Classification des utilisateurs à partir de leurs caractéristiques de frappe de mot de passe

Léo Pichery, Dejoie Adèle

Février 2025

Table des matières

Introduction

Nous nous proposons d'étudier des données relatives à la frappe d'un mot de passe. Pour cela, 51 personnes ont été invitées à taper le mot de passe `".tie5Roanl"` à 376 reprises, tout en enregistrant les instants d'appui et de relâchement de chaque touche.

Ce mot de passe étant constitué de 10 caractères, la base de données associée contient 21 temps différents pour chaque tentative de chaque individu. Ces données se présentent ainsi : l'instant noté $H.x$ correspond au moment où la touche x est enfoncée, tandis que $UD.x.y$ désigne l'instant où la touche x est relâchée pour passer à la touche y. Le dernier enregistrement correspond à la pression de la touche "Entrée". En déduisant $UD.x.y - H.x$, on obtient la durée pendant laquelle la touche x est maintenue enfoncée. En examinant les valeurs de la base, on observe que certains de ces écarts sont négatifs, indiquant que la touche y a été enfoncée avant même que la touche x ait été relâchée. Ce phénomène est fréquent chez les personnes tapant rapidement et anticipant la position des lettres.

Dans le cadre de cette étude, nous utiliserons uniquement la moitié des essais réalisés par chaque individu, soit 188 tentatives par personne. Cela nous amène à analyser 21 variables sur un total de $51 \times 188 = 9588$ observations.

L'objectif est de caractériser ces individus à partir de leur manière de taper, en exploitant des techniques de classification. Cette approche pourrait notamment renforcer la sécurité des systèmes informatiques en vérifiant que la frappe du mot de passe correspond bien au profil habituel de l'utilisateur, réduisant ainsi les risques d'usurpation d'identité. Par ailleurs, elle permettrait d'identifier des changements de comportement, par exemple liés au stress, ou encore d'adapter les interfaces aux capacités motrices des utilisateurs (ralentissement de l'affichage des suggestions pour une frappe lente, ajustement des prédictions en fonction du rythme de saisie). Pour cela, nous expérimenterons plusieurs types de classifications : binaire, multi-classe, et par regroupements de niveau, afin de mettre en évidence les caractéristiques permettant de différencier les utilisateurs.

1 Analyse en composante principale

1.1 Préambule

Le point de départ de notre analyse est de déterminer si nos données doivent être standardisées ou non. Étant donné qu'elles représentent des instants, les centrer ne fait pas perdre d'information et facilite leur interprétation. Cependant, en examinant la matrice des moyennes de chaque instant pour l'ensemble des individus, on observe que les ordres de grandeur sont globalement similaires, mais que certains points apparaissent comme des outliers. Ces derniers correspondent aux instants de relâchement des touches, c'est-à-dire aux moments de transition vers une autre touche.

Compte tenu de la nature des données, il semble important de conserver l'information relative à des temps de transition plus longs (ou plus courts) que la moyenne, car cela permet de quantifier la recherche des touches. C'est pourquoi nous choisissons de centrer nos variables sans les réduire. Toutefois, afin de valider cette approche, nous mènerons une analyse parallèle en utilisant des données standardisées et comparerons les résultats obtenus.

Dans un premier temps, nous allons effectuer une analyse en composantes principales (que nous appellerons ACP par la suite), dans le but de réduire le nombre de variables explicatives tout en conservant une quantité d'information satisfaisante.

1.2 Réalisation de l'ACP

En effectuant l'ACP numériquement, nous obtenons tout d'abord la Figure 1, qui montre le pourcentage d'inertie expliquée par chacune des composantes, ainsi que le pourcentage cumulé tracé en rouge. Nous constatons que 99 % de l'inertie est expliquée par les 11 premières composantes. Nous représentons donc ce graphique à nouveau en ne conservant que ces 11 premières composantes, ce qui donne la Figure 2.

L'ACP permet ainsi de diviser par deux le nombre de variables explicatives. Comme le nombre initial n'est pas très élevée, elle ne joue pas un rôle central dans la réduction de dimension de notre jeu de données mais elle reste pertinente pour la visualisation et l'extraction d'informations utiles.

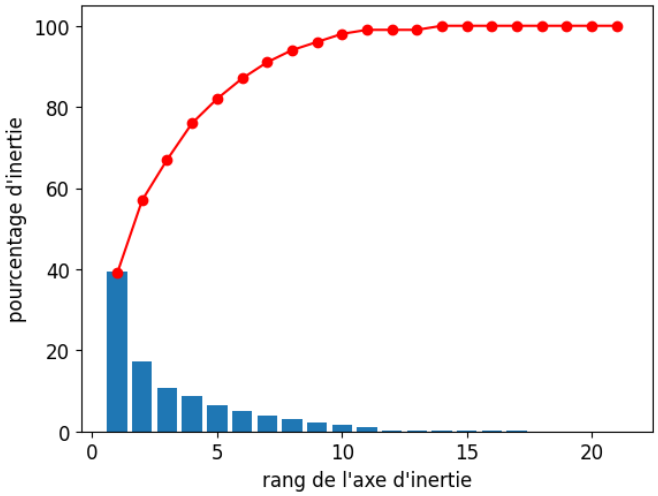

FIGURE 1 – Répartition de l'inertie expliquée par les différentes composantes (données uniquement centrées)

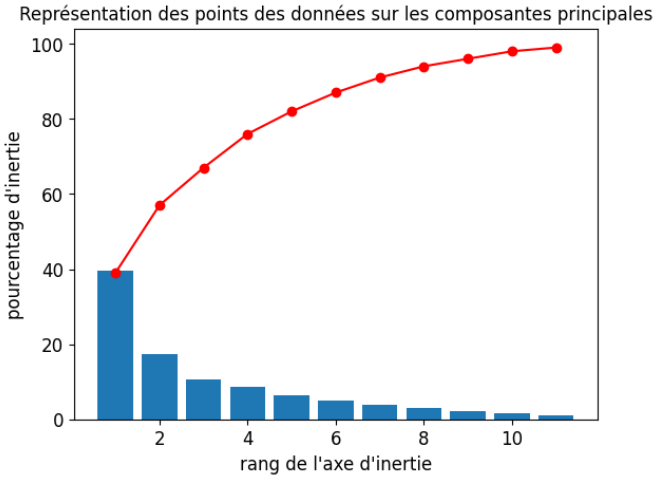

FIGURE 2 – Répartition de l'inertie expliquée par les 11 premières composantes (données uniquement centrées)

Dans la Figure 2, nous observons que les deux premières composantes représentent 56,7 % de l'inertie totale, ce qui n'est pas suffisant pour se limiter à celles-ci. Toutefois, il peut être intéressant de représenter les données dans le plan factoriel formé par ces deux composantes afin d'explorer une classification binaire consistant à distinguer un individu de tous les autres. La Figure 3 illustre cette approche.

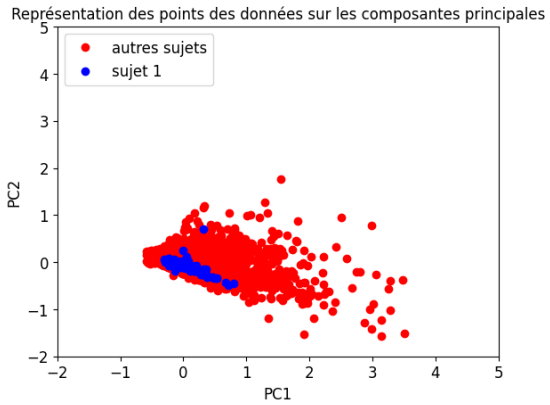

FIGURE 3 – Nuage de points représentant les données du sujet 1 face à tous les autres sujets (données uniquement centrées)

Cette représentation montre que les instants mesurés semblent peu varier d'un individu à un autre, comme en témoigne l'amas de points dense. Cela suggère qu'une classification par classes sera probablement plus efficace pour différencier les individus.

Avant d'interpréter plus en détail cette ACP, nous devons vérifier notre hypothèse selon laquelle il est préférable de seulement centrer nos variables. Pour cela, nous effectuons une nouvelle ACP sur le jeu de données standardisé et obtenons la Figure 4.

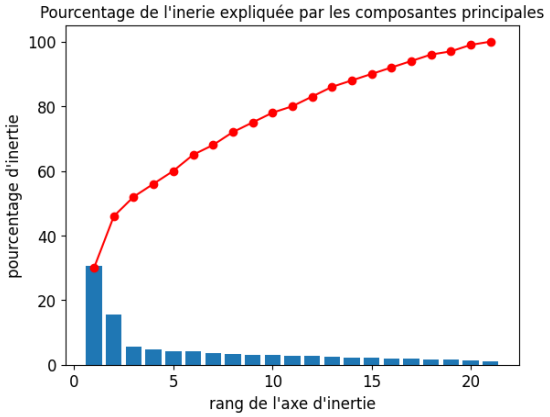

FIGURE 4 – Répartition de l'inertie expliquée par les différentes composantes (données standardisées)

Nous constatons que, dans ce cas, les 11 premières composantes n'expliquent plus que 80 % de l'inertie totale, ce qui est inférieur au cas précédent. De plus, les représentations en nuages de points se font généralement dans un plan défini par

deux composantes principales. Or, il est préférable que celles-ci capturent 56,7 % de l'inertie totale (cas des données centrées uniquement) plutôt que 46 % (cas des données standardisées), afin de maximiser la rétention d'information.

Ainsi, notre hypothèse initiale est validée. Toutefois, nous poursuivrons la comparaison entre les résultats obtenus avec les données centrées et ceux obtenus avec les données standardisées afin d'affiner notre analyse.

