UDACITY

DATA ANALYST NANODEGREE

PROJECT Number 4

WRANGLING, ASSESSING, CLEANING AND ANALYZING TWEET DATA

Student :
• HASSAINE Abdellah

6 april 2020

1 Introduction

Real-world data rarely comes clean. Using Python and its libraries, we will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling. and this is the purpose of this report, to document our work and bring each step of the followed process.

2 Gathering

In this part we have been asked to gather data from diffrent source:

- download a file manualy which contain The WeRateDogs Twitter archive,
- download a file programmatically which contain the result of the dog breed prediction based on the picture of the dog passing by 3 different algorithm, and
- use the tweeter API to gather more data about tweets in the archive

3 Assessing

In this part we have been asked to asses the data visualy and programmatically using pandas fonction like *head(), info(), summary()* and then extract data quality issue from data and data tidiness issue.

3.1 Quality

archive_df table:

- row that are not an original tweet when ever we have a value in *(retweeted_status_id, in_reply_to_status_id)*.
- the dog stage is available only for (380 dog).
- sevral tweet have a wrong rating_numerator.
- sevral tweet have a wrong rating_denominator.
- sevral tweet have no name on them.
- tweet with id=”840696689258311684” Kevin is a person not a dog
- tweet with id=”832645525019123713” is not a dog
- change the data type of timestamp to *pd.datetime*.
- make dog stage from 4 columns to 1 column.

img_predection_df table:

- get the best bread from the 3 algorithm based on precision of the algorithm.

3.2 Tidiness

- we have several useless columns in all the data frames that need to get ride of.
- in our data we have two entity we have *tweets* and we have *dog* if we separate each entity in a single file, and connect then with the tweet_id we will have a better data schema.

4 Cleaning

In this part, we've been asked to clean each of the issues documented in the previous part.
Define: define how we going to clean the issue step by step, the more detail we put here the more easy our work will be in the coding part.
Code: write the necessary code to clean the issue
Test: check if the data has been cleaned

Note: in addition to the library needed for the project I've used the **NLTK** library and the brown corpus on the library.