Ibodullaev Fazliddin
Udacity student
July 8, 2020

Wranlge Report

Introduction

This project validates that I mastered the Data wrangling by learning the core concepts of Data wrangling process by Udacity nano-degree program. 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.

Gathering data

There are ways of three datasets that I will work with:

Twitter archive file: the twitter_archive_enhanced.csv was provided by Udacity and downloaded manually.

The tweet image predictions, i.e., what breed of is present in each tweet according to a neural network. This file (image_predictions.tsv) is hosted on Udacity's servers and was downloaded programmatically using the Requests library and URL information

Twitter API & JSON: I downloaded the tweet_json.txt from my classroom and I used JSON data in the file. Because I could not get the tokens of Twitter API. I read this .txt file line by line into a pandas dataframe with tweet ID, favorite count, retweet count, followers count, friends count, source, retweeted status and url.

Assessing data

If we speak visually, I have done this step with two tools. One of them was by printing the three entire dataframes separate in Jupyter Notebook. Second process is checking the csv files in Google Sheets.

If we speak programmatically, I have used a lot of methods such as info, value_counts, sample, duplicated, groupby and etc.

At that point I isolated the issues experienced in quality issues and tidiness issues. Key focuses to remember for this procedure was that original ratings with pictures were needed.

Cleaning data

This piece of the data wrangling was separated in three sections: define, code and test the code. These three stages were on every one of the issues portrayed in the assess section.

First and extremely accommodating advance was to make a duplicate of the three unique dataframes. I composed the codes to control the duplicates. If in the event that there was a mistake, I could create a new copy from the original.

At whatever point I committed an error, I could make another duplicate of the dataframes and keep taking a shot at the cleaning part.

There were two or three cleaning steps that were challenging. One of them was in the picture prediction table. I needed to make a 'nested if' inside a function so as to catch the first correct prediction of the kind of dog. The original table had three predictions and confidence levels. I sifted this into one column for dog type and one column for confidence level.

Other intriguing cleaning code was to melt the dog stages in a single column rather than four columns as original introduced in twitter archive.

One challenging cleaning step was the point at which I needed to address a few numerators that were actual decimals. This issue was drawn out into account after the first Udacity review. Utilizing Google sheets and visual assessment was not adequate to validate those decimals. Hence, I needed to run a code so as to check those actual tweets.

Conclusion

Data wrangling is a center ability that whoever handles information ought to be acquainted with. I have utilized Python programming language and a portion of its packages. There are a few points of interest of this tool (when contrasted with for example Google sheets) that is utilized by a lot of data scientists.

For gathering data there are a few bundles that help scratching information off the web, that help utilizing APIs to gather information (Tweepy for Twitter) or to speak with SQL databases.

It is solid in managing large information (obviously better than Google sheets).

It can manage an enormous assortment of information (unstructured data like JSON or additionally organized information from ERP/SQL databases.

It is simple to archive each single step and if necessary re-run each single step. Consequently, one can leave an ideal review trail.

One can re-run analysis every period. Accordingly, we could really re-run the dog analysis consistently with considerably less effort and time since I have set it up once.