Wrangle Report

Yu Tao 2020/12/26

Introduction:

This is a brief report on the twitter data wrangling steps: gather, assess, and clean. The Twitter data used is from **WeRateDogs**, a Twitter account that rates people's dogs with a humorous comment about the dog.

Data Gathering:

Three datasets have been gathered from various sources for this project.

The WeRateDogs Twitter archive:

This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of the tweets from WeRateDogs as they stood on August 1, 2017. This file is directly downloadable, with the name **twitter_archive_enhanced.csv**.

The tweet image predictions:

This file shows what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (**image_predictions.tsv**) is hosted on Udacity's servers and is downloaded programmatically using the Requests library and a link URL.

Additional Twitter data:

Each tweet's favorite count and retweet count are important for analysis. I created a Twitter API account and used the tweet IDs in the WeRateDogs Twitter archive, queried the Twitter API for each tweet's JSON data using Python's **Tweepy** library and store each tweet's entire set of JSON data in a file called **tweet_json.txt** file. From this JSON file, I extracted the tweet ID, retweet count, and favorite count, and utilized them in the final analysis.

Data Assessing:

For each of the datasets, I import them into separate pandas dataframes, and assessed their properties based on the **quality and tidiness** of the data. This is because, for data downloaded from various sources, it might contain issues such as missing values, outliers, etc. There might also be wrong or inaccurate values when we programmatically extract information.

I used basic pandas commands (**head, info, value_count, sample, groupby, duplicate**) to assess the data from different aspects. After exploration, I found **11 quality issues and 2 tidiness issues** for the three dataframes.

df_twitter_archive (quality):

- there are missing values in the dataframe.
- there are retweets and replies in the dataframe.
- the data type of timestamp should be change to datetime.
- some ratings are not correctly extracted from the text (float number issue).
- sometimes there are more than one ratings for single tweet.
- some of the dog names are wrong (with names like 'a', 'the', 'an').
- many of the dogs don't have a dog type (doggo/floofer/pupper/puppo).

df_image_prediction (quality):

- there are duplicate predictions from retweets.
- some pictures got predicted even though they don't show dogs.
- the prediction results are inconsistent in using lower/upper case letters.
- we should condense the 3 predictions into one based on the confidence level.

df_twitter_archive (tidiness):

- the four columns (doggo/floofer/pupper/puppo) should be combined into one.

all 3 dataframes (tidiness):

- we should merge the three tables since they share tweet_id.

Data Cleaning:

Data cleaning is divided into three parts: **define, code, and test**. For each of the issues listed in the assessing section, I used commands like drop, merge, count to fix them one by one.

I have also learned other skills from the wrangling process, such as text processing when I dealt with rating numerator and demoninator, and string operations when combine columns.

Besides, it is a good practice to make copies of the original dataframe, because whenever you make a mistake cleaning you data, you can easilty track back using the copies data.

```
In [ ]:
```