ALX-T DATA ANALYST NANODEGREE

WRANGLE REPORT

Introduction

We were tasked to gather data from data from different sources using different methods: manually downloading the data and uploading it, using the `request` library and scraping off twitter using its API and the tweepy package. I was able to do the first two and opted for the manual download for the last one because I couldn't get my twitter developer account approved. The three dataset were named enchanced_df, prediction_df and extracted_df.

Data Wrangling

After loading the three datasets into a dataframe, I had looked through the dataset using `.head()` and `.sample()` to see If I could discover any inconsistencies. I was able to discover the weird names in the name column of the enhanced_df which I also confirmed using the the `enchanced_df.name.unique()`. The weird names include None, indefinite articles, possessive pronouns, all of which started with a lowercase apart from none. I was able to take care of this by initiatlizing a list and iterating through the array to append lowercase words to the list since the actual names started with a capital letter.

Columns doggo, floofer, puppo and pupper are columns that represent stages of a dog and they are in separate columns. This columns needs to be merged into one. To do this I replaced, the None with empty strings then add the text in each column together after which I manual add a comma in between ones that had two values.

I also noticed some columns where duplicates of each other. Every column ending in `id_str` were essentially duplicates of a column ending in `id`. I also dropped this columns using `.drop(cols_to_drop,1, inplace=True).`

I also checked for missing values using `df.isna().any().any()` which was written into a function that I can reuse since I would be checking multiple times. There were missing values in the extracted_df and enhanced_df. Since most of the columns contained lots of missing values. I decided to drop them all leaving only `expanded_urls` from enhanced_df and possibly_sensitive and possibly_sensitive_appealable from the extracted_df which I filled with None and 0.0. I filled using `.fillna()`

I also noticed the a column called source in the extracted_df and enhanced_df were filled with div tags instead of the actual source, I extracted the actual source by applying a lambda function that splits the text any of these signs, `>`, `<` and `-`. The first two sign where to get the words in between the div tag while the last was to eliminate the description after one of the source (Vine).

The newline special character was confirmed by a function I wrote to detect it. The were simple removed by replacing them with an empty string, “”. This was done using `.str.replace("\n", "")` on columns where they were present.

The number of tweet in each columns were not the same. This mean we would have missing values if the three datasets were joined together. I prevented this by created a set of ids that were present in all datasets then dropping any row that wasn’t present in this set.

All this three dataset contained data related to the same observation unit, a tweet, so we need to merge then into the same dataframe before we can qualify the dataset as tidy. We did renaming columns that contain the same dataset in each of the columns to have a common name then merging on common columns, id, created_at, full_text, source. This was done using the `pd.merge(...,...,how="inner",...)`

Lastly, I dropped of all retweets by getting rows starting with `RT` and dropping them.