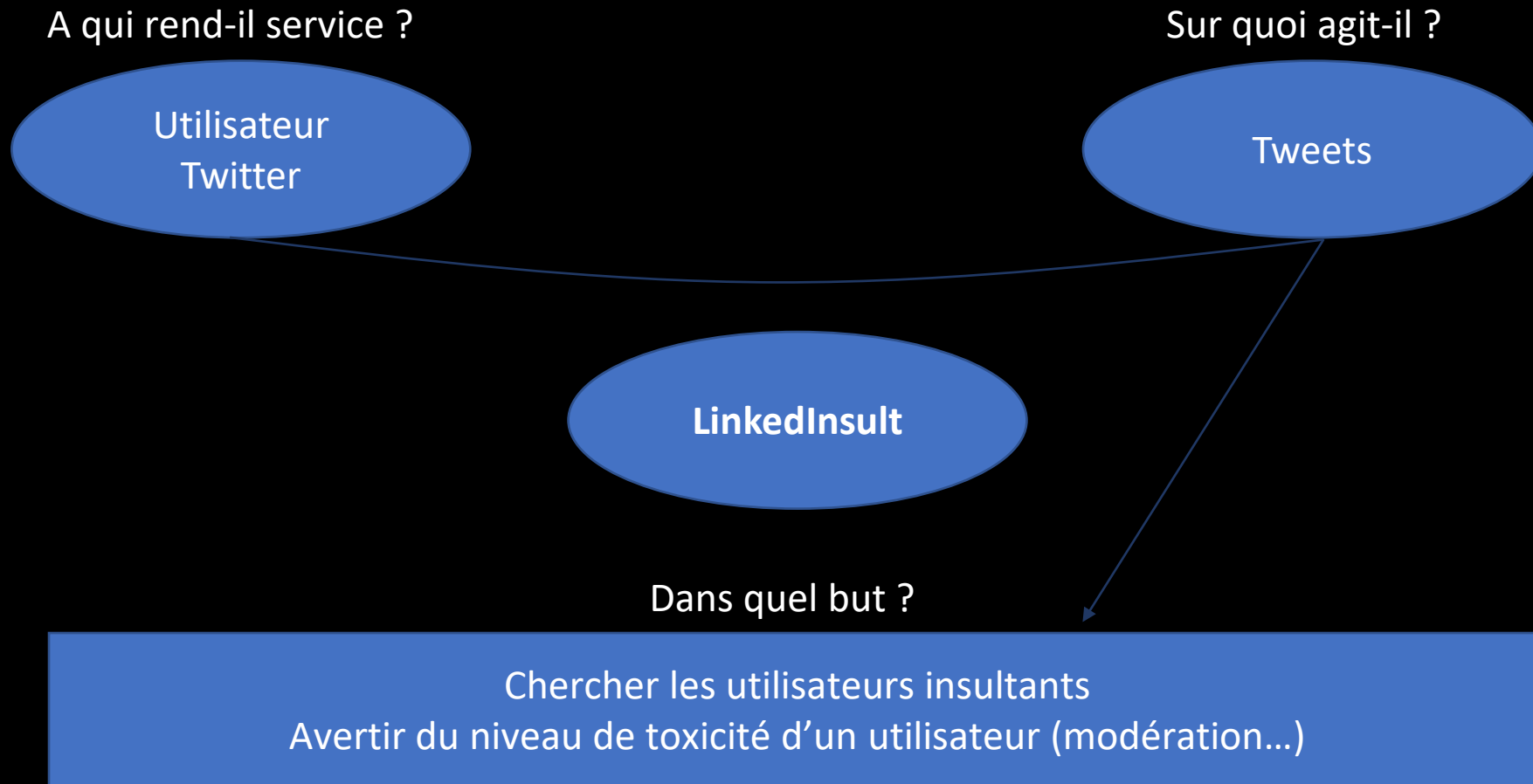


# LinkedInsult



Gabriel Martin  
Ilann Amiaud-Plachy  
Pauline Bidard  
Alexandre Caillot  
Julien Rosenberger  
Robin Dumas

# Expression du besoin



# Objectifs

## MVPs:

☑ MVP 1 : Détecteur d'insulte et profilage (un profil = une liste). Les queries tweepy mènent à compléter la base de donnée + Interface terminal

☑ MVP 2 : MVP1 + interface graphique avec Tkinter

MVP 3 : MVP2 + niveau de créativité, originalité ou répétitions ? vocabulaire réduit ?