1.3 Interprétations des résultats de l'ACP

Afin de comprendre l'importance de nos différentes variables et leur représentation dans les composantes principales, nous devons tout d'abord effectuer une décomposition en valeurs singulières. Cela permet d'obtenir les directions des composantes principales ainsi que les valeurs singulières, qui sont liées à la quantité d'inertie expliquée par chaque composante. Le tableau 1 ci-dessous présente les poids associés à chaque variable dans les deux premières composantes principales, d'abord pour les données uniquement centrées, puis pour les données standardisées.

TABLE 1 – Poids associés à chaque variable dans les deux premières composantes principales

Variables	**V1centrée**	**V2centrée**	**V1stand**	**V2stand**
H.period	-0.02	-0.00	-0.27	-0.09
UD.period.t	0.35	0.06	0.16	-0.30
H.t	-0.01	0.00	-0.28	-0.24
UD.t.i	0.12	0.00	0.13	-0.17
H.i	-0.01	0.00	-0.28	-0.14
UD.i.e	0.38	0.89	0.08	-0.19
H.e	0.00	0.00	-0.25	-0.23
UD.e.five	0.44	-0.34	0.13	-0.26
H.five	-0.01	0.00	-0.25	-0.12
UD.five.Shift.r	0.46	-0.24	0.11	-0.36
H.Shift.r	-0.01	0.00	-0.28	-0.16
UD.Shift.r.o	0.30	-0.11	0.20	-0.29
H.o	-0.01	0.00	-0.30	-0.13
UD.o.a	0.11	-0.03	0.13	-0.26
H.a	0.00	-0.00	-0.22	-0.25
UD.a.n	0.12	-0.00	0.17	-0.24
H.n	-0.02	0.00	-0.26	-0.21
UD.n.d	0.05	-0.00	0.12	-0.14
H.d	0.00	0.00	-0.25	-0.19
UD.d.l	0.02	-0.00	0.13	-0.20

Pour les données uniquement centrées, on constate qu'il existe des variables pour lesquelles les poids associés sont très supérieurs à ceux des autres variables. Comme mentionné précédemment, cela correspond aux variables qui mettent en relief le temps nécessaire au sujet pour trouver un autre caractère sur le clavier. Ce tableau met également en évidence, pour les données uniquement centrées, l'existence de variables qui varient significativement entre les sujets, comme "UD.five.Shift.r", qui possède le poids le plus élevé dans V1centrée et qui est nettement supérieur à la plupart des autres.

En revanche, pour les données standardisées, on constate que toutes les variables ont des poids relativement similaires, ce qui renforce encore notre hypothèse sur la non-standardisation des données. Ces informations se visualisent clairement grâce au cercle des corrélations, que nous pouvons représenter dans le plan formé par les deux premières composantes principales pour les deux cas. Cela est illustré par la Figure 5 et la Figure 6.

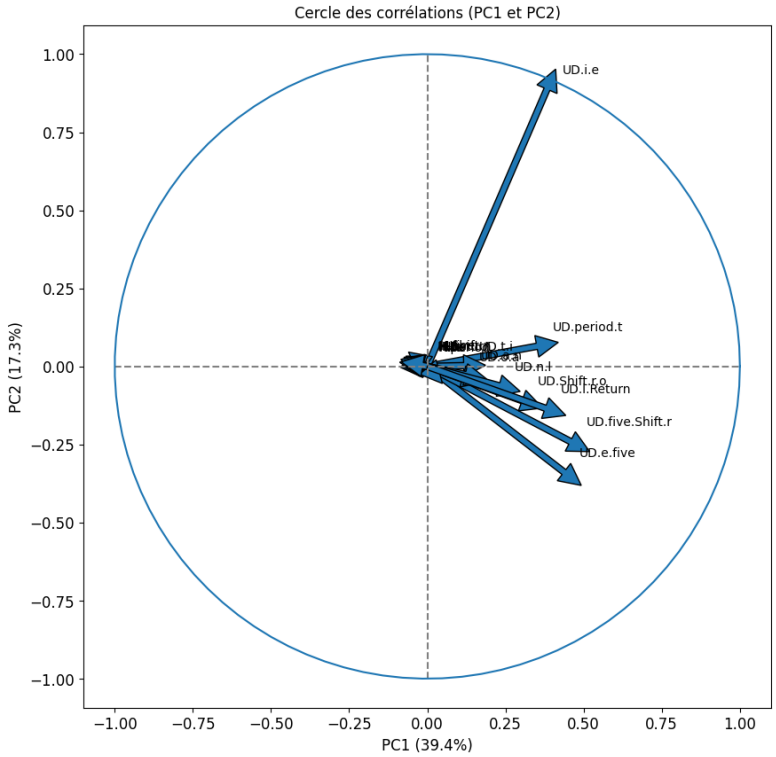

FIGURE 5 – Cercle des corrélations dans le plan formé par les deux premières composantes principales (données uniquement centrées)

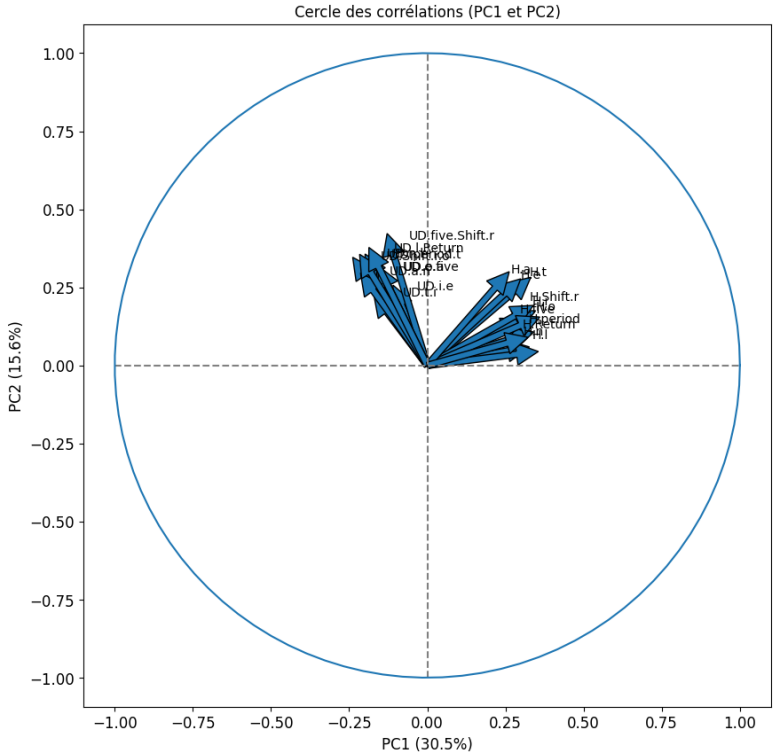

FIGURE 6 – Cercle des corrélations dans le plan formé par les deux premières composantes principales (données standardisées)

Dans le cas des variables uniquement centrées, on observe des flèches de longueurs et de directions variées, ce qui permet d'identifier les variables bien expliquées par les composantes principales ainsi que certaines corrélations. En revanche, pour les données standardisées, toutes les flèches ont une taille similaire et se répartissent uniquement selon deux directions principales. On remarque alors que les instants correspondant à une même action (enfoncement ou relâchement d'une touche) sont corrélés entre eux, tandis que ces deux types d'actions sont totalement décorrélés. Ce second cercle perd ainsi beaucoup d'informations par rapport au premier. C'est pourquoi dans la suite de cette étude on considérera les données uniquement centrées.

2 La classification pour la cyber-sécurité

2.1 Préambule

Dans le cadre de la cyber-sécurité, il est important de pouvoir détecter les comportements frauduleux. En effet, si une personne souhaite accéder à des infor-

mations confidentielles en usurpant l'identité de quelqu'un possédant ces informations, il lui suffit de connaître son mot de passe. Il est donc primordial d'ajouter des sécurités supplémentaires lors de l'authentification afin de rendre l'accès plus difficile pour une personne qui n'est pas celle autorisée. Nous allons donc essayer de classifier notre jeu de données, par exemple en deux classes : l'une contenant l'unique sujet autorisé et l'autre contenant tous les autres sujets, les intrus. Nous souhaitons trouver des moyens de distinguer efficacement les deux classes pour réussir à savoir, compte tenu d'un essai, à quelle classe il appartient.

2.2 Seul contre tous

2.2.1 Analyse discriminante linéaire

Nous avons donc effectué une analyse discriminante linéaire (LDA) sur l'ensemble des essais du sujet autorisé, choisi aléatoirement grâce à notre numéro étudiant (le sujet numéro 27). Pour réaliser cette LDA, nous avons décidé d'entraîner notre modèle sur les 188 essais du sujet autorisé et sur 4 essais de chacun des 50 sujets intrus. En effet, cela teste 188 essais du sujet autorisé contre 200 essais des sujets intrus, ce qui permet d'avoir le même ordre de grandeur et empêche que le modèle soit constamment égal à 0 ou 1, car une des tailles d'échantillon prédomine l'autre. La réalisation numérique nous permet d'afficher la Figure 7, qui représente la matrice de confusion, ainsi que la Figure 8, qui représente le boxplot.

