JANUARY 4, 2020

PROJECT #4: WRANGLLE AND ANALYZE DATA

WRANGLE REPORT

MASHAEL ALSAADAN

Introduction

Real-world data rarely comes clean. Using Python and its libraries, I gathered data from a variety of sources and in a variety of formats, assess its quality and tidiness, then cleaned it. This is called data wrangling.

The dataset that I have 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.

Wrangling Process

The process consists of:

- Gathering data
- Assessing data
  - Quality issues
  - Tidiness issues
- Cleaning data
  - Define
  - Code
  - Test

1. Gathering Data

Data was gathered from three different resources:

1.1. WeRateDogs Twitter Archive file that I was given (manually downloaded):

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 is used to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo) to make this Twitter archive "enhanced." Of the 5000+ tweets, tweets were filtered with ratings only (there are 2356).

1.2. The Tweet image predictions which is hosted on Udacity's server (programmatically downloaded):

Every image in the WeRateDogs Twitter archive were ran through a neural network that can classify breeds of dogs*. The results: a table full 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).

1.3. Using python's library tweepy and twitter API I retrieved each tweets' retweet and favorite counts:

Additional data (retweet count and favorite count) were gathered from Twitter's API.

2. Assessing Data

After gathering each of the above pieces of data, I assessed them both visually and programmatically for quality and tidiness issues. Below are the issues that were found in the three tables `twitter_archive` table, `image_predictions` table and `tweet_json` table:

- Quality Issues (Dirty Data):

  Content issues: completeness, validity, accuracy and consistency

  `twitter_archive` table:

  - Column tweet_id int instead of str (in all tables)
  - The dataset includes retweets (when the columns retweeted_status_id, retweeted_status_user_id and retweeted_status timestamp are not NaN it means they are retweets)
  - Missing data 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
  - Some dogs' names contain invalid names such as: a, an or the
  - Timestamp datatype is object instead of datetime
  - Source column includes html tags

`image_predictions` table:

- There is no column to detect the breed of the dog with highest confidence
- P1, p2 and p3 columns are inconsistent in capitalizations
- P1, p2 and p3 columns have invalid data (banana, paper_towel and bagel)

- Tidiness Issues (Messy Data):

  Contains structural issues, tidy datasets have specific structure:

  `twitter_archive` table:

  - The four columns (doggo, floofer, pupper and puppo) related to each other

  `Image_predictions` table:

  - Table's data related to the twitter_archive table

  `tweet_json` table:

  - Table's data related to the twitter_archive table

3. Cleaning Data

In this section, I cleaned the data using this process on copies of the dataset:

- Define
- Code
- Test

The following are the actions that were taken to clean the data:

1. Merge all three tables into one table
2. Convert tweet_id column from int to object (str) datatype
3. Remove tweets with column retweeted_status_id that isn't NaN
4. Remove the columns since we don't need them
5. Change the names with lower letters to None
6. Convert timestamp to datetime datatype
7. Clean source column from html tags
8. Create a new column with the breed of dogs based on the highest confidence rate

9. Capitalize the derived column (breed)
10. Remove rows with dog false prediction
11. Combine the four columns into one
12. Drop columns we don't need (img_num, expanded_urls)