La classification croisée appliquée aux données textuelles usuelles

Mohamed BEN HAMDOUNE, Lucas ISCOVICI

Paris, France

Université de Paris Descartes

Abstract

La classification est une méthode d'apprentissage automatique populaire couramment utilisée dans l'analyse de texte exploratoire. De nombreuses méthodes de classification ont déjà été proposées et de nombreuses se concentrent sur la classification en une seul dimension.
En pratique, il est souvent souhaitable de co-classifier simultanément des documents et des mots en exploitant la co-occurrence parmi eux. Il a été démontré que la mise en co-cluster est plus efficace que la mise en cluster sur un seul dimension dans de nombreuses applications.

Keywords: Visualisation, Classification Croisée, Fouille de texte, Modèle des blocs latents, Classification

Contents

1. Introduction

Un thème de recherche standard pour la classification de texte est la création de représentations compactes de l'espace de fonctions et la découverte de relations complexes qui existent entre les caractéristiques, les documents et les classes. Dans cette optique, un important domaine de recherche où le regroupement est utilisé pour faciliter la classification du texte est la réduction de dimensionnalité.
Le clustering est utilisé comme méthode d'extraction et / ou de compression d'entités: les entités sont classées en groupes en fonction de critères de classification sélectionnés. Les méthodes de clustering des fonctionnalités créent de nouveaux espaces événementiels de taille réduite en rejoindre des fonctionnalités similaires dans des groupes.
En règle générale, les paramètres du cluster devenir la moyenne pondérée des paramètres de ses caractéristiques constitutives. Dans ce projet de co-clustering (classification croisée), nous aborderons le sujet en deux parties.

1. Données de messagerie électronique (Spam).
2. Choix de 4 documents ().

Il s'agit d'un volume de données textuelles important et nous utiliserons beaucoup de méthodes comme ACP, AFC, MDS, et nous utiliserons de la classification non supervisée (K-means, CAH) puis aussi la NMF. Ensuite nous évaluerons l'impact d'une normalisation et les conséquences sur les résultats.
Ensuite nous serons amenés à traiter des données sparses et le co-clustering est une méthode appropriée à notre problématique.

2. Analyse exploratoire des données

2.1. SPAM

Dans ce jeu données, il y a 58 variables dont une colonne correspond à la classe cible. Nous avons deux possibilités qui est soit le message est un spam ou non, nous sommes façe à un problème de classification binaire.

spam	make	address	all	X3d	CapLM	CapLsup	CapLtot
1	0.00	0.64	0.64	0	3.756	61	278
1	0.21	0.28	0.50	0	5.114	101	1028
1	0.06	0.00	0.71	0	9.821	485	2259
1	0.00	0.00	0.00	0	3.537	40	191
1	0.00	0.00	0.00	0	3.537	40	191
1	0.00	0.00	0.00	0	3.000	15	54

Figure 1: Exemple du jeu de données

Le concept de spam est divers (annonces de produits avec les sites Web en occurrence, newsletter, etc).
La collection de courriers électroniques autres que du courrier indésirable provient de courriers électroniques professionnels et personnels. Le mot "george" et l'indicatif régional "650" sont donc des indicateurs de non-spam. Celles-ci sont utiles lors de la création d'un filtre anti-spam personnalisé. Il faudrait soit masquer ces indicateurs non-spam, soit disposer d'un très grand nombre de non-spams pour générer un filtre anti-spam à usage général.

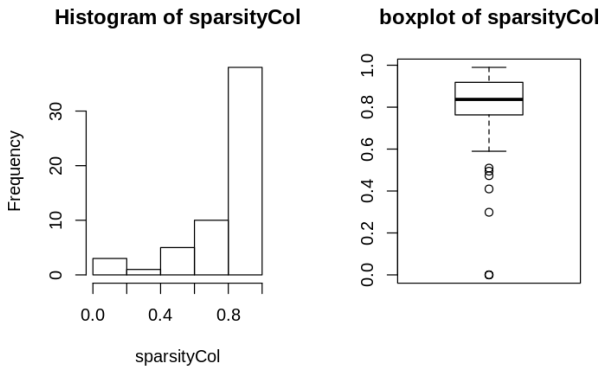

Figure 2: Étude sur la sparsité des colonnes *SPAM*

Il y'a beaucoup de variables creuses a plus de 50% (il y a 3 variables qui n'ont pas de 0, ce sont les variables "dénombrant le nombre de lettres majuscules").

On effectuera une transformation car il y a des écarts de valeurs et de variance élevé. La transformation que nous effectuerons est le *log*. Nous avons comme valeur maximal (1.514402) et ensuite la valeur minimal (0.04834942) puis une moyenne (0.2849189).

3. Approche quantitative

3.1. Analyse en composantes principales sans transformation ni normalisation

Nous examinons les valeurs propres pour déterminer le nombre de composantes principales à prendre en considération.

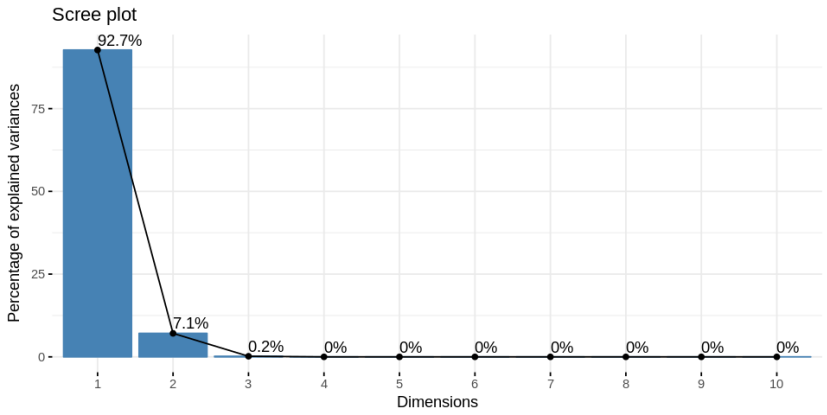

Figure 3: Pourcentage de variance expliqué par composantes principales

Nous avons 92.7% de l'information contenu dans la première composante principal, ensuite 99.8% avec la deuxième et finalement 100% en 3 composantes principales. L'ACP ne possède aucune problème a capté la variance dans notre matrice de variance-covariance.
Nous passons à la représentation graphique sur les deux premiers plan factoriel de l'ACP:

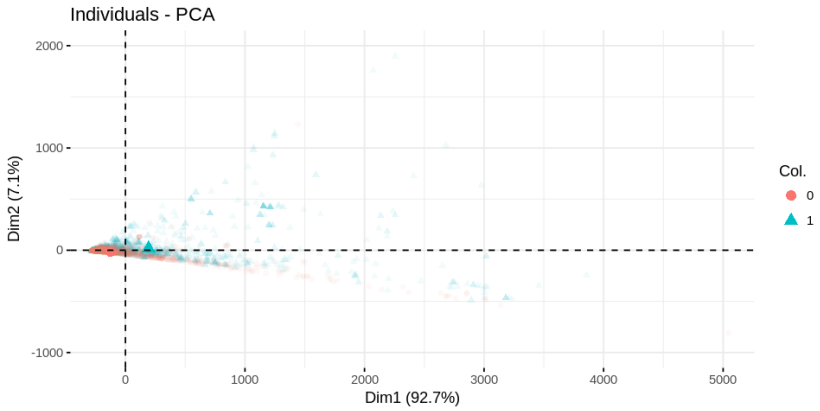

Figure 4: Individus sur les deux premiers plan factoriel

On ne semble pas dissocié les classes, on voit que le groupe 1, est très éparpillé sur l'axe 2. Nous allons dès à présent regarder si les groupes sont dissociable, avec une courbe de densité.

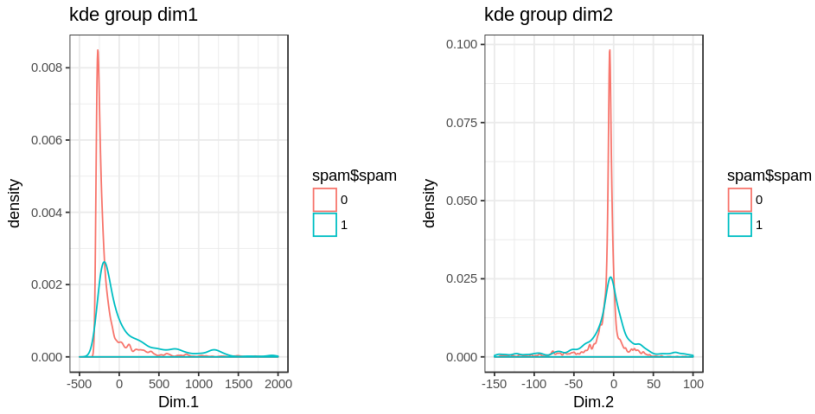

Figure 5: Densité des deux classes

On voit nettement que les groupes ne sont pas du tout séparés.

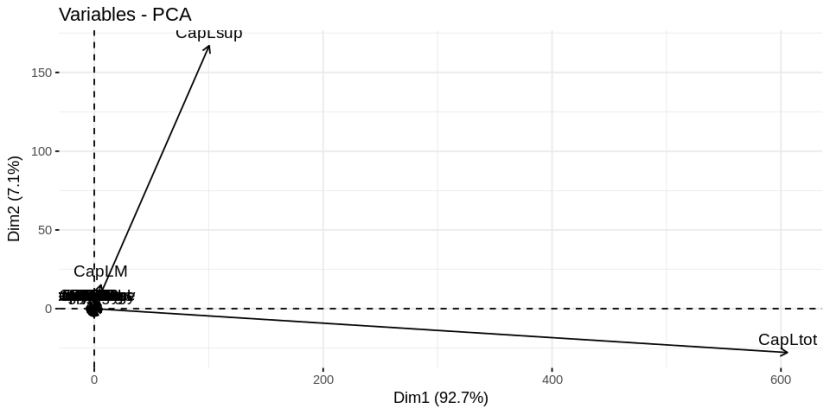

Figure 6: Variables sur le plan factoriel

Les données n'étant ni transformé et normalisé, on voit que les variables avec les valeurs les plus grandes, prennent le plus de variance, (elles prennent beaucoup de poids). Sur le cercle de corrélation on voit nettement que les variables *capLtot*, et *CapLsup* prennent le plus de variance. On peux en conclure que les individus qui sont éparpillé sur l'axe 1, ont de forte valeur sur *CapLtot* n et les individus qui ont une forte valeurs sur l'axe2, ont de forte valeur sur *CapLSup*.
Le groupe 1 (les spams) étant positifs sur l'axe 1 et 2 cela nous dit comme première explication que ces deux variable explique les spams. Pour rappel la moyenne des valeurs pour *CapLsup* et *capLtot* est: 283.29, 52.17, alors que le reste des variables, elles sont vers [0,1]. Cela montre l'importance de standardisé les variables, pour ne pas donnée trop de poids au variables qui ont de forte valeurs.

