UDACITY

Return to "Data Analyst Nanodegree" in the classroom

Wrangle and Analyze Data

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear Student,

You have put dedicated effort into this project and it paid off. Congratulations on meeting all the specifications of the project! You have demonstrated a very good python coding skills and understanding of data wrangling process. You also did a fantastic job of incorporating suggestions from **the previous reviewer. As a different reviewer, I have left some additional comments.** If you are uploading this project to your portfolio or sharing it with your potential employer, it is a good idea to address these comments. It also gives you an opportunity to appreciate the complete essence of this project. Keep up all the great work you are doing.

Good luck with your future projects!

Note: I made the comments marked as **Suggested** to help you improve the project. It does not require you to resubmit the project. You have already passed the project. Congratulations!

Code Functionality and Readability

All project code is contained in a Jupyter Notebook named wrangle_act.ipynb and runs without errors.

Suggested:

Please note that the notebook you submit must be free of any kind of error. However, the notebook you submitted contained the following error. Please run your entire notebook and make sure there are no errors. In your notebook, in Menu bar > Cell > Run All.

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-71-30d3976dc01d> in <module>()
      1 #Code
----> 2 df_123c['popularity'] = df_123c['favorite_count'] + df_123c['retweet_count']

NameError: name 'df_123c' is not defined
```

The Jupyter Notebook has an intuitive, easy-to-follow logical structure. The code uses comments effectively and is interspersed with Jupyter Notebook Markdown cells. The steps of the data wrangling process (i.e. gather, assess, and clean) are clearly identified with comments or Markdown cells, as well.

Good job clearly identifying the steps of the data wrangling process in markdown cells. The notebook is structured well. This helps to easily follow your code.

Gathering Data

Data is successfully gathered:

- From at least the three (3) different sources on the Project Details page.
- In at least the three (3) different file formats on the Project Details page.

Each piece of data is imported into a separate pandas DataFrame at first.

Assessing Data

Two types of assessment are used:

- Visual assessment: each piece of gathered data is displayed in the Jupyter Notebook for visual assessment purposes. Once displayed, data can additionally be assessed in an external application (e.g. Excel, text editor).
- Programmatic assessment: pandas' functions and/or methods are used to assess the data.

At least eight (8) data quality issues and two (2) tidiness issues are detected, and include the issues to clean to satisfy the Project Motivation. Each issue is documented in one to a few sentences each.

Suggested:

Having 3 columns for dog breed prediction is not a tidiness issue, as these are indeed three different variables; algorithms' first, second and third most likely predictions. This should be addressed as a quality issue. Please note that although this issue seems to be to untidy in literal sense, it does not violate any rules of tidiness (in tidy data each variable forms a column, each observation forms a row, each type of observational unit forms a table).

Cleaning Data

The define, code, and test steps of the cleaning process are clearly documented.

Copies of the original pieces of data are made prior to cleaning.

All issues identified in the assess phase are successfully cleaned (if possible) using Python and pandas, and include the cleaning tasks required to satisfy the Project Motivation.

A tidy master dataset (or datasets, if appropriate) with all pieces of gathered data is created.

Suggested:

You only want original dog ratings (no retweets) that have images (a user can retweet their on tweet), so you need to remove all rows that have values (not blank or non-null) in retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp columns. As mentioned in the **Key Points** section of **Project. Wrangle and Analyze Data - Sublesson 2. Project Motivation** (link), this is an important quality issue to be addressed, as it has a direct bearing on your analysis.
Just removing the columns like retweeted_status_id is not going to address the retweet issue. You can remove these columns only after removing rows with retweets.

With regards to outliers, please note that all outliers are not bad outliers. Some of the outliers may indicate actual trend and you should retain such outliers. For instance, there is rating of 1776. This dog is rated on the independence day of USA so it is just a creative way to rate this dog very high. You should preserve such interesting data. That is why it is explicitly mentioned that **The fact that the rating numerators are greater than the denominators (in some cases extreme values) does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.**

One of the issue you cleaned is,

- The tweet-json dataframe has duplicates!

But your assessment shows that there are no duplicates. Please see below.

```
print('Number of duplicated rows are given below:')
print('{} for {}'.format(msg07, 'twitter-archive-enhanced'))
print('{} for {}'.format(msg08, 'image-predictions'))
print('{} for {}'.format(msg09, 'tweet-json'))

Number of duplicated rows are given below:
0 for twitter-archive-enhanced
0 for image-predictions
0 for tweet-json
```

As you correctly identified, there are non-dog names in **name** column such as 'a', 'such', 'the', 'just', 'getting' etc. Please note that these cases are in lowercase as a result of the way the names were extracted from the tweet text, whereas a proper dog name is capitalized. These observations can be used to handle this issue programmatically, but you handled this issue individually which is not robust and you missed many non-dog names.

Removing rows with denominator != 10 without further inspection is not a good idea. For instance, take a look at this tweet https://twitter.com/dog_rates/status/704054845121142784; in this case the denominator is intended to be 50, because there are 5 puppers. Project instruction says denominator is **almost always** 10, it does not say it is **always** 10.

Did you notice that certain tweets have more than one stage (previous reviewer also noted this). For instance take a look at the following tweet, where both doggo and pupper is present.

808106460588765000			2016-12	<a href=	Here we have Burke (pupper) and Dexter (doggo). Pupper wants to be exactly like doggo. Both 12/10 would pet at same time https://t.co/ANBpEYHaho					https://t	12	10	None	doggo	None	pupper

This is because some tweets may have more than one dog with different stages (https://twitter.com/dog_rates/status/808106460588765185/photo/1). When there are multiple stages for a tweet (e.g., doggo and pupper) like this, your code capture only one stage. Instead you should capture all the stages as a list delimited by comma (e.g., 'doggo, pupper'), or you can capture them something like 'multiple_stages'. However, there is one more issue. In certain cases, although there are more than one stage, if you look at the text, there is supposed to be only one stage. For example, take a careful look at the following tweet;

854010172552949000			2017-04	<a href=	At first I thought this was a shy doggo, but it's actually a Rare Canadian Floofer Owl. Amateurs would confuse the two. 11/10 only send dogs https://t.co/TXdT3tmuYk					https://t	11	10	None	doggo	floofer	None

This is supposed to be floofer, but it is captured as doggo and floofer, which is wrong. So if you want to clean this column perfectly, you may have to do some manual cleaning. This shows how data wrangling can get really complicated on occasions.

Storing and Acting on Wrangled Data

Students will save their gathered, assessed, and cleaned master dataset(s) to a CSV file or a SQLite database.

The master dataset is analyzed using pandas or SQL in the Jupyter Notebook and at least three (3) separate insights are produced.

At least one (1) labeled visualization is produced in the Jupyter Notebook using Python's plotting libraries or in Tableau.

Students must make it clear in their wrangling work that they assessed and cleaned (if necessary) the data upon which the analyses and visualizations are based.

Report

The student's wrangling efforts are briefly described. This document (wrangle_report.pdf or wrangle_report.html) is concise and approximately 300-600 words in length.

The three (3) or more insights the student found are communicated. At least one (1) visualization is included.

This document (act_report.pdf or act_report.html) is at least 250 words in length.

Project Files

The following files (with identical filenames) are included:

- wrangle_act.ipynb
- wrangle_report.pdf or wrangle_report.html
- act_report.pdf or act_report.html

All dataset files are included, including the stored master dataset(s), with filenames and extensions as specified on the Project Submission page.

DOWNLOAD PROJECT

RETURN TO PATH