Data Wrangling Report

Introduction

The purpose of this project was to put to use what I have practiced over this last course: data wrangling (gathering), assessing, cleaning, and using those practices to then analyze and visualize the data. The dataset that was given was from the Twitter handle @dog_rates, or WeRateDogs from 11/15/2015 through 08/01/2017. WeRateDogs is a comical account that rate's peoples' dogs with humor and has an irrational rating system (ie: 12/10).

The purpose of this report is to describe my wrangling efforts through a variety of sources, formats, and code.

Overview of Project Details

I was tasked with gathering data from multiple sources, assessing all the data (both visually and programmatically), and cleaning the data so that, when merged, all data was accurate and tidy.

1) ***Gathering Data***
    - The first task was to gather the data for this project from three different datasets. The first, **Twitter Archive File** (twitter-archive-enhanced.csv), was provided by Udacity and manually downloaded into my computer, then uploaded into my Jupyter workspace.
    - The second dataset, **Tweet Image Predictions** (image-predictions.tsv), had three levels of predictions as to what breed was being represented in each image_url. This particular file was also provided by Udacity, but was downloaded programmatically through my Jupyter workspace using the requests library.
    - The third dataset, **Twitter API & JSON** (twitter-api.py), had a ton of unnecessary data that needed to be cleaned before we moved on to assessment. I was able to use the tweet_id column to query the Twitter API's for each tweet JSON using Pythons' Tweepy library, then stored the data in the final file (tweet_json.txt). I then condensed the file down to three columns for easier reading, and merging (later): tweet_id, favorite_count, and retweet_count.

2) ***Assessing Data***
    - Once all three dataframes were imported and saved into my Jupyter workspace, I then moved on to assessing each individual one, both visually and programmatically.

- Visually, I was able to assess the dataframes by using the .head() function. This function helped show if there was any non-null, or NaN, data, or how each row looked (duplicate, missing, egregious data).
- Programmatically, I was able to assess each dataframe a number of ways, which included: .info(), .value_counts(), .duplicated(), and .groupby().
- After visually and programmatically assessing each dataframe, I was able to pinpoint 8 quality issues and 2 tidiness issues. These are sectioned out underneath each dataframes' section in my final project file Wrangling and Analyzing Twitter Data.ipynb.

3) Cleaning Data

- When cleaning data, we must focus on three basic principles: define, code, and test.
- The first step was to create a copy of the three original dataframes. This helps us protect the original should we make any egregious mistakes and/or errors.
- I found that cleaning certain parts of each dataframe was the hardest part of this project as the code was longer than expected. Some columns, such as the text column, had multiple forms of data: this one included text and a url. I was able to filter out the url, leaving just the main body of the tweet in this column.
- Another challenging code during the cleaning stage was cleaning up the last four columns in the df_archive_clean dataframe: doggo, floofer, pupper, puppo. These had too much NaN information, and should be condensed into one column, which could then be filtered per 'level' of dog at the users' discretion. I was able to condense them down into a new column, level_of_dog, and delete the original columns.

Conclusion

Data wrangling is a very important core skill as we have been told multiple times it can account for 80% of the job of a Data Analyst.

I will be the first one to admit that I do not have advanced Python/Jupyter skills, but I try my hardest as I do enjoy working with the program. Everything that I did in this course I know how to complete in Excel, as it is a program I have used on a daily basis for over 12 years now. However, I can see that Python is by-far more efficient and effective in getting the same work done, especially when working with multiple datasets.