DATA-WRANGLING REPORT

Project: Data Wrangling and Analysis using Tweepy to get WeRateDog tweet data via Twitter API

Introduction:

In this project, Using Python and its libraries, I gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. which is called data wrangling. I am documenting my wrangling effort in this report which I will analyze and visualize in the analysis report.

This project is part of the requirement of data-wrangling section of my Udacity Data Analyst Nanodegree Program.

Project Details:

The dataset that i will be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. 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.

WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for students to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.)

The tasks for this project are:

a) Data-wrangling
   - (i) Gathering data
   - (ii) Assessing data
   - (iii) Cleaning data

b) Storing

c) Analyzing & Visualization

d) Report

Gathering the data

I make use of three (3) different formats of data for this project, which are gathered as follows:

1. Twitter Archive enhanced data - This data is downloaded from my student project page (Note: in the introduction, this is the data sent to Udacity for the project purpose)

2. Image Prediction data (tsv) - The tweet image prediction is downloaded programmatically from the Udacity server (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv) via the requests library, and then converted to a data frame. This is a neural network present in each tweet to predict the exact dog image according to the information provided.

3. Twitter API-JSON data - This data is gotten from Twitter API using tweepy. By using the tweet_id in the archive data above, I queried the Twitter API for each tweet data which was returned in JSON format and converted to a data frame, and cleaned.

Assessing the data:

After gathering all three pieces of data, I assess them visually and programmatically for quality and tidiness issues. Keeping in the mind the following specifications that must be assessed.

I only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets. While I only need the data till August 1st, 2017 for the sake of the prediction.

And I identified the following quality and tidiness issues:

- **Quality**

1. False datatype - ids shouldn't be int and timestamp not object type
2. Duplicate data on df1 (Twitter archive table)
3. The tables do not contain the same amount of tweet_id's, 2075 in predict table while more in the archive table.
4. Untidy data in the source column contains html tag (<a></a>)(<a></a>)
5. Dog name like a, an, none not extracted well (Programmatically)
6. Unneeded columns should be deleted (drop)
7. Alphabet format should be constant (not small letter starting some while capital letter state the other).
8. Different rating denominator, which mean different rating standard which i will like to be fixed to 10 as that is the common standard.

   Tidiness

9. The last 4 columns (doggo,floofer, pupper & puppo) in df1 are dog stages which should be collapse into one column (Each variable forms a column). And the none are taken out with an empty space to ease analysis.
10. The three data frame (df1,predict, tweepy_df) needs to be merge.

Cleaning Data

Most of the tidiness and quality issues sighted were with the archive data (df1) which I did justice to by changing the datatype where needed, cleaning out the tweet_id that was not matching with the image prediction, concatenating the stages into one column name growth, deleted unnecessary columns, works on the rating_denominator and then merge the cleaned data to carry out my analysis.

Storing Data

After cleaning the data's, I merge the data set into a single file and then save on my computer as twitter_archive_master.csv

In line with the objective of the project, data rarely comes clean so my activities so far have been about integrating data from different sources and format and cleaning them to make sense out of the information.