Data Wrangling Report

Introduction

The purpose of this project is to put in practice what I learned in data wrangling data section from Udacity Data Analysis Nanodegree program. The dataset that is 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.

This report briefly describes my wrangling efforts.

Project Details

The tasks of this project are as follows:

- Gathering data
- Assessing data
- Cleaning data
- Storing data

Gathering Data:

- **Introduction about the dataset:**
  - ✓ 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 2333 basic tweet data from November, 2015 to August, 2017.
  - ✓ Based on the images in the above dataset (i.e. WeRateDogs Twitter archive), another dataset is created which consists of image predictions 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.

- **Twitter archive CSV file**

  Using the link provided by Udacity, I downloaded the WeRateDogs Twitter archive manually as twitter_archive_enhanced.csv *(https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archiveenhanced/twitter-archive-enhanced.csv)* file and imported this file into a data frame.

- **Tweet image predictions**

  The archive contained predictions of the breed in each tweet by a neural network designed by Udacity. I downloaded the tweet image predictions file hosted on Udacity's servers programmatically using Python's Requests library and saved it locally to image_predictions.tsv file. Then, I imported this file into a dataframe.

- **Twitter API & JSON**

  Using the tweet IDs in the WeRateDogs Twitter archive, I accessed the entire data for every tweet from Twitter API and stored every tweet's entire set of JSON data in a file called tweet_json.txt file. Created a dataframe from this JSON file including tweet ID, favorite count, retweet count, followers count, friends count, source, retweeted status and url.

Assessing Data:

Visual Assessment

- In order to visually assess the data, I first printed the three entire dataframes separate in Jupyter Notebook. After that, I glanced at the CSV files in excel.
- 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 followng quality and tidiness issues:
    - ✓ **Quality**: unnecessary html tags in source column of twitter archive in place of utility name e.g.<a href=""http://twitter.com/download/iphone"" rel=""nofollow"">Twitter for iPhone</a>
    - ✓ **Tidiness**: doggo, floofer, pupper and puppo columns in twitter_archive_df table should be merged into one column named "stage".
    - ✓ **Tidiness**: Twitter_archive_df 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 twitter_archive_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. I also used sample and duplicated methods to find some issues.

This entire activity helped me to identify the following 6 quality issues.

- ✓ Keep the original ratings that have images. Churn out the retweets.
- ✓ Erroneousdtypes(in_reply_to_status_id,in_reply_to_user_id,timestamp columns,etc)
- ✓ The numerator and denominator columns have invalid values
- ✓ Name column has invalid names i.e. a,an,'None' and less than 3 characters.
- ✓ In several columns, null objects are non-null(None to NaN).
- ✓ Unnecessary html tags in source column in place of utility name.

There are a few other quality issues I identified during the assessment process. They are duly listed in wrangle_act.ipynb file.

Cleaning Data:

- This part of the data wrangling was divided in three parts: Define, code and test the code. These three steps were on each of the quality and tidiness issues described in the assess section.
- Firstly I created a copy of the three original dataframes. I wrote the codes to manipulate the copies. If there was an error, I could create a new copy from the original. This was also suggested by the instructors in the lesson.
- For me, the most challenging step in the cleaning process was working with the numerator and denominator columns. There were a few exceptions where dogs were given unusually high ratings. These were counted as outliers. But for the vast majority of them, I wrote a code that checked for decimal numerators in the actual tweets.
- Lastly, there were four columns for the dog stages named doggo, floofer, pupper and puppo in the original archive. And as per the rules of tidy data, I melted all the four dog stages in a single column named stage.

Storing Data:

After the completion of the whole cleaning process, I stored the twitter_archive_df_clean dataframe in twitter_archive_master.csv file. This file could be used for further analysis. This is not the end of the cleaning process. Since data wrangling is a continuous process, we may have to jump to step 1 again.

Conclusion

Data wrangling is a very important skill that every data analyst should be proficient with. The Twitter account WeRateDogs (@dog_rates) is devoted to humorously reviewing pictures of dogs doing adorable poses. Dogs are rated on a scale of one to ten, but are invariably given ratings in excess of the maximum. It has acquired over 8.19 million followers.