WRANGLING REPORT - WeRateDogs Twitter Data Analysis

This project report is to highlight the activities that took place in wrangling data from the WeRateDogs Twitter account. The project's goal was to gather data from 3 different sources in different formats, assess and clean the data, and produce a master data frame for analysis and visualization. The dataset used for this project was gotten from 3 different sources;

1. WeRateDogs enhanced the Twitter archive a csv file, which was provided by Udacity
2. Image predictions file, a tsv file, also provided by Udacity
3. Twitter API, which I had to query using Tweepy library

Gathering Data

The first activity was importing necessary packages needed for this project, some of them include; pandas, NumPy, JSON, os, etc. The first source Twitter enhanced archive, this source was provided by Udacity which they downloaded programmatically from WeRateDogs. The dataset was manually downloaded, and imported into the project workspace using pandas read_csv.

The second source, The Image Predictions file was provided by Udacity as a URL. Dataset from this source was programmatically downloaded using python's Requests library. A file that did not exist prior was created and requests were used to download the data from the Url. A response of 200 proved successful, then the file was opened in the folder using write binary, because of the images. Then the file was read into a pandas DataFrame.

Because all the information needed for the project was not available in the provided Datasets, it was required of me to query the WeRateDogs Twitter account using an API for this additional information, making this the third source. So firstly I had to apply for a Twitter developer's account. The access keys were provided. The information needed was extra information for the tweet ids already existing in the Twitter archive DataFrame. A loop was created to iterate over each tweet id in the archive DataFrame. try and except codes were also implemented in the loop to avoid and note down errors. This took about 30 minutes and 32 errors were obtained. After this, an empty list was created and each line of the JSON file was read, into a new file, using the read format. The following variables; tweet_id, favorite_count, and retweet_ count were accessed and then appended to the empty list initially created. Then it was converted into a pandas DataFrame.

Time to assess

Assessing Data

The Datasets were assessed visually and programmatically assessed. The programmatic assessment was better than the visual. The essence of the assessment was to seek out Data

quality and Tidiness problems, and they were very much present. In the archive data frame, there were completeness issues i.e missing data, consistency issues, where the name column had invalid strings and Tidiness issues where dog stages had different columns which broke each variable as a column rule. some columns also had data type issues like timestamp was a string column instead of date time and tweet id was an integer. The rating denominator column also had consistency issues, the source HTML string had the source names embedded in the string. Also, some retweeted and reply rows had non-null values. Also, It was observed that the image prediction data and the JSON data had a number of tweet ids short of that present in the archive data frame. The fact that columns meant to be in the archive dataset were in the JSON and image prediction DataFrame, further broke the tidy rule that says each type of observational unit is a table. All these issues were documented and data was ready for cleaning.

Cleaning and Storing Data

In this stage, data were transformed from dirty and untidy to clean data. The retweeted non-null values and reply rows were dropped because we needed just original tweets. The missing values were dropped and some that I felt would have affected the analysis were filled with none values. The source names were extracted from the HTML string. The dog stages columns were melted and merged into one column.

Irrelevant columns were dropped and data types were fixed. For the Image predictions data, the predicted and confidence level columns were assessed for the best prediction and the remaining columns were dropped. This was addressed because of it broke the tidy rule. DataFrames were eventually merged together into one master DataFrame. Then the master DataFrame was stored as a csv file.