UDACITY

Wrangle Report

Yousef Majeed

This report is for the wrangling data project (fourth project) in the Data Analyst Nanodegree Program that running by Udacity corporate.

Data Analyst Nanodegree

May-29, 2019

Overview

In this report I will describe the data wrangling process performed in the project named #Data Wrangling - Enhanced Twitter Archive.

Data wrangling process consists of 3 steps:

- Gathering data
- Assessing data
- Cleaning data

Gathering Data

Gathering data is the first step of data wrangling. For this project the data needed to be gathered was the following as described below in a Jupyter Notebook titled wrangle_act.ipynb:

About the Dataset(s)

The dataset I'll be wrangling is the tweet archive of Twitter user @dog_rates (https://twitter.com/dog_rates), also known as WeRateDogs. This archive/dataset consists of 2356 basic tweet data from November, 2015 to August, 2017. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog.

Based on the images in the above dataset (i.e. WeRateDogs Twitter archive), another dataset is created which consists of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images). Though no wrangling will be done directly on this image predictions dataset, it will definitely provide some additional data for our main tweet archive dataset.

1. ***Gather Twitter archive CSV file:*** The WeRateDogs Twitter archive stored in a csv file: twitter_archive_enhanced.csv. The WeRateDogs Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. One column the archive does contain though: each tweet's text, which I was used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced."
2. ***Gather tweet image predictions:*** The tweet image predictions file, i.e., what breed of dog (or another object, animal, etc.) 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 the followingURL:https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv
3. ***Gather data from Twitter API:*** Additional Data via the Twitter API: each tweet's retweet count and favorite ("like") count. Using the tweet IDs in the WeRateDogs Twitter archive, Twitter API was queried for each tweet's JSON data using Python's Tweepy library and stored in a file called tweet_json.txt file.

Gathering data for this project was done in Python Jupyter Notebook using pandas, requests and tweepy libraries. What I found challenging is using tweepy library to query Tweepy API, especially because it's the first time using such library, it's the first-time accessing Tweeter via secret keys and access. It took a while to download the data and many attempts and many failures, but in the end, I managed to extract it successfully.

Assessing Data

Assessing data is the second step in data wrangling. Assessing means inspecting the dataset for two things: data quality issues (i.e. content issues) and lack of tidiness (i.e. structural issues).

In this project I found the following: I assessed the datasets visually and programmatically and I found the following issues:

First of all, I was able to identify 2 quality issues just by going through the Key Points in the Project Motivation page.

Visual Assessment

I opened the twitter_archive_enhanced.csv and image_predictions.tsv in Excel and scrolled through them, looking for quality and tidiness issues. I was able to spot the following 2 quality and 2 tidiness issues:

- **Quality:** unnecessary html tags in source column of twitter archive in place of utility name e.g. Twitter for iPhone
- **Quality:** text column of twitter archive contains untruncated text instead of displayable text
- **Tidiness:** doggo, floofer, pupper and puppo columns in arc_df table should be merged into one column named "stage"
- **Tidiness:** Twitter archive data without any duplicates (i.e. retweets) will have empty retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp columns, which can be dropped

Programmatic Assessment

I used pandas' info method on arch_df to spot erroneous datatypes and other quality issues, if any. Then I used value_counts method on rating_numerator, rating_denominator and name columns to look up the range of their values and its distribution. Also, to verify 1 tidiness issue that I found during the visual assessment, I queried the archive dataframe to see if any of its tweets has more than one dog-stage mentioned. This entire activity helped me to identify the following 7 quality issues.

- contains retweets and therefore, duplicates
- many tweet_id(s) of arch_df table are missing in images_df (image predictions) table

- erroneous datatypes (in_reply_to_status_id, in_reply_to_user_id and timestamp columns)
- rating_denominator column has values higher than 10
- erroneous dog names starting with lowercase characters (e.g. a, an, actually, by)
- some records have more than one dog stage

The info method on the other 2 dataframes (images_df and tweets_df) didn't reveal any quality issues. However, after taking a look at the sample of each of these dataframes, I was able to identify the following 2 tidiness issues:

- the three dataframe should be merged into one master dataframe
- "id" column name from tweets_df dataframe not aligned with the rest of data frames

Cleaning Data

As all the quality and tidiness issues were related to arch_df table, I created copy of all three tables and named it with _clean. For each quality/tidiness issue, I performed the programmatic data cleaning process in 3 stages - Define, Code & Test. During the cleaning process, I converted the datatypes of source and newly created stage columns of arch_df_clean to category datatype. Then I merged all the three dataframe into one “twitter_master” dataframe using the tweet_id.

Storing Data

After the completion of the cleaning process, I stored the twitter_master DataFrame in twitter_archive_master.csv file.