WRANGLING

STEPS

1- GATHERING THE DATA

- KNOW WHAT DATA IS NEED TO BE COLLECTED IN WHTAT FORM , AND FROM WHERE TO COLLECT IT

2- Asserting

- Reading the datasets
- EXAMINIG THE DATA AND THE QULITY OR TIDNY ISSUES WITH IT

3- Cleaning

- **Phase one : Data Cleaning**

```
1- Dealing with duplicate

2- Removing Non-exexting Data
```

- **Phase Two : Data Analysis in Python**
- **Phase Three : Visulaization and sharts**

4- OUTPUT AND FINAL DATA AND Visulaization

WRANGLING

PART 1 :: GATHERING THE DATA

IN THIS PROJECT I HAVE TO DEAL WITH THREE DATA FROM THREE $\leqslant$ SEPREATS ORGINS

* First

BY including the provided link of the first tow data and throw request in pytohn i was able to download the tow files and save both in ".csv" file format , and start to wokr on them.

* Second

And to get the tweets from tweeter on the accoutn of "dog_rate" , started by creating tweeter accoutn with more than five repalys on my request to get the approval on the "tweeter Developer accoutn" and examin how to use the tweepy api to get tweets from certin ids that is proviede by the previslu downloaded in step one and save thes tweets with in ".csv" file format to start work on that file

Type *Markdown* and LaTeX: α^2

PART 2 :: Asserting

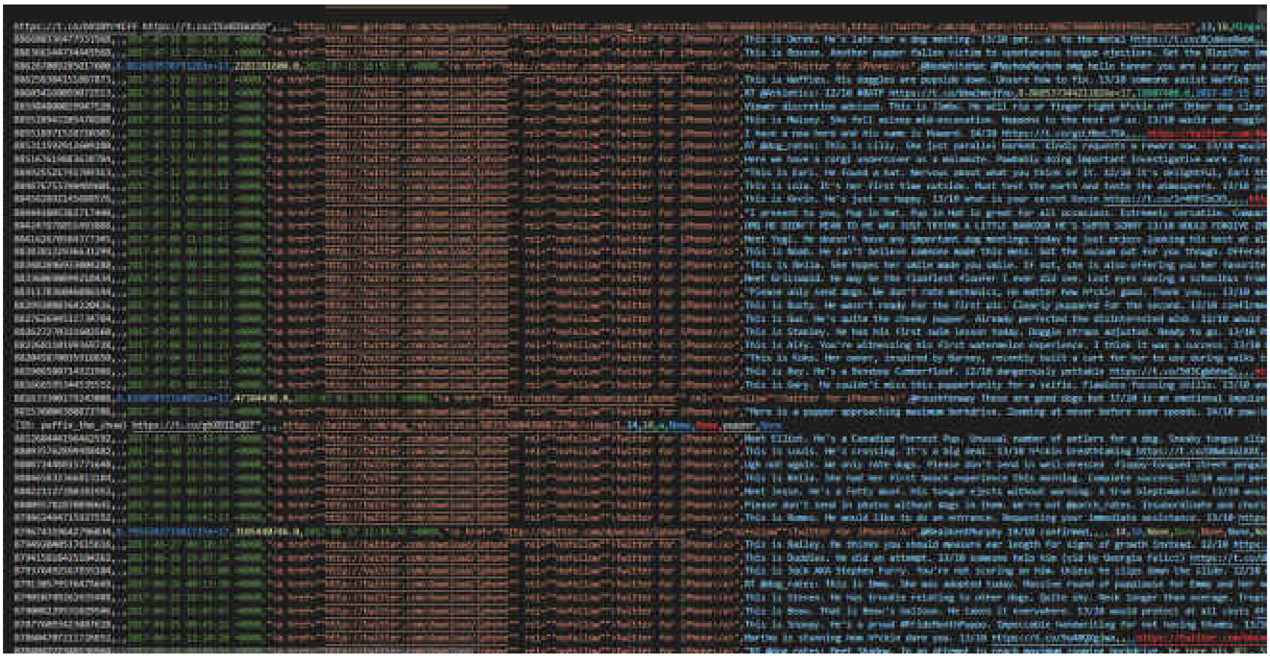
Data Assessment:

