Projet SY09

Julia CAVICCHI Bilel DEBBABI Pierre ROMON

12 juin 2020

1 Introduction

Dans le cadre du projet de SY09 nous avons été amenés à analyser un jeu de données portant sur les caractéristiques biomécaniques de patients orthopédiques. Pour cela une première phase d'étude a consisté en l'observation des données. Dans un second temps et après avoir appliqué un prétraitement sur l'échantillon à disposition, nous avons cherché à prédire la maladie (et donc la classe) d'un individu. Tout au long de ce rapport nous détaillerons les corrélations entre les différentes variables, ainsi que les caractéristiques qui déterminent l'appartenance à une classe.

2 Analyse des données

2.1 Présentation des données

Nous avons à notre disposition 2 fichiers *CSV*. Ils contiennent tous les deux les mêmes individus et les mêmes variables descriptives. On compte 310 patients, décrits par 6 caractéristiques quantitatives :

- *pelvic tilt* (bascule pelvienne) : angle d'inclinaison du bassin.
- *sacral slope* (pente sacrée) : angle entre la vertèbre supérieur S1 et une ligne horizontale.
- *pelvic incidence* (incidence pelvienne) : somme de *pelvic tilt* et *sacral slope*.
- *lumbar lordosis angle* (angle de lordose pelvienne).
- *pelvic radius* (rayon pelvien).
- *degree spondylolisthesis* : degré de glissement vers l'avant d'une vertèbre par rapport à celle située juste en dessous.

On dispose également d'une 7ecolonne qualitative *class* qui diffère entre les deux fichiers. Dans le fichier *column_3C_weka.csv* cette dispose de 3 modalités : *Normal*, *Hernia* et *Spondylolisthesis*. En revanche dans le fichier *column_2C_weka.csv* les deux maladies sont regroupées. La colonne répartit donc les individus en 2 classes : *Normal* et *Abnormal*.

Ces données proviennent de l'Université de Californie (School of Information and Computer Science).

Nous avons choisi de travailler majoritairement avec le jeu de données regroupé en 3 classes. En effet, c'est celui qui contient le plus d'informations, et donc celui à partir duquel nous serons plus à même de prédire la maladie d'un individu. Par ailleurs il est possible de passer à une classification à 2 classes si les conclusions de nos analyses nous amènent en ce sens.

2.2 Analyse exploratoire des données

2.2.1 Répartition des individus

On cherche à trouver la proportion des individus appartenant à une certaine classe au sein du jeu de données. Pour cela, on utilise un diagramme en bâtons (figure 1). A noter qu'on aurait également pu utiliser un diagramme circulaire afin d'accentuer la notion de proportion au lieu de celle de quantité ici mis en évidence. Il apparaît donc clairement qu'il y a plus d'individus de la classe *Abnormal* (210 individus) que*Normal* (100 individus), ces individus étant répartis en majorité dans la classe *Spondylolithesis* au détriment de la classe *Hernia* (150 contre 60 individus).

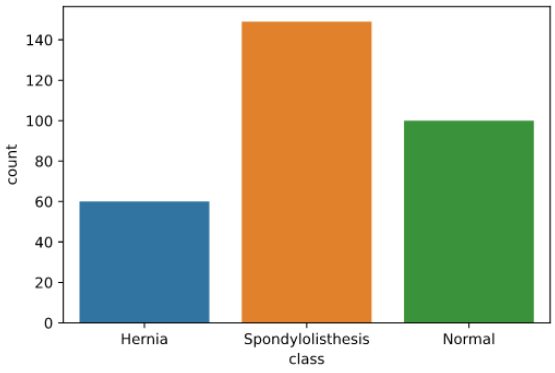

FIGURE 1 – Répartition des individus par classe

2.2.2 Répartition des variables

On s'intéresse ensuite à la répartition des variables et à leur valeur au sein des différentes classes. On effectue ici un pivotage du jeu de données du format large au format long sur la composante *class*. Ce pivotage permet de faire facilement des analyses bidimensionnelles impliquant la composante de pivot.

Avec ce jeu de données au format large, on produit des boîtes à moustache (une par classe) des différentes composantes (figure 2). Ces boites à moustache permettent de comparer la répartition de chaque variable selon les classes d'appartenance.

Globalement on remarque pour plusieurs variables une distinction des valeurs de la classe *Spondylolithesis* (dont la médiane est généralement plus élevée que celle des autres classes sauf pour le *pelvic radius*). Par contre, les valeurs de *Hernia* et *Normal* sont souvent proches. La répartition des différentes variables ne permet pas de distinguer ces deux classes ce qui justifie notamment une approche multivariée.

