act-report

January 5, 2021

0.1 Quality Issues

0.1.1 df:

1) Missing data in the following columns: in_reply_to_status_id, in_reply_to_user_id, Retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls
2) This dataset includes retweets, which means there is duplicated data
3) Timestamp and retweeted_status_timestamp is an object and not correct datetime frame.
4) The source column still has the HTML tags
5) Dogs name have 'None', or 'a', or 'an.' and some more lower case words as names
6) Multiple dog stages occurs such as 'doggo puppo', 'doggo pupper', 'doggo floofer'

0.1.2 image_df:

1) Dog breeds are not consistently in p1,p2,p3 columns: this column contains dog breeds name starts with upper case or lower case.

0.1.3 df_tweet_json:

1) Date type of column tweet_id is an object. It should be an int in order to merge it to the master df.

0.2 Tidiness Issues

0.2.1 df:

1) The variable for the dog's stage (dogoo, floofer, pupper, puppo) is spread in different columns which should be in one column.

0.2.2 image_df:

2) This data set is part of the same observational unit as the data in the archive_df

0.2.3 df_tweet_json:

3) This data set is also part of the same observational unit as the data in the archive_df

1 Cleaning Data

```
In [31]: #Making a copy of the dataframes before cleaning
         df_clean = df.copy()
         image_df_clean = image_df.copy()
         tweet_json_clean = df_tweet_json.copy()
```

1.1 DEFINE-CODE-TEST

1) Convert the tweet_id in tweet_json_clean dataframe into int type for merging into master dataframe
2) Creates a predicted dog breed column, based on the the confidence level of minimum 20% and 'p1_dog', 'p2_dog' and 'p3_dog' statements
3) Create one column for the various dog types: doggo, floofer, pupper, puppo, 'doggo, puppo', 'doggo, pupper', 'doggo, floofer' ascolumn name ' type ' with the categorical dtype
4) Merge the copied df_clean, image_df_clean, and tweet_json_clean dataframes
5) Convert the tweet_id in master_df into object type as there is no use for maths operation in tweet_id
6) Replace 'a', 'an', 'the', 'None' and other lower case words with NaN in name column
7) Remove Inconsistency in pred_breed
8) Delete retweets
9) Remove columns no longer needed: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp
10) Change the timestamp to correct datetime format
11) Removing HTML tags from source column
12) Dog ratings get standardized for denom of 10.

1.1.1 1. Convert the tweet_id in tweet_json_clean dataframe into int type for merging into master dataframe

```
In [32]: tweet_json_clean['tweet_id'] = tweet_json_clean['tweet_id'].astype('int64')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2331 entries, 0 to 2330
Data columns (total 3 columns):
tweet_id          2331 non-null int64
retweet_count     2331 non-null object
favorite_count    2331 non-null object
dtypes: int64(1), object(2)
memory usage: 54.7+ KB
```

1.1.2 2. Creates a predicted dog breed column, based on the the confidence level of minimum 20% and 'p1_dog', 'p2_dog' and 'p3_dog' statements

In [34]:
```
image_df_clean.sample()
```

Out[34]:
```
                tweet_id                                           jpg_url  \
1170  736225175608430592  https://pbs.twimg.com/media/CjeY5DKXEAA3WkD.jpg

      img_num                  p1   p1_conf  p1_dog  \
1170        1  Labrador_retriever  0.399217    True

                                p2  p2_conf  p2_dog             p3   p3_conf  \
1170  West_Highland_white_terrier  0.13771    True  cocker_spaniel  0.062033

      p3_dog
1170    True
```

In [35]:
```
image_df_clean['pred_breed'] = [df['p1'] if df['p1_dog'] == True and df['p1_conf'] > 0.
                           else df['p2'] if df['p2_dog'] == True and df['p2_conf'] > 0.2
                           else df['p3'] if df['p3_dog'] == True and df['p3_conf'] > 0.2
                           else np.nan for index, df in image_df_clean.iterrows()]
```

In [36]:
```
# Drop 'p1', 'p1_dog', 'p1_conf','p2', 'p2_dog', 'p2_conf','p3', 'p3_dog', 'p3_conf' co
image_df_clean.drop(['p1', 'p1_dog', 'p1_conf','p2', 'p2_dog', 'p2_conf','p3', 'p3_dog'
```

