Wrangle report

This report is for wrangle processes that I made in the project; wrangle process is divided into 3 processes:

- **Gathering**
- **Accessing**
- **Cleaning**

Those are the packages which are used in the project:

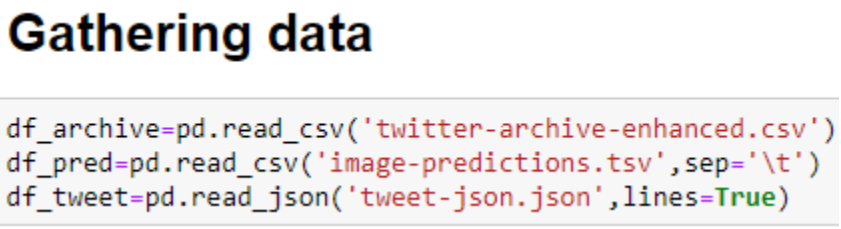
```
import pandas as pd
import numpy as np
import requests as rq
import tweepy
import json
```

```
import seaborn as sns
```

```
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
```

The three wrangle processes in details:

1) **Gathering:**



I gathered three different types of file which are (twitter archive enhanced) which is a .csv file, (image predications) which is .tsv file and (tweet json) which is .txt file and I converted to .json file for ease of reading the file.

2) **Accessing:**
   I used functions like head(), info(),describe(),value_counts() and count() to illustrate the three data frames and information about them to show issue that need to be cleaned

```
df_archive.head(5)
```

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id
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
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	NaN
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	NaN
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	NaN
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	NaN

```
df_pred.describe()
```

	tweet_id	img_num	p1_conf	p2_conf	p3_conf
count	2.075000e+03	2075.000000	2075.000000	2.075000e+03	2.075000e+03
mean	7.384514e+17	1.203855	0.594548	1.345886e-01	6.032417e-02
std	6.785203e+16	0.561875	0.271174	1.006657e-01	5.090593e-02
min	6.660209e+17	1.000000	0.044333	1.011300e-08	1.740170e-10
25%	6.764835e+17	1.000000	0.364412	5.388625e-02	1.622240e-02
50%	7.119988e+17	1.000000	0.588230	1.181810e-01	4.944380e-02
75%	7.932034e+17	1.000000	0.843855	1.955655e-01	9.180755e-02
max	8.924206e+17	4.000000	1.000000	4.880140e-01	2.734190e-01

```
df_archive.count
```

```
<bound method DataFrame.count of                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
0      892420643555336193                    NaN                  NaN
1      892177421306343426                    NaN                  NaN
2      891815181378084864                    NaN                  NaN
3      891689557279858688                    NaN                  NaN
4      891327558926688256                    NaN                  NaN
5      891087950875897856                    NaN                  NaN
6      890971913173991426                    NaN                  NaN
7      890729181411237888                    NaN                  NaN
8      890609185150312448                    NaN                  NaN
9      890240255349198849                    NaN                  NaN
10     890006608113172480                    NaN                  NaN
11     889880896479866881                    NaN                  NaN
12     889665388333682689                    NaN                  NaN
13     889638837579907072                    NaN                  NaN
14     889531135344209921                    NaN                  NaN
15     889278841981685760                    NaN                  NaN
16     888917238123831296                    NaN                  NaN
17     888804989199671297                    NaN                  NaN
```

I detected some issues need to be cleaned which are classified into two parts:

- **Quality**
- **Tidiness**

1)quality:

in tweeter archive data frame:

a) drop unused columns
b) drops useless rows
c) change timestamp to date_time type
d) change id to string type
e) rating_numerator and rating dominator to be float type
f) in dog stages columns there is 'none' values

in image predication data frame:

a) after tidiness of this data frame, there're names in prediction column in lowercase
b) change id to string type

in tweet json data frame:

a) after tidiness drop unused columns
b) change id to string type
c) drop unused rows

2)tidiness:

a) merge df_archive and df_tweet
b) col. names in df_pred

c) id column name in df_tweet should be repalced by tweet_id

d) 4 dog_stages to be in one column

3) Cleaning:

Clean

```
#make copy for each dataframe
df_clean1= df_archive.copy()
df_clean2= df_pred.copy()
df_clean3= df_tweet.copy()
```

In cleaning process first of all I made a copy for each data frames for ease of cleaning and compare between the original data to the cleaned one, note: there are some quality issues are cleaned after tidiness part.

