Data Wrangling Process for Twitter Data of WeRateDogs Account

Introduction

Tweet's information of WeRateDogss account were gathered through three different sources: given file, downloading file programmatically, and using twitter API. Three data frames out of those sources. The data frames were assessed for quality and tidiness issues. After that there were cleaned, and stored in csv file. Finally, data visualization was applied over the final product, in addition to analyzing this data and come out with some insights and observations about it.

In this report only the data wrangling process is explained. The process includes three main stages: gathering, assessing, and finally cleaning. Each stage will be explored thoroughly in the report.

Brief description of the WeRateDogs account is needed for the reader to understand the information what is being analyzed. This account basically rates dogs out of 10. The rating could be above or below 10. So, the tweets and the rating of the dogs is being gathered, wrangled, and analyzed to find, for example, what are the dogs that have the most likes, retweets, or the highest ratings.

Data gathering stage

The data were gathered from three different sources. The first source was a given csv file named "twitter-archive-enhanced.csv". This file was given by Udacity. Python coding was used to transfer it to data frame. The piece of code used for transferring is: *pd.read_csv('file name').* The first data frame was named *df1*. The information contained in was the tweet's ID, replaying information, time stamp, the source of the tweet, retweets status information, URL of the tweet, rating mentioned in the tweet for the dog, the text of the tweet, the name of the dog, and dog description.

The second file was downloaded programmatically. The file was downloaded using *Requests* library from python. The contains image of the dogs mentioned in each tweet with breed recognition results (using deep learning). The information in this file was converted to data frame named *df2.*

The last file was got through twitter API, tweepy. This API allowed for downloading the json file which contains additional tweets information from WeRateDogs page. The information, which are the focus of the interest, are the number of retweets, and likes for each tweet. That information makes the last data frame which is called *df3*.

Data Assessing stage

The three data frames were assessed for quality and tidiness issues. Quality issues can be missing data, invalid ones, low accuracy, or not consistent format. Tidiness is more of structural issues not having the same data under the same variable, or row don't represent values and so on. The following data issues were documented during the assessing phase:

- Date frame 1:
    1) Quality:
        - ☒ The values under timestamp column are object type instead of being time type.
        - ☒ Having None value under the following columns (doggo, floofer, pupper, puppo) instead of NaN
        - ☒ Large numbers of missing values for the following columns (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, retweeted_status_timestamp)
        - ☒ There is a rating of zero for numerator are less than the denominator. It can be sign of quality issue.
        - ☒ There is a rating of zero or less than 10 for the denominator which needs to be checked.
        - ☒ The type of data under the columns (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, retweeted_status_timestamp) are float which is not correct
    2) Tidiness:
        - ☒ Dog's stage is divided over 4 columns instead of having it under one column. This is tidiness issue.
- Data frame 2:
    1) Quality:
        - ☒ Minor quality issue: the name of the breed doesn't start with capital for all dogs
        - ☒ Some values mentioned under columns (p1, p2, and p3) are not valid. e.g., for row# 1270 it predicts the dog as bow_tie, and sunglasses which doesn't make any sense
    2) Tidiness:
        - ☒ The main issue here is tidiness. The last 9 columns have the same information and can be merged into one.
        - ☒ The columns names are not descriptive which is another tidiness issue.
- Data frame 3:
    1) Tidiness:
        - ☒ The only remark that can be highlighted here is that this data frame can integrated with the other two.

It worth mentioning that the assessment process used the following codes to detect issues programmatically:

- Using .Info method to get general information
- Using .isnull().sum() methods to check for missing values
- Using .sample to check random samples and apply visual assessment
- Using .describe to get statistical information about the data

Tidiness issues were discovered visually, with programmatical approach used.

Data cleaning stage

The data cleaning section will be divided to three subsections. Each will explain the work done to clean each data frame. First will go over the issues mentioned in assessing stage. After that I will go through the cleaning steps and them explained, and connect them with the issues found in assessment stage.