In [37]:
```
image_df_clean.head()
```

Out[37]:
```
             tweet_id                                          jpg_url  \
0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2  666033412701032449  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3  666044226329800704  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4  666049248165822465  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg

   img_num             pred_breed
0        1  Welsh_springer_spaniel
1        1                redbone
2        1         German_shepherd
3        1     Rhodesian_ridgeback
4        1      miniature_pinscher
```

1.1.3 3) Create one column for the various dog types: doggo, floofer, pupper, puppo, 'doggo, puppo', 'doggo, pupper', 'doggo, floofer' ascolumn name ' type ' with the categorical dtype

In [38]:
```
#Number of columns in df_clean
df_clean.columns
```

Out[38]:
```
Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
       'source', 'text', 'retweeted_status_id', 'retweeted_status_user_id',
```

```
       'retweeted_status_timestamp', 'expanded_urls', 'rating_numerator',
       'rating_denominator', 'name', 'doggo', 'floofer', 'pupper', 'puppo'],
      dtype='object')
```

In [39]:
```
# as there are separate columns for dogs type 'doggo','floofer','pupper'and so on...
#i will convert them into one column
df_clean.doggo.replace(np.NaN, '', inplace=True)
df_clean.floofer.replace(np.NaN, '', inplace=True)
df_clean.pupper.replace(np.NaN, '', inplace=True)
df_clean.puppo.replace(np.NaN, '', inplace=True)
df_clean.doggo.replace('None', '', inplace=True)
df_clean.floofer.replace('None', '', inplace=True)
df_clean.pupper.replace('None', '', inplace=True)
df_clean.puppo.replace('None', '', inplace=True)
```

In [40]:
```
df_clean['stage'] = df_clean.doggo + df_clean.floofer + df_clean.pupper + df_clean.pupp
df_clean.loc[df_clean.stage == 'doggopupper', 'stage'] = 'doggo, pupper'
df_clean.loc[df_clean.stage == 'doggopuppo', 'stage'] = 'doggo, puppo'
df_clean.loc[df_clean.stage == 'doggofloofer', 'stage'] = 'doggo, floofer'
```

In [41]:
```
# Convert the stage in df_clean into categorical dtype
df_clean['stage'] = df_clean['stage'].astype('category')
```

In [42]:
```
# drop 'doggo', 'floofer', 'pupper', 'puppo' columns
df_clean.drop(['doggo', 'floofer', 'pupper', 'puppo'], axis=1, inplace=True)
df_clean.stage.replace('', np.nan, inplace=True)
```

In [43]:
```
df_clean.info()
df_clean.stage.value_counts()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 14 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
stage                         380 non-null category
dtypes: category(1), float64(4), int64(3), object(6)
memory usage: 242.0+ KB
```

```
Out[43]: pupper            245
         doggo              83
         puppo              29
         doggo, pupper      12
         floofer             9
         doggo, puppo        1
         doggo, floofer      1
                             0
         Name: stage, dtype: int64
```

```
In [44]: df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 14 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
stage                         380 non-null category
dtypes: category(1), float64(4), int64(3), object(6)
memory usage: 242.0+ KB
```

1.1.4 4) Merge the copied df_clean, image_df_clean, and tweet_json_clean dataframes

```
In [45]: from functools import reduce
         data = [df_clean, image_df_clean, tweet_json_clean]
         main_df = reduce(lambda left, right:  pd.merge(left, right,on = 'tweet_id'), data)
```

```
In [46]: main_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2059 entries, 0 to 2058
Data columns (total 19 columns):
tweet_id                      2059 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2059 non-null object
source                        2059 non-null object
```

```
text                          2059 non-null object
retweeted_status_id           72 non-null float64
retweeted_status_user_id      72 non-null float64
retweeted_status_timestamp    72 non-null object
expanded_urls                 2059 non-null object
rating_numerator              2059 non-null int64
rating_denominator            2059 non-null int64
name                          2059 non-null object
stage                         318 non-null category
jpg_url                       2059 non-null object
img_num                       2059 non-null int64
pred_breed                    1460 non-null object
retweet_count                 2059 non-null object
favorite_count                2059 non-null object
dtypes: category(1), float64(4), int64(4), object(10)
memory usage: 308.0+ KB
```

