Wrangle Report

Project: Data Wrangling WeRateDog Twitter Archive

Description: The popular twitter account WeRateDogs with the username @dog_rates with over 4m followers on twitter downloaded their Twitter archive and sent it to Udacity via email exclusively to be used in this project. The archive contains basic tweet data (tweet ID, timestamp, text, etc.) not later than August 1, 2017.

The task is to gather, assess, clean and analyze the data using python and its various libraries.

Tools: python, pandas, numpy, matplotlib, seaborn, request and tweepy.

Datasets:

1. Twitter-enhanced-archive: this dataset was provided by udacity in csv format
2. Image_prediction: this dataset need to be downloaded programmatically using the request library, saved as tsv file
3. Tweet_json.txt: this file was gathered through the use of twitter API

The Data Wrangling Steps are as follows:

1. Gathering the Dataset

i. Twitter-enhanced-archive: This particular dataset was given by udacity in csv format, need to be loaded in the working environment using pandas library as df_1 = pd.read_csv(‘name_of_the_file.csv’)

ii. Image_prediction: this dataset is in tsv format and needs to be gathered programmatically using the request library specifically ‘response = requests.get(url)’, save the file in a directory and read the file using pandas as df_2 = pd.read_csv(‘name_of_the_file.tsv’, sep=’\t’).

iii. Tweet_json.txt: this dataset was gathered using the twitter API, firstly we need to apply to the twitter company for request access to the API, have to filled some information and the reason why the need for the API before it can be granted, the API will be accessed with the given credentials, three fields was extracted (id, retweet_count, favorite_count) and assigned to a dataframe df_3.

2. Assessment of the Dataset

Visual and programmatic assessment was employed on the three datasets, MS Excel is used for the visual assessment, during the programmatic assessment the use pandas functions like .info(), .describe(), .value_counts(), .unique(), accessing the string content of an object type using the .str() method and so on. Below are the key assessments that was administered on the dataset.

Table I: Quality Issues

S/N	Table	Dimension	Issue
1	Twitter-archived-enhanced	validity	invalid datatype for tweet_id
2	Twitter-archived-enhanced	validity	invalid Timestamp datatype
3	Twitter-archived-enhanced	completeness	retweeted data not needed
4	Twitter-archived-enhanced	completeness	reply data not needed
5	Twitter-archived-enhanced	completeness	drop retweets and replies columns
6	Twitter-archived-enhanced	Accuracy	names in lowercase like 'an', 'the', 'a', 'my', 'by', 'not', 'one', 'mad', 'all', 'old', doesnt make sense, hence are invalid
7	Twitter-archived-enhanced	Accuracy	record with tweet_id 776201521193218049 has a name of O'Malley instead of O
8	Twitter-archived-enhanced	Accuracy	record with tweet_id 770414278348247044 has a name of Al Cabone instead of Al

9	Twitter-archived-enhanced	Accuracy	source column contains links and HTML tags
10	Twitter-archived-enhanced	Accuracy	inaccurate values in rating_numerator and in rating_denominator
11	Twitter-archived-enhanced	Accuracy	strange characters like '&amp' and '\n' in text column
12	Image_prediction	validation	invalid datatype for tweet_id
13	Image_prediction	Accuracy	the characters (- , _) in p1, p2, p3 columns
14	Image_prediction	Consistency	inconsistent case in p1, p2, p3 columns (lowercase, uppercase)
15	Tweet_json	validation	invalid datatype for id
16	Tweet_json	completeness	id should be rename to tweet_id for consistency with other tables

Table II: Tidiness Issues

S/N	**Table**	**Dimension**	**Issue**
1	Twitter-archived-enhanced	completeness	doggo, floofer, puppo, pupper columns to single column stage
2	Twitter-archived-enhanced	completeness	rating column from rating_numerator and rating_denominator
3	All Three	completeness	merge the dataframes into a single tidy dataframe

3. Cleaning

First as the ethics, all the three dataframes need to be copied to a new dataframes to avoid making erroneous change on the original data.

All the aforementioned issues were address in the Define – Code – Test, A definition of how to clean, the code execution and the test code to ascertain the code implemented.

Few takeaways from the cleaning, all datatypes issues are change correctly using .astype() method, erraneuos rating_numerator and rating_denominator were extracted from the text using regular expression through the .str() method of a text column, all incorrect names are checked and replace as ‘None’, weird characters in text and p1,p2,p3 columns are cleaned and replaces where the need arise respectively.

Lastly, all the dataframes are merged into a single master dataframe for the next task to be implemented which is the analysis and visualization.