Udacity Data Analyst Nanodegree

04 Wrangle And Analyze Data Part 1

DAVID LASSIG

02-28-2019

Contents

1 Project 4: Wrangle and Analyze Data Part 1: Gathering, Assessing, Cleaning

1.1 Introduction

Your tasks in this project are as follows:

Data wrangling, which consists of: * Gathering data (downloadable file in the Resources tab in the left most panel of your classroom and linked in step 1 below). * Assessing data * Cleaning data * Storing, analyzing, and visualizing your wrangled data * Reporting on 1) your data wrangling efforts and 2) your data analyses and visualizations

Key points to keep in mind when data wrangling for this project:

- You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
- Assessing and cleaning the entire dataset completely would require a lot of time, and is not necessary to practice and demonstrate your skills in data wrangling. Therefore, the requirements of this project are only to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset.
- Cleaning includes merging individual pieces of data according to the rules of tidy data.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.
- You do not need to gather the tweets beyond August 1st, 2017. You can, but note that you won't be able to gather the image predictions for these tweets since you don't have access to the

algorithm used.

1.2 Environment Preparation

```
import pandas as pd
import numpy as np
import getapi
import json
import tweepy
import requests
import sys
import re
import matplotlib
import matplotlib.pyplot as plt

if sys.version_info[0] < 3:
    from StringIO import StringIO
else:
    from io import StringIO

%matplotlib inline
```

1.3 Data Gathering

1.3.1 Open We Rate Dogs Archive

```
wrd_df = pd.read_csv("twitter-archive-enhanced.csv")
```

1.3.2 Download and process associated twitter stats

For having the **Retweet Counts** and the **Favourite Counts** for each entry in the **twitter-archive-enhanced.csv**. I will download the whole Twitter API stats by using the Tweet ID. As the Twitter API allows only a certain amount of requests per time it will take a while. Moreover I will collect in the variable **missing** the Tweet ID's for which it wasn't possible to retrieve any additional information.

```
file_name="tweet_json.txt"
missing = []

api = getapi.get_twitter_api()

with open(file_name,mode="w") as file:
    for tid in wrd_df['tweet_id']:
        try:
            output = api.get_status(tid)
        except tweepy.TweepError as e:
            print(str(tid)+":"+str(e))
            missing.append(tid)
        file.write(json.dumps(output._json)+"\n")
```

```
Output:

    888202515573088257:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]


    Rate limit reached. Sleeping for: 34


    873697596434513921:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    872668790621863937:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    869988702071779329:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    866816280283807744:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    861769973181624320:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    845459076796616705:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    842892208864923648:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    837012587749474308:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    827228250799742977:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    812747805718642688:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    802247111496568832:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    775096608509886464:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    770743923962707968:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]
    754011816964026368:[{'code': 144, 'message': 'No status found with that
    ↪  ID.'}]


    Rate limit reached. Sleeping for: 679
```

```python
tweet_df = pd.DataFrame()


with open("tweet_json.txt","r") as file:
```

```
    for index,line in enumerate(file):
        output = json.loads(line)
        tweet_df = tweet_df.append(pd.DataFrame.from_dict(output).head(1),sort=True)
file.close()
```

```
tweet_df = tweet_df.reset_index(drop=True)
tweet_df = tweet_df[['id','retweet_count','favorite_count']]
```

1.3.3 Download and process image predictions

```
pred_url = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv"
response = requests.get(pred_url)

image_df = pd.read_csv(StringIO(response.text),sep="\t")
image_df.to_csv('image_predictions.tsv',sep='\t')
```

1.4 Data Assessing

1.4.1 First View

wrd_df

```
wrd_df.head(1)
```

	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
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892420643...	13	10	Phineas	None	None	None	None

```
wrd_df.info()
```

```
Output:

    <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
```

```
wrd_df[wrd_df.doggo != 'None'].iloc[:5]
```

