Data Wrangling Project for WeRateDog

By Ashwin Paramashivan

Introduction

In this report I will be explaining my step of my data wrangling and how I was able to assess the data visually and programmatically. The data I have wrangled and cleaned visually is the archive of the WeRateDogs.

Gather

From this project there were three data that I had to gather. The three files that I have gathered are

1. “twitter_archive_enhanced.csv”,
2. “image_prediction.tsv”
3. “tweet_JSON.txt”.

The first file I downloaded manually from the website and uploaded them manually to my project workspace. Then using pandas, I created a dataframe for that file which is “df_1 = pd.read_csv(filename).
The second file I downloaded them programmatically into my workspace. I scraped the url from the website. Then used response=requests.get(url) to gather the data from that webpage. Then I used file.write() and with open(filename, wb) to create the tsv file and write all the data to that tsv file.
The third data I used an API to gather the twitter data using the first data frame created and writing them into the “tweet_JSON.txt” file. Afterwards, I used python list dictionaries to read the data from the third file I created and create a new data frame. I used pd.DataFrame(twitter_API, column name) to create the third data frame.

Assess

While I was assessing the data programmatically and visually, there were 13 quality issues and 3 tidiness issues I found. The ones I visually assessed through the excel file, I found that there were odd names given to some dogs such as ‘a, an, the, his, very etc’. Another one was inaccurate ratings. In the twitter page there was a dog rating which showed 9.75/10, but in the csv file it showed 75/10. I found three tidiness issues.

1. Dog stage is in 4 different columns instead of 1
2. TweetId is in 3 different tables and they can be merged into one dataframes
3. Rating Numerator and Rating Denominator can be merged into one column.

I found rest of the quality issues programmatically using the following commands:
-df.info()
-df.head()

-df[col].value_counts()
-df.describe()

Clean

Before cleaning each steps I copied each of the dataframe into a clean version of the data frame. “df_1_clean = df_1.copy()”.
For the rating numerator which had inaccurate ratings I used df_1_clean.text.str.extract(REGEX, expand=True). Then convert the numerator and denominator to float. Majority of the denominator rating is 10 but there were some denominator ratings which are 0. For that specific column I had to change the numerator in index 313 from 960 to 13 and the denominator from 0 to 10.
Then for the tidiness of the issue I used df_1_clean[columns].astype(str).sum(axis=1) to merge all the columns of the dog stage into one. I merged all the three dataframes into one master dataframe. For this one I used commands such as df_master.dropna(), df_master.drop(cols, axis=1) to drop the unnecessary columns and the columns with the null values. Then I filtered the outlier for ratings which were above 1.5 using df_master[df_master[‘rating] < 1.5]. Once I cleaned all the dataframes, I put them into the master csv file using df_master.to_csv(filename, inplace=False).

Summary/Conclusion

According to my data wrangling skills, there were lots of skills I have used to clean the data programmatically and it was very efficient. I was able to remove all the unnecessary columns. I had to change the datatype for some columns. Now the data table is much cleaner than before.