wrangle_report

December 17, 2018

1 Wrangle report

1.1 Data

We use three datasets in this project, `twitter-archive-enhanced.csv`, `image_predictions.tsv` and `tweet_json.txt`. We also create a new dataframe `dogs_information.csv`

`twitter-archive-enhanced.csv`:

This dataset record almost every basic attribute about tweet data.

`image_predictions.tsv`:

This dataset do a prediction about pictures in tweets in `twitter-archive-enhanced.csv`.

`tweet_json.txt`:

This file record every detail about tweets in `twitter-archive-enhanced.csv`.

`dogs_information.csv`

This dataset include 'name', 'doggo', 'floofer', 'pupper', 'puppo', 'rating_numerator', 'rating_denominator', 'tweet_id' and some columns in image_predictoin.tsv.

1.2 Collect

`twitter-archive-enhanced.csv`:

We use this dataset as default one, just upload it to the workspace.

`image_predictions.tsv`:

We use `requests` packaege to download the data from internet and save it into the file `image_predictions.tsv`.

`tweet_json.txt`:

We download it from provided link and upload it into workspace.

Conclude

We have three dataset to store above information:

`tweet_info` to store `twitter-archive-enhanced.csv` (pandas dataframe)

`image_info` to store `image_predictions.tsv` (pandas dataframe)

`tweet_json_info` to store `tweet_json.txt` (list, if we decide to use some attributes, we'll extract those attributes directly from this list)

1.3 Assess

1.3.1 Quality

`twitter-archive-enhanced.csv`:

First, as requested, we notice there are some retweeted tweets in this dataset, and tweets don't have photos.

Next, we observe type of each columns, and find:

- `tweet_id` is type int64
- `in_reply_to_status_id` is type float64
- `in_reply_to_user_id` is type float64
- `retweeted_status_id` is type float64
- `retweeted_status_user_id` is type float64
- `timestamp` is object type
- `retweeted_status_timestamp` is type object

Then, we take a close look at each column. From left to right, we have found following problems:

- `timestamp` column has +0000 at end
- `source` column seems has tag a, and href link
- `retweeted_status_id`, `retweeted_status_user_id` and `retweeted_status_timestamp` maybe not need.(If we delete all retweeted tweets, then there's no need to keep these three columns.)
- `name` column has name displayed as None, which isn't np.nan in pandas
- `name` column have name begin with [a-z], like 'a', 'an', 'the', and others, obviously not a dog name.
- In `doggo`, `puppo`, `pupper` and `floofer` columns, there are data displayed as None, which is not a np.nan type.
- There are 12 records have `doggo` and `pupper`, 1 record has `doggo` and `puppo`, and 1 has `doggo` and `floofer`.
- There are some wrong values in `rating_denominator` and `rating_numerator` columns.

`image_predictions.tsv`:
First, it has too many columns we don't need.
Second, `tweet_id` column is type int64.
`tweet_json.txt`:
It has too many information won't be used.

1.3.2 Tidy

`twitter-archive-enhanced.csv`

- Last four columns, `doggo` `floofer` `pupper` `puppo` can be integrate into one column `Stage`
- columns `numerator` and `denomitor` can be one column `score`

`image_predictions.tsv`:

- image_info should be a part of tweet_info

`tweet_json.txt`

- There are some attributes we need, these attributes should in `tweet_info` dataframe.

1.4 Clean

We copy three original datasets.

Retweeted tweets

We choose to solve this quality problem first. We find those retweeted tweets and delete them.

Redundant columns

After delete those retweeted tweets, we find `retweeted_status_id`, `retweeted_status_user_id`, `retweeted_status_timestamp`, `in_reply_to_status_id` and `in_reply_to_user_id`seems useless, so we drop them.

Wrong score

We first extract score from text directly,then we examin some strange data one by one and fix them.

Missing data

To add attributes in `tweet_json.txt` into `tweet_info_clean` dataframe(which is the copy of tweet_info), we extract `followers_count`, `retweet_count` and `favorite_count` from `tweet_json_info` then merge them into `tweet_info_clean`.

Multi Stage

We build a new dataframe `df_stage` to solve this problem.

From task, we only need `doggo`, `floofer`, `pupper`, `puppo`, `rating_numerator` and `rating_denominator` columns in `df_stage` and drop those columns in `tweet_info_clean` at the same time. For completeness, we also include `tweet_id` in case we need to query from `tweet_info_clean`.

Then we check tweets have multi stage, and fix them.

Image info drop

We choose the most confident prediction and preserve it to merge with `tweet_info_clean` and `df_stage`.

Merge four Stage columns

It make sence to merge four stage columns `doggo`, `puppo`, `pupper` and `floofer` into one column `Stage`. Set None value in and `Stage` at the same time. This work is done on `df_stage`.

Calculate numerator/denominator

Calculate result of numerator/deominator, store it in the new column `score`, then drop columns `rating_numerator` and `rating_denominator`, this work is done on `df_stage`.

Add image_info

We merge image information processed in **Image info drop** step with `tweet_info_clean` and `df_stage`.

Timestamp

Extract time in timestamp column as form: %Y-%m-%d %H%M%S, make it easier for next process.

Wrong types

- change tweet_id type to string (both `tweet_info_clean` and `df_stage`)
- change in_reply_to_status_id to type string
- change in_reply_to_user_id to type string
- change timestamp to object datetime
- chnge img_num to type string

Some columns have been deleted, so there are less columns need change type.

Source with tag < a >

We extract text betweet tag < a >, and save them in source column.

Name column

We turn None in name column into np.nan, and extract those words begin with [a-z], set them to np.nan, too.

1.5 Save

Finally, we save `tweet_info_clean` into `twitter_archive_master.csv` , save `df_stage` into `dogs_information.csv` and use them in the following analyse.