	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
9	890240255349198849	NaN	NaN	2017-07-26 15:59:51 +0000	<a href="http://twitter.com/download/iphone" r...	This is Cassie. She is a college pup. Studying...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/890240255...	14	10	Cassie	doggo	None	None	None
43	884162670584377345	NaN	NaN	2017-07-09 21:29:42 +0000	<a href="http://twitter.com/download/iphone" r...	Meet Yogi. He doesn't have any important dog m...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/884162670...	12	10	Yogi	doggo	None	None	None
99	872967104147763200	NaN	NaN	2017-06-09 00:02:31 +0000	<a href="http://twitter.com/download/iphone" r...	Here's a very large dog. He has a date later. ...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/872967104...	12	10	None	doggo	None	None	None
108	871515927908634625	NaN	NaN	2017-06-04 23:56:03 +0000	<a href="http://twitter.com/download/iphone" r...	This is Napolean. He's a Raggedy East Nicaragu...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/871515927...	12	10	Napolean	doggo	None	None	None
110	871102520638267392	NaN	NaN	2017-06-03 20:33:19 +0000	<a href="http://twitter.com/download/iphone" r...	Never doubt a doggo 14/10 https://t.co/AbBLh2FZCH	NaN	NaN	NaN	https://twitter.com/animalcog/status/871075758...	14	10	None	doggo	None	None	None

```
wrd_df['doggo'].value_counts()
```

```
Output:

    None      2259
    doggo       97
    Name: doggo, dtype: int64
```

```
wrd_df['pupper'].value_counts()
```

```
Output:

    None       2099
    pupper      257
    Name: pupper, dtype: int64
```

```
wrd_df['floofer'].value_counts()
```

```
Output:

    None       2346
    floofer      10
    Name: floofer, dtype: int64
```

```
wrd_df['puppo'].value_counts()
```

```
Output:

    None     2326
    puppo      30
    Name: puppo, dtype: int64
```

```
wrd_df.groupby(['pupper','floofer','puppo']).doggo.value_counts()
```

```
Output:

    pupper  floofer  puppo  doggo
    None    None      None   None     1976
                             doggo      83
                      puppo  None       29
                             doggo       1
            floofer  None   None        9
                             doggo       1
    pupper  None      None   None      245
                             doggo      12
    Name: doggo, dtype: int64
```

```
type(wrd_df['timestamp'][0])
```

```
Output:

    str
```

```
wrd_df['tweet_id'].nunique()
```

```
Output:

    2356
```

```
wrd_df['name'].value_counts()[:5]
```

```
Output:

    None       745
    a           55
    Charlie     12
    Lucy        11
    Cooper      11
    Name: name, dtype: int64
```

```
[row for row in wrd_df['text'] if row.startswith('RT')][:5]
```

```
Output:

    ['RT @dog_rates: This is Canela. She attempted some fancy porch pics. They
    ↪   were unsuccessful. 13/10 someone help her https://t.co/cLyzpcUcMX',
     'RT @Athletics: 12/10 #BATP https://t.co/WxwJmvjfxo',
     'RT @dog_rates: This is Lilly. She just parallel barked. Kindly requests a
     ↪  reward now. 13/10 would pet so well https://t.co/SATN4If5H5',
     'RT @dog_rates: This is Emmy. She was adopted today. Massive round of
     ↪  pupplause for Emmy and her new family. 14/10 for all involved
     ↪  https://...',
     'RT @dog_rates: Meet Shadow. In an attempt to reach maximum zooming
     ↪  borkdrive, he tore his ACL. Still 13/10 tho. Help him out
     ↪  below\n\nhttps:/...']
```

```
wrd_df[wrd_df['rating_numerator'] > 30][:5]
```

