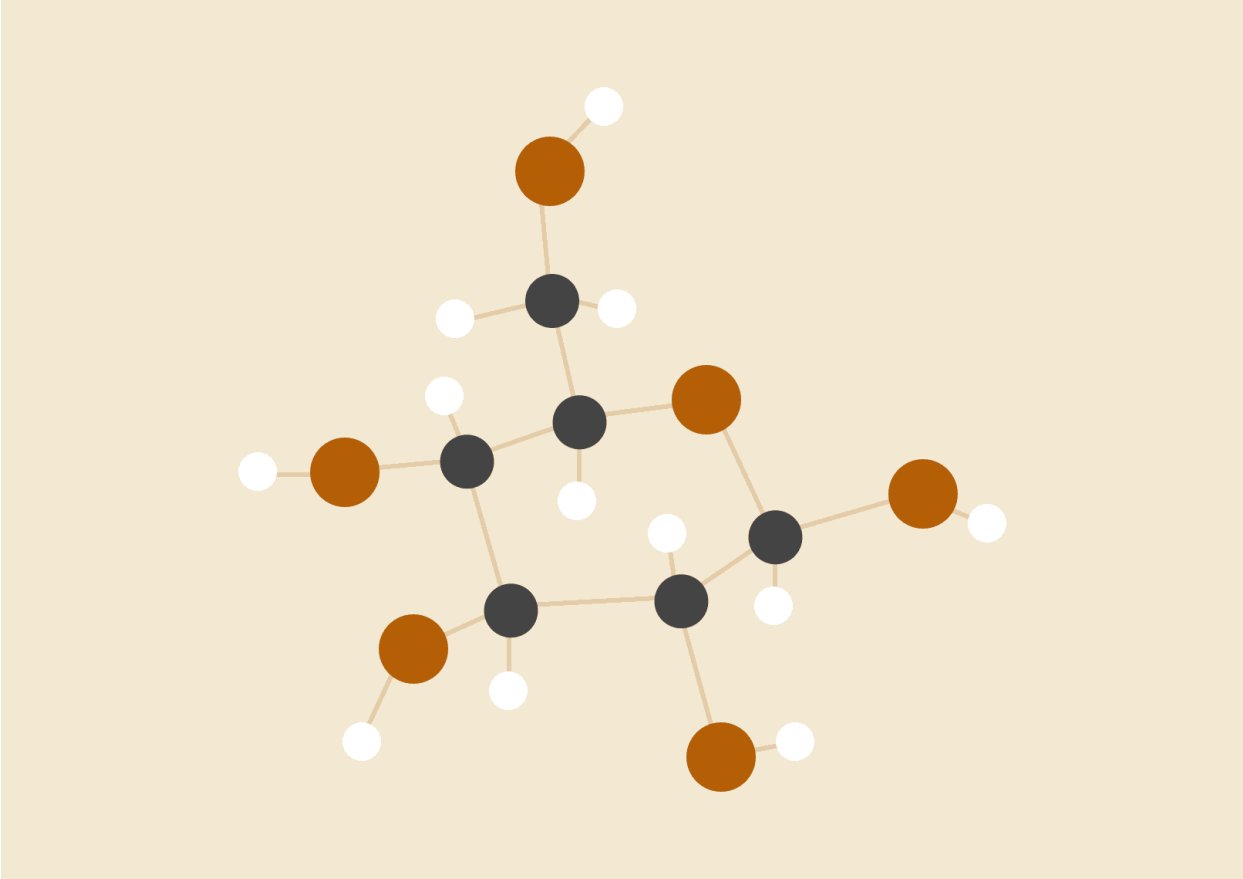
WRANGLING AND ANALYSING DATA

Report 1: Wrangling Efforts



Alham O. Hotaki

@Masterschool

Dec 14, 2022

INTRODUCTION

Real-world data rarely comes clean. Using Python and its libraries and gathering data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling. The wrangling efforts are documented in a Jupyter Notebook, plus showcase them through analyses and visualisations using Python (and its libraries) and/or SQL.

