wrangle_report

September 15, 2020

Data Wrangling and Analysis

A case study by Siddartha Thentu

Introduction

What is this project about - In this project, we look at ways to gather data, supplement the incomplete data by wrangling the required data from the internet thorugh API, manually download available data files. After that, we assess the data for issues in it's quality and tidiness. Then, we clean the issues and save the clean file.

Libraries Used - pandas : data manipulation - numpy : numerical manipulation - requests : downloading data from the internet - re : extracting the regular expressions - tweepy : downloading data through twitter API - json : handling json format data - timeit : calculate the time taken

Steps Taken - Data gathering - Data assesing - Data cleaning - Data analysis

Case Study WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage. In this case study, we currently have **two pieces of data/dataframes available with us**. Let's take a minute to explore them.

1. Through Udacity, we have access to a few tweet data provided by WeRateDogs. However, this data has only basic features like 'tweet_id','dog_name','retweet_id' etc. Some important features like the number of retweets and number of favorites (which show the popularity of the dog) were not provided. Luckily, with the given tweet_id's, we can wrangle this tweets from WeRateDog's archive and extract the required content. **Name : twitter-archive-enhanced-2.csv Source : Udacity Access : manual download from website**

2. Each tweet in the twitter-archive-enhanced-2.csv is thought to have images of the dog. These images were fed into a Dog breed classifier model to identify the type and breed of dog. This was done by the Udacity team and the file was hosted on Udacity's server. **Name : image-predictions.tsv Source : Udacity Access : Download the file from server through requests library**

Gathering

- For gathering data, I created a developer's account through Twitter and registerd my OAuth.
- Used tweepy to extract the relevant tweet content
- Manually downloaded given files and loaded them into dataframes through pd.read_csv()
- Downloaded image_prediction.tsv file from the internet through requests library

- Create three seprate dataframes for three pieces of data

Assessment

For data assessment, I utilized two types both visual and programmatic assesment. Some of the most import functions used were - pandas.dataFrame.info() for identifying the data types - pandas.dataFrame.describe() for identifying the min and max values - pandas.value_counts() to count the number of occurences - pandas.duplicated() to find the duplicate values - pd.Series.query() helped me to extract the part of dataframes that I need. I - Indexing the dataframe was done with pd.loc and pd.iloc.

Quality Issues

1. given_tweet_data dataframe
   - Impossible dog names should be replaced with Nan
   - tweet_id datatype erroneously points to Int. Should be changed to object
   - timestamp datatype incorrect. Change to datetime
   - Missing values in certain columns like in_reply..etc
   - Erroneous numerator and denominator values should be corrected
   - Retweet tweets should be dropped
   - Source column has html text to be removed
2. image_df dataframe
   - tweet_id dataype correction
   - duplicate jpg_urls due to retweets
   - rows whose images are not classified as a dog
   - Floofer and etc columns have string None instead of python Nan
   - names of dogs have inconsistencies. Lower and Upper case, joining words with ‘-’ and ’_’
3. rt_df dataframe
   - retweet_count column datatype correction
   - fav_count column datatype correction

Tidiness Issues

1. given_tweet_data dataframe
   - Unused columns can be dropped
   - multiple dog class columns can be collapsed into 1 column
   - Merge 3 dataframes into 1 based on tweet_id
2. image_df dataframe
   - 3 dog breed probabilities can be replaced with the top most probability
   - 3 probability confidences can be replaced by the top most accurate

Data Cleaning

The most important part befor data cleaning is to create copies of dataframe which I did. - pd.DataFrame.copy() to copy dataframes - lambda functions to manipulate columns - pd.dropna() to drop rows with Nan values - pd.drop() to drop columns - pd.Series.str.extract() to identify a specific patter in the string - pd.Series.replace() to replace contents of a series - Merged tables using pandas.merge()

Steps taken :

- **Create copies of dataframes**
- **Drop rows in predict_df/image_df dataframe that does not have images predicted as a dog**
- **Change predict_df/image_df tweet_id column datatype from int to string**
- **Handle the inconsistencies in predict_df/image_df dog breed names by replacing - and _ with space. Also changed all names to lowercase**
- **Replace multiple p1,p2,p3 columns into 2 columns : breed and accuracy**
- **Changing data types of columns tweet_id and timestamp**
- **Merge doggo, floofler, pupper and puppo columns into one**
- **Replaced “None” values in dog_class with space and later into np.nan
- **Rectify the dog_class by checking the text by concatenating the columns and correcting inconsistencies**
- **Delte rows with tweet_id’s that were retweeted by checking retweeted_status columns**
- **Drop unwanted columns like doggo,floofer,in_reply_to_status_id etc..**
- **Correct dog names by replacing unnecessary columns with np.nan**
- **Correct source column by removing unnecessary html text**
- **Correct multiple patterns in ratings by manually studying the tweet.text**
- **Correct floating numerator in ratings by manually studying the tweet.text**

[ ]:

Conclusion:

I learned a lot through this data analysis project. I realized that the pipeline for a data analysis process and the steps needed to succesfully navigate the pipeline step by step. I am well equipped with pandas and matplotlib skills now. I also learned how to wrangle missing data from the internet and systematically assess and clean the data