	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
188	855862651834028034	8.558616e+17	194351775.0	2017-04-22 19:15:32 +0000	<a href="http://twitter.com/download/iphone" r...	@dhmontgomery We also gave snoop dogg a 420/10...	NaN	NaN	NaN	NaN	420	10	None	None	None	None	None
189	855860136149123072	8.558585e+17	13615722.0	2017-04-22 19:05:32 +0000	<a href="http://twitter.com/download/iphone" r...	@s8n You tried very hard to portray this good ...	NaN	NaN	NaN	NaN	666	10	None	None	None	None	None
290	838150277551247360	8.381455e+17	21955058.0	2017-03-04 22:12:52 +0000	<a href="http://twitter.com/download/iphone" r...	@markhoppus 182/10	NaN	NaN	NaN	NaN	182	10	None	None	None	None	None
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	NaN	NaN	NaN	960	0	None	None	None	None	None
340	832215909146226688	NaN	NaN	2017-02-16 13:11:49 +0000	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: This is Logan, the Chow who liv...	7.867091e+17	4.196984e+09	2016-10-13 23:23:56 +0000	https://twitter.com/dog_rates/status/786709082...	75	10	Logan	None	None	None	None

```
wrd_df[wrd_df['rating_denominator'] > 10][:5]
```

	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
342	832088576586297345	8.320875e+17	30582082.0	2017-02-16 04:45:50 +0000	<a href="http://twitter.com/download/iphone" r...	@docmisterio account started on 11/15/15	NaN	NaN	NaN	NaN	11	15	None	None	None	None	None
433	820690176645140481	NaN	NaN	2017-01-15 17:52:40 +0000	<a href="http://twitter.com/download/iphone" r...	The floofs have been released I repeat the flo...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/820690176...	84	70	None	None	None	None	None
784	775096608509886464	NaN	NaN	2016-09-11 22:20:06 +0000	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: After so many requests, this is...	7.403732e+17	4.196984e+09	2016-06-08 02:41:38 +0000	https://twitter.com/dog_rates/status/740373189...	9	11	None	None	None	None	None
902	758467244762497024	NaN	NaN	2016-07-28 01:00:57 +0000	<a href="http://twitter.com/download/iphone" r...	Why does this never happen at my front door......	NaN	NaN	NaN	https://twitter.com/dog_rates/status/758467244...	165	150	None	None	None	None	None
1068	740373189193256964	NaN	NaN	2016-06-08 02:41:38 +0000	<a href="http://twitter.com/download/iphone" r...	After so many requests, this is Bretagne. She ...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/740373189...	9	11	None	None	None	None	None

tweet_df

```
tweet_df.head(1)
```

	id	retweet_count	favorite_count
0	892420643555336193	8264	37871

```
tweet_df.info()
```

```
Output:

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 2356 entries, 0 to 2355
    Data columns (total 3 columns):
    id                  2356 non-null int64
    retweet_count       2356 non-null int64
    favorite_count      2356 non-null int64
    dtypes: int64(3)
    memory usage: 55.3 KB
```

image_df

```
image_df.head(1)
```

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	True	Shetland_sheepdog	0.061428	True

```
image_df.info()
```

```
Output:

    <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
```

```
type(image_df.p1_conf[0])
```

```
Output:

    numpy.float64
```

```
wrd_df_clean = wrd_df.copy()
tweet_df_clean = tweet_df.copy()
image_df_clean = image_df.copy()
```

1.4.2 Quality Issues

With the overall impression of the assessed data I can identify several quality issues I need to clean for drawing any further conclusions.

WeRateDogs_df (wrd_df)

1. Remove columns that are unneccesary for further analysis from **wrd_df**.
2. Remove columns that have almost only null values from **wrd_df**.
3. Remove rows for which we didn't obtain a twitter status.
4. Convert timestamp in **wrd_df** from string to datetime.
5. Remove names from **name** in **wrd_df** that seems to be unvalid.
6. Remove tweets that are retweeted. That's appearing in form of the string "RT" in beginning of tweet texts.
7. Convert all values in **rating_nominator** and **rating_denominator** to float and correct values that deviates strongly from the margin of values.

image_df

8. Remove columns that are unneccessary for further analysis from **image_df**.
9. Remove second **p2** and third **p3** estimation from dataframe.

tweet_df

10. Rename Column **id** to **tweet_id** for more easier merging.

General Issues

11. Convert **tweet_id** for all dataframes to string object.

1 Define

Remove **source** and **expanded_urls** from **wrd_df**

1 Code

```
wrd_df_clean.drop(columns=['source','expanded_urls'],inplace=True)
```

1 Test

```
wrd_df_clean.head(1)
```

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

```
image_df_clean.head(1)
```

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	True	Shetland_sheepdog	0.061428	True

2 Define

Remove * in_reply_to_status_id * in_reply_to_user_id * retweeted_status_id * retweeted_status_user_id * retweeted_status_time_stamp

from **wrd_df** as it has almost only null values.

