Data-Analysis-NanoDegree-Udacity-Project-4

Wrangle-Report done-by-Cherif.Arsanious

Wrangle-Overview divided into 3 parts

1- Gather Data

I have to gather data from a csv file prepared by the instructor, a tsv file from a url using the requests l
using tweepy.

2- Assess Data I have to assess the collected data visually and programatically

3-Clean the data clean them from dirty and messy data and test to make sure of my cleaning process

First step in my wrangle effort is to import the required libraries that will help me to gather, assess an

I made a repository on github to store the file and save my progress

Gathering Step

My first source of data is twitter_archive.csv file.

That file was ready available on Udacity platform and just needed to be imported through the right libr
to 1-8-2017 time frame. Running the describe function on the rating numerator and rating denominato
0 which is weird that some rating were 0s and the maximum rating_numerator is 1776 which is most
misleading, and the same goes for the 170 in the denominator rating. The task is to deal with original
78 replies to tweets

My second source of data is image_prediction.tsv file

That file was ready to be downloaded through Udacity platform but it was asked to do this task throug

The other way to import the image_prediction file by requests library Ok I will start by checking the rov
and open their urls to check if these tweets contained dogs pictures or not as based on that I will kee
checking these rows by random check, I found out that these tweets do contain dogs and the model v

My third source of data is the twitter api

I will use the twitter api itself to collect two additional columns required in this project(retweet_count,
account on twitter and requested access to their api and was granted access. After that I used twwep
to a text file and then read them line by line and create my third dataframe which is ret_fav_count.

Assessing Step

Quality Issues noticed

- timestamp in twitter_archive is of type object and not timedate
- 'in_reply_to_status_id', 'in_reply_to_user_id','retweeted_status_id', 'retweeted_status_user_id', 'ret twitter_archive have alot of missing values and are of no signicficance to our analysis
- None values in name column in twitter_archive instead of np.nan
- 'a' values in name column in twitter_archive instead of np.nan
- None values in dogs stages are not interpreted as np.nan in twitter_archive
- dogs' breeds are not consistent (some are capital and some are not) in image_prediction datafr
- timestamp should be treated as timestamp not object type
- 'p1' column name in image_prediction dataframe is not a suitable name to indicate that this colu confusing name
- 'p2' is a confusing name and not indicating dog breed indincated
- 'p2-dog' is a confusing name to indicate that it is the probability of being a dog or not
- Do i need all the columns in image_prediction data frame
- image_prediction assumed that some images don't belong to dogs and interpreted other stuff t
- timestamp in ret_fav_count dataframe of object type
- 181 retweets found and not original tweets
- 78 reply to tweets and not original tweets
- 59 missing values in expanded urls
- extreme min and max values in numerator and denominator ratings

Tidiness

- 3 tables that have common columns and should be probably merged in one
- doggo, floofer, pupper, puppo should be stacked in one column

Cleaning and Testing Step

I will first create three copies of our three dataframes to start the cleaning process without missing w

```
#making new copies of our dataframes to start the cleaning process
twitter_archive_clean=twitter_archive.copy()
image_prediction_clean=image_prediction.copy()
ret_fav_count_clean=ret_fav_count.copy()
```

1-Change timestamp in twitter_archive is from type object to type timedate Test

2-Drop rows where expanded_urls are missing

3-Drop 181 rows where there are retweets since we are only interested in orginal tweets

4-Drop 78 rows where there are replies to tweets since we are only interested in original tweets Test

5-Drop 'in_reply_to_status_id', 'in_reply_to_user_id','retweeted_status_id', 'retweeted_status_user_id', 're
twitter_archive as they are all missing values and are of no signicficance to our analysis as we are onl

6-Replace None values in name column in twitter_archive with np.nan Test

7-Replace 'a' values in name column in twitter_archive with of np.nan Test

8- make breed name consistent with lower case only(some are capital and some are not) in image_p

9- change 'p1' column name in image_prediction dataframe which is not a suitable name to indicate t
is a confusing name so I will replace with dog_breed Test

10- Drop columns ['p2','p2_conf','p2_dog','p3','p3_conf','p3_dog'] by inspection on google sheets, theses
dogs so they are not reliable to our project as by random inspection all tweets contained dogs picture
breeds in this dataframe so they are quality issues for me and i will drop these columns

Test

11- Change ['p1_conf','p1_dog'] to ['prediction_confidence','dog'] respectively Test

12- change the values in breed column that do not belong to dogs to np.nan image_prediction assum
and interpreted other stuff than dogs and which is not true

by random inspection of these urls, I found that they contained dogs picture and the model failed to p
breed values to np.nan

Test

13- Drop dog column now that we don't need it anymore

Test

14- Change 'name' column name to 'dog_name' to avoid confusion

Test

15- change timestamp type from object to datetime in ret_fav_count dataframe

Test

16-Drop rows where denominator are not equal to our standard 10

17-Drop rows where rating_numerator are equal to 0 or those whose have rating above 20

Test

Tidiness

1-Stack dog stages in one column

```
#the code did not work for me and created a 4 times bigger dataframe that i could not hand
#twitter_archive_clean=pd.melt(twitter_archive_clean,id_vars=['tweet_id', 'timestamp', 'so
#'floofer', 'pupper', 'puppo'],var_name='dog_stage')
```

This cleaning step combines both a quality cleaning issue and a tidiness issue

the documentation in this website helped me https://pandas.pydata.org/pandas-docs/stable/referen

```
dog_stage = ['doggo','pupper', 'floofer', 'puppo' ]
for i in dog_stage:
        twitter_archive_clean[i] = twitter_archive_clean[i].replace('None', '')
#concat the four strings together
twitter_archive_clean['stage'] = twitter_archive_clean.doggo.str.cat(twitter_archive_clean

twitter_archive_clean = twitter_archive_clean.drop(['doggo','floofer','pupper','puppo'], a


twitter_archive_clean['stage'] = twitter_archive_clean['stage'].replace('', np.nan)
```

Test

2-Merge the three dataframes together in a master one

```
twitter_archive_master=pd.merge(left=twitter_archive_clean,right=image_prediction_clean,on
```

```
twitter_archive_master=pd.merge(left=twitter_archive_master,right=ret_fav_count_clean, on=
```

I will rearrange columns for better convenience

```
twitter_archive_master=twitter_archive_master[['tweet_id','favorites', 'retweets','dog_nam
twitter_archive_master.sample(10)
```

Storing

I will store the twitter_archive_master as a csv file and also a database table and this will end my wra

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

```
from sqlalchemy import create_engine
```

```
engine = create_engine('sqlite:///twitter_archive_master.db')
```

```
#@title
twitter_archive_master.to_sql('master',engine,index=False)
```