1.1.5 5) Convert the tweet_id in master_df into object type as there is no use for maths operation in tweet_id

```
In [47]: main_df['tweet_id'] = main_df['tweet_id'].astype('object')

In [48]: main_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2059 entries, 0 to 2058
Data columns (total 19 columns):
tweet_id                      2059 non-null object
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2059 non-null object
source                        2059 non-null object
text                          2059 non-null object
retweeted_status_id           72 non-null float64
retweeted_status_user_id      72 non-null float64
retweeted_status_timestamp    72 non-null object
expanded_urls                 2059 non-null object
rating_numerator              2059 non-null int64
rating_denominator            2059 non-null int64
name                          2059 non-null object
stage                         318 non-null category
jpg_url                       2059 non-null object
img_num                       2059 non-null int64
pred_breed                    1460 non-null object
retweet_count                 2059 non-null object
favorite_count                2059 non-null object
dtypes: category(1), float64(4), int64(3), object(11)
```

```
memory usage: 308.0+ KB
```

1.1.6 6. Replace 'a', 'an', 'the', 'None' and other lower case words with NaN in name column

```
In [49]: words = main_df[main_df.name.str.islower()].name.unique()

In [50]: main_df['name'] = main_df['name'].replace(words, np.nan)
         main_df['name'] = main_df['name'].replace('None', np.nan)

In [51]: main_df['name'].dropna()

Out[51]: 0          Phineas
         1            Tilly
         2           Archie
         3            Darla
         4         Franklin
         6              Jax
         8             Zoey
         9           Cassie
         10            Koda
         11           Bruno
         13             Ted
         14          Stuart
         15          Oliver
         16             Jim
         17            Zeke
         18         Ralphus
         19          Gerald
         20         Jeffrey
         22          Canela
         25            Maya
         26          Mingus
         27           Derek
         28          Roscoe
         29         Waffles
         30           Jimbo
         31          Maisey
         32           Lilly
         34            Earl
         35            Lola
         36           Kevin
                      ...
         1972          Dook
         1974          Hall
         1975      Philippe
         1978         Reese
         1979       Cupcake
         1983         Biden
```

```
1984          Fwed
1986     Genevieve
1987        Joshwa
1990       Timison
1993      Clarence
1994       Kenneth
1995       Churlie
1996       Bradlay
1997         Pipsy
1999          Gabe
2000         Clybe
2001          Dave
2003          Keet
2005        Klevin
2006         Carll
2011          Jeph
2012       Jockson
2015         Josep
2016         Lugan
2018    Christoper
2020       Jimothy
2021      Kreggory
2022         Scout
2028        Walter
Name: name, Length: 1386, dtype: object
```

```
In [52]: main_df.name.value_counts()
```

```
Out[52]: Cooper       10
         Penny        10
         Oliver       10
         Tucker       10
         Charlie      10
         Lucy          9
         Sadie         8
         Bo            8
         Lola          8
         Winston       8
         Daisy         7
         Toby          7
         Scout         6
         Stanley       6
         Koda          6
         Milo          6
         Bella         6
         Jax           6
         Dave          6
         Rusty         6
```

```
Bailey        6
Louis         5
Alfie         5
Oscar         5
Larry         5
Leo           5
Buddy         5
Chester       5
Sophie        4
Bear          4
              ..
Barney        1
Strider       1
Miguel        1
Ridley        1
Chaz          1
Kota          1
Mairi         1
Enchilada     1
Mattie        1
Brudge        1
Noosh         1
Bodie         1
Huck          1
Alexander     1
Mojo          1
Venti         1
Schnozz       1
Howie         1
Bode          1
Keurig        1
Rodney        1
Lilah         1
Nida          1
Pawnd         1
Shawwn        1
Samsom        1
Harrison      1
Reagan        1
Burt          1
Lucky         1
Name: name, Length: 911, dtype: int64
```

1.1.7 7) Delete Retweets

```
In [53]: # Delete the rows which contains retweets
         main_df = main_df.drop(main_df[(main_df['in_reply_to_status_id'].isnull() == False) | (
```

```
In [54]: main_df.shape[0]
```

```
Out[54]: 1964
```

```
In [55]: main_df.shape
```

```
Out[55]: (1964, 19)
```