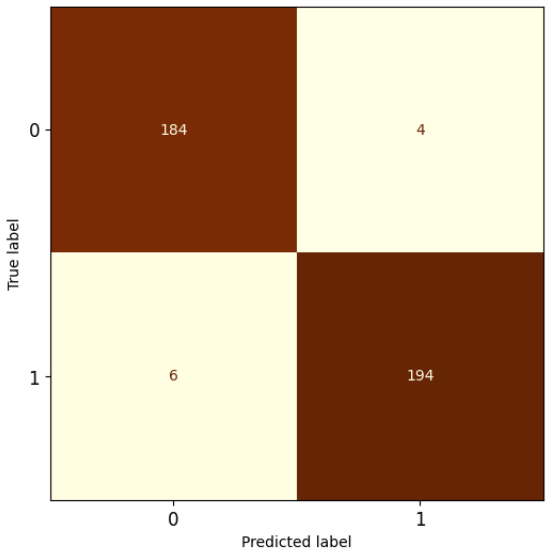

FIGURE 7 – Répartition des différents essais entre les deux classes

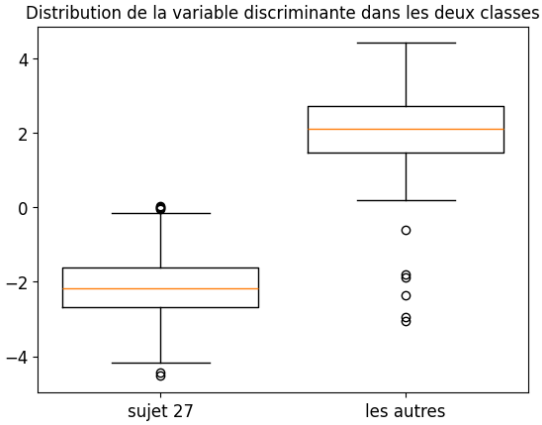

FIGURE 8 – Boxplot représentant la distribution de la variable discriminante dans les deux classes

La réalisation numérique de cette LDA nous donne un taux d'erreur d'environ 3%. Néanmoins, nous sommes obligés d'utiliser tous les essais du sujet autorisé pour entraîner le modèle, car sinon il n'y en aurait pas assez et nous ne pourrions donc pas tester le modèle sur de nouvelles valeurs. C'est pourquoi nous avons effectué une validation croisée à k plis (pour différentes valeurs de k) afin d'appuyer éventuellement nos résultats, ce qui nous affiche un score moyen de 96,6%. Cela nous garantit que cette méthode est très efficace pour différencier un individu d'un groupe et qu'il n'y a pas eu de phénomène de sur-apprentissage. Cette conclusion était aussi possible à l'aide du boxplot Figure 8 : en effet, les corps des bougies étant très éloignés et les mèches supérieures et inférieures des deux bougies ne se chevauchant pas, cela met en relief le fait que la LDA réussit à bien distinguer un individu particulier du reste du groupe.

2.2.2 Méthode de détection des anomalies

La source dont est issu notre jeu de données compare les performances d'algorithmes de détection d'anomalies. Par ailleurs, toujours dans le cadre de la cybersécurité, il peut être interessant de réussir à détecter si un essai d'un sujet semble être une anomalie car cela peut donner une alerte sur l'éventuelle possibilité que ce soit un intru qui soit à l'origine de cet essai. Nous avons donc décidé de mettre en œuvre une détection d'anomalies de la façon suivante : on considère la moitié des essais du sujet autorisé (on considérera encore le sujet 27) et on calcule la moyenne de chacune des variables associées à ses $188/2 = 94$ essais. Ensuite, pour détecter si un essai est une anomalie (c'est-à-dire si un essai n'est pas suffisamment "proche" de notre pivot), on introduit la métrique euclidienne pour définir une notion de distance entre un essai d'un sujet et cette moyenne. Il est important de relever

deux types d'erreurs pour évaluer les performances de cette méthode :

• L'erreur qui consiste à dire qu'un essai appartient à notre sujet autorisé alors qu'il appartient à un intrus

• L'erreur qui consiste à dire qu'un essai n'appartient pas à notre sujet autorisé alors qu'il lui appartient

Dans le cadre de la cyber-sécurité, on cherchera en priorité à réduire le taux de la première erreur : nous préférons avoir trop de signaux d'alertes plutôt que de potentiellement laisser passer un intrus. Pour déterminer ces taux d'erreurs, nous allons chercher graphiquement un seuil basé sur la distance entre les essais et la moyenne (on considère la norme du vecteur essai - vecteur moyenne des essais du sujet autorisé). Pour ce faire, nous allons considérer deux essais par sujet intrus afin de conserver le même nombre d'essais pour les deux classes. Nous avons représenté dans la Figure 9 le taux d'essais du sujet autorisé détecté comme des essais intrus en fonction du seuil (vrai négatif). Ensuite, nous avons aussi représenté dans la Figure 10 le taux d'essais des sujets intrus détectés comme des essais intrus en fonction du seuil.

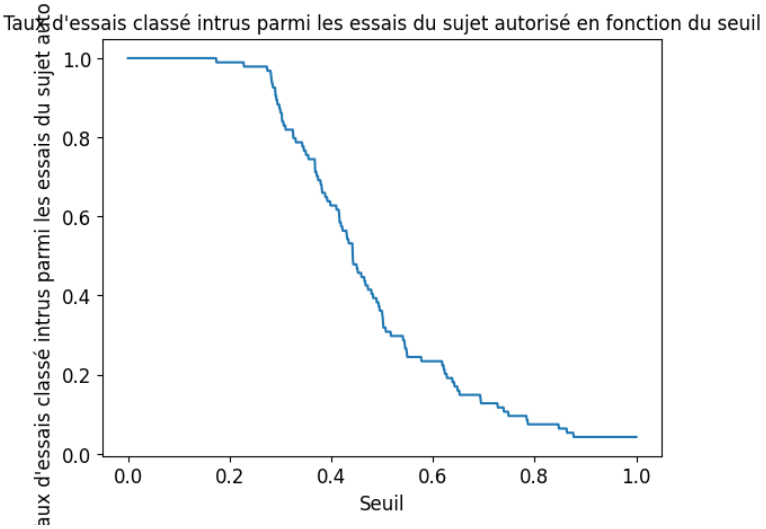

FIGURE 9 – Taux de vrai négatif en fonction du seuil

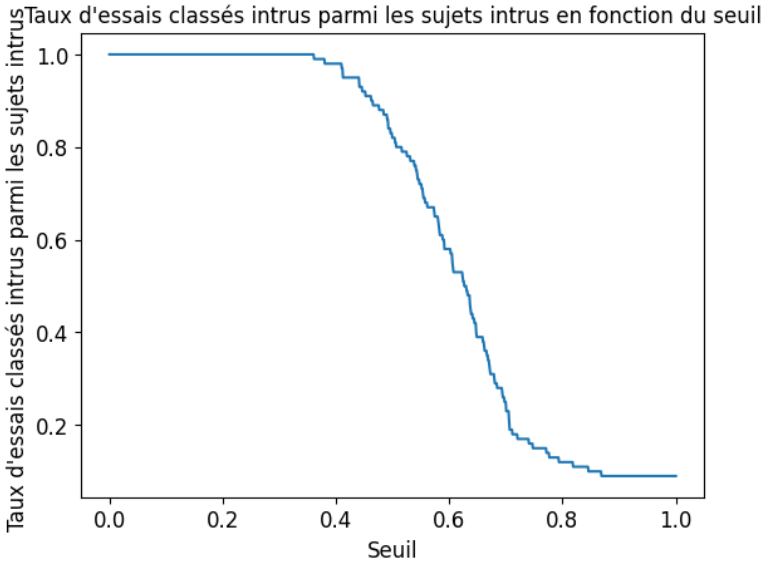

FIGURE 10 – taux d'essais des sujets intrus détectés comme des essais intrus en fonction du seuil

Nous avons ensuite numériquement essayer différentes valeurs de seuil entre 0,2 et 0,8 et nous avons finalement choisit 0,5 afin d'avoir le meilleur compromis entre une détection assez précise des intrus et un taux de fausses alertes (situation où le sujet autorisé est considéré comme intrus) le plus bas. Nous obtenons ainsi, après calcul avec ce seuil, un taux de fausses alertes de 0,42 et de non-détection des intrus de 0,12. Réduire à 0,05 la non-détection des intrus induit un taux de fausses alertes de plus de 0,5 et nous pouvons donc rester sceptiques quant à l'utilisation de cette méthode de détection d'anomalies.

Nous pouvons donc nous demander si nous ne devrions pas utiliser une autre métrique. Pour cela, nous utilisons la distance Cityblock (Manhattan). Comme précédement, nous obtenons les Figure 11 et Figure 12

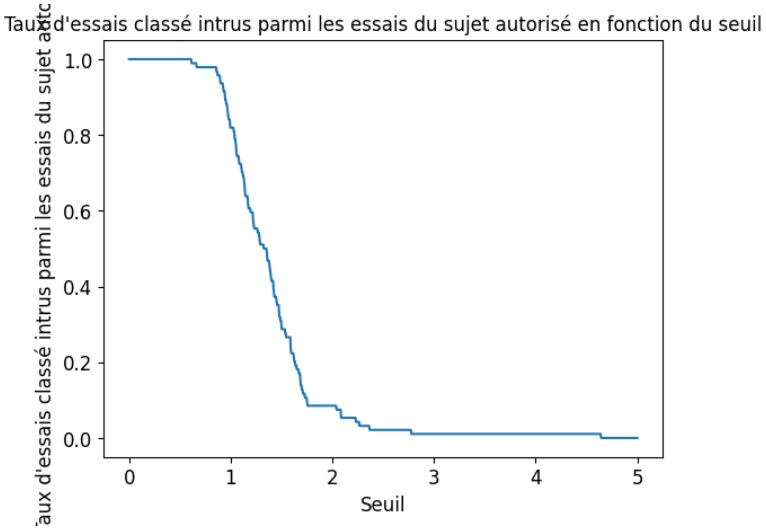

FIGURE 11 – Taux de vrai négatifs

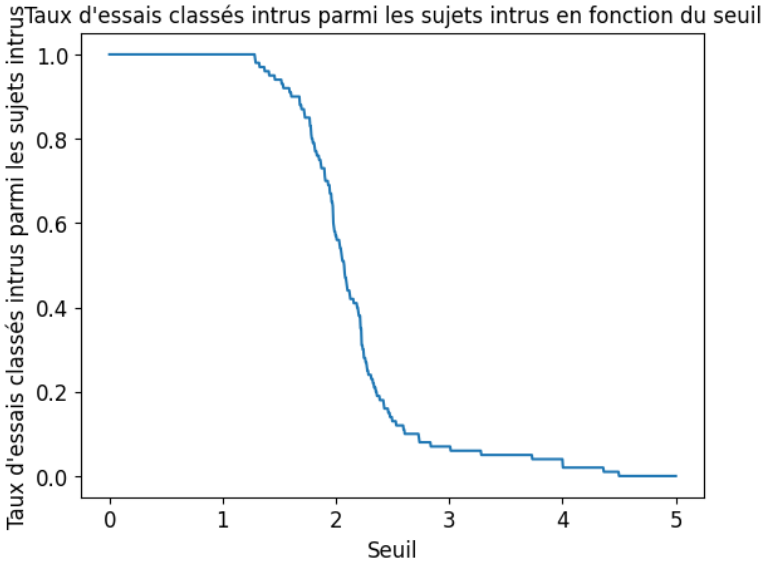

FIGURE 12 – taux d'essais des sujets intrus détectés comme des essais intrus en fonction du seuil

