Wrangle_Report

September 1, 2022

0.1 WRANGLE_REPORT

The `wrangle report` shows all the data wrangling process carried out in the `wrangle_act` jupyter notebook as well as the code & comments.

```
In [ ]: #Programmatically opening the tweet-json.txt file
        df_list = []
        with open("tweet-json.txt", "r", encoding = 'utf-8') as file:
            for i in file:
                data = json.loads(i)
                df_list.append(data)

In [ ]: #Loading the dataset
        df = pd.DataFrame(df_list, columns = df_list[0].keys())
        df = df[["id", "favorite_count"]]
        df.head(6)

In [ ]: #Investigating the archive dataset
        df.info()

In [ ]: #Checking for the datatypes of the archive dataset
        df.dtypes

In [ ]: #More investigation on the archive dataset
        df.describe()

In [ ]: #Removing the retweeted_status_id & retweeted_status_user_id, retweeted_status_timestamp
        archive = archive.drop(["retweeted_status_timestamp", "retweeted_status_id", "retweeted_

In [ ]: #Checking to confirm the removal
        archive.info()

In [ ]: #Loading the image predictions dataset as predictions
        predictions = pd.read_csv("image_predictions.tsv", sep = '\t')
        predictions.head(6)

In [ ]: #Investigating the predictions dataset
        predictions.info()
```

```
In [ ]: #Checking the datatypes of the predictions dataset
        predictions.dtypes

In [ ]: #Checking for null values in the expanded column
        archive[archive.expanded_urls.isnull() == True]

In [ ]: #Checking the rating_numerator column for the maximum value
        archive[archive.rating_numerator == archive.rating_numerator.max()]

In [ ]: #checking the rating denominator for the maximum value
        archive[archive.rating_denominator == archive.rating_denominator.max()]

In [ ]: #checking the rating numerator whose value is greater than 20
        archive[archive.rating_numerator > 20]

In [ ]: #using regular expressions to check the numerator values where they have been incorrectl
        import re
        archive[archive.text.str.contains("r(\d+\.\d*\/\d+)")][["text", "rating_numerator"]]

In [ ]: #Check for duplicate values in the dataset
        archive[archive.tweet_id.duplicated()]

In [ ]: #value count of the ima_num column
        predictions.img_num.value_counts()

In [ ]: #checking for duplicated tweet id
        predictions[predictions.tweet_id.duplicated()]

In [ ]: #checking for duplicated jpg_url
        predictions[predictions.jpg_url.duplicated()].jpg_url.values

In [ ]: #filtering duplicated jpg_url in the dataset
        predictions[predictions.jpg_url.apply(lambda x: x in predictions[predictions.jpg_url.dup

In [ ]: #checking for duplicated  values in the twitter json converted dataset
        df[df.duplicated() == True]

In [ ]: #checking for duplicated id's
        df[df.id.duplicated()]

In [ ]: #checking for duplicated retweet count & favourite count
        df.favorite_count.duplicated()
```

After all the processes above was carried out, I highlighted some `Quality issues` **&** `Tidiness Issues` **I noticed about the data as shown below**

0.1.1 Quality issues

`archive table, predictions table, df table :` 1.There are columns with missing values namely - expanded_urls

2.Name, doggo, floofer, pupper and puppo columns have value with the name `None`

3.Values in `source` column are not human readable

4.Use of _instead of `space` in p1, p2 and p3 column values. Also, values upper case sometimes lowercase other times in p1, p2 and p3 columns

5.column names are not clearly descriptive

6.There are image duplicate predictions present for duplicate `jpg_url` with different tweet ids and rest all the data same.

7.tweet id title is different, `id` here `tweet_id` in others.

8.There are `image duplicate predictions` present for duplicate `jpg_url` with different `tweet ids` and rest all the data same

0.1.2 Tidiness issues

1.One variable in 4 columns - dog_stage(doggo, floofer, pupper, puppo)

2.Merge the tables - `archive_tweet` and `df`

Now, all the issues have been cleared. The data is ready to be stored and analysed to get insights.

0.2 Storing Data

Save gathered, assessed, and cleaned master dataset to a CSV file named "twitter_archive_master.csv".

tweet_data_master = pd.merge(archive_clean, predictions_clean, on = 'tweet_id', how = 'inner')

```
In [ ]: tweet_data_master.to_csv('twitter_archive_master.csv')
```

0.2.1 Storing predictions_clean

```
In [ ]: predictions_clean.to_csv('image_predictions_master.csv')
```