wrangle_report

August 16, 2019

Gathering Data There are three main pieces of data involved in the wrangling process:

1. A twitter archive dataset from WeRateDogs in CSV format (provided and was added to workspace manually, called 'twitter-archive-enhanced.csv'). It contains tweet information like tweet id, timestamp, tweet text, source, dog rating numerator and denominator that were extracted from the tweet texts, dog stages, etc. It was read into dataframe 'twitter_archive_enh'.

2. Dog breed or other object prediction information based on each tweet's images running through a neural network, in a TSV file format to download programmatically from a Udacity server (image-predictions.tsv, read into dataframe 'image_predictions');

3. Each tweet's retweet and favorite counts as well as tweet text display range indexes extracted from tweet JSON data by calling the Twitter python API Tweepy. (Tweet entire JSON data was saved as 'tweet_json.txt', subset dataframe created with the needed info is 'df_tweet_json'.)

Assessing and Cleaning Data

Visual Assessing: Opened the twitter-archive-enhanced.csv in Excel to do some initial visual check and assessment. There were quite several issues spotted with both quality and tidiness in the dataset:

1. The source column has dirty html tag characters that makes it hard to tell the field values.

***Define**: Use .replace() to replace the html tag characters to empty string pattern identified by regular expression, cast field dtype to category using .astype().*

2. tweet_id field is incorrectly interpretated as a numeric column.

***Define**: Cast the field dtype to str using .astype().*

3. text column contains not just texts, but some urls at the end.

***Define**: use the display_text_range indices to slice the text field. Join tweet_json_clean with twitter_clean (.merge()), split 'display_text_range' column into two columns 'start_index', 'end_index' (.str.strip().str.split()), then use the index columns to slice 'text' field for cleaned text (.apply()). Drop the no-longer-needed index columns at the end.*

4. The four dog stage columns should be one variable and the 'None' should be null to indicate there's a missing value.

***Define**: First set all rows with multiple dog stages to 'None' (.loc[]), change all 'None' values to empty string, add a new 'dog_stage' field concatenating all four stage field values (.loc[].apply()), set record with no stage to null, drop the no-longer-needed columns (.drop()) and finally update the stage column to category type (.astype()).*

5. Several columns rarely contain much data and will not be used, i.e. 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp'.

***Define**: drop unused columns with .drop().*

Programmatic Assessing: Used several panda's methods and functions like .info(), .describe(), .value_counts(), boolean array indexing, .str.extractall() to examine the dataframes one step further, confirmed some of the issues spotted above as well as found more others:

6. twitter_archive_enh contains retweets besides original tweets.

***Define**: retweets can be distinguished from typical Tweets by the existence of a retweeted_status attribute. Thus used boolean array mask indexing, i.e. twitter_clean[twitter_clean.retweeted_status_id.isnull()]*

7. Incorrect dtype for twitter_archive_enh's 'tweet_id', 'timestamp', 'source', 'dog_stage' and image_predictions' 'tweet_id'.

***Define**: Cast the field types to string, datetime, category using .astype() and pd.to_datetime().*

8. Field 'name' contains incorrect values like 'a', 'an', 'the', starting with non-capitalized letter in twitter_archive_enh.

***Define**: Use boolean array mask to filter out all non-capitalized names (.name.str.match()) and assign to null, update "None" names to null as well.*

9. Some rating numerators and denominators are incorrect in twitter_archive_enh.

***Define**: use regular expression to extract all matching patterns from 'text' (.str.extractall()), visually check the few records that have more than one matches to record the correct ratings. Join the new extracted rating dataframe with twitter_clean (.join()), split the column to update the numerators and denominators with the true derived ratings to include decimal point ratings, etc.*

10. All three tables/dataframes are describing each tweet's info thus should be combined to one.

***Define**: merge image_predictions df with twitter_clean, the master dataframe.*

```
In [ ]:
```