Projet Classification

1) Classification des phonèmes sans prétraitement

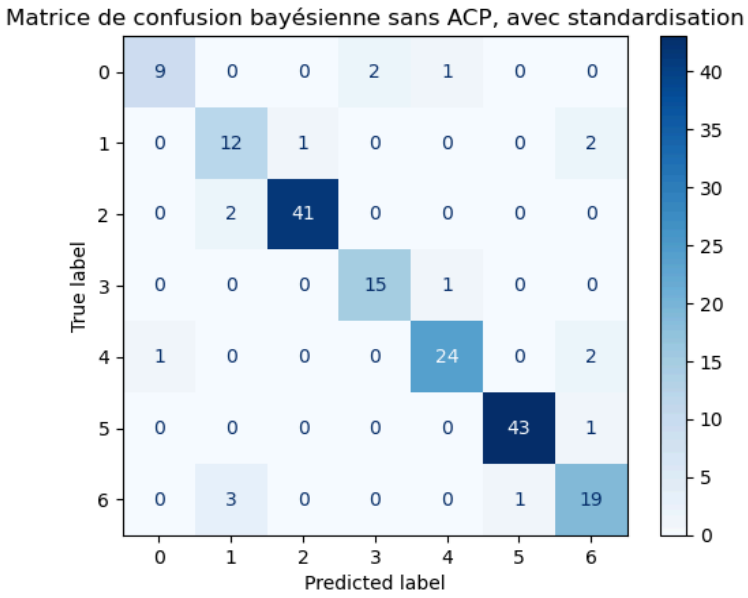
1.1) Méthodes supervisées

Les deux méthodes de classification supervisées utilisées sont la méthode Bayésienne et la méthode des k plus proches voisins.

Méthode Bayésienne

L'implémentation de la méthode bayésienne que nous avons utilisée provient de la classe GaussianNB (cette classe ne prend pas de paramètres). Nous avons dans un premier temps réalisé la classification sans effectuer de standardisation, puis nous avons réalisé cette même classification, cette fois avec les données standardisées à l'aide de la classe de standardisation MinMaxScaler.
Score avec méthode Bayésienne sans ACP, ni standardisation : 55%.
Score avec méthode Bayésienne sans ACP, avec standardisation : 90%.

La meilleure matrice de confusion que nous obtenons est la suivante.



Nous pouvons observer que:

- Les classes 0, 2 et 5 sont les mieux identifiées avec seulement 1 prédiction incorrectes
- Les classes 4 et 3 sont également bien identifiées avec 2 prédictions incorrectes
- Finalement, les classes 1 et 6 sont les classes les moins bien identifiées avec 5 prédictions incorrectes
- En conclusion, ce sont de très bons résultats.

