Data Wrangling Report (Second Project)

Data Gathering:

1- Twitter-archive-enhanced.csv file, this file was delivered by email and I downloaded it manually then I imported it into the working environment using **pandas**.

```
df = pd.read_csv('twitter-archive-enhanced.csv')
```

2- Image-predictions.tsv file, this file was delivered by email and I downloaded it manually then I imported it into the working environment using **requests**

```
url =
'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'
dfimage = 'image-predictions.tsv'
response = requests.get(url)
if not os.path.isfile(dfimage):
  with open(dfimage, 'wb') as f:
    f.write(response.content)
```

3- Tweet_json.txt the final file, this file was delivered by email and I downloaded it manually then I imported it into the working environment using **json** (because I did not get the developer tweeter account yet).

```
df_list = []
with open('tweet_json.txt', 'r') as f:
  for line in f:
    tweet = json.loads(line)
    tweet_id = tweet['id']
    retweet_count = tweet['retweet_count']
```

```
    favorite_count = tweet['favorite_count']
    user_count = tweet['user']['followers_count']
    df_list.append({'tweet_id':tweet_id,
                    'retweet_count': retweet_count,
                    'favorite_count': favorite_count ,
                    'user_count': user_count})

dfApi = pd.DataFrame(df_list)
dfApi.head()
```

Data Assessment:

1- The visual assessment done on the Windows Excel sheet.
2- The programmatically assessment done on Jupyter notebook by pandas functions such **.info () .head ()** etc...
3- first I addressed inconsistence and missing data .
   -separating between concatenated values in dog_stage column in archive table by '_'.
   -fixing different entries like None and NaN values in archive table (inconsistence issue).
   -dropping tweets that have no images (validity issue).
   -getting the real tweets from dfimage table (Accuracy issue).
   -replacing NaN values in name column in archive table to Dog (completeness issue).
   -checking rating_denominator values in archive table (Accuracy issue).
   -checking rating_numerator values in archive table (Accuracy issue).
   -replacing columns names in dfimage1 table (Accuracy issue).
   -fixing some incorrect value (like a value) in name column in archive table (validity issue).
   -dropping duplicates in twitter_archive_master table.

4- then I addressed tidiness issues

- dropping doggo, floofer, pupper and puppo columns and adding a new column that called dog_stage.
- merging all the data frames that called twitter_archive_master.

Data Cleaning:

*-cleaning inconsistence and missing data issues:

1-Define: separating between concatenated values in dog_stage column in archive table by '_'.

Code:**archive.loc[archive['dog_stage']=='doggopupper','dog_stage']='doggo_pupper'**

archive.loc[archive['dog_stage']=='doggofloofer','dog_stage']='doggo_floofer'

archive.loc[archive['dog_stage']=='doggopuppo','dog_stage']='doggo_puppo'

test: **archive['dog_stage'].value_counts()**

2-Define: fixing different entries like None and NaN values in archive table (inconsistence issue).

-Code: **archive['dog_stage'].replace('','NaN')**

-test:**archive.info()**

3-Define: dropping tweets that have no images (validity issue).

-Code:

tweets_with_image = list(dfimage.tweet_id.unique())

len(tweets_with_image) ==
archive.tweet_id.isin(tweets_with_image).sum()

archive = archive[archive.tweet_id.isin(tweets_with_image)]

retweet_entries = archive.retweeted_status_id.notnull()

archive[retweet_entries].shape[0]

archive = archive[~retweet_entries]

np.logical_not(dfimage.tweet_id.isin(list(archive.tweet_id)))

dfimage[~np.logical_not(dfimage.tweet_id.isin(list(archive.tweet_id)))]

-test: **archive.info()**

4-Define: getting the real tweets from dfimage table (Accuracy issue)

_code:

tweets_with_image = list(dfimage.tweet_id.unique())

len(tweets_with_image) ==
archive.tweet_id.isin(tweets_with_image).sum()

archive = archive[archive.tweet_id.isin(tweets_with_image)]

retweet_entries = archive.retweeted_status_id.notnull()

archive[retweet_entries].shape[0]

archive = archive[~retweet_entries]

np.logical_not(dfimage.tweet_id.isin(list(archive.tweet_id)))

dfimage[~np.logical_not(dfimage.tweet_id.isin(list(archive.tweet_id)))]

-test: **archive.info()**

5-Define: replacing NaN values in name column in archive table to Dog (completeness issue).

-Code: **archive['name']=archive['name'].replace('None','Dog')**

-test: : **archive['name'].value_counts()**

6-Define: checking rating_denominator values in archive table (Accuracy issue).

-Code:
archive.loc[archive['tweet_id']==832088576586297345,'rating_denominator']=15

