We Rate Dogs Project - Wrangle and Analyze Data

Sources:

I gathered data from the following source:

1. Given CSV file (imported)
2. HTML URL programmatically
3. Twitter's API using tweepy to access the API to gather the JSON data for the tweets, this data is stored as txt file and then wrangled using pandas and other libraries

Mentioned below are all the steps followed to get the results:

1. Gathering:

Gathering Data for this Project involved obtaining three different datasets from three different sources. Each one testing a different way of obtaining a dataset.

The 3 files had untidy and subpar quality level, so they needed wrangling to reach the needed status

They are originally named:

- twitter_archive_df (CSV File)
- tweets_df (From API)
- img_predictions_df (From HTML Link)

At the end, the API keys are then removed by erasing the keys from the file as it is the shortest and most functional method

2. Assessing:

Several aspects are then loaded to the jupyter notebook and checked visually and programmatically using methods such as (.info(), .value_counts(), .head(), .tail(), … and others)

The data itself is checked as well using excel to check if any errors occurred in the csv

The datasets were accessed under two criteria, quality and tidiness. When an issue was detected it was documented under one of these two criteria

The goal was to proceed to the next step understanding the needed and prepare them for cleaning

Cleaning:

The final step is to clean the data here are some of the issues that were addressed:

In Tidiness:

1. Combine data frames
2. Unify multiple columns mentioning dogs (doggo, floofer, pupper and puppo)
3. Use best prediction in regard to images prediction

In Quality:

1. Remove retweets and meme replies
2. Remove unwanted and extra columns
3. Fix data types
4. Fix timestamp data types
5. Correct Numerators
6. Correct Denominators
7. Fix some generic dogs names
8. Fix column names with better indicators

Some extensive methods were used especially in the ratings and breeds

In Ratings: the following regex '((?:\d+\.)?\d+)\/(\d+)' was used to re-import the ratings and the values were then re-assed and further modified

The maximum rating was normalized as 10 and anything in the numerator greater than 10 was converted to 10

In Breeds: It was modified to accept more than one entry, such as:

```
Out[49]:                    1662
         pupper              201
         doggo                62
         puppo                22
         doggo, pupper         8
         floofer               7
         doggo, floofer        1
         doggo, puppo          1
         Name: stage, dtype: int64
```

None and empty value are converted to empty space

4. Storing:

The data is then stored to twitter_archive_master.csv file where all the clean data resides waiting to be conquered

It is worth mentioning that not all the data are corrected

As so many more aspects can be tackled but this what I ultimately decided to include in my pack and go on this adventure

Overall this project was a blast and I was so happy to work on it, hope my project gets your liking

Thanks and Regards

Hossam Tarek