3.2. Analyse en composantes principales sans transformation mais normalisation

Nous examinons les valeurs propres pour déterminer le nombre de composantes principales à prendre en considération. Quand les données sont standardisés, on peux dire qu'on veux garder les composantes qui ont une valeur propre supérieur à 1, car cela indique que la composante principale (PC) concernée représente plus de variance par rapport à une seule variable d'origine.

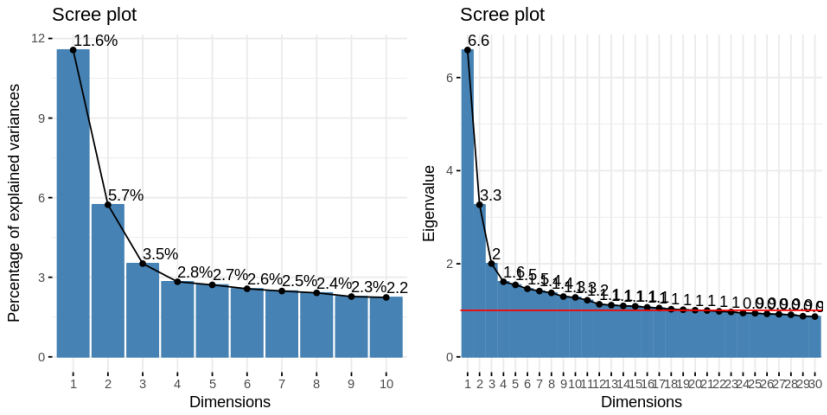

Figure 7: Pourcentage de variance expliqué par composantes principales et Valeur propres

On voit que la 1ère composante principale capte 11.6% de variation contenue dans le jeu de données La 2ème composantes principales, ne capte que 5.7% et la troisième 3.5%. Ainsi nous pourrions prendre en considération les trois premières composantes principales, sur le première plan factoriel (17.3%),et sur le plan 2:3 (9.2%), le plan 1:3 (15.1%) Si nous prenons comme critère le nombre de composantes principales qui ont un valeur propre strictement supérieur à 1, nous devrions prendre jusqu'à 16 composantes principales.

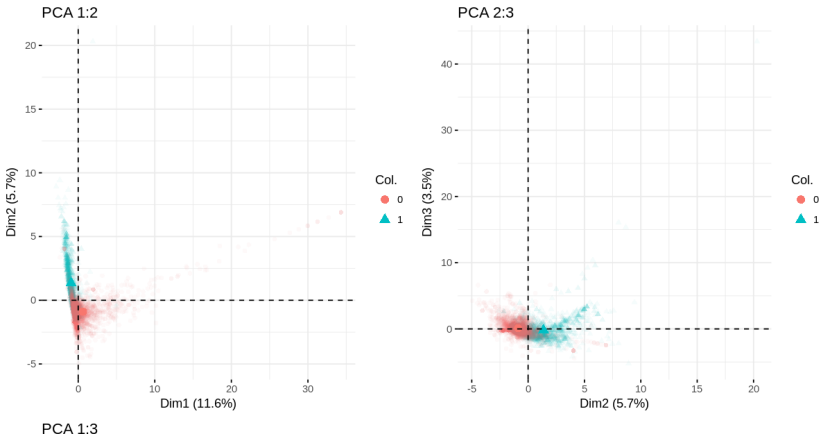

Figure 8: Plan factoriel 1:2 et 2:3

Sur le plan factoriel 1 et 2, Il semblerait que les classes soit assez bien discriminé sur l'axes 2. Sur l'axe 1e groupe 0, est très éparpillé. Ensuite sur le plan 2 et 3, les classes assez bien séparé sur l'axe 2, ce qui confirme les deux premiers plan factoriel. Finalement, l'axe 3 ne nous apporte pas grand chose.

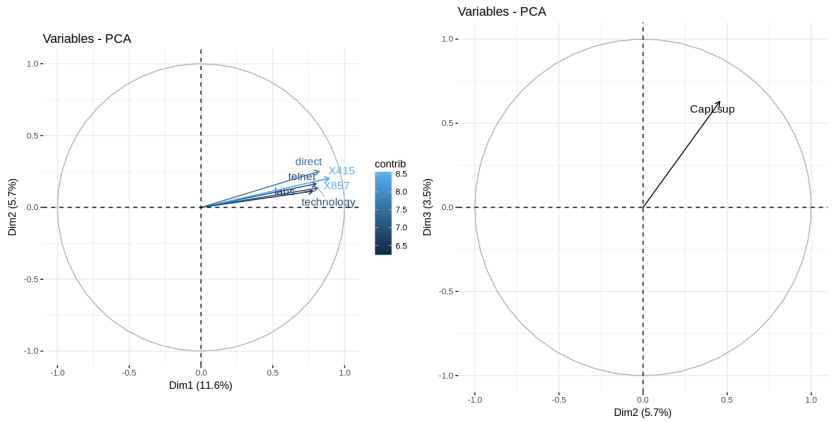

Figure 9: Cercle de corrélation sur les plan factoriel 1,2 et 2,3

On voit que les variables qui contribue le plus et qui sont bien projeté sur l'axe 1, sont *X415*, *X857*, *direct*, *telnet*, *technology*. Le groupe de *SPAM* est plutôt négatif sur l'axe 1, donc le groupe de *SPAM* est plutôt anti-corrélé à ces variables. Sur l'axe 2, toutes les variables sont mal projetés.

L'axe 1 est corrélé à la variable: *X415*, *X857*, *telnet*, *direct*, *X85*, *technology*. On peux conclure que ces variables sont anti-corrélé au spams *CapLsup* et *CapLTot* sont corréleé à l'axe 2. L'axe 2 étant l'axe qui sépare le mieux les données, nous pouvons remarqué, que dans les spams, il y aurait beaucoup de "yours", de CapLsup (c'est a dire le nombre maximum de capitales par mot), *Cdollar* (c'est à dire le caractère dollar), et *CapLTtot* (Nombre totale

de lettres capitales) et *CapLM* (Nombre moyen de capitales par mot).

3.3. Analyse en composantes principales avec transformation mais sans normalisation

Nous examinons les valeurs propres pour déterminer le nombre de composantes principales à prendre en considération

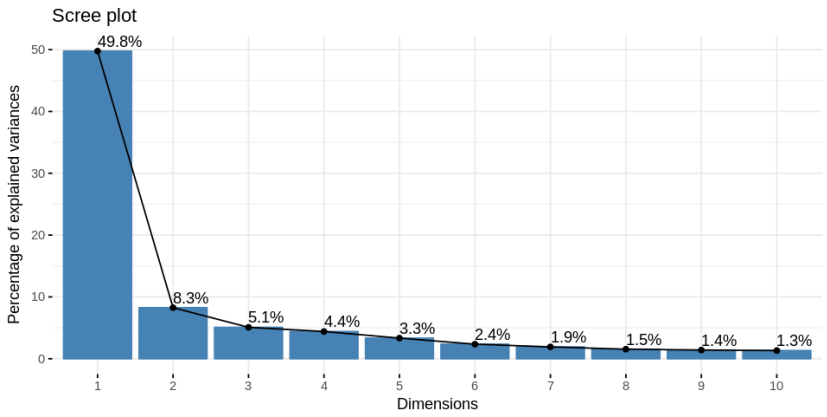

Figure 10: Pourcentage de variance expliqué par composantes principales

On voit que la 1er composante principale capte 49.8% de variation contenue dans le jeu de données La 2ème cp, ne capte que 8.3% et la troisième 5.1% Ainsi nous pourrions prendre en considération les trois premieres cp, sur le premiere plan factoriel (58.02%),et sur le plan 2:3 (13.34%), le plan 1:3 (54.84 %).

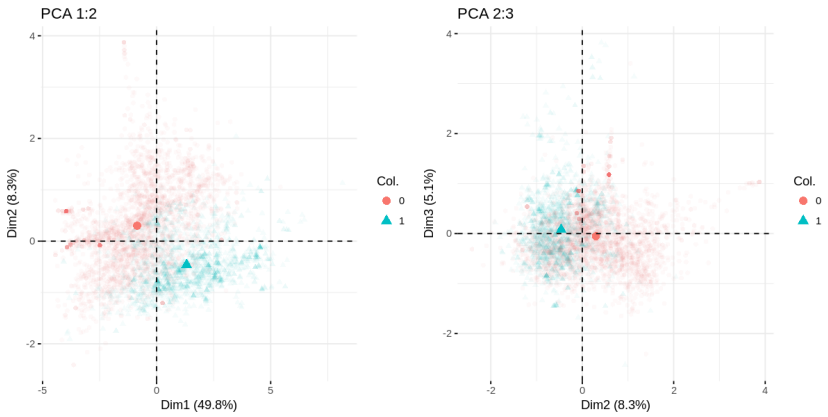

Figure 11: Plan factoriel 1:2 et 2:3

Il semblerait que les classes soit assez bien discriminé sur l'axe 2, moyen sur l'axe 1, et pas du tout sur l'axe 3.

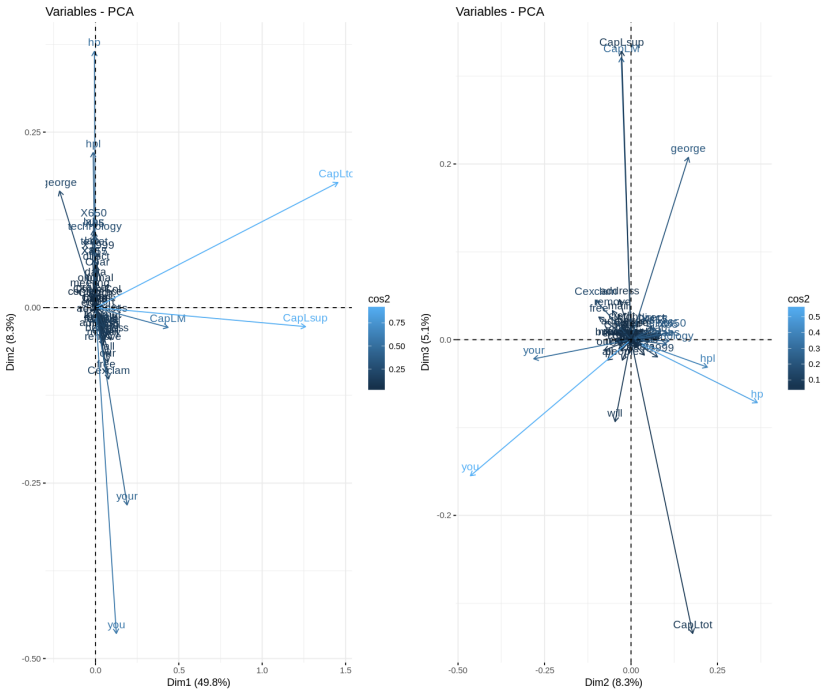

Figure 12: Cercle de corrélation sur les plan factoriel 1,2 et 2,3

On voit que comme on a pas scaler, les variables capLtot, CapLsup, et

CapLM, prenne bcp de place, et de variance. idem . que pour pour pca sur données original sans scale, l'axe 2, ne semble pas trop gené par ces variables.

l'axe 1 est corrélé à CapLtot, CapLsup, CapLM, your, Cdollar.(ce qui est plutôt logique). l'axe 2 est corrélé à hp, hpl, technology, X650. Cé qui conforte nos resultats avec "pca donnée orginale et scale"

3.4. Analyse en composantes principales avec transformation et normalisation

Nous examinons les valeurs propres pour déterminer le nombre de composantes principales à prendre en considération. Quand les données sont standardisés, on peux dire qu'on veux garder les composantes qui ont une valeur propre supérieur à 1, car cela indique que la composante principale (PC) concernée représente plus de variance par rapport à une seule variable d'origine.