archive.loc[archive['tweet_id']==820690176645140481,'rating_denominator']=10

archive.loc[archive['tweet_id']==810984652412424192,'rating_denominator']=7

archive.loc[archive['tweet_id']==775096608509886464,'rating_denominator']=11

archive.loc[archive['tweet_id']==758467244762497024,'rating_denominator']=10

archive.loc[archive['tweet_id']==740373189193256964,'rating_denominator']=11

archive.loc[archive['tweet_id']==731156023742988288,'rating_denominator']=11

archive.loc[archive['tweet_id']==722974582966214656,'rating_denominator']=10

archive.loc[archive['tweet_id']==716439118184652801,'rating_denominator']=10

archive.loc[archive['tweet_id']==713900603437621249,'rating_denominator']=12

archive.loc[archive['tweet_id']==710658690886586372,'rating_denominator']=10

archive.loc[archive['tweet_id']==709198395643068416,'rating_denominator']=10

archive.loc[archive['tweet_id']==704054845121142784,'rating_denominator']=10

archive.loc[archive['tweet_id']==697463031882764288,'rating_denominator']=10

archive.loc[archive['tweet_id']==684222868335505415,'rating_denominator']=10

archive.loc[archive['tweet_id']==682962037429899265,'rating_denominator']=11

archive.loc[archive['tweet_id']==677716515794329600,'rating_denominator']=10

archive.loc[archive['tweet_id']==675853064436391936,'rating_denominator']=10

archive.loc[archive['tweet_id']==666287406224695296,'rating_denominator']=10

archive.loc[archive['tweet_id']==810984652412424192,'rating_denominator']=7

archive.loc[archive['tweet_id']==740373189193256964,'rating_denominator']=11

archive.loc[archive['tweet_id']==731156023742988288,'rating_denominator']=11

archive.loc[archive['tweet_id']==713900603437621249,'rating_denominator']=12

archive.loc[archive['tweet_id']==682962037429899265,'rating_denominator']=11

-test: **(archive[archive['rating_denominator']!=10]).head(23)**

7-Define: checking rating_numerator values in archive table (Accuracy issue).

-Code:

(archive['rating_numerator']>15).sum()

(archive['rating_numerator']<6).sum()

archive.loc[archive['tweet_id']==832215909146226688,'rating_numerator']=10

archive.loc[archive['tweet_id']==820690176645140481,'rating_numerator']=14

archive.loc[archive['tweet_id']==810984652412424192,'rating_numerator']=14

archive.loc[archive['tweet_id']==778027034220126208,'rating_numerator']=12

archive.loc[archive['tweet_id']==758467244762497024,'rating_numerator']=15

archive.loc[archive['tweet_id']==749981277374128128,'rating_numerator']=10

archive.loc[archive['tweet_id']==731156023742988288,'rating_numerator']=12

archive.loc[archive['tweet_id']==716439118184652801,'rating_numerator']=11

archive.loc[archive['tweet_id']==713900603437621249,'rating_numerator']=11

archive.loc[archive['tweet_id']==710658690886586372,'rating_numerator']=10

-test: **(archive['rating_numerator']>15).sum()**

(archive['rating_numerator']<6).sum()

8-Define: replacing columns names in dfimage1 table (Accuracy issue).

-Code:

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']

dfimage1.columns = cols

-test: **dfimage1.head()**

9-Define: fixing some incorrect value (like a value) in name column in archive table (validity issue).

-Code:

archive.loc[archive['tweet_id']==885518971528720385,'name']='Howard'

archive.loc[archive['tweet_id']==666781792255496192,'name']='Octavia th'

archive.loc[archive['tweet_id']==671743150407421952,'name']='Jacob'

archive.loc[archive['tweet_id']==671147085991960577,'name']='Rufus'

archive.loc[archive['tweet_id']==670427002554466305,'name']='Spork'

archive.loc[archive['tweet_id']==670361874861563904,'name']='Cherokee'

archive.loc[archive['tweet_id']==670303360680108032,'name']='Hemry'

archive.loc[archive['tweet_id']==669923323644657664,'name']='Alphred'

archive.loc[archive['tweet_id']==669564461267722241,'name']='Alfredo'

archive.loc[archive['tweet_id']==668955713004314625,'name']='Leroi'

archive.loc[archive['tweet_id']==668507509523615744,'name']='Chuk'

archive.loc[archive['tweet_id']==668171859951755264,'name']='Alfonso'

archive.loc[archive['tweet_id']==667861340749471744,'name']='Cheryl'

archive.loc[archive['tweet_id']==667773195014021121,'name']='Jessiga'

archive.loc[archive['tweet_id']==667538891197542400,'name']='Klint'

archive.loc[archive['tweet_id']==667470559035432960,'name']='Kohl'

archive.loc[archive['tweet_id']==666983947667116034,'name']='Pepe'

archive.loc[archive['tweet_id']==666781792255496192,'name']='Octaviath'

archive.loc[archive['tweet_id']==666701168228331520,'name']='Johm'

-test: **archive['name'].value_counts()**

10-Define: -dropping duplicates in twitter_archive_master table.

-Code:
twitter_archive_master.drop_duplicates(subset=['tweet_id'],keep=False,inplace=True)

-test: **twitter_archive_master['tweet_id'].duplicated().sum()**

*-cleaning of tidiness issues:

1-Define: dropping doggo, floofer, pupper and puppo columns.

Code**: archive['doggo']=archive['doggo'].replace('None','')**

archive['floofer']=archive['floofer'].replace('None','')

archive['pupper']=archive['pupper'].replace('None','')

archive['puppo']=archive['puppo'].replace('None','')

archive.pop('doggo')

archive.pop(' floofer ')

archive.pop('pupper')

archive.pop('puppo')

: **archive['dog_stage'] = archive.doggo + archive.floofer +**

archive.pupper + archive.puppo

test: **archive.info()**

2-Define: merging all the data frames that called twitter_archive_master.

-Code **: datafarames=[archive,dfimage,dfApi]**

twitter_archive_master= pd.concat(datafarames)

-Test: twitter_archive_master.head()