Wrangle Report

By Divya Nitin Naidu

Introduction

The goal of this project was to wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. This project was part of the data wrangling section of the Udacity Data Analyst Nanodegree program and is primarily focused on wrangling data from the WeRateDogs Twitter account using Python, documented in a Jupyter Notebook (wrangle_act.ipynb).

Project Details

Real-world data rarely comes clean. Using Python and its libraries, I had to gather, assess, and clean the data, in order for it to be used for analysis and visualization. Fully assessing and cleaning the entire dataset would require exceptional effort so only a subset of its issues (eight quality issues and two tidiness issues at minimum) needed to be assessed and cleaned.

The tasks for this project were:

1. Data wrangling, which consisted of:
    - Gathering data
    - Assessing data
    - Cleaning data
2. Storing, analyzing, and visualizing the wrangled data
3. Reporting on my data analyses and visualizations (act_report.pdf)

Gathering the Data

The data for this project was in three different formats and they were obtained as mentioned below:

Twitter Archive File – WeRateDogs: This was extracted programmatically by Udacity and provided as twitter_archive_enhanced.csv to use.

Image Predictions File: The tweet image predictions, breed of dog present in each tweet according to a neural network. This file (image_predictions.tsv) was hosted on Udacity's servers and downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

Twitter API & Tweet JSON File: By using the tweet IDs in the WeRateDogs Twitter archive, I queried the Twitter API for each tweet’s JSON data using Pythons tweepy library and stored each tweet's entire set of JSON data in a file called tweet_json.txt file.

Assessing the Data

After gathering the data, the three tables were saved and assessed Visually and Programmatically. With both the assessments I looked for Unclean data i.e Dirty data with content issues and messy data with structural issues. So basically I looked for Tidiness and quality issues.

Visual Assessment provided me with issues such as columns 'doggo', 'floofer', 'pupper', 'puppo' in df_tw should be a single column named stage and the column names in image_predictions were not clear and straightforward such as p1,p2.

Programmatic Assessment actually gave me most of the quality issues that were present in the three datasets. Then I separated the issues encountered in two groups quality and tidiness. I also provided a gist of my assessment in the jupyter notebook. I divided the quality issues according to the datasets and checked for completeness, validity, accuracy, and consistency.

Cleaning the Data

This part of the data wrangling was divided in three parts as per the data sets and was further divided as per the three steps Define, code and test. I have provided a headline and code and test blocks below it to make is easy for understanding.

First and the most important step was to create copies of all the three data frames. SO that I can do trial and errors in the copy frames rather than the originals.

In the twitter archive data, I changed the datatype of timestamp and made the dog names consistent i.e. first letter capital. As mentioned earlier the standard for "rating_denominator" is 10, but on checking we found that it includes some other numbers, which could be the mis parse. So, I check the text corresponding to those ratings and noticed that few of them were analyzed incorrectly due to the presence of another fraction in the text. I corrected the same for both rating denominator and numerator.

In the Image Predictions data, I found out that there were 66 duplicate values for jpg_url i.e. same url was added to the data multiple times. So, I dropped the duplicated data. I also changed the column names to make it more descriptive and readable. Again, the dog breeds of the all the three prediction columns involved both upper and lowercases for the first letter. I rectified the same to make it consistent.

In the generated tweet JSON data, most important and required columns were retweet_count and favorite_count, others were actually redundant as the same were present in the twitter archive data. So deleted the unnecessary columns.

Storing the Data

After cleaning the data, I found out that there was no need for three data sets. All the data could be easily made into a single file. So, I joined 'tweet_json' and 'image_predictions' to 'df_tw', to create the twitter_archive_master.csv.

Conclusion

A good data wrangler knows how to integrate information from multiple data sources, solving common transformation problems, and resolve data cleansing and quality issues. A data wrangler also knows their data intimately and is always looking for ways to enrich the data. I have done the same using amazing python Libraries.