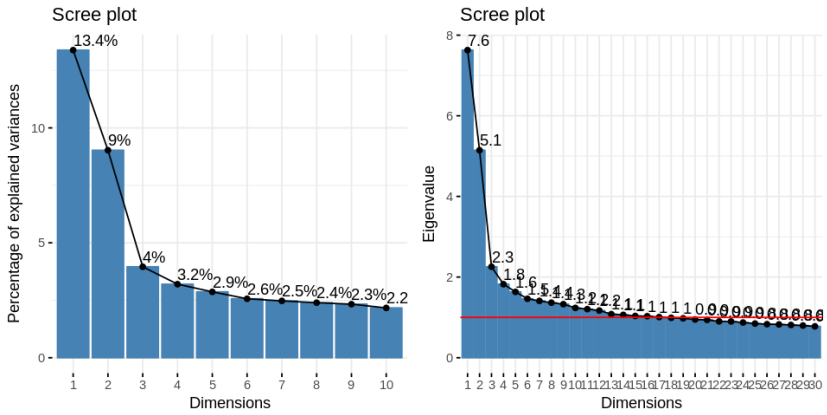

Figure 13: Pourcentage de variance expliqué par composantes principales et Valeur propres

On voit que la 1er composante principale capte 13.4% de variation contenue dans le jeu de données La 2ème cp, ne capte que 9% et la troisième 4% Ainsi nous pourrions prendre en considération les trois premieres cp, sur le premiere plan factoriel (22.4%),et sur le plan 2:3 (13%), le plan 1:3 (17.4%) Si nous prenons comme critère le nb de cp qui ont un valeur propre > 1, nous devrions prendre jusqu'a 14 cp.

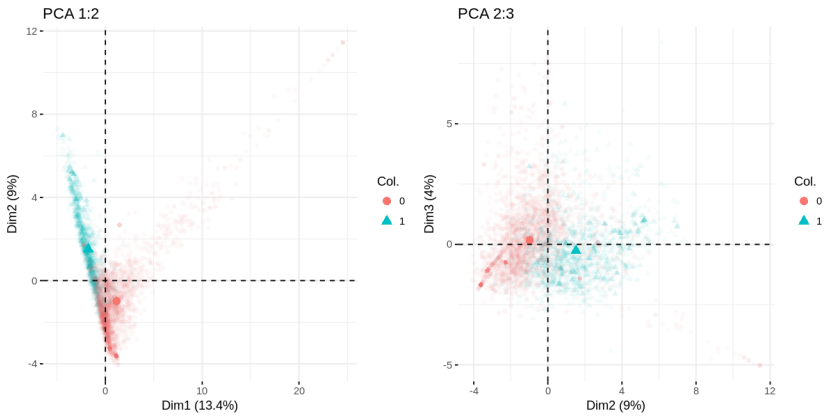

Figure 14: Plan factoriel 1:2 et 2:3

Sur le plan factoriel 1 et 2, Il semblerait que les classes soit assez bien discriminé sur l'axes 1 ET 2. Ensuite sur le plan 2 et 3, les classes assez bien séparé sur l'axe 2. Finalement, l'axe 3 ne nous apporte pas grand chose.

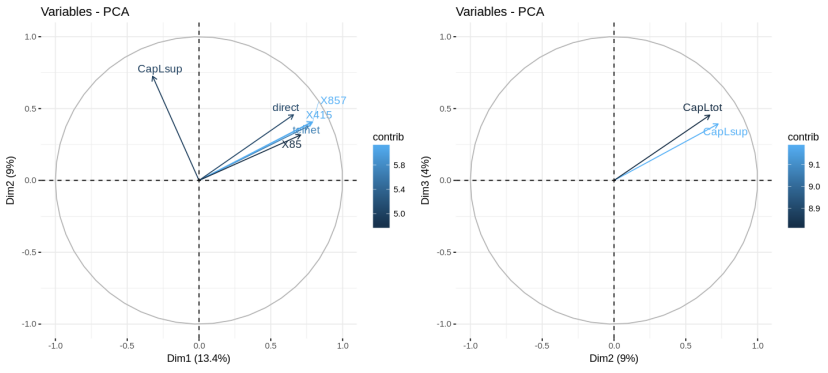

Figure 15: Cercle de corrélation sur les plan factoriel 1,2 et 2,3

On voit que les variables qui contribue le plus et qui sont bien projetés sur l'axe 1, sont X415,X857, telnet, direct, X85 le groupe de spam étant négatif sur l'axe 1, on peux conclure que ces variables sont anti-corrélé au spams. CapLsup et CapLTot sont corréleé a l'axe 2. le groupe de spam étant positif sur l'axe 2, on peux on conlure que ces variables sont corrélé aux spams.

En conclusion, les données non normalisé produisent des résultats non interprétables alors que la transformation logarithmique et la normalisation rendent les résultats interprétables. Avec les deux axes, nous pouvons discriminer les informations.

3.5. Classification ascendante hiérarchique (Classification des variables)

La proportion de la variance expliquée par les classes doit être le plus proche possible de 1 sans avoir trop de classes exprime le rapport entre la variance expliquée par le modèle et la variance totale. Le choix a été de choisir les données transformés (logarithmique) et d'utiliser Ward.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{1}$$

Une mesure de dissimilarité $1 - R^2$

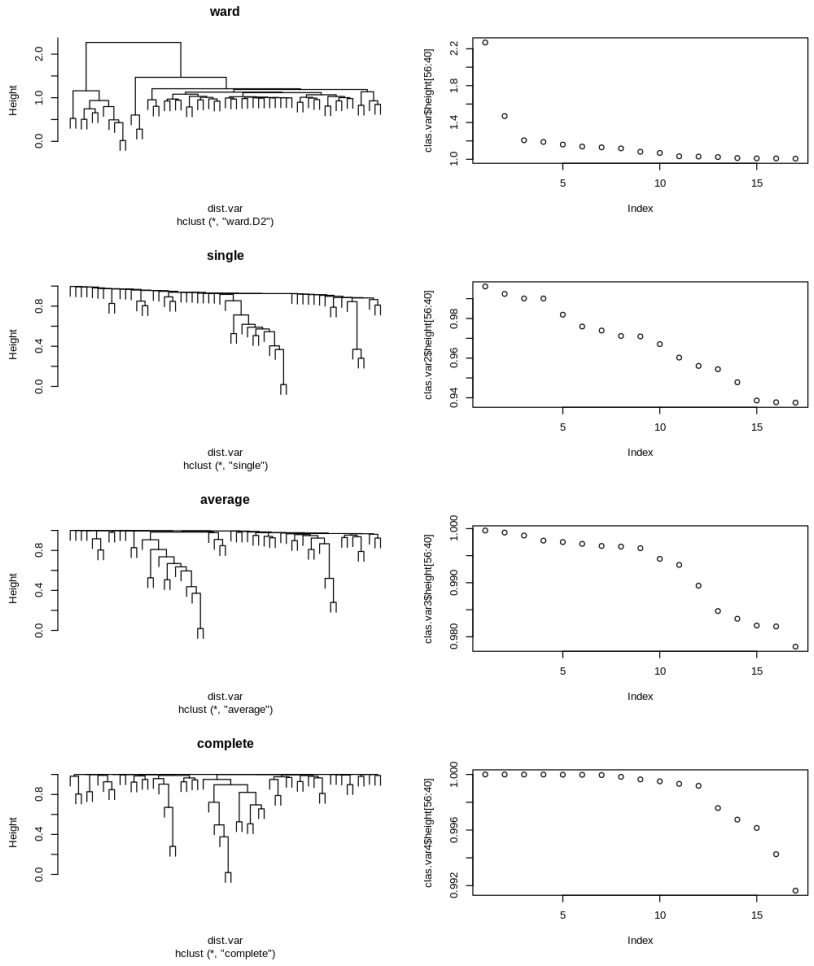

Figure 16: Dendogramme CAH Ward

Pour Ward, on voit qu'il y a 2 groupes qui se dégage de dendogramme, et le saut de hauteur est le plus grand entre 1 et 2. C'est a dire qu'il faudrait couper pour 2 classes (entre 2.3 et 1.5). Pour ce qui concerne le 1er groupe (en rouge) ,on retrouve les variables corrélés trouvé dans l'ACP données original et celle transformé et scaler. Il reste beaucoup de variable pour le groupe 2 (en bleu), mais on note que les 3 variables portant sur les majuscules sont regroupées, et qu'il corresponde bien aux resultats trouvé dans les ACP.

Pour Single, Average et Complete : Le dendogramme n'est pas très interprétable.

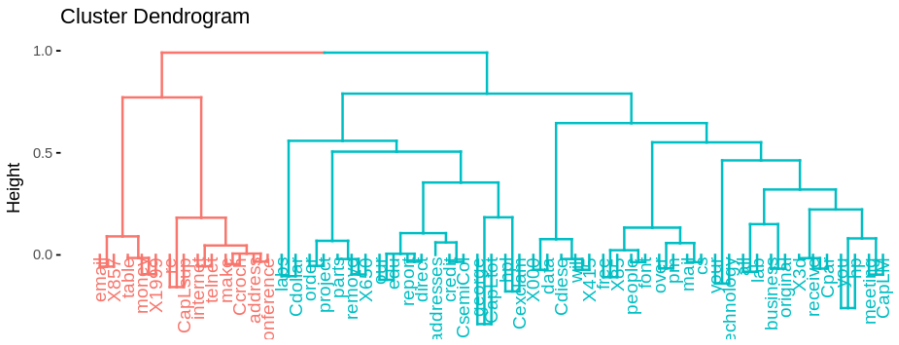

Figure 17: Résultat des différentes CAH

Pour Single et Average avec distance euclidienne (les résultats sont disponible dans le notebook), le dendogramme n'est pas très interpretable car on ne peux rien dire sur le nombre de classe.

