Data Wrangling & Data Modeling

UDACITY

February 11, 2023
Logan Scott McQuillan

Project Overview

Introduction

A dataset of over 5000 tweets from the popular Twitter site called WeRateDogs (twitter.com/dog_rates) will be wrangled, analyzed, and visualized to extract data on our most precious family members, our pets. The tweets feature humorous comments about people's beloved pets with a score out of ten, but most visitors leave a score better than ten because of how adorable the pets are.

In order to analyze the given dataset, information from multiple sources must be first gathered and cleansed. The goal is to use the basic tweet information and extrapolate data to create awe-inspiring analyses and visualizations.

Project steps

The goal of this project is to:

- Gather data
- Assess data
- Clean data
- Store data
- Analyze & visualize data
- Report data findings

Two sweet golden retriever puppies playing in the grass.

Methods used

The project will be performed using Jupyter Notebook with Pandas, Numpy, Requests, Tweepy, and Json libraries. Other libraries which add to the functionality of the project are additionally used, such as Seaborn, Matplotlib, WordCloud, PIL, and IPython using HTML.

The Data

Three datasets will be used to obtain the necessary information to conduct the analysis. First, the WeRateDogs Twitter archive (twitter_archive_enhanced.csv) containing over 5000 tweets and provides the attributes needed for analysis: Tweet id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls, rating numerator, rating_denominator, name, doggo, floofer, pupper, and puppo attributes. The twitter_archive_enhanced.csv file is already provided in the file folders.

The next dataset is the image_predictions.tsv. Using the Requests library at the URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv, information about the dog breed predictions will be uploaded with the attributes needed for analysis: Tweet id, jpg_url, img_num, p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf, and p3_dog.

The final dataset used is the tweet_json.txt, obtained from the Twitter API, using the Tweepy library. The json text file will be saved as a Pandas DataFrame with the attributes of tweet id, retweet count, and favorite count.

Gather Data

The WeRateDogs Twitter data will be directly downloaded via the file that Udacity provided. It will be saved as df_enhanced for later analysis and visualizations.

Using the Requests library, the data for the dog breed image predictions (image-predictions.tsv) will be downloaded for future analysis and visualizations.

The Tweepy library using the Twitter API is used to extrapolate the tweet_json.txt information.

Assess Data

The project requires eight quality issues and at least two tidiness issues to assess and clean. Data quality issues are problems that diminish the credibility and reliability of data in a dataset, such as incorrectly entered data, missing or duplicated data, etc. Tidiness issues are problems that affect the organization of the dataset format, such as a column containing a variable and a row containing multiple pieces of information.

The quality issues found during the assessment:

1. In the twitter archive enhanced data, the timestamp column is set to an object type. It needs to be changed to a timestamp datatype.
2. In the twitter archive enhanced data, the tweet_id column is set to an int64 datatype and needs to be set as an object or string datatype.
3. There are several invalid dog names in the twitter archive enhanced data that need to be removed.
4. In the twitter archive enhanced data, the retweeted rows and replies must be removed in order to remove the unwanted columns.
5. In the twitter archive enhanced data, there are several columns missing a lot of data and need to be dropped (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp).

6. The rating_denominator column contains 23 denominators larger than 10, so the numbers need to be replaced with 10 to make them consistent.
7. In the image predictions data, columns p1, p2, and p3 are inconsistent. Some of the dog names are capitalized, whereas others are lowercase.
8. In the image predictions data, the tweet_id column's datatype needs to be changed from int64 to object or string to match the other tables.
9. In the tweet json data, the id column needs to be changed to tweet_id and converted from an int64 datatype to an object or string data type to match the other tables' id columns.

The tidiness issues found during the assessment were:

1. The dog stages need to be combined into one column, not a separate column for each stage.
2. The three separate tables need to be merged into a single table to allow for easier data analysis.

The data assessment requires both visual assessment and programmatic assessment, whereby the former is visually inspecting the data for any issues, and the latter uses pandas to programmatically assess the data.

Clean Data

In the cleaning phase of the project, copies of the original datasets are made to keep the integrity of the original data. After the copies are made, the quality issues are fixed.