2 Code

```
wrd_df_clean.drop(columns=['in_reply_to_status_id',
                           'in_reply_to_user_id',
                           'retweeted_status_id',
                           'retweeted_status_user_id',
                           'retweeted_status_timestamp'],inplace=True)
```

2 Test

```
wrd_df_clean.info()
```

```
Output:

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 2356 entries, 0 to 2355
    Data columns (total 10 columns):
    tweet_id                2356 non-null int64
    timestamp               2356 non-null object
    text                    2356 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: int64(3), object(7)
    memory usage: 184.1+ KB
```

3 Define

Remove the rows for the **tweet_id** we collected in the list **missing**.

3 Code

```
# hardcoded missing values, as if you change the the notebook after kernel restart it would
# take a big amount of time to generate these values again
missing = [888202515573088257,873697596434513921,872668790621863937,869988702071779329,
           866816280283807744,861769973181624320,845459076796616705,842892208864923648,
           837012587749474308,827228250799742977,812747805718642688,802247111496568832,
           775096608509886464,770743923962707968,754011816964026368]



for tweet_id in missing:
    wrd_df_clean.drop(wrd_df_clean[wrd_df_clean['tweet_id'] == tweet_id].index[0],inplace=True)
```

3 Test

```
# if there is removed the right amount of rows, the calculation should result in zero
wrd_df.shape[0] - wrd_df_clean.shape[0] - len(missing)
```

```
Output:

    0
```

4 Define

Convert the **timestamp** column from **wrd_df** to datetime.

4 Code

```
wrd_df_clean['timestamp'] = pd.to_datetime(wrd_df_clean['timestamp'])
```

4 Test

```
wrd_df_clean['timestamp'][1] - wrd_df_clean['timestamp'][0]
```

```
Output:

    Timedelta('-1 days +07:53:31')
```

5 Define

Remove names from **name** in **wrd_df** that seems to be unvalid like “a” and “an”.

5 Code

```
wrd_df_clean['name'] = wrd_df_clean['name'].apply(lambda x: "none" if (x == "a") or (x == "an") else x)
```

5 Test

```
# should be zero if all names that are "a" are removed
(1*(wrd_df_clean['name'] == 'a')).sum()
```

```
Output:

0
```

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

6 Define

Remove tweets that are retweeted. That's appearing in form of the string "RT" in beginning of tweet texts.

6 Code

```
len_df_before = wrd_df_clean.shape[0]
retweet_indexes = [row[0] for row in wrd_df_clean.iterrows() if row[1][2].startswith('RT')]
wrd_df_clean.drop(wrd_df_clean.index[[retweet_indexes]], inplace=True)
```

6 Test

```
# if all columns that includes retweets are removed, the solution should be zero
len_to_drop = len(retweet_indexes)
wrd_df_clean.shape[0] + len_to_drop - len_df_before
```

```
Output:

    0
```

7 Define

This cleaning includes several steps:

- extract and split the ratings from the tweet texts if possible
- check and apply the extracted rating values to the previous dataset values if they are deviate strongly
- **if** there couldn't be extracted a rating and **if** the previous rating deviate strongly
  - I will **drop the whole row from the dataset** when
  - the **rating_numerator > 100** or the **rating_denominator > 100**

7 Code

I guess this code part needs some explanation:

- In this first part I'm compiling a regex for capturing the rating strong from every tweet text
- afterwards follows a function to extract the captured ratings in case of success and **else** applying the old one

```
rating_re = re.compile(" [0-9]{1,2}\/[0-9]{1,2} ")

def apply_rating_num_denum(row):


    if row['ex_rating'] != "none":
        numerator = float(row['ex_rating'].split('/')[0])
        denominator = float(row['ex_rating'].split('/')[1])
        return numerator,denominator
    else:
        return float(row['rating_numerator']),float(row['rating_denominator'])
```

