utt
université de technologie
Troyes

RAPPORT IF29

Détermination de profils suspects sur Twitter

Aurélien Cecille

François Gillot

Hazem Zrig

Thomas Della Vedova

Timothee Jobard

Encadrant : Babiga Birregah

Semestre : Printemps 2020

SOMMAIRE

INTRODUCTION

Nous avons récupéré deux datasets de tweets envoyés pendant la coupe du monde 2018.
L'objectif est de repérer les profils suspicieux sur les 4,6 millions de tweets.

Dans un premier temps, puisque ce dataset est très lourd, il nous faudra explorer les données à partir d'un logiciel de gestion de base de données et les visualiser afin de les comprendre pour ensuite en étudier la sémantique.

Pour répondre à la problématique, nous allons utiliser deux approches :

- un algorithme de machine learning non-supervisé
- un algorithme de machine learning supervisé

Pour le premier nous allons utiliser une méthode de clustering afin de repérer différents clusters de types de profil : l'objectif est donc de repérer les *outliers*. Le choix des indicateurs est donc très important pour un clustering efficace et pour repérer facilement les profils suspects.
Pour le deuxième nous allons prendre le dataset de sortie de l'algorithme non-supervisé qui sera un dataset des profils twitter avec, pour chaque profil, une étiquette suspect ou non-suspect.

Il devra donc trouver les nouveaux suspects pour le dataset des tweets de base.
On comparera ensuite les deux méthodes par rapport aux résultats trouvés et avec différentes visualisations.

DATA EXPLORATION ET TRAITEMENT DES DONNÉES

a. Exploration des données

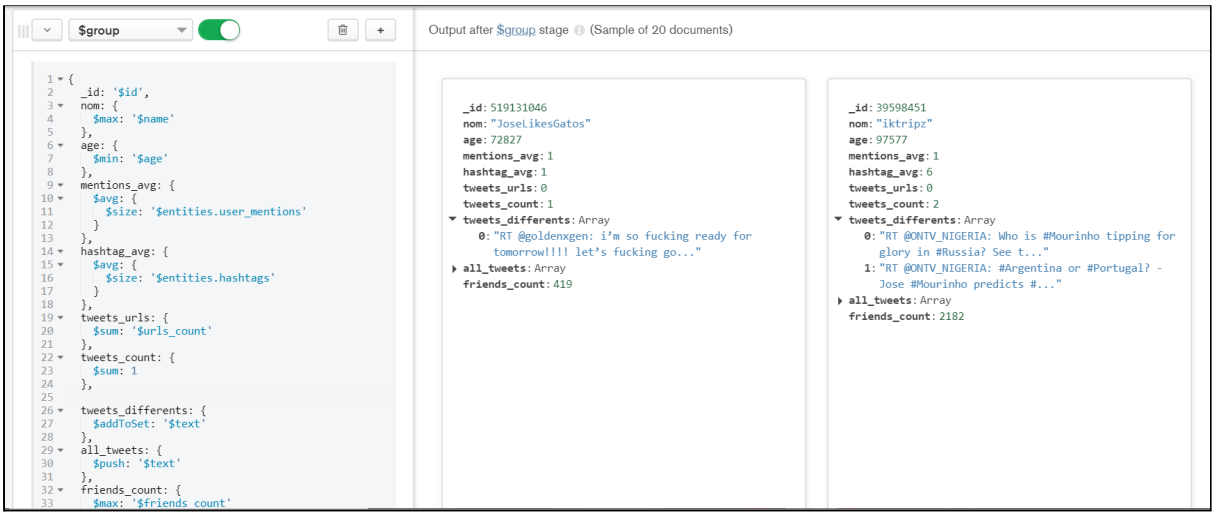
Nous avons décidé d'utiliser MongoDB pour toute cette première partie : pour le stockage, l'exploration et le traitement des données.

En effet l'API twitter nous fournit un bon nombre d'informations sur chaque tweet object (cf. figure 1).

Cependant peu de ces indicateurs sont pertinents pour repérer les suspects : nous avons donc explorer les différents tweets individuellement pour voir quels étaient les plus indicatifs pour trouver les suspects :

Age du compte
Nombre de followers
Nombre de profils suivi
Fréquence de publication de tweets
Fréquence d'ajouts d'amis
Nombre moyen d'URL (détecter les https)
Nombre moyen d'hashtag
Nombre moyen de mentions
Nombre moyen de retweets
Nombre de réponses moyen
Longueur moyenne des tweets
Nombre de tweets identiques
Analyse du texte des tweet (tweets malicieux)

Nous avons donc masqué les autres indicateurs et affiché les plus pertinents dans un premier temps, en groupant par ID des profils des tweets. Il s'agissait ici d'explorer les possibles bons indicateurs et faire une rapide analyse en filtrant par ordre décroissant du nombre d'url par tweets par exemple, de la moyenne des mentions et des hashtags.



A partir de ce traitement des données nous pouvons commencer à travailler dessus.
Nous avons ainsi exporté les 2000 tweets traités, dans un fichier CSV pour faire nos premiers tests avec l'approche non-supervisée.

b. Classification des suspects à la main

Par ailleurs, nous avons également décidé de classifier les suspects à la main en rajoutant une colonne "suspect" à notre dataset de 2000 tweets (1 pour suspect, 0 pour non suspect), pour une raison : comparer les résultats de l'algorithme non-supervisé et aussi le supervisé

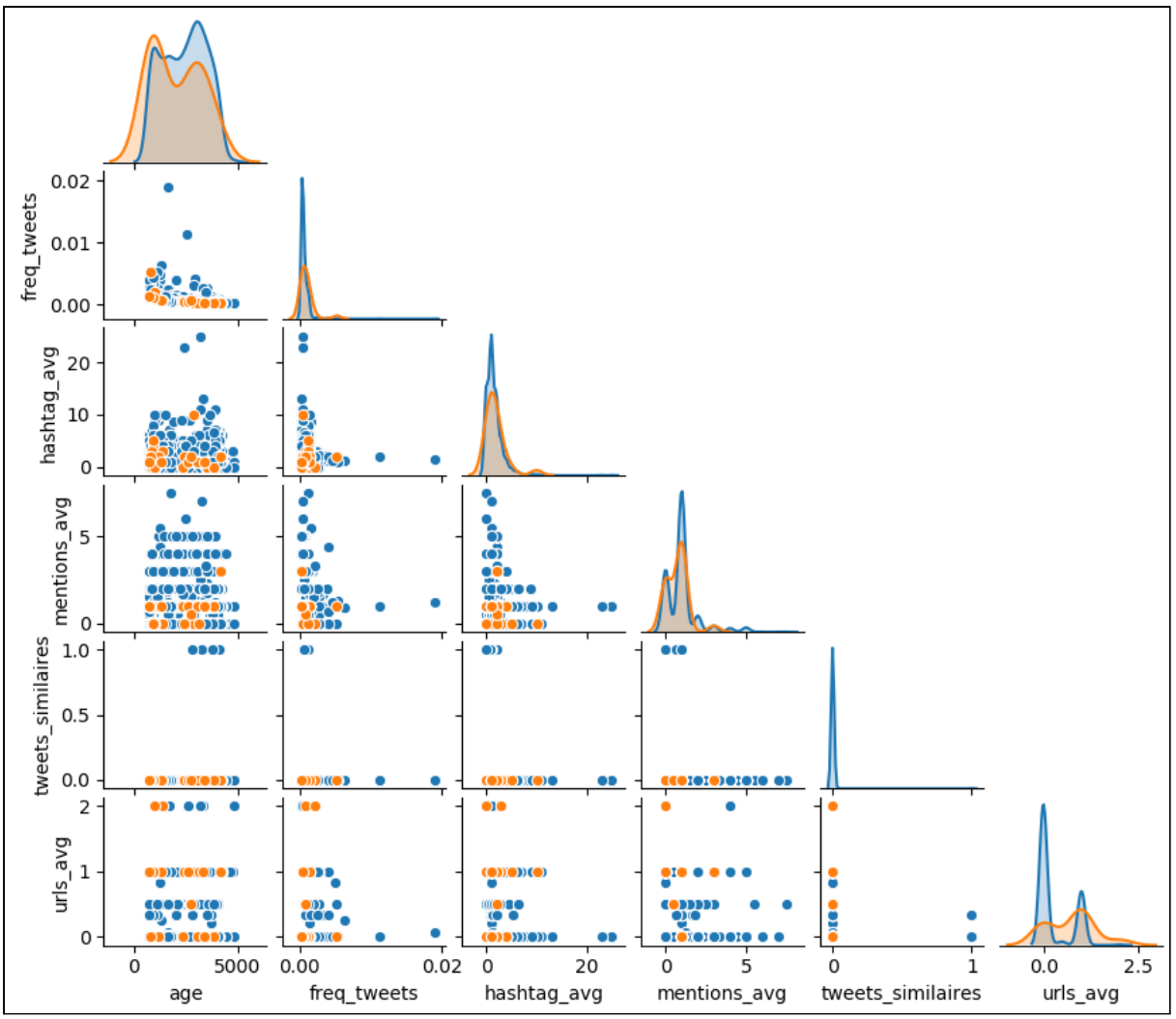
avec le dataset des 2000 tweets comprenant les suspects trouvés à la main et donc pour vérifier si l'algorithme fait aussi bien que nous voire mieux ou pire.