Numériquement, en faisant varier le seuil pour des valeurs comprises entre 1 et 4, on choisit 1,5 comme seuil et cela nous donne un taux de fausses alertes de 0,36 et de non-détection des intrus de 0,04. Ces résultats sont nettements plus intéressants que ceux que l'on a obtenu précédemment. Nous pouvions prétendre à de meilleurs résultats avant même d'avoir effectué la simulation car la première

métrique que nous avions considéré "mélangeait" tous les écarts alors qu'en considérant la distance Cityblock, comme nous l'avions mentionné dans l'introduction, nous prenons davantage en compte les écarts liées aux temps de changements de touche. Or, nous avions fait remarquer que ces temps précisement peuvent permettre de caractériser des sujets qui ont l'habitude de taper sur un clavier de ceux qui ont moins l'habitude.

Compte tenu de cette dernière remarque, on peut donc vouloir affiner l'étude en considérant la métrique qui ne fais la somme plus que des écarts en valeur absolue des coordonées paires (qui correspondent aux moments où les sujets recherchent la nouvelle touche). Cela nous permet d'obtenir les Figure 13 et Figure 14

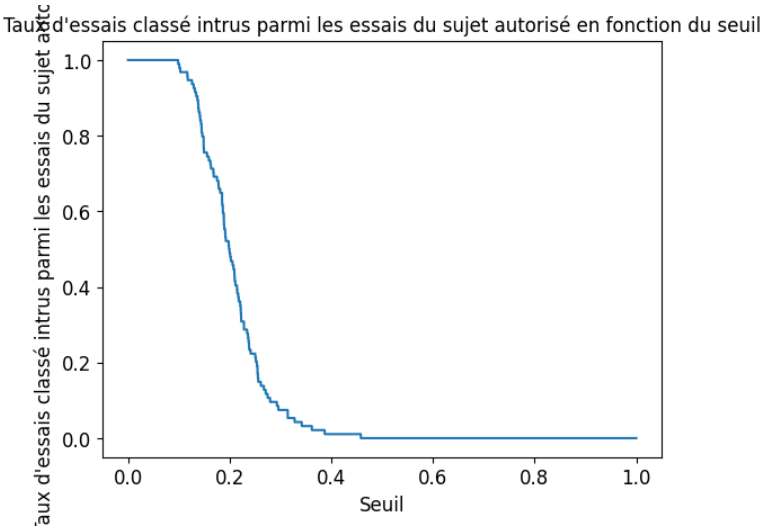

FIGURE 13 – le taux d'essais du sujet autorisé détecté comme des essais intrus en fonction du seuil

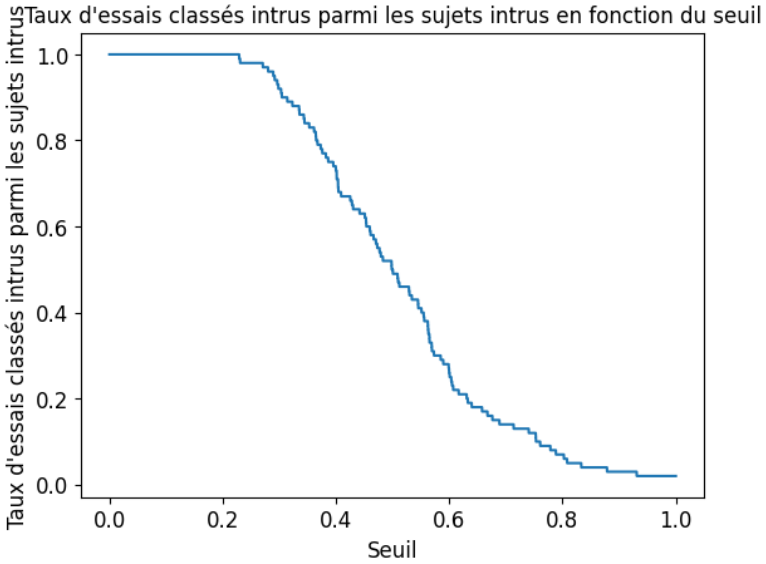

FIGURE 14 – taux d'essais des sujets intrus détectés comme des essais intrus en fonction du seuil

Après divers essais on choisit de fixer le seuil à 0,3 et nous obtenons encore de meilleurs résultats : en conservant un taux de non-détection des intrus à 0,04, nous avons réussit à baisser le taux de fausses-alertes à 0,1 ! Ces derniers résultats sont très encourageants vis à vis de notre problématique car cela laisse sous-entendre que l'on puisse éventuellement considérer d'autres métriques qui pourraient acroitre encore plus notre précision tout en conservant le confort du sujet autorisé (cela n'est pas toujours apprécié de devoir taper plusieurs fois son mot de passe alors que c'est vraiment nous...)

2.3 Caractérisation de tous les sujets

Une autre approche consiste à réaliser une LDA en prenant une classe par sujet. On se retrouve donc avec 51 classes. Ainsi, lorsqu'un nouveau sujet tape le mot de passe, on regarde s'il rentre dans l'une des classes autorisées. Si ce n'est pas le cas, on considère le sujet comme interdit d'accès.

De manière plus concrète, comme nous possédons relativement peu de données, nous avons utilisé l'ensemble des données fourni : les 188×51 relevés. Comme précédemment nous utilisons les données brutes (non standardisées). On possède de plus dans nos données le numéro de l'individu qui a tapé le mot de passe. On considère alors naturellement comme classes le numéro de chaque sujet. A l'aide de ces 188×51 relevés, nous avons procédé à une LDA dont le résultat est affiché ci-dessous (figure 15).

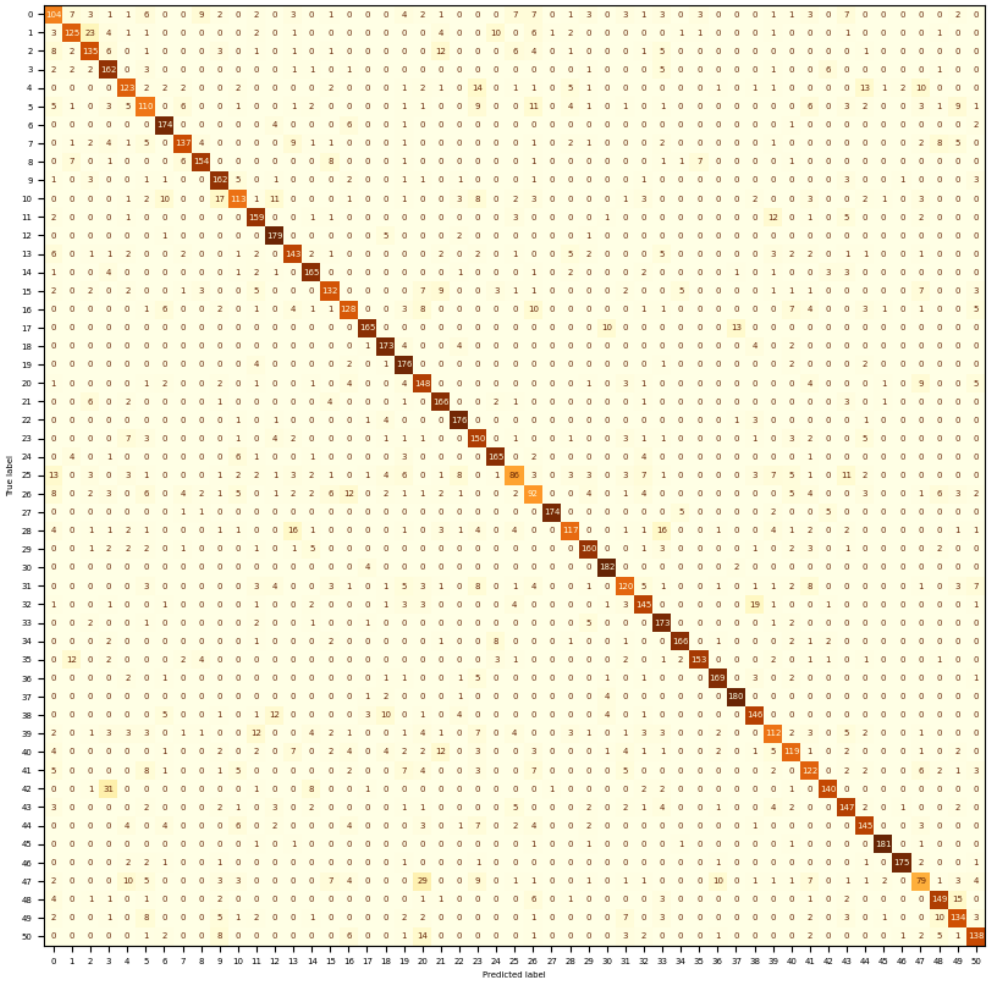

FIGURE 15 – LDA avec une classe par sujet

Le résultat observé semble satisfaisant. La majorité des relevés de mots de passe sont associés au bon sujet.

