Data Wrangling Report

Mohamed Hassan

Introduction

This project focused on wrangling data from the WeRateDogs Twitter account using Python, documented in a Jupyter Notebook (wrangle_act.ipynb). This Twitter account rates dogs with humorous commentary. The rating denominator is usually 10, however, the numerators are usually greater than 10. They're Good Dogs Brent wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. 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.

The goal of this project is to wrangle the WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The challenge lies in the fact that the Twitter archive is great, but it only contains very basic tweet information that comes in JSON format. I needed to gather, asses and clean the Twitter data for a worthy analysis and visualization.

Data wrangling, which consists of:

- Gathering data
- Assessing data
- Cleaning data
- Storing, analyzing, and visualizing our wrangled data Reporting on:
  (1) our data wrangling efforts and
  (2) our data analyses and visualizations

Gathering the Data

There were three data sources that I gathered from; a csv file that was given from the start, a tsv file that I had downloaded programmatically, and json data that I had queried twitter's API for. The csv file (twitter-archive-enhanced.csv) was originally @dog_rates twitter archive, but it had been slightly wrangled. The tsv file (image_predictions.tsv) contained dog breed predictions for each tweet with a picture as generated from a neural network. The data from twitter's API was extracted and written to a text file (tweet_json.txt) in json format. This txt file just contained each tweet's tweet id, favorite count, and retweet count.

Assessing the Data

I used pandas .describe() , .value_counts() .sample() and .info() mainly to assess the data. I didn't quite know what 'one' quality or tidiness issue was so I had to make some executive decisions. The issues I found with the data were as follows:

a- Quality Issues

archive_copy

- Missing values for dog stage (incomplete data for doggo, floofer, pupper, puppo).

- Replace 'None' with 'NaN' for all dog stages.

- Rating numerators and rating denominators values are incorrect.

- Remove entries that are retweets and expanded URLs are unnecessary

- tweet_id is numeric. Should be string.

- Change tweet_archive timestampe from object type to datetime type.

image_predictions_copy

- Remove entries where p1_dog, p2_dog, and p3_dog are all "False".

- Values for p1, p2, and p3 sometimes capitalized but not always.

- tweet_id is numeric. Should be string.

tweet_data_copy

1. tweet_id is numeric. Should be string.

b- Tidiness Issues

1. Doggo, floofer, pupper, puppo are one variable spread across different columns in archive_copy.

2. rating_numerator and rating_denominator can be combined into one column in archive_copy.

3. Combine data frames by tweet_id.

Cleaning the Data

For the consistence of datasets, three new datasets were created, **archive_copy**, **image_predictions_copy**, and **tweet_data_copy**.

Both manual and programmatic methods are being used here to clean the datasets, and quality and tidiness issues were solved in the cleaning section.

For the consistency of the analysis, all three datasets are joined together using unique tweet_ids, and the final csv file is created.

Conclusion

Data wrangling provided the very solid foundation of the further data analysis, and using data wrangling methods in this project, we can fix quality and tidiness issues and create a much cleaner dataset.