Udacity Project

Twitter Data Wrangling Report

Author: Mr. Burak Gunbatan

Data Gathering

The first dataset that was meant to be downloaded manually was relatively easy to import. This was done by clicking on the hyperlink provided and moving the data into the relevant directory, importing using Python's Pandas package (pandas as pd) using the pd.read_csv() function.

The next dataset was meant to be downloaded programatically using the Requests package. The package takes a URL and saves the response to a variable which can then be saved or written to a relevant file before being imported again withPandas.

I created my all files in my computer because pandas version was not up to date. Also, I came across with a problem of twitter API, regarding this situation I used another step which was described in project folder of Udacity.

After importing all the data, I made sure to create copies of the data frames using pandas.DataFrame.copy() so that nothing strange would happen to the original dataframes that I imported.

Data Assessment

Next, the imported data needed to be assessed for quality issues. This was done programmatically within my ipynb file.

Visual assessment was not very well documented not only because it is difficult to point directly to what you are viewing without messing around with image files. This meant that I ran into more problems during the cleaning stage later in the wrangling process.

Visual Assessment

I could recreate what I wrote in my jupiter notebook file, but this would be a waste of time so I'll just keep this brief.

Basic problems I found during the visual assessment were that some IDs that were handed which might cause problems for me down the line.

One thing that I missed as well was that the columns containing the dog stage not only was structured as a sort of "sparse matrix" when it could just be one column (tidiness issue), but it also was the case that roughly 90/200 of the values were present, which wasn't immediately noticeable because the missing values were denoted with a "None" string instead of a NaN value. There were other columns as well which were not important so I just removed them.

Programmatic Assessment

The programatic assessment is where I found most of the issues that I wanted to address during the wrangling phase. It is worth noting however that there is not a hard line btween what constitutes programmatic and visual, as i noted within the notebook itself. The general problems that I found was missing values using pandas.DataFrame.info(), which I used quite a lot throughout the wrangling process. The tweet_id was always an integer when it should have been a string (see Ordinal vs Categorical values).
I used pandas's filtering capabilities to help me figure out where to draw the line between which pictures were and werent dogs. This means that I had to look at the top three predictions that were either predicted to be a dog or not and then look at the image visually in order to assert if this categorisation was true or not. I found that images were unlikely to be dogs if the top three predictions were false or if there was a True prediction with a probability of 0.2.

Data Cleaning

During the cleaning I solved the problems. Some of the issues I had were due to not thoroughly assessing the data. This meant that I had to skip a lot throughout the document to make sure all data entries were properly updated, which was cumbersome.

There were also a lot of times when commands that I was using that working for one reason or another. One time I tried to remove duplicate rows using pandas.DataFrame.drop_duplicates() but this did not work because the rows that sometimes contained the same names of dogs sometimes had different tweet_ids.

However, having defined the small "step-by-step" tasks helped me a lot to focus on each individual one and execute them, which meant that this step took much less time than it might have if I did not have a plan, so I am very happy about that.

Data Output

After the data cleaning was completed, I created a CSV file and a SQLite database using pandas and sql respectively.
The SQLite database was made by creating the pandas.DataFrame.to_sql() command using the engine I had just created to export the data.
The CSV creation was much less interesting, that was just done using pandas.DataFrame.to_csv().