☑ MVP 4 : MVP3 + apprendre à une IA à reconnaître les insultes

# README

## Projet des Rageux pour les Coding Weeks 2021

### LinkedInsult ou le Twitter des rageux

Ce projet est un projet d'analyse de tweets mené durant les Coding Weeks de CentraleSupelec

Le but du projet est de détecter les rageux.ses qui sévissent sur Twitter par une détection d'insultes au hasard sur le réseau social puis une analyse du profil de l'insultant (fréquence des insultes, violence des insultes, créativité des insultes, catégorie des insultes...). Les tweets analysés sont en anglais car le français n'est pas pris en charge par TextBlob. Les statistiques sur les insultes d'un utilisateur sont stockés dans une base de donnée et les profils sont consultables via une interface graphique.

### Rageux

- Amiaud--Plachy Ilann : contributor who wants to do the perfect insult censor.
- Bidard Pauline : contributor & Twitter developer
- Caillot Alexandre : contributor
- Dumas Robin : contributor & Twitter developer
- Martin Gabriel : Host and contributor
- Rosenberger Julien : Grammatical mistakes sensitive

### Utilisation

Modules utilisés:

- sys
- tweepy
- sqlite3
- Pandas
- Random
- TextBlob
- Tkinter
- Matplotlib.pyplot
- Matplotlib.figure
- Matplotlib.backends\_tkagg
- Json

pour la partie IA :

- sklearn
- numpy
- pickle

## Prérequis et déploiement

Avoir un compte Twitter Developer est indispensable.

Avant toute utilisation, **créer un fichier *filepath\_perso.py*** en parallèle de *main.py* et y coller la ligne suivante: >>> filepath = "chemin\_d'accès\_absolu\_vers\_dossier\_CONTENANT\_credentials"  
Avec *credentials.py* un fichier contenant les informations nécessaires à la connection à l'API Twitter (CONSUMER\_KEY, CONSUMER\_SECRET, ACCESS\_TOKEN, ACCESS\_SECRET)

Pour lancer l'application, il suffit d'exécuter main. Il est possible de choisir l'interface entre terminal (INTERFACE = 'command') et Tkinter (INTERFACE = 'graphic') directement dans main.  
L'utilisateur peut aussi fixer le nombre maximal de tweets récupéré par compte Twitter avec la variable NB\_MAX.

## Avertissement fonctions défaillantes

Certaines fonctionnalités ne sont pas encore opérationnelles :  
- faire des recherches de tweets en direct

# Structure et modularité

```
<insulteddetector_s2_rageux>
- 0001 filepath_perso.py
- 0002 .gitignore
- 0003 exemple_tweet.py
- 0004 bac_a_sable.py
- 0005 main.py
- 0006 README.md
- 0007 conception.md
- 0008 db_insulter.db
- 0009 Plan du projet.png
- <src>
  - 0001 insultDetection.py
  - 0002 profileAnalysis.py
  - 0003 data.py
  - 0004 findTweet.py
  - 0005 gui_main_window.py
  - 0006 from_twitter_to_database.py
  - 0007 getStats.py
  - 0008 gui_profile_window.py
  - 0009 tweetAnalysis.py
  - 0010 dataBase.py
  - 0011 cli.py
  - 0012 gui_add_menu.py
  - <IA_supervised_learning>
    - 0001 str_info.py
    - 0002 test_toxic_classifier.py
    - 0003 common_functions.py
    - 0004 toxic_classifier_generate.py
    - 0005 db_toxic.db
    - 0006 type_classifier
  - <tests>
    - 0001 test_insultDetection.py
    - 0002 test_findTweet.py
    - 0003 test_profileAnalysis.py
    - 0004 test_tweetAnalysis.py
    - 0005 test_dataBase.py
    - 0006 test_getStats.py
```

# Interface Graphique

Menu d'ajout

Mode de recherche

☒ Chercher une insulte

☐ Streamer une insulte

Mode d'analyse

☐ Mot à mot

☒ IA

Chercher

?