Pour trouver manuellement les profils suspects nous nous sommes répartis dans l'équipe, 500 tweets chacun à analyser et indiquer s'ils sont suspects ou non. Nous nous sommes basés sur le texte du tweet en lui-même mais aussi en regardant le contenu de l'url posté dans le tweet, ou l'image. Si le tweet comprenait par exemple, *FREE LIVE STREAM ou BITCOIN $$,* il est évident que le profil associé au tweet est suspect. De plus pour être très précis et ne pas nous tromper dans nos validations de type suspect ou non suspect, nous avons vérifié le contenu du profil twitter en lui-même.
Cela nous a pris un peu de temps mais ce fut par la suite un travail essentiel pour vérifier l'efficacité des algorithmes que nous avons décidé d'utiliser par la suite. Sans cela il aurait été compliqué de choisir un algorithme non-supervisé efficace.

⇒ Cette classification manuelle nous permet donc essentiellement d'évaluer les performances des deux algorithmes.

c. Choix des indicateurs finaux

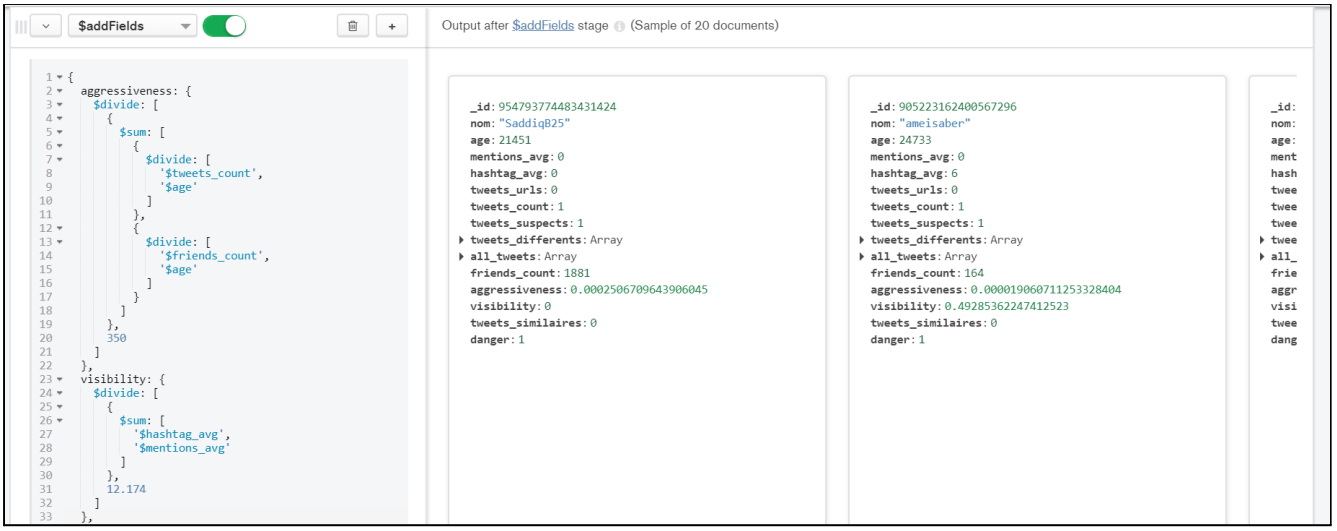
Pour choisir nos indicateurs finaux pertinents pour la recherche de profils suspects, nous avons donc fait une matrice de graphe de corrélation ce qui nous permet de voir rapidement les indicateurs intéressants par rapport aux autres.



On trouve peu de corrélation entre chaque indicateur ce qui montre qu'ils ont tous une importance et permettent chacun d'expliquer les données.

De plus, les tweets similaires étant quasiment égaux à 0 pour chaque profil, nous décideront plus tard de prendre un dataset de tweet "augmenté" avec non pas 2000 tweets aléatoires (en général on trouve un tweet par profil) mais un dataset de 2000 profils (1700 exactement) pour continuer nos tests et avoir des résultats plus représentatifs sur nos algorithmes de machine learning.

On remarque que plusieurs indicateurs expriment la même idée : la fréquence des tweets correspond à des profils agressifs si celle-ci est élevée, et la moyenne des mentions et celle des hastags, à la visibilité du profil si ces deux valeurs sont élevées.
Nous avons donc décider d'inclure ces indicateurs dans 2 indicateurs calculés : ***visibility*** et ***agressiveness***.