- now I will use the regex to extract the ratings from the tweet texts
- in the second line I'm applying the previously created function to extract the new ratings
- as the result is a **Pandas Series** of tuples I'm seperating them and writing back to **rating_numerator** and **rating_denominator**

```
wrd_df_clean['ex_rating'] = wrd_df_clean['text'].\
                         apply(lambda x: rating_re.search(x).group() if
                                                                      rating_re.search(x) != None
                                                                    else
                                                                      "none")

new_rating =  wrd_df_clean.apply(apply_rating_num_denum, axis=1)

wrd_df_clean['rating_numerator'] = new_rating.apply(lambda x: x[0] if x!= None else "")
wrd_df_clean['rating_denominator'] = new_rating.apply(lambda x: x[1] if x!= None else "")
```

```
wrd_df_clean.drop(wrd_df_clean[wrd_df_clean['rating_numerator'] > 100].index,inplace=True)
wrd_df_clean.drop(wrd_df_clean[wrd_df_clean['rating_denominator'] > 100].index,inplace=True)

wrd_df_clean.drop(columns=['ex_rating'],inplace=True)
```

7 Test

```
# result should be null if everything that deviates strongly is removed
wrd_df_clean[wrd_df_clean['rating_numerator'] > 100].shape[0] +\
wrd_df_clean[wrd_df_clean['rating_denominator'] > 100].shape[0]
```

```
Output:

    0
```

8 Define

Remove **img_num** from **image_df**.

8 Code

```
image_df_clean.drop(columns=['img_num'],inplace=True)
```

8 Test

```
image_df_clean.head(5)
```

	tweet_id	jpg_url	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	Welsh_springer_spaniel	0.465074	True	collie	0.156665	True	Shetland_sheepdog	0.061428	True
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	redbone	0.506826	True	miniature_pinscher	0.074192	True	Rhodesian_ridgeback	0.072010	True
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	German_shepherd	0.596461	True	malinois	0.138584	True	bloodhound	0.116197	True
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	Rhodesian_ridgeback	0.408143	True	redbone	0.360687	True	miniature_pinscher	0.222752	True
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	miniature_pinscher	0.560311	True	Rottweiler	0.243682	True	Doberman	0.154629	True

9 Define

Remove **p2**, **p2_dog**, **p2_conf**, **p3**, **p3_dog** and **p3_conf** from **image_df** as it is enough for our purpose to remain the estimation with the highest confidence.

9 Code

```
image_df_clean.drop(columns=['p2','p2_dog','p2_conf','p3', 'p3_dog','p3_conf'],inplace=True)
```

9 Test

```
image_df_clean.head(1)
```

	tweet_id	jpg_url	p1	p1_conf	p1_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	Welsh_springer_spaniel	0.465074	True

10 Define

Rename Column in **tweet_df** from **id** to **tweet_id**.

10 Code

```
tweet_df_clean.rename(columns={'id':'tweet_id'},inplace=True)
```

10 Test

```
tweet_df_clean.head(1)
```

	tweet_id	retweet_count	favorite_count
0	892420643555336193	8264	37871

11 Define

Convert **tweet_id** from **image_df**, **tweet_df** and **wrd_df** to string.

11 Code

```
wrd_df_clean['tweet_id'] = wrd_df_clean['tweet_id'].astype(str)
image_df_clean['tweet_id'] = image_df_clean['tweet_id'].astype(str)
tweet_df_clean['tweet_id'] = tweet_df_clean['tweet_id'].astype(str)
```

11 Test

```
print("wrd_df_clean tweet_id filetype: {}\n".format(type(wrd_df_clean['tweet_id'][0])) +\
      "image_df_clean tweet_id filetype: {}\n".format(type(image_df_clean['tweet_id'][0])) +\
      "tweet_df_clean tweet_id filetype: {}\n".format(type(tweet_df_clean['tweet_id'][0])))
```

```
Output:

    wrd_df_clean tweet_id filetype: <class 'str'>
    image_df_clean tweet_id filetype: <class 'str'>
    tweet_df_clean tweet_id filetype: <class 'str'>
```

1.4.3 Tidiness Issues

1. Replace four dog type columns into one categorical column in **wrd_df**.
2. Convert **rating_nominator** and **rating_denominator** in **wrd_df** to a single fraction.
3. **retweetCount** and **favouriteCount** should be merged by **tweet_id** from **tweet_df** to **wrd_df**.
4. The columns from **image_df** should be merged by **tweet_id** with **wrd_df** for creating one master dataset.

1 Define

Take string values from **doggo, floofer, pupper and puppo** and put the not **None** values into one primary categorical column **dogtype** and taking the risk to remove a secondary label.

1 Code

```
categories = wrd_df_clean.keys()[-4:].tolist()
categories.append("none")
categories.append("multiple")
categories
```

```
Output:

    ['doggo', 'floofer', 'pupper', 'puppo', 'none', 'multiple']
```

```
wrd_df_clean['dogtype'] = pd.Series(pd.Categorical(values=["none"]*len(wrd_df_clean),categories=categories))


def check_dogtype(df, dogtype, dogtype_string):

    mask = dogtype != "None"
    for index,entry in df[mask].iterrows():
        if df.loc[index,'dogtype'] == "none":
            df.loc[index,'dogtype'] = dogtype_string
        else:
            df.loc[index,'dogtype'] = "multiple"

check_dogtype(wrd_df_clean,wrd_df_clean.doggo,'doggo')
check_dogtype(wrd_df_clean,wrd_df_clean.pupper,'pupper')
check_dogtype(wrd_df_clean,wrd_df_clean.floofer,'floofer')
check_dogtype(wrd_df_clean,wrd_df_clean.puppo,'puppo')


wrd_df_clean.drop(columns=['doggo','floofer','pupper','puppo'],inplace=True)
```

1 Test

```
wrd_df_clean['dogtype'].value_counts()
```

```
Output:

    none        1631
    pupper       227
    doggo         71
    puppo         23
    multiple      11
```

```
floofer        9
Name: dogtype, dtype: int64
```

2 Define

Convert **rating_numerator** and **rating_denominator** in **wrd_df** to a single fraction **rating** and remove them.

2 Code

```
wrd_df_clean['rating'] = wrd_df_clean['rating_numerator'] / wrd_df_clean['rating_denominator']
wrd_df_clean.drop(columns=['rating_numerator','rating_denominator'],inplace=True)
```

2 Test

```
wrd_df_clean.head(1)
```

	tweet_id	timestamp	text	name	dogtype	rating
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	Phineas	none	1.3

3 Define

Merge **wrd_df** with **tweet_df** by using **tweet_id** as the key and remove occuring nan values in merged dataset. Especially for later applications of possible linear regression it's crucial not having **any nan or infinite** values anymore.

3 Code

```
wrd_df_clean = wrd_df_clean.merge(tweet_df_clean,how='outer',left_on='tweet_id',right_on='tweet_id')
wrd_df_clean.dropna(subset = ['timestamp', 'rating','retweet_count'],inplace=True)
```

3 Test

```
wrd_df_clean.head(5)
```

	tweet_id	timestamp	text	name	dogtype	rating	retweet_count_x	favorite_count_x	retweet_count_y	favorite_count_y	retweet_count	favorite_count
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	Phineas	none	1.3	8264.0	37871.0	8264	37871	8264	37871
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	Tilly	none	1.3	6107.0	32540.0	6107	32540	6107	32540
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	Archie	none	1.2	4043.0	24500.0	4043	24500	4043	24500
3	891689557279858688	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	Darla	none	1.3	8411.0	41228.0	8411	41228	8411	41228
4	891327558926688256	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	Franklin	none	1.2	9110.0	39403.0	9110	39403	9110	39403

```
wrd_df_clean.tail(5)
```

