Wrangle Report

Introduction

This project mainly focus on the data wrangle process (gathering data, assessing data and cleaning data). The dataset for this project 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. The ratings almost always have a denominator of 10.

This report briefly describes my wrangling efforts on this project.

Wrangling Details

Gathering Data for this Project

The project consists of three different datasets. Gather each of the three pieces of data as described below in a Jupyter Notebook titled wrangle_act.ipynb:

1. **Twitter archive:** this dataset is provided by the Udacity and called **twitter_archive_enhanced.csv**. I just downloaded it using the link and imported it as DataFrame by Pandas.

2. **The tweet image predictions:** i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (**image_predictions.tsv**) is hosted on Udacity's servers and should be downloaded programmatically using the Requests library and the provided URL. Then it was imported as the DataFrame by Pandas.

3. **The tweet JSON data from Twitter's API: this file contains** each tweet's retweet count and favorite ("like") count, and additional data of interest. Given the tweet IDs in the WeRateDogs "**Twitter archive**", I call the Twitter API for each tweet's JSON data using Python's Tweepy library and store those JSON data in a file named **tweet_json.txt**. Tweet's JSON data are written line by line in the .txt file. Then read this .txt file line by line into a pandas DataFrame with tweet ID, retweet count, and favorite count.

Assessing Data

After the datasets had been imported in the Jupyter Notebook, I started assessing the data as below:

1. **Visual Assessment:** I both used the Jupyter Notebook and Excel to check the entire three datasets firstly to get the general idea of each dataset. However, there were some folded parts in three DataFrames' rows and columns when showing in the Jupyter Notebook.

2. **Programmatical Assessment:** since all three datasets are too large to assess comprehensively via visual assessment, they were assessed using pandas methods like .info(), .value_counts(), .duplicated(), .isnull() and other useful methods to further find the quality issues and tidiness issues.

Eventually, all the quality issues and tidiness issues were summarized correspondingly to each dataset for later cleaning process.

Quality issues

1. twitter_archive

   - erroneous data type('tweet_id'should be object not int, 'in_reply_to_status_id','in_reply_to_user_id', 'retweeted_status_id ' and 'retweeted_status_user_id'should be object not the float, 'timestamp'and 'retweeted_status_timestamp 'should be datetime not the object)
   - missing values in expanded_urls
   - incorrect rating_numerators (only take the number behind the decimal point of the original data in text column)
   - 23 wrong rating_denominator that not equal to 10
   - 745 dogs named none
   - some entries are retweets
   - some entries are wrong named(not the name in text)

2. image_prediction

   - erroneous data type('tweet_id'should be object not int)
   - the duplicates in jpg_url(66)
   - inconsistant upper case value in p1, p2, p3
   - missing data (only has 2075 entries instead of 2356

3. twitter_data

   - erroneous data type('tweet_id'should be object not int)
   - missing data (only has 2331 entries instead of 2356)

Tidiness issues

Three data frames twitter_archive, image_predictions, and twitter_data should be one (combined table) since all tables' entries are each describing one tweet

1. twitter_archive

   - one variable in four columns (doggo, floofer, pupper, and puppo)
   - retweeted data not the original

2. image_prediction

   - Create 1 column for image prediction and 1 column for confidence level

Cleaning Data

I divided each cleaning process into three parts: Define (the solution to address the issues found), code and test (confirm the code worked well and issues were solved).

Before the cleaning, I use .copy() to made three copy for each dataset to keep the original ones. In this case even if I made the mistake in code, I would not ruin the original dataset and affect the following analysis.

The most challenging part is to use regex and .str.extract() to retake the numerator of the rating in the text. There are both integers and floats data in the text, especially some rating have the same format as rating, which are just date or other information. So I only focus on the rating format that the denominator is 10 as introduced in project detail, and then generate the new column called rating_denominator_new for later analysis.

Some missing values I found should not be simply dropped, like expanded url and jpg url. I check the twitter website for the explanation and check some tweets manually, and it turns out that those missing urls were tweets without dog image or just retweets.

There are still some issues I found quite interesting, and I would try to address them later for much comprehensive analysis.

Last but not least, documenting is a really nice step in data wrangling since it is easy to forget the issues and redo the assessing or scrolling back is time-consuming.