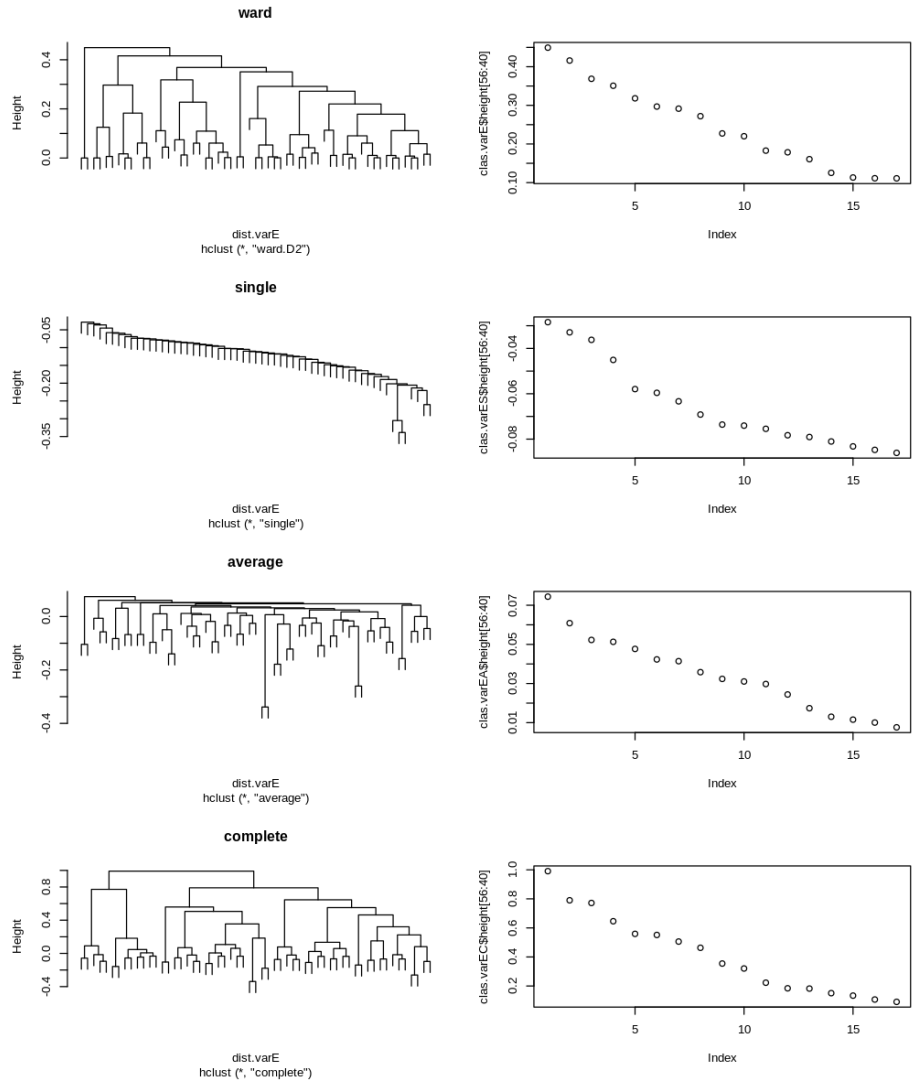

Figure 18: CAH avec la matrice de dissimilarité euclidienne

De plus, pour Complete avec distance euclidienne le dendogramme est intérpretable, on peux dire qu'il y a deux classes avec le saut d'hauteur.

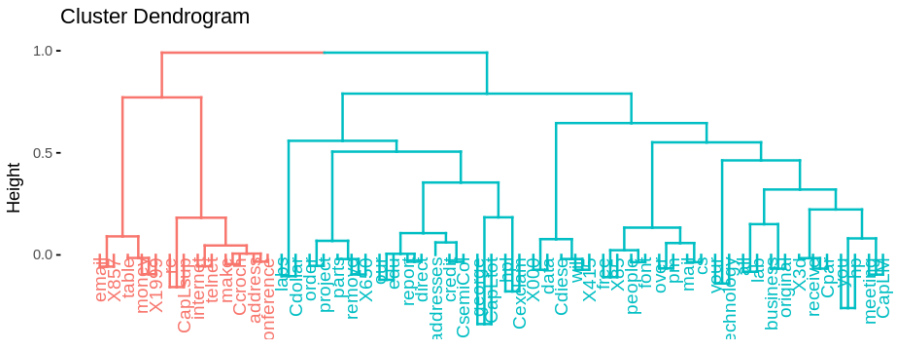

Figure 19: Dendogramme avec le lien maximum

On voit qu'enfaite le groupe 1 (rouge) n'est pas logique car il y'a des variables spécifique pour les spams(CapLsup,money) et les non-spams(conference, X1999).
Pour conclure, on voit qu'avec la distance $1 - R^2$,et Ward: Les groupes sont bien séparées et correspondent bien aux variables trouvées avec l'ACP.

3.6. Positionnement multidimensionnel

Nous utiliserons la technique du MDS (Multidimensional scaling).

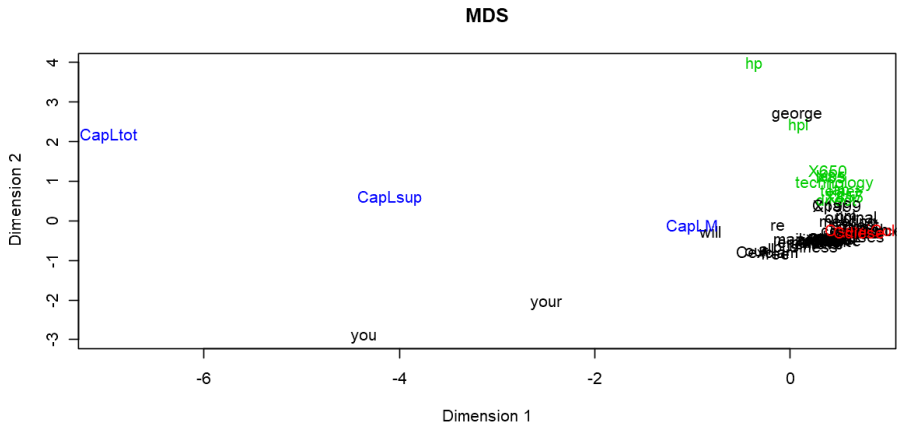

Figure 20: Résultat du clustering sur MDS

On voit que le groupe vert, corespond bien au variable corrélé avec l'axe 1 dans lorsqu'on effectue une ACP avec données transformer et scaler. On

voit que le groupe bleu, correspond bien au variable corrélé avec l'axe 2 dans l'ACP (transformer et scaler).

4. Approche qualitative

4.1. Recodage

Changement de stratégie, en considérant les aspects qualitatifs des variables : présence / absence, d'un mot ou caractère plutôt que les comptages. De même les variables concernant le nombre de lettres majuscules sont recodées en trois classes.Les statistiques globales sont disponibles fig .43 et aussi les statistiques par groupes .44.
On voit que pour les spams les variables suivantes peuvent discriminer:

1. CDollar (Cdol).
2. Cexclam (Cexc).
3. CapLtotq (Mt3).
4. CapLMq (Mm3).
5. CapLsupq (Ms3).
6. money (mone).
7. free (inte).
8. Et d'autres variables en appendice.

4.2. Analyse factorielle multiple des correspondances

C'est une méthode factorielle de réduction de dimension pour l'exploration statistique de données qualitatives complexes puis cette méthode est une généralisation de l'analyse factorielle des correspondances permettant de décrire les relations entre p variables qualitatives simultanément observées sur n individus.

Elle sert à résumer et visualiser un tableau de données contenant plus de deux variables catégorielles. On peut aussi la considérer comme une généralisation de l'analyse en composantes principales lorsque les variables à analyser sont catégorielles plutôt que quantitatives.

L'objectif est d'identifier:

1. Un groupe de personnes ayant un profil similaire dans leurs réponses aux questions.

2. Les associations entre les catégories des variables.

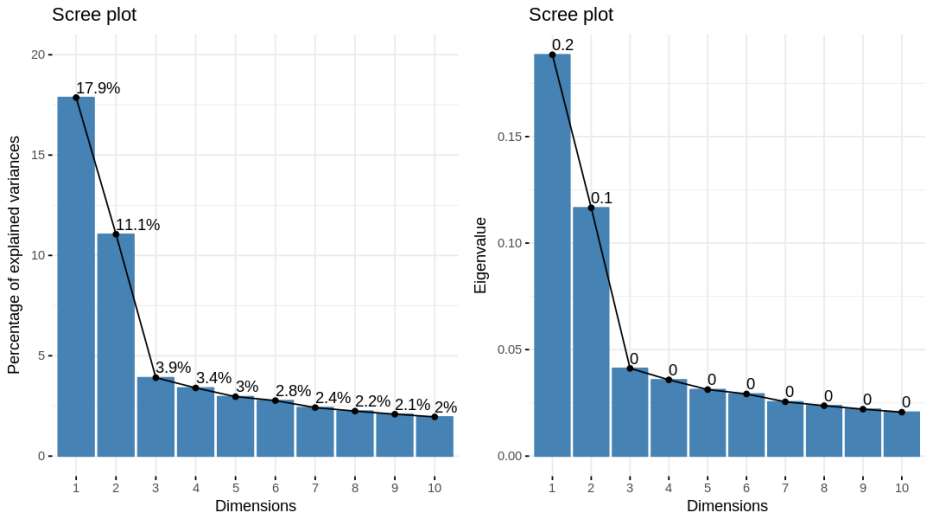

Figure 21: Valeur propres et pourcentage d'explication sur AFCM

On va choisir les deux premiers axes (17.9% et 11.1%) avec un plan de (28.92%).

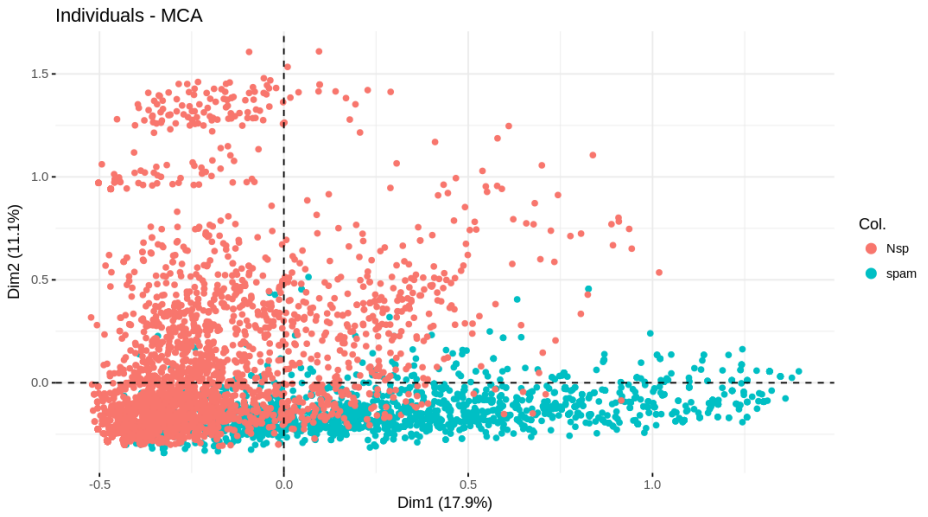

Figure 22: Individus sur le plan factoriel de l'AFCM

On voit que pour le groupe *SPAM*, il est plutôt en long de l'axe 1 et le groupe de messages normales est plutôt vers les valeurs négatifs des deux axes et est éparpille. L'axe 2 ne discrimine pas les groupes.

4.3. Classification ascendante hiérarchique (Classification des modalités)

Nous effectuons une CAH avec le critère de Ward, la distance euclidienne est utilisée pour effectuer la matrice de dissimilarité avec les coordonnées des variables provenant de l'AFCM.