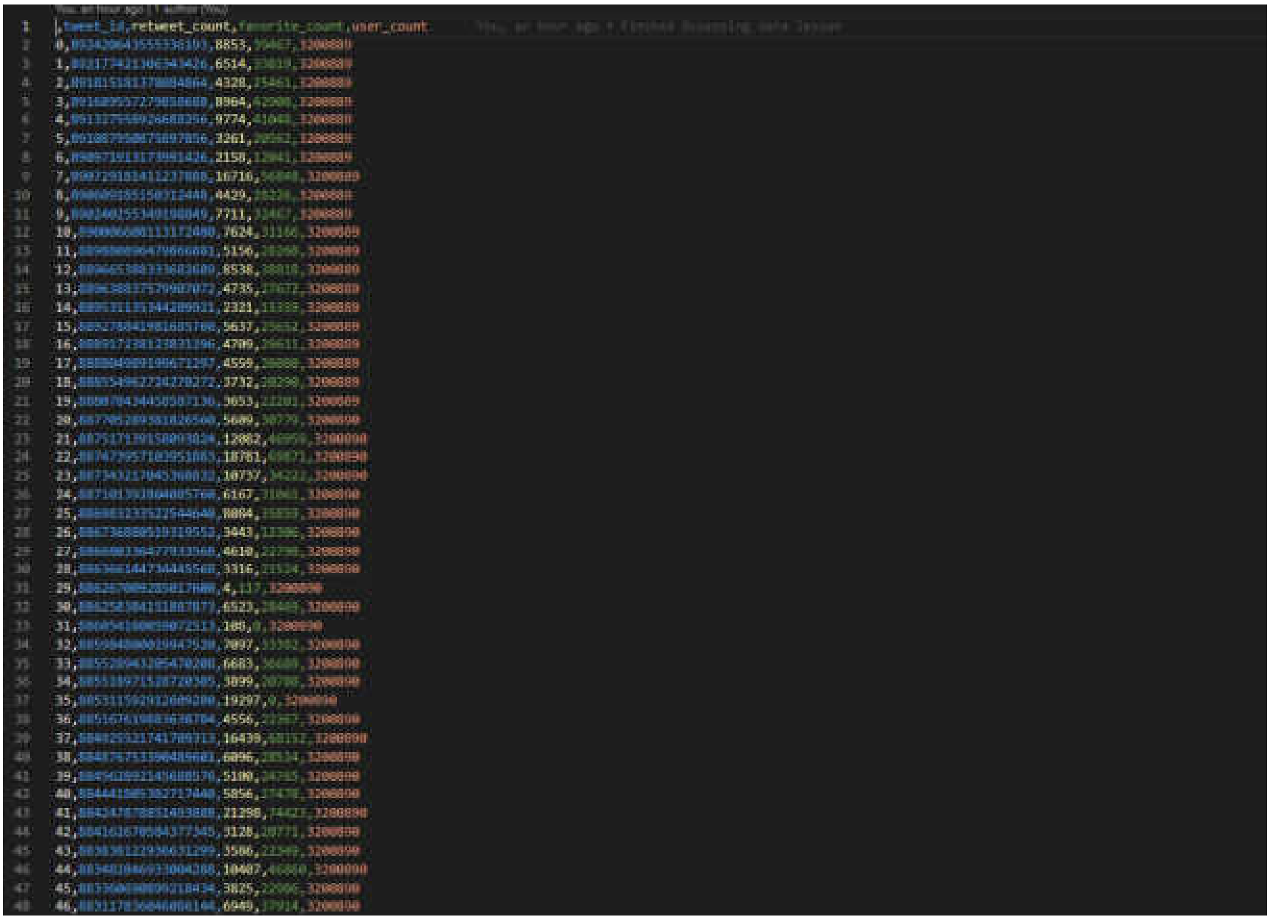
Visualy and Programmatic Examinig the Data

