Data Wrangling of Tweets by WeRateDog

Vishruth Srinath 19th June 2018

This report describes and documents the wrangling effort for the data associated with WeRateDog's (@dog_rates) tweets. WeRateDogs is a Twitter account that rates dogs and includes a wholesome and funny comment. I not only wrangled different data sources into one clean and tidy table but also performed analyses and presented my insight.

The data wrangling and analysis has been performed using Python 3 in the *"wrangle_act.ipynb"* Jupyter notebook. This report with describe the process while following the structure in the notebook.

Data Wrangling

The wrangling was broken into the following 3 steps

Gathering

Data was gathered from the following three sources and loaded into corresponding Pandas dataframes (DF):

Data	**Original Source**	**Local File Storage**	**Important Tasks Performed**
Enhanced tweet archive	Local .CSV file	.CSV file	Load CSV to DF
Image predictions	.TSV file stored on Udacity server	.TSV file	- Read file from url and write to file - Load TSV to DF
Tweet details	Twitter API	.txt file with data serialized as JSON	- Use API to access tweet details - Write JSON to file - Read JSON from file into DF

Assessing

The data were assessed, primarily using programmatic assessments, and both quality and tidiness issues were listed.

Some of the techniques and methods used are listed below:

- Pandas dataframe and series indexing (.<column_name>, .iloc[], and .loc[]) and ability to pass Boolean series as masks
- Pandas dataframe and series methods: .head((), .info(), .value_counts(), .duplicated(), .isnull(), .sample()

In general, tidiness, such as the same observation type being in separate tables, or systemic quality issues, such as bad datatypes, were straightforward to identify. However, one-off or rare errors such as incorrectly parsed ratings for a few tweets, were harder to identify.

Cleaning

The data were cleaned, i.e., the issues identified in the assessment steps were addressed, in the following order:

- Missing Data
- Tidiness

- Other Quality Issues

Within each of the above three steps the issues were addressed in an order that resulted in the least amount of re-work and not in the order they were identified. Each cleaning operation was performed using the Define-Code-Test framework

Some of the techniques used in "Code" phase are listed below:

- Pandas dataframe and series indexing (.<column_name>, .iloc[], and .loc[]) and ability to filter using Boolean series as masks
- String manipulations with Pandas: str.extract(), str.title(), str.slice(), str.replace()
- Pandas dataframe manipulation: .merge(), .drop(), .apply(), .reset_index(), .set_index(), .replace(), .dropna()
-

Though the cleaning of tidiness or systemic quality issues resulted in more changes to the data or the data structure and sometimes required elaborate code, the code was more general purpose and the results easily tested. However, the one-off errors and issues that needed to be fixed needed very specific code that may not work for different data of the same type.

At the end of the cleaning process I had a tidy and clean dataframe.

Data Storage

The dataframe that resulted from the wrangling was stored as a CSV file, *"twitter_archive_master.csv"*.

Analysis and Visualization

I analysed the dataset for the following features:

- The accounts tweet rate over time
- The variation of the rating over time
- The accounts popularity over time
- The most popular dog breeds

Apart from the techniques and tools mentioned previously, I used the following for analysis and visualization:

- Pandas .groupby() method
- Matplotlib library and its pyplot feature: .figure(), .plot(), .ax()

Having had a lot of experience with data visualization in Pandas this section was less googling and more fun.