Data Wrangle Project:

Waterdogs twitter archive

Data wrangling, which consists of:

1. Gathering data
2. Assessing data
3. Cleaning data

1. Gathering Data we gather data from 3 different resources as follow

- Download file manually
- Download file from the internet
- Access twitter API

- Download file manually

- Download file (twitter_archive_enhanced.csv) and upload it in jupyter environment folder.
- Load it in pandas dataframe using pandas read_csv function then test it using head function

- Download file from the internet

- Use requests package to get the file from the URL provided then save it. - Load it in pandas dataframe using pandas read_csv function then test it using head function

- Query Data from twitter API

- Create Twitter developer account and get the tokens and credentials needed to access the API.
- Connect to twitter API using tweepy package.
- Get the tweets that has tweet_id in the twitter_archive_enhanced file and save it in tweet_json.txt each tweet in new line.
- Read the tweet_json.txt and load tweets in dataframe, we grape only the interesting data, which are 'tweet_id', 'retweet_count', 'favorite_count', and 'followers_count', 'source'.
- we save each tweet data in dictionary then append the dictionary to list then convert the list to the dataframe

2. Assess Data: Assess Data visually and programmatically

1. Assess Data visually by open files using excel program and show them in the jupyter notebook.
2. Assess Data programmatically using functions like:

info () to get main information about datatype and number of null values in each column, head () to get a look at the first rows, sample () to get random rows, shape to get dimension of the dataframe how many rows and columns and other known functions that can be applied to get closer look for the data, you can find all the functions used in wrangle_act.ipynb From the visual and programmatic assessment, I found the following issues:

Quality Issues:

1. Should remove tweets in twitter_df and not in api_df
2. Should convert tweet_id from int64 to object –string to be easily used

- twitter_df

2. Should convert timestamp of type object to be datetime

3. Should convert None value in the columns name, doggo, floofer pupper and puppo column

4. Inaccurate names in name column like ’a’ and ’an’

5. 59 null values for the expanded_urls column

6. Data must contain original tweets no retweets and reply (no tweets has retweeted_status_id,in_reply_to_status_id, in_reply_to_user_id)

7. unnecessary column like source hard to read

8. numerator column has inaccurate data

9. dominator column has inaccurate data

10. numerator column type int64

- image_predictions_df

13. tweet_id of type int64

14. it has 2075 records which need to be reflected in the other df

15. Inconsistent lower case for some of the predicted bread in the predicted column (p1,p2,p3)

Tidiness Issues:

1. In twitter_df there are four columns (doggo, floofer pupper and puppo) which are values related to one variable

2. In image_predictions_df the column names (p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3,p3_conf, p3_dog) not descriptive and need to be merged to three column because each 3 is values to one variable

3. The two dataframes twitter_df and api_df should merged in one dataframe because it is related to one observation

3. Clean Data

Clean Data by using the points found in assess data phase using define, code, test procedure .

You can find the detailed code and description in wrangle_act.ipynb file → before start clean; make clean copy for the three data frame

Quality Clean:
First for twitter_df

1. Should remove tweets in twitter_df and not in api_df

• Drop records in twitter_df and not in api_df because it is missing tweets, delete. Using isin and drop functions

2. tweet_id of type int64 in all three df

- convert tweet_id column type from int64 to object -string-
- use .as type(str) function

3. timestamp of type object

- convert timestamp column datatype from object to datetime
- use pd.to_datetime() function

4. Many None value in the columns name, doggo, floofer pupper and puppo column

- convert "None value for the column name to np.nan
- convert "None" value for doggo, floofer pupper and puppo to ” to be able to merge in other point in tidness
  - use .replace function

5. inaccurate names in name column like ’a’ and ’an’
- Convert the in accurate value name ’a’ and ’an’ by an accurate name extracted from the text column
- apply reg expression to text column to extract the right names

6. 59 null values for the expanded_urls column
- Drop rows that has null value on the column expanded_urls
- use the dropna functions

8. Data must contain original tweets no retweets and reply (no tweets has retweeted_status_id, in_reply_to_status_id, in_reply_to_user_id) • Drop rows that has value in retweeted_status_id, in_reply_to_user_id and in_reply_to_status_id
• use notnull function and invert sign ~

9. Some Columns not needed in the analysis like "in_reply_to_user_id", "retweeted_status_id", "retweeted_status_user_id", "retweeted_status_timestamp"
• Drop column that don’t have any value no after deleted retweet and reply which are ["in_reply_to_status_id", "in_reply_to_user_id", "retweeted_status_id", "retweeted_status_user_id", "retweeted_status_timestamp"]
• use drop function

10. Numerator column of type int64
• convert numerator type to float and re-extract it from the text value as there is some inaccurate data
•use astype ('float') and extract () functions, regular expression

11. Denominator column has inaccurate data

- Correct rating_denominator values (denominator should equal 10)
- Re-calculate numerator column for rows that have Denominator value not equal 10

12. Source column has hard to read values

- Drop source column as we already have same column in api_df which will merge later to this df
- use drop function

Second for image_predictions_df

13. It has 2075 records, which need to reflecte in the other df

- Drop rows from twitter_df with tweet_id that not in image_predictions_df and rows from image_predictions_df that not in twitter_df
- use drop and isin functions and invert sign ~

14. Inconsistent lower case for some of the predicted bread in the predicted column (p1, p2, p3)

- Change values for p1 , p2, p3 column to be capitalized
- use .str.capitalize() function

Tidiness Clean:

1. In twitter_df there are four columns (doggo, floofer pupper and puppo) which are values related to one variable

- Combine the four columns (doggo, floofer pupper and puppo) in one column and named it dog_stage
- use + sign to combine and then .replace to replace empty value with np.nan then drop the four columns
- replace value in dog_stage column that have multiple bread with mixed bread value

2. In image_predictions_df the column names (p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf, p3_dog) not descriptive and need to be merged to three column because each 3 is values to one variable

- Rename the 9 column ('tweet_id', 'jpg_url', 'img_num','propability_1', 'confidence_1', 'dog_1','propability_2', 'confidence_2', 'dog_2','propability_3', 'confidence_3', 'dog_3') to be more descriptive and merge them into 3 columns 'probability', 'confidence', 'dog'
- use wide_to_long function

3. The two dataframes twitter_df and api_df should merged in one data frame because it is related to one observation

- Merge twitter_df_clean and api_df_clean into
- use merge function with inner and on tweet_id

Storing wrangled data

1. Store twitter_clean into twitter_archive_master.csv
2. Store image_predictions_df_clean into image_predictions_master.csv