WRANGLE AND ANALYZE DATA – WRANGLE REPORT

JANUARY 3, 2019

UDACITY
Authored by: Jolene Yao

Table of Contents

Objective

For project 7 of Udacity's Data Analyst Nanodegree (DAND), our objective is to "wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations".

Project Details

In this project, we are going to gather data from a variety of sources, assess it, and clean it. This is a process called wrangling data. We will wrangle data from twitter's @dog_rates profile, also known as WeRateDogs. This is a very popular twitter and we owe the popularity of this to cute dog pictures and their high ratings, usually more than 10 out of a denominator of 10.

The enhanced twitter archive is given to us in a csv file. This contains data that has been extracted from the twitter archive such as dog rating, name, and "stage". These are just some examples of the column identifiers we will be using. The stages of a dog in the WeRateDog culture range from puppo to pupper, floofer, and doggo. Not all of the extracted pieces of information are clean so we will wrangle this enhanced twitter archive.

In addition to the twitter_archive_enhanced.csv file, we will wrangle the twitter api for retweet_count and favorite_count. We will query most recent 3000 tweets according that we have for tweet_id in the twitter_archive_enhanced.csv file. This is achieved with setting up a twitter account, twitter developer account, and generating access tokens and keys.

Last source of data we are going to gather from is the image predictions file. This was created through a neural network which predicted the breed of dog based on the image of the tweet. There are three predictions of the image. Columns in this file include a prediction of the dog breed, confidence level of the prediction, and whether the prediction was true or false.

Key Points

There are a few key points we want to keep in mind while working on this project. They are as follows.

- We want only the original tweets and not the retweets. The enhanced twitter archive file has roughly 5000 tweets and we are going to lose some entries after removing the retweets.
- We only want 8 quality issues and 2 tidiness issues, even though there are more than that. Creating a perfectly clean file would take too long of a process and for the sake of this project, we will demonstrate skills we have learned.
- We do not need to clean up the rating numerators that are higher than denominators since this is part of the rating system and culture of WeRateDogs
- We do not need to gather the tweets beyond August 1st, 2017. We will not be able to gather the image predictions for the tweets beyond this date since we do not have access to the algorithm used.
- We will not include access tokens and keys in our final submission

Data Wrangling Process

1. Gathering Data

There are three sources of data that we are going to gather from – a file given to us from the twitter archive, the twitter api, and an image predictions file.

- The **twitter_archive_enhanced.csv** file is given to us through a download link on Udacity
- The twitter image predictions file predicts which dog breed is which based on its tweet image. This was predicted using an algorithm through Udacity's neural network and can be downloaded programmatically through the following link or retrieved using python's Requests library:
  https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv
- Tweet data, such as retweet count and favorite count, will be queried from twitter's api. We will need to query this JSON data into a file called **tweet_json.txt**. We will need to use our twitter developer account and access keys and tokens for this step.

2. Assessing Data

We will assess data programmatically and visually to find 8 quality issues and 2 tidiness issues. Keep in mind the key points in the previous page while finding these issues. All these issues will be defined, coded and tested in the wrangle_act.ipynb.

3. Cleaning Data

The 8 quality issues and 2 tidiness issues I have found are as follows:

QUALITY ISSUES:

Issue #1 Dog names "None", "a", "the" and "an" should be replaced.

- Clean this by using the replace function on the df_twitter_archive_clean["name"] column

Issue #2 Timestamp format is incorrect. Change timestamp data type from object to datetime. The +0000 will drop

- Clean this by using the to_datetime function on the df_twitter_archive_clean[“timestamp”] column

Issue #3 Data type of tweet_id should be string, not object.

- Clean this by using the astype() function on df_twitter_archive_clean[“tweet_id”] column

Issue #4 The expanded_urls column has null values that need to be dropped.

- Clean this by using the .dropna() function on df_twitter_archive_clean[“expanded_urls”] column

Issue #5 The rating_denominator column should only have the denominator 10

- Clean this by setting df_twitter_archive_clean["rating_denominator"] = 10

Issue #6 The numerator column should exclude numbers less than 10 but just for an example, we will only exclude the values 0, 1, and 2.

- Clean this by setting the rating_numerator column != 0, 1, and 2

Issue #7 Extract numerator ratings from the tweet text and fill the missing values in rating_numerator column by finding the pattern ##.##

- Clean this by using a regex pattern in the format of ##.## on df_twitter_archive_clean[“rating_numerator”] column

Issue #8 p1, p2, and p3 columns have uppercased and lowercased dog breed names that need to be standardized

- Clean this by using .str.lower() on the df_twitter_archive_clean p1, p2, and p3 columns

TIDINESS ISSUES:

Issue #1 Remove the retweeted images because we only want original tweets

- Clean this by using the drop() function on the columns in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp.

Issue #2 Combine all three datasets into one dataset called twitter_master_df.

- Clean this by combining the df_twitter_archive_clean dataset with image_predictions_clean dataset. Then we will combine that dataset with the df_json_clean dataset to make one master dataset. Use the merge function with a left join and on tweet_id

Issue #3 There are extra issues cleaned in the wrangle_act.ipynb file for fun

Storing Data

We will store the main dataframe in a file called **twitter_archive_master.csv** or if we have multiple dataframes, then the biggest one is named twitter_archive_master.csv. In this project we have included a second dataframe called df_2.csv and df_2_new for the act report's visualizations.