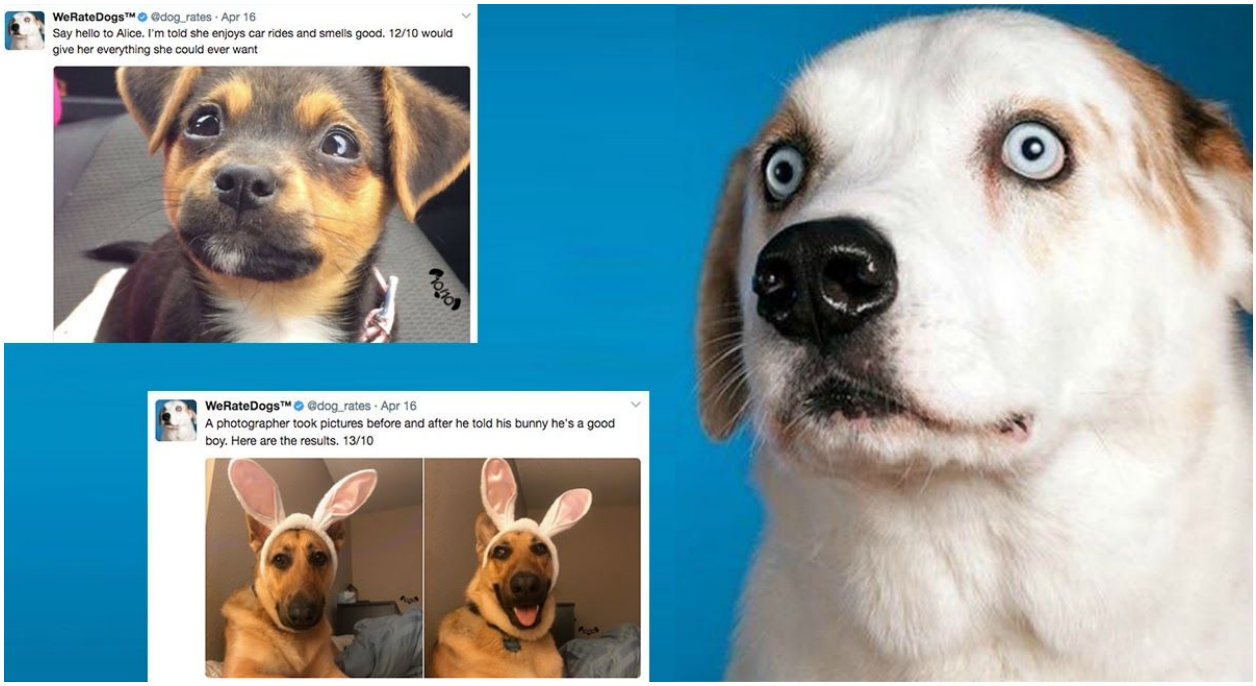
WeRateDogs Wrangle Report



Prepared By: Salma Abdelfattah

10.14.2020

Data Wrangling report

INTRODUCTION

As an assignment project for the Udacity Data Analysis Nanodegree, this report includes the full process of data wrangling of the WeRateDogs Twitter account data. The data wrangling process includes three stages, first data gathering, second data assessment, and third data cleaning. Each stage will be described in this report thoroughly.

DATA GATHERING

Data gathering is the process in which we collect the required data. The data was to be gathered from three different sources, so the process of data gathering was done in three steps:

1- Downloading (manually) the twitter-archive-enhanced.csv directly from Udacity workspace, it was then read using pandas library giving the dataframe `archive`.

2- Downloading (programmatically) the image-predictions.tsv file using the requests package & the given URL in Udacity workspace, it was then read using pandas library giving the dataframe `image_predictions`.

3- Using Twitter's API and the tweepy package to download a JSON file that contains all the required tweets & then extracting the ids, favorite count, and retweet count of every tweet & saving them to a dataframe called `api_df`.

DATA ASSESSMENT

Data assessment is the process of investigating the datasets. Assessment in this project is done visually & programmatically. Each dataset was examined visually using Atom text editor, Google Sheets, & in the jupyter notebook. The programmatic assessment was also done thoroughly using pandas functions (head, tail, info, describe, duplicated, value_counts, isnull). Data were assessed for quality issues and each one was categorized into Completeness, Consistency, Validity, and Accuracy. Tidiness issues were also detected.

DATA CLEANING

Data cleaning is the process of putting the results from assessment into action. This part took the most effort in the whole Data Wrangling process, this indicates its importance and shows how much it facilitates data analysis afterward.

ISSUES & SOLUTIONS

TABLE	ISSUE	TYPE OF ISSUE	SOLUTION
Archive	78 replies	Quality (Validity)	Remove these rows using the `drop` function.
Archive	181 retweets	Quality (Validity)	Remove these rows using the `drop` function.
Archive	Null values in `expanded_urls` column	Quality (Validity)	Remove these rows using the `drop` function.
Archive	`rating_numerator` column has illogical values (max value is 1776)	Quality (Validity)	These values will be investigated and corrected, programmatically & manually, the invalid ones will be deleted
Archive	`rating_denominator` column has illogical values (max value is 170)	Quality (Validity)	These values will be investigated and corrected, programmatically & manually, the invalid ones will be deleted
Archive	Some names in the `name` column aren't capitalized	Quality (Consistency)	Capitalize values of the mentioned columns using `str.title()` function

Archive	Some names in the `name` column are wrong like 'a'	Quality (Accuracy)	Correct names can be extracted from the tweet's text if they exist, the names are usually after the word 'named'
Archive	Erroneous Data Types `tweet_id` & `timestamp` columns	Quality (Accuracy)	For the `tweet_id` column, we change them to string using `astype()` function. For the `timestamp` column we change it to datetime object using the `to_datetime` function
Image Predictions	Some predictions are capitalized and some of them are not	Quality (Consistency)	Capitalize values of the mentioned columns using `str.title()` function
Image Predictions	Erroneous Data Type `tweet_id` column	Quality (Accuracy)	Changeto string using `astype()` function.
Archive	Date & Time are both stored in the same column `timestamp`	Tidiness	Split the `timestamp` column into `date` and `time` columns using `split` function
Image Predictions	Predictions are distributed into 3 columns `p1`, `p2`, and `p3`	Tidiness	Merge the columns together using the `melt` function
API	This table isn't an observational unit	Tidiness	The `API` table should be merged into the `archive` table using the `merge` function

REFERENCES

1. Python requests library
2. StackOverflow: get tweets from twitter API
3. StackOverflow: rate limit reached error
4. StackOverflow: reading text files
5. StackOverflow: indexing
6. Twitter's developer guide
7. Pandas Melt Function documentation
8. Pandas Drop Function documentation
9. Rules of Tidy data