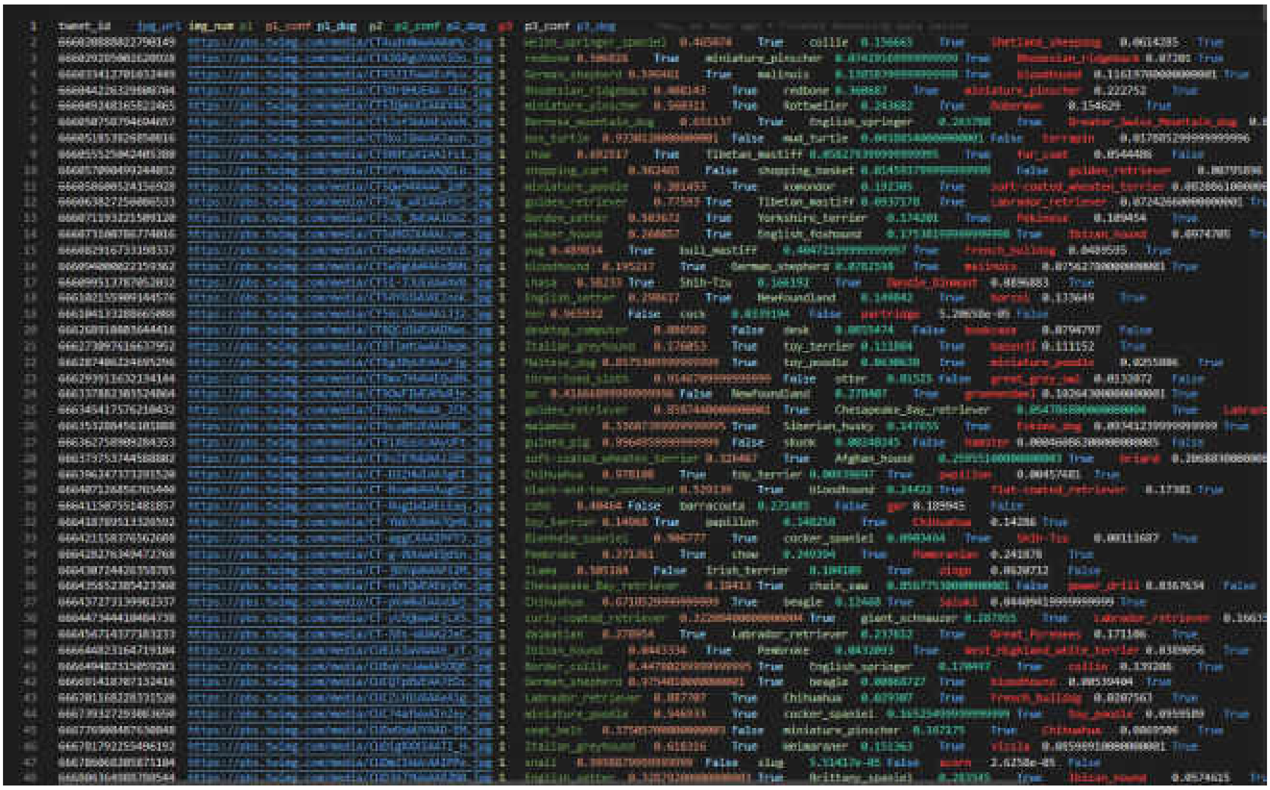
A - Visualy Assessment

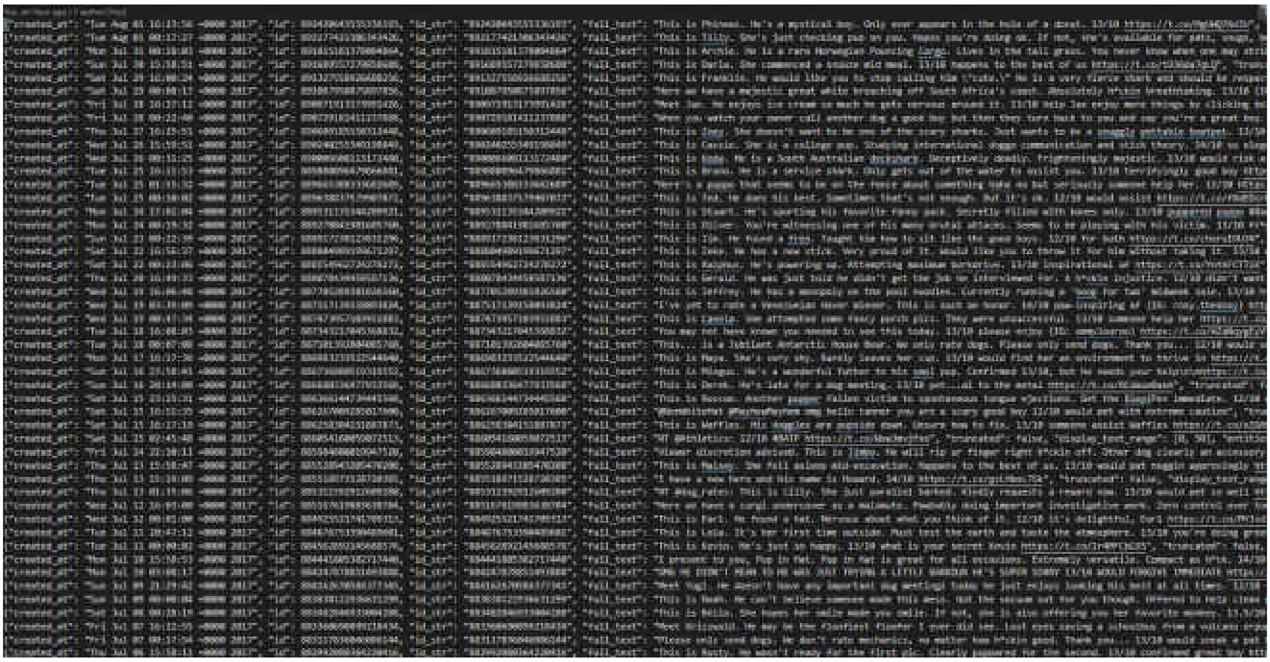
Visual assessment:

- By Openin the FILES ['api_df.csv' , 'archive_df.csv' , 'image_predictions_df.csv'] , In Exel APP and VSCODE APP ON MY WINDOWS DESKTOP AND CHECK THE DATA









B - Programmatic Assessment ::

- BY USING THE pandas supported function to examin the data in the data frames to see how it lock like and what it contain ,

by looping thorw each data to see its shape columns , and what it look like throw describe and info functions :

SOME OF THE PANDAS FUNCTIONS USED IN PROGRAMMATIC ASSESSMENT :

```
PD >> MEANS PANDAS DATA FRAME
```

1- PD.head()

2- PD.shape()

3- PD.info()

4- PD.describe()

5- PD.count()

6- PD.sum()

7- PD.nunique()

