Data Wrangling - Twitter-archive

Gather

For gathering the data, we have loaded the data from three different sources which are:

1. CSV File

This is initial file which was provided along with the project to start with. The file was downloaded from the provided link directly from outside program. The file was ten loaded into notebook, by using pandas - Read_CSV function into to the pandas Dataframe.

2. Online TSV File

For this, a link was provided in the project. The file was downloaded programmatically using python request module. When download, the file was again saved into .tsv format for a backup, and the data was loaded then jupyter using pandas into Dataframe.

3. Twitter API

This source was used to gather additional data like favourite-counts and retweet-counts, and also sometime for real time investigation for tweet ID in case of an ambiguity.

Assess

Assessment can be summed up to 4 major categories:

1. Inspecting for different features, their datatype and null value counts

The data was initially investigated for number of features available, with number of null values in each feature and total no. of observations inside each feature. It was found that few columns has large amount of NULL values, these correspond to retweets and in-reply-to tweets, which are not actual tweet of WeRateDog twitter handle

2. Checking for accuracy in mapping of Dog Data (Name, stage and Rating)

Next was to asses correct mapping of dog data in terms of name, stage and rating. First i asses for the dog stage lables i.e. doggo, puppo etc. Then programatic assesment was done to check the accuracy in maping names and corresponding ratings score. It was found that there were lots of errors in correct mapping of dog data which was needed to be corrected.

3. Tweets which are not from WeRateDogs twitter handle (Retweets and Reply Tweets)

By visual inspection, it was found that there were many tweets which were from other sources but not WeRateDog twitter handle, hence it is required to programatically access number of such cases.

4. Assessing tweet JSON data

The complete JSON data for a particular tweet was assessed, to understand what all keys are there that can be of any importance for this project, which can be included at the later stage.

By performing assessments, some of them stated above, following quality and tidiness issue were found:

1. Quality

A. twitter-archive

1. *The Retweets all correspond to the duplicity of original tweets of WeRateDogs. Hence must be removed.*
2. *Tweets corresponding to valid 'in_reply_to_status_id' data are not original review of Dog by WeRateDog, nor it has link for it.*
3. *Erroneous data types - Datetime, Dog categories - Doggo, puppo etc are string object.*
4. *Wrongs IDs - Some of the Tweet IDs are wrongly mentioned.*
5. *The source column, has a html element of HTTP link. It must have only the text value of html element.*
6. *The source column must be of category data type*
7. *Some of the values of the 'Rating numerator' and 'Rating denominator' has junk values*
8. *The dog stage - doggo, floofer, puppo, pupper is also not correctly mapped.*
9. *The dog name has sometimes invalid values.*
10. *There were lots of NaN value in expanded_url columns.*
11. *There is lot of data that is tweets not from the twitter handle of 'WeRateDogs' but others or from WeRateDog VINE.CO website, which is not a tweet. They all has to be removed as it is required to Wrangle only We Rate Dogs original twitter tweets*
12. *tweet_id must be sorted in one direction.*

B. prediction table

1. *The tweet_ids must be sorted in one direction : ascending or descending*
2. *The prediction table must consist of only tweet IDs corresponding to those in twitter_archive_master table.*

2. Tidyness

1. *The column for retweet IDs and in-reply-to-IDs is not desired and required, as it creates duplicate, and hence be removed*
2. *The twitter data for 'retweets_count' and 'likes_count' has to merged with the table.*
3. *The dog type columns i.e doggo, floofer, puppo, pupper must be in single column named as Dog type*
4. *The prediction column must be part of twitter_archive master table.*

Clean

1.1.1 Removing Retweets

Retweets actually creating duplicity in the dataset, as they are retweeting their own tweets. Sometimes they are just retweeting someone else's tweets, which is actually not a Dog's Rating. Hence we simply delete all the rows which are the retweets by WeRateDog twitter handle.

1.1.2 Removing in_reply_to tweets

Reply tweets also do not are the actual dog ratings tweets, hence they can be removed from the table.

