REPORTING: WRANGLE_REPORT ON WERATEDOGS

BY: Timothy Yaji

Contents

Introduction

Real-world data rarely come clean. Using Python and its libraries, i will gathered data from various sources and in a variety of formats, assess its quality and tidiness, then clean it this is called data wrangling.

The dataset that I wrangled 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. Details here

Software Used for Wrangling

The entirety of this wrangling project was in Jupyter Notebook with the following packages (libraries) installed: pandas, NumPy, requests, tweepy ,JSON, os, and seaborn.

Step1: Data Gathering

In this project we used three different pieces of data from various sources using different methods to gather each data for the purpose of wrangling.

- **The WeRateDogs Twitter archive** I am given this file in my Udacity work space. I downloaded this file manually by clicking the following link: twitter_archive_enhanced.csv. Once it was downloaded and saved in C:/Users/USER/3D Objects/Udacity_Project_02/wrangling_analyzing_data/twitter-archive-enhanced.csv, i upload it and read the data into a pandas DataFrame.

- **The tweet image predictions** This file (image_predictions.tsv) is hosted on Udacity's servers and i downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

- **Additional data from the Twitter API** I gather each tweet's retweet count and favorite ("like") count a Using the tweet IDs in the WeRateDogs Twitter archive, I query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Each tweet's JSON data was then written to its own line and read as .txt file line by line into a pandas DataFrame with tweet ID, retweet count, and favorite count.

Step 2: Assessing Data

After gathering all three pieces of data. I assessed the data using the following two types of assessment: **Visual assessment:** Each piece of gathered data was displayed in the Jupyter Notebook for visual assessment purposes. This was achieved by setting the display limits in pandas DataFrame (pd.set_option("display.max_colwidth",150) and pd.options.display.max_rows = 3000) **Programmatic assessment:** I used pandas' functions and/or methods to assess the data.

In order to meet project specifications, i made sure that:

- Only original ratings (no retweets) that have images were retained.
- I gathered the tweets and image predictions for these tweets up to August 1st, 2017 only.
- The Following issues were identified:

Quality Issues:

1. Some records are retweets or replies, the information to identify them can be found in the following columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp.

2. tweet_id,for all the three tables are stored as int instead of string/object type and timestamp column stored as string/object type instead of datetime data type in Df_twitter_archive.

3. Source column is a html element with tweet source wrapped in the text part of the element.

4. Inapproprate names ('a','an','all'etc in lowercase letter) in name column in Df_twitter_archive.

5. They is rating_denominator that has value zero,this does not make sense mathematically rating_denominator is zero, this does not make sense mathematically and also observations with decimal ratings.

6. Null values in expanded_urls in Df_twitter_archive

7. Text column has retweet combined with url.

8. The different predictions (p1, p2, p3) and their respective confidence level (p1_conf, p2_conf, p3_conf) columns with predicted breed (p1_dog, p2_dog, p3_dog) can be reduced into two columns to contain dog_breed and confidence_probability variables.

9. None descriptive column names

Tidiness issues

1. Tweet_id stored in multiple tables i.e(Twitter_archive, image_predictions and twitterApi_data should be one table).

2. doggo ,floofer, pupper and puppo column are dog stages which should be one column but there are in separate columns in the twitter_archive dataset.

Step 3: Cleaning Data

In this section, we will clean all the issues documented while assessing. This also includes merging individual pieces of data according to the rules of tidy data.

During the cleaning process i used the defined Define-Code-Test Framework to clearly document the process in this sequence:

- ❖ I make a copy of the original data sets.
- ❖ Define-Code-Test

a. Quality Issues:

Issue #1: Some records are retweets or replies, the information to identify them can be found in the following columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp, but we need only original tweet by project specifications.

Solution: I Drop records where retweeted_status_id and in_reply_to_status_id are not null from df_archive_clean. and then drop the redundant retweets and reply columns.

Issue #2*:* tweet_id, and timestamp is stored as int instead of string/object type and timestamp column stored as string/object type instead of datetime data type in Df_twitter_archive *solution:* I convert the tweet_id column to string/object type using pandas.astype() and the timestamp column to a datetime object using pandas.to_datetime()

Issue #3: Source column is a html element with tweet source wrapped in the text part of the element. we need only the tweet source.

Solution: Extract source from which tweets were made using regex with str.extract functions

Issue #4: Inapproprate names ('a','an','all'etc in lowercase letter) in name column in Df_twitter_archive.

Solution: Inappropriete names such as a, an, like, by, old, all etc are written in lowercase letters. We will use regular expression to get all lower-case names and drop those that are not real dogs and replace the remaining with None if there is any.

Issue #5*:* They is rating_denominator that has value zero,this does not make sense mathematically

Solution: Search for rating_denominator whose value is zero and correct it with appropriate rating of 10 gotten if it still exists.

Issue #6*:* Null values in expanded_urls in Df_twitter_archive

Solution: I dropped the column.

Issue #7: -Text column has tweets text combined with url.

Solution:

- I Extract the text from the column using regex and str.replace() and save into a tweets_text column

- Extract url and save in new column, "tweet_url".

- Account for missing urls by replacing them with None

- Drop text column.

Issue #8:

The different predictions (p1, p2, p3) and their respective confidence level (p1_conf, p2_conf, p3_conf) columns with predicted breed (p1_dog, p2_dog, p3_dog) can be reduced into two columns to contain dog_breed and confidence_probability variables

Solution:

- create empty lists to save the best value from each row in the dataset

- create a function that will iterates through prediction columns to find the best prediction and confidence probability

- Apply our breed_confidence function on our df_master_clean

- Assign the values in dog_breed_list into new columns in our df_master_clean

- Drop columns rename columnscolumns=['p1', 'p1_conf', 'p1_dog', 'p2','p2_conf', 'p2_dog', 'p3', 'p3_conf', 'p3_dog'])from the data set

b. Tidiness issues

Issue #1:

tweet_id stored in multiple tables i.e (Twitter_archive, image_predictions and twitterApi_data should be one table)

Solution:

I Merged all the three datasets into one and called it df_master_clean

Issue #2:

doggo, floofer, pupper and puppo column are dog stages which should be one column but there are in separate columns in the twitter archive dataset.

Solution: I combined the four columns doggo, floofer, pepper, and Puppo into column dog_stage.

Step4: Storing Data

I Save gathered, assessed, and cleaned the twitter archive data and saved master dataset to a CSV file with the name "twitter_archive_master.csv".