The dataset used for wrangling (and analysing and visualising) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with humorous comments about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for this report to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. More on this soon.

Project Steps

Steps in this project are as follows:

Step 1: Gathering data

Step 2: Assessing data

Step 3: Cleaning data

Step 4: Storing data

Step 5: Analysing, and visualising data

Step 6: Reporting

- data wrangling efforts (current report)
- data analyses and visualisations

Step 1: Gathering data

Three data frames were gathered for this project.

1. The WeRateDogs Twitter archive: Twitter archive contains basic tweet data for all 5000+ of their tweets, but not everything. This file is downloaded and called df_1 and loaded from .csv file format.
2. The tweet image predictions: This contains the predictions of dog breeds after running every image in WeRateDogs Twitter archive through a neural network. This file is downloaded and called df_image and loaded from .tsv file format.
3. Additional data from the Twitter API: This file contains retweet count and favourite count for each tweet obtained via the Twitter API. Despite several attempts, this file was not obtained from API due to lack of having an elevated account in Twitter, therefore was downloaded from Udacity as a .json file and loaded as df_2.

Step 2: Assessing Data

Two types of assessments have been conducted:

1. 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).
2. Programmatic assessment: pandas' functions and/or methods are used to assess the data.

After assessment, following two types of issues were identified:

Quality issues:

1. Incorrect Names: the incorrect names shall be replaced either extracting them from the 'text' column, if not available, shall be replaced with NaN.
2. Timestamp: extra +0000 values in the timestamp column to be removed.
3. Drop rows: where they contain 'dont send' in the 'text' column as they are irrelevant photos for rating.
4. Drop rows: In order to have original rating, replies and retweets should be dropped.
5. Timestamp column type to be reassigned properly.

6. Nulls are defined as 'None' in several columns, and shall be replaced with 'NaN'.
7. ID fields dtype to be reassigned to objects.
8. Incorrect ratings extracted from the 'text' column shall be corrected.
9. Removing columns to be more appropriate.
10. Renaming columns to be more appropriate.
11. Numerator and denominator dtypes to be reassigned to float

Tidiness issues:

1. Combining three data frames into one master data frame with appropriate data.
2. Data in four dog columns: 'doggo', 'floofer', 'pupper' and 'puppo' shall be into one category column.

Step 3: Cleaning data

It is made sure that following issues are considered during this stage:.

- Before performing the cleaning, a copy of the original data is made.
- During cleaning, the define-code-test framework is used and clearly documented.
- Cleaning includes merging individual pieces of data according to the rules of tidy data. The result is displayed in a high-quality and tidy master pandas DataFrame (or DataFrames, if appropriate).

Issue 1:

Combining three scattered data frames into one master data frame with appropriate data

Solution:

The data frames have been merged together into one on tweet_id as master key. ‘Tweet_id’ column was ‘id’ in df_2, first that was renamed then, the main merge was done.

Issue 2:

Incorrect Names: the incorrect names shall be replaced either extracting them from the 'text' column, if not available, shall be replaced with NaN.

Solution:

A function applied to extract names from the 'text' column contains 'named_____' or 'name is ____' and replaces it with the incorrect names in the dataframe. If such is not

available, then replace it with NaN. A list of messy names also defined as Messy_Names = ['None', 'a', 'an', 'the', 'O', 'my', 'by', 'one', 'his'].

Issue 3:

Timestamp: extra +0000 values in the timestamp column to be removed.

Solution:

It was done by defining a function and its application on the timestamp column.

Issue 4:

Drop rows: where they contain 'dont send' in the 'text' column as they are irrelevant photos for rating.

Solution:

A list of messy words were defined, if one of those were existing in the ‘text’ column, then the corresponding row was deleted. ["dont send", "don't send", "do not send", "Dont send", "Don't send", "Do not send"]

Issue 5:

Drop rows: In order to have original rating, replies and retweets should be droped

Solution:

Drop rows with 'in_reply_to_status_id' is not NaN then 'retweeted_status_id' is not NaN

Issue 6:

Timestamp column Data type

Solution:

‘Timestamp’ format was converted to datetime.

