Wrangle Report

By Mohamed Gamal

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 FWD Udacity Data Analyst Nanodegree Professional 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 data-frame 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)

Data Gathering:

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.

Data Assessing:

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 twitter_archive that should be a single column named “dog_stage” 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 data-frames. 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 data-frames and checked for completeness, validity, accuracy,and consistency.

Data cleaning:

I started by copying to original data-frames so if I did something wrong I can always go back a don’t affect the original data. This part is divided in three process first you define the problem briefly then you write the code to fix it then you test that code.

First in the twitter_archive_copy data, I removed the retweets and replies then filtered the tweet_ID based on image_predictions to remove the tweets without images and then I combined the three columns I mentioned before into one “dog_stage”column then fixed lost rating.

Second in the image_predictions_copy data, I did the same and removed the retweets and replies then I changed the names of the non-descriptive columns then removed duplicates in the jpg_url column.

Finally in the tweets_df_copy data, I removed unnecessary columns and then I did the same and removed the retweets and replies and then changed the tweet_id name to match other data_frames then I changed those columns data type to be able to preform analysis.

Storing the Data:

After cleaning the data, I found out that there was no need for three data sets. So,I joined “twitter_archive_copy” and “image_predictions_copy” to “tweet_df_copy”, to create the “twitter_archive_master.csv”.