Cleaning the first data frame

First let us show the main issues extracted during the assessing stage for the first data frame:

1) Quality:
    - ☒ The values under timestamp column are object type instead of being time type.
    - ☒ Having None value under the following columns (doggo, floofer, pupper, puppo) instead of NaN
    - ☒ Large numbers of missing values for the following columns (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, retweeted_status_timestamp)
    - ☒ There is a rating of zero for numerator are less than the denominator. It can be sign of quality issue.
    - ☒ There is a rating of zero or less than 10 for the denominator which needs to be checked.
    - ☒ The type of data under the columns (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, retweeted_status_timestamp) are float which is not correct
2) Tidiness:
    - ☒ Dog's stage is divided over 4 columns instead of having it under one column. This is tidiness issue.

The first point was cleaned changing the data type from being an object type to date type. For the second point, the none values were changed to 'NaN' this is needed to detect the number of null values, since this is the way the ***pandas*** library recognize them. Once they were recognized, it was found out that more than 90% of those data are actually missing. Hence, it was fixed in the same manner as point three, by removing the columns that have more 90% of their values missing. The reason to use this approach is that the values under those columns are not easily retrievable and it would very time consuming as they have to be filled manually most probably. By removing the columns in the third point, this by default solve the last point.

Regarding the denominator rating that is different than 10, first the data frame was filtered to show only those values that are less than 10. After going through them by checking the URLs associated with them. It was found out that the rating might not always be out of 10, hence this was excluded from being an issue.

However, during the assessing stage an issue was discovered where the not all the texts are original tweets from WeRateDogs account, instead they were retweets. Those retweets start with RT @, and they were eliminated.

After that, for the numerator issue, it was found out that about 900 out of 2000 rows have the same remark. It was first thought of it as invalid or inaccurate data, since dogs are actually cute and would not expect one to have a rating less than 10. However, after verifying some of them it seems that this can be excluded as an issue.

Finally, the tidiness issues where there were four columns (doggo, floofer, pupper, puppo), that represent the dog stage. This is clearly a tidiness issue contradicting with the fact that every row is an observation. Those were clearly observations that shall be combined under one variable. The cleaning method followed was to have them combined under one column called dog stage.

Cleaning the second data frame

First let's recap over the issues captured regarding the second data frame

- Quality:
    1) Minor quality issue: the name of the breed doesn't start with capital for all dogs
    2) Some values mentioned under columns (p1, p2, and p3) are not valid. e.g., for row# 1270 it predicts the dog as bow_tie, and sunglasses which doesn't make any sense
- Tidiness:
    1) The main issue here is tidiness. The last 9 columns have the same information and can be merged into one.
    2) The columns names are not descriptive which is another tidiness issue.

The first quality issue, as stated in the assessing stage is minor and has no real impact on the data quality, so it was not modified. However, the second quality issue was resolved over to stages. The first stage was to have all the rows with the columns (*p1_dog, p2_dog, p3_dog*) having False value under all of them. This indicates that the neural network didn't identify the dog correctly. After that those rows were eliminated. The second stage were to eliminate any False indication by replacing the identification result with space. For example, if the identification result for a dog was chair, the value chair will be replaced white space.

The first tidiness issue, was resolved by having all the identification result collected under one column called *dog_breed*. It worth mentioning that a further step could have been taken by eliminating the predictions with low confidence score. However, instead, an assumption was placed that a dog can be a mix of all the breeds indicated. Moreover, this elimination process would have been lengthy and won't serve the analysis part.

The second tidiness issue were resolved by changing the names of the columns. However, those columns were removed as they were deemed unnecessary.

Cleaning the third data frame

The last data frame didn’t have any quality issue. However, it was contradicting with the third role of tidy data, which was having each type of observational unit forms a table.

This data frame by itself doesn't tell much and doesn't make an observational unit. Hence it had to be combined with the other two data frames.