Data Wrangling Report

by Lakshmi Rajasekhar

Introduction

Data wrangling is an important art (more than science) which deals with cleaning up a ‘raw’ data into a much more useful form appropriate for further data analysis and modelling. Data wrangling is a required skill for any aspiring data-scientist as it gives her/him the power to transform a given raw data into a useful form for further analysis. There are basically three steps in any data wrangling project, namely,

- Gathering data
- Assessing data
- Cleaning data

This report details some of the wrangling effort (i.e., gathering, assessing and cleaning) for datasets related to ‘WeRateDogs’ twitter account. This report is part of my work towards Udacity’s data wrangling project from Data Analyst Nanodegree Program.

Data Gathering

Data gathering is the first step which deals with collecting required data from various sources. The datasets might be in various formats. This project required gathering three datasets as explained below:

(a) **DATA 1 - WeRateDogs Twitter archive -** given by Udacity direclty.

This dataset contains information like tweet_id, tweet text, dog ratings, image url etc for WeRateDogs tweets. This dataset has the original tweets information which would be used later on for further data gathering as detailed below.

(b) **DATA 2 - Tweet image predictions**

A ‘.tsv’ file downloaded from Udacity’s server using python Requests library. This tab separated file contains information like the breed of the dog (or other object, animal, etc.), three predictions for the breed of the dogs and the prediciton confidences. Udacity had done this for each

tweet in the dataset gathered above (WeRateDogs Twitter Archive) using a neural network model for predictions. This can help us identify the breed of the dog in the tweeted image as this information is missing most of the time in the twitter archive dataset above.

(c) **DATA 3 - Twitter API datadump**

This effort was to suppliment the above two datasets for more information like number of re-tweets, favourite counts, etc., for each tweet. This was done using python’s tweepy library by querying through a twitter API (Twitter provides a REST search API for searching tweets from Twitter’s search index).

This took some time as Twitter API has a rate limit on the number of tweets searched per 15-minute window. The data gathered using twitter API using tweepy is dumped into a ‘.txt’ file in a json format. Finally this text file is read back into the jupyter notebook and the interesting columns are extracted for further cleaning, analysis, and visualizations.

Data Assessing

Assessing the three datasets gathered in the section above required a lot of effort. I feel, a lot more needs to be done to complete the full assessing and cleaning up of these datasets. Mainly two techniques were used for assessing the datasets – manual and programmatic assessment. Manual assessment helps in splotting the most visually evident quality and tidiness issues. Manual assessment sometimes leads to maunal cleaning also if some errors are isolated and is difficult to be cleaned programmatically. Programmatic assessment is faster than manual but both types of aseessment are required for overall success in assessing any dataset. Also, since pandas collapses the records while displaying in jupyter notebook, programmatic assessment helps to identify and fix quality and tidiness issues not evident through simple visual assessment.

Any dataset can have mainly two types of issues – data quality and data tidiness issues. The steps below outline the assessing steps performed for this project so far:

Data Quality issues:

Data quality is what defines if a dataset is fit for a given use/purpose. The quality rquired for a dataset differs from project to project. This project required cleaning up the datasets mainly to check if the records/columns were valid, accurate, and consistent for further use.

Some of the data quality issues identified for each dataset were:

I. Data 1 - dogsTwitter_archive:

0. *The 'text' column is truncated in certain records* - replace the text column from the twitter API data dump.

1. *The dataset has retweets whereas we want only original 'dog' tweets with images* - Filter the dataset for enforcing this condition.

2. *Tweet text contains urls which needs to be cleaned* - This could very well be treated as a tidiness issue first. The text description most of the times has ‘tweet description’, rating, image url, breed etc. All these information can be separated out into columns and then the original ‘text’ column can be treated for ‘quality’ issue by removing the extra information other than the tweet description from the column. But, since this is a highly time consuming task and also because not all this information is present for each tweet text, I am currently considering this as just a quality issue and then removing the extra url embeddings from the text.

3. *HTML elements in 'source column'* - *t*This coulmn has html embeddings which can be removed and cleaned to find the source (type of phone etc) used for tweeting.

4. *Errors in dog names* - Some names are 'None'. Some names are listed as smaller case 'not', 'an', etc, which might most probably be wrong. Analyzing the tweet text, I found that sometimes, there are no names mentioned and hence can be treated as ‘None’ itself.

