wrangle_report

September 13, 2020

1 Data Wrangling Project

by Mohammed Elshafei

1.1 WeRateDogs Twitter Data

1.1.1 Introduction

For this project, we have been asked to wrangle data from twitter account "WeRateDogs" and analyze and provide insights with suitable visualizations. Using the built in Jupyter Notebook workspace provided by Udacity, the wrangling activities were conducted. Before starting with the data gathering, pythong libraries are imported which will be helpful for data gathering, cleaning, storing & visualizing.

1.1.2 Gathering Data

Data for this project was gathered using three different methods.

1- Manually downloading the "tweet-archive-enhanced.CSV" file from Project Details page and uploading to the Jupyter Notebook workspace provided.

2- Downloading the image_predictions.tsv file using python's requests library.

3- Using Twitter API to gather additional data of tweets using tweepy library and stored to a file called json_data.txt.

After gathering the data, the data is casted to a pandas dataframes using pandas.read_csv method.

1.1.3 Assessing Data

After visually & programatically assessing the data, the following quality and tidiness issues appeared which require cleaning and fixing before analyzing the data. Visual assessment was done using excel, while programatic assessment was done using python pandas methods.

Quality Issues:

df_tweets:

- Column timestamp dtype is string
- Column retweeted_status_timestamp dtype is string

- Some tweets are retweets
- Some tweets are not dog ratings
- doggo, floofer, pupper, puppo have NaN values depicted as none
- alot of no name values in name column
- "None" is not a name of dog in names column
- some columns with all entries Nan values

df_json_data:

- Columns contributors, coordinates, geo are empty columns.
- Column id name inconsistant with columns name tweet_id in df_tweets and df_image_predict
- column id (tweet_id) not at begining of table
- df_json_data table has multiple representations of id in int64 and object dtypes
- df_json all columns except retweet_count, favourate_count, id are irreleavent

df_image_predict

- some predictions show that the tweeted image is not a dog.
- entries which have 4 as img_num or wrong entry img_num value

Tidiness Issues:

- df_tweets doggo, floofer, pupper, puppo are not variables
- tweet_id is spread across all data frames

1.1.4 Cleaning Data

With the above issues that are a result of the assess task of this project, below are the cleaning & fixing methods used for the subject data sets.

1.1.5 First --> Define

Clean df_tweets

Quality Issues:

- change data type for timestamp from string to datetime
- change data type for retweeted_status_timestamp from string to datetime
- Drop entries (tweets) which begin with "RT @" as they are retweets & Dropping tweets where values of columns in_reply_to_status_id are non null

- Drop tweets that are not dog ratings by checking the text of the tweet for strings "we only rate dogs" & "don't rate"
- doggo, floofer, pupper, puppo none entires change to empty strings
- Change not names "lowercase" entires in name column to Nan
- Change "None" to Nan in names column
- drop columns with all entries that are Nan values

Clean df_json_data:

- drop empty columns from data set df_json_data
- change column name id in df_json_data to tweet_id
- rearrange column tweet_id to become the first column in the table
- drop multiple representations of the same information
- Drop irreleavnet columns from df_json_data

Clean df_image_predict

- drop entries that dont have dog images by passing condtion p1_dog, p2_dog, p3_dog False entries.
- fix entries which have 4 as img_num or wrong img_num value by making a new column and iterating through the confidence levels of each image.

Clean Tidiness Issues:

- combine df_tweets doggo, floofer, pupper, puppo in to "stage" column
- join df_tweets with df_json_data & df_image_predict

1.1.6 Second --> Code & Test

Before Coding, a copy of the data sets where taken. And Using methods like; drop, to_datetime, astype(), regex, tolist, merge and others to clean and fix the above issues with the data sets. Each step while cleaning was followed by the test codes like info(), describe(), value_counts() and others.

1.1.7 Store

Stored the clean data set in twitter_archive_master.csv

1.1.8 Analyze

Using max(), min(), loc(), mean() was able to gather some insights

1.1.9 Visualize

using the matplotlib library vizualized some attributes which have further shed light on other insights.