Communication and generation language: We don’t even speak the same language.

**Description:** Language one of the main tools we used as human beings to communicate, without language we will not be able to explain our needs feeling or even ours though.

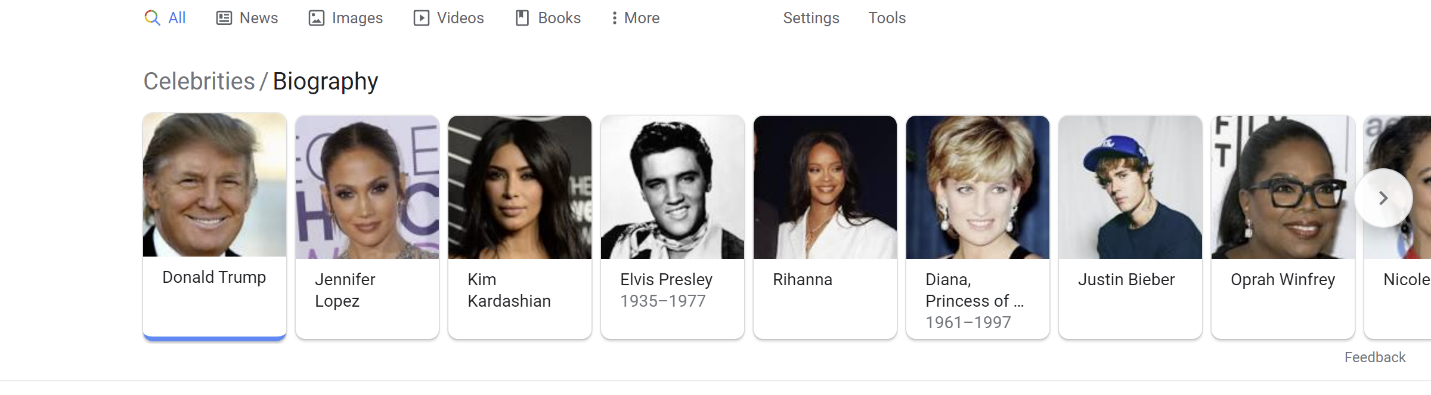
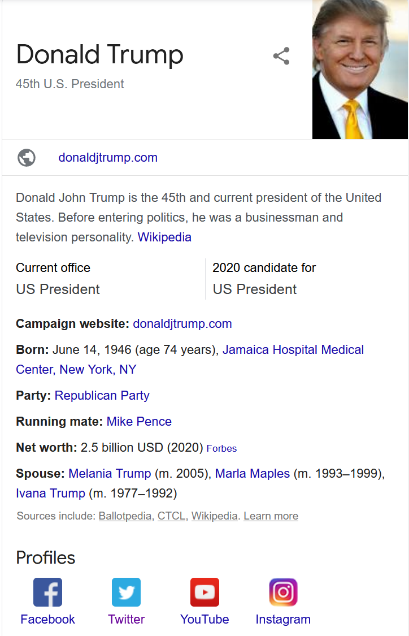
There are many languages in this world, but we still be able to communicate, English is ranked as first language in the world over 1.268 billion speak English, 369.7 million people speaks English as a first language while 898.4 million English speakers English is their second language.

It doesn’t mean we speak the same language we will be able to understand each other well, there is some difficulty you will face like if this different generation or different education or where you come from.

This study will focus in different generations, we will deep be looking at what is the difference between different generation by analyze the words each generation say and try to guess the age from the language or finding the best words we can use when we try to communicate with the other generation.

1. Data collection
2. Link data
3. Data analysis and discover
4. Data cleansing
5. Data preparation, model and testing

**Data collection:**

First part of the project was data collection, I did chose twitter over facebook or other social media for flexibility, the only issue I found in twitter API that they don’t share anything about the age or even the date of birth so I used Google to collect some celebrities profile with them twitter account and them date of birth, also I found more details it might be useful in other project or if decide to expand in this project like relative, place of birth, position, career …

For getting random profile I was just search by jobs or key word famous people.

