WeRateDogs – Wrangle Report

Contents

Introduction:

The dataset wrangled in the project is the tweet archive of Twitter user **@dog_rates**, also known as **WeRateDogs**. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. WeRateDogs has over 4 million followers and has received international media coverage.

I have wrangled WeRateDogs Twitter data to create interesting and "*Wow!*"-worthy analyses and visualizations.

Data Wrangling:

In Data Wrangling, our goal is to gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then cleaning it. This is called **data wrangling**.

We need to wrangle our data for good outcomes, otherwise there could be consequences. If we analyze, visualize, or model our data before we wrangle it, our consequences could be making mistakes, missing out on cool insights, and wasting time. So, best practices say wrangle always.

Steps in Data Wrangling,

- Data Gathering
- Data Assessing
- Data Cleaning

Data Gathering

To make the trustworthy analysis, I have gathered the WeRateDogs Twitter data from three various resources.

- **Enhanced Twitter Archive**
  I have extracted the basic tweet data (tweet ID, rating, dog name, and dog stage,timestamp, text,etc.) from `twitter_archive_enhanced.csv` - Twitter archive provided by Udacity. Stored the extracted data into `archive_df` dataframe.

- **Image Predictions File**
  Extracted a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction from `image_predictions.tsv` - image-predictions provided by Udacity. Stored the gathered predictions data into `img_predict_df` dataframe.

- **Twitter API**
  To query the additional information from Twitter API, I obtained the API retrieved data provided by Udacity in the txt file. The JSON file was converted to csv using an online tool called JSON-To-CSV . The obtained file was then stored in the `tweet_api_df` dataframe.

Data Assessing

To find out the quality and tidiness issues performed two types of assessment,

1. **Visual Assessment**
   Each piece of gathered data assessed manually displayed using Jupyter Notebook and other external applications like excel.
2. **Programmatic Assessment**
   Assessing data using python Pandas functions and methods.

Quality Issues are analyzed based on the below quality dimensions,

- ✓ Completeness
- ✓ Validity
- ✓ Accuracy
- ✓ Consistency

Quality Issues

Twitter Archive Data

- Erreneous datatypes: tweet_id, timestamp
- Retweet and Reply observations present along with original tweet ratings
- Unwanted columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp
- Source not extracted properly from hyperlink tag
- Rating numerator and denominator can be converted from int to float as the ratings have decimal values
- Unstandardized ratings:
  - Some Rating numerator and denominator are not matching with the tweet text
  - Some tweets doesn't having ratings

        - Decimal ratings are present for some tweets which is not matching with Rating numerator and denominator
- Ambiguous names present
- &amp ; and \n codes directly present in tweet text
- Url present in both tweet text and expanded_urls

Image Predictions Data

- Erreneous datatype tweet_id
- Predicted dog names contains difference kinds of casing
- Dog breed contains underscore and hyphen characters

Twitter API Data

- Erreneous datatype tweet_id

Tidiness Issues

Twitter Archive Data

- Individual columns for dog stages (doggo, floofer, pupper, puppo) which shoule be combined into single column Dog stages

Image Predictions Data

- Most confident image predictions are enough. Contains three set of image predictions including False predictions

General

- tweet_id column is duplicated in gathered datasets(archive, img_pred, tweet_api). Gathered three datasets can be combined into one single dataframe to improve tidiness

Data Cleaning

Before making cleaning operations, I took copy of assessed datasets using `.copy()` method so that we can still view the original dirty and/or messy dataset later.

```
archive_clean = archive_df.copy()

pred_clean = img_predict_df.copy()

tweet_clean = tweet_api_df.copy()
```

Data cleaning process contains below three sets,

❖ Define
❖ Code
❖ Test

Further Steps,

- Removed retweets and replys observations in `archive_clean` by ignoring the rows which has values for `retweeted_status_id` and `in_reply_to_status_id` using 'isna' method from `archive_clean`
- Dropped unwanted columns and columns had more null values using `drop()` method `in_reply_to_status_id`, `in_reply_to_user_id`, `retweeted_status_id`, `retweeted_status_user_id`, `retweeted_status_timestamp` from `archive_clean`
- Combined individual dog stages columns (doggo, floofer, pupper, puppo) into single column `dog_stages` and replaced 'None' with empty string. For the rows which contain multiple dog stages assigned names manually and dropped individual columns.
- Selected most confident TRUE image predictions and ignored the observations which have FALSE predictions. Stored the combined result in new columns `breed` and `confidence`
- Merge gathered datasets into single dataframe `twitter_data_cleaned` and dropped duplicate column `tweet_id`
- Corrected Erreneous datatypes : `tweet_id`, `timestamp`, `rating_numerator`, `rating_denominator` using **.astype()** and **to_datetime()** methods.
- Extracted source from hyperlink tag properly and removed html tags using ***re.sub()*** method.
- Corrected the dog breed which contains underscore and hyphen characters with empty string and normalized the dog breed string casing using ***str.title()***
- Removed HTML code entities from tweet text by replacing '&amp' with '&' and '\n' with empty string.
- Removed Url from tweet text as both tweet text and expanded url contain same urls using regex expression and replaced the urls in text with empty string.
- Dropped all null values to optimize the dataframe using ***dropna()*** method.

Conclusions

After Performing Data wrangling on the WeRateDogs Twitter Data, I have stored the wrangled data into ***twitter_archive_master.csv*** which contains 1963 entries and 15 columns.

Note: The master data set isn't fully free of issues always reassess and iterate on any of the data wrangling steps if necessary as Data Wrangling is an iterative process.