WRANGLE REPORT ON THE TWITTER DATA ARCHIVE OF THE WERATEDOGS TWITTER ACCOUNT

BY:

OJODU UTHMAN

Introduction

The aim of this report is to gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it using python and its libraries. The dataset I am attempting to wrangle is the twitter data archive of the user @dog_rates also as WeRateDogs which was provided as one of the projects of the Udacity Nanodegree Program. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

Project Details

The steps I went through while wrangling data involves gathering, assessing and cleaning the parts where I assessed to be dirty. Complete assessing and cleaning is a monumental task which requires a great deal of work, but this report is limited to the scope of the Udacity Nanodegree program project which requires 8 quality issues and 2 tidiness issues at minimum.

The tasks carried out in this project include

1. Gathering the data.
2. Assessing the data.
3. Cleaning the data.
4. Storing the data.
5. Creating the wrangle report.

Gathering the Data

The data gathering in this project requires three files which were of different formats.

- The first one was provided with the course and is the twitter data archive of the WeRateDogs account (twitter_archive_enhanced.csv).
- The tweet image predictions file (image-predictions.tsv) which was hosted on the Udacity servers and I had to be programmatically download using the Requests library and the URL (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv) provided.
- Additional data from the twitter API which I got by querying 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. Each tweets json was then read line by line into a pandas dataframe called twitter_counts.

Assessing the Data

I assessed the data both visually and programmatically. During the visual assessment, I was able to find some issues such as a lot of null values in some columns of the twitter archive dataframe, and the non-descriptive nature of some columns in the image predictions dataframe.

During the programmatic assessment of the data, I was able to delve in deeper and find some other data issues which were separated into Quality and Tidiness issues. Some of the Quality issues found were incorrect datatypes in some columns, columns not needed for the analysis, and missing values represented as None instead of NaN. The tidiness issues where primarily with the twitter archive which had four columns (doggo, floofer, pupper and puppo) that were supposed to be just one column violating one of the laws of tidy data which states that each variable must be a column.

Cleaning the Data

This involves three steps which include: Defining what to clean, writing the code to fix the cleaning issue, and testing the code to see if it works.

First of all, we have to make a copy of the three dataframes to allow flexibility and have access to the initial data collected.

For the twitter archive dataframe, I found that there were some columns not needed for the analysis which I dropped, I changed the datatypes of the tweet_id, timestamp and source columns to string, timestamp and category respectively. There were some rows with retweets instead of original tweets and these rows were filtered out. I cleaned up the messy source column and made it easier to understand and also changed all the 'None' values which should have been 'NaN' to 'NaN'. I collapsed the four columns (doggo, floofer, pupper and puppo) which were all different aspects of the same variable into one column dog growth. I also split the timestamp into two different columns date and time. And thereafter I dropped the timestamp column.

For the Image predictions table, I dropped columns that weren't needed for the analysis. I changed the non-descriptive column names into more suitable column names, and also changed the datatype of the tweet id to the type string. I also changed the prediction confidence value to percentages from proportions. I also had to merge the dataframe to the twitter archive dataframe because they are part of the same observational unit. I made sure that all the dog breeds started with the same case and replaced all the (_) values with spaces within the dog breeds column.

For the twitter counts table, I had to merge the dataframe to the twitter archive dataframe because they are also part of the same observational unit.

Storing the Data

The newly gathered, assessed, cleaned and merged dataset was saved to a CSV file named 'twitter_archive_master.csv'.