Wrangle and Analyze Data Project

By: Niladri Ghosh

Introduction

The wrangle and analyze data project was provided by Udacity team as a part of the Data Analyst Nanodegree Project. The project involves wrangling of data in order to achieve a perfect analysis. Twitter user @dog_rates, commonly known as WeRateDogs provided their twitter archive for analysis. WeRateDogs rate's picture of other people's dogs in a humorous manner and often provide ratings more than 10 and denominator always 10.

To wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information. In addition, gathering, then assessing and cleaning is done for generating crucial analyses and visualizations.

Part I - Gathering data

The data of this project consists of three different data sets:

- The WeRateDogs Twitter archive twitter_archive_enhanced.csv.
  Udacity Team: [[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 I 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, I have filtered for tweets with ratings only (there are 2356).]]

- 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 has been downloaded programmatically using the Requests library.

- Each tweet's retweet count and favorite (i.e. "like") count at minimum, and any additional data we 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 has been written to its own line. Then we read this .txt file line by line into a pandas DataFrame with tweet ID, retweet count, and favorite count.

Part II - Accessing data

This step allows us to identify quality and tidiness issues. Low-quality data are considered as dirty data and have content issues. We have to assess at least height quality issues. As for untidy data, they are considered as messy data and are structural issues.

There are two types of assessment:

- Visual assessment, which consists in scrolling through the data
- Programmatic assessment, for which we use python statistical libraries such as pandas, numpy, etc.

Issues:

Tidiness Issues:

- Merging the three dataframes into one using tweet_id.
- Joining the dog stages into a single column instead of four different columns. doggo, floofer, pupper and puppo columns in archive should be merged into one column named "stage"
- If we remove duplicates from archive (i.e. retweets) we will have empty retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp columns, therefor we must drop them for tidiness also there are many other useless columns in the table.

Quality Issues:

- Tweet_id, timestamp, sources, img_num and dog_stages need to be converted into the right datatype
- Sources column has to be filtered out to get the useful data. < href > should be removed.
- Removing the values in name which are not names.
- Missing values in the dog stages column showing up as 'None'
- Separate timestamp into day - month - year (3 separate columns) for making it simple and efficient.
- Removing the retweets. In text field we can easily find that by first two letters "RT".
- In the text, we can notice some decimal numbers for the ratings numerator part wrongly extracted
- Some numerator values were higher than 10, must be rectified if not removed

Part III - Cleaning data

With this method, we need to define (definition or instruction list) the cleaning task. Then, we code the issue to get it cleaned (drop, extract, islower, loc, etc., methods). At the end, we test the dataset, visually or with code, to assure that the cleaning operations work correctly.

Part IV - Storing, Analyzing and Visualizing data

In this part we have to store our results into files and make documentation on it. After analyzing the data, we can provide many insights from the data set that will come in handy. Also making visual plots or charts that will further enhance the findings and will make things much simpler.

Conclusion

Through the data wrangling and analysis, we used many libraries such as pandas, NumPy, requests, tweepy, and json, which allow us to gather, assess, and clean the data. Finally, we put the following documents together:

- wrangle_act.ipynb: code for gathering, assessing, cleaning, analyzing, and visualizing data
- wrangle_report.pdf: documentation for data wrangling steps: gather, assess, and clean
- act_report.pdf: documentation of analysis and insights into final data
- twitter_archive_enhanced.csv: file provided
- image_predictions.tsv: file downloaded programmatically
- tweet_json.txt: file constructed via API
- twitter_archive_master.csv: combined and cleaned data