Issue 7:

Nulls are defined as 'None' in several columns, and shall be replaced with 'NaN'.

Solution:

Replaced 'None' with np.nan in 'doggo', 'floofer', 'pupper','puppo' columns. In the ‘name’ column, it was already removed. (see Issue 2)

Issue 8:

ID fields dtype to be reassigned to objects.

Solution:

The ID fields, like tweet_id, in_reply_to_status_id etc. should be objects, not integers or floats because they are not numeric and aren't intended to perform calculations.

Issue 9:

Incorrect rating values: There are numerator and denominator values extracted improperly from the 'text' column. They will be assessed, if proper values are found in the 'text' column, if not, they will be removed.

Solution:

- tweet_id 740373189193256964 should be 14/10 instead of 9/11
- tweet_id 722974582966214656 should be 13/10 instead of 4/20
- tweet_id 716439118184652801 should be 11/10 instead of 50/50
- tweet_id 682962037429899265 should be 10/10 instead of 7/11
- tweet_id 666287406224695296 should be 9/10 instead of 1/2
- tweet_id 883482846933004288 should be 14/10 instead of 5/10
- tweet_id 786709082849828864 should be 10/10 instead of 75/10
- tweet_id 778027034220126208 should be 11/10 instead of 27/10
- tweet_id 680494726643068929 should be 11/10 instead of 26/10
- tweet_id 749981277374128128 will be removed (1776/10) improper rating
- tweet_id 670842764863651840 will be removed (420/10) improper rating
- tweet_id 810984652412424192 will be removed (24/7) improper rating

Issue 10:

Data in four dog columns: 'doggo', 'floofer', 'pupper' and 'puppo' shall be placed in one category column.

Solution:

A function will be applied that assigns the correct dog stage based on each tweet, 'multiple' will be a new category if we have multiple dog stages for tweet and NaN if there is no stage defined.

Issue 11:

By testing the outcomes of issue 9, it was found that some categories are misplaced or some are not defined properly.

Solution:

The following problems were identified and resolved:

- Some tweet texts contain 'floof', 'Floof', 'floofs', or 'Floofs' while they are not categorised as 'floofer' due to mismatch in spelling.
- Some tweet texts contain 'puppers', 'Puppers' while they are not categorised as 'pupper' due to mismatch in spelling.
- tweet_id 817777686764523521 was wrongly classified as doggo while it is 'pupper'
- tweet_id 855851453814013952 was wrongly classified as 'doggo' while it is 'puppo'
- tweet_id 854010172552949760 was wrongly classified as 'doggo' while it is 'floofer'

Issue 12:

Removing columns:

Solution:

After introducing a new column as 'dog_stage', there is no need to keep other four columns: 'doggo', 'floofer', 'pupper' and 'puppo' columns.

Besides that columns with the highest number of empty cells such as:

- in_reply_to_status_id
- in_reply_to_user_id
- retweeted_status_id
- retweeted_status_user_id
- retweeted_status_timestamp

will be removed.

Issue 13:

Renaming Columns

Solution:

In order to be more appropriate for the user, following columns were renamed:

- - 'timestamp' : 'tweet_timestamp'
- - 'text' : 'tweet_text'
- - 'name' : 'dog_name'
- - 'p1' : 'dog_breed_1st_prediction'
- - 'p1_conf' : 'prediction_confidence_1stguess'
- - 'p2' : 'dog_breed_2nd_prediction'
- - 'p2_conf' : 'prediction_confidence_2ndguess'
- - 'p3' : 'dog_breed_3rd_prediction'
- - 'p3_conf' : 'prediction_confidence_3rdguess'

Issue 14:

Numerator and denominator dtypes to be reassigned to float

Solution:

Numerator and denominator columns are subject for further analysis and are preferred to be reassigned to float.

Step 4: Storing data

The cleaned data frame was stored as a df_twitter_archive_master.csv file.

End of Report_1