wrangle_report

February 10, 2019

0.1 PROJECT: WRANGLE AND ANALYZE DATA

BY: SAURABH KULKARNI

This document describes the data wrangling steps that were done. The whole process is divided into three main sections:

- Gather data
- Assess data
- Clean data

0.2 Gather Data

Data has been gathered from three different sources.

- Twitter Archive CSV File: provided directly
- Image Predictions TSV File: URL provided
- Tweet information using APIs: Tweepy

Getting first piece of data: Twitter Archive CSV File The first piece of data was directly given to us in a CSV file. This file was manually downloaded and saved in working directory. Pandas has been used to directly read this CSV file into a dataframe.

Getting second piece of data: Image Predictions TSV File URL to this file is provided in project details. Using "Requests", this file has been programmatically downloaded and read into a dataframe using Pandas.

Getting third piece of data: Tweet information using Tweepy Tweepy, a Python library for accessing the Twitter API has been used to retrieve tweet data like favorite and retweet count. This data is stored in dataframe using Pandas. All retrieved data has also been stored into a text file.

0.3 Assess Data

The next step is to assess all three pieces of gathered data to identify 8 quality issues and tidyness issues. Python in-built functions like info, dtypes,head, tail and sample were used to visualize and inspect data for issues.

0.3.1 The following observations were noted:

Quality Issues

- Missing values in df_imgpre. 2356 tweet ids in df_tarch and 2075 tweet ids in df_imgpre
- Data type for timestamp in df_tarch is not datetime (will be needed to filter out only tweets before August 1st 2017)
- Dog names might be incorrect in df_tarch
- Dog stage might be incorrect in df_tarch
- Rating numerator might not be correctly extracted in df_tarch
- Rating denominator might not be correctly extracted in df_tarch
- Not meaningful column headers in df_imgpre
- No consistency in dog names (lower and upper case) in df_imgpre

Tidyness issues

- Dog stages does not have a column (variables form a column violated)
- Timestamp has a mix of three values, month, date, time (each variable forms a column)
- Two tables for tweet data information, one is favorite count from API and other df_tarch

0.4 Clean Data

To preseve original data during cleaning, we first created copies of each dataframe we created in "Gather Data" step. Once that was done, we started tackling each quality and tidyness issue. First, we decided to tackle missing data issue since we do not want to repeat cleaning steps again if missing data issue is handled later.

QUALITY ISSUE 1

Missing values in df_imgpre. There are 2356 tweet ids in df_tarch and 2075 tweet ids in df_imgpre.

We have a twitter archive of 2356 tweets and image predictions of 2075 tweets. We will identify the tweet_ids without image data and flag them using a newly created column image_prediction having boolean value True or False. This will not be deleted as we might need it for analysis.

- Get list of missing tweets between df_imgpre_clean and df_tarch_clean
- Create a new column image_prediction in df_tarch_clean
- Assign True or False depending on whether that tweet id is present in the list from step 1.

QUALITY ISSUE 2 Data type for timestamp in df_tarch is not datetime (will be needed to filter out only tweets before August 1st 2017)

- Using pandas, change dtype of timestamp column in df_tarch_clean to datetime.
- Check if any timestamp exists beyond August 1, 2017

Quality Issue 3 Dog names might be incorrect in df_tarch_clean

- Using regex extract name of dog
  - Dog name is followed by "This is" or "Meet"
- Create a new column regex_name and save these to this column

- Compare this new column to "name" column to see if there are any discrepancies
- If there are discrepancies, verify the text and drop "name" column
- Rename "regex_name" to "name"
- To check this we will see if the number of columns with names has increased or not

Quality Issue 4 and Tidyness Issue 1

- Quality Issue: Dog stage might be incorrect in df_tarch
- Tidyness Issue: Dog stage does not have a column, variables form a column violated

To tackle the possibility of dog stages being incorrect, a function was defined that uses regex to match dog stages in tweet texts for each tweet id. The dog stages obtained were then passed to a new column. The old column (which came with the original data) was then deteled.

Dog stage did not have its own column. However, there were four columns for each dog stage and that is a tidyness issue. To resolve this, all four columns were deleted and a new column "dog_stage" was created which has dog stage information for every tweet.

Quality Issue 5 Rating numerator might not be correctly extracted in df_tarch

- Extract numerator value from tweet text using regex
- Create new column with these numerator values
- Delete (Drop) old column rating_numerator

Explanation of the function created to get numerator rating Objective: To extract numerator from ratings using 'text' column from df_tarch_clean and create new column for this data

STEPS:

1) Checks if there are two ratings (with denominator 10) in tweet text using regex
2) If only one rating (out of 10) found, it extracts numerator and returns it
3) Else, it moves to the next loop which finds the number of ratings in the text that are NOT out of 10
4) If only ONE rating is present, it will extract numerator and return it. The reason we are extracting when only ONE rating is present is because tweets might contain strings like 24/7 or 50/50 which are not ratings.
5) If all these conditions are not meet, "None" is returned.

Note: There are 26 rows that either have multiple ratings (for two dogs or same dog) or no ratings. These have been filled with "None" values.

Quality Issue 6 Rating denominator might not be correctly extracted in df_tarch
This will be similar to what was done when extracting numerator from 'text' column.
Explanation of the function to get denominator rating below:

1) Checks if there are two ratings (with denominator 10) in tweet text using regex
2) If only one rating (out of 10) found, '10' is denominator and that is returned.

3) Else, it moves to the next loop which finds the number of ratings in the text that are NOT out of 10

4) If only ONE rating is present, it will extract denominator and return it. The reason we are extracting when only ONE rating is present is because tweets might contain strings like 24/7 or 50/50 which are not ratings.

5) If all these conditions are not meet, "None" is returned.

Note: There are 26 rows that either have multiple ratings (for two dogs or same dog) or no ratings. These have been filled with "None" values.

Quality Issue 7 Not meaningful column headers in df_imgpre
Using df.rename, change column names to something more meaningful

Quality Issue 8 No consistency in dog names (lower and upper case) in df_imgpre

- Change predict_no_1 to lower case
- Change predict_no_2 to lower case
- Change predict_no_3 to lower case

Tidyness Issue 2 Timestamp has a mix of three values, month, date, time (each variable forms a column)
Split timestamp column into four columns - Year - Month - Date - Time

Tidyness Issue 3 Two tables for tweet data information, one is favorite count from API and other df_tarch
Merge df_tarch_clean and df_tweet_data on tweet_id

0.4.1 Saving master tweets data file in csv format

Finally, all the master cleaned data was saved in a csv format.