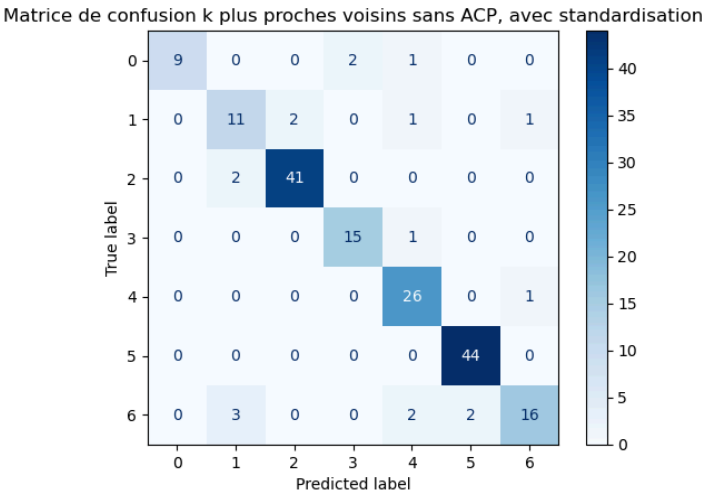
Méthode des k plus proches voisins

L'implémentation de cette méthode provient de la classe KNeighborsClassifier

Cette classe prend un paramètre, le paramètre "k". Celui-ci représente le nombre de voisins qui devra être pris en compte lors du décompte du nombre de voisins dans chaque classe. Par exemple si k=10, pour chaque point "x" nous allons prendre les 10 points les plus proches de "x", et compter le nombre de points qui appartient à chaque classe. La classe qui a le plus grand nombre de points parmi les 10 gagne, et est désignée comme étant la classe dont fait partie "x".
Ici, afin de trouver la meilleure valeur "k", nous avons décidé de toutes les essayer à l'aide d'une boucle (la valeur "k" qui produit le meilleur score gagne). Cette méthode nous a permis de trouver que la meilleure valeur de "k" est 96.

Nous obtenons donc les résultats suivants.
Score avec méthode des k plus proches voisins sans ACP, ni standardisation : 69.4%.
Score avec méthode des k plus proches voisins sans ACP, avec standardisation : 90%.

La meilleure matrice de confusion que nous obtenons est la suivante.



Nous pouvons observer que:

- La classe 0 est parfaitement identifiée
- Les classes 2, 3, 5 et 6 sont bien identifiées avec 2 prédictions incorrect
- Finalement, les classes 1 et 4 sont les classes le moins bien identifiées avec 5 prédictions incorrect
- En conclusion, ce sont également de très bons résultats.

1.2) Méthodes non supervisées

Les deux méthodes non supervisées utilisées sont la méthode K-means et la méthode Gaussian Mixture.

Méthode K-means

L'algorithme a été utilisé avant et après utilisation de la normalisation ("scaler"), ce qui a permis de voir une différence de score de performance notable.

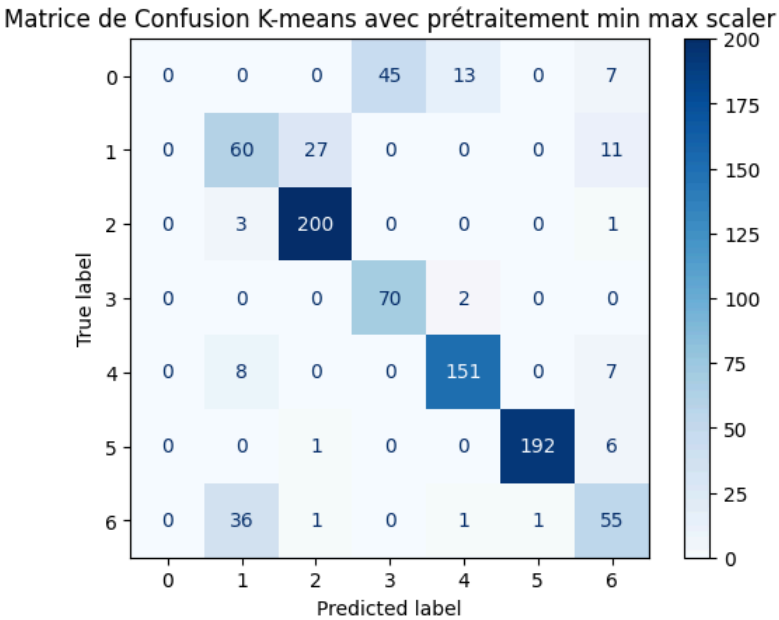
Score K-means avant normalisation : 55,79%.
Score K-means après normalisation : entre 74 et 81% selon la méthode de normalisation choisie.

Les paramètres utilisés pour cette méthode sont le nombre de clusters, le type d'initialisation de l'algorithme (pour la position initiale des classes) et "random_state".
Ici, nous avons 7 familles de fruits, donc le premier paramètre "n_clusters" est valorisé à 7.
Le paramètre 'init' a été fixé à **random** car la méthode 'k-means++' ne réalisait pas de meilleurs résultats. En effet, cette deuxième méthode se base sur des probabilités pour une convergence plus rapide, mais ne permet pas de meilleurs résultats dans cette situation.
Le paramètre "random_state" a été fixé à 20 pour toutes les méthodes mises en place afin de stabiliser et comparer les résultats. En annexe (fin du fichier), deux algorithmes ont cherché la meilleure valeur, qui est 20 pour Kmeans et 51 pour Gaussian Mixture, donc les résultats peuvent différer selon ce paramètre.
Un autre paramètre utilisable est "max_iter" qui a pour valeur par défaut **300**, ce qui est assez pour classifier les résultats, l'augmentation de ce paramètre n'a donc pas d'effet sur les performances.
Les autres paramètres ont été laissés par défaut. Des tests ont été réalisés avec le paramètre "algorithm="elkan"" mais les résultats étaient sensiblement similaires.

La matrice de confusion avant normalisation présente les résultats suivants :

- La classe 2, 4, 5 ont bien été trouvées un bon nombre de fois (146, 145 et 173). La classe 1 a été bien identifié 37 fois ce qui reste assez peu.
- Les autres classes (0, 3 et 6) n'ont jamais bien été identifiées (toutes à 0 en diagonale)
- Beaucoup de données ont été identifiées en classe 1 ou 2 alors qu'elles appartenaient à d'autres classes

Le meilleur score obtenu avec normalisation est en utilisant "min max scaler". La matrice est alors la suivante :



Cette fois-ci, la matrice montre de meilleurs résultats (ce qui se retrouve dans le score de performance) :

- Les classes 2, 4 et 5 sont également bien identifiées, et les classes 3 et 6 ne sont plus à 0.

- Quelques éléments restent mal identifiés : notamment la classe 0 qui est souvent identifiée en tant que classe 3, la classe 6 identifiée comme 1 et la classe 1 comme classe 2.

- Aussi, aucune donnée n'a été identifiée comme appartenant à la classe 0, ce qui montre une mauvaise performance pour identifier cette classe.

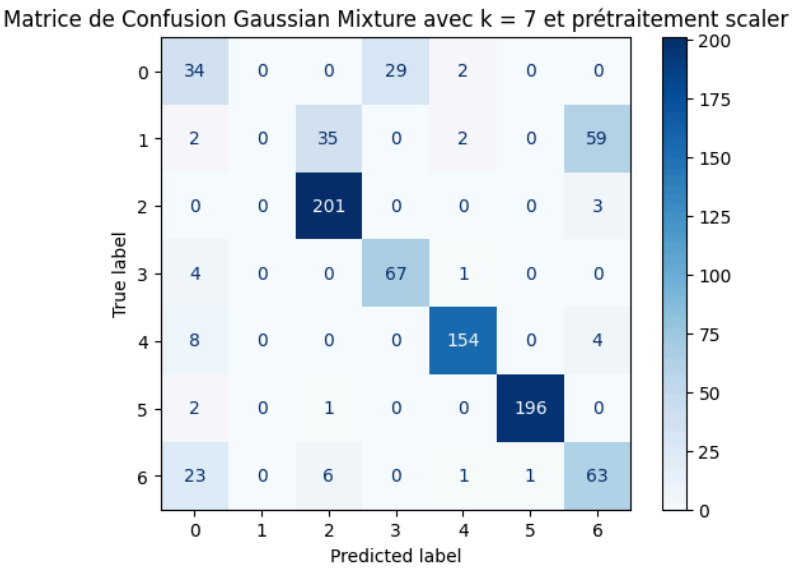
Après normalisation des données, les résultats de la méthode K-means sont donc bien meilleurs.

Méthode Gaussian Mixture

L'algorithme a été utilisé avant et après utilisation de la normalisation ("scaler"), ce qui a permis de voir une différence de score de performance notable.
Score K-means avant normalisation : 68%.
Score K-means après normalisation : 79,6% avec la méthode scaler "min max".

