wrangle_report

April 28, 2019

1 Wrangling Report

* * * * * * * * * *	**Udacity**
Nanodegree:	*Data Analyst*
Student:	*Theresa Kocher*
Date:	*28th April 2019*

1.1 Data Sources

There are 3 sources were we get different parts of our data.

1. The twitter archive of the twitter account **WeRateDogs**, which can be found here: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv

2. The predicted dog breeds of the images of a **WeRateDogs** tweets can be downloaded here: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

3. The twitter API `tweepy` which you can use to access the *favorite_count*, and *retweet_count* for every given tweet_id available in the **WeRateDogs** twitter archive (1). To use the twitter API you need to make a twitter developer account first, and add this project as application, to get an access key and token to request twitter data.

1.2 Gathering

To gather the data from the three sources three different methods were used.

1. The twitter archive was downloaded manually and was loaded with the pandas function `pd.read_csv()`.
2. The predictions were downloaded programatically using the `requests` library.
3. The retweet and favourite counts were gathered with the `tweepy` twitter API. To avoid automatically blocakge of the API when accessing too much data, you need to use the `wait_on_rate_limit=Tru` and `wait_on_rate_limit_notify=True`parameters in the `tweepy.API()` function. Also you need to use `try except` to avoid that deleted tweets will stop your script.

1.3 Assessing

To assess the data, the following aspects were considered.

- The data types were viewed by using the pandas `pd.DataFrame.info()` function.
- The number of unique values of colum `twitter_id` were observed with pandas `pd.Series.nunique()`.
- Duplicates in the `twitter_id` column were checked with pandas `pd.Series.duplicated()`. (Same result as nunique)
- All columns were checked for missings.
- Check columns and text to find out how retweets can be found.
- Check the validity of the rates by viewing `denominator` values.

Result:

- The columns `timestamp` of the first dataframe and `created_at` of the last dataframe are not of type datetime.
- The columns `doggo`, `floofer`, `pupper`, `puppo` are used like type *boolian* but are from *string* data type.
- The column `twitter_id` doesn't have duplicates in the first dataframe.
- There are rows with missing names in the first dataframe.
- Retweets are tweets without image url (`expanded_url` in first dataframe).
- Retweets are tweets with a retweet reference (`retweeted_status_id` in first dataframe).
- Not all ratings have valid `denominator`. Only 10 is a valid denominator.

1.4 Cleaning

Taking the results from the assessing step, we can derivate the following cleaning steps:

- Convert `timestamp` and `created_at` columns to datetime.
- Convert `doggo`, `floofer`, `pupper` and `puppo` columns to bool by replacing the values.
- Remove tweets without images (`expanded_url`).
- Remove retweets with retweet id (`retweeted_status_id`).
- Reomving tweets without dog `name`. This was decided to have a name to every tweets dog later in analysis.
- Remove rows where rating_denominator != 10.
- Remove when prediction `p1_dog` is `False` in third dataframe.
- Remove newer tweets than August 1st, 2017.

1.5 Tidying

The three tables contain information about mostly the same tweets. So it is necessary to bring the information of all three tables togehther into one. This is why we join the tables and only keep those columns, we need for analysis later. Following steps were taken for tidying the data:

- Inner join on tweet_id (keep only tweets where data is available in all 3 dataframes).
- Take ranking columns from first dataframe(`rating_numerator`, `rating_denominator`).
- Take `name` column from first dataframe.
- Extract image from second dataframe (`jpg_url`).
- Take `retreat_count` from third dataframe.
- Take `favorite_count` from third dataframe.
- Take all 3 predictions and confidence values from third dataframe.
- Remove needless columns.