Below i provided some photos of the results of using code to visually Programmatic the data given :

```
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Unnamed: 0                  2356 non-null   int64
 1   tweet_id                    2356 non-null   int64
 2   in_reply_to_status_id       78 non-null     float64
 3   in_reply_to_user_id         78 non-null     float64
 4   timestamp                   2356 non-null   object
 5   source                      2356 non-null   object
 6   text                        2356 non-null   object
 7   retweeted_status_id         181 non-null    float64
 8   retweeted_status_user_id    181 non-null    float64
 9   retweeted_status_timestamp  181 non-null    object
 10  expanded_urls               2297 non-null   object
 11  rating_numerator            2356 non-null   int64
 12  rating_denominator          2356 non-null   int64
 13  name                        2356 non-null   object
 14  doggo                       2356 non-null   object
 15  floofer                     2356 non-null   object
 16  pupper                      2356 non-null   object
 17  puppo                       2356 non-null   object
dtypes: float64(4), int64(4), object(10)
memory usage: 331.4+ KB
```

```
max    2355.000000  8.924206e+17          8.862664e+17          8.405479e+17

       retweeted_status_id  retweeted_status_user_id  rating_numerator  \
count         1.810000e+02              1.810000e+02       2356.000000
mean          7.720400e+17              1.241698e+16         13.126486
std           6.236928e+16              9.599254e+16         45.876648
min           6.661041e+17              7.832140e+05          0.000000
25%           7.186315e+17              4.196984e+09         10.000000
50%           7.804657e+17              4.196984e+09         11.000000
75%           8.203146e+17              4.196984e+09         12.000000
max           8.874740e+17              7.874618e+17       1776.000000

       rating_denominator
count         2356.000000
mean            10.455433
std              6.745237
min              0.000000
25%             10.000000
50%             10.000000
75%             10.000000
max            170.000000
********************

Count of actual data in each column against the total number of rows:
```

```
Count of actual data in each column against the total number of rows:

Unnamed: 0                    2356
tweet_id                      2356
in_reply_to_status_id           78
in_reply_to_user_id             78
timestamp                     2356
source                        2356
text                          2356
retweeted_status_id            181
retweeted_status_user_id       181
retweeted_status_timestamp     181
expanded_urls                 2297
rating_numerator              2356
rating_denominator            2356
name                          2356
doggo                         2356
floofer                       2356
pupper                        2356
puppo                         2356
dtype: int64
********************
```

```
Non existing  data in each row:

Unnamed: 0                       0
tweet_id                         0
in_reply_to_status_id         2278
in_reply_to_user_id           2278
timestamp                        0
source                           0
text                             0
retweeted_status_id           2175
retweeted_status_user_id      2175
retweeted_status_timestamp    2175
expanded_urls                   59
rating_numerator                 0
rating_denominator               0
name                             0
doggo                            0
floofer                          0
pupper                           0
puppo                            0
dtype: int64
********************
Duplicated data :
```

```
********************
Shape of the data:
(2075, 13)
********************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  2075 non-null   int64  
 1   tweet_id    2075 non-null   int64  
 2   jpg_url     2075 non-null   object 
 3   img_num     2075 non-null   int64  
 4   p1          2075 non-null   object 
 5   p1_conf     2075 non-null   float64
 6   p1_dog      2075 non-null   bool   
 7   p2          2075 non-null   object 
 8   p2_conf     2075 non-null   float64
 9   p2_dog      2075 non-null   bool   
 10  p3          2075 non-null   object 
 11  p3_conf     2075 non-null   float64
 12  p3_dog      2075 non-null   bool   
dtypes: bool(3), float64(3), int64(3), object(4)
```

C - Asserting issues with Data

Quality and tidiness issues

Quality

1- Missing values :

A - Missing vlues in "twitter-archive-enhanced" Data set, , AS AN EXAMPLE - in :

- "in_reply_to_status_id"
- "in_reply_to_user_id"
- "retweeted_status_id"
- "retweeted_status_user_id"
- 'name'
- "source"

columns .

B - Missing values in "twitter-archive-enhanced" , AS AN EXAMPLE - in :

```
* Row # "2351"  with tweet id     "666049248165822465"  , have name with "None" vlaue .
* Row # "2352"  with tweet id    "666044226329800704"  , have name with "a" letter vlaue .
* Row # "2353"  with tweet id    "666033412701032449"  , have name with "a" letter vlaue .
* Row # "2354"  with tweet id    "666029285002620928"  , have name with "a" letter vlaue .
* Row # "2355"  with tweet id    "666020888022790149" , have name with "None" vlaue .
```

ROWS .

2- Duplicated Tweets in "twitter-archive-enhanced" Data in

```
"retweeted_status_id"        ,with count of    181
"retweeted_status_user_id"   ,with count of    181
"retweeted_status_timestamp" ,with count of    181
```

columns.

3- DATA TYPES

- IN "twitter-archive-enhanced", THE "retweeted_status_timestamp" COLUMN DATA TYPE IS 'object' NOT "TIME STAMP" OR "DATE TIME " TYPE.
- IN "twitter-archive-enhanced" , "tweet_id" column is "int" type with no need to do analysis on it so it may convert to string .