1.1.8 8) Remove columns no longer needed: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp

```
In [56]: # drop the reply status and retweet status columns
         main_df.drop(['in_reply_to_status_id', 'in_reply_to_user_id','retweeted_status_id', 're
                'retweeted_status_timestamp'], axis=1, inplace=True)
```

```
In [57]: main_df.columns
```

```
Out[57]: Index(['tweet_id', 'timestamp', 'source', 'text', 'expanded_urls',
                'rating_numerator', 'rating_denominator', 'name', 'stage', 'jpg_url',
                'img_num', 'pred_breed', 'retweet_count', 'favorite_count'],
               dtype='object')
```

```
In [58]: main_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1964 entries, 0 to 2058
Data columns (total 14 columns):
tweet_id              1964 non-null object
timestamp             1964 non-null object
source                1964 non-null object
text                  1964 non-null object
expanded_urls         1964 non-null object
rating_numerator      1964 non-null int64
rating_denominator    1964 non-null int64
name                  1342 non-null object
stage                 302 non-null category
jpg_url               1964 non-null object
img_num               1964 non-null int64
pred_breed            1395 non-null object
retweet_count         1964 non-null object
favorite_count        1964 non-null object
dtypes: category(1), int64(3), object(10)
memory usage: 217.1+ KB
```

1.1.9 9) Change the timestamp to correct datetime format

```
In [59]: main_df['timestamp'].sample(5)
```

```
Out[59]: 842     2016-06-17 00:05:25 +0000
         1991    2015-11-19 03:10:02 +0000
         198     2017-03-23 18:07:10 +0000
         596     2016-09-21 17:42:10 +0000
         1038    2016-03-17 00:58:46 +0000
         Name: timestamp, dtype: object

In [60]: main_df['timestamp'] = pd.to_datetime(main_df['timestamp'], format='%Y-%m-%d %H:%M:%S')

In [61]: main_df['timestamp'].sample(5)

Out[61]: 260    2017-02-16 17:00:25
         1887   2015-11-24 03:29:51
         64     2017-06-23 16:00:04
         344    2017-01-12 00:55:47
         363    2017-01-05 21:29:55
         Name: timestamp, dtype: datetime64[ns]

In [62]: main_df['timestamp'].describe()

Out[62]: count                    1964
         unique                   1964
         top       2016-08-04 22:52:29
         freq                        1
         first     2015-11-15 22:32:08
         last      2017-08-01 16:23:56
         Name: timestamp, dtype: object
```

1.1.10 10) Removing HTML tags from source column

```
In [63]: href = main_df["source"].str.split('"', expand = True)
         main_df["source"] = href[1]

In [64]: main_df.head()

Out[64]:             tweet_id           timestamp                              source  \
         0  892420643555336193 2017-08-01 16:23:56  http://twitter.com/download/iphone
         1  892177421306343426 2017-08-01 00:17:27  http://twitter.com/download/iphone
         2  891815181378084864 2017-07-31 00:18:03  http://twitter.com/download/iphone
         3  891689557279858688 2017-07-30 15:58:51  http://twitter.com/download/iphone
         4  891327558926688256 2017-07-29 16:00:24  http://twitter.com/download/iphone

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

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

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

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

  retweet_count favorite_count
0          7437          35276
1          5527          30527
2          3652          22954
3          7616          38561
4          8197          36842
```

