wrangling_steps

December 13, 2020

1 We_rate_dogs data rangling steps:

2 Firstly Gathering data.

2.1 Gathering is the first step in data wrangling. Where we started gatheing data from different sources.

2.1.1 Input data sources :

1. **Enhanced Twitter Archive.csv :** this file was `downloaded manually` into the directory of the project.
2. **Image Predictions File (.tsv) :** this is a file contaning prediction data about the dogs from running the tweets images through neural network and other info is given too, this file will be downloaded programmatically using the `Requests` library.
3. **Additional Data via the Twitter API :** Be it more information about each tweent in the given period like (retweet count and favorite ("like") count, and any additional data, this should be handled through `tweepy using tweeter developer account`, **But** we used `tweet-json.txt` file provided as we had difficaulty setting developer account on twitter.

2.1.2 Output data files :

1. **enhanced_archive_df**
2. **Image_predictions_df** & image-predictions.tsv
3. **Tweet_json.txt / api_df**

2.1.3 1. importing the `Enhanced Twitter Archive.csv`

After downloading in mannually in the project dir. we imported the first CSV file

2.1.4 2. programmatically download the `Image Predictions File.tsv`

We downloaded the second TSV file using the given like programatically.

2.1.5 3. Additional Data via the Twitter API

we accessed the project data without a twitter account as we hade problems with the twitter developer account. so we used the given "tweet-json.txt" and read it line by line to extract our data.

-

2.1.6 Here we finished the Gathering Phase of the data and we have 3 data frames that we will use to asses and clean

3 Assessing data

3.1 Assessing is the second step in data wrangling. Where the inspection of our collected data sets of both the Quality and Tidiness perspectives will be conducted.

3.1.1 Input data :

1. **enhanced_archive_df**
2. **image_predictions_df**
3. **api_df**

3.1.2 Process :

`Two types of assessment are used`**:**

1. `Visual assessment`: each piece of gathered data is displayed in the Jupyter Notebook for visual assessment purposes. or additionally be assessed in an external application (e.g. Excel, text editor).
2. `Programmatic assessment`: pandas' functions and/or methods are used to assess the data.

3.1.3 Output :

`Assessment Summary report containing :`

1. quality issues found
2. tidiness issues found

3.2 Visual assessment

- we scrolled through the data in diffirent programs tring to find and quality or tidiness issues.

3.3 Programmatic assessment

- by using pandas functions/methods to try and extract any hidden issues with the data

3.4 Assessment documentaion.

3.4.1 Quality aspects:

enhanced_archive_df:

1. `Validity issue`: number of records are retweets and replaies, we only need tweets with images in them.
2. `Validity issue`: number of records are tweets with no image prediction data in **"image_predictions_df"**.
3. `Completeness issue`: pets classification columns : "doggo", "floofer", "pupper" and "puppo" have missing data 'None'.
4. `Consistency issue`: **"name"** feature needs consistant data for missing names : "None" and some don't make sense "a" , "the", "an" usually dog name is not mintioned in the tweet need to change all to 'null'.
5. `Accuracy issue`: **"name"** feature have number of inaccurate names "a" , "the", "an", "O" to name a few.
6. `Accuracy issue`: **"name"** feature have number of inaccurate names starting with lower case.
7. `Consistency`: **"time stamp"** is in string formate should be changed to date time formate and stored in deffirent colomns.
8. `Use case validity issue`: Some columns that will not be of use in our analysis are: **"in_reply_to_status_id"** , **"in_reply_to_user_id"**, **"retweeted_status_id"**, **"retweeted_status_user_id"** and **"retweeted_status_timestamp"** .
9. `Completeness issue`: **"expanded_url"** column of the archive_df have missing values for tweets without photos.

image_predictions_df:

1. more descriptive colomn names.
2. p1_dog, p2_dog and p3_dog, have 324 images that have are not dogs, all dog predections are False.
3. number of retweets and replaies are present and we only need tweets with images in them. Validity issue.

api_df:

1. number of retweets, replaies and tweets that do not have image prediction present in the data frame and we only need tweets with images in them. Validity issue.

3.4.2 Tidiness aspects:

enhanced_archive_df:

1. "`values are column names`: pets classification features: **"doggo"**, **"floofer"**, **"pupper"**, **"puppo"** should be in one colomn of (pet_class) of one or compination pet stage.

image_predictions_df:

1. **"image_predection_df** Column headers are values, not variable names.

api_df:

1. Two tables **"api_df"** and **"enhanced_archive_df"** have a common observation unit (the tweet it silfe) need to combined in one df.

4 cleaning data

4.1 defining the issues and the solutions we used.

4.1.1 quality issues resolving :

1. Removing all **archive_clean_df_1** rows that does not contain imeges in **image_clean_df_1**.
2. Removing all the rows that are retweets and replays in **archive_clean_df_1**.
3. Removing all the rows that are retweets and replays in **image_clean_df_1**.
4. Removing colomns (in_reply_to_status_id , in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp) in **archive_clean_df_1**.
5. removing **"expanded_url"** column of missing values for tweets without photos.
6. Removing all the rows that are retweets, replays and tweets without images from in **api_clean_df_1**.
7. changing some **image_clean_df_1** colomn names to a more descriptive names.
8. **name** feature needs consistant data for missing names : "None".
9. **name** feature some data don't make sense "a" , "the", "an" ,(a dog name allways start with an upper case)usually dog name is not mintioned in the tweet need to change all to 'NA'.
10. **timestamp** colomn is in string formate will be changed to date time formate and stored in deffirent colomns.
11. **"doggo"**, **"floofer"**, **"pupper"**, **"puppo"** All "None" values will be changed to NaN.

4.1.2 Tidiness issues resolving:

1. pets classification features: **"doggo"**, **"floofer"**, **"pupper"**, **"puppo"** will be in one colomn of (pet_class) of {**pupper, doggo, puppo, doggo-pupper, floofer, doggo-floofer, doggo-puppo}.**
2. The two tables **"api_df"** and **"enhanced_archive_df"** will be merged.
3. **"image_predection_df** have values of predection algorithm **p1_dog**, **p2_dog** and **p3_dog** as colomn variables.

- Note : The dataset has two observational units the first is the tweet (the archive and api datasets) and second is the images (the image prediction dataset).

4.2 code of the cleaning process.

4.2.1 taking a copy of the raw imported data frames

4.2.2 1. removing all rows from `archive_clean_df_1` with no images in `image_clean_df_1`.

4.2.3 2. Removing all the rows that are retweets and replays in `archive_clean_df_1`.

4.2.4 3. Removing all the rows that are retweets and replays in `image_clean_df_1`.

4.2.5 4. Removing all unused colomns `archive_clean_df_1`.

4.2.6 5. changing `image_clean_df_1` colomns names to more discriptive names

4.2.7 6. Removing records that are retweets, replays and tweets without images from in api_clean_df_1.

4.2.8 7. `"name"` feature needs consistant data for missing names.

4.2.9 8. changing time stamp to date time formate

4.2.10 9. "doggo", "floofer", "pupper", "puppo", All "None" values will be changed to NaN.

4.2.11 10. image_clean_df_2 columns names tideness.

4.2.12 11. Merging the two archive_clean_df_2 and api_clean_df_2 into one Data frame "master_tweets_df".

4.3 test of the cleaning code

- the testing code is used to verify the effectivness of the cleaning steps on the data.

5 Storing data to external CSV files

-

-