Data Analyst Nanodegree

We Rate Dogs @dog_rates

Wrangle and Analyze Data

Bayan Bin Manea - Project 5

Introduction

Real-world data rarely comes clean. In this project, I will be using Python and its libraries, to gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling.

The dataset that I will 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.

1. Gather

- **The WeRateDogs Twitter archive file:** The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. One column the archive does contain though: each tweet's text, which is used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced." Of the 5000+ tweets, the tweets have been filtered with ratings only (there are 2356).
- **The tweet image predictions file**, 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

- ❏ Unfortunately, I do not have access to Twitter API, so I will be using the tweet json.text file instead. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count.

2. Asses

Quality

Quality: issues with content. Low quality data is also known as dirty data.

The WeRateDogs `twitter_archive_data` table

- Timestamp is captured as a string object not datetime
- Incorrect dog names that starts with lowercase letters
- None values in dogs name
- Tweet_id is captured as an int not object string
- Archive data contains retweets along with original tweets
- Retweeted_status_id retweeted_status_user_id , retweeted_status_timestamp, columns are not needed for the anlaysis

Image Predictions File `image_pred` table

- Inconsistent capitalization in p1,p2,p3 column, some are written in title case and lowercase
- Missing data, Image Predictions File table has 2075 tweets information

  Twitter API `tweet_json` table
- missing data, Twitter API table has 2354 tweets information

Tidiness

Tidiness: issues with structure that prevent easy analysis. Untidy data is also known as messy data.
Tidy data requirements:

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.

- twitter_archive_data , image_pred , tweet_json tables describe one tweet

The WeRateDogs twitter_archive_data table

- four variables (doggo, floofer, pupper, puppo) in one column dog_stage

Image Prediction File image_pred table

- p1, p2, p3 columns names are not clear

Twitter API tweet_json table

- id columns name needs to be tweet_id to match with twitter_archive_data and image_pred table

3. Clean

To clean the data, I began by creating a copy of the data into a new dataframe to fix the quality and tidiness issues identified in the assessment stage previously. The programmatic data cleaning process has three steps: define, code and test. I began cleaning the data by converting necessary data types, dropping incorrect dog names and tackling the rest of the problems programmatically. After I'm done with the cleaning process, I merged the cleaned data into a new csv file called twitter_archive_master.csv.