In [65]: href

Out[65]:

```
              0                                      1      2         3  \
0     <a href=  http://twitter.com/download/iphone   rel=  nofollow
1     <a href=  http://twitter.com/download/iphone   rel=  nofollow
2     <a href=  http://twitter.com/download/iphone   rel=  nofollow
3     <a href=  http://twitter.com/download/iphone   rel=  nofollow
4     <a href=  http://twitter.com/download/iphone   rel=  nofollow
5     <a href=  http://twitter.com/download/iphone   rel=  nofollow
6     <a href=  http://twitter.com/download/iphone   rel=  nofollow
7     <a href=  http://twitter.com/download/iphone   rel=  nofollow
8     <a href=  http://twitter.com/download/iphone   rel=  nofollow
9     <a href=  http://twitter.com/download/iphone   rel=  nofollow
10    <a href=  http://twitter.com/download/iphone   rel=  nofollow
11    <a href=  http://twitter.com/download/iphone   rel=  nofollow
12    <a href=  http://twitter.com/download/iphone   rel=  nofollow
13    <a href=  http://twitter.com/download/iphone   rel=  nofollow
14    <a href=  http://twitter.com/download/iphone   rel=  nofollow
15    <a href=  http://twitter.com/download/iphone   rel=  nofollow
16    <a href=  http://twitter.com/download/iphone   rel=  nofollow
```

```
17    <a href=  http://twitter.com/download/iphone   rel=  nofollow
18    <a href=  http://twitter.com/download/iphone   rel=  nofollow
19    <a href=  http://twitter.com/download/iphone   rel=  nofollow
20    <a href=  http://twitter.com/download/iphone   rel=  nofollow
21    <a href=  http://twitter.com/download/iphone   rel=  nofollow
22    <a href=  http://twitter.com/download/iphone   rel=  nofollow
23    <a href=  http://twitter.com/download/iphone   rel=  nofollow
24    <a href=  http://twitter.com/download/iphone   rel=  nofollow
25    <a href=  http://twitter.com/download/iphone   rel=  nofollow
26    <a href=  http://twitter.com/download/iphone   rel=  nofollow
27    <a href=  http://twitter.com/download/iphone   rel=  nofollow
28    <a href=  http://twitter.com/download/iphone   rel=  nofollow
29    <a href=  http://twitter.com/download/iphone   rel=  nofollow
...       ...                                  ...    ...       ...
2029  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2030  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2031  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2032  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2033  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2034  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2035  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2036  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2037  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2038  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2039  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2040  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2041  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2042  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2043  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2044  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2045  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2046  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2047  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2048  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2049  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2050  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2051  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2052  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2053  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2054  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2055  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2056  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2057  <a href=  http://twitter.com/download/iphone   rel=  nofollow
2058  <a href=  http://twitter.com/download/iphone   rel=  nofollow

                            4
0     >Twitter for iPhone</a>
1     >Twitter for iPhone</a>
```

```
2     >Twitter for iPhone</a>
3     >Twitter for iPhone</a>
4     >Twitter for iPhone</a>
5     >Twitter for iPhone</a>
6     >Twitter for iPhone</a>
7     >Twitter for iPhone</a>
8     >Twitter for iPhone</a>
9     >Twitter for iPhone</a>
10    >Twitter for iPhone</a>
11    >Twitter for iPhone</a>
12    >Twitter for iPhone</a>
13    >Twitter for iPhone</a>
14    >Twitter for iPhone</a>
15    >Twitter for iPhone</a>
16    >Twitter for iPhone</a>
17    >Twitter for iPhone</a>
18    >Twitter for iPhone</a>
19    >Twitter for iPhone</a>
20    >Twitter for iPhone</a>
21    >Twitter for iPhone</a>
22    >Twitter for iPhone</a>
23    >Twitter for iPhone</a>
24    >Twitter for iPhone</a>
25    >Twitter for iPhone</a>
26    >Twitter for iPhone</a>
27    >Twitter for iPhone</a>
28    >Twitter for iPhone</a>
29    >Twitter for iPhone</a>
...                        ...
2029  >Twitter for iPhone</a>
2030  >Twitter for iPhone</a>
2031  >Twitter for iPhone</a>
2032  >Twitter for iPhone</a>
2033  >Twitter for iPhone</a>
2034  >Twitter for iPhone</a>
2035  >Twitter for iPhone</a>
2036  >Twitter for iPhone</a>
2037  >Twitter for iPhone</a>
2038  >Twitter for iPhone</a>
2039  >Twitter for iPhone</a>
2040  >Twitter for iPhone</a>
2041  >Twitter for iPhone</a>
2042  >Twitter for iPhone</a>
2043  >Twitter for iPhone</a>
2044  >Twitter for iPhone</a>
2045  >Twitter for iPhone</a>
2046  >Twitter for iPhone</a>
2047  >Twitter for iPhone</a>
```

```
2048  >Twitter for iPhone</a>
2049  >Twitter for iPhone</a>
2050  >Twitter for iPhone</a>
2051  >Twitter for iPhone</a>
2052  >Twitter for iPhone</a>
2053  >Twitter for iPhone</a>
2054  >Twitter for iPhone</a>
2055  >Twitter for iPhone</a>
2056  >Twitter for iPhone</a>
2057  >Twitter for iPhone</a>
2058  >Twitter for iPhone</a>

[1964 rows x 5 columns]
```

```
In [66]: main_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1964 entries, 0 to 2058
Data columns (total 14 columns):
tweet_id              1964 non-null object
timestamp             1964 non-null datetime64[ns]
source                1964 non-null object
text                  1964 non-null object
expanded_urls         1964 non-null object
rating_numerator      1964 non-null int64
rating_denominator    1964 non-null int64
name                  1342 non-null object
stage                 302 non-null category
jpg_url               1964 non-null object
img_num               1964 non-null int64
pred_breed            1395 non-null object
retweet_count         1964 non-null object
favorite_count        1964 non-null object
dtypes: category(1), datetime64[ns](1), int64(3), object(9)
memory usage: 217.1+ KB
```

```
In [67]: main_df.describe()
```