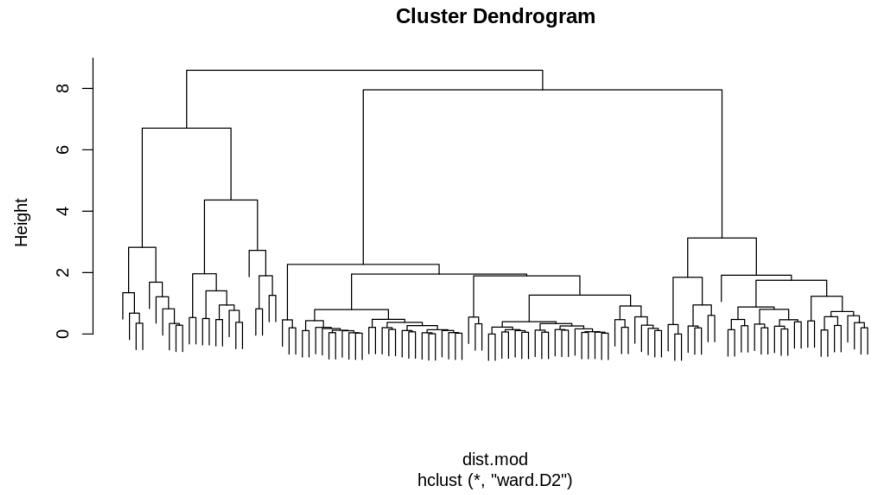

Figure 23: Individus sur le plan factoriel de l'AFCM

Avec le saut d'hauteur, on couperai entre 3 et 4. On choisit de prendre 4 classes.

4.3.1. CAH et K-means

Comparaison avec la méthode de partitionnement K-means sur 4 classes.

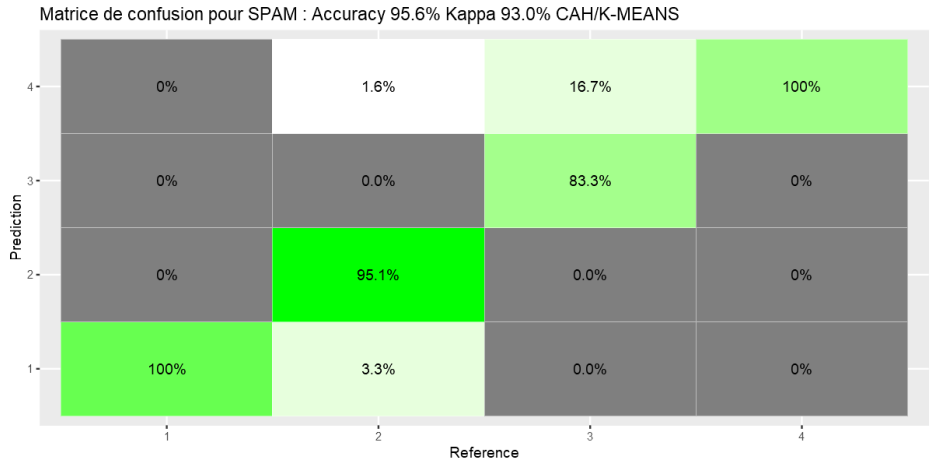

Figure 24: Matrice de confusion avec CAH et K-means

On remarque que dans la matrice de confusion avec K-means et CAH en utulisant le critère de Ward donne la même chose. Les classes étants presque les mêmes, logique puisque que Ward et K-means sont liées avec l'inertie intra-classe.

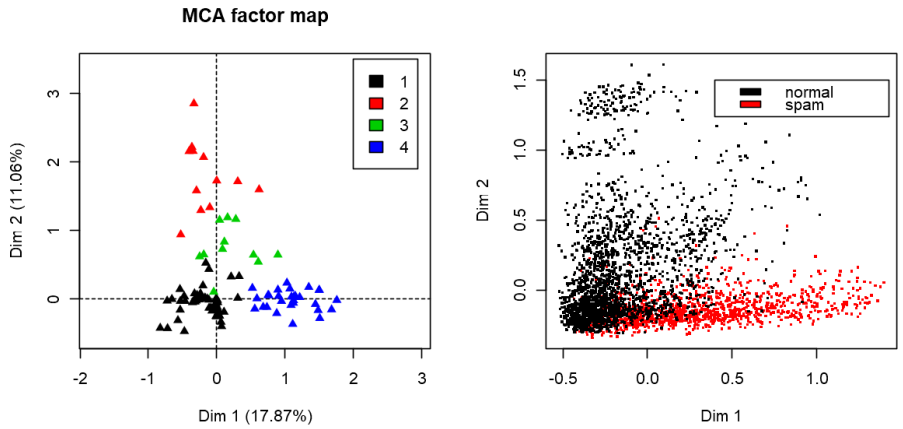

Figure 25: Visualisation des 4 classes

On voit que le groupe 1 (noir) est pour la catégorie des *SPAM*, ensuite le groupe 2 (rouge) est confondu pour spams et non spams. Ensuite, on voit que le groupe 3 et 4 est pour les messages normaux.
Une étude des variables est effectué dans le notebook associé.

5. Approche par NMF

Les données quantitatives sont reconsidérées mais en intégrant le caractère essentiellement "creux" de la matrice des données. Cette situation couramment répandue a suscité une nouvelle forme d'analyse dite Non Negative Matrix Factorization (NMF) dont le principe est de rechercher deux matrices de faible rang r de telle sorte que leur produit approche au mieux les valeurs observées. Contrairement à L'ACP où les facteurs sont recherchés orthogonaux 2 à 2, cette méthode impose la contrainte de non négativité des matrices pour construire les facteurs de la décomposition. Ces facteurs ne permettent plus de représentation comme en ACP ou en MDS mais au moins une classification non supervisée tant des lignes que des objets lignes et colonnes de la matrice. Cette approche est testée sur les données de spam pour en comparer les résultats obtenus.

Nous utilisons 4 algorithmes :

1. brunet - Kullback-Leibler divergence
2. lee - Euclidean distance
3. snmf/l - Euclidean-based objective function,
4. snmf/r - Euclidean-based objective function,

5.1. Choix de la méthode

L'évaluation de la "stabilité" de plusieurs exécutions de NMF repose sur des critères (silhouette, consensus, corrélation cophénétique) issues des méthodes de classification non supervisée de se déterminer pour une méthode de moindres carrés (snmf/l) convergeant plus rapidement et présentant des valeurs optimales (cophenetic, residuals,...) ainsi qu'une meilleure stabilité sur plusieurs exécutions.

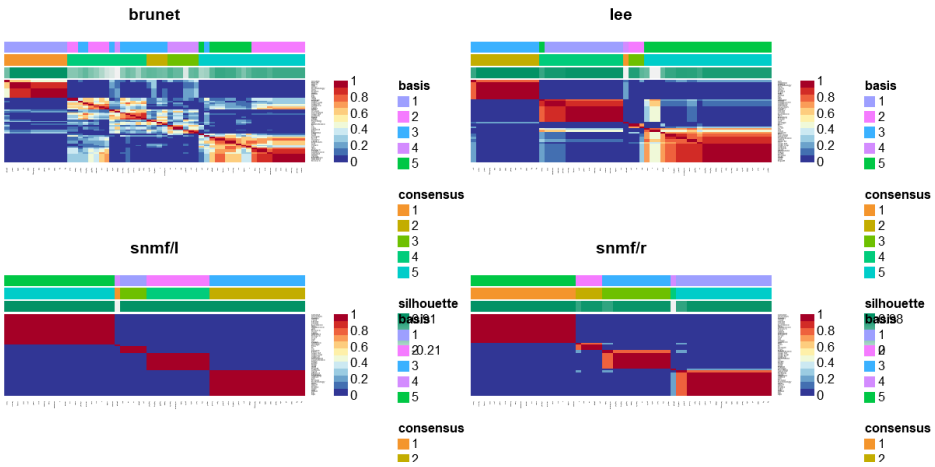

Figure 26: NMF

La snmf/l est choisi car convergeant plus rapidement et présentant des valeurs optimales (cophenetic, residuals,...) ainsi que la meilleure stabilité sur plusieurs exécutions.

5.2. Choix du rang de la matrice

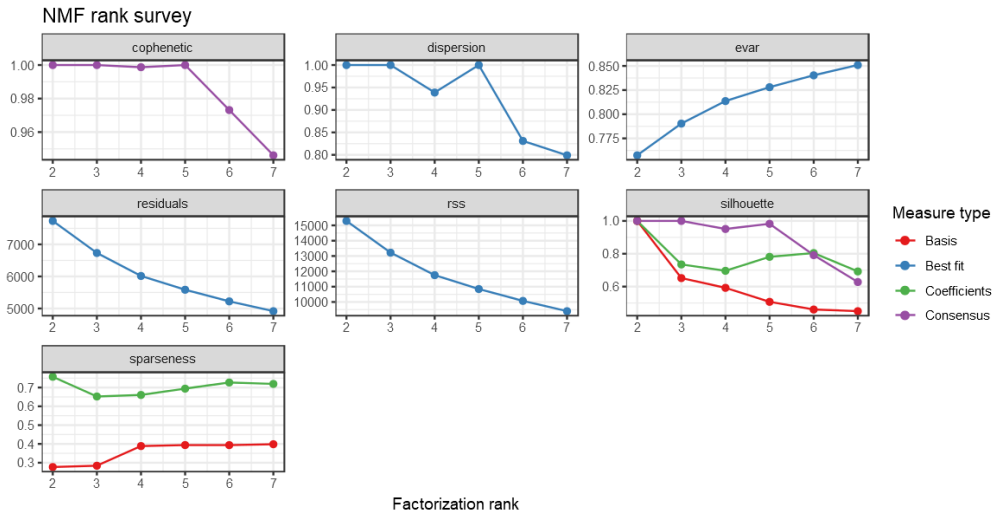

Figure 27: Rang de la matrice

Les figures conduisent au choix de r à 5, la corrélation cophénétique de 1 avant décroissance et meilleur graphique de consensus.

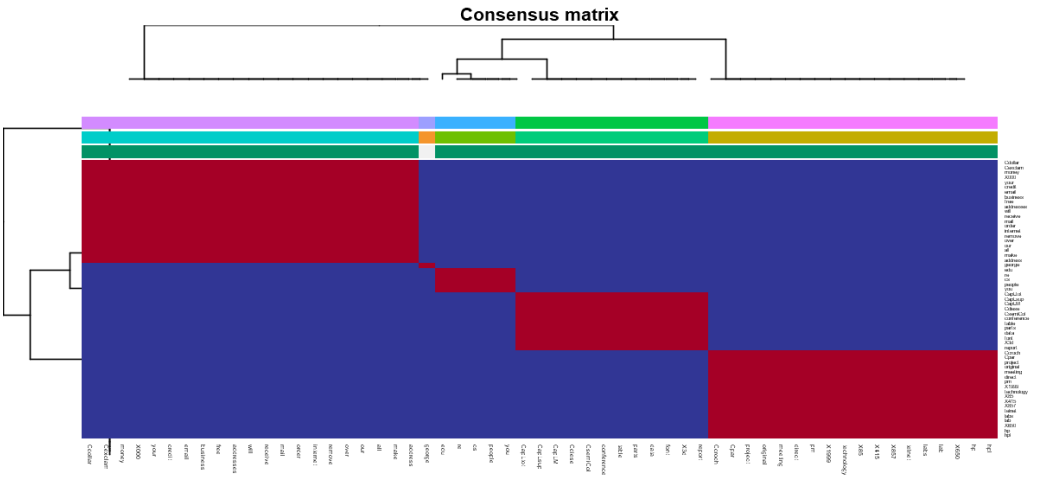

Figure 28: NMF Consensus Map

Les principaux critères (nombre de lettre capitales,etc) sont regroupés dans une même classe et correspondant simultanément à une classes de spams.
En revanche le mot-clef *georges*, qui est le prénom du destinataire, est isolé et caractérise des courriels correctes.