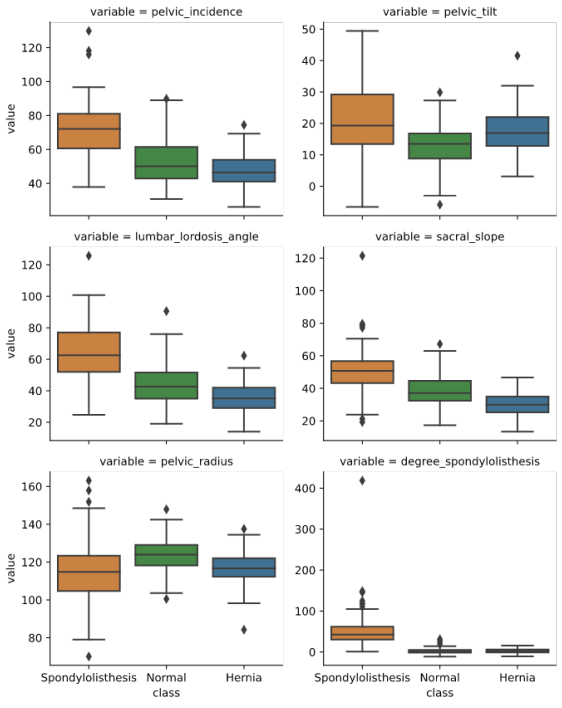

FIGURE 2 – Répartition des variables

2.2.3 Correction des valeurs aberrantes

On remarque dans la figure précédente (figure 2) une valeur totalement aberrante au niveau des composantes *degree spondylolisthesis* et *sacral slope*.

Il se trouve que c'est le même individu qui possède ces deux valeurs. On a donc décidé de le supprimer du jeu de données. On aurait pu lisser le bruit en remplaçant les valeurs aberrantes par la moyenne ou la médiane de la variable correspondante par exemple. Mais puisqu'un seul individu est concerné, cette suppression n'a pas de grandes conséquences sur les résultats. A noter qu'à partir de cette étape, le jeu de données considéré compte 309 individus.

2.2.4 Répartition des variables 2 par 2

En comparant les composantes entre elles (2 par 2), il est possible d'effectuer une analyse dimensionnelle complète. On obtient alors une matrice recensant les nuages de points des composantes 2 par 2 (figure 3).La diagonale de cette matrice contient les histogrammes de ces composantes.

Il est possible de distinguer 2 clusters : *Hernia-Normal* et *Spondylolisthesis*. Par contre distinguer les classes *Normal* et *Hernia* est une tâche plus ardue. On remarque cependant certaines différences (et donc une possibilité de clustering) sur certaines composantes, notamment *sacral slope*.

On peut également constater une corrélation entre certaines variables (*pelvic incidence* avec *pelvic tilt*, *lumbar lordosis angle* et *sacral slope*). La variable *pelvic radius* ne semble en revanche corrélée à aucune autre.

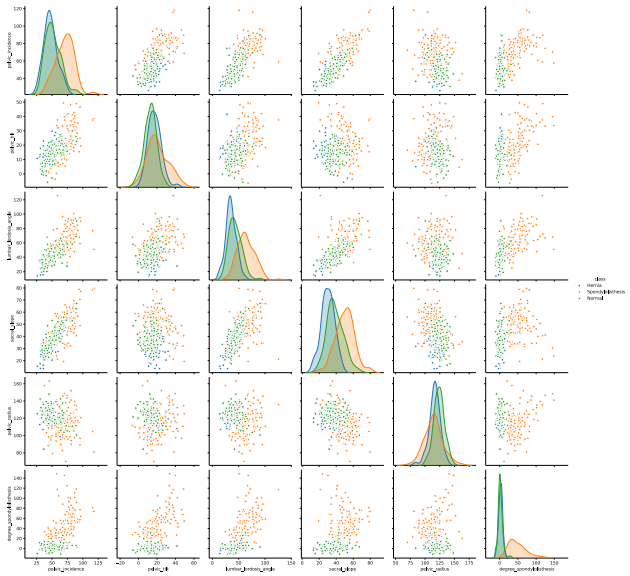

FIGURE 3 – Répartition des variables 2 par 2

2.2.5 Corrélation des variables

Il s'agit maintenant trouver des relations plus globales sur l'ensemble des composantes.Il est intéressant de faire apparaître une matrice de corrélation (figure 4) afin de déterminer plus précisément ces relations. Pour faciliter l'analyse, on utilise une heatmap (ou carte thermique) afin d'obtenir un rendu visuellement explicite.

On remarque alors que la composante *pelvic radius* est indépendante des autres composantes (ce qui étaye les observations réalisées suite à la comparaison des variables deux à deux).On observe par ailleurs une forte corrélation entre les composantes *sacral slope* et *pelvic incidence*, ainsi qu'entre *pelvic incidence* et *lumbar lordosis angle*.

