UDACITY Data Analysis Nanodegree

Project:- Wrangle & Analyze Data from WeRateDogs Twitter, Wrangle Report

Grant Patience, 29th August 2019

The task required of us was to wrangle numerous data sources together pertaining to the WeRateDogs Twitter account. This would involve gathering, assessing and identifying data quality and tidiness issues and clean them.

1. Gather

There were three data sources Data sources in total, each of a distinct format, location and extraction method;

- Tweet Archive,
- Image Predictions,
- Tweet Information

1.1. Tweet Archive

This file contains basic tweet information of 5000+ Tweets from WeRateDogs. The Tweet Archive data was provided via the UDACITY Project portal as `'twitter-archive-enhanced.csv'`. This was downloaded manually and stored locally into a Data subfolder within the local project directory.

From there, the file was imported into Python using `Pandas` and read into a dataframe called `twitter_archive`.

1.2. Image Predictions

Image Prediction file contained predictions of the Dog Breeds for each image in the Tweet. It was stored online at https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv).

To extract the information programmatically, the Python `requests` library was utilised to extract the data via Python using `requests.get(url)` and saved locally to the Data subfolder in the Project directory as a tsv file called `image_predictions.tsv`. This was read back into Python for analysis as a `Pandas` dataframe caled `image_predictions`

1.3. Additional Data via the Twitter API

The third and final data source pertained to any additional data that could be gleaned from Twitter directly, using Twitters API, most pertinant for analysise was Tweet Favourite and Retweet data.

To extract the data, first a Twitter Developer account was created to give the necessary access required for the Twitter API. The Pytyhon library `tweepy` (http://www.tweepy.org/) (http://www.tweepy.org/) was utilised to extract the data directly from Twitter. Data was extracted into a `Pandas` dataframe and then saved locally to a JSON file called `tweet_json.txt` utilising the `JSON` Pandas library in UTF-8 format.

It was then read back into Python as a `Pandas` data frame called `tweet_data`.

2. Assess

To assess the data, several approaches were carried out in Python.

First of all a manual assessment was made on each data frame utilising `.shape` to determine row count, column count and `.columns` for the column names. From this we could see roughly what attributes contained nulls, the uniqueness contained within the attribute, the datatype of each attribute. Some of the attributes were incorrectly assigned by Python, and some of the columns were not readable such as 'p1', 'p2' and 'p3'

A manual visual assessment of the data was carried out using `.sample(10)` so that a snapshot of rows throughout the datasets could be viewed. This identified the usage of columns and certain values such as `source` attribute being quite unreadable, `expanded_url` being used to identify retweets.

Delving deeper `.info()` and `.nunique()` were used to gather information on the datatypes, and number of unique values in each attribute respectively. `.describe()` wa used to programmaticaly assess different statistics of the datasets such as mean, standard deviations and quartile information.

A function `missing_values_table` was written to programmatically assess the number and % of Null values contained within each attribute in a dataset using `isnull().sum()`. No Null values were found in `image_predictions` or `tweet_data`. Some Nulls were found in `twitter_archive` but were related to retweet and reply information `[in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp]`.

Duplicate Tweets were identified using `tweet_id.duplicated()`, which highlighted duplicates in the `image_predictions` dataframe.

A visual assessment of name values was carried out using `.value_counts()`, identifying several issues with names including the value of 'None' occuring 745 times, several erroneous names such as 'a', 'the', 'an' and other lowercase words which looked to have been prorammatically picked up from the verbatim of the tet body.

Contained within the `twitter_archive` were 4 columns pertaining to the 'stage' or 'class' of a dog; code> [doggo, puppo, floofer, pupper] which could be categorised as one column. Note; it was since discovered that potentially could be multiple dogs in an image, and therefore the tweet could be one or many of each type and not one as assumed.

3. Clean

Before cleaning the data, we copied the data to new "clean" versions of the dataframes to work on, independently of the originals.

Erroneous, mistaken, or incorrect dog names were corrected using Python assessments to get a list of the index locations of each affected row, and we reassessed the text body of the Tweet to find a replacement.

Duplicate rows were dropped using `drop_duplicates(inplace=True)`

The uneccesary 'doggo', 'floofer', 'pupper' and 'puppo' columns were merged into one column 'dog_stage' by adding the contents to a new column called `dog_class`.

The three datasets were combined into one using Pandas `merge` into the 'master' dataset `twitter_archive_clean`.

Retweets were removed using `.dropna` on the expanded_urls attribute and unused columns `['retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp']` were dropped using `.drop`.

The `source` attribute was cleaned down to a readable format using string replace function `.str.replace`.

Incorrect datatypes were reassigned to a more appropriate datatype using `.astype` for dog_class (category), timestamp(datetime), in_reply_to_status_id(string), in_reply_to_user_id(string). While innapropriate column names were renamed using `.rename` on the Dog Prediction columns `{'p1':'Breed_Probability1', 'p2':'Breed_Probability2', 'p3':'Breed_Probability3'}`

Incorrectly named dog names 'None' were replaced using `replace('None', np.NaN)`.

The cleaned dataset was saved to the Data subfolder in the Project directory as `'twitter_archive_master.csv'`

In [ ]: