 non-null object
retweet_count         2173 non-null int64
favourite_count       2173 non-null int64
dtypes: datetime64[ns](1), int64(5), object(8)
memory usage: 254.6+ KB
```

4.0.3 Test:

```
In [48]: archive_clean.sample(3)

Out[48]:                tweet_id           timestamp              source  \
         795   750086836815486976 2016-07-04 22:00:12          TweetDeck
         1518  680970795137544192 2015-12-27 04:37:44  Twitter for iPhone
         251   837471256429613056 2017-03-03 01:14:41  Twitter for iPhone

                                                             text  \
         795   This is Spanky. He was a member of the 2002 US...
         1518  I thought I made this very clear. We only rate...
         251   This is Vincent. He's suave as h*ck. Will be y...

                                                expanded_urls  rating_numerator  \
         795   https://twitter.com/dog_rates/status/750086836...                12
         1518  https://twitter.com/dog_rates/status/680970795...                 9
         251   https://twitter.com/dog_rates/status/837471256...                12

               rating_denominator     name doggo floofer pupper puppo  retweet_count  \
         795                   10   Spanky  None    None   None  None            591
         1518                  10     None  None    None   None  None            717
         251                   10  Vincent  None    None   None  None           2477

               favourite_count
```

```
795              2305
1518             2564
251             13483
```

4.0.4 Define:

2- Create a new column " dog_stage" as 'categorical' datatype in the archive table with variables doggo, floofer, pupper, and puppo.

4.0.5 Code:

```
In [49]: archive_clean['dog_stage'] = archive_clean.text.str.extract('(doggo | floofer | pupper
```

```
In [50]: archive_clean.dog_stage=archive_clean.dog_stage.astype('category')
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2173 entries, 0 to 2172
Data columns (total 15 columns):
tweet_id              2173 non-null int64
timestamp             2173 non-null datetime64[ns]
source                2173 non-null object
text                  2173 non-null object
expanded_urls         2116 non-null object
rating_numerator      2173 non-null int64
rating_denominator    2173 non-null int64
name                  2173 non-null object
doggo                 2173 non-null object
floofer               2173 non-null object
pupper                2173 non-null object
puppo                 2173 non-null object
retweet_count         2173 non-null int64
favourite_count       2173 non-null int64
dog_stage             240 non-null category
dtypes: category(1), datetime64[ns](1), int64(5), object(8)
memory usage: 257.0+ KB
```

```
In [52]: archive_clean.drop(['doggo', 'floofer', 'pupper', 'puppo'], axis=1, inplace=True)
```

4.0.6 Test:

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2173 entries, 0 to 2172
Data columns (total 11 columns):
tweet_id              2173 non-null int64
```

```
timestamp             2173 non-null datetime64[ns]
source                2173 non-null object
text                  2173 non-null object
expanded_urls         2116 non-null object
rating_numerator      2173 non-null int64
rating_denominator    2173 non-null int64
name                  2173 non-null object
retweet_count         2173 non-null int64
favourite_count       2173 non-null int64
dog_stage             240 non-null category
dtypes: category(1), datetime64[ns](1), int64(5), object(4)
memory usage: 189.1+ KB
```

```
In [54]: archive_clean.head(20)
```

Out[54]:

	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	
5	891087950875897856	2017-07-29 00:08:17	Twitter for iPhone	
6	890971913173991426	2017-07-28 16:27:12	Twitter for iPhone	
7	890729181411237888	2017-07-28 00:22:40	Twitter for iPhone	
8	890609185150312448	2017-07-27 16:25:51	Twitter for iPhone	
9	890240255349198849	2017-07-26 15:59:51	Twitter for iPhone	
10	890006608113172480	2017-07-26 00:31:25	Twitter for iPhone	
11	889880896479866881	2017-07-25 16:11:53	Twitter for iPhone	
12	889665388333682689	2017-07-25 01:55:32	Twitter for iPhone	
13	889638837579907072	2017-07-25 00:10:02	Twitter for iPhone	
14	889531135344209921	2017-07-24 17:02:04	Twitter for iPhone	
15	889278841981685760	2017-07-24 00:19:32	Twitter for iPhone	
16	888917238123831296	2017-07-23 00:22:39	Twitter for iPhone	
17	888804989199671297	2017-07-22 16:56:37	Twitter for iPhone	
18	888554962724278272	2017-07-22 00:23:06	Twitter for iPhone	
19	888078434458587136	2017-07-20 16:49:33	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...	
5	Here we have a majestic great white breaching ...	
6	Meet Jax. He enjoys ice cream so much he gets ...	
7	When you watch your owner call another dog a g...	
8	This is Zoey. She doesn't want to be one of th...	

```
9   This is Cassie. She is a college pup. Studying...
10  This is Koda. He is a South Australian decksha...
11  This is Bruno. He is a service shark. Only get...
12  Here's a puppo that seems to be on the fence a...
13  This is Ted. He does his best. Sometimes that'...
14  This is Stuart. He's sporting his favorite fan...
15  This is Oliver. You're witnessing one of his m...
16  This is Jim. He found a fren. Taught him how t...
17  This is Zeke. He has a new stick. Very proud o...
18  This is Ralphus. He's powering up. Attempting ...
19  This is Gerald. He was just told he didn't get...

                                          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
5   https://twitter.com/dog_rates/status/891087950...                 13
6   https://gofundme.com/ydvmve-surgery-for-jax,ht...                 13
7   https://twitter.com/dog_rates/status/890729181...                 13
8   https://twitter.com/dog_rates/status/890609185...                 13
9   https://twitter.com/dog_rates/status/890240255...                 14
10  https://twitter.com/dog_rates/status/890006608...                 13
11  https://twitter.com/dog_rates/status/889880896...                 13
12  https://twitter.com/dog_rates/status/889665388...                 13
13  https://twitter.com/dog_rates/status/889638837...                 12
14  https://twitter.com/dog_rates/status/889531135...                 13
15  https://twitter.com/dog_rates/status/889278841...                 13
16  https://twitter.com/dog_rates/status/888917238...                 12
17  https://twitter.com/dog_rates/status/888804989...                 13
18  https://twitter.com/dog_rates/status/888554962...                 13
19  https://twitter.com/dog_rates/status/888078434...                 12

    rating_denominator      name  retweet_count  favourite_count dog_stage
0                   10   Phineas           8378            38257       NaN
1                   10     Tilly           6186            32798       NaN
2                   10    Archie           4092            24698       NaN
3                   10     Darla           8520            41586       NaN
4                   10  Franklin           9229            39761       NaN
5                   10      None           3070            19956       NaN
6                   10       Jax           2036            11681       NaN
7                   10      None          18607            64571       NaN
8                   10      Zoey           4212            27429       NaN
9                   10    Cassie           7279            31482    doggo
10                  10      Koda           7229            30255       NaN
11                  10     Bruno           4902            27413       NaN
12                  10      None           9917            47483     puppo
```

```
13                  10        Ted          4480            26803       NaN
14                  10     Stuart          2209            14901     puppo
15                  10     Oliver          5304            24917       NaN
16                  10        Jim          4432            28688       NaN
17                  10       Zeke          4236            25226       NaN
18                  10    Ralphus          3499            19584       NaN
19                  10     Gerald          3447            21474       NaN
```