$$Aggressiveness = \frac{f_{tweets} + f_{f\,riends}}{350}$$

avec f_{tweets} : fréquence des tweets publiés par heure

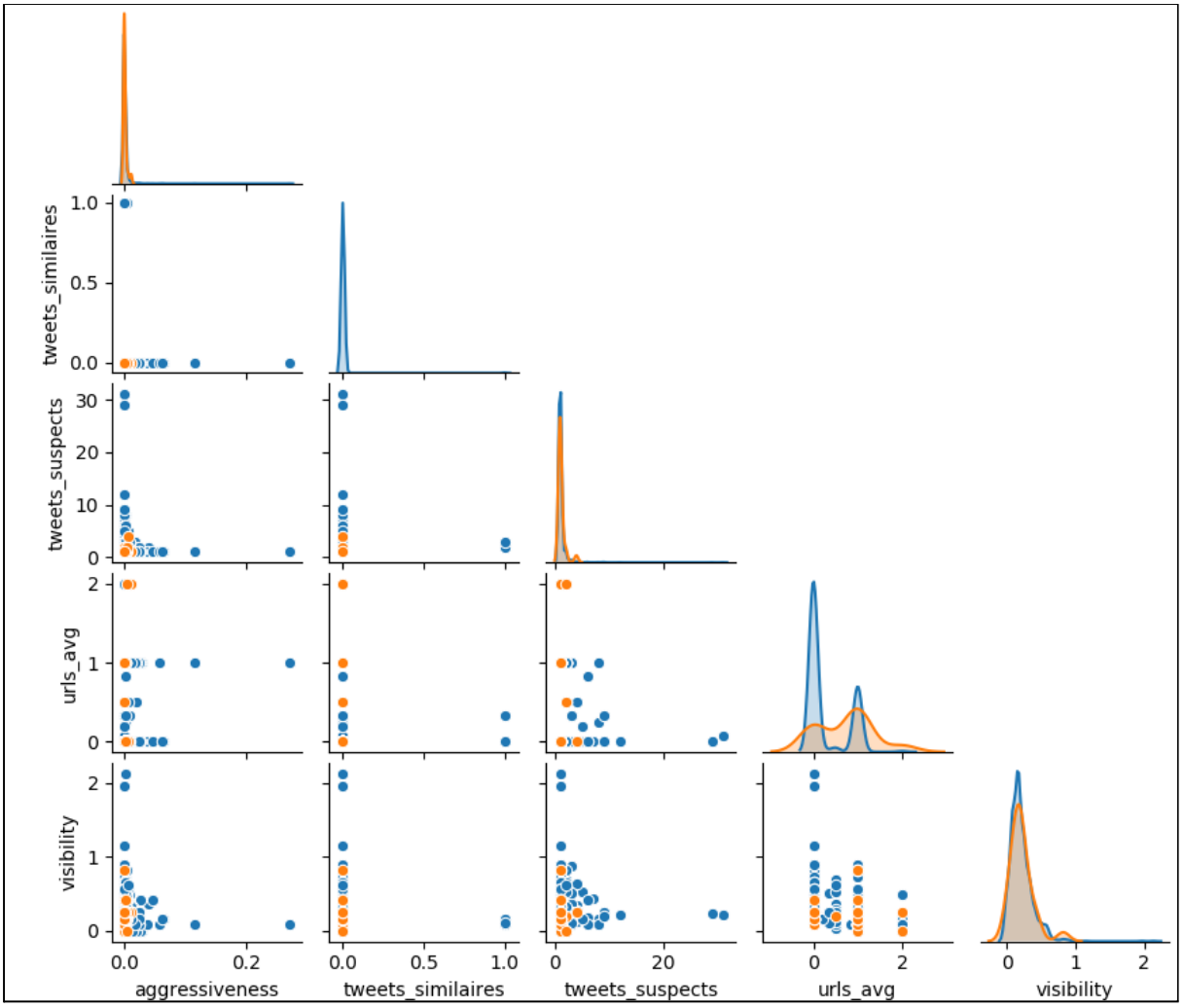
$f_{f\,riends}$: fréquence du nombre d'amis ajoutés par heure

350 : nombre d'actions maximum possibles imposé par l'API

$$Visibility = \frac{\sum_{E \in \{@,\#\}} Avg(E) \cdot C(E)}{140}$$

avec C(E) : cout moyen de caractère nécessaire pour une référence de @ ou # ($=11,5$)

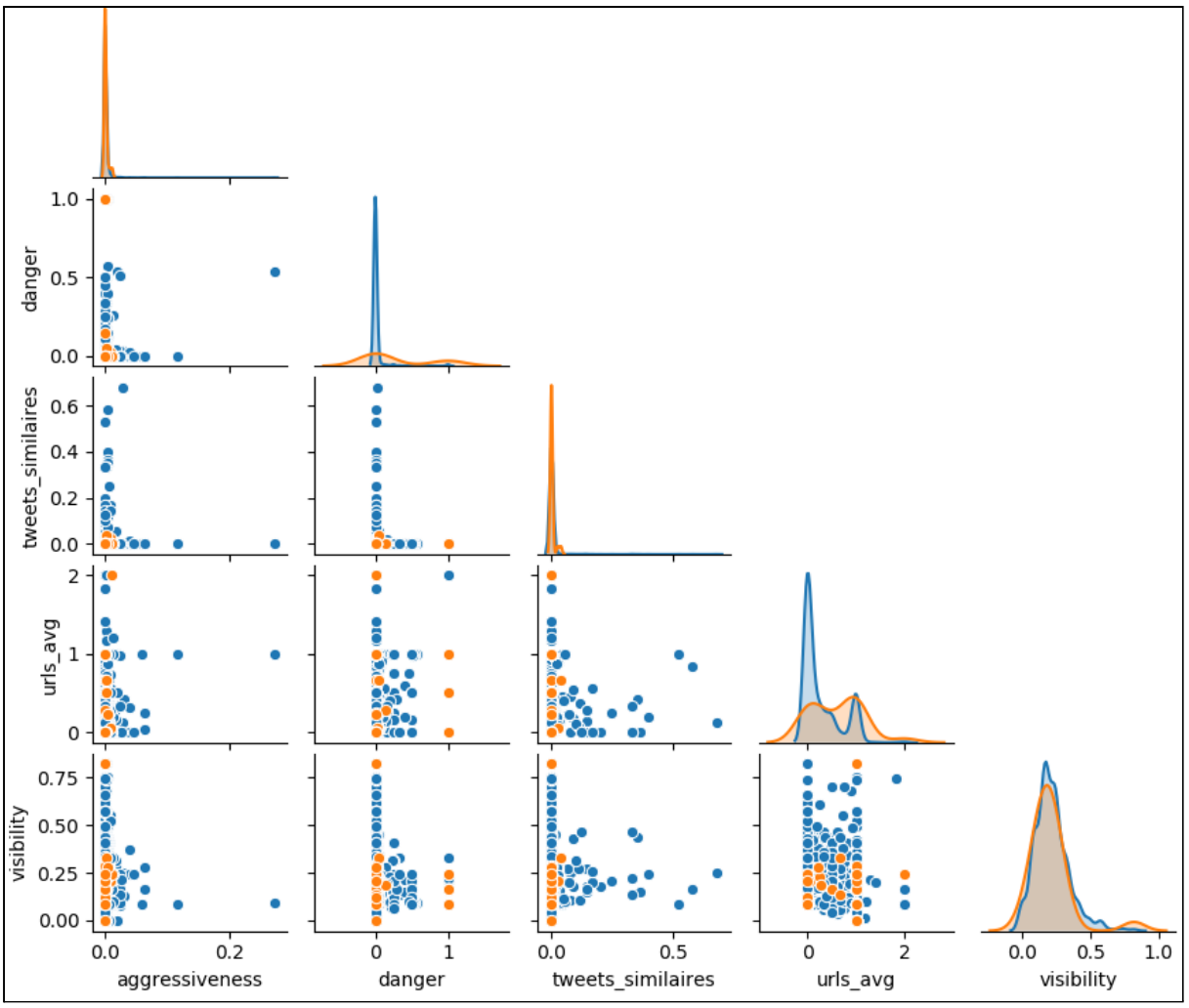
Ces nouveaux indicateurs renvoient des valeurs entre 0 et 1 et incluent les autres précédemment choisis. Ce regroupement est efficace et nous avons besoin de trois indicateurs calculés pour visualisés les résultats des algorithmes de machine learning.
De plus nous avons rajouté un indicateur sur mongoDB permettant de trouver un tweet malicieux. Pour cela nous avons fait une liste de mot suspects (sorte de blacklist) et donc un indicateur *tweets_suspects* qui somme le nombre de mot de la blacklist sur tous les tweets du profil en question.