Cleaning process is consisting of three parts:

a) Define

b) Code

c) Test

Each clean issue will be illustrated in sequence of those parts:

in tweeter archive data frame:

a) **drop unused columns**
define:
drop unused columns in df_1 using drop()
code:
code

```
df_clean1.drop(['in_reply_to_status_id','in_reply_to_user_id','retweeted_status_id','retweeted_status_user_id'
                ,'retweeted_status_timestamp','source'],axis=1,inplace=True)
```

test:
test

```
df_clean1
```

	tweet_id	timestamp	text	expanded_urls	rating_numerator	rating_denominator
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	https://twitter.com/dog_rates/status/892420643...	13	10
1	892177421306343426	2017-08-01 00:17:27 +0000	This is Tilly. She's just checking pup on you....	https://twitter.com/dog_rates/status/892177421...	13	10
2	891815181378084864	2017-07-31 00:18:03 +0000	This is Archie. He is a rare Norwegian Pouncin...	https://twitter.com/dog_rates/status/891815181...	12	10
3	891689557279858688	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal	https://twitter.com/dog_rates/status/891689557...	13	10

b) **drops useless rows**
define:
drop unused rows (i.e. only rows which don't have image)
code:

code

```
cond= df_clean1['tweet_id'].isin(df_clean2['tweet_id'])
df_clean1.drop(df_clean1[~cond].index,inplace=True)
```

```
df_clean1.reset_index(drop=True)
```

	tweet_id	timestamp	text	expanded_urls	rating_numerator	r
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	https://twitter.com/dog_rates/status/892420643...	13	
1	892177421306343426	2017-08-01 00:17:27 +0000	This is Tilly. She's just checking pup on you....	https://twitter.com/dog_rates/status/892177421...	13	
2	891815181378084864	2017-07-31 00:18:03 +0000	This is Archie. He is a rare Norwegian Pouncin...	https://twitter.com/dog_rates/status/891815181...	12	
3	891689557279858688	2017-07-30 15:58:51 +0000	This is Darla. She commenced a snooze mid meal...	https://twitter.com/dog_rates/status/891689557...	13	

test:

test

```
#the count of tweet id in the tweeter archive shoul be equal to e count of tweet id in the image prediction
df_clean1.tweet_id.count()
```

```
2075
```

```
df_clean2.tweet_id.count()
```

```
2075
```

c) **change timestamp to date_time type**

define:

change timestamp to date_time type

code:

code

```
df_clean1['timestamp']=pd.to_datetime(df_clean1['timestamp'])
```

test:

test

```
df_clean1.timestamp.dtype
```

```
dtype('<M8[ns]')
```

d) **change id to string type**

define:

change id to string type

code:

code

```
df_clean1['tweet_id']=df_clean1.tweet_id.astype(str)
```

test:

test

```
df_clean1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2075 entries, 0 to 2355
Data columns (total 11 columns):
tweet_id              2075 non-null object
```

e) **rating_numerator and rating dominator to be float type**

define:

rating_numerator and rating dominator to be float type

code:

code

```
df_clean1['rating_numerator']=df_clean1.rating_numerator.astype(float)
df_clean1['rating_denominator']=df_clean1.rating_denominator.astype(float)
```

test:

test

```
df_clean1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2075 entries, 0 to 2355
Data columns (total 11 columns):
tweet_id              2075 non-null object
timestamp             2075 non-null datetime
text                  2075 non-null object
expanded_urls         2075 non-null object
rating_numerator      2075 non-null float64
rating_denominator    2075 non-null float64
```

f) **in dog stages columns there is 'none' values**

define:

in dog stages columns there is 'none' values

code:

code

```
df_clean1['doggo']=df_clean1.doggo.str.replace('None','')
df_clean1['pupper']=df_clean1.pupper.str.replace('None','')
df_clean1['puppo']=df_clean1.puppo.str.replace('None','')
df_clean1['floofer']=df_clean1.floofer.str.replace('None','')
```

test:

test

```
df_clean1.doggo.head(5)
```

```
0
1
2
3
4
Name: doggo, dtype: object
```

```
df_clean1.pupper.head(5)
```

```
0
1
2
3
4
Name: pupper, dtype: object
```

```
df_clean1.puppo.head(5)
```

```
0
1
2
3
4
Name: puppo, dtype: object
```

```
df_clean1.floofer.head(5)
```

```
0
1
2
3
4
Name: floofer, dtype: object
```