Out[67]:

	rating_numerator	rating_denominator	img_num
count	1964.000000	1964.000000	1964.000000
mean	12.223014	10.479124	1.202138
std	41.708155	6.865424	0.559615
min	0.000000	2.000000	1.000000
25%	10.000000	10.000000	1.000000
50%	11.000000	10.000000	1.000000
75%	12.000000	10.000000	1.000000
max	1776.000000	170.000000	4.000000

1.1.11 12. Standardize dog ratings

```
In [68]: ratings = main_df.text.str.extract('((?:\d+\.)?\d+)\/(\d+)', expand=True)

In [69]: main_df.rating_numerator = ratings
         main_df['rating_numerator'] = main_df['rating_numerator'].astype('float64')

In [70]: # standardizing to a denominator of 10 for groups of dogs:

         rating_num = [int(round(num/(denom/10)))  if denom != 10 and num/denom <= 2
                                    else num for num, denom in zip(main_df['rating_numerator'],
         rating_denom = [10 if denom != 10 and num/denom <= 2
                                     else denom for num, denom in zip(main_df['rating_numerator
         main_df['rating_numerator'] = rating_num
         main_df['rating_denominator'] = rating_denom

         main_df = main_df.drop(main_df[((main_df['rating_denominator'] != 10) | (main_df['ratin

In [71]: main_df['rating_numerator'].unique()

Out[71]: array([ 13.  ,  12.  ,  14.  ,  13.5 ,  11.  ,   6.  ,  10.  ,   0.  ,
                  9.75,   5.  ,  11.27,   3.  ,   7.  ,   8.  ,   9.  ,   4.  ,
                  2.  ,  11.26,   1.  ])

In [72]: main_df['rating_denominator'].unique()

Out[72]: array([10])
```

2 Storing, Analyzing, and Visualizing Data

```
In [73]: # storing main dataframe as csv
         main_df.to_csv('twitter_archive_master.csv', encoding='utf-8', index=False)

In [74]: # read twitter_archive_master.csv
         df1 = pd.read_csv('twitter_archive_master.csv')
         df1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1961 entries, 0 to 1960
Data columns (total 14 columns):
tweet_id              1961 non-null int64
timestamp             1961 non-null object
source                1961 non-null object
text                  1961 non-null object
expanded_urls         1961 non-null object
rating_numerator      1961 non-null float64
rating_denominator    1961 non-null int64
name                  1340 non-null object
stage                 302 non-null object
```

```
jpg_url               1961 non-null object
img_num               1961 non-null int64
pred_breed            1394 non-null object
retweet_count         1961 non-null int64
favorite_count        1961 non-null int64
dtypes: float64(1), int64(5), object(8)
memory usage: 214.6+ KB
```

```
In [75]: df1.describe()
```