4.0.7 Storing and Analysing

```
In [118]: archive_clean.to_csv('twitter_archive_master.csv', index= False)
          img_clean.to_csv('image_prediction_master.csv', index=False)

In [119]: archive=pd.read_csv('twitter_archive_master.csv')
          prediction=pd.read_csv('image_prediction_master.csv')

In [121]: prediction.describe()

Out[121]:            tweet_id      img_num  conf_percentage_1  conf_percentage_2  \
          count  2.075000e+03  2075.000000        2075.000000        2075.000000
          mean   7.384514e+17     1.203855          59.454826          13.458861
          std    6.785203e+16     0.561875          27.117352          10.066574
          min    6.660209e+17     1.000000           4.433340           0.000001
          25%    6.764835e+17     1.000000          36.441200           5.388625
          50%    7.119988e+17     1.000000          58.823000          11.818100
          75%    7.932034e+17     1.000000          84.385500          19.556550
          max    8.924206e+17     4.000000         100.000000          48.801400

                 conf_percentage_3
          count       2.075000e+03
          mean        6.032417e+00
          std         5.090593e+00
          min         1.740170e-08
          25%         1.622240e+00
          50%         4.944380e+00
          75%         9.180755e+00
          max         2.734190e+01
```

From reading the above table we can conclude that alogirthm 1 is the most confident between the three alogorithms. This is clear from the readings of the 25%,50% and the 75% percentiles.

```
In [122]: prediction[prediction.breed_prediction_1==True].prediction_1.value_counts().head(5)

Out[122]: Golden_retriever      150
          Labrador_retriever    100
          Pembroke               89
          Chihuahua              83
          Pug                    57
          Name: prediction_1, dtype: int64
```

These are the top 5 predicted breed dogs from the 1st algorithm

```
In [123]: prediction[prediction.breed_prediction_2==True].prediction_2.value_counts().head(5)

Out[123]: Labrador_retriever    104
          Golden_retriever       92
          Cardigan               73
          Chihuahua              44
          Pomeranian             42
          Name: prediction_2, dtype: int64
```

These are the top 5 predicted breed dogs from the 2nd algorithm

```
In [124]: prediction[prediction.breed_prediction_3==True].prediction_3.value_counts().head(5)

Out[124]: Labrador_retriever    79
          Chihuahua             58
          Golden_retriever      48
          Eskimo_dog            38
          Kelpie                35
          Name: prediction_3, dtype: int64
```

These are the top 5 predicted breed dogs from the 3rd algorithm

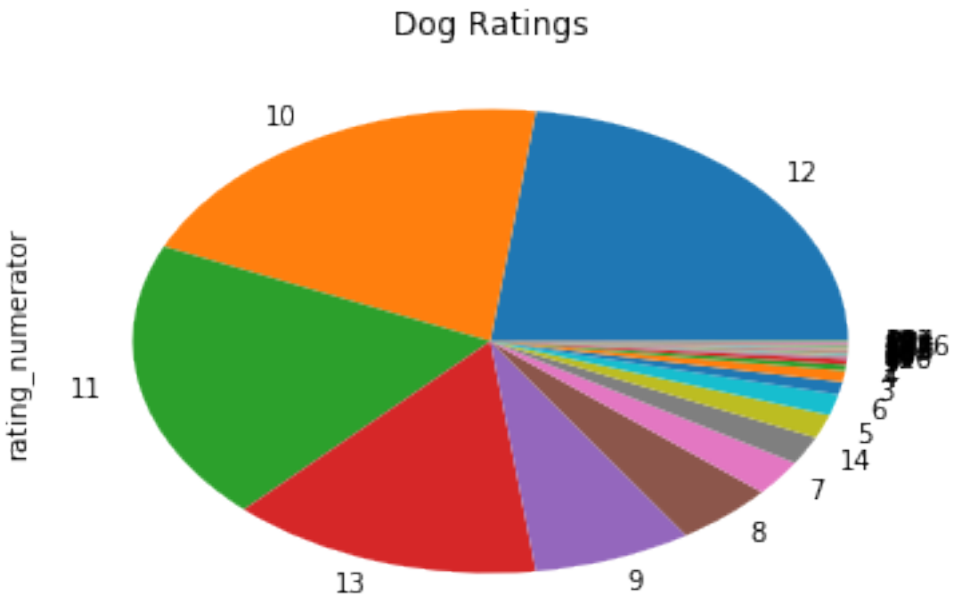
```
In [125]: prediction.groupby('breed_prediction_1')['conf_percentage_1'].mean()

Out[125]: breed_prediction_1
          False    54.016655
          True     61.382324
          Name: conf_percentage_1, dtype: float64

In [126]: prediction.groupby('breed_prediction_2')['conf_percentage_2'].mean()

Out[126]: breed_prediction_2
          False    11.708954
          True     14.047046
          Name: conf_percentage_2, dtype: float64

In [127]: prediction.groupby('breed_prediction_3')['conf_percentage_3'].mean()

Out[127]: breed_prediction_3
          False    5.689345
          True     6.164244
          Name: conf_percentage_3, dtype: float64
```