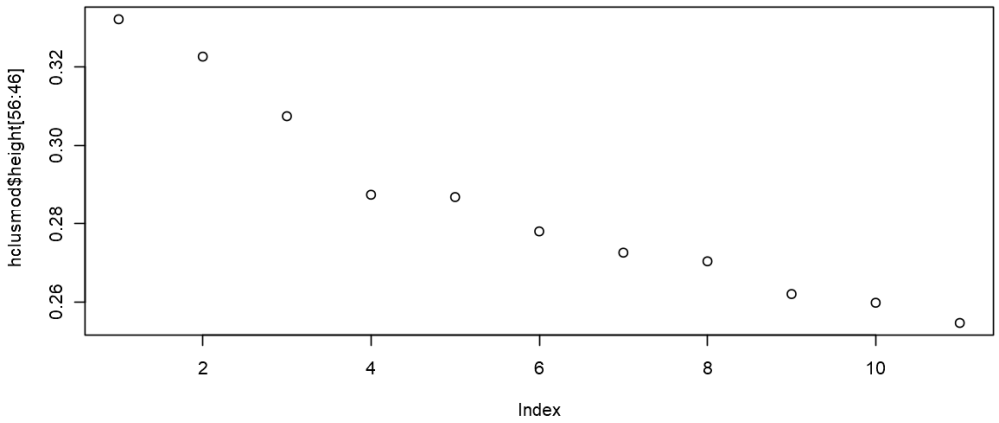

Figure 29: CAH avec le critère de Ward sur H

Nous choisissons un nombre de classe à 4 et décidons de faire une représentation grâce à la MDS **??** disponible en annexe.
Il n'est pas possible comme en ACP ou AFCM de mettre en relation les deux représentations des lignes et colonnes, individus et variables de la matrice factorisée. Cela peut être fait de façon détournée à l'aide d'une heatmap qui peut intégrer deux classifications obtenues par ailleurs. Donc nous avons normaliser la matrice W et effectuer une dissimilarité dessus pour l'envoyer en entrée d'une CAH Ward .48.
La NMF est bien pour des matrices creuses présentent des valeurs très disparates.

6. Autres méthodes répondant aux mêmes objectifs

6.1. Classification Hiérarchique sur Composantes Principales

Algorithme de la méthode HCPC

- En premier lieu on effectuée une ACP, AFC, ACM, AFDM ou AFM en fonction du type de données.

- Ensuite on applique la classification hiérarchique sur le résultat de la première étape.

- On choisit un nombre le nombre de classe en fonction du dendrogramme obtenu à CAH. Un partitionnement initial est effectué.

- Puis on effectue le k-means pour améliorer le partitionnement initiale obtenu.

Dans notre cas, on se base sur l'ACP sur les données tranformées et scaler. On utilise l'ACP car nous avons des données continues. L'étape ACP peut être considérée comme une étape réduisant le bruit de fond dans les données, ce qui peut conduire à une classification plus stable.
Pour le nombre de composants principales, on va utiliser la fontion de *FactoMineR* estimer à partir d'une cross-validation. (5)

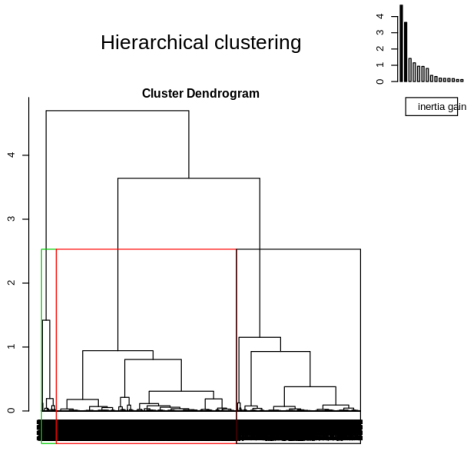

Figure 30: Dendogramme de HCPC

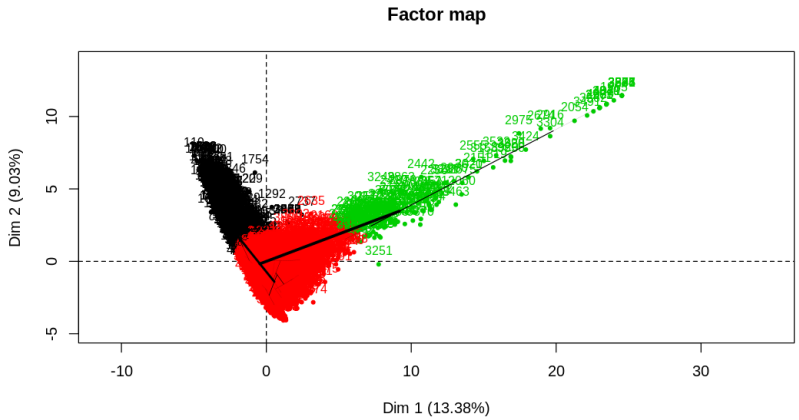

Figure 31: Plan factoriel HCPC

Trois clusters sont trouvés. Une étude détaillé est disponible dans le notebook, et on voit que comme attendu, vu les graphes et les variables lié aux groupes, que le groupe 1(Spam) est lié aux valeur négative sur l'axe 1 et positive sur l'axe 2 ('CapLsup' 'your' 'CapLM' 'CapLtot') (par rapport aux moyenne).
Le groupe 2(Non-spam) est lié aux valeur moyenne sur l'axe 1 et négative sur l'axe 2 (par rapport aux moyenne)('george' 'hp' 'edu' 're' 'meeting'). Comme attendu,les graphes et les variables lié au groupe 3 (Non-spam) est lié aux valeur positive sur l'axe 1 et positive sur l'axe 2 (par rapport aux moyenne) ('george' 'hp' 'edu' 're' 'meeting').

6.2. Analyse sémantique latente

Nous avons effectué une lsa et ensuite un kmeans.
Le packages NbClust nous a permis de trouver qu'il fallait 2 classes.

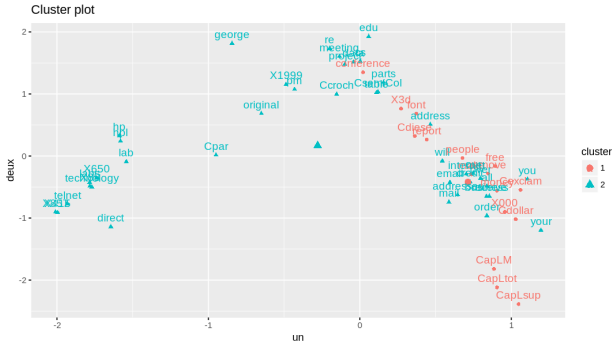

Figure 32: Plan factoriel LSA

On retrouve un groupe de non-spams (en bleu) avec notamment les variables "george", "hp", "X1999", "X650", "meeting", "address", et un groupe de spam (en rouge) avec notamment "CapLSup", "CapLM", "CapLtot", "free", "money".
Les variables sont vraiment bien repérées.

6.3. Synthèse des résultats

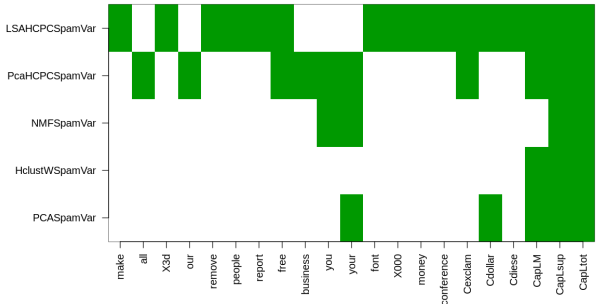

Figure 33: Synthèse sur les *SPAM*

On voit que CapLtot et CapLsup est repéré par toutes les méthodes, de plus CapLM est quasiment repéré par tous le monde.
LsaHCPp repère le plus de variables pour *SPAM*.

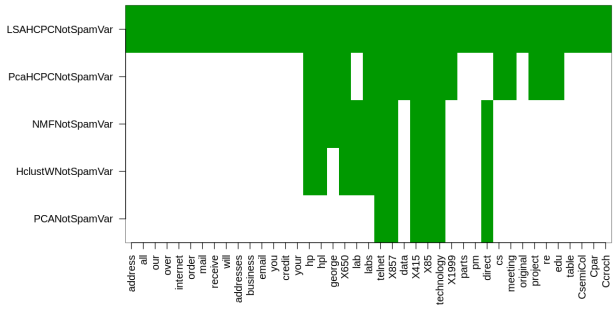

Figure 34: Synthèse sur les non *SPAM*

On voit que telnet, X857, X415, X85, technology est repéré par toutes les methodes, de plus hp,hpl,direct est quasiment repéré par tous le monde. lsahcpc repere le plus de var Logique vu que les Xnombre variables sont les infos sur georges

7. Co-Clustering

7.1. Blockcluster

Co-clustering Package for Binary, Categorical, Contingency and Continuous Data-Sets
Simultaneous clustering of rows and columns,

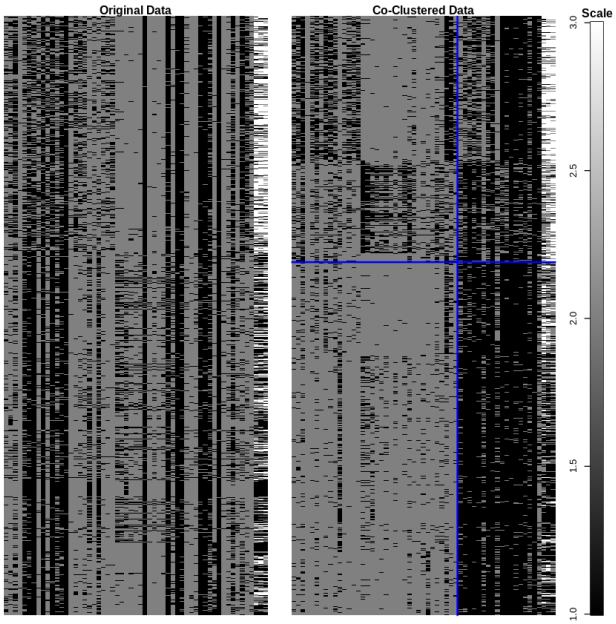

Figure 35: BlockCluster

Visuellement nous voyons qu'il n'y a que deux block qui sont assez homogènes ((2,1),(2,2)).
Le groupe (2,1) est caractérisé par les variables: 'CapLM' 'CapLsup' 'CapLtot', donc les spams.
Le groupe (2,2) est caractérisé par les variables: 'CapLM' 'CapLsup' 'CapLtot' 'cs' 'meeting' 'edu' 'conference' , donc les pas spams.

