The Data Wrangling Report

Introduction

In this project I have done the wrangle process in 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.

The Data Wrangling process is divided into three Steps:

Gathering

I have gathered the data from the following three recourses:

1. The WeRateDogs Twitter archive, which I Downloaded this file manually by clicking the following link: twitter_archive_enhanced.csv (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv)

2. The tweet image predictions, i.e., 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 should be downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

3. Each tweet's retweet count and favorite ("like") count at minimum. Using the tweet IDs in the WeRateDogs Twitter archive, I have 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. Each tweet's JSON data is written to its own line. Then read this .txt file line by line into a pandas DataFrame with tweet ID, retweet count, and favorite count.

Accessing

After gathering each of the above pieces of data, I assessed them visually and programmatically for quality and tidiness issues. I have detected and documented the following quality and tidiness issues:

Quality

For `twitter_archive` table

1. There is a value in the following columns ((in_reply_to_status_id, in_reply_to_user_id), (retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp)), and we want original ratings (no retweets) that have images.
2. Missing values in (expanded_urls) column.

3. The link tag ("\\") in (source) column
4. None value in ('doggo', 'floofer', 'pupper', 'puppo') columns to NaN.
5. The values ('doggo', 'floofer', 'pupper', 'puppo') as the dog have been categorized in more than one.
6. Erroneous data type of (timestamp).
7. The url in the (text) column.
8. Wrong values in (rating_numerator, rating_denominator).
9. rating_denominator not always 10

For `image_predictions` table

10. The _ and the first letter in each word capital in the following columns (p1, p2, p3)

Tidiness

1. `df_json` should be part of the `twitter_archive` table
2. one variable in four column in `twitter_archive` table (dog_stage)
3. In `twitter_archive`, the following columns ((in_reply_to_status_id, in_reply_to_user_id), (retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp)) is not needed any more.
4. The following columns' name in `image_predictions` ('p1', 'p1_conf', 'p1_dog', 'p2', 'p2_conf', 'p2_dog', 'p3', 'p3_conf', 'p3_dog') is not self-expressive
5. `image_predictions` should be part of the `twitter_archive` table

Cleaning

I have cleaned the each of issues that I have documented in the accessing step. The cleaning step have been done after make copies of the 3 files and then follow the three cleaning steps:

1. Define: convert our assessments into defined cleaning tasks.
2. Code: convert those definitions to code and run that code.
3. Test: test the dataset to make sure your cleaning operations worked.