wrangle_report

September 5, 2022

0.1 Reporting: wrangle_report

DATA GATHERING Data was gathered in three different formats (csv, tsv, txt), from three sources. The csv file was downloaded and read in using pandas while the tsv file was retrieved using the requests library and written to working directory using os and a context manager. It was then read in with pandas. To read in the txt file, an empty list was created and a context manager was used to open the file. The opened txt file was looped through with a for loop to read in the lines as json objects. The output was added as values to a dictionary of predefined keys (eventual column name). The dictionary was written to a data frame using pandas and dictionary keys as column names.

DATA ASSESSMENT All three data sets were previewed using pd.head() for visually noticeable anomalies and to get an overall grasp of the data sets. Programmatic assessment was done using the standard methods to check for data type conformity, nulls and duplicate entries. After assessment, eight quality issues and two tidiness issues were identified.

DATA CLEANING Copies of the original data frames were made and the identified issues were addressed using the define – code – test framework. The wrong data types were amended with standard methods and missing values were dropped. Non-compliant null entries were replaced with their standard values (np.nan). The unrequired retweet texts were identified by filtering out rows where the retwee_status_id column is not null. The index for these affected rows were assigned to a variable using the pandas .index() method and dropped on the index using .drop(index=index). The ratings were unified into a single column using simple string concatenation and the initial split columns were dropped. The dog breeds were identified and predictions that are not dogs were removed from the data frame. Only the most confident predictions were taken into account. Having dropped all unrequired columns, all three data frames were merged with an inner join to prevent nulls and ensure that all entries have corresponding values across data frames.

```
In [ ]:
```