UDACITY NANODEGREE (ALX-T) DATA ANALYTICS

WRANGLE REPORT (PROJECT #2)

We Rate Dogs Twitter Data Analysis

By OLAMILEKAN OLUGBAYILA

INTRODUCTION

This project's objective was to aggregate WeRateDogs Twitter data in order to produce insightful and reliable analyses and visualizations. This project, which is primarily focused on manipulating data from the WeRateDogs Twitter account using Python and is described in a Jupyter Notebook (wrangle act.ipynb), was a component of the data wrangling phase of the Udacity Data Analyst Nanodegree program.

PROJECT DETAILS

Real-world data is rarely perfect. I had to acquire, evaluate, and clean the data using Python and its libraries so that it could be used for analysis and visualization. Only a portion of the dataset's issues (no less than eight quality issues and two tidiness issues) required to be fully analyzed and cleaned because doing so would require a lot of work.

The project tasks included :

- Gathering of Data (Programmatically and via API interaction)
- Assessing the gathered data (To spot quality and tidiness issues)
- Cleaning the assessed data and resolving the quality issues.
- Storing the cleaned data
- Generating insights and visualizations
- Creating a comprehensive report to detail the workflow

DATA GATHERING PROCESS

The three different formats in which the data for this project were obtained are listed below:

1) **WeRateDogs Twitter Archive File**: Udacity programmatically extracted this and made Twitter archive enhanced.csv available for usage.
2) **Image Predictions File**: According to a neural network, each tweet's image predicts the breed of dog that is present. The URL https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad image-predictions/image-predictions.tsv was used to programmatically download the file (image predictions.tsv), which was hosted on Udacity's servers.
3) **Tweet JSON File & Twitter API**: I used Python's tweepy module to query the Twitter API for each tweet's JSON data using the tweet IDs from the WeRateDogs Twitter archive, and I saved each tweet's whole set of JSON data in a file called tweet json.txt.

ASSESSING THE DATA

I assessed the data using visual and programmatic means. The majority of the quality concerns that were present in the three datasets were actually revealed to me by programmatic assessment. Then I divided the problems into two groups: quality and orderliness. I also summarized my evaluation in the Jupyter notebook. I split the quality concerns into groups based on the datasets and performed consistency, correctness, validity, and completeness checks.

CLEANING THE DATA

The three data sets that made up this portion of the data wrangling were further broken into the three processes of define, code, and test. To make things simple to grasp, I've included a headline with code and test blocks in the main jupyter note attached in the submission.

The first and most crucial step was to make duplicates of each of the three data frames. So that rather than using the original frames, I can experiment with the copy frames.

Furthermore, I cleaned each data frame individually to fix the issues.

In the **Twitter Archive Dataset**: The following issues were cleaned

- HTTP link attached to the end of the string in the text column
- Tweet_id datatype recorded as int instead of string
- Timestamp column has a datatype of string instead of DateTime
- Null values represented as none instead of Nan
- Invalid names in dog name columns. ('a','an', none etc)
- Denominator values for dog ratings are greater than the standard 10
- Duplicates are found in the expanded_url column.
- No interests in columns like (retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp)
- No interests in columns (in_reply_to_status_id and in_reply_to_user_id).
- Source column in twitter archive data frame values messy change datatype to category
- The Columns (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id and tweet_id) dataypes are float instead of objects.

In the **Additional Twitter dataset:**

The most crucial and necessary columns in the generated tweet JSON data were retweet count and favourite count; other columns were essentially redundant because they were available in the Twitter archive data. Therefore, the extra columns were eliminated.

Finally; the **tidiness issues**: The following issues were fixed.

- dog features (doggo, floofer, pupper, puppo) are spread in different columns
- Breed Predictions, Confidence intervals and Dog tests are spread in three columns
- All datasets should be merged into one data frame

CONCLUSION

I discovered that there was no need for three data sets after cleansing the data. A single file containing all the data might be created with ease. To build the Twitter archive master.csv, I linked the “additional_twitter_df" and "image_predictions_df" together using Tweet_id to join them.