	tweet_id	timestamp	text	name	dogtype	rating	retweet_count_x	favorite_count_x	retweet_count_y	favorite_count_y	retweet_count	favorite_count
2298	666049248165822465	2015-11-16 00:24:50	Here we have a 1949 1st generation vulpix. Enj...	None	NaN	0.5	42.0	106.0	42	106	42	106
2299	666044226329800704	2015-11-16 00:04:52	This is a purebred Piers Morgan. Loves to Netf...	a	NaN	0.6	136.0	292.0	136	292	136	292
2300	666033412701032449	2015-11-15 23:21:54	Here is a very happy pup. Big fan of well-main...	a	NaN	0.9	43.0	123.0	43	123	43	123
2301	666029285002620928	2015-11-15 23:05:30	This is a western brown Mitsubishi terrier. Up...	a	NaN	0.7	46.0	125.0	46	125	46	125
2302	666020888022790149	2015-11-15 22:32:08	Here we have a Japanese Irish Setter. Lost eye...	None	NaN	0.8	498.0	2529.0	498	2529	498	2529

4 Define

4 Code

```
twitter_archive_master = wrd_df_clean.merge(image_df_clean,how='outer',left_on='tweet_id',right_on='tweet_id')
twitter_archive_master.dropna(subset = ['timestamp', 'rating'],inplace=True)
```

4 Test

```
twitter_archive_master.head(5)
```

	tweet_id	timestamp	text	name	dogtype	rating	retweet_count_x	favorite_count_x	retweet_count_y	favorite_count_y	retweet_count	favorite_count	jpg_url	p1	p1_conf	p1_dog
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	Phineas	none	1.3	8264.0	37871.0	8264.0	37871.0	8264.0	37871.0	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	orange	0.097049	False
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	Tilly	none	1.3	6107.0	32540.0	6107.0	32540.0	6107.0	32540.0	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	Chihuahua	0.323581	True
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	Archie	none	1.2	4043.0	24500.0	4043.0	24500.0	4043.0	24500.0	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	Chihuahua	0.716012	True
3	891689557279858688	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	Darla	none	1.3	8411.0	41228.0	8411.0	41228.0	8411.0	41228.0	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	paper_towel	0.170278	False
4	891327558926688256	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	Franklin	none	1.2	9110.0	39403.0	9110.0	39403.0	9110.0	39403.0	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg	basset	0.555712	True

```
twitter_archive_master.tail(5)
```

	tweet_id	timestamp	text	name	dogtype	rating	retweet_count_x	favorite_count_x	retweet_count_y	favorite_count_y	retweet_count	favorite_count	jpg_url	p1	p1_conf	p1_dog
2298	666049248165822465	2015-11-16 00:24:50	Here we have a 1949 1st generation vulpix. Enj...	None	NaN	0.5	42.0	106.0	42.0	106.0	42.0	106.0	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	miniature_pinscher	0.560311	True
2299	666044226329800704	2015-11-16 00:04:52	This is a purebred Piers Morgan. Loves to Netf...	a	NaN	0.6	136.0	292.0	136.0	292.0	136.0	292.0	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	Rhodesian_ridgeback	0.408143	True
2300	666033412701032449	2015-11-15 23:21:54	Here is a very happy pup. Big fan of well-main...	a	NaN	0.9	43.0	123.0	43.0	123.0	43.0	123.0	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	German_shepherd	0.596461	True
2301	666029285002620928	2015-11-15 23:05:30	This is a western brown Mitsubishi terrier. Up...	a	NaN	0.7	46.0	125.0	46.0	125.0	46.0	125.0	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	redbone	0.506826	True
2302	666020888022790149	2015-11-15 22:32:08	Here we have a Japanese Irish Setter. Lost eye...	None	NaN	0.8	498.0	2529.0	498.0	2529.0	498.0	2529.0	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	Welsh_springer_spaniel	0.465074	True

```
# should be zero if all nan and infinite retweets are removed
(~np.isfinite(twitter_archive_master['retweet_count'])).sum()
```

```
Output:

0
```

1.4.4 Write finished dataframes to csv

```
twitter_archive_master.to_csv('twitter_archive_master.csv',sep=',',index=False)
image_df_clean.to_csv('twitter_image_prediction.csv',sep=',',index=False)
```