M.Gameel
DAND student

Wrangle Report

First file:

Gathering:

The first file was on hand file I downloaded manually ('twitter-archive-enhanced.csv').

Assessing:

I assessed the data visually to discover some issues of quality and tidiness that are:

For tidiness: the four stages of dogs are stored in 4 columns the should be under one variable named dog stage.

doggo	floofer	pupper	puppo
None	None	None	None
None	None	None	None
None	None	None	None
None	None	None	None

For quality:

- Through visual assessment I noticed that the data contain retweets and replies. our main concern here is actual tweets with ratings and pictures

I used a programmatic approach to detect quality issues. I designed a simple for loop to display all the variables value counts.

```
for col in list(twitter_enhanced):
  print (col)
  print(twitter_enhanced[col].value_counts())
```

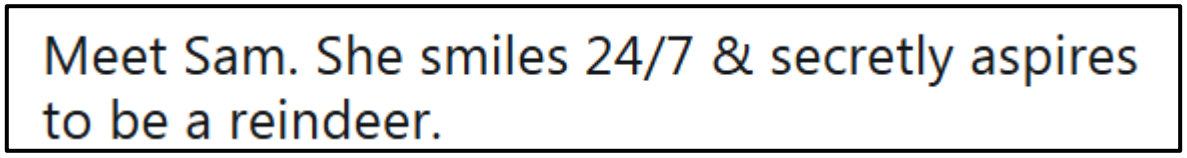
going through the value counts I discovered various issues:

- The name of dog variables contains observations like 'a' and 'an' which may result from the program that extracted the names as I extracted the next word after "this is". As this phrase usually followed by the dog name.

```
name
None        745
a            55
```

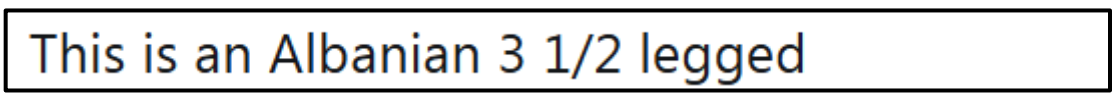
- Also, the program extracted the word "incredibly" as name due to the reason explained earlier.

- 24/7 was mistaken as a rating in status 810984652412424192. which I used my browser to check:



Meet Sam. She smiles 24/7 & secretly aspires to be a reindeer.

- 3 1/2 legged was mistaken as a rating in status 666287406224695296.



This is an Albanian 3 1/2 legged

checking the data types of the data I discovered that:

- timestamp stored as object

during the cleaning phase later on I discovered that:

- some dogs have two stages like doggo pupper or doggo floofer

second file:

gathering:

I used the requests library to download the image prediction file as follows:

```
with open('image_predictions.tsv','w') as file:
    r = re.get("https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv")
    text = r.content.decode('utf-8')
    file.write(text)
```

and then I read it into a pandas data frame:

```
image_prediction = pd.read_csv('image_predictions.tsv',sep='\t')
```

assessing:

through the visual assessment I discovered:

- some confidence of pictures is too low to rely on

```
image_prediction.p1_conf.sort_values()
```

38	0.044333
136	0.055379
1093	0.059033
1370	0.063152
246	0.070076
250	0.071124
145	0.071536
680	0.072885
701	0.081101

- some of the pictures are not for dogs

hen	0.965932	False
desktop_computer	0.086502	False

- the columns of prediction 2 ,3 have too low predictions and we can't rely on them

in programmatic assessment I used a simple for loop to look at all the variables' value counts:

```
for col in list(image_prediction):
  print()
  print('------------------------------------------------------------')
  print (col)
  print(image_prediction[col].value_counts())
```

then I discovered that:

- some of the pictures are duplicated

```
image_prediction.jpg_url.duplicated().any()
```

```
True
```

For the tidiness:

- I needed to create a new column with the whole rating "num/den"
- The predictions columns need to be renamed to appropriate names

