wrangle_report

June 26, 2022

1 DATA WRANGLING REPORT

1.0.1 written by: Toluwalase Tawak

1.1 ## Project Goal

The goal of this project is to practice and familirise myself with data wrangling skills which include; Gathering, Assessment, Transformation and Cleaning. These activities will be carried out on the twitter archive of the [@RateDogs](https://twitter.com/dog_rates) account. This account rates people's dogs with a humorous comment about the dogs.
The report summarises how approached the data wrangling for this project.

1.2 ## Project Details

1.2.1 Gathering Data

The data used for this project consists of three different dataset. How these datasets were gathered are as follows:

`Twitter Archive`: This data was provided by Udacity and was downloaded onto my local machine and uploaded into my Jupyter workspace in a virtual machine provided by Udacity. After importing the pandas library, the Twitter Archive data was read into a pandas dataframe using the **pandas.read_csv()** function.

`Tweet Image Prediction`: A link was provided with the data here. I imported the Python requests library and use the the **get()** function from library to download the page into a variable.

Using the **with open()** function, I wrote the webiste content to a tsv file in the same working directory. I proceeded to read the downloaded tsv file into a pandas dataframe.

`Tweet Json`: To carry out gathering of this data, a twitter developer account was required. While the approval for my developer account was taking time, I proceeded to use the data already gathered and stored by Udacity for students who might face difficulties with having their developer accounts approved. The data was downloaded unto my local machine from here as a txt file, and then uploaded to my Jupyter workspace in a virtual machine provided by Udacity.

Using the **with open()** function again, I loaded each line of JSON format and appended the lines into list named **status**. This list was then fed as an argument to the **pandas.DataFrame()** function and converted into a pandas dataframe.

1.2.2 Assessing Data

After gathering the datasets and creating DataFrame objects with them, I went on to assess them;

1. I first of assessed each dataframe **visually** by printing them out and visually observing the content and structure of the three different dataframes individually. This was done in jupiter notebook by scrolled through each dataframe in every direction.

2. I then went out to carry out **programmatic** assessments of the three dataframes using various pandas methods and functions such as **shape, info(), describe(), notnull(), head(), nunique(), sample(), duplicated(), value_counts(), query()** and **columns**.

1.2.3 Cleaning Data

This section of the wrangling process was broken down into three parts: >1. `Define`: Where the cleaning process to be carried out was explained >2. `Code`: Code needed to achieve the cleaning goal defined was written and run. >3. `Test`: Code was written and run to confirm that the cleaning goal was achieved.

To begin, copies of the three datasets were created. These copies were used to carry out the cleaning activities.
The cleaning proccesses carried out on each dataset are as follows:

1. Non-null rows in the in_reply_to_status_id and retweeted_status_id columns were removed. This was to provide us with a dataset containing only original tweets and no retweets or replies.

2. Five columns, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, in_reply_to_status_id and in_reply_to_user_id columns were dropped because they now contained only null values.

3. The id column in Image Prediction was renamed to tweet_id. Values in the tweet_id columns for all three tables were converted from integer to string. The timestamp column in the archive dataframe was converted to datetime format.

4. The four columns containing stages of dogs being rated were concanated into one column (description) and cleaned to carry only one description for the dogs.

5. All rows whose the name attribute began with lower case were renamed and given the value None, the previous names were confirmed to not be actual names. Two names were renamed to match the complete names of the dogs.

6. Special characters in the text column were replaced with the appropriate strings and links in text column were replaced with empty strings.

7. Values in the p1, p2, and p3 columns cleaned by replacing underscores with a space and making the first letter of each word upper case.

8. p1, p2 and p3 was melted into one column. p1_conf, p2_conf, p3_conf was melted into one column. A function was created to keep the corresponding values for the first **True** value in either p1_dog, p2_dog or p3_dog.

9. The three dataframes were merged on their tweet_id column to become one dataframe.

10. Suspicious values in both rating columns were addressed by dropping rows were the tweets was rating more than one dog, correcting and rounding decimal ratings, rows with the wrong ratings and dropping rows with no rating in their text.

1.2.4 Storing the Data

After gathering, assessing and cleaning, the merged dataset was converted and saved as a csv file called twitter_archive_master.csv.

```
In [7]: # from subprocess import call
        # call(['python', '-m', 'nbconvert', 'wrangle_report.ipynb'])
        !jupyter nbconvert --to pdf wrangle_report.ipynb
```

```
[NbConvertApp] Converting notebook wrangle_report.ipynb to pdf
[NbConvertApp] Writing 22446 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 30453 bytes to wrangle_report.pdf
```

```
In [ ]:
```