Pour les paramètres, de la même manière que pour K-means, le nombre de clusters (ici appelé "n_components") a été fixé à **7**. Des essais ont été réalisés avec 6 et 8 mais cela a fait baisser la performance. De la même manière, plusieurs essais ont été faits en variant le paramètre "covariance_type", et la meilleure performance a été obtenue avec le type "**full**".
Enfin, de la même manière que pour la méthode Kmeans, random_state a été fixé à 20.

La matrice de confusion après normalisation est la suivante :



Les résultats sont quelques peu similaires à la version K-means après normalisation:

- Les valeurs les plus élevées se trouvent dans la diagonale, ce qui représente toutes les données bien identifiées. Les classes les mieux identifiées sont encore 2, 4 et 5.
- Encore une fois, la classe 0 a quelquefois été identifiée comme classe 3 et la classe 1 comme classe 2. En plus, la classe 6 a été identifiée parfois comme la classe 0.
- Cette fois-ci la classe 0 a été identifiée quelquefois, mais aucune donnée n'a été classée dans la catégorie "1".

1.3) Comparaisons

Comparaison des deux méthodes supervisées

Sans standardisation, la méthode des k plus proches voisins (k-NN) produit un meilleur score que la méthode Bayésienne. Ce meilleur score est obtenu au détriment du temps de calcul. En effet, comme dit plus haut, afin de trouver la meilleure valeur de “k”, nous essayons toutes les valeurs possibles. Par conséquent, là où le temps de calculs pour la classification Bayésienne est d'environ 250 ms, celui-ci est d'environ 6 secondes pour la classification k-NN.

Cela dit, le temps de recherche du meilleur “k” est l’étape qui prend le plus de temps, sans cette étape les deux temps de calculs se valent. Il semblerait donc que si un individu privilégie la qualité de la classification, il choisira k-NN, tandis que s’il privilégie le temps de calcul il choisira la méthode Bayésienne. C’est une détermination qui doit être faite en considérant la taille des données, car si celle-ci est trop grande, essayer toutes les valeurs possibles n’est plus une méthode viable, et il faudra essayer de trouver le meilleur “k” avec d’autres méthodes.

Avec standardisation, les deux méthodes se valent.

Ainsi, si nous standardisons nos données, il vaut mieux utiliser la méthode Bayésienne, car cela sera vastement plus rapide que k-NN (recherche de meilleur “k”). Il faut cependant noter que cela n’est pas universel et dépendant des données que nous devons classifier.

Comparaison des deux méthodes non supervisées

Concernant les deux méthodes non supervisées (Kmeans et Gaussian Mixture), les résultats restent quelque peu similaires même si certaines mauvaises identifications ne se trouvent pas au même endroit. La différence de performance n’est pas très grande entre les deux. Gaussian Mixture a une meilleure performance que Kmeans sans la normalisation, mais avec normalisation, Kmeans est meilleur que Gaussian Mixture, en prenant en compte le paramètre “random_state” fixé à 20.

Lorsque l’on utilise random_state avec 51 (meilleure valeur pour Gaussian Mixture), les résultats sont les suivants :

- K-means sans normalisation : 53,8% / avec normalisation minmax : 79,3%
- Gaussian Mixture sans normalisation : 71,4% / avec normalisation minmax : 81,5%

On peut donc en déduire qu’en fonction des paramètres choisis, les deux méthodes ont à peu près les mêmes scores de performance.

Comparaison méthodes supervisées VS non supervisées

Après normalisation, les méthodes supervisées obtiennent de meilleurs résultats que les méthodes non supervisées. Cela s'explique par le fait que les méthodes de classification supervisées s'appuient sur des étiquettes de classes connues pour apprendre des modèles précis, en optimisant directement la séparation des classes alors que les méthodes non supervisées ne connaissent pas les classes, ce qui rend la séparation moins précise.