M.Gameel
DAND student

Third file:

Gathering:

I have applied for twitter developer account and responded to their emails that I need the access for my Udacity project but the reply that they don't have enough information for my intended usage of the API. they eventually refused my application

So, I used the provided file:

tweet_json = pd.read_json('tweet-json.txt',lines=True)

and extracted the retweet count and the favorite count:

tweet_json = tweet_json[['id','favorite_count','retweet_count']]

for tidiness:

- the tweet_json must be joined with twitter enhanced as they both contain information about the tweets

cleaning:

I divided each of the issues documented in the assess section into the three part for cleaning (define – code -test). all the detailed code is provided in the wrangle_act.ipynb jupyter notebook

I began with the tidiness issues:

Which I solved and tested with the following code:

- the four last columns need to be under on variable column named 'dog_stage'.

code:

```
import numpy as np
twitter_enhanced_clean = twitter_enhanced.copy()

twitter_enhanced_clean.replace({'None':''},inplace=True)

twitter_enhanced_clean['dog_stage'] = twitter_enhanced_clean['doggo']+twitter_enhanced_clean['floofer']+\
twitter_enhanced_clean['pupper']+twitter_enhanced_clean['puppo']
twitter_enhanced_clean.replace({'':np.nan},inplace=True)
twitter_enhanced_clean.drop(columns=['doggo','floofer','pupper','puppo'],inplace=True)
```

test:

```
twitter_enhanced_clean.head()
```

- the tweet_json must be joined with twitter enhanced as they both contain information about the tweets

```
twitter_enhanced_clean=twitter_enhanced_clean.merge(tweet_json,left_on='tweet_id',right_on='id')
```

Cleaning quality issues:

- the name (a, an) in the name column

```
twitter_enhanced_clean.name.replace({'a':np.nan,'an':np.nan,'A':np.nan,'An':np.nan},inplace=True)
```

- the data should include only tweets no retweets or replies

```
#remove reply tweets
twitter_enhanced_clean = twitter_enhanced_clean[twitter_enhanced_clean.in_reply_to_status_id.isna()]
```

```
#remove retweets
twitter_enhanced_clean = twitter_enhanced_clean[twitter_enhanced_clean.retweeted_status_id.isna()]
```

- timestamp stored as object

```
twitter_enhanced_clean.timestamp = pd.to_datetime(twitter_enhanced_clean.timestamp)
```

- wrong extracted name 'incredibly'

```
twitter_enhanced_clean.loc[541,'name'] = np.nan
```

- 24/7 mistaken as rating in tweet 810984652412424192

```
twitter_enhanced_clean =twitter_enhanced_clean.drop(515)
```

- 3 1/2 legged mistaken as rating in id 666287406224695296

```
twitter_enhanced_clean = twitter_enhanced_clean[twitter_enhanced_clean.tweet_id != 666287406224695296]
```

- some dogs have two stages like doggo, pupper or doggo, floofer

```
twitter_enhanced_clean =\
twitter_enhanced_clean[~twitter_enhanced_clean.dog_stage.isin(['doggopupper','doggofloofer','doggopuppo'])]
```

- some pictures are duplicated

```
image_prediction_clean.drop_duplicates(subset=['jpg_url'] , inplace=True)
```

- some confidences are too low

```
#lets drop values less than 0.4
image_prediction_clean = image_prediction_clean[image_prediction_clean.p1_conf >=0.4]
```

- some pictures are not for dogs

```
image_prediction_clean = image_prediction_clean[image_prediction_clean.p1_dog == True]
```

- the columns of prediction 2 ,3 have too low predictions and we can't rely on them

```
image_prediction_clean = image_prediction_clean[cols[:-6]]
```

Finally, I joined the data and stored it in the twitter_archive_master.csv File

Final note: I reduced the size of details on cleaning phase because of the word count limit of the report. **But all the commented code is provided in wrangle_act.ipynb jupyter notebook**

thanks