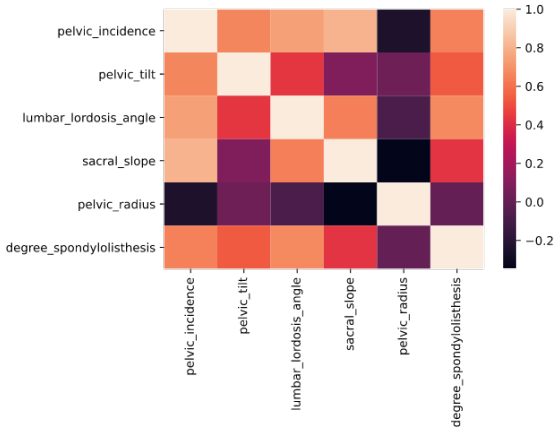

FIGURE 4 – Corrélation entre les variables

Par définition, on sait que *pelvic incidence* est égal à la somme entre *sacral slope* et *pelvic tilt*. Afin de nous assurer que les données à notre disposition respectent bien cette contrainte, nous avons calculé le coefficient de Pearson entre la variable *pelvic incidence* et la variable nouvellement créée *sacral slope + pelvic tilt*. Ce coefficient étant égal à 1, cela valide l'hypothèse énoncée précédemment.

3 Prétraitements par ACP

Comme on vient de le voir, il existe une corrélation entre plusieurs variables. Il est donc intéressant de diminuer le nombre de variables afin de supprimer les informations redondantes et obtenir de nouvelles variétés linéaires proches du nuage initial. L'Analyse en Composantes Principales est une méthode adaptée à notre situation puisque l'on dispose d'un tableau individu-variables comportant des valeurs quantitatives.

On applique ainsi l'ACP sur le jeu de données auquel on a retiré l'information de classe (on obtient dès lors un jeu de données uniquement quantitatif). On récupère l'inertie expliquée par les composantes principales obtenues (en pourcentage). A noter qu'on obtiendrait le même résultat avec le fichier *column_2C_weka.csv* qui ne diffère que par le nombre de classes étudiées.

Afin de déterminer le nombre d'axes à retenir, on applique la méthode du coude (figure 5). On remarque que le 1eraxe (68.5% d'inertie), et le 2^{e}axe (14.6%) expliquent une grande partie de la répartition des données (83.1%).

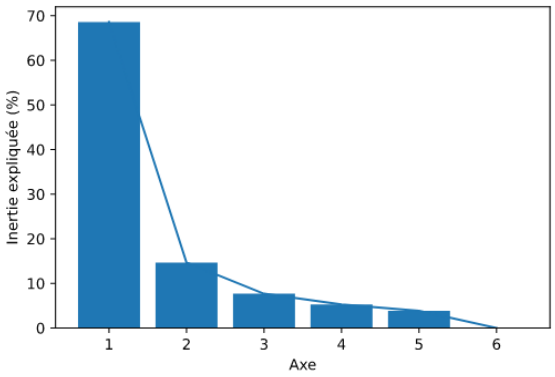

FIGURE 5 – Méthode du coude

Sur la figure, on repère visuellement un lieu de rupture après la seconde composante : on choisit donc une représentation dans le premier plan factoriel (figure 6).

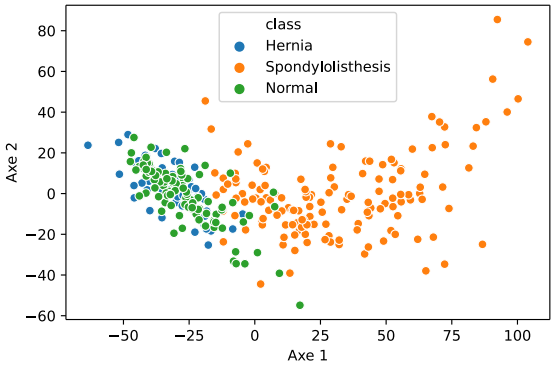

FIGURE 6 – Visualisation de l'ACP à 2 composantes

Encore une fois il est possible de distinguer 2 clusters : *Hernia-Normal* et *Spondylolisthesis*. Cela corrobore les observations préalables réalisées sur le jeu de données, mais ne nous permet pas se séparer clairement les classes *Hernia* et *Normal*.

Le cercle de corrélation (figure 7) nous permet d'analyser plus en détail les composantes principales obtenues.

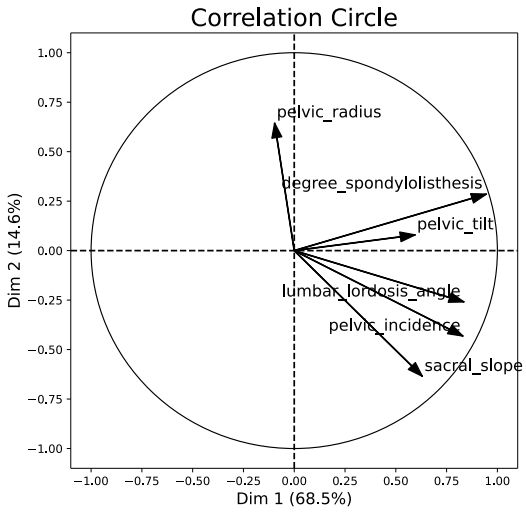

FIGURE 7 – Cercle de corrélation

D'une part, on remarque