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 have done an excellent job wrangling the given data and producing some interesting insights like positive correlation between retweet count and favorite count of tweets by WeRateDogs twitter account.

You also did a fantastic job of incorporating the previous reviewer suggestions. As a different reviewer, I have left some additional comments. I made these comments marked as Suggestions to help you improve the project. It does not require you to resubmit the project. You have already passed the project. Congratulations! 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!

Here are a few resources that may help your continued learning:

- A critical skill for data scientist/analyst is data visualization. With python seaborn libray, you can plot many different kind of charts apart from the ones you plotted. You can take a look at it.
- Having developed a wide array of data wrangling skills, the next thing you may want to learn more about is predictive analytics. Luckily, there is a free machine learning course available with this nanodegree. You can see a lesson Intro to Machine Learning in EXTRACURRICULA section. You can take a look at it.
- PEP8 is the style guide for python. This style guide provides guidelines and best practices on how to write Python code to improve the readability of code and make it consistent across the wide spectrum of Python code. You can take a look at this guide here; https://www.python.org/dev/peps/pep-0008/ and should strive to adhere to these guidelines.

Code Functionality and Readability

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

Good job adding a hyper-linked Table of Contents so that it is very easy to navigate through your notebook. This shows your attention to details.

Excellent job writing functional code, executing the code and displaying the output without any 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.

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. A good notebook structure also makes code maintenance easier.

Gathering Data

Data is successfully gathered:

- From at least the three (3) different sources on the Step 1: Gathering Data page.
- In at least the three (3) different file formats on the Step 1: Gathering Data page.

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

Excellent job successfully gathering data from local file `twitter_archive_enhanced.csv` and from a URL (`image_predictions.tsv`) and imported them into separate pandas dataframes.

Suggestions

You seem to have used `tweet_json.txt` provided in the supporting material in the project instruction. It is fine as far as completing the project for this nanodegree is concerned. However, I strongly encourage you to query twitter API and gather data by yourself when possible as it is an invaluable skill.

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.

Suggestions

Assessment is the most important part for data cleaning, which has huge impact on your downstream data analysis or predictive modelling. The following are some of the pandas functions you can use in the data assessment. You have already used some of these functions. You can try others.

• testing.assert_series_equal
• Various methods of indexing and selecting data - .loc(), .iloc()
• .duplicated()
• .isnull()

- .nunique()
- .info()
- .describe()
- .value_counts()
- .head()
- .tail()
- .sample()

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.

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.

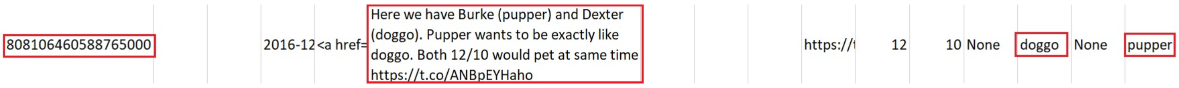
All issues identified in the assess phase are successfully cleaned (if possible) using Python and pandas, and include the cleaning tasks required to satisfy the Project Motivation.

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

Good job copying the data prior to cleaning. If you want to know more about why it is important to copy the dataframes please see this stack overflow thread. Copying is also important if at some point you need to trace back on your steps.

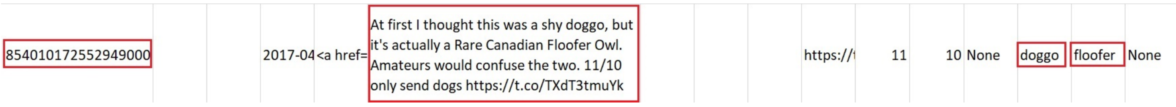
Suggestions

Did you notice that certain tweets have more than one stage? 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.

You have used the string `None` to represent missing dog stage data. Instead, you should use `np.nap` to represent missing dog stage data, which is a standard way to represent Null values in pandas. This makes it easy to handle null values programmatically as there are many function available in pandas (e.g., isnull()).

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. This fact can be used to handle this issue programmatically, but you handled this issue individually which is not robust and missed some non-dog names. Moreover, instead of removing them you should set them to `np.nan` so that you do not loose entire row.

Here is a simple way to clean the non-dog names in the `name` column;

```
mask = DataFrameName.name.str.islower()
column_name = 'name'
DataFrameName.loc[mask, column_name] = np.nan
```

There is no reason to drop rows with decimal numerators. Instead you need to correctly extract them. Here is a programmatic way to handle this issues is;

```
rating = df_1_clean.text.str.extract('((?:\d+\.)?\d+)\/(\d+)', expand=True)
rating.columns = ['rating_numerator', 'rating_denominator']
df_1_clean['rating_numerator'] = rating['rating_numerator'].astype(float)
df_1_clean['rating_denominator'] = rating['rating_denominator'].astype(float)
```

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.

You have done a good job using index argument in `to_csv()` function and setting it to `False` to avoid adding a unwanted index column in the saved file.

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.

Suggestions

You used heatmap in this project, which is fine as far as completing this project as project requires you to create just one visualization. But it is a good idea to know about different kind of charts one can use to represent different kinds of insights. For instance, for time series data (month, year etc.) line chart works best (e.g., favorite count over time). To depict relationship between two quantitative variables scatterplot works best (e.g., relationship between retweet count and favorite count). To depict distribution of variables, histogram is suitable (distribution of rating). Here are some resources that help you choose right kind of visualizations to represent various types of data/insights.

Choosing the right chart type for your insight.
How to pick the right chart type.
When to use line chart vs area chart.
The difference between a bar chart and a histogram.
Why pie charts are not an ideal choice in most cases.
Quickly plot correlation between multiple variables is pandas using scatter matrix.

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.

You have done an excellent job producing this very interesting report explaining the insights you gained from your analysis.

Suggestions

Since this report should be like a blog post or magazine article, you can also include a screenshot of a specific tweet, a specific breed of dog, etc. (anything to get the reader engaged).

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.