7.2. Scikit-Learn Visualisation

7.2.1. t-SNE

t-SNE tente de trouver une configuration optimale selon un critère de théorie de l'information pour respecter les proximités entre points : deux points qui sont proches (resp. éloignés) dans l'espace d'origine devront être

proches (resp. éloignés) dans l'espace de faible dimension.
Une distribution de probabilité est également définie de la même manière pour l'espace de visualisation.

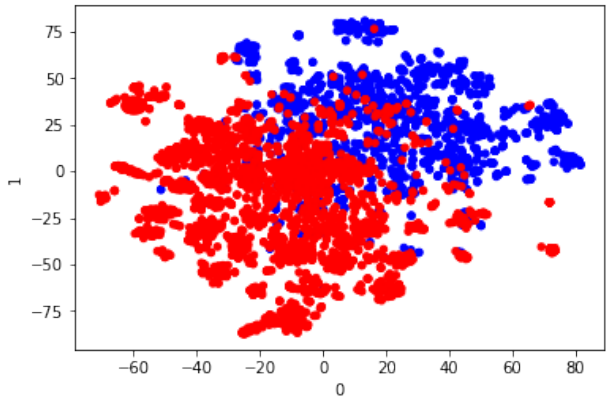

Figure 36: Tsne

La tsne trouve bien des groupes malgré qui ne connaisse point les classes.

7.2.2. Isomap

Isomap peut être considéré comme une extension de MDS ou de K-PCA. Cet algorithme cherche un espace de dimension réduite qui conserve les distances "géodésiques" entre tous les points. Cette méthode se base sur les plus proches voisins de chacune des instances afin de conserver cette distance.

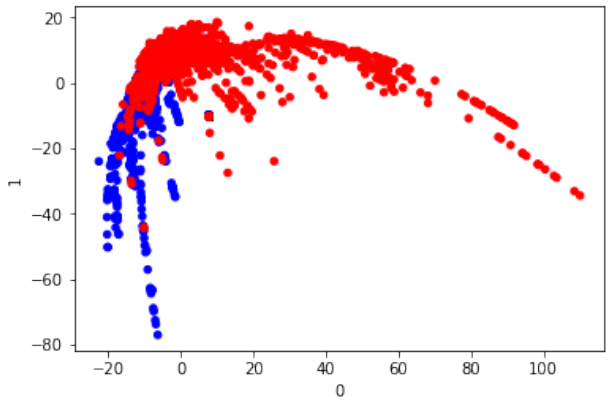

Figure 37: Isomap

Isomap trouve bien des groupes malgré qui ne connaisse point les classes.

7.3. Scikit Learn Classification

7.3.1. DBSCAN

L'algorithme DBSCAN (Density-Based Spatial Clustering of Applications with Noise)itère sur les points du jeu de données. Pour chacun des points qu'il analyse, il construit l'ensemble des points atteignables par densité depuis ce point : il calcule l'epsilon-voisinage de ce point, puis, si ce voisinage continent plus de n_min points, les epsilon-voisinages de chacun d'entre eux, et ainsi de suite, jusqu'à ne plus pouvoir agrandir le cluster. Si le point considéré n'est pas un point intérieur, c'est à dire qu'il n'a pas suffisamment de voisins, il sera alors étiqueté comme du bruit. Cela permet à DBSCAN d'être robuste aux données aberrantes puisque ce mécanisme les isole.

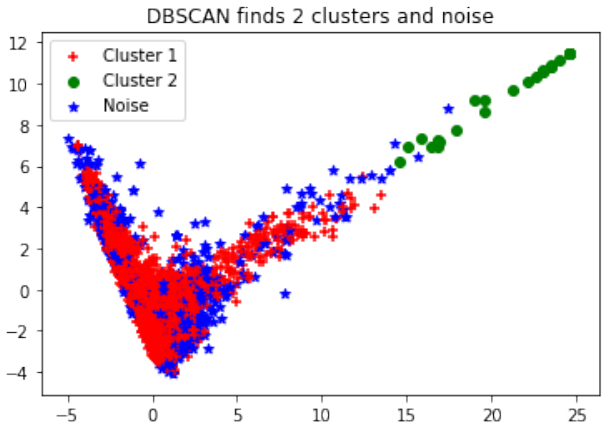

Figure 38: DBSCAN

ça ne marche pas, il y a trop de bruit.

7.3.2. Birch

BIRCH1 (« balanced iterative reducing and clustering using hierarchies ») est un algorithme d'exploration de données non-supervisé utilisé pour produire une segmentation hiérarchisée sur des volumes de données particulièrement importants.

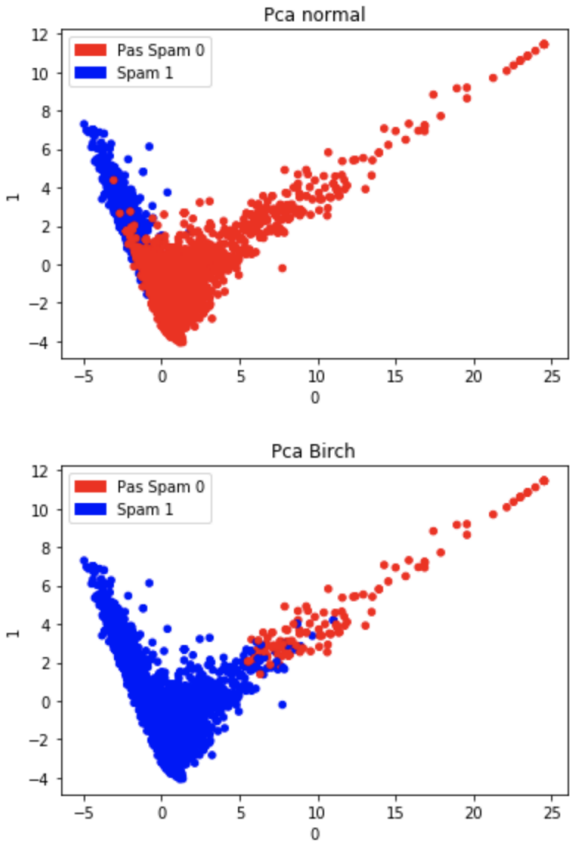

Figure 39: Birch 2 cluster

On voit que birch predit des spams qui ne sont pas des spams.
On va regarder la matrice de confusion, ainsi que le rapport de classification.

Table 1: Classification Rapport Birch 2 clusters

	precision	recall	f1-score	support
0	1.00	0.04	0.09	2788
1	0.40	1.00	0.58	1813
avg/total	0.77	0.42	0.28	4601

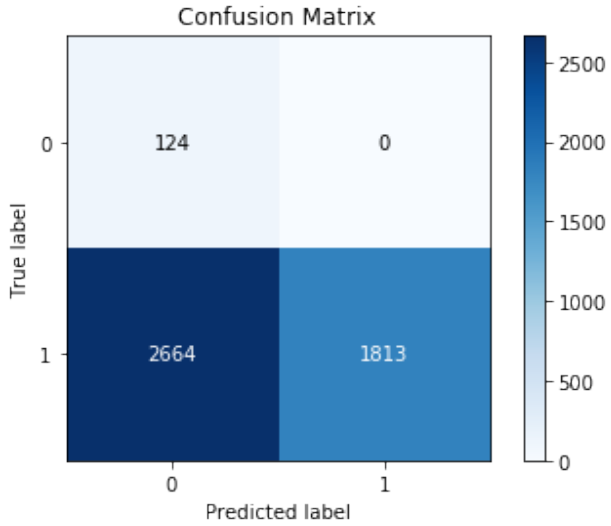

Figure 40: Birch 2 cluster confusion Matrix

Comme on voit sur le graph et sur les matrices, il repère les non-spams(précisions pour les non-spams) vraiment bien, mais mets trop de messages en spams alors qu'ils ne sont pas des spams(FP)(recall pour les non-spams).
On voit que Birch avec 2 clusters a du mal, on va rajouter troisième cluster de "message indéfini" , regardons le plot et la matrice de confusion.

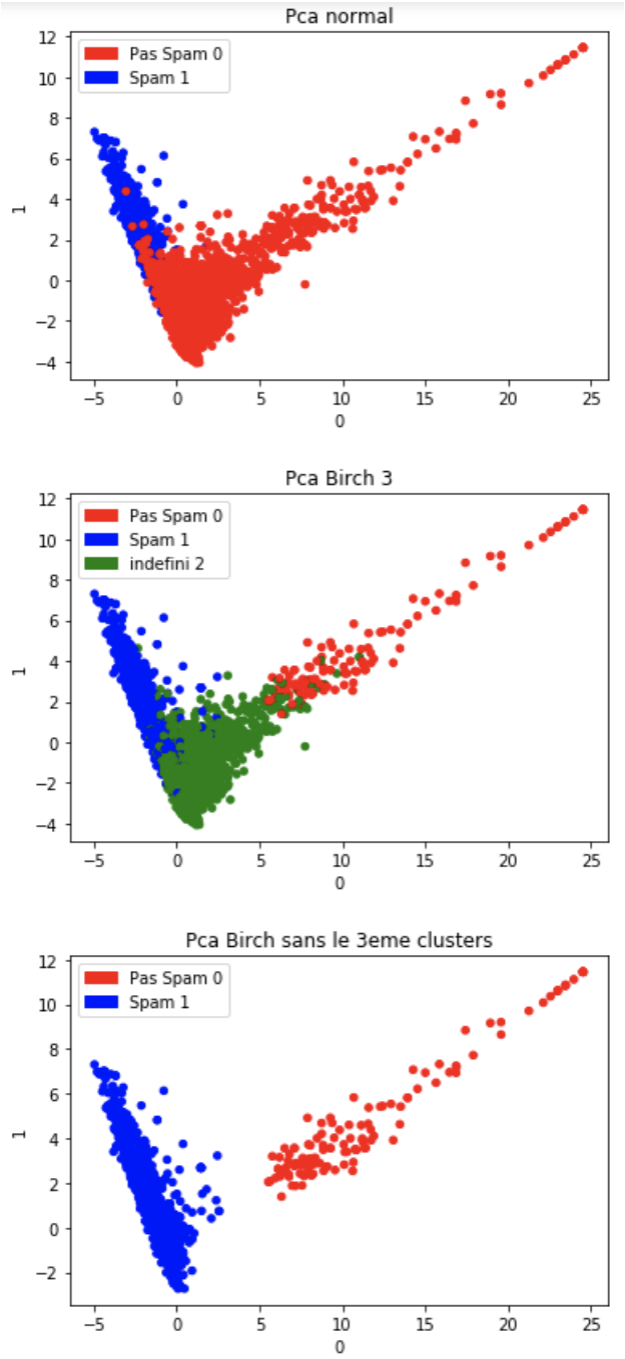

Figure 41: Birch 3 cluster

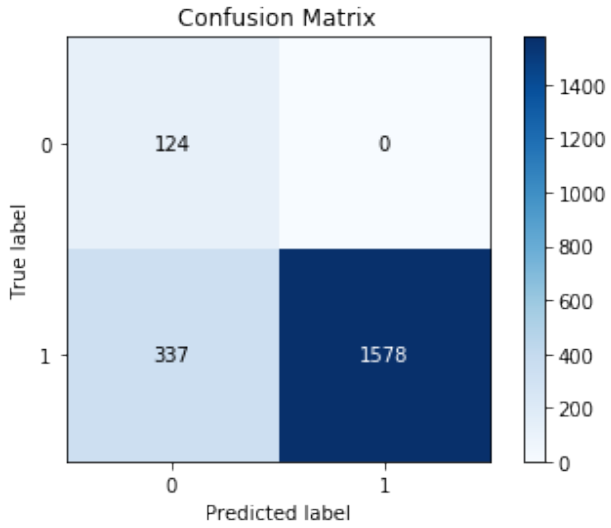

Figure 42: Birch 3 cluster confusion matrix

On voit qu'il repère bien les spams à 82%. On voit qu'il repère très bien les non-spams 100% mais predict des spams alors que non-spams 27% recall.