On peut essayer de quantifier cela plus précisément, par exemple en calculant un taux d'erreur empirique : le pourcentage d'attribution de classe erronée parmi l'ensemble des essais. On trouve un taux d'erreur empirique de 0.225, i.e un score d'entraînement $\approx 77\%$.

Comme nous avons utilisé l'ensemble des données, afin de quantifier la performance du modèle, on passe par une validation croisée. Pour le choix du nombre de plis, nous avons fonctionné de manière expérimentale en testant plusieurs valeurs. En prenant 15 plis ($cv = 15$), on obtient un bon compromis entre temps de calcul et résultats obtenus. Pour $cv = 15$ la validation croisée marche bien : on observe un coefficient de variation de 3.2% ($= \frac{\sigma(scores)}{E(scores)}$) ce qui dénote une grande stabilité

dans les scores obtenus. Stabilité confirmée par la faible amplitude des bornes des scores : 13.6 % de la moyenne.

La validation croisée confirme ce qui était suspecté : notre modèle est efficace. On trouve un score moyen de 77%, ce qui est très largement supérieur au score moyen de 2% d'un modèle naïf où l'on attribue systématiquement la classe majoritaire. On remarque aussi que ce score moyen est identique au score d'entraînement, ce qui montre qu'il y a peu de surapprentissage.

Caractériser l'ensemble des sujets à l'aide d'une LDA est donc une bonne option pour garantir la sécurité de données sans trop impacter l'expérience utilisateur.

3 La classification pour aider les utilisateurs

3.1 Préambule

Dans la partie précédente, nous avons vu qu'il était possible de séparer les individus en les regroupant en deux classes ou bien en attribuant une nouvelle classe pour chaque individu. Cependant, cette classification a été faite de façon supervisée : nous décidions combien nous souhaitions de classes ; mais il est légitime de se demander en combien de classes un modèle pourrait trier nos sujets, sans que l'on intervienne. Nous passons donc du côté de la classification non supervisée. Pour cela, nous allons effectuer une classification ascendante hiérarchique (CAH) afin de pouvoir par la suite utiliser la méthode des k-means. Cependant, nous allons tout d'abord effectuer une méthode de classification naïve qui consiste à regrouper les individus en fonction de la moyenne de leurs essais. Bien que nous soyons conscients que l'on perd de l'information, nous avons tenu à faire cette étude malgré tout afin de pouvoir confronter les résultats à ceux du CAH ainsi que de la méthode k-means.

3.2 Méthode naïve

Pour notre méthode naïve, nous considérons deux matrices différentes :

• La matrice du jeu de données sur laquelle nous avons effectué une ACP dans la partie Analyse en composante principale. Cette matrice de taille $(51\times188) \times 21$, une fois que l'on lui a effectué une ACP, est de nouveau une matrice de taille $(51\times188) \times 21$ dont les coefficients représentent les coordonnées des différents essais des différents sujets selon les composantes principales.

• La matrice de taille 51×21 dont le coefficient $[i][j]$ représente la moyenne de la variable j des 188 essais du sujet i.

La première matrice nous permet donc, lorsque l'on calcule la moyenne des coefficients selon la première composante principale des 188 essais pour chaque sujet, de représenter la moyenne de l'information contenue dans la première composante principale pour chacun des sujets. Pour exploiter la deuxième matrice, nous lui avons appliqué une ACP. Nous pouvons désormais représenter la "moyenne" des sujets dans le plan formé par les deux premières composantes principales. Ces deux résultats donnent respectivement lieu aux images Figure 16 et Figure 17.

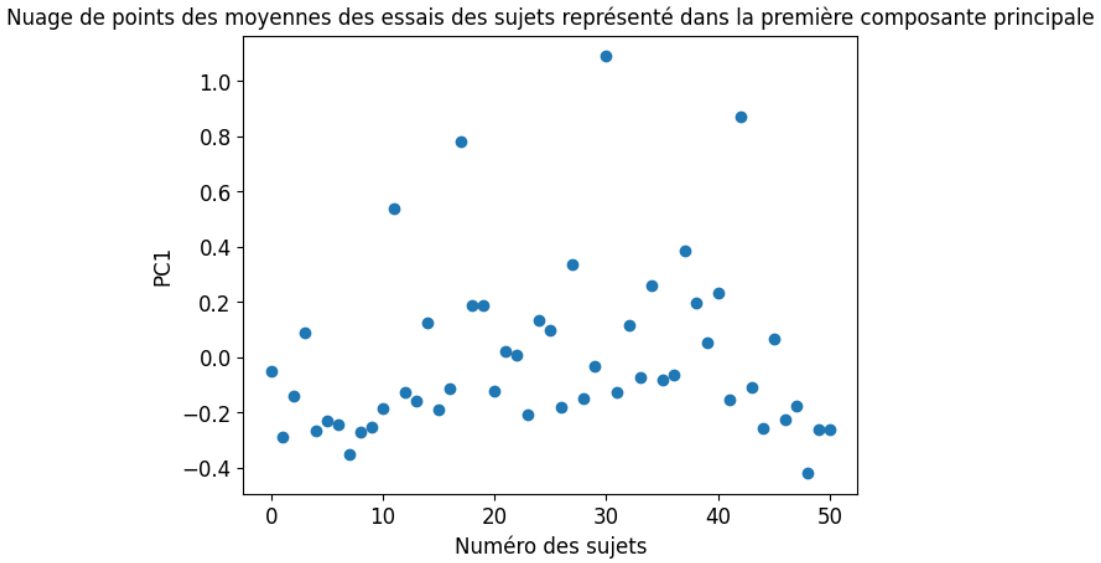

FIGURE 16 – Moyenne des essais des sujets représentée dans la première composante principale

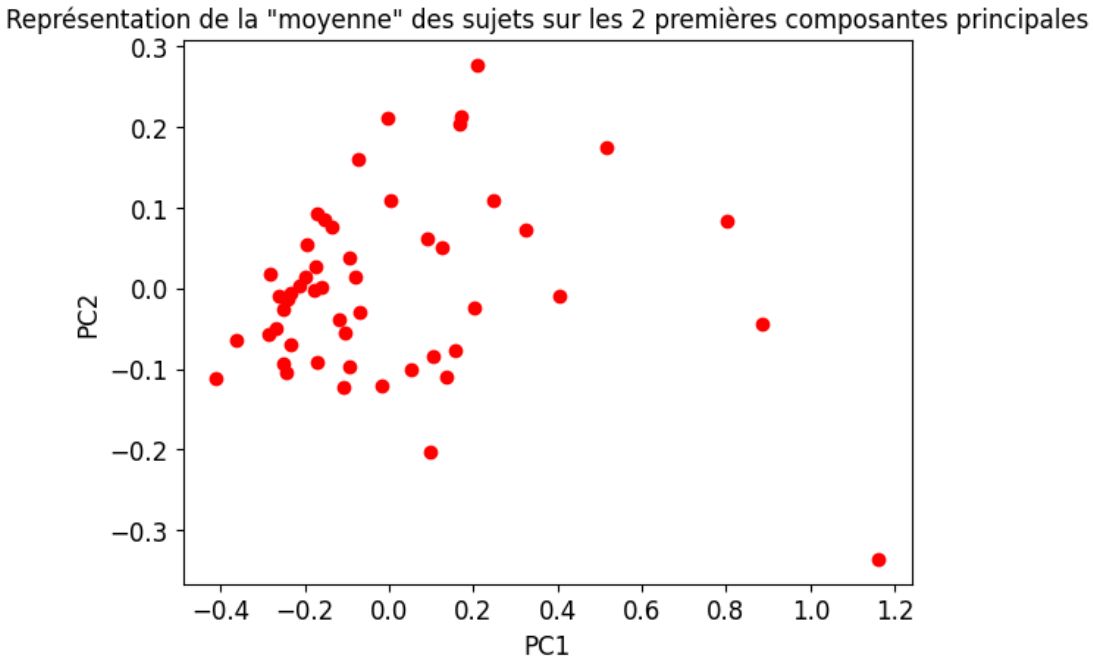

FIGURE 17 – Moyenne des essais des sujets représentée dans la première composante principale

Nous remarquons à travers ces deux figures que trois sujets se démarquent du reste du groupe. Il est donc normal d'espérer que ces sujets se retrouvent dans une même classe à part dans la suite de notre étude. Ainsi, bien que cette méthode ne soit pas optimale pour regrouper les individus, elle nous permet tout de même d'avoir certaines attentes et d'appréhender un peu mieux notre jeu de données.

3.3 Classification ascendante hiérarchique

Afin de pouvoir utiliser la méthode k-means pour comparer les différents résultats, il est nécessaire de déterminer le nombre optimal de groupes à former parmi nos sujets. C'est pourquoi nous allons réaliser une classification ascendante hiérarchique (CAH), dans l'espoir de trouver ce nombre optimal. Nous avons effectué la CAH en utilisant les métriques suivantes :

— Euclidean
— Cityblock (Manhattan)
— Cosine
— Precomputed

Ainsi que pour les méthodes de liaison suivantes :

— Single

— Complete
— Median
— Average
— Weighted
— Centroid
— Ward

Les résultats croisés pour les métriques Cityblock et Euclidean, ainsi que pour l'ensemble des méthodes de liaison, sont présentés dans la Figure 18.