4- DATA VALUES

- Validity:

```
IN "twitter-archive-enhanced" there is :

* "rating_denominator" HAVE DATA VLUES LESS THAN 10 WICH IS THE MINMIMUM RATING FOR DOG , with count of 3 rat
ings

* "rating_numerator"   HAVE DATA VLUES LESS THAN 10 WICH IS THE MINMIMUM RATING FOR DOG , with count of 440 r
atings
```

5 - More than one column in "image_predictions" , for the smae vlaue and messure in AKA (DOGS IS seperated IN 4 COLUMNS)

```
[p1, p2 ,p3]

[ p1_dog , p2_dog , p3_dog ]

[ p1_conf , p2_conf , p3_conf ]
```

6 - In "image_predictions" , columns header are vales not variables name

---

Tidiness

1 - All the data is in relation in each other but seperated into three taples

2 - COLUMN with no useful , or repated data , must be droped from the final data

Type *Markdown* and LaTeX: α^2

PART 3 :: Cleaning

- **Phase one : Data Cleaning**

```
1- Dealing with duplicate

2- Removing Non-exexting Data
```

- **Phase Two : Data Analysis in Python**
- **Phase Three : Visulaization and sharts**

Data Cleaning:

- AT THE END OF THE process we need data that contain the "dog name" if avilable, the "dog type" AKA "dog type" if avilable, and "rating tweets" , that dog type get over the time .

In Cleaning ethier dorp the repeated columns the give the same answer or merge them each catigory in one column

in order to do so I need a data with only "tweet id" that is not duplicated , and original tweet no retweets and also a rating with the most predection avilable,

- To do so i wll do the follow process :

1- merge the types of the dog in one column "dog type" contain the type rather than (Fulse , Ture) boolean column

2- will dorp the other tow less predection , columns , drop text name after extraxting the dog type from the txt column

3- drop the ratings that viloate the rating mesures

4- creating a cealn data and save it to new ".csv" data type .

Type *Markdown* and LaTeX: α^2

WRANGLING

THE END RESULT

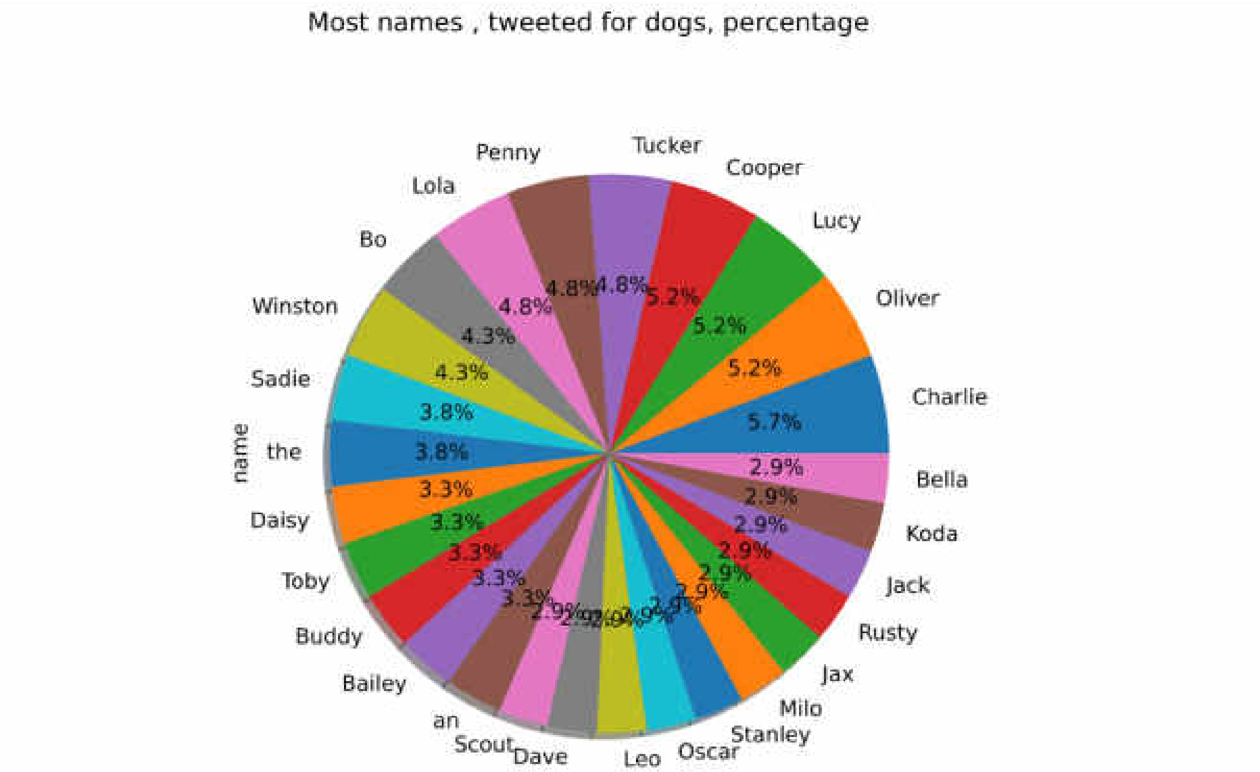
Output:

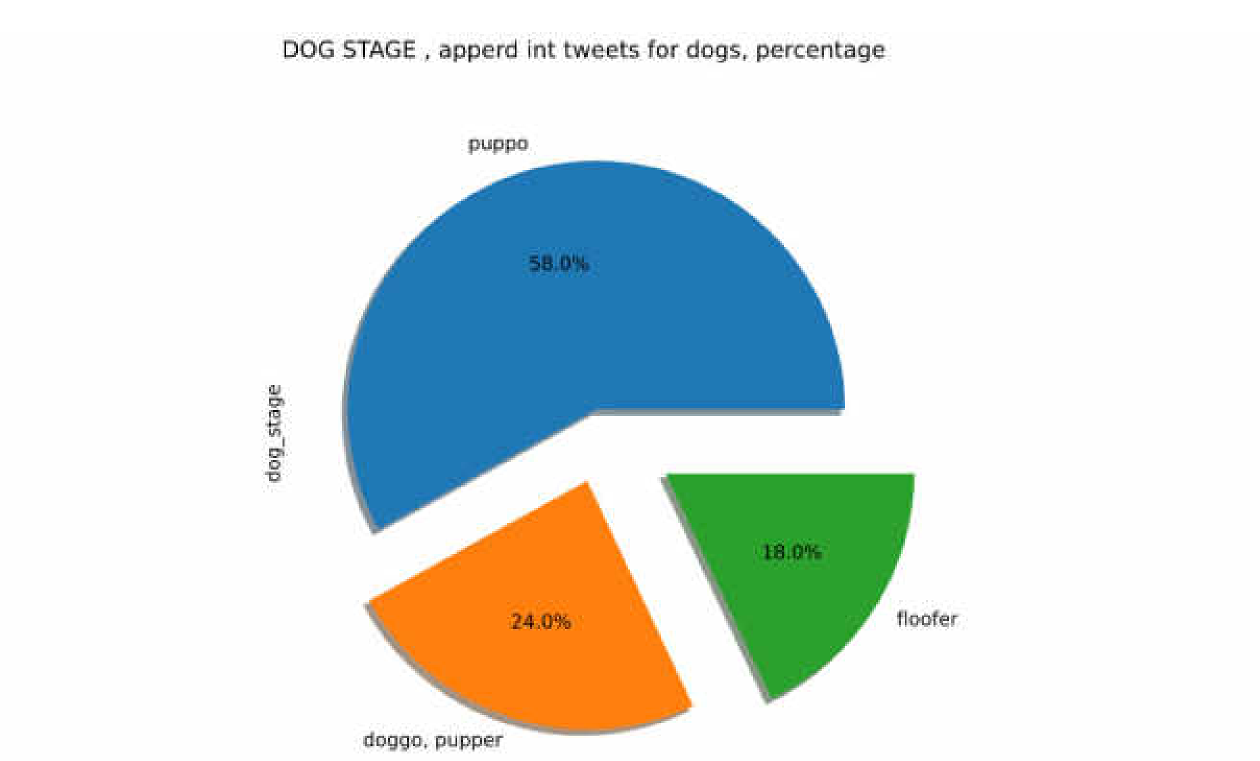
The file created as result of this process.

