Wrangle report

Table of contents

0. Introduction

Udacity data wrangle project aims to teach the ability to gather data from different sources and to clean it to be able to operate on it.
The project uses tweets from the very good bois Twitter account **WeRateDogs®** and uses the collected data for data wrangling and analysis.
The steps through the project are summarized as follows:

1) Gathering data
2) Assessing data
3) Cleaning data
4) Data analysis

1. Gathering data

It was required to gather data from 3 sources into Pandas dataframes before working on them.

1.1 Twitter_archive_enhanced.csv :

The CSV file is downloaded from resources and is parsed directly into a data frame using this code:

```
#read WeRateDogs archived tweets
type_dict = {'rating_numerator': 'float32', 'rating_denominator': 'float32'}
twitter_archive = pd.read_csv('twitter-archive-enhanced.csv', low_memory=False, dtype=type_dict)
#show info about input (entries, columns names and Dtypes , etc.)
twitter_archive.info()
```

Rating_numerator is explicitly specified as float after data visualization (mainly to capture the floating point ratings). I parsed rating denominator as a float to after which I saw no need but it is what it is.

1.2 image_prediction.tsv :

The file is downloaded from Udacity resources and is parsed directly to a dataframe while specifying tab character as the delimiter using this code :

```
image_prediction = pd.read_csv('image-predictions.tsv', sep='\t' )
#show info about input (entries, columns names and Dtypes , etc.)
image_prediction.info()
```

1.3 tweet_json.txt :

The file is downloaded directly from Udacity resources due to difficulties in getting a Twitter developer account and is parsed into a dataframe using JSON load :

```
json_file = open('tweet_json.txt', encoding='utf-8')
tweet_json = pd.read_json(json_file, lines=True)
tweet_json= tweet_json.rename(columns={"id": "tweet_id"})

json_file.close()
tweet_json.info()
```

2. Assessing data

Each data frame is saved as CSV to visualize data using Excel, also Panda functions are used to find duplicate images, Retweets, duplicate tweets and incorrect ratings.
Jupyter notebook has more details on the functions used.

3. Cleaning data

After using Pandas functions and manually visualizing data, I defined the following issues:
1- There are ids that are actually retweets (quality issue #1).
2- There are duplicate images from the same tweet that needs to be handled.(quality issue #2)
3- Some images do not have a dog prediction at all so it's needed to collect the image that is most likely correct before removing the duplicates.(quality issue #3)
4- Some tweets do not have an identified dogs at all which are to be removed from the collected data.(quality issue #4)
5- Some tweets are confused while parsing rating due to the presence of a simillar regex pattern (e.g: 24/7). (quality issue #5)
6- Some tweets do not have a correct rating in the text at all which are to be removed from the collected data. (quality issue #6)
7- Some tweets have a floating point rating which is not correctly parsed.(quality issue #7)
8- Some tweets have a multiple dogs rating which are also to be removed from the collected data.(quality issue #8)

9- There is no need to have many columns for the dog type(pupper,puppo,etc.)(Tidiness #1)

10- There are some columns that are not needed in the data analysis which are to be removed from the collected data.(Tidiness #2)

11- There are tweets that do not have an image at all which are also to be removed after merging.(quality issue #9)
12- Third table had inconsistent key (id instead of tweet id) that needed to be changed.(quality issue #10)
13- Time stamp in twitter archive is parsed as an object not a datetime

I followed the follwing procedure while cleaning the data:

3.1 Removing retweets

I removed the retweets using the following snippet:
(there were no retweets in tweet_json but I did its step as a general approach for wrangling data)

```
#delete the retweets found above in twitter archive
twitter_archive2 = twitter_archive2[pd.isnull(twitter_archive2['retweeted_status_user_id'])]

#delete the retweets in tweet_json (None found above but just in case)
tweet_json2 = tweet_json2[tweet_json2['retweeted'] == False]
```

3.2 Adjust ratings

To adjust ratings, I made three steps
A) Research the tweets whose denominator != 10 and find a pattern whose denominator = 10
B) Reassign rating to tweets having floating points
C) Remove the remaining tweets whose denominator != 10

3.3 Handle duplicate images(images corresponding to the same tweet)

To handle duplicate images I made the following steps:

A) Make 2 columns for prediction and species, which will carry the first guess that is actually a dog. For example: if p1 is an orange , p2 is a golden retriever then the function will assign the golden retriever to the species column with probability p2
B) Sort the dataframe by probability before removing the duplicate images which will then take the highest pobability guess among all the images corresponding to the same tweet while removing
C) Remove duplicates with keep first option

3.4 Remove unused columns from the three tables

3.5 merge the tables with twitter_archive as the base table

3.6 Remove tweets that have no jpg url

3.7 Change time stamp to datetime

4. Data analysis

After merging data in one data frame, some analysis is made to get some insights which are described in act_report.pdf.