Cependant nous n'obtenons pas plus de résultats intéressant ici.
Nous avons, donc, décidé d'augmenter le dataset comme décrit en page précédente.
Nous avons également rajouté un 3me indicateur calculé : ***danger***

$$Danger = \frac{tweets\ suspects}{tweets\ count}$$

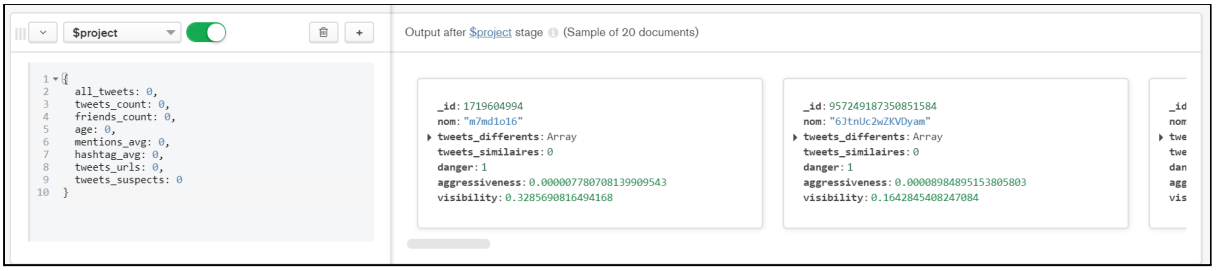
Avec tweets_suspects comptant le nombre de tweet du profil qui contiennent au moins un mot malicieux de la blacklist, et tweets_count, la somme des tweets du compte.



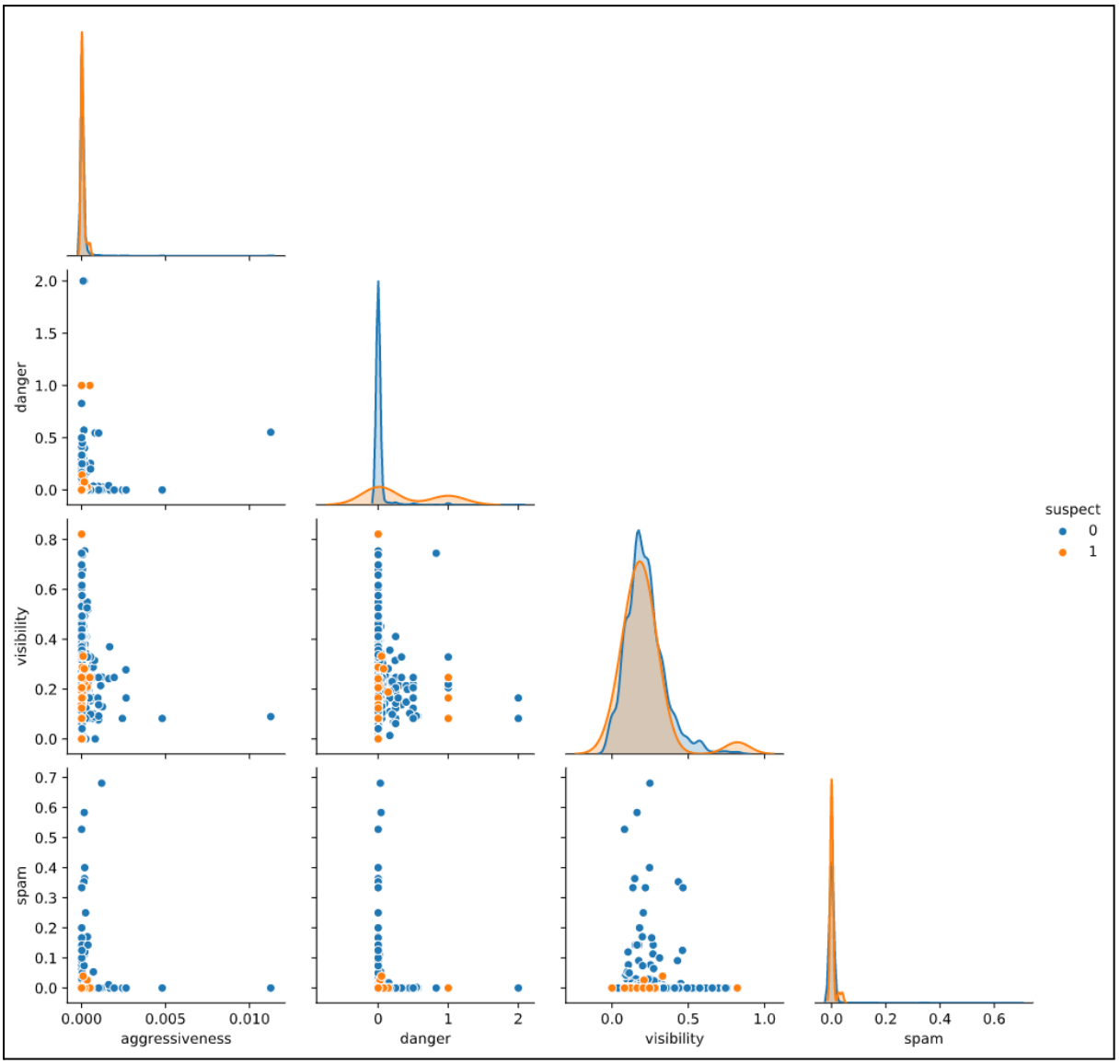
Les résultats étant un peu plus intéressant pour les tweets similaires que nous avons renommé ***spam*** qui est donc la fréquence des tweets similaires (indicateur précédemment choisi et correspondant à la somme des tweets similaires divisé par le nombre de tweets du compte twitter).

De plus, on observe que les profils suspects sont des tweets contenant au moins 2 urls ou plus. Nous avons donc créé un indicateur tweets_url comptant le nombre de tweets contenant au moins 2 urls ou plus, sur un même compte.
Nous avons rajouté cela dans l'indicateur calculé ***danger.***

$$Danger = \frac{tweets\ suspects\ +\ tweets\ urls}{tweets\ count}$$



On conserve donc, sous MongoDB, uniquement les 4 indicateurs avec le texte du tweet.



Nous nous sommes inspirés de SPOT pour 3 des indicateurs, une recherche similaire fourni par notre référent en IF29 par M. BIRREGAH.

Il nous reste donc 4 indicateurs : ***aggressiveness***, ***danger***, ***visibility***, ***spam*** qui nous semblent pertinents dans l'identification de profils suspects sur Twitter.

Dans ces indicateurs calculés nous avons choisi d'utiliser la fréquence au lieu de sommes afin d'éviter de détecter les comptes les plus actifs à l'inverse de détecter les comptes ayant le plus de chance d'être frauduleux.

Pour conclure voici nos 4 indicateurs :

$$Aggressiveness = \frac{f_{tweets} + f_{friends}}{350}$$

avec f_{tweets} : fréquence des tweets publiés par heure

$f_{friends}$: fréquence du nombre d'amis ajoutés par heure

350 : nombre d'actions maximum possibles imposé par l'API

$$Danger = \frac{tweets\ suspects + tweets\ urls}{tweets\ count}$$

$$Visibility = \frac{\sum_{E \in \{@, \#\}} Avg(E) \cdot C(E)}{140}$$

avec C(E) : cout moyen de caractère nécessaire pour une référence de @ ou # ($=11,5$)