After that from the list in the top I loop in each person to extract the information from his/her card.

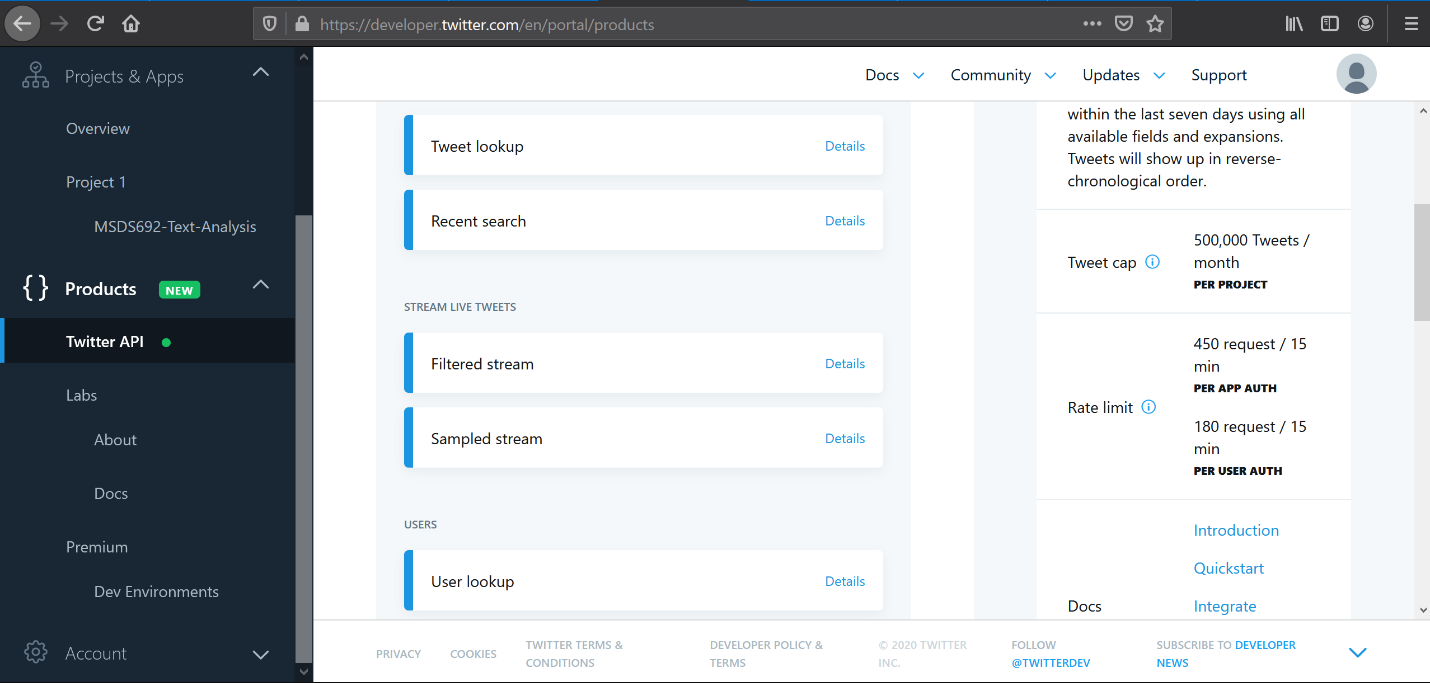
And store all the data in csv file after the data scraping.

Also create twitter API to call twitter profile I found in them bio, I did filter after word the users without Twitter account or when I found his/her account not work or not been use or private.