5. *Unusual values for rating_numerator and rating_denominator columns* – Some of the denominator values were 0 and the numerator values were incorrect. Sometimes, when there are more than one rating given for a tweet image, the first one is extracted by default. This most of the times is wrong and we need to extract out the last rating listed in the ‘text’ column. Hence the rating values needs to be corrected after proper extraction from the ‘text’ column.

6. *Convert columns to appropriate datatypes for easier analysis*

   i. tweet_id, in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id columns should be categories (i prefer it to be categories rather than strings)

   ii. retweeted_status_timestamp column should be datetime format.

7. *Check for 'None' or null values for dog_stage after merging the columns* - doggo, floofer, pupper, and pupp. Sometimes the dog_stage column has ‘None’ even after merging. Check if the dog stage is actually not recorded and hence ‘None’.

8. *Variation of null data representations in many columns (like 'None', NaN etc).*

II. Data 2 - twitter image predictions:

1. *Convert tweet_id column to category datatype*
2. *Missing records* - not all tweet_id records are present in the image predicitons file

III. Data 3 - twitter API datadump:

1. *Convert into appropriate dtypes:*
    i. 'tweet_id' column to category datatype
    ii. 'retweet_created_at' column to date_time format

Tidiness issues:

I. Data 1 - dogsTwitter_archive:

1. doggo, floofer, pupper, pupp columns should be merged into 1 column to represent the stage of dog. The original columns can be dropped after this step.

II. General for the whole project:

1. Merge all 3 datasets (twitter archive, image predictions and twitter datadump) to form one file for analysis (perform an inner join based on tweet_id) – All three datasets gathered needs to be merged based on tweet id to have a single cleaned master dataset for further analysis and visualization.

Data Cleaning

Data cleaning effort for this project consisted of correcting all the quality and tidiness issues detected in the above section. The data cleaning section is mostly divided into Define/Code/Test sub-sections. Sometimes, this was not possible as intermediate tests needed to be performed to check an intermediate step’s correctness and achieve the code cleaning task definition. As far as possible, I have tried to maintain the three sub-sections for each code cleaning task.

Final Thoughts/Future Work

The assessing and cleaning is an iterative process. I still have some ideas to clean the dataset further but due to lack of time, I am restricting myself to the effort I have detailed above.

As a future work, I would like to manually assess the few tweet_ids which returned errors while twitter searching and data scrapping. Since, one of my intention after cleaning and merging the datasets is to find the tweet with the largest number of re-tweets, a few errors might make a lot of difference (in case the highest re-tweeted one happened to be one of the tweet_ids in the error list). Currently, I am assuming that this is not the case.

I would also like to try various methods to extract the maximum amount of information from the twitter_archive ‘text’ column. A lot of things like tweet description, dog ratings, dog names, breeds, image urls etc are hidden in this one column. Hence, this column has a lot of quality and tidiness issues but also has a reserve of information, if extracted properly. Since, performing full text mining on the ‘text’ column in the twitter archive dataset itseld would take a lot of time, I have limited myself to just cleaning the ‘text’ column to remove the embedded urls and extract the correct ratings.

Another intersting thing to see would be to see what factors cause some images (dogs) to get higher rating than usual.

Summing up, the future work I intent to do is as listed below:

1. Check tweet_ids which gave status error while extracting using Twitter API and tweepy.
2. The image prediction file has dog breed prediction. Some of the tweet images has dogs + some other element (e.g., a shopping cart). The prediction model predicts the image as shopping cart with the highest confidence, which is errorenous. This should be corrected in the file. Also, some images are not of dogs. Such records should also be removed from the dataset.
3. Correct '&amp' in tweet 'text' column to just '&'.
4. The tweet 'test' column can be further cleaned to remove all ratings and to have only text.
5. The twitter API scrapped data dump file could be used to see if missing image files for tweets are available. If so, those records needn't be dropped just because they don't have images in the given file.

6. Optimising the code for faster execution. Currently, only 'cleaning' the data has been concentrated on. Ideally, the code needs to be made faster too.

References:

1. Wikipedia: https://en.wikipedia.org/wiki/Data_wrangling
2. https://www.karambelkar.info/2015/01/how-to-use-twitters-search-rest-api-most-effectively./