$$Spam = \frac{tweets\ similaires}{tweets\ count}$$

Résultat de l'ACP

Variable	1	2	3	4
Pourcentage expliqué	27%	25%	24%	22%

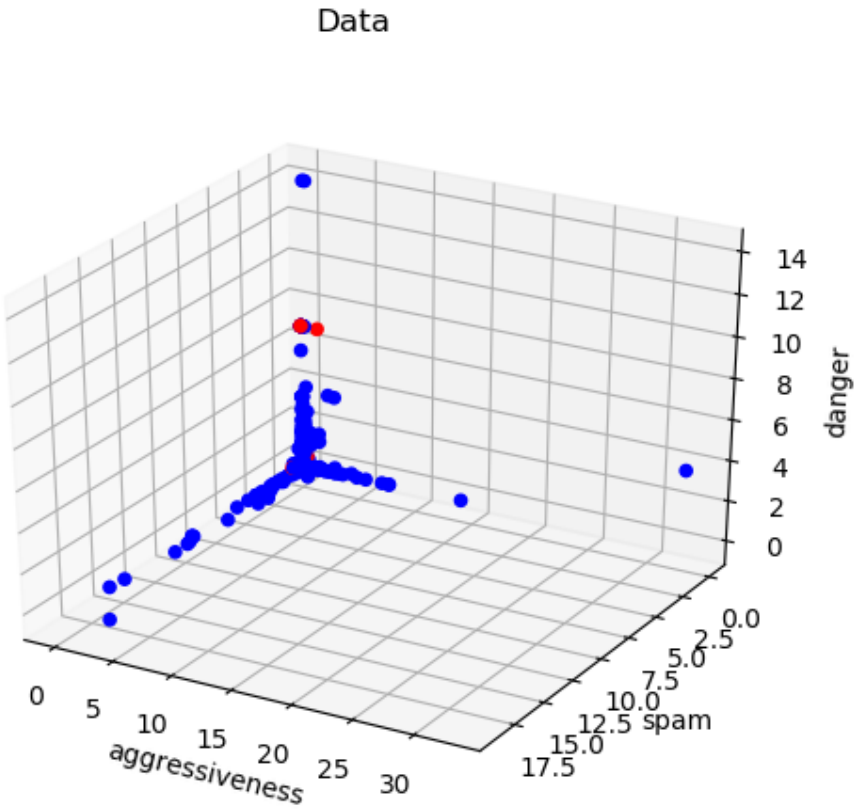
Si on supprime une variable de l'ACP on remarque que au moins 22% de l'explication des données sera supprimée.

Les 4 indicateurs expliquent donc tous les données pour chacun, il est donc pas intéressant d'en garder moins c'est à dire 3 ou 2.

⇒ L'utilisation de l'ACP n'a donc pas d'intérêt ici.

Visualisation en 3 dimensions :

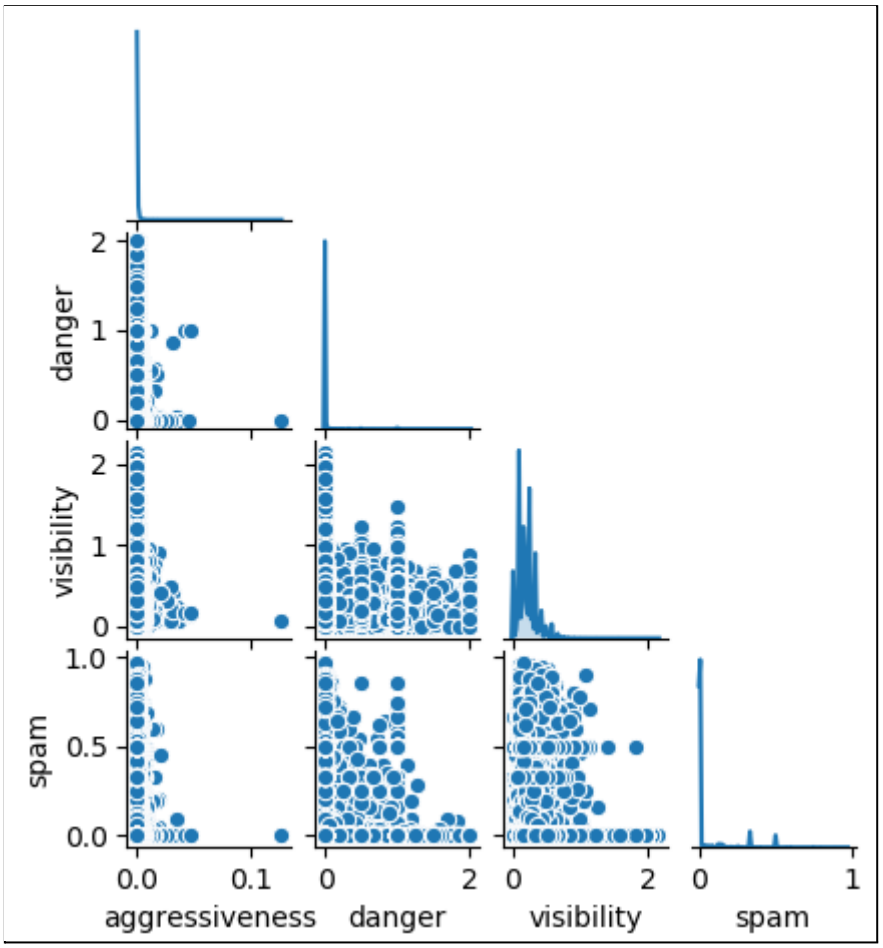
Maintenant que nous avons nos 4 indicateurs, nous allons visualiser en 3 dimensions les 2000 tweets (non “augmenté”) pour lesquels nous avons manuellement indiqué ceux suspects et ceux non suspects.



Les profils suspects parmi ces 2000 tweets, sont particulièrement notés comme *dangereux,* ce qui est logique car nous nous sommes basés essentiellement sur le texte du tweet et donc sur les mots malicieux (mais nous avons vérifié le profil de chaque tweet suspect pour être sûr).

Visualisation sur les 1.8 millions de compte du dataset total de tweets

Pour finir l'exploration nous avons cherché à voir les corrélations possibles des indicateurs sur tous les profils par rapport à nos critères.



On note que l'agressivité par rapport aux 3 autres critères, est souvent proche de 0 et ne varie pas en fonction des autres indicateurs.

APPROCHE NON SUPERVISÉE

Pour la partie non supervisée, il fallait tout d'abord qu'on essaye plusieurs algorithmes sur la tranche de 2000 données pour savoir lesquelles sont les plus pertinent. En effet, il est très important qu'on est de bon résultat sur le non-supervisée, cela permettra d'avoir de bon résultat dans la partie supervisée aussi. Au niveau des indicateurs, nous avons décidé de n'afficher que 3 indicateurs, l'agressivité, le danger et le spam pour pouvoir afficher les graphiques en 3 dimensions. Nous avons jugé que l'indicateur le moins pertinent était la visibilité, en effet, il y a une répartition plus dense des données avec la visibilité et c'est finalement l'indicateur qui permet le moins d'identifier les anomalies.

Alors, nous avons essayé bon nombre d'algorithme :

- Dbscan
- Kmeans
- Isolation forest
- LocalOutlierFactor
- EllipticEnvelope

Pour chacun des algorithmes, l'utilisation de la matrice de confusion a été très utile.

Matrice de confusion

Valeur à la main / Prédiction	**Positif**	**Négatif**
Positif	Comptes non-suspects correctement prédit (Vrai positif)	Comptes non-suspects incorrectement prédit (Faux positif)
Négatif	Comptes suspects incorrectement prédit (Faux négatif)	Comptes suspects correctement prédit (Vrai négatif)

Grâce à cette matrice, nous avons déterminé si les algorithmes étaient pertinent ou non. En effet, on doit avoir le moins de possible de faux négatif et de faux positif. De ce fait, après l'étude de la matrice de confusion pour chaque algorithme, nous avons déterminé que

l'algorithme IsolationForest est le plus adapté. L'IsolationForest est très pertinent au vu de sa matrice, de plus le temps d'exécution est assez court, ce qui permet une utilisation sur l'entièreté des données.

Pour l'IsolationForest, nous avons trouvé comme matrice de confusion :

1633	**43**
16	**8**

C'est un assez bon résultat, puisque nous trouvons 8 vrai négatifs, donc suspects et 43 faux positif qui sont donc les valeurs que le modèle détecte comme suspect mais que nous n'avions pas trouvé manuellement. Néanmoins il est intéressant de voir que lorsque nous somme aller voir ces faux positifs plusieurs de ces comptes était en réalité suspects.