**Files:**

|  |  |
| --- | --- |
| [scrapingGoogleSearch.ipynb](http://localhost:8888/notebooks/Desktop/textAnalysisProject/scrapingGoogleSearch.ipynb) | Code for data scraping the bio information from google search. |
| Celebrities bio.csv | Csv file for the bio data after scrapping and formatted |
| [TwitterAPI.ipynb](http://localhost:8888/notebooks/Desktop/textAnalysisProject/TwitterAPI.ipynb) | Code to connect with twitter API and retrieve the tweets by twitter user name |

**Link data:**

In this stage I linked the user profile using his google bio to twitter account by looping over google profiling collected from scraping and use the twitter account in the bio information to get most his tweets from his/her account, the only challenge here I takes long time since twitter strict them API call with number of calls before go sleep for 15 min. to be able to place any new call.

The data been stored in csv file to save time from calling the API my account authentication is included in the code to be able to use it for testing.

**Files:**

|  |  |
| --- | --- |
| [extractUserProfileAndTweets.ipynb](http://localhost:8888/notebooks/Desktop/textAnalysisProject/extractUserProfileAndTweets.ipynb) | Code to extract the data from the profiling and link it with twitter account. Repeating the code to catch any network failed records |
| data.csv | The profiling data after added generation column with excel |
| tweetsData.csv | All the tweets for all users with generation column this file will save a lot of time for you since generate this file may take over 5 h estimated, this file been generated in [extractUserProfileAndTweets.ipynb](http://localhost:8888/notebooks/Desktop/textAnalysisProject/extractUserProfileAndTweets.ipynb) |

**Data analysis and discover:**

Start to discover the data and group the data with generation groups which was:

|  |  |
| --- | --- |
| 30s | People who born from 1930 to 1939 |
| 40s | People who born from 1940 to 1949 |
| 50s | People who born from 1950 to 1959 |
| 60s | People who born from 1960 to 1969 |
| 70s | People who born from 1970 to 1979 |
| 80s | People who born from 1980 to 1989 |
| 90s | People who born from 1990 to 1999 |
| 00s | People who born from 2000 and after |

These groups been generated from date of birth it is not in the code because I did use excel to generate this column

The column been added to the csv file to save the time for later

Also generate graph to compare between percentage of the user per generation and tweets per generation.

Files:

|  |  |
| --- | --- |
| DataDiscovery.ipynb | Comparison between user percentage and tweets per generation |
| data.csv | The profiling data after added generation column with excel |

**Data cleansing:**

It was one of the important process where deleted the duplicate profiles duplicate tweets for same user also clean the text from image, links, emojis, mentions, spaces and punctuations

Also delete the empty tweets after the cleanup.

Files:

|  |  |
| --- | --- |
| dataCleansing .ipynb | Code to clean the tweets text and generate word matrix |
| wordMatrix.csv | The data for word matrix to be used in the model count of the words repeated per generation |

**Data preparation, model and testing:**

One of the long processors is break sentences to words where take each sentence and break it to word and classify it in which generation if it new word will be added if it was before it will add to the number.

Word matrix example

|  | **30s** | **40s** | **50s** | **60s** | **70s** | **80s** | **90s** | **00s** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **word** |  |  |  |  |  |  |  |  |
| **flying** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **high** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **above** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **president** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **trumps** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **threw** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **open** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **borders** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **ravaged** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **cities** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

After words create the model based on this table and train and test the model.

|  |  |
| --- | --- |
|  |  |

**Reference:**

<https://pypi.org/project/tweepy/>  
<https://www.geeksforgeeks.org/python-api-get_status-in-tweepy/?ref=rp>  
<http://gettwitterid.com/?user_name=ddlovato&submit=GET+USER+ID>  
<https://groups.google.com/forum/#!topic/twitter4j/Nibyf30jIBs>  
<https://stackoverflow.com/questions/753052/strip-html-from-strings-in-python>  
<https://www.geeksforgeeks.org/python-string-replace/>  
<https://hackernoon.com/how-to-scrape-google-with-python-bo7d2tal>  
<https://developers.google.com/people>  
<https://realpython.com/beautiful-soup-web-scraper-python/><https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2>