Data-Analysis-NanoDegree-Udacity-Project-4

Wrangle-Report done-by-Cherif.Arsanious

In this project, my task was to wrangle data from the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. It is a twitter account that rates people's dogs.

Introduction

Wrangle-efforts were divided into 3 parts:

1- Gather Data

I have to gather data from a csv file prepared by the instructor on Udacity platform, a tsv file from a url using the requests library and gather data from twitter api using tweepy library.

2- Assess Data

I have to assess the collected data visually and programatically

3-Clean the data

I should clean the data from dirty data (problems with the quality of the data) and messy data (problem with the structure of the data), so first, I define each cleaning step, write the appropriate code and finally test to make sure of my cleaning process was successful.

4-Repeat the steps if necessary (sometimes the cleaning process reveals problems that were hidden before the primary cleaning.

This report was asked to be brief so I omitted any code and the code is in my wrangle_act.ipynb with enough markdown cells to explain my steps.

First step in my wrangle efforts was to import the required libraries that will help me to gather, assess and clean the data, which are (pandas, numpy, json, requests,and tweepy libraries)
I made a repository on github to store the files and the new dataframes I created and to save my progress in the ipynb notebook.

Data Gathering

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 pandas library.

My second source of data is image_prediction.tsv file

That file was ready to be downloaded through Udacity platform also and simply be imported by the pandas library with the tab separator, but it was asked to do this task through requests library so i did it by both ways and the needed code is in the attached wrangle_act.ipynb. The other way to import the image_prediction file by requests library.

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, favorite_count). I created a development account on twitter and requested access to their api and was granted access. After that I used tweepy library to download tweets json content to a text file and then read them line by line and create my third dataframe which is ret_fav_count.

Data Assessment

Our data extends from 15-11-2015 to 1-8-2017 time frame and although the twitter account is still running to this day but the project focused only on this time frame as another source of data didn't deal with tweets beyond 1-8-2017.

This is the step where I just take notes of the quality issues and structures issues that I discover by viewing the data frames visually on google sheets and programmatically through methods like info(), describe(), and other methods to check for missing values and inconsistent data.

Quality Issues noticed

- The column 'timestamp' in twitter_archive data frame is of type object and not timedate, type object is not suitable for dates and time columns.
- The project specified to deal with original tweets only and not retweets or reply to tweets, 181 retweets found and not original tweets, 78 replies to tweets and not original tweets were also found.
- After addressing the previous step, I notice that ('in_reply_to_status_id', 'in_reply_to_user_id','retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp') columns are of no significance to our analysis.
- None values in 'name' column in twitter_archive instead of np.nan, and when running the info() method on this column, python didn't consider the None values as missing values and that is misleading
- 'a' values in name column in twitter_archive instead of np.nan and that also is not interpreted as missing values in python and also is misleading
- None values in dogs stages are not interpreted as np.nan in twitter_archive

- Dogs' breeds ‘p1’ are not consistent (some are capital and some are not) in image_prediction dataframe.
- 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 column is dog breed predicted. It is a confusing name
- 'p2' is a confusing name and not indicating dog breed indicated
- '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 than dogs and which is not true
- timestamp in ret_fav_count data frame of object type
- 59 missing values in expanded urls
- extreme min and max values in numerator and denominator ratings

Tidiness Issues

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

Data Cleaning

Before the cleaning step, I will first create three copies of our three data frames to start the cleaning process without missing with the original dataframe. It is better to be organized than to be sorry. In the cleaning step, there are 3 substeps which are a-Define my solution to each issue I discovered in the assessment step, b-Code the appropriate code to execute my solution, and finally c-Test my solution also by code to make sure the issue was resolved. Due to the concise nature of this report, I will write only the define steps and the code will be found clearly in the wrangle_act.ipynb attached with this report.

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

4-Drop 78 rows where there are replies to tweets

5-Drop 'in_reply_to_status_id', 'in_reply_to_user_id','retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp' columns in twitter_archive data frame.

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_prediction dataframe Test

9- change 'p1' column name in image_prediction dataframe with ‘dog_breed’

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

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

12- change the values in breed column that do not belong to dogs to np.nan image_prediction assumed that some images don't belong to dogs 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 predict them so I need to change these breed values to np.nan

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

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

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

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

Tidiness

1-Stack dog stages in one column (This cleaning step combines both a quality cleaning issue and a tidiness issue) the documentation in this website helped me pandas.Series.str.cat — pandas 1.0.1 documentation

2-Merge the three dataframes together in a master one

I will rearrange columns for better convenience

Storing Data

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