Wrangle and Analyze Data

Udacity Project

Executive Summary

Social media platforms are filled with animal love pages, and Twitter is no exception. The dataset for this project is the tweet archive of the Twitter account WeRateDogs (@dog_rates) which boasts 8.9 million followers. The account, started in 2015 by a college student, is home to dog "ratings." However, it could more so be characterized as a "cute" dog photo sharing, and "good-boy" moments. Nonetheless, we examine perform the following tasks in this project:

- Gathering the Data: CSVs, GET Request, Tweepy
- Assessing Data Quality Issues: Image Predictions, WeRateDogs Archive, RTs & Favorites
- Tidy Data Issues
- Cleaning the Data: Quality Issues, Tidy Issues

Gathering the Data

This project made use of acquiring data from multiple sources, downloaded local files, GET requests, and an API.

Local CSV Files

The WeRateDogs Twitter archive was provided as a downloadable .csv file from the course site and could be saved to the project directory for use. The archive contains basic tweet data for 5000+ of the account's tweets.

GET Request

The second data source for the project is a tab-separated file (.tsv) downloaded from a provided URL. A GET request is made with the requests library to return a response object. The response content is written to a file which is then converted into a pandas DataFrame.

Tweepy

The last data source for the project makes use of Tweepy, a Python library for accessing the Twitter API. After registering a developer profile with an account, API and access tokens can be created for a project application. These credentials are used by Tweepy methods to authenticate, and ultimately return a Status object for each available tweet_id in the archive data. Retweet and favorite counts were extracted from the json attribute of the Status object which contained this information.

Assessing Data Quality Issues

Image Prediction Data

1. [tweet_id] should be a string.
2. [p1], [p2] and [p3] predictions have underscores in the dog name predictions.

WeRateDogs Archive

3. Missing data in in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls fields.
4. [tweet_id] should be a string

5. [timestamp] should be in datetime format.
6. [name] field includes invalid dog names that are definite articles.
7. [text] field contains a URL of the dog image which is not informative textual information.

RTs and Favorites

8. [tweet_id] should be a string.

Tidy Data Issues

1. The retweet and favorite counts are in a separate table than the tweet archive table.
2. The [doggo], [floofer], [pupper], and [puppo] fields represent an original ranking that should be one column.

Cleaning the Data

Quality Issues

Image Prediction Data:

1. Change [tweet_id] to string with pandas Series astype( ).
2. Use applymap( ), lambda function, and string methods to fix dog breed predictions.

WeRateDogs Archive:

3. Remove cols with missing data using DataFrame drop( ).
4. Change [tweet_id] to string with pandas Series astype( ).
5. Use pd.to_datetime to change [timestamp] to dt.
6. Remove invalid dog names with Series replace( )
7. Remove URL from [text] field with str.replace( ) and regex pattern.

RTs and Favorites:

8. Change [tweet_id] to string.

Tidy Issues

1. Join archive data to retrieved RTs and favorite counts data by joining on index.
2. Merge “Dogtionary” columns into one column with str.extract( )