2) Classification des phonèmes avec prétraitement

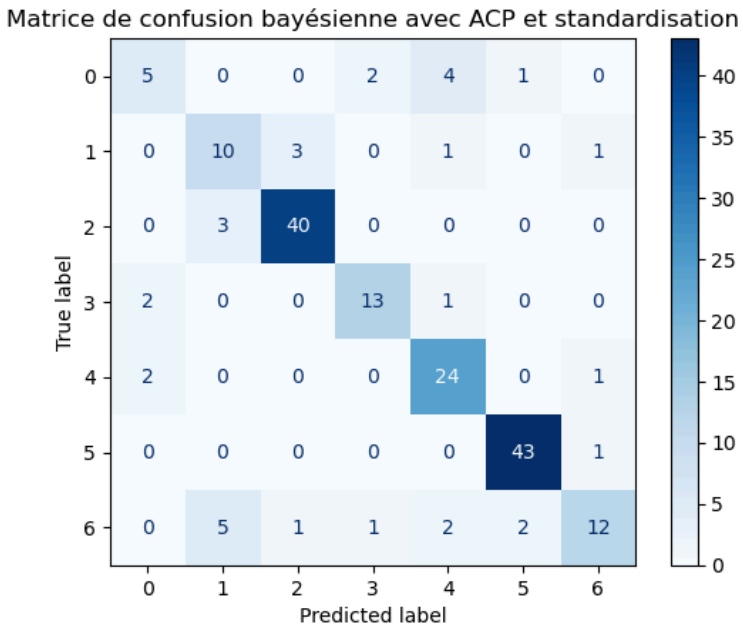
2.1) Méthodes supervisées

Encore une fois, les deux méthodes de classification supervisées utilisées sont la méthode Bayésienne et la méthode des k plus proches voisins. Après l'ACP, nous avons des données en 2 dimensions.

Méthode Bayésienne

Nous utilisons la même classe pour la classification Bayésienne. Nous obtenons les résultats suivants.
Score avec méthode Bayésienne avec ACP, ni standardisation : 60%.
Score avec méthode Bayésienne avec ACP, avec standardisation : 81.6%.

La meilleure matrice de confusion que nous obtenons est la suivante.



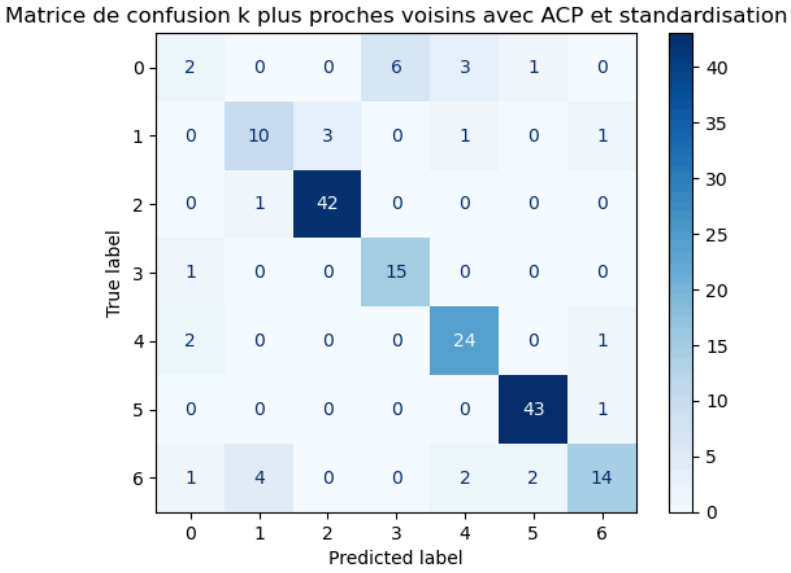
Nous pouvons observer que:

- Les classes 3, 5 et 6 sont également bien identifiées avec 3 prédictions incorrectes
- Les classes 0 et 2 sont relativement bien identifiées avec 4 prédictions incorrectes
- Finalement, les classes 1 et 4 sont les classes les moins bien identifiées avec 8 prédictions incorrectes
- En conclusion, ce sont de bons résultats.

