Data Wrangling Project

Wrangling action Report

Introduction

In this project, we are working on wrangling data about rating dogs. This data came from 3 different sources of varying degrees of quality and tidiness, which necessitates the process of Data wrangling, in order to obtain clean, tidy and consistent data.

Gathering data

The data came from 3 sources

- A downloadable csv file, which is an enhanced twitter archive containing information about
    - Tweet technical data: like tweet_id, in_reply_to_user_id, retweeted_status_id, expanded_url, text
    - Data parsed from tweet text like dog name, dog stage name, rating expressed in terms on numerator and denominator

- Data accessible via the python “requests” library. This dataset is a result of passing the tweet images through a neural network to identify the dog breeds in the images. This dataset includes information about:
    - Tweet id from which the image was used
    - Top 3 predictions of the neural network

- Twitter API. For this, I had to create a twitter developer’s account to query for tweet data using that API. The data was loaded into a text file to protect my access tokens. After that, this text file was used to create a pandas dataframe out of it. This dataset was mainly used
    - Correct some issues in the enhanced archive dataset
    - Retrieve more vital information about the tweets, such as retweet count, and favorite count

---

Assessing Data

Before jumping in to clean the data, we need to first identify these problems.

Upon visual assessment, I could identify the following issues:

- While loading the data from the twitter API, some tweets failed to load, because they were probably deleted. This meant that the twitter API dataset had fewer rows than the enhanced archive dataset.
    - This was furtherly corroborated by programmatic assessment when checking the shapes of both datasets

- Enhanced archive dataset had columns that were values instead of being variables of their own. Specifically, they were the columns “doggo”, “floofer”, “puppo” and “pupper” which could all be grouped under a single column like “stage_name”

- Rating denominator had a column of its own. I would argue that this shouldn’t be the case, and that the rating should be contained in one variable, not two. Upon further inspection, it turned out that the denominator was sometimes not 10 for these reasons:
    - Some tweets had many dogs, so the twitter user multiplied both denominator and numerator by a factor
    - Some tweets had errors in parsing, which led to incorrect results in both numerator and denominator columns
- Some tweets in the archive were not rating tweets. They were replies and retweets of the twitter user, so these records need to be removed.

- Not each table formed an observational unit. Tweet data is spread across all dataframes.

- Column names in the image prediction dataset can be ambiguous. Columns such as “p1”, “p2”, “p1_dog”,... do not reflect the meaning of the variable.

Now as for programmatic assessment, I noticed the following issues:

- Columns “name”, “floofer”, “doggo”, “puppo” and “pupper” all had the string value “None” instead of Nan. This would adversely affect any analysis procedure that searched for null values, as it would return false.

- Dog names were inconsistently lower case and upper case. This is because the names are not parsed correctly.

- Some columns had incorrect datatypes, like the “timestamp” column in the enhanced archive which was object, instead of datetime

---

Cleaning the data

Before anything, I created 3 copies of the 3 dataframes. After that I proceeded with cleaning the data as follows

1. First, Missing data issues:
    - Removed records in the archive data that did not exist in the data retrieved by the API
    - Removed records in the archive data that were retweets or replies, instead of being original tweets. This also meant that I had to remove the corresponding data in the twitter API dataset and the image prediction datasets.

2. Second: The denominator column issue
    - I had to fix this issue manually, because I couldn’t think of a way to do it programmatically. The rating was not placed systematically across all tweets and some random text can resemble the rating format. To do that, I did the following:
        - Filtered all tweets that had a denominator of value not equal to 10
        - Wrote these tweets inside a csv file, with their url
        - Used the urls to visit the tweet, and corrected the numerators and denominator manually
        - For pictures whose numerator and denominator were multiplied by a factor, I divided the numerator and the denominator by the factor to make all ratings out of 10
        - Read the csv file again and modify the archive data accordingly.
        - NOTE: the cells used to create the csv file are now omitted, because running them would overwrite the manual work.
        - After that, I dropped the denominator column, since it would be redundant.

3. Third: tidiness issues:
    - Grouped “doggo”, “floofer”, “pupper”, “puppo” columns under one column of name “stage_name”
    - Managed to group the data into 1 entity “observation unit” , which is the “tweet_df”, which holds all information about the tweets.

4. Other quality issues
    - Converted the string “None” to Nan in the name column. The first tidiness issue automatically solved this issue for “doggo”, “pupper”... columns
    - Corrected dog names by checking manually for the dog names that are not parsed successfully. This was done similar to the way incorrect denominators were fixed
        - Filtered all tweets that had a lowercase dog name
        - Wrote these tweets inside a csv file, with their url
        - Used the urls to visit the tweet, and corrected the dog names manually.
        - Read the csv file programmatically, and modify the archive data accordingly.
        - NOTE: the cells used to create the csv file are now omitted, because running them would overwrite the manual work.
    - Whilst checking the dog names manually, I also noticed that some tweets are about other animals-referencing dogs in a funny way-. So I removed these tweets as well from the dataset.
    - Changing datatypes to more suitable datatypes, like timestamp from object to datetime
    - Changing the column names in the image prediction dataset into more meaningful names
      “P1”, “p2”,... to “prediction_1”, “prediction_2”, ...