2.1 Remove columns correponding to reply_to and re_tweets IDs and status_ids

1.1.10 Lot of NaN in 'expanded_urls' column in twitter_archive table

Since the retweet data and in-reply-to data has been removed, the need for the corresponding column is not required. Hence these column can be easily dropped. Such tweets data has all 'NaN' in the expanded_url section. It is assumed that every tweet data must has its url link for reference. Hence the tweets IDs having 'NaN' in expanded must be removed.

1.1.3 Changing datatype of timestamp and dog-stages

The datetime data column is string-type. But for analysis in later stage programmatically, datatime data must be of type 'Datetime' category. Dog-stage type in the table are set as string-type. But for further analysis and data exploration, it would be better if it is set as category-type. It can easily done by using 'astype' function in pandas dataframe.

1.1.4 Correcting the incorrect tweet IDs

2.2(Tidiness) Extracting 'favorite_count' and 'retweet_count' from the downloaded json data and adding them as column to twitter_archive table.

It has been found that some of the tweet IDs are incorrect or are not callable using twitter API. So i will start downloading the json data by calling twitter API one by one. If any failure occur, i will extract the tweet ID from the 'expanded_url' and again make a twitter API call. The downloaded JSON data will stored in the form of list and is then saved to a text file - 'tweet_json.txt'

1.1.5 The source column, has a html element of HTTP link. It must have only the text value of html element.

1.1.6 The source column must be of type category

The source column has value represented in XML element. The only important content is the 'text' value of that element. This can be done by first converting the string data into html -datatype. And subsequently extracting text value of that element. Then converting the datatype of column to 'category' type.

1.1.7 Correcting Rating: Numerator and Denominator

We can extract the correct marks from the tweet status itself and assigned to numerator and denominator column. It has been found that the numerator can be floating point number also. Hence *we can create 'Regex' value for different range of values for the numerator and subsequently use it to extract the score from the numerator.*

1.1.8 Error in correctly mapping of dog category

It has been found that there are multiple instances, that a dog-stage category is present in the tweet-text, slightly in different form, but the corresponding marked as none. This can be done by reading each tweet's ID text, and searching for the desired category name in most general form.

1.1.9 Many dogs names are not mapped accurately. Needed to be correct.

For this we have to go through the tweet text data throughout the table, and needed to evaluate if there is any consistency in part of text that has dog name present in there. Since there are so many incorrect names extracted, it is required to be done programmatically. I am doing this, by re-extracting dog names for all the tweet IDs, wherever possible, though there still must be some errors. Based on my analysis of tweet text, i have shortlisted 4 types of string-text, that has name in it. Such string start as :

1. *"named ...."*
2. *"name is ..."*
3. *"This is ..." (Excluding "This is a ...")*
4. *"say hello to ..."*

2.3 The dog type columns i.e doggo, floofer, puppo, pupper must be merged into single column named as Dog Stage

All of the labels of Dog Stage i.e. doggo, floofer, pupper, pupper or None are of same category type. Hence they must be merged into single column named 'DogStage' which shows these stages for different IDs. This can be done using pandas 'melt' function.

1.1.11 Eliminating all tweets data other than from WeReDogs

Since we have to analyse only tweets (No retweet) of WeReDog twitter handle, all other tweets must be removed from the table.

1.1.12 Sorting twitter_archive table and prediction by tweet_id

Since all the tweets are jumbled up, they must be sorted in one direction, which help in easy visual/programmatic tracing of an ID. So sorting in ascending order. Same is the case for IDs in prediction table.

1.2.2 The prediction table should only tweet IDs corresponding to in twitter_archive table.

The prediction table has 2074 observations, but 'twitter_archive_master.csv' has 1971 observation. Hence the prediction table must have only those IDs which are there in twitter_archive_master table.

2.4 One of prediction test must be part of twitter_archive master table

The test results of prediction algorithm must be part of twitter_archive_master table for a complete analysis and making data more tidy. Hence i will merged the data of first prediction test i.e. 'p1', 'p1_conf' and 'p1_dog' with the main table.