Sélectionner le type de recherche

Annuler

LinkedInsult

Insulteurs Répertoriés

@chomiaaa

@hellno\_bitch

@saraescartin00

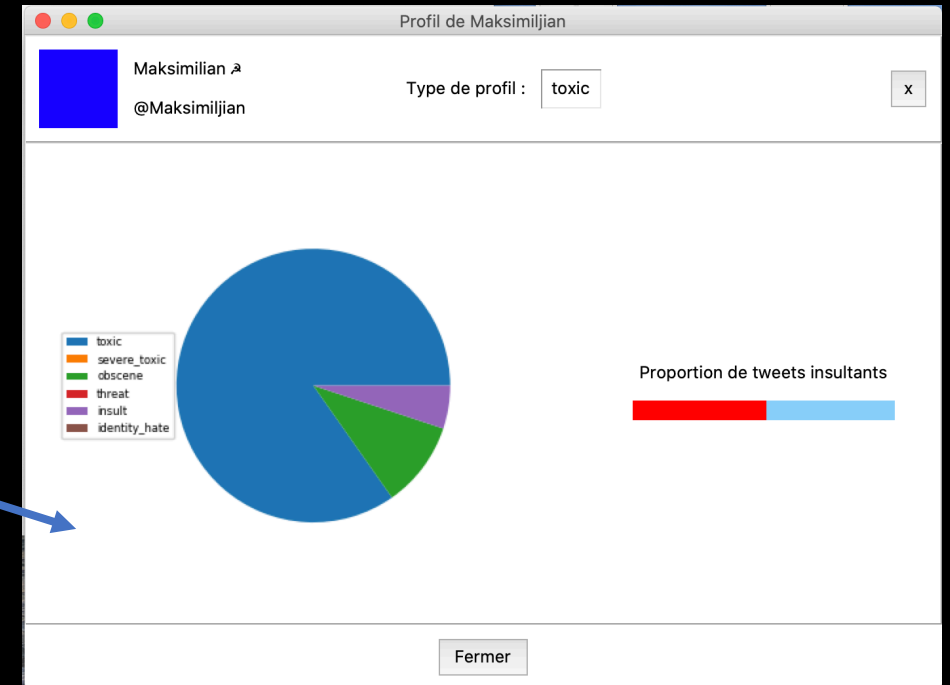
@wickedmouthed

@Maksimiljjan

+

Cliquer sur un insulteur pour consulter son profil

Fermer



# IA de classification

Accuracy of training model : 0.986461641317065					
Classification report :					
		precision	recall	f1-score	support
Toxic	0	0.89	0.67	0.76	3158
Severe_toxic	1	0.66	0.10	0.18	340
Obscene	2	0.88	0.67	0.76	1706
Threat	3	0.65	0.12	0.20	92
Insult	4	0.79	0.53	0.63	1600
Identity_hate	5	0.52	0.04	0.08	289
micro avg		0.86	0.58	0.69	7185
macro avg		0.73	0.36	0.44	7185
weighted avg		0.83	0.58	0.67	7185
samples avg		0.22	0.19	0.20	7185
Accuracy score : 0.7096378830083565					

Evaluation IA : max\_features = 1500  
n\_estimators = 100

# Paramétrage

- Nombre max tweets pris en compte par utilisateur
- Mode de recherche (Chercher une insulte, stream, chercher un utilisateur)
- Mode d'analyse (dictionnaire, IA)
- max\_features
- n\_estimators



# Pytest

```
===== test session starts =====
platform win32 -- Python 3.8.3, pytest-5.4.3, py-1.9.0, pluggy-0.13.1
rootdir: C:\Users\gaby2\Documents\CS\Cours\CW\rageux\insulteddetector_s2_rageux
plugins: cov-2.10.1
collected 17 items

tests\test_dataBase.py ..... [ 29%]
tests\test_findTweet.py . [ 35%]
tests\test_getStats.py .... [ 58%]
tests\test_insultDetection.py . [ 64%]
tests\test_profileAnalysis.py .. [ 76%]
tests\test_tweetAnalysis.py .... [100%]

===== 17 passed in 4.11s =====
```

# Vidéo de présentation



# LinkedInsult



Gabriel Martin  
Ilann Amiaud-Plachy  
Pauline Bidard  
Alexandre Caillot  
Julien Rosenberger  
Robin Dumas