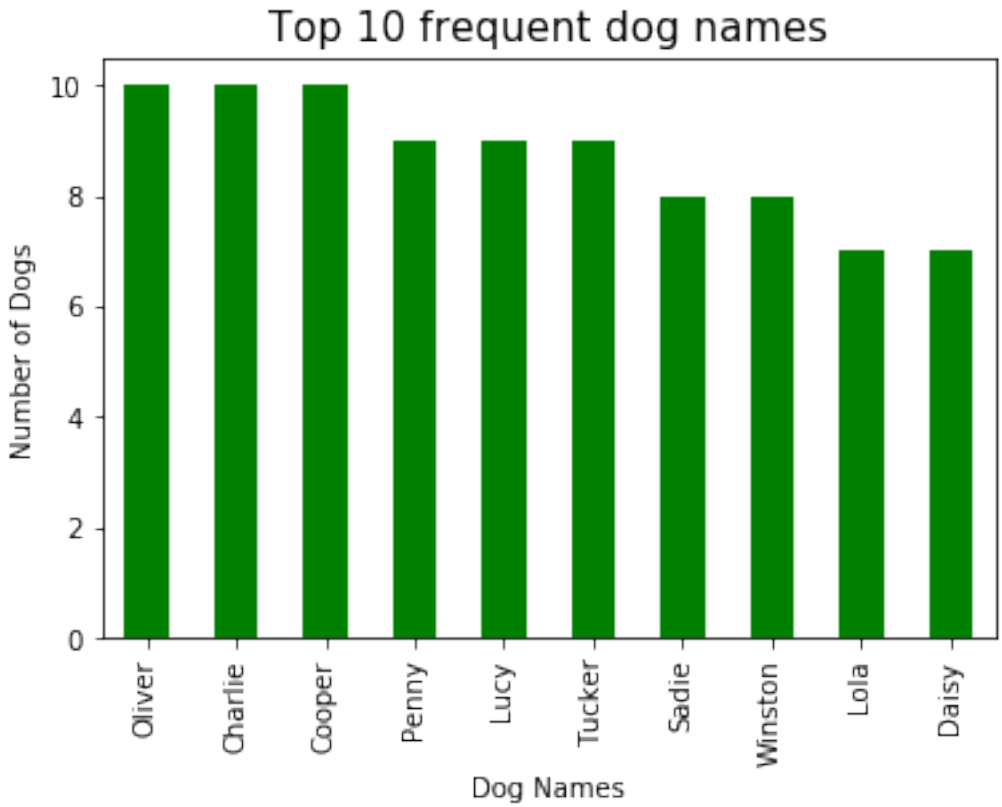
Out[75]:

	tweet_id	rating_numerator	rating_denominator	img_num	\
count	1.961000e+03	1961.000000	1961.0	1961.000000	
mean	7.358030e+17	10.528700	10.0	1.202448	
std	6.745542e+16	2.179024	0.0	0.559987	
min	6.660209e+17	0.000000	10.0	1.000000	
25%	6.758457e+17	10.000000	10.0	1.000000	
50%	7.087111e+17	11.000000	10.0	1.000000	
75%	7.877176e+17	12.000000	10.0	1.000000	
max	8.924206e+17	14.000000	10.0	4.000000	

	retweet_count	favorite_count
count	1961.000000	1961.000000
mean	2385.981642	8108.614482
std	4269.800622	11936.349859
min	11.000000	69.000000
25%	530.000000	1738.000000
50%	1154.000000	3648.000000
75%	2722.000000	10122.000000
max	75085.000000	151947.000000

2.1 What are the 10 most frequent dog names?

```
In [84]: df1['name'].value_counts()[0:10].sort_values(ascending=False).plot(kind = 'bar', color=
         plt.ylabel('Number of Dogs')
         plt.title('Top 10 frequent dog names', size=15)
         plt.xlabel('Dog Names')
         plt.plot();
```



Most of the dogs are of names: OLiver,Charlie, Cooper, Penny, Lucy, Tucker, Sadie, Winston, Lola, Daisy Also, check the count below:

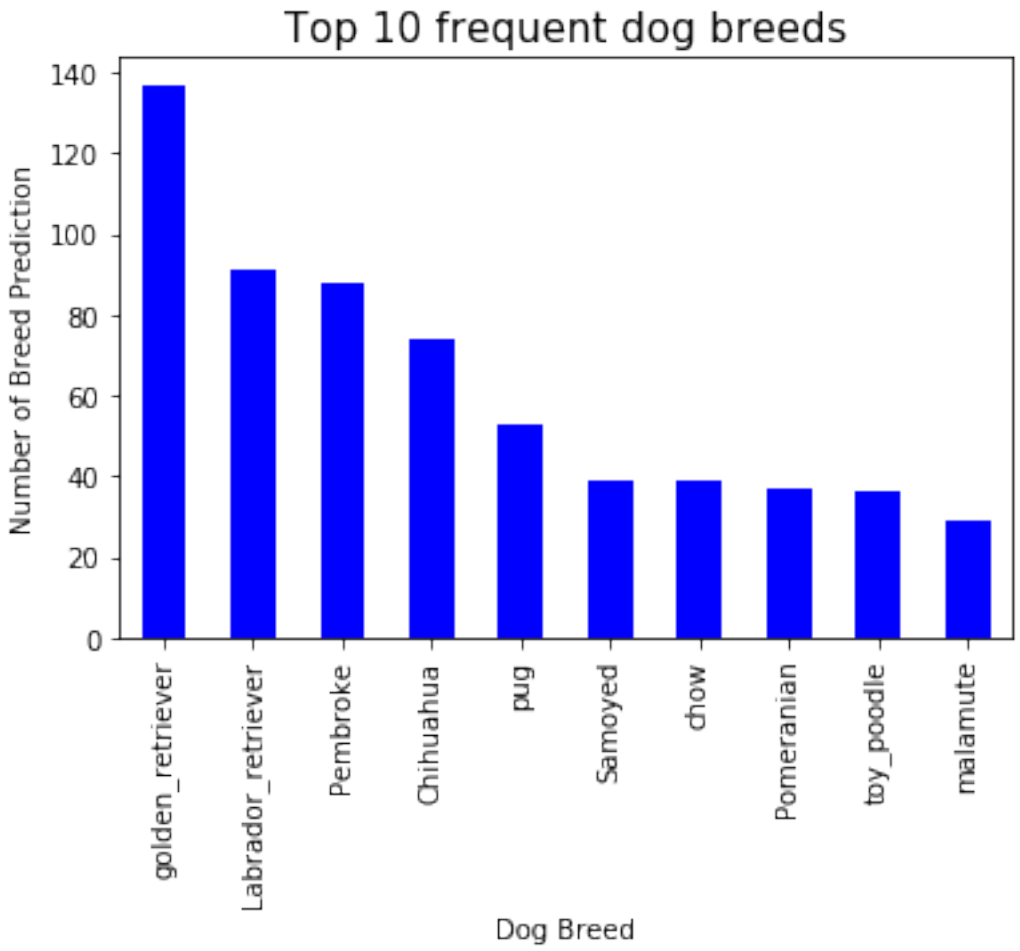
```
In [77]: #Top 10 frequent dog names
         df1['name'].value_counts()[0:10].sort_values(ascending=False)
```

```
Out[77]: Oliver     10
         Charlie    10
         Cooper     10
         Penny       9
         Lucy        9
         Tucker      9
         Sadie       8
         Winston     8
         Lola        7
         Daisy       7
         Name: name, dtype: int64
```

2.2 What are the 10 most frequent predicted dog breeds?

```
In [89]: df1['pred_breed'].value_counts()[0:10].sort_values(ascending=False).plot(kind = 'bar',
         plt.ylabel('Number of Breed Prediction')
```

```
plt.title('Top 10 frequent dog breeds', size=15)
plt.xlabel('Dog Breed')
plt.plot();
```



Most of the dogs have golden retriever, labrador retriever as a breed which all are rated

```
In [79]: #Top 10 frequent dog breeds
         df1['pred_breed'].value_counts()[0:10].sort_values(ascending=False)
```

```
Out[79]: golden_retriever      137
         Labrador_retriever     91
         Pembroke               88
         Chihuahua              74
         pug                    53
         Samoyed                39
         chow                   39
         Pomeranian             37
```

```
toy_poodle              36
malamute                29
Name: pred_breed, dtype: int64
```

3 Findings of the analysis

1) The pred_breed column is created based on the the confidence level of minimum 20% and 'p1_dog', 'p2_dog' and 'p3_dog' statements

2) Based on dog types: doggo, floofer, pupper, puppo, 'doggo, puppo', 'doggo, pupper', 'doggo, floofer', only one categorical column is created named as 'stage'

3) tweet_id is set as object type as it is not going to use for calculation.

4) A main dataframe is created using df_clean, image_df_clean, and tweet_json_clean dataframes

5) Dog Names Issue got rectified

6) Inconsistency in pred_breed got removed

7) All retweets get deleted to get unique tweets

8) The columns such as in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp is removed which is not needed

9) Timestamp format got corrected to datetime format

10) Extra HTML tags from source column get refracted

11) Dog ratings get standardized for denom of 10.

```
In [80]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'wrangle_act.ipynb'])
```

```
Out[80]: 0
```