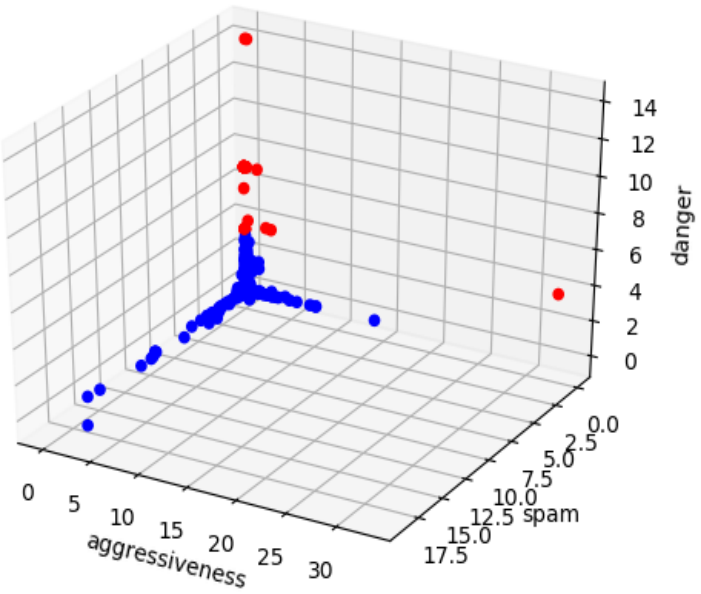
Exemple de faux positif :

RT @casinobitcoin: Bet with confidence and the best odds for the #WorldCup with #bitcoin and #adk only at https://t.co/s5Z1cFLQTC
🏆 WORLD CUP SPECIALS 🏆\n\nGet price boosts, free bets and tons more ahead of Saturday's games in Russia ! 👇… https://t.co/nEZ2kc8Ddp
📐Science seed\nhttps://t.co/RGDgpW9iPs\n\n#HealthXPh #Jobs #life #love #animals #startup #music #photography #poetry… https://t.co/icKkbYICvN
Get 100% up to £50 when you sign up for an account during the #WorldCup ⬇️\n\n18+. T&Cs apply.… https://t.co/JwjYDQGXmr
We are doing a #WorldCup2018 Challenge Pool. Sign up for free and win some $$: https://t.co/zGJoJgMAYd https://t.co/2BvJjOJXCn

Voici quelques graphiques et leur matrice de confusion sur les différents algorithmes testés sur la tranche de 2000 données :

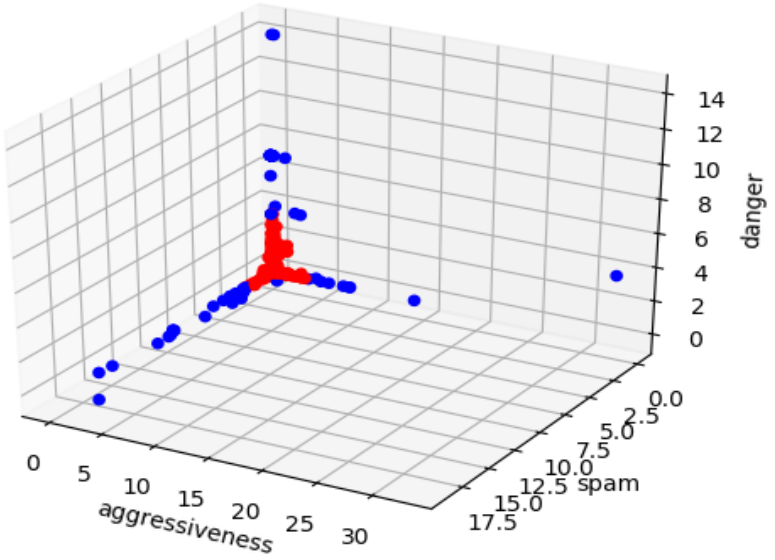
KMEANS :

1649	**27**
16	**8**



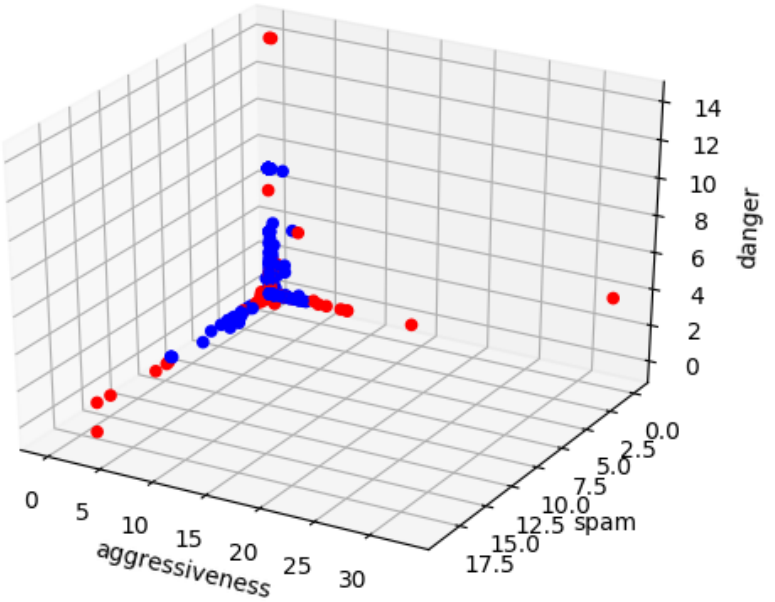
DBSCAN :

1624	**52**
16	**8**



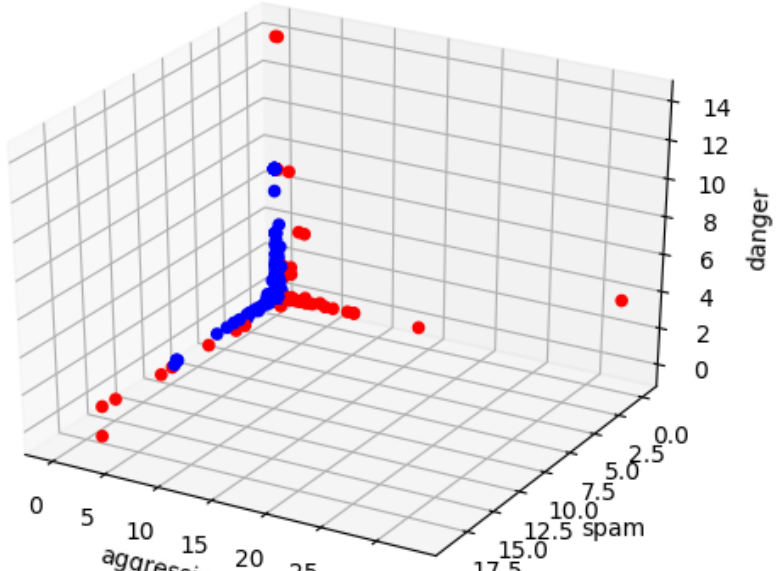
LocaloutlierFactor :