Méthode des k plus proches voisins

Nous utilisons la même classe pour la classification des k plus proches voisins, ainsi que la même méthode d'obtention du meilleur k. Nous obtenons les résultats suivants.
Score avec méthode des k plus proches voisins avec ACP, sans standardisation : 63.3%.
Score avec méthode des k plus proches voisins avec ACP, avec standardisation : 83.3%.

La meilleure matrice de confusion que nous obtenons est la suivante.



Nous pouvons observer que:

- Les classes 2, 5 et 6 sont assez bien identifiées avec 3 prédictions incorrectes
- Les classes 0 et 1 sont assez mal identifiées avec 4 et 5 (respectivement) prédictions incorrectes
- Finalement, les classes 4 et 3 sont les classes les moins bien identifiées avec 6 prédictions incorrectes
- Toutes les erreurs de prédiction pour la classe 3 sont causées par la mis-classification des points vers la classe 0
- En conclusion, ce sont également de bons résultats.

2.2) Méthodes non supervisées

Les deux méthodes non supervisées utilisées sont toujours la méthode K-means et la méthode Gaussian Mixture. Deux types d'ACP ont été réalisés : ACP 2 dimensions et ACP 3 dimensions.

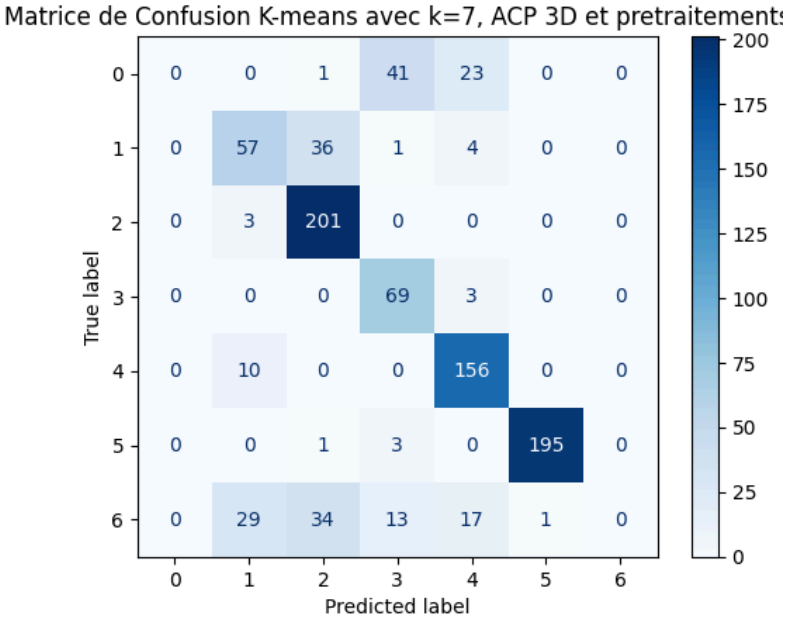
Méthode K-means

L'algorithme a été utilisé avant et après utilisation du normalisation, ce qui a permis de voir une différence de score de performance notable.

	ACP 2 dimensions	ACP 3 dimensions
Sans normalisation	56%	56%
Avec normalisation	73,5%	75,5%

Concernant les paramètres utilisés, ce sont les mêmes que pour la classification sans ACP. (Voir ici).

Les meilleures performances étant avec ACP 3 dimensions et après normalisation, nous allons analyser la méthode de confusion dans ce cas de figure.



- Les classes les mieux analysées sont encore une fois les classes 2, 4 et 5. Les classes 1 et 3 ont également bien été identifiées la plupart du temps.

