UDACITY

‹ Return to Classroom

Wrangle and Analyze Data

REVIEW

HISTORY

Meets Specifications

Excellent work incorporating all previous suggestions. It is great that you have cleaned the dataset quite thoroughly, including the extraction of dog stages and handling multiple dog stages. This is a challenging project, but you managed to push through.

Now, I know that this advice may sound unconventional, but a great way to appreciate the importance of data wrangling, and to learn further, is to try doing one or two Kaggle competitions. It may seem like a daunting task, but this is very good to get you to learn and get used to wrangling techniques that are (somewhat) relevant to real-world tasks. Oftentimes, you will see competition participants share their wrangling routines; those are invaluable learning materials.

Good luck in your learning journey, and I wish you a wonderful career in the field of data analysis!

Code Functionality and Readability

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

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.

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.

There are now more than 8 quality and 2 tidiness issues listed and cleaned in your work. It is also great that you have written a summary of the issues cleaned, making the notebook easier to read. Well done!

Note: Apologies for the confusion, but I can confirm that removing unneeded columns is indeed a quality issue. I have reported this inconsistency to the coaches for further investigation, but you may rest assured the fact is correct.

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.

DataFrame objects had been copied before cleaning, and a final cleaned dataset was created and filled with the cleaned data. All the important issues have also been cleaned, excellent work here.
Below, I have some suggestions to improve your wrangling skills even further. Please take a moment to read each of them and, if possible, incorporate them into your project.

(Optional) The rating value 0 is a correct rating

You have removed the ratings with the value 0 in numerator or denominator. This is not entirely correct as 0 is a valid value, at least for the numerator.

(Optional) Find and update incorrect ratings - Quality issue

For rows with rating denominator != 10, there are cases where they are valid ratings and there are also invalid ones. The only way to find out (with what you have learned so far) is by manually reading the text. Fortunately, we do not have that many of those, so this is still doable. I'll give you three examples of possible scenarios:

- Tweet ID 810984652412424192. Text: "Meet Sam. She smiles 24/7 & secretly aspires to be a reindeer. \nKeep Sam smiling by clicking and sharing this link:\nhttps://t.co/98tB8y7y7t https://t.co/LouL5vdvxx". Extracted rating numerator and denumerator were 24 and 7. This is not correct. There shouldn't be any rating in this tweet.
- Tweet ID 835246439529840640. Text: "@jonnysun @Lin_Manuel ok jomny I know you're excited but 960/00 isn't a valid rating, 13/10 is tho". Extracted rating numerator and denumerator were 24 and 7. Correct ones should be 13 and 10
- Tweet ID 820690176645140481. Text: "The floofs have been released I repeat the floofs have been released. 84/70 https://t.co/NIYC820tmd". The extracted rating numerators and numerators of 84 and 70 are both correct.

(Optional) Ratings with decimal values incorrectly extracted - Quality issue

The current pipeline captures incorrect values when rating numerators contain decimals. You have mentioned it was not possible to clean this programmatically. In reality, however, it is possible to do just that to a certain degree. For example, here is a value from one observation with tweet id 786709082849828864:

> "This is Logan, the Chow who lived. He solemnly swears he's up to lots of good. H*ckin magical af 9.75/10 https://t.co/yBO5wuqaPS"

Currently, the value 75 would be captured as the rating numerator. Try to capture the entire value from the text instead. Here is a code snippet as an example, where `df` here is the twitter archive dataset:

```
ratings = df.text.str.extract('((?:\d+\.)?\d+)\/(\d+)', expand=True)
```

`ratings` series object will then contain all rating numerators with decimals and rating denominators (without decimals). The next step is to extract only the numerators and denumerators from `ratings` dataframe, and then update your dataset's fields with extracted rating numerators and denominators (**NOTE: Do not forget to convert the field datatype into Float, `astype` function may be used here**):

```
df.rating_numerator = ratings
```

To improve it even further, you may also want to try adjusting the code so rating denumerators would also capture decimal values.

I find tools such as this one to be helpful in finding the correct regex.

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.

The master dataset has been properly analyzed and several insights have been produced. Visualizations are included in the report.

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.

The act report document is well-written and it is such a pleasure to read, well done.

(Optional) The report is aimed at general audiences, so we need to make it as readable as possible. We suggest removing all code blocks and any other artifacts that may have made the report harder to read.

(Optional) This is optional, but we suggest including pictures for aesthetic and additional context purposes on top of the required visualizations. Example: include a screenshot of a specific tweet, a specific breed of dog, etc. Anything to get the reader engaged. Picture this report like a blog post or magazine article; we want people to be engaged and have fun while reading.

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

Rate this review

START