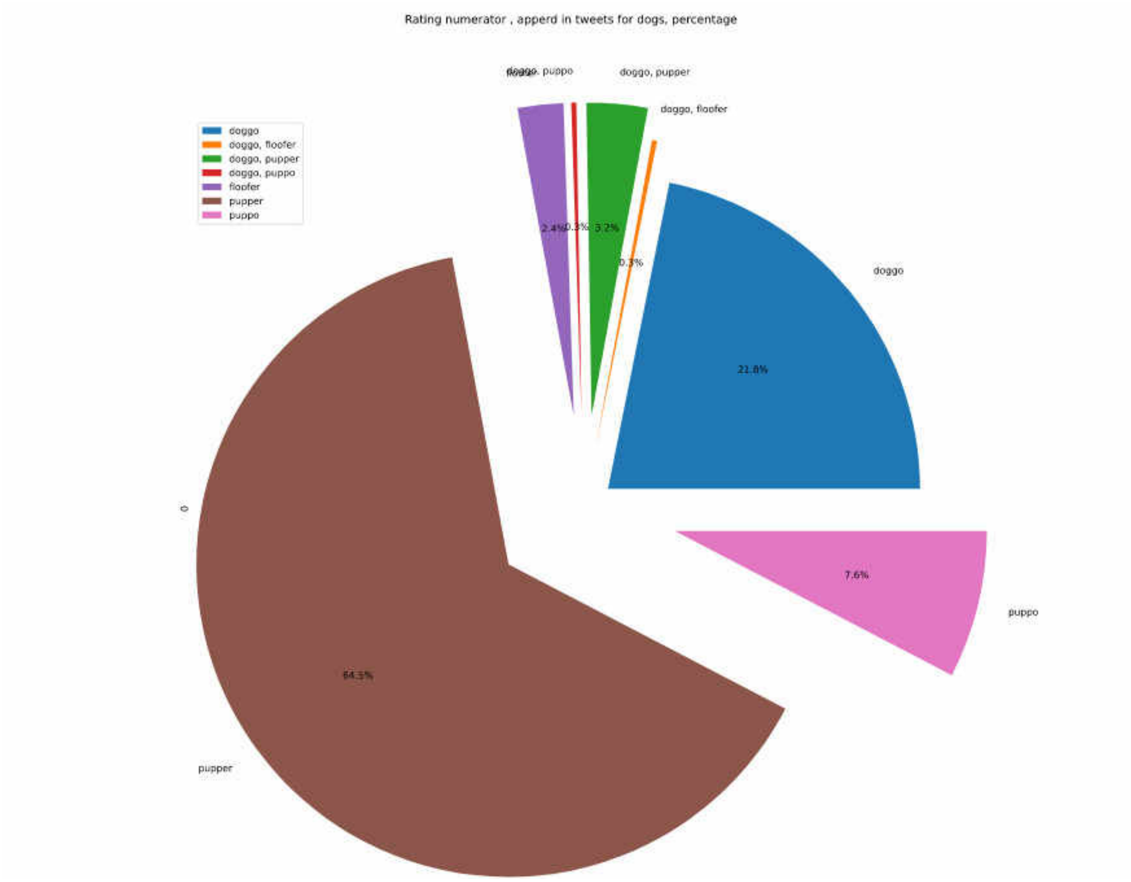
THE OUTPUT FINAL FILE IS "twitter_archive_master.csv" DATA WICH IS HAVE IN MY VIEW THE OPTIMAL DATA CLEANED

WICH WILL BE USED IN THE NEXT PHASES :

- **Phase Two : Data Analysis in Python**
- **Phase Three : Visulaization and sharts**






Rating numerator , apperd in tweets for dogs, percentage
doggo
doggo, floofer
doggo, pupper
doggo, puppo
floofer
pupper
puppo
2.4%
0.3%
3.2%
0.3%
21.8%
7.6%
64.5%
0

In [ ]:

```
1
```