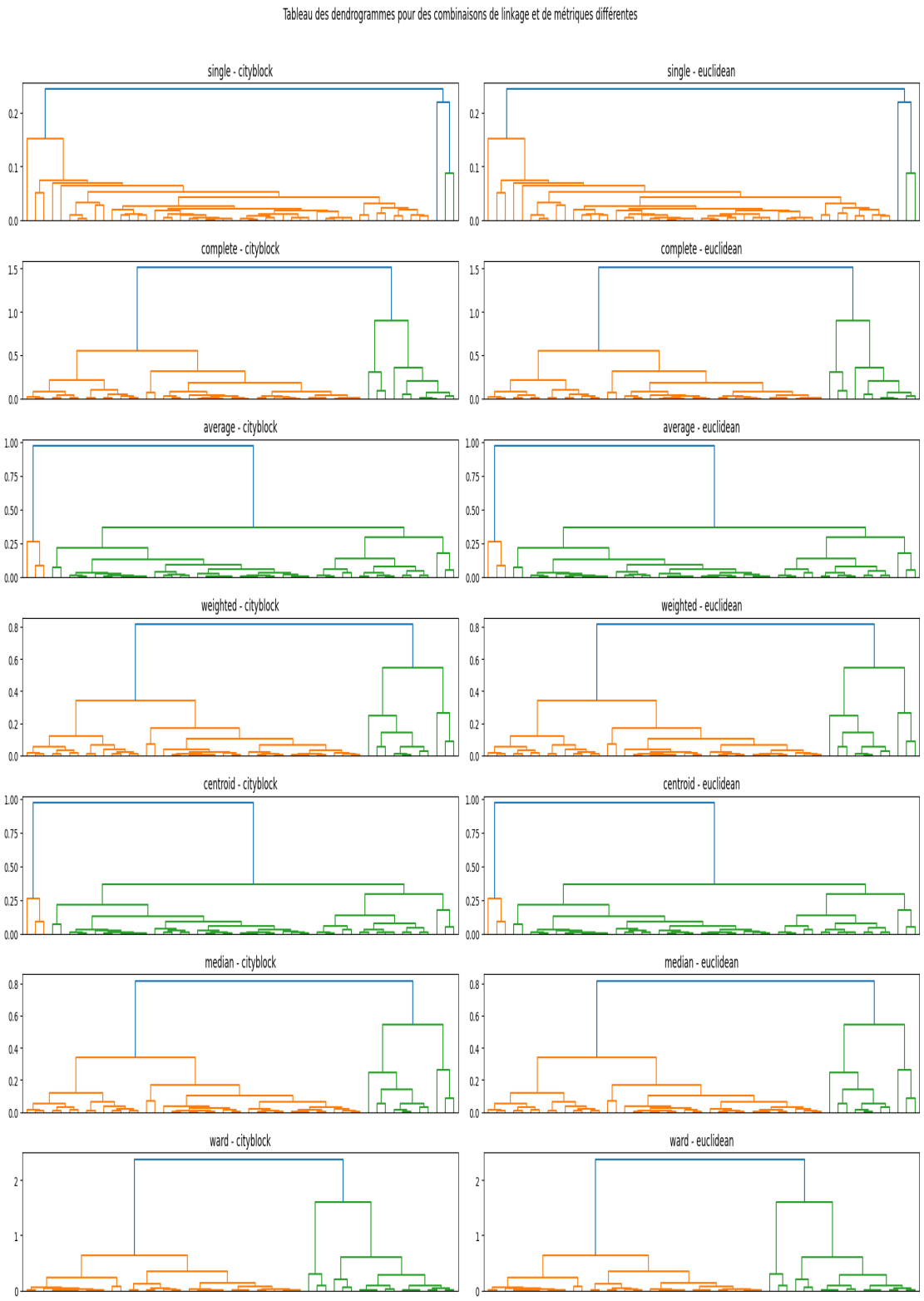

FIGURE 18 – Résultats croisés pour les métriques Cityblock/Euclidean et pour l'ensemble des méthodes de liaison

Il semble que la métrique Euclidean et la méthode de liaison Ward soient plus appropriées pour notre étude (d'un point de vue théorique on souhaite maximiser l'inertie inter-classe) d'autant plus que c'est la méthode la plus courante. Ce sont donc ces paramètres que nous conserverons par la suite, et nous affichons dans la Figure 19 le dendrogramme associé.

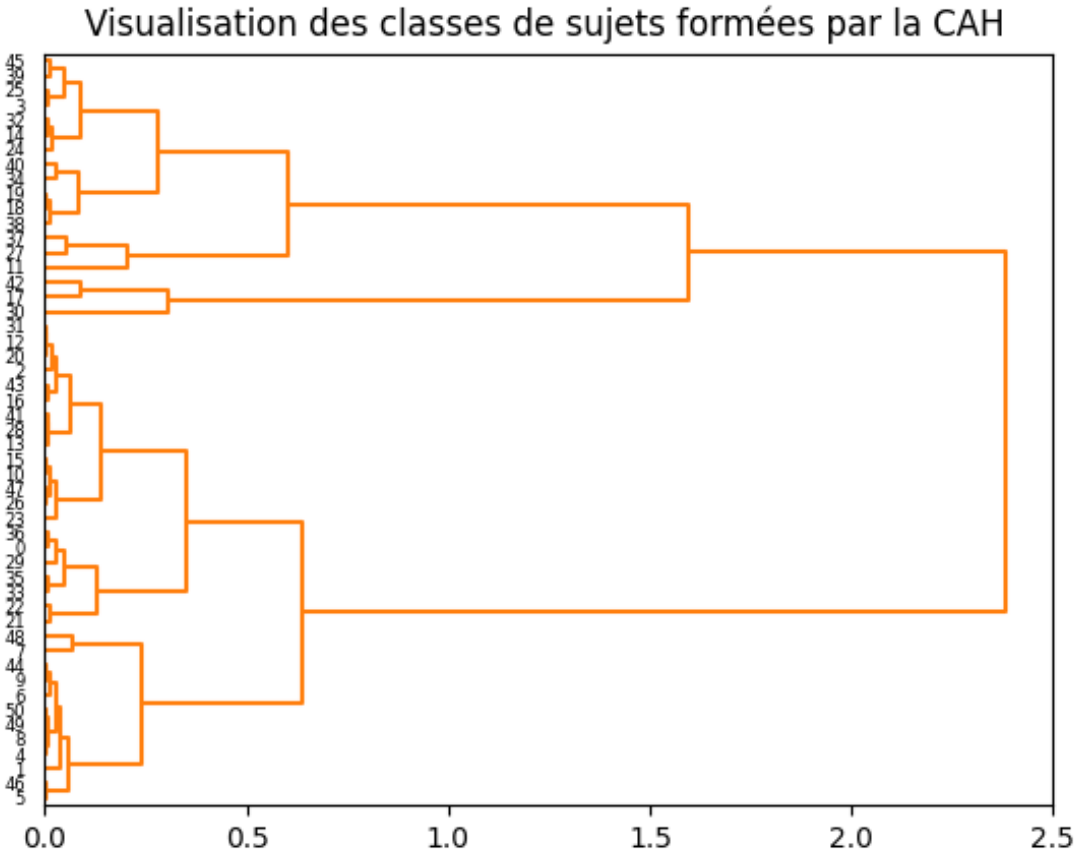

FIGURE 19 – Dendrogramme associé à la métrique Euclidean et à la méthode de liaison Ward

Il faut désormais déterminer le seuil pour la distance Euclidienne qui permettra de classer les sujets en différents groupes. Sachant que la longueur des branches du dendrogramme quantifie la perte d'information (plus une branche est longue, plus on perd d'information), nous souhaitons, avec la Figure 19, répartir les sujets en 4 ou 3 classes. Numériquement, cela représente respectivement un seuil de 0,62 et de 1. Nous affichons dans les Figure 20 et Figure 21 les dendrogrammes mettant en couleur les classes respectives pour ces deux classifications.