- On remarque qu'aucune donnée n'a été identifiée comme appartenant à la classe 0 et 6. De nombreuses données de la classe (réelle) 6 ont été identifiées dans diverses classes, et toutes les données de la classe 0 ont été réparties dans les classes 4 et 5.

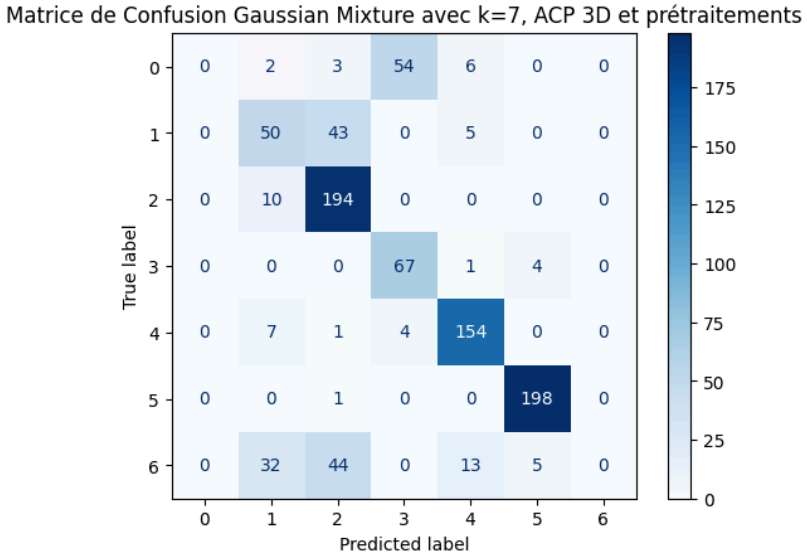
Méthode Gaussian Mixture

L'algorithme a été utilisé avant et après utilisation du normalisation, ce qui a permis de voir une différence de score de performance notable.

	ACP 2 dimensions	ACP 3 dimensions
Sans normalisation	61,8%	61,6%
Avec normalisation	73,5%	73,8%

Concernant les paramètres utilisés, ce sont les mêmes que pour la classification sans ACP. (Voir ici).

Les meilleures performances étant avec ACP 3 dimensions et après normalisation, nous allons analyser la méthode de confusion dans ce cas de figure.



- Les classes 2, 4 et 5 sont majoritairement bien identifiées.

- De la même manière que pour Kmeans, aucune donnée n'a été identifiée comme appartenant à la classe 0 et 6. Toutes les données appartenant réellement à la classe 0 ont été identifiées comme appartenant à la classe 3 et les données de la classe 6 ont été partagées entre la classe 1 et 2.

2.3) Comparaisons

Comparaison des deux méthodes supervisées

Contrairement à la classification pré-ACP, nous n'obtenons de majeures différences entre la classification Bayésienne et la classification k-NN, ni pré, ni post-standardisation. En effet, la classification k-NN n'est que marginalement meilleure (par moins de 3%).

Il semblerait donc qu'avec les données que nous avons, si la qualité de la précision est une priorité absolue, alors nous choisirons la méthode k-NN tandis que si nous pouvons nous permettre quelques pourcentages d'erreurs, le choix de la méthode Bayésienne peut nous faire gagner une quantité non-négligeable de temps de calculs.

Nous remarquons également que l'utilisation de l'ACP réduit la précision de la classification. Cela est sûrement dû à la perte d'information, peut-être que de garder plus de dimension nous permettrait de réduire la quantité de données à traiter, tout en maintenant une précision maximale.

Comparaison des deux méthodes non supervisées avant et après ACP

De la même manière que lors de la première comparaison de Kmeans et Gaussian Mixture, on remarque que les deux ont similairement les mêmes scores, même si ici Gaussian Mixture est légèrement meilleur.
En comparant les matrices de confusion avec et sans utilisation d'ACP, on remarque une perte d'information qui a pour conséquence que les données ne sont pas identifiées dans certaines classes.