WRANGLE REPORT

Ebenezer Maradesa
ALX-T Data Analytics
Project 2: Data Wrangling – Wrangle Report

Table of Contents

INTRODUCTION....1
TOOLS....1
GATHERING DATA....1
archive_df:....2
image_df:....2
add_info_df:....2
ASSESSING DATA....3
Quality Issues:....3
Tidiness Issues:....3
CLEANING DATA....3
STORING DATA....4

INTRODUCTION

This report details the wrangling efforts applied on the dataset relating to the tweets of the twitter account WeRateDogs (@dogrates).

TOOLS

Wrangling was done using Python, particularly the Python Pandas and NumPy libraries in a Jupyter Notebook.

GATHERING DATA

Data was gathered from three sources using three different methods:

- The WeRateDogs twitter archive, available in a file, ‘twitter-archive-enhanced.csv’. This was loaded into the Jupyter Notebook using the pandas.read_csv method as ‘archive_df’ dataframe.
- The tweet image predictions file, programmatically downloaded from this url, using Python ‘requests’ library, saved into ‘image_predictions.tsv’, and read into the Notebook ‘image_df.’

- Twitter API using Python's tweepy and json libraries. The tweet jsons were stored into 'tweet_json.txt', and the newly scraped data was stored in 'add_info_df.'

The dataframes to work with are thus:

archive_df:

Contains:

- tweet_id: unique identifier for each tweet
- in_reply_to- columns: reply data
- timestamp: of tweet
- source: the url of the tweet
- retweeted_status- columns: retweets data
- expanded_urls
- rating- columns: rating of dog in tweet
- name: of dog in tweet
- doggo, floofer, pupper, puppo: dog stage columns

image_df:

Contains:

- tweet_id
- jpg_url
- img_num
- probability prediction columns

add_info_df:

Contains:

- tweet_id
- retweet_count and favorite_count
- tweet_length
- hashtags

ASSESSING DATA

Assessment was done both visually and programmatically to gain insights into the quality and tidiness issues in the dataset. After assessing, the following were noted (some on iterations):

Quality Issues:

archive_df:

1. None Dog: tweets not about dogs
2. DataType: tweet_id is int, rather than string
3. DataType: timestamp columns are string
4. Missing Data: NaN values in in_reply_to- and retweeted_status- columns
5. Missing Data: in expanded_urls column
6. Retweets
7. Dog Names: None, ‘a’, etc.
8. Incorrect rating_numerator
9. Inconsistent scaling in rating_denominator

image_df:

8. None dog tweets
9. DataType: tweet_id is int

add_info_df:

10. DataType: tweet_id is int

Tidiness Issues:

1. Single variable (dog stage) in four columns (doggo, floofer, pupper and puppo)
2. tweet information present in three different tables

CLEANING DATA

The cleaning was not done sequentially as noted, as steps depended on others and some required iterations. Cleaning was done using the Python Pandas library. Summarily, using the Define, Code, Test pattern, the following were done:

- copies of each dataframe were created
- dog_stage columns were melted into a single column, according to this stackoverflow answer
- missing values were recovered by revisiting the tweet_json.txt file, while others were replaced by none
- wrong datatype columns were converted to the appropriate ones, e.g. timestamp was converted to datetime, tweet_id converted to string
- retweets were dropped
- None Dog tweets were dropped, using the predictions from the image_df to filter the archive_df, after the image_df had been trimmed to contain only the most likely prediction for each image
- incorrect ratings were re-extracted
- all ratings were scaled to be over 10
- other unneeded columns were dropped, including: retweeted_status- columns, rating_denominator column
- a new dog_info dataframe was created containing tweet_id, name, rating, favorite_count, retweet_count
- all initial three dataframes were merged into one: ‘archive_master’

STORING DATA

Three resultant dataframes were then stored as follows:

archive_master was stored into ‘twitter_archive_master.csv’

archive_clean was stored into ‘archive_clean.csv’

image_clean was stored into ‘image_clean.csv’

dog_info was stored into ‘dog_info.csv’