Wrangle Report Processes

Data wrangling is the process of removing errors and combining complex data sets to make them more accessible and easier to analyse. Due to the rapid expansion of the amount of data and data sources available today, storing and organizing large quantities of data for analysis is becoming increasingly necessary.

A data wrangling process, also known as a data munging process, consists of reorganizing, transforming, and mapping data from one "raw" form into another in order to make it more usable and valuable for a variety of downstream uses including analytics.

The wrangle and analyse data project was provided by Udacity team as a part of the Data Analyst Nanodegree Project. The project involves wrangling of data to achieve a perfect analysis. Twitter user @dog_rates, commonly known as WeRateDogs provided their twitter archive for analysis.

WeRateDogs rate's picture of other people's dogs in a humorous manner and often provide ratings more than 10 and denominator always 10.

This report includes the steps taken to wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information.

STAGE 1: Data Gathering

The data of this project consists of three different data sets:

1. ***twitter_archive_enhanced.csv***: Directly download CSV file using *pd.read_csv* import into pandas data frame.

2. ***image_predictions.tsv***: Programmatic download from Udacity's server. The tweet image predictions is present in each tweet according to a neural network. This file is

hosted on Udacity's servers and downloaded programmatically using the *requests* library.

3. ***Tweet_json.txt:*** Each tweet's retweet count and favorite (i.e. "like") count at minimum, and any additional data I will find interesting. Using the tweet IDs in the WeRateDogs Twitter archive, using tweet_json.txt(I was not able to successfully open a Twitter Developer account) gotten from the resulting data from twitter_api.py.

STAGE 2: Accessing Data

This step allows us to identify quality and tidiness issues. Low-quality data are considered as dirty data and have content issues. We must assess at least eight quality issues. As for untidy data, they are considered as messy data and are structural issues.

There are two types of assessment:

● Visual assessment, which consists in scrolling through the data

● Programmatic assessment, for which we use python statistical libraries such as pandas, numpy, etc.

Issues:

Tidiness Issues:

1.Merging the three dataframes into one using tweet_id.

2.full_text: Some tweets reference more than one dog and include multiple stages and ratings. I will create a new column called stage to include the dog stage and delete the four columns. For tweets that reference multiple stages, the stage will include additional stages, such as: doggo|pupper, doggo|floofer, and doggo|puppo.

Quality Issues:

1.Remove null retweeted_status_id from the dataset

2.Delete the timestamp column from the merged dataset as I included the created_at column in the tweet_df dataframe,Separate created_at into day - month - year (3 separate columns) for making it simple and efficient.

3.Parse the column source to show the direct source (for ex: iphone, tweetdeck, etc.) rather than the HTML statement.

3.I found 1 dog that is both doggo and floofer, I found 12 dogs that are both doggo and pupper and I found 1 dog that is both doggo and puppo.

4.retweeted_status_id, retweeted_status_id, retweeted_status_user_id: I found 181 retweets. The retweets and the respective columns were removed from the dataset.

5.p1, p2, p3: Some entries are lower case. also, some entries have underscore.

6.In the text, we can notice some decimal numbers for the ratings numerator part wrongly extracted.

7.Correct erroneous names

8.Tweet_id, sources and img_num need to be converted into the right datatype

STAGE 3: Cleaning Data

1. Merged the 3 datasets using INNER join.

- The number of rows of the merged dataset is 2074.
- I created a new dataframe called df_master to work on and kept the original dataframe for reference.

2. Deleted retweets from the dataset

3. Deleted the extra columns:

- I deleted the following columns: retweeted_status_id, retweeted_status_user_id,retweeted_status_timestamp, timestam, text, in_reply_to_status_id, in_reply_to_user_id

4. Created separate columns for day, month and year based on the created_at column.

5. Parsed the column source to show the direct source (for ex: Tweeter for iPhone) rather than the full link.

6. Converted tweet_id to string.

7. Dropped duplicates.

8. Re-formatted the content of columns p1, p2, p3.

- converted to upper case and removed underscores.

9. Corrected erroneous values in the name column and converted the (None) values to Null.

10. Created a new categorical column called stage to include the dog stages and deleted the old 4 columns.

11. Resorted the dataframe columns.

STAGE 4: Storing, Analysing and Visualizing data

In this part had have to store our results into files and make documentation on it. After analysing the data, we can provide many insights from the data set that will come in handy. Also making visual plots or charts that will further enhance the findings and will make things much simpler.

Conclusion

Through the data wrangling and analysis, we used many libraries such as pandas, NumPy, requests and json, which allow us to gather, assess, and clean the data. Finally, we put the following documents together:

● wrangle_act.ipynb: code for gathering, assessing, cleaning, analyzing, and visualizing data

● wrangle_report.pdf: documentation for data wrangling steps: gather, assess, and clean

● act_report.pdf: documentation of analysis and insights into final data

● twitter_archive_enhanced.csv: file provided

● image_predictions.tsv: file downloaded programmatically

● tweet_json.txt: file provided via twitter_api.py

● twitter_archive_master.csv: combined and cleaned data