In image predication data frame:

a) **after tidiness of this data frame, there're names in prediction column in lowercase**
define:
there're names in prediction column in lowercase
code:
code

```
df_clean2['prediction']=df_clean2.prediction.str.title()
```

test:

test

```
df_clean2.head(3)
```

tweet_id	jpg_url	img_num	prediction_level	index	prediction	confidence	breed
			1	0	Welsh_Springer_Spaniel	0.465074	True
666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	2	0	Collie	0.156665	True
			3	0	Shetland_Sheepdog	0.061428	True

b) **change id to string type**
define:
change id to string type
code:

code

```
: df_clean2['tweet_id']=df_clean2.tweet_id.astype(str)
```

test:

test

```
df_clean2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null object
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(1), object(5)
memory usage: 152.1+ KB
```

In tweet json data frame:

a) **after tidiness drop unused columns**
define:
drop unused columns
code:

```
list(df_clean3)
```

```
['contributors',
 'coordinates',
 'created_at',
 'display_text_range',
 'entities',
 'extended_entities',
 'favorite_count',
 'favorited',
 'full_text',
 'geo',
 'id',
 'id_str',
 'in_reply_to_screen_name'
 'in_reply_to_status_id',
 'in_reply_to_status_id_st
 'in_reply_to_user_id',
 'in_reply_to_user_id_str'
 'is_quote_status',
 'lang',
 'place',
 'possibly_sensitive',
 'possibly_sensitive_appea
 'quoted_status',
 'quoted_status_id',
 'quoted_status_id_str',
 'retweet_count',
 'retweeted',
 'retweeted_status',
 'source',
 'truncated',
```

```
df_clean3.drop(['in_reply_to_screen_name','in_reply_to_status_id','in_reply_to_status_id_str',
                'in_reply_to_user_id','in_reply_to_user_id_str','is_quote_status','lang','place','retweeted',
                'retweeted_status','contributors', 'coordinates','created_at','display_text_range','entities',
                'extended_entities','favorited','full_text','geo','truncated','user','possibly_sensitive'
                ,'possibly_sensitive_appealable','quoted_status','quoted_status_id',
                'quoted_status_id_str','id_str'],axis=1,inplace=True)
```

test:

test

```
df_clean3.head(5)
```

	favorite_count	id	retweet_count	source
0	39467	892420643555336193	8853	<a href="http://twitter.com/download/iphone" r...
1	33819	892177421306343426	6514	<a href="http://twitter.com/download/iphone" r...
2	25461	891815181378084864	4328	<a href="http://twitter.com/download/iphone" r...
3	42908	891689557279858688	8964	<a href="http://twitter.com/download/iphone" r...
4	41048	891327558926688256	9774	<a href="http://twitter.com/download/iphone" r...

b) **change id to string type**
define:
change id to string type
code:
code

```
df_clean3['id']=df_clean3.id.astype(str)
```

test:
test

```
df_clean3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 4 columns):
favorite_count    2354 non-null int64
id                2354 non-null object
retweet_count     2354 non-null int64
source            2354 non-null object
dtypes: int64(2), object(2)
memory usage: 73.6+ KB
```

c) **drop unused rows**
define:
drop unused rows (i.e. only rows which don't have image)
code:
code

```
cond1= df_clean3['tweet_id'].isin(df_clean1['tweet_id'])
df_clean3.drop(df_clean3[~cond1].index,inplace=True)
df_clean3.reset_index(drop=True)
```

test:

test

```
(df_clean3['tweet_id'].shape,df_clean1['tweet_id'].shape)
```

```
((2073,), (2075,))
```

tidiness:

a) **col. names in df_pred:**
define:
df_clean2 need to be reshaped
code:
code

```
# Renaming the dataset columns
cols = ['tweet_id', 'jpg_url', 'img_num',
       'prediction_1', 'confidence_1', 'breed_1',
       'prediction_2', 'confidence_2', 'breed_2',
       'prediction_3', 'confidence_3', 'breed_3']
df_clean2.columns = cols
```

```
# Reshaping the dataframe
df_clean2 =pd.wide_to_long(df_clean2.reset_index(0), stubnames=['prediction', 'confidence', 'breed'],
    i=['tweet_id', 'jpg_url', 'img_num'], j='prediction_level', sep="_")
```