Issue number one is to change the timestamp column from an object, or a string, to a timestamp datatype. The code used to transform the datatype is:

```
df_enhanced_cleaned['timestamp'] =
pd.to_datetime(df_enhanced_cleaned['timestamp'])
```

Issue number two is to correct the datatype for the tweet_id column in the twitter archive enhanced data. The code to correct problem:

```
df_enhanced_cleaned.tweet_id = df_enhanced_cleaned.tweet_id.astype(str)
```

Issue number three is to remove any invalid dog names, such as none, lowercase, or blank values. The code to fix the invalid dog names issues:

```
df_enhanced_cleaned.loc[df_enhanced_cleaned.name.str.islower(),'name']= np.nan
df_enhanced_cleaned.loc[df_enhanced_cleaned.name == 'None','name'] = np.nan
df_enhanced_cleaned.loc[df_enhanced_cleaned.name == '', 'name'] = np.nan
```

Issue number four is to remove the retweeted rows and replies from the Twitter archive enhanced data to delete the unwanted columns later. The code used to delete the retweeted rows and replies:

```
df_enhanced_cleaned =
df_enhanced_cleaned[pd.isnull(df_enhanced_cleaned.retweeted_status_id)]
df_enhanced_cleaned =
df_enhanced_cleaned[pd.isnull(df_enhanced_cleaned.in_reply_to_status_id)]
```

Issue number five drops the unwanted columns from the dataset (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp). A user-defined function is created to drop the columns that are not required for data analysis.

```
def drop_data_func(column_name):
    """returns a drop function to drop selected columns from dataset"""
    return df_enhanced_cleaned.drop([column_name], axis = 1, inplace = True)

drop_data_func('in_reply_to_status_id')
drop_data_func('in_reply_to_user_id')
drop_data_func('retweeted_status_id')
drop_data_func('retweeted_status_user_id')
drop_data_func('retweeted_status_timestamp')
```

Issue number six is to correct the rating_denominator column that contains 23 denominators larger than 10. The code used will change the values from their larger numbers to the number ten:

```
df_enhanced_cleaned.rating_denominator = 10
```

Issue number seven is to correct the capitalization of the dog names in the name column. Some of the dogs' names are capitalized, whereas others are not. The code used to correct the issue:

```
    df_predictions_cleaned.p1 = df_predictions_cleaned.p1.str.lower()
    df_predictions_cleaned.p2 = df_predictions_cleaned.p2.str.lower()
    df_predictions_cleaned.p3 = df_predictions_cleaned.p3.str.lower()
```

Issue number eight is to correct the image predictions data tweet_id column's datatype to an object, or string, to match the other tables' tweet_id datatypes in order to join the tables for future analysis. The code to fix the issue:

```
    df_predictions_cleaned['tweet_id'] =
df_predictions_cleaned['tweet_id'].astype(str)
```

Issue number nine is to fix the tweet json data. The id column needs to match the other tables datatypes. The tweet_id column datatype (int64) will be changed to an object, or string, datatype.

Next are the tidiness issues to fix. The first tidiness issue is the dog stage columns (doggo, floofer, pupper, and puppo) need to be joined together to form one column, not four separate columns to make future analysis and visualizations easier to perform. First the none values need to be replaced, followed by joining the values. After the joining, the empty strings need to be replaced with the NaN values, and then finally, the unwanted columns will be dropped. The code to fix the issues:

```
    df_enhanced_cleaned['doggo'].replace('None', '', inplace = True)
    df_enhanced_cleaned['floofer'].replace('None', '', inplace = True)
    df_enhanced_cleaned['pupper'].replace('None', '', inplace = True)
    df_enhanced_cleaned['puppo'].replace('None', '', inplace = True)
    df_enhanced_cleaned['dog_stage'] = 
df_enhanced_cleaned[df_enhanced_cleaned.columns[8:]].apply(lambda x: 
''.join(x), axis = 1)
    df_enhanced_cleaned['dog_stage'].replace('',np.nan)
    df_enhanced_cleaned.drop(['doggo','floofer','pupper','puppo'], axis = 1,inplace = 
True)
```

Issue number two of the tidiness issues is to create a data object to store the three sperate tables into one master table. First the enhanced table will be joined with the predictions table, and then that joined table will be joined to the tweet json table. The code used to join the three tables is:

```
    df_merge_tables = pd.merge(df_enhanced_cleaned, df_predictions_cleaned, 
on='tweet_id', how='inner').merge(df_tweet_json_cleaned, on='tweet_id', 
how='inner' )
```

Store Data

After the data is fixed and joined, the next phase is to store the joined master table as “Twitter_archive_master.csv.” The code used to save the joined tables is:

```
    df_merge_tables.to_csv('twitter_archive_master.csv', index=False)
```

Conclusion

This project has expounded on what data wrangling is and the steps needed to perform data wrangling.

Data wrangling is the process of gathering, assessing, cleaning, and organizing raw data to make it useful for future analysis and data visualization. The unstructured data is converted into structured data utilizing many different tools, such as Python, Pandas, Numpy, Seaborn, Matplolib, and SciPy, to name a few.