Wrangle_Report

Author: Gour Bera

Date: 1st-Jul-18

Introduction

The dataset that I am be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for us to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017.

Software used

Below the following packages (libraries) used in this project

- pandas
- numpy
- matplotlib
- requests
- tweepy
- json
- tqdm

Data wrangling

Data wrangling consists of:

1. Gathering data
2. Assessing data
3. Cleaning data

Gathering data

Gathering Data for this Project composed of three pieces of data as described below:

1.1 The WeRateDogs Twitter archive. Manually downloaded this file by clicking the following link: twitter_archive_enhanced.csv

1.2 The tweet image predictions, i.e., what breed of dog (or another 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 I have downloaded it programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

1.3 Each tweet's retweet count and favorite ("like") count at minimum, and any additional data you find interesting. Using the tweet IDs in the WeRateDogs Twitter archive, query 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. Each tweet's JSON data written to its own line. Then I read the file line by line into a pandas DataFrame.

Assessing data

After gathering each of the above pieces of data, assess them visually and programmatically for quality and tidiness issues was next step. I could detect and document the following quality issues and tidiness issues.

Quality issues

2.1.1 Could not fetch data from Twitter API for some tweet_ids

2.1.2 Type of 'timestamp' column in df_clean is Object instead of timestamp

2.1.3 Possible incorrect data in 'name' column like- 'a', 'the', 'this'

2.1.4 'numerator' and 'denominator' columns have some unusual values

2.1.5 column name 'id' in tweet_df_clean DataFrame should be changed to 'tweet_id' in-order to merge with other DataFrame

2.1.6 Missing values in images_clean DataFrame (2075 rows instead of 2356)

2.1.7 We have null values in various column

2.1.8 Type of Various column like in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id in tweet_df_clean DataFrame is Float instead of Int

2.1.9 Some tweet_ids does not actually contain a picture of dog like- (666052000000000000,666412000000000000)

2.1.10 Calculate rating based on 'numerator' and 'denominator'

Tidiness issues

2.2.1 Various unwanted columns in df_clean, and tweet_df_clean DataFrame
2.2.2 We can have one column 'dog_stage' instead of doggo floofer pupper and puppo
2.2.3 df_clean, images_clean and tweet_df_clean should be merged to one DataFrame

Cleaning data

3.1 Convert column 'timestamp' from Object to timestamp

'timestamp' column in df_clean DataFrame changed from Object to timestamp and created one more column called ‘date’ based on timestamp data

3.2 Rename all possible incorrect name to NaN

There are some unusual dog names like 'a','all','an','by','his','my','just','this','the' removed from DataFrame as they does not seem real dog name.

3.3 Rename column 'id' to 'tweet_id'

Column name 'id' in tweet_df_clean DataFrame modified to 'tweet_id'

3.4 Drop all unwanted column from all 3 DataFrame

All unwanted column removed for DataFrame

3.5 Create one column 'dog_stage' instead of doggo floofer pupper and puppo

Created one column called 'dog_stage' instead of doggo floofer pupper and puppo

3.6 Create column 'rating' to calculate rating based on 'numerator' and 'denominator'

Created new column called ‘rating’ based on 'numerator' and 'denominator' columns value.

3.7 Merge all 3 DataFrame

All three different DataFrame merged to one DataFrame called df_master

Conclusion

Data wrangling indeed is a core skill that everyone who works with data should be familiar with since so much of the world’s data isn’t clean. If we analyze, visualize, or model our data before we wrangle it, our consequences could be making mistakes, missing out on cool insights, and wasting time. We couldn’t be able to make some of the visualizations without wrangling (i.e dog gender partition). **So best practices say wrangle Always.**