Test:
test

```
df_clean2
```

tweet_id	jpg_url	img_num	prediction_level	index	prediction	confidence	breed
			1	0	Welsh_springer_spaniel	0.465074	True
666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	2	0	collie	0.156665	True
			3	0	Shetland_sheepdog	0.061428	True
			1	1	redbone	0.506826	True
666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	2	1	miniature_pinscher	0.074192	True
			3	1	Rhodesian_ridgeback	0.072010	True
			1	2	German_shepherd	0.596461	True
666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	2	2	malinois	0.138584	True
			3	2	bloodhound	0.116197	True
			1	3	Rhodesian_ridgeback	0.408143	True
666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	2	3	redbone	0.360687	True
			3	3	miniature_pinscher	0.222752	True
			1	4	miniature_pinscher	0.560311	True

b) **id column name in df_tweet should be replaced by tweet_id**
define:
id column name in df_tweet should be replaced by tweet_id
code:

code

```
df_clean3=df_clean3.rename(columns={'id':'tweet_id'})
```

test:

test

```
df_clean3
```

	favorite_count	tweet_id	retweet_count	source
0	39467	892420643555336193	8853	<a href="http://twitter.com/download/iphone" r...
1	33819	892177421306343426	6514	<a href="http://twitter.com/download/iphone" r...
2	25461	891815181378084864	4328	<a href="http://twitter.com/download/iphone" r...
3	42908	891689557279858688	8964	<a href="http://twitter.com/download/iphone" r...
4	41048	891327558926688256	9774	<a href="http://twitter.com/download/iphone" r...
5	20562	891087950875897856	3261	<a href="http://twitter.com/download/iphone" r...
6	12041	890971913173991426	2158	<a href="http://twitter.com/download/iphone" r...
7	56848	890729181411237888	16716	<a href="http://twitter.com/download/iphone" r...
8	28226	890609185150312448	4429	<a href="http://twitter.com/download/iphone" r...
9	32467	890240255349198849	7711	<a href="http://twitter.com/download/iphone" r...
10	31166	890006608113172480	7624	<a href="http://twitter.com/download/iphone" r...
11	28268	889880896479866881	5156	<a href="http://twitter.com/download/iphone" r...
12	38818	889665388333682689	8538	<a href="http://twitter.com/download/iphone" r...
13	27672	889638837579907072	4735	<a href="http://twitter.com/download/iphone" r...

c) **4 dog_stages to be in one column**

define:

4 dog_stages to be in one column

code:

code

```
df_clean1['stages_of_dogs'] = df_clean1.floofer + df_clean1.doggo+ df_clean1.pupper + df_clean1.puppo
df_clean1.drop(['doggo','pupper','puppo','floofer','name'],axis=1,inplace=True)
```

```
df_clean1.loc[df_clean1.stages_of_dogs == 'doggopupper', 'stages_of_dogs'] = 'doggo-pupper'
df_clean1.loc[df_clean1.stages_of_dogs == 'doggopuppo', 'stages_of_dogs'] = 'doggo-puppo'
df_clean1.loc[df_clean1.stages_of_dogs == 'doggofloofer', 'stages_of_dogs'] = 'doggo-floofer'
```

```
df_clean1['stages_of_dogs']=df_clean1.stages_of_dogs.replace('',np.nan)
```

test:

test

```
df_clean1.stages_of_dogs.value_counts()
```

```
pupper          211
doggo            67
puppo            23
doggo-pupper     11
floofer           7
flooferdoggo      1
doggo-puppo       1
Name: stages_of_dogs, dtype: int64
```

d) **merge df_archive and df_tweet**

define:

merge the data frames of archive and tweet json to be one data frame

code:

code

```
df_mrge = pd.merge(df_clean1, df_clean3, how='left', left_on='tweet_id',right_on='tweet_id')
```

test:

```
df_mrge.head(5)
```

	tweet_id	timestamp	text	expanded_urls	rating_numerator	rating_denominator	stages_of_dogs	favorite_cou
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	https://twitter.com/dog_rates/status/892420643...	13.0	10.0	NaN	3946
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	https://twitter.com/dog_rates/status/892177421...	13.0	10.0	NaN	3381
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	https://twitter.com/dog_rates/status/891815181...	12.0	10.0	NaN	2546

After wrangling we save the data frames to a new .csv file (twitter archive master) and another file for image predication data frame (image predication) for analyze the cleaned data.

save cleaning data

```
df_mrge=df_mrge.to_csv('twitter_archive_master.csv',index=False)
df_clean2=df_clean2.to_csv('image_predication.csv',index=False)
```