In the three algorithms the % of confidence is higher when breed dog was predicted. However, the 1st algorithm is the most accurate between the 3 algorithms with True % cofidence 61.3%

```
In [130]: archive.groupby('rating_numerator')['retweet_count'].mean().sort_values(ascending = Fa

Out[130]: rating_numerator
          14     8198.325581
          75     6719.000000
          13     6386.302932
          420    4537.000000
          84     3542.000000
          Name: retweet_count, dtype: float64

In [129]: archive.groupby('rating_numerator')['favourite_count'].mean().sort_values(ascending =

Out[129]: rating_numerator
          14    25845.116279
          13    21030.413681
          75    19560.000000
          0     13382.500000
          84    13039.000000
          Name: favourite_count, dtype: float64
```

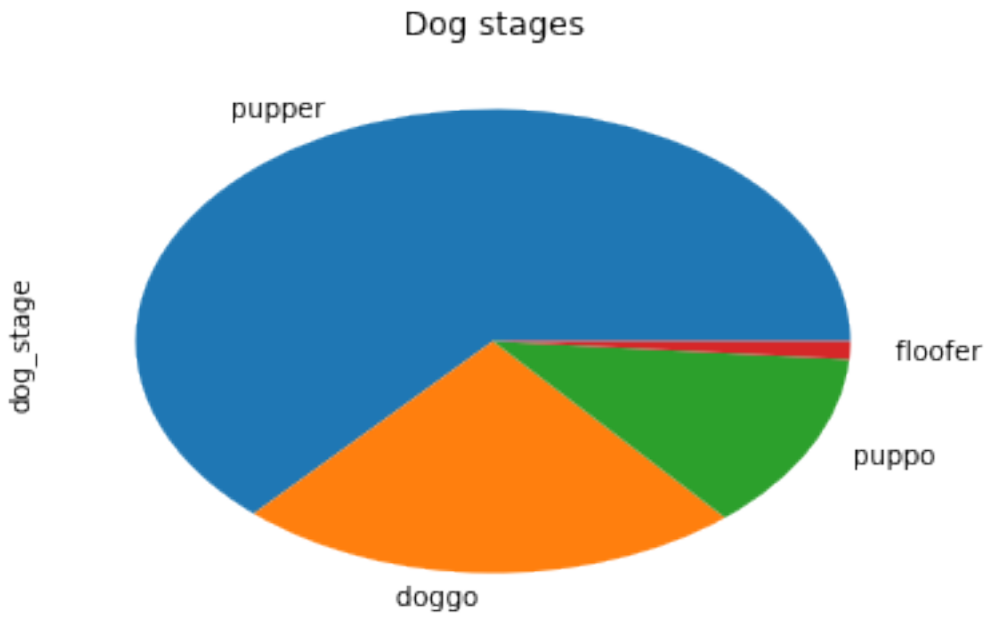
These results shows the top 5 average number of 'retweets' and 'favorite' respectively for the rating_numerator used in ratings.

```
In [131]: archive.rating_numerator.value_counts().plot(kind='pie')
          plt.title('Dog Ratings')

Out[131]: Text(0.5,1,'Dog Ratings')
```



This Pie shows that the 10,11,12,13 are the most numbers used to rate the dogs.

```
In [132]: archive.dog_stage.value_counts().plot(kind='pie')
          plt.title('Dog stages')

Out[132]: Text(0.5,1,'Dog stages')
```



Pupper is the most dog stage used then doggo, puppo and floofer