7.4. Coclust

On a effectué coclustInfo, qui utilise la théorie de l'information pour faire le co-clustering.
On a 3 blocs en colonnes:

- 1er: spams avec variables CapLM, CapLsup, CapLtot,you
- 2ème: spams avec variables money, free, Cexclam, receive, Cdollar
- 3ème: Pas spams avec variables george, hp, hpl, X857, X415,X85,X1999, cs, meting

7.5. biclust et particulièrement la méhode BCQuest

On a 2 blocs en colonnes:

- 1er: non-spams avec variables 'hp' 'hpl' 'george' 'X650' 'lab' 'labs' 'cs' 'meeting' 'project' 'table' 'conference'
- 2ème: spams avec variables 'money' 'Ccroch' 'Cdollar' 'Cdiese'

8. Conclusion

Nous avons vu beaucoup de méthode classification et leurs variantes suivant le type de données que nous avons (qualitatives ou quantitatives).
Ce projet, nous a permis d'aborder le co-clustering à travers des données textuelles mis sous forme de document-terme. Cette représentation éparse est utile dans le cas d'une classification en ligne et en colonne de manière simultané afin d'optimiser des blocs homogènes.

Toutes ces techniques peuvent en effet donner de bon résultat mais il ne faut pas oublier le pré-processing. On a pu constater des différences majeurs lors qu'un type de normalisation était réaliser ou non ainsi qu'une transformation des données.

Figure .43: Statistiques globales

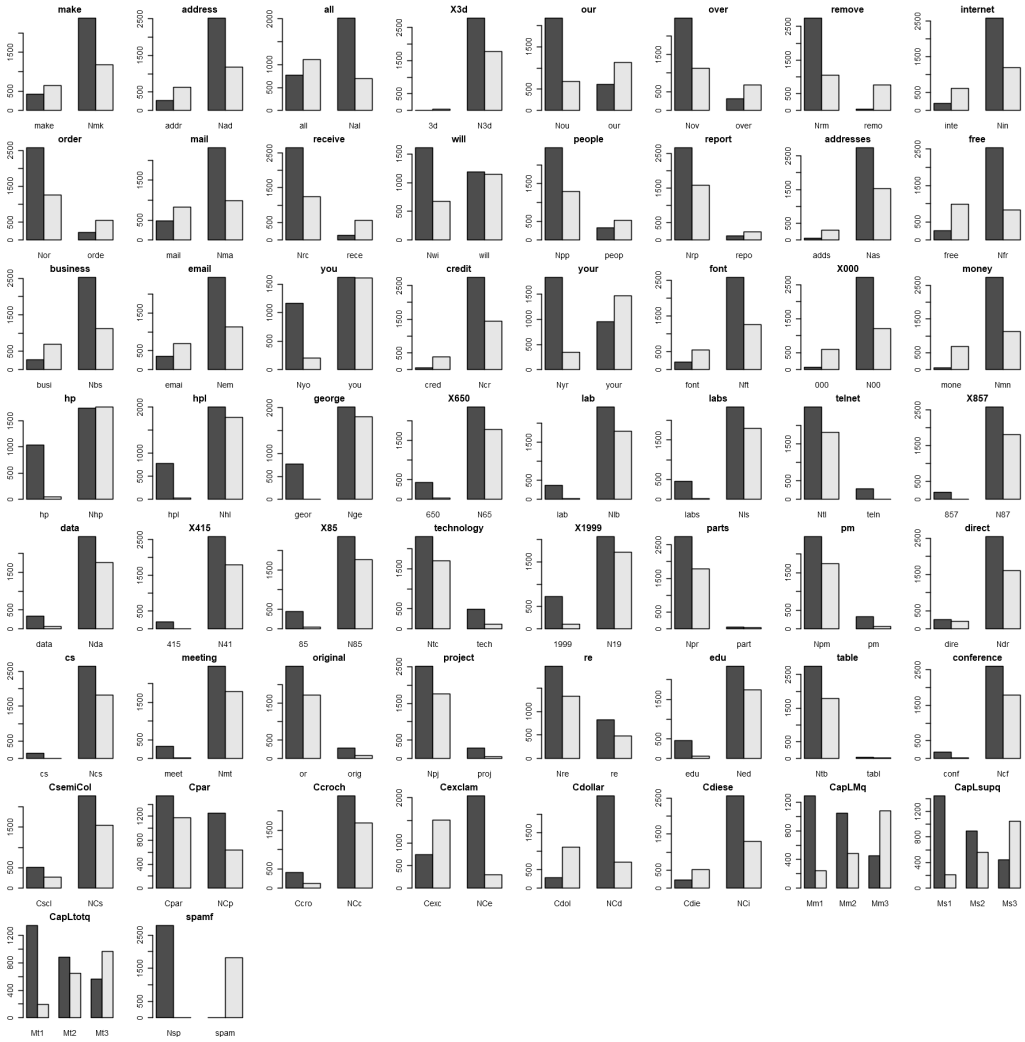

Figure .44: Statistiques par groupes

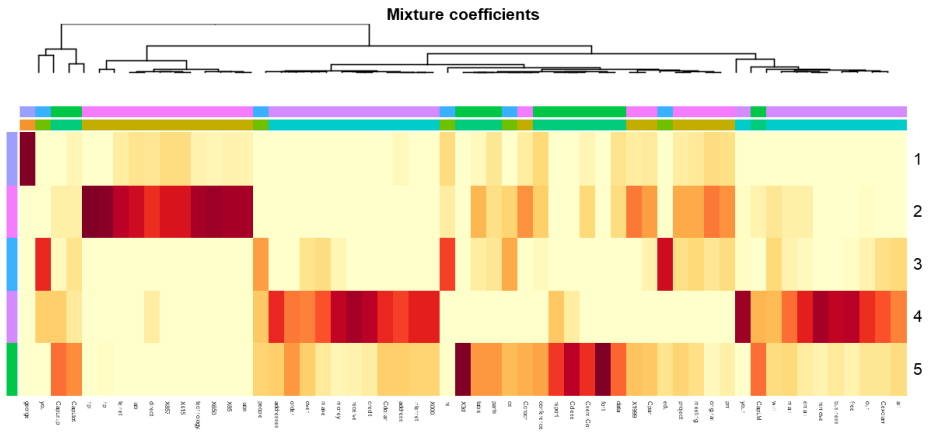

Figure .45: NMF mixture components

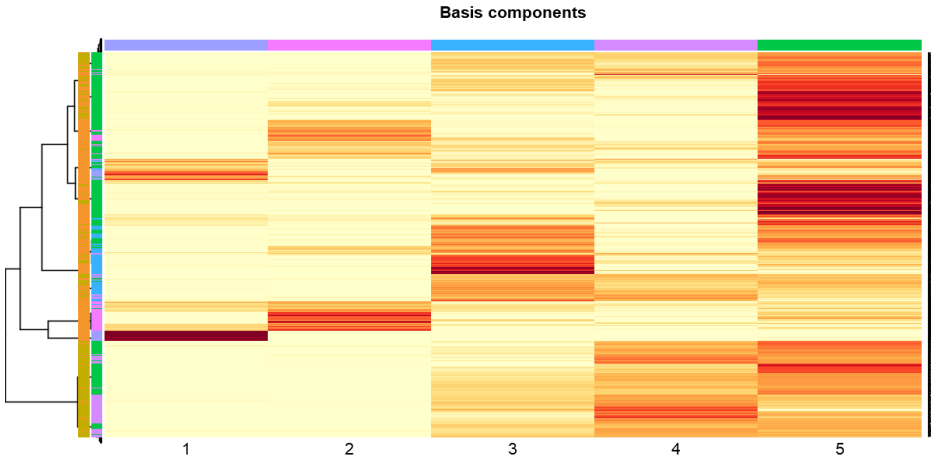

Figure .46: NMF Basis component

Figure .47: MDS sur les 4 clusters de H par la CAH

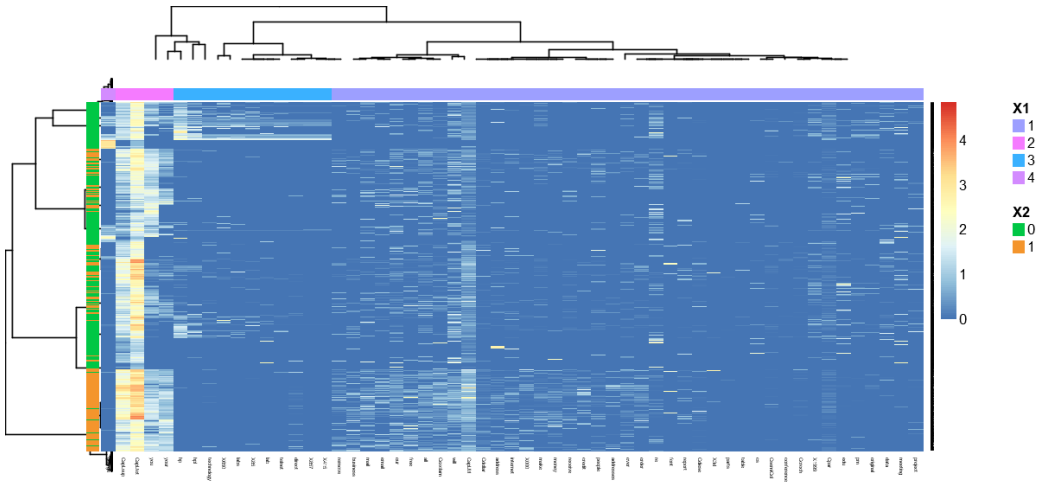

Figure .48: Heatmap sur la matrice dissimilarité W normaliser et CAH Ward