Wrangle Report

Author: Otto Roberson¶

Table of Contents

Introduction

The purpose of this project is to put in practice what I've learned so far. The dataset that is wrangled is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. The commentary is intended to be humorous with part of the humor in the use of "illogical" ratings.

In order to better understand and provide a basic analysis of the data, four steps were performed, gather, assess, clean, and visualize:

1. Gather:

For this project, we gathered the data from three different resources and saved it initially in three separate files.

twitter_archive_enhanced.csv: A comma-delimited file which was provided to us as a local archive.
image_predictions.tsv: A tab-delimited file which was downloaded from Udacity's servers using the Requests library.
Twitter API & JSON: I created a developer account on Twitter, and then queried their API using the Tweepy library. Each tweet's set of JSON data was saved in a file called tweet_json.txt.

2. Assess:

During this step, I stored each of the files in a separate DataFrame using the Pandas library. I then used Panda's methods to evaluate the data. Some of the methods used for each of the DataFrames, represented by 'df', were:

- df.head()
- df.tail()
- df.info()
- df.sample()
- dr.str.contains()
- df.sort_values()

- df.value_counts()
- df.describe()
- df.unique()
- df.duplicated().sum()
- df.isnull.sum()

Data Selection

I filtered the data based on the following criteria:

- Don't include retweets, as including them would skew the data.
- Include only tweets with images.
- Only include original tweets, as there are reply tweets which produce multiple data points for the same dog.

I also filtered the issues with the data and selected to handle those most pertinent to my analysis. There is one notable issue that I decided not to treat, which is that there are numerous lower-case values in the `names` field which aren't really names. There are too many to drop unless I do it as a special case, after the rest of the data has been treated. It doesn't look as if it would add much value to the analysis otherwise.

3. Clean:

During this step I attempted to work iteratively, and structuring the process in logical steps to progress through the DataFrames. The results of this process where stored in the file twitter_archive_master.csv.

I addressed the following issues with the data:

- **Quality**
  - Rows without images in `expanded_url` need to be removed.
  - Rows with retweets need to be removed.
  - Retweeted columns need to be removed.
  - Some name values are `None`, should be changed to `NaN`.
  - Some ratings have decimals and the datatype needs to be changed.
  - The corresponding numerators in these ratings will need to be updated.
  - Wrong datatype in `timestamp`, should be changed to `datetime`.
  - Assign categorical datatype to `dog_stage`.
  - Assign string datatype to `tweet_id`, `in_reply_to_status_id`, `in_reply_to_user_id`.
- **Tidiness**
  - Extraneous dog stages, merge into one variable.

- Drop unneeded columns.
- Data is spread across three tables which can be combined.

4. Visualize

For this stage I employed the matplotlib.pyplot library, creating different graphs of the dataset to gain insights and reach conclusions.

Resources

> -twitter API tutorial
>
> -JSON resource with examples
>
> -python JSON encoder & decoder documentation
>
> -pandas.set
>
> -pandas str.contains
>
> -pandas str.extract
>
> -pandas.merge
>
> -pandas.loc
>
> -pandas.to_datetime
>
> -seaborn.set_context
>
> -twitter data visualization
>
> -More twitter data visualization
>
> -wordcloud