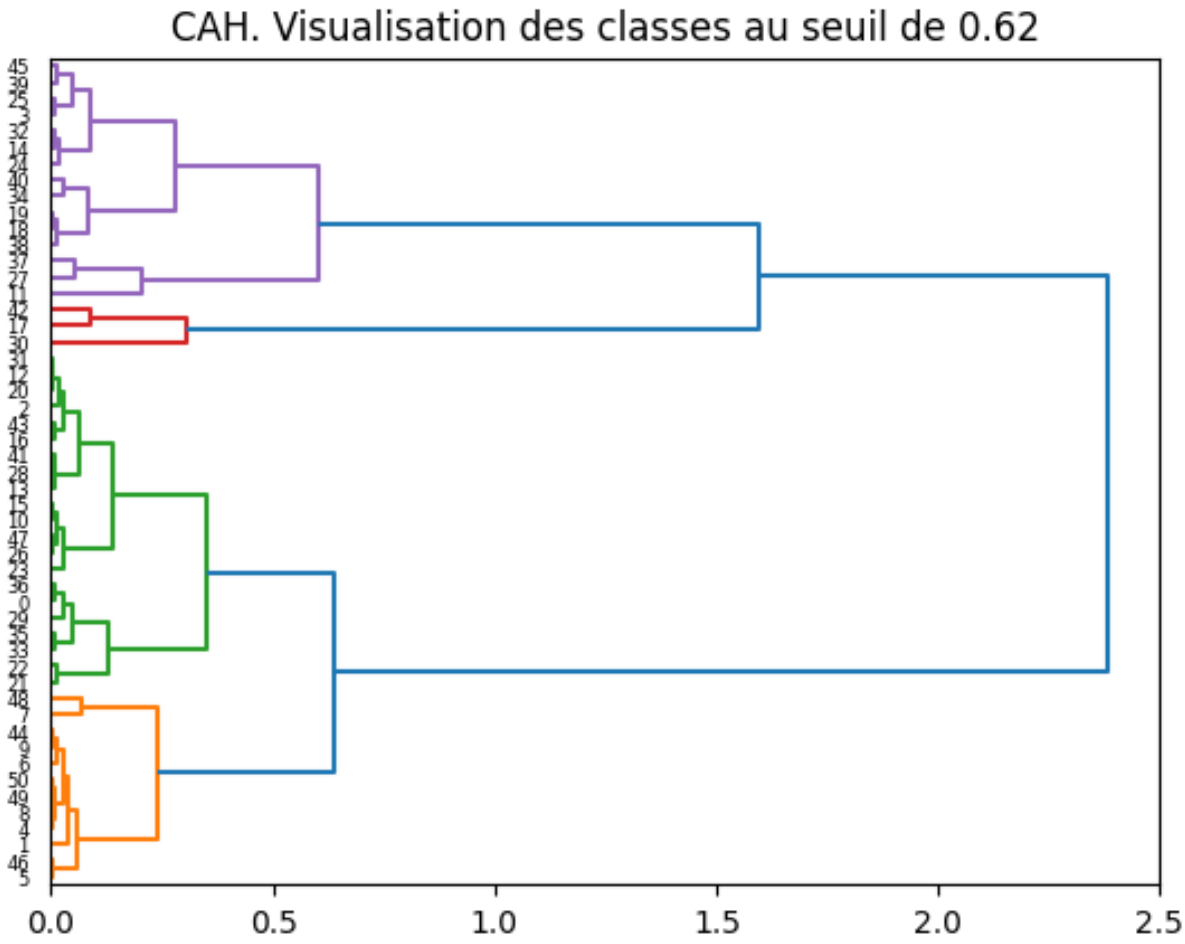

FIGURE 20 – Dendrogramme représentant les 4 classes par couleur

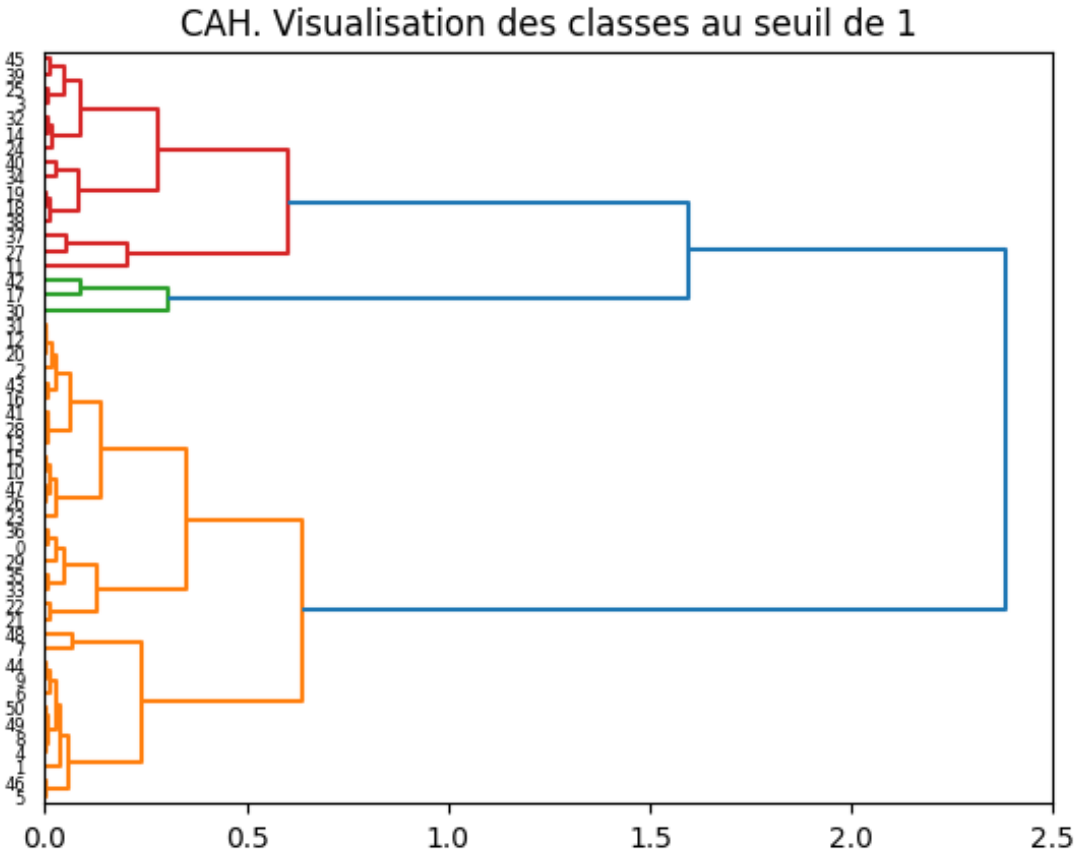

FIGURE 21 – Dendrogramme représentant les 3 classes par couleur

Nous pouvons également afficher les numéros des sujets appartenant à chacune de ces classes, comme le montre les tableaux 2 et 3.

Classe	**Sujets**
1	1, 4, 5, 6, 7, 8, 9, 44, 46, 48, 49, 50
2	0, 2, 10, 12, 13, 15, 16, 20, 21, 22, 23, 26, 28, 29, 31, 33, 35, 36, 41, 43, 47
3	17, 30, 42
4	3, 11, 14, 18, 19, 24, 25, 27, 32, 34, 37, 38, 39, 40, 45

TABLE 2 – Les 4 classes données par la CAH.

Classe	Sujets
1	0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16, 20, 21, 22, 23, 26, 28, 29, 31, 33, 35, 36, 41, 43, 44, 46, 47, 48, 49, 50
2	17, 30, 42
3	3, 11, 14, 18, 19, 24, 25, 27, 32, 34, 37, 38, 39, 40, 45

TABLE 3 – Les 3 classes données par la CAH.

On constate que, dans les deux cas, une classe est formée des trois sujets qui se distinguent complètement du groupe. Pour mieux visualiser ces résultats, nous pouvons représenter cette classification dans le plan factoriel formé par les deux premières composantes principales de notre matrice de données. Nous obtenons ainsi les Figures 22 et Figure 23, qui mettent en évidence ces 4 et 3 classes de sujets.