1509	**167**
21	**3**



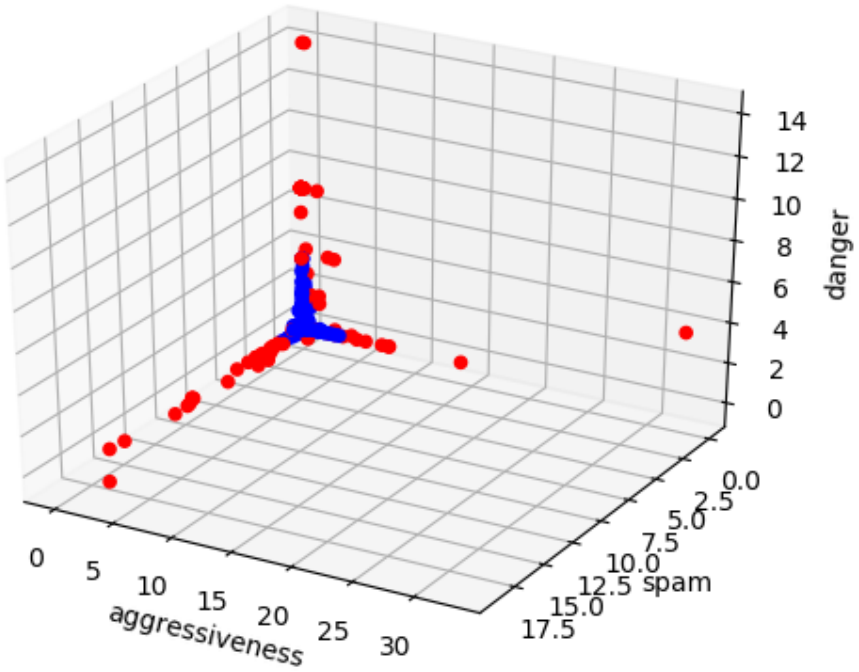
EllipticEnvelope :

1594	**82**
21	**3**



IsolationForest :

1633	**43**
16	**8**

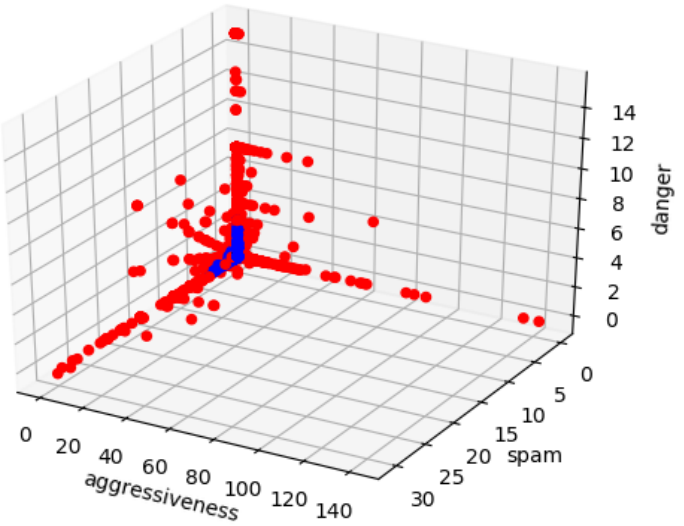


Sur les graphiques ci-dessus, nous pouvons remarquer que les intrus sont en rouge pour le Kmeans, l'IsolationForest, l'EllipticEnvelope et le LocaloutlierFactor alors que pour le Dbscan, les suspects sont en bleus.

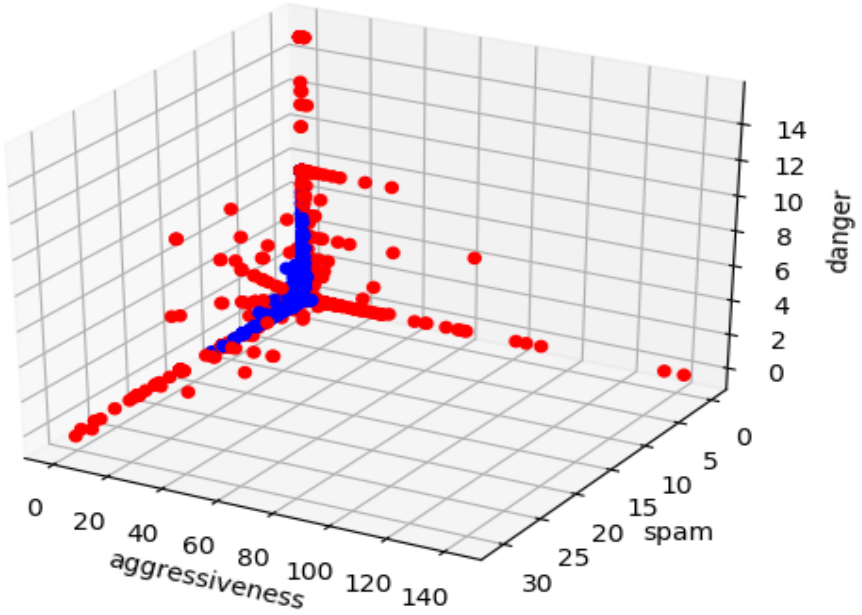
On peut noter que Kmeans a de meilleurs résultats que IsolationForest. Cependant, Kmeans ne prend en compte que le danger et pas nos deux autres critères. Nous supposons alors que le Kmeans se généralisera moins bien sur un plus grand échantillon de données. Sur un petit échantillon de données le spam est moins important puisqu'il y a peu de tweets. Par contre sur un grand échantillon de données, l'indicateur spam est très important au vu du nombre de tweets. C'est pour cela que si nous ne prenons pas en compte cet indicateur cela peut baisser la pertinence nos résultats, c'est pourquoi on a choisi IsolationForest.

Voici les graphiques que nous avons retenu sur l'entièreté des données en fonction du paramètre contamination, qui est la proportion de suspect dans le jeu de données :

IsolationForest, contamination = 5%



IsolationForest, contamination = 2%



Ce graphique a été obtenu en émettant comme hypothèse que les comptes étant différents des autres sont alors suspects suivant nos critères. Nous pensons que 2% de contamination est plus intéressant puisqu'il reflète les résultats que nous avons trouvés à la main.

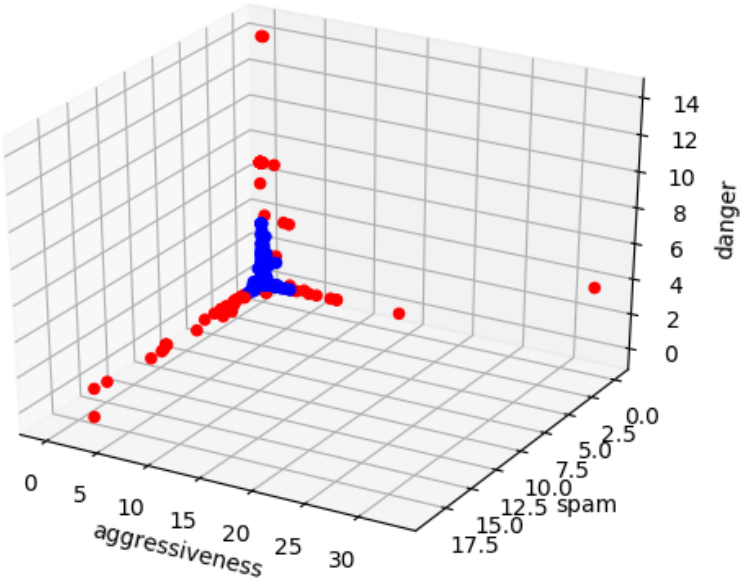
APPROCHE SUPERVISÉE

L'approche supervisée consiste à apprendre une fonction à partir d'une base d'apprentissage. Nous avons choisi d'utiliser l'algorithme SVM (Support Vector Machine) afin de détecter les tweets malveillants.

Il fallait tout d'abord entraîné notre modèle, pour cela il faut que le modèle sache quel compte est suspect et quel compte ne l'est pas. Ce que nous n'avons pas au préalable dans notre jeu de données. Donc nous avons choisi d'utiliser les données trouvées à l'aide de l'approche non supervisée, afin d'entraîner notre modèle supervisé. Nous avons donc utilisé 1700 comptes suspects et non suspects pour notre jeu d'entraînement, qui ont déjà été définis dans l'approche non supervisée et nous ont permis d'entrainer notre modèle actuel.