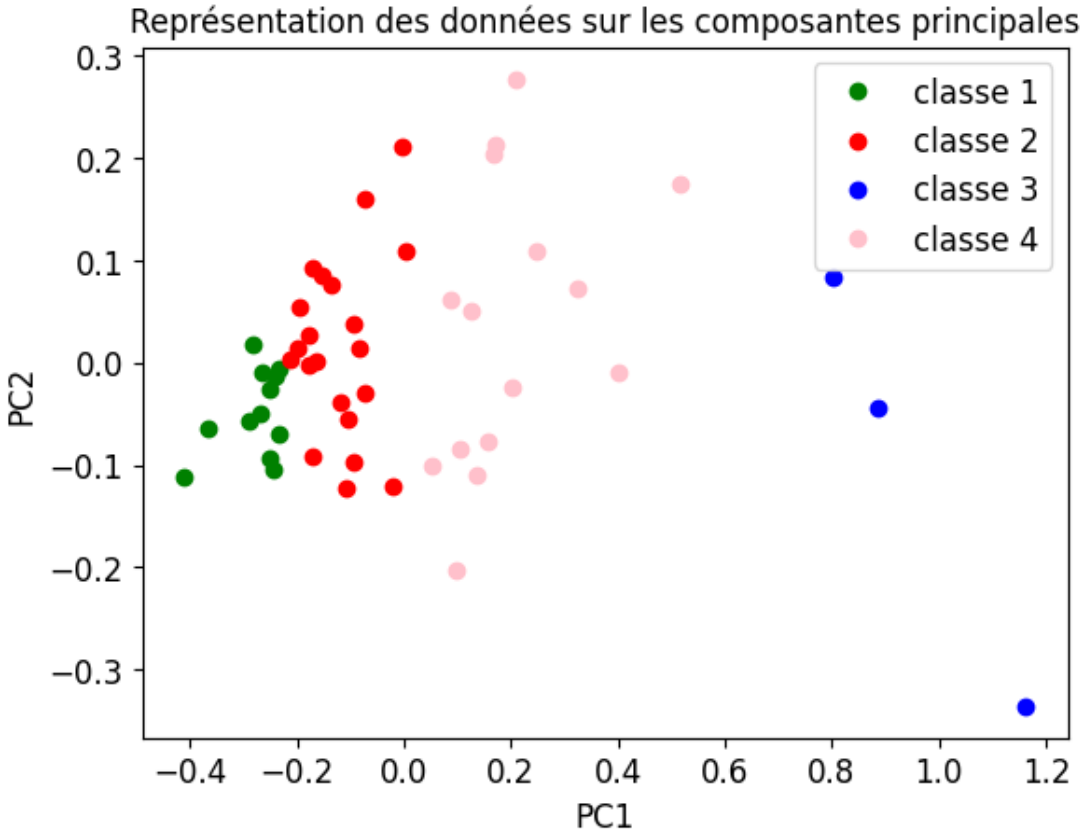

FIGURE 22 – Nuage de points représentant les 4 classes obtenues avec la CAH

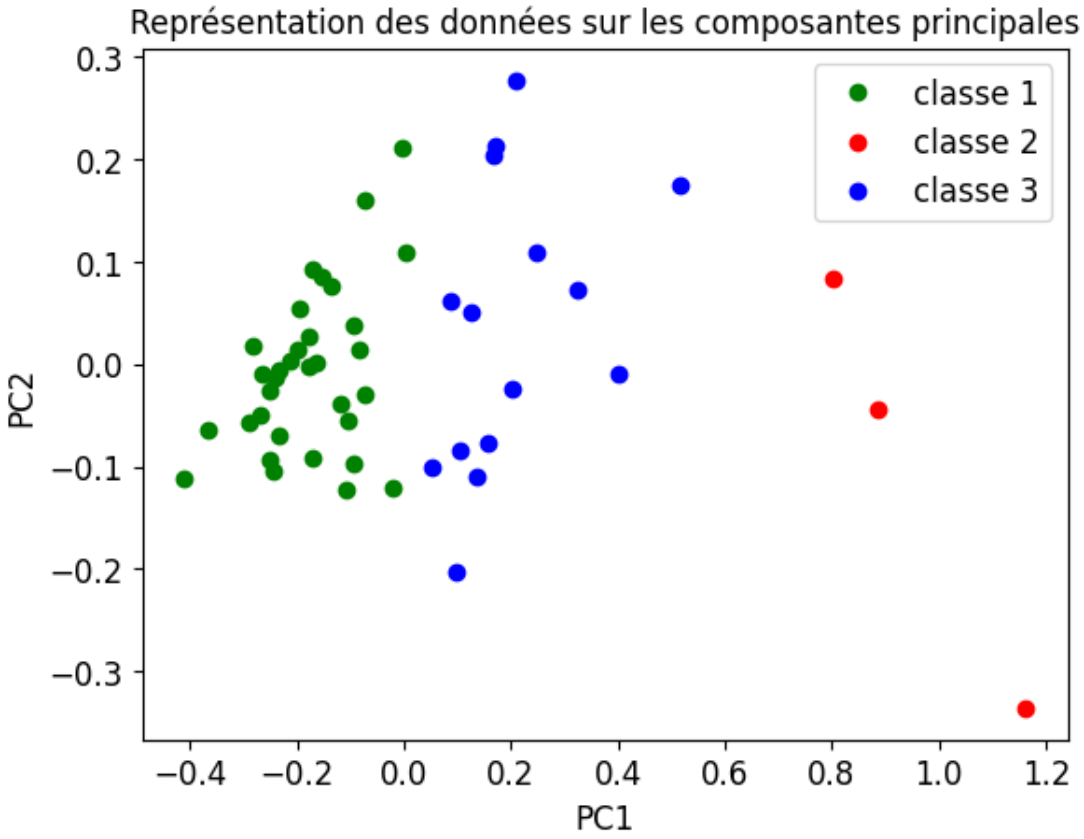

FIGURE 23 – Nuage de points représentant les 3 classes obtenues avec la CAH

3.4 Méthode k-means

3.4.1 Réalisation

Maintenant que l'on sait que l'on peut classifier nos individus en 4 ou 3 groupes, nous pouvons appliquer la méthode des k-means à k = 4 et 3. Nous obtenons les classes respectives qui sont indiqués dans les tableaux 4 et 5.

Classe	**Sujets**
1	3, 14, 18, 19, 21, 22, 24, 25, 27, 32, 34, 37, 38, 39, 40, 45
2	11, 17, 42
3	0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16, 20, 23, 26, 28, 29, 31, 33, 35, 36, 41, 43, 44, 46, 47, 48, 49, 50
4	30

TABLE 4 – Les 4 classes données par la méthode k-means.

Classe	Sujets
1	0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16, 20, 23, 26, 28, 29, 31, 33, 35, 36, 41, 43, 44, 46, 47, 48, 49, 50
2	3, 11, 14, 18, 19, 21, 22, 24, 25, 27, 32, 34, 37, 38, 39, 40, 45
3	17, 30, 42

TABLE 5 – Les 3 classes données par la méthode k-means.

3.4.2 Interprétation des résultats

On constate que dans la classification en 4 classes, la méthode k-mean nous isole totalement un individu ce qui ne correspond pas à nos observations. On peut donc retirer cette hypothèse de notre étude et on se concentre alors sur la séparation de nos sujets en 3 classes. On peut tout d'abord affirmer que le résultat de la méthode k-means sont cohérents avec ceux de la CAH et sont pertinents à la vue des observations que nous avons relevées. Ensuite, nous pouvons interpréter une telle classification comme la reconnaissance de groupes de niveau parmi les sujets. En effet, en ayant remarqué que des moyennes de temps plus élevés correspondent à des coordonnées selon la première composantes principales plus élevées, on peut réinterpréter notre classification (voir Figure 23) comme cela est indiqué dans le tableau 6 :

Classe	Sujets
1	Confirmé
2	Faible
3	Moyen

TABLE 6 – Niveau des classes de sujets formé par l'ACH et la méthode k-means

Maintenant, pour tester la validité de ces 3 classes, nous avons effectué une LDA ainsi qu'une validation croisée pour obtenir un taux d'erreur ainsi qu'un taux moyen de prédiction juste lors d'une telle classification. Pour la validation croisée, nous avons considéré 10 plis, cela permet d'avoir suffisamment de plis pour avoir une validation croisée pertinente sans pour autant faire de l'isolement de situation. Nous obtenons ainsi la Figure 24

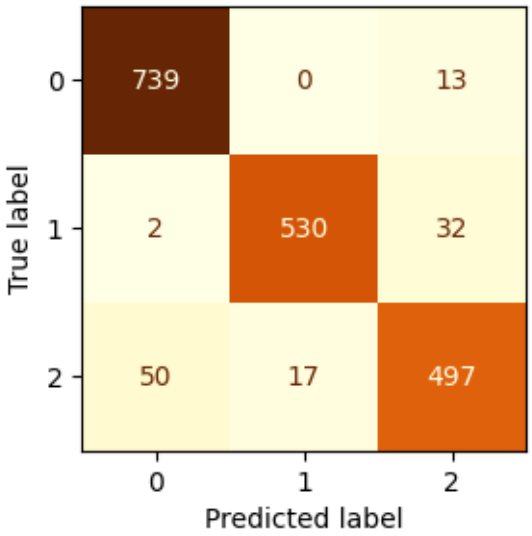

FIGURE 24 – Répartition des différents essais entre les 3 classes

Pour 10 plis, on obtient un taux d'erreur d'environ 0,06. C'est très proche de celui de la LDA qui est de 0,08. Cela nous permet d'affirmer d'une part que notre classification est pertinente et efficace et d'autre part que notre modèle n'est pas surentraîné. Pour visualiser comment sont prédites les données par notre modèle, nous avons tracé les nuages de points 25 26 27 des prédictions dans le plan formé par les deux premières composantes de la LDA en mettant au premier plan à chaque fois une classe différente afin de mettre en lumière où est-ce que notre modèle s'est trompé. Cela correspond à une représentation sous forme de nuage de points de la Figure 24

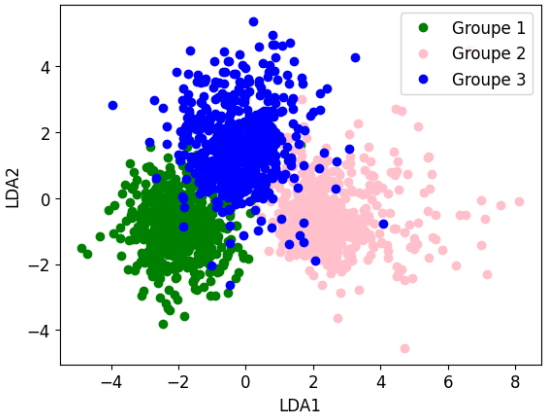

FIGURE 25 – Visualisation de la projection du groupe 3

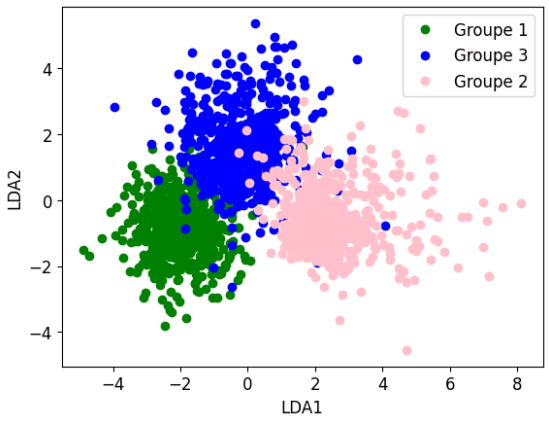

FIGURE 26 – Visualisation de la projection du groupe 2

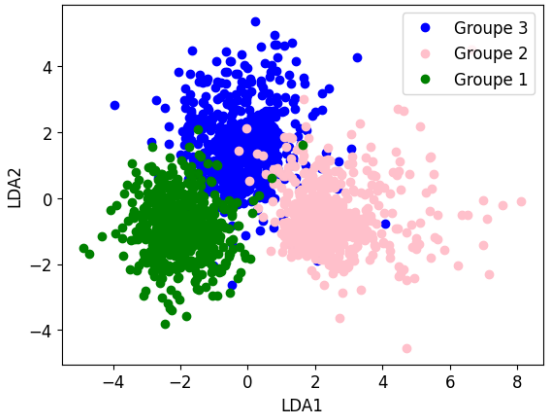

FIGURE 27 – Visualisation de la projection du groupe 1

4 Conclusion

En conclusion, ce projet nous a permis d'appliquer les résultats des trois premiers chapitres du cours à un jeu de données de frappe de mot de passe, en utilisant des techniques telles que l'Analyse en Composantes Principales (ACP) et la Classification Ascendante Hiérarchique (CAH). À travers ces analyses, nous avons non seulement pu identifier des regroupements naturels de sujets, mais aussi extraire des patterns significatifs dans la manière dont les individus frappent leur mot de passe.

Les résultats obtenus ouvrent des perspectives intéressantes pour des applications pratiques dans le domaine de la cybersécurité, notamment pour renforcer la fiabilité des systèmes d'authentification. En effet, comprendre les comportements

de frappe permettrait d'améliorer la détection des tentatives de fraude ou de compromission, en intégrant des critères biométriques basés sur la manière de taper plutôt que de se baser uniquement sur les mots de passe eux-mêmes.

De plus, ces résultats pourraient également être utilisés pour améliorer l'interface utilisateur des systèmes de saisie de mot de passe, en optimisant l'expérience utilisateur en fonction de la manière dont les utilisateurs interagissent avec le clavier.

Pour aller plus loin, il serait pertinent de recenser d'autres données, telles que le taux d'erreur dans la saisie des mots de passe (par exemple, la fréquence d'utilisation de la touche `Suppr` pour supprimer des caractères), ou même la pression appliquée sur les touches, afin de détecter des signes de stress ou de colère. De plus, un plus grand nombre de données permettrait d'augmenter la fiabilité des résultats présentées dans cette étude car comme nous avons pu le voir, dans certains cas il pouvait être compliqué d'évaluer nos modèles sur des nouvelles données. Ces informations supplémentaires pourraient permettre de renforcer davantage la personnalisation et la sécurité des systèmes d'authentification en ligne.