Par la suite grâce au modèle obtenu, nous avons classifié l'ensemble des données. Nous obtenons donc une modélisation en 3D comprenant différentes échelles : spam, agressivité et danger. Les comptes qui peuvent être considérés comme malveillant sont, ici, obtenus en rouge.

Cette première modélisation représente seulement les 1700 comptes que nous avons défini comme suspect ou non à la main pour évaluer le modèle, s'il fonctionnait correctement.



Dans un premier temps nous avons évaluer la performance avec les données de test pour comparer ce modèle avec le non-supervisée.

482	**0**
7	**21**

Nous obtenons la matrice suivante :

Dans ce cas le modèle ne devrait avoir ni faux positif ni faux négatif et on peut voir que les résultats ne sont pas mauvais puisqu'il ces valeurs sont assez faible.

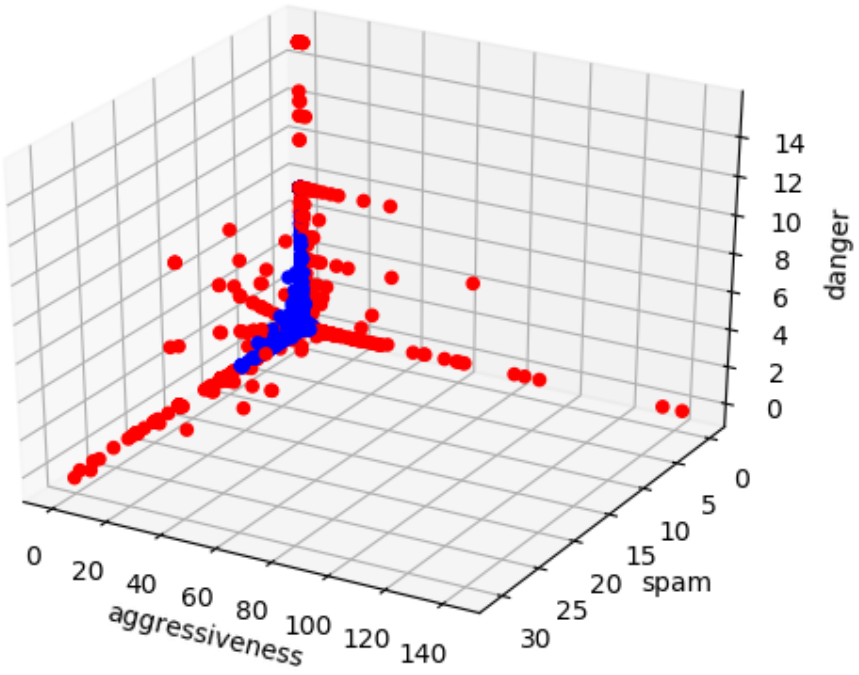
1614	**62**
15	**9**

Dans un second temps nous avons comparé les résultats avec les données récolté manuellement comme nous le faisions précédemment et avons obtenu la matrice ci-dessous très similaire aux modèle précédents :

Puis nous avons ensuite entrainé le modèle sur l'ensemble des données et l'avons évalué avec ses données de test. Cela nous a permis d'obtenir la seconde modélisation ci-dessous.
La matrice de confusion obtenu montre que le modèle est pertinent, puisque l'on obtient seulement 70 faux négatif et 22 faux positif pour 645 vrai négatifs.

IsolationForest avec toutes les données

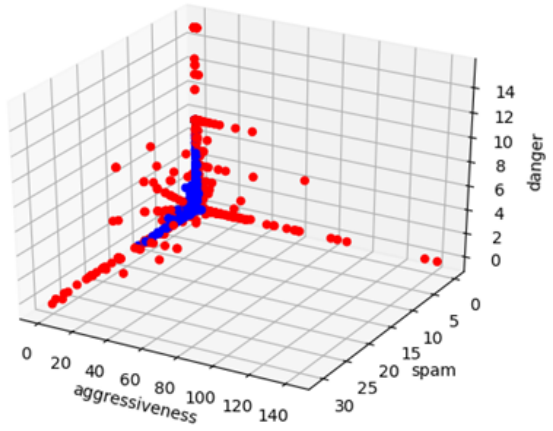
36147	**22**
70	**645**



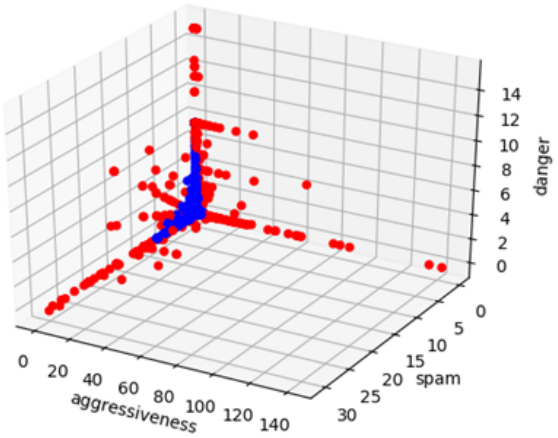
Temps d'entraînement du modèle : 31 min 42

COMPARAISON DES DEUX MÉTHODES

Après avoir réalisé les deux approches, on peut donc les comparer, même si intuitivement le modèle supervisé devrait être plus efficace que le modèle non-supervisé.



Non-supervisé



Supervisé

On note quelques légères différences au niveau des faux positif et faux négatifs notamment concernant le spam, mais elles sont minimes.

Il s'avère que les résultats sont finalement équivalents ! Ce qui est en fin de compte logique, car on utilise les résultats du modèle non-supervisé pour entraîner le modèle supervisé, et comme les résultats obtenus à partir de celui-ci sont très bon, le modèle supervisé apprend de ce modèle et obtient donc logiquement des résultats semblables.

CONCLUSION

L'objectif de trouver des profils suspects sur Twitter fut passionnant ! Tout au long du projet nous nous sommes chacun réparti des tâches et avons trouvé des solutions que ce soit pour stocker, explorer, traiter visualiser les données de profils Twitter ou chercher des indicateurs pertinents.

Il est intuitivement impossible de trouver des profils suspects à la main sur l'ensemble des 4,6 millions de tweets mais cela est possible grâce au machine learning notamment !
Nous avons obtenu de très bons résultats en s'appuyant sur un fichier de profils de suspects trouvés à la main pour trouver des algorithmes efficaces en fonctions des 4 indicateurs pertinents qui finalement s'avèrent être les pièces maîtresses du projet.

Dans un objectif d'amélioration continue de notre projet, nous aurions aimé avoir des critères supplémentaires concernant les caractéristiques d'un tweet, comme le nombre de retweet par exemple, indicateur qui aurait pu être intéressant d'utiliser dans nos modèles de machine learning.

Nous avons réutilisé les notions apprises en cours d'IF29 dans le cadre de notre projet et ce fut très formateur pour acquérir de nouvelles compétences en Data Analytics et que nous pourrons appliquer en milieu professionnel, ce fut un projet très intéressant avec de très bon résultats dans lequel nous nous sommes beaucoup impliqués et qui a plu à tous les membres du groupe du projet.

Bibliographie

Kmeans :
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html?highlight=kmeans#sklearn.cluster.KMeans

DBSCAN :
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html?highlight=dbscan#sklearn.cluster.DBSCAN

Détection d'anomalies :
https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection

LocalOutlierFactor :
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html#sklearn.neighbors.LocalOutlierFactor

EllipticEnvelope :
https://scikit-learn.org/stable/modules/generated/sklearn.covariance.EllipticEnvelope.html#sklearn.covariance.EllipticEnvelope

IsolationForest :
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html?highlight=isolation#sklearn.ensemble.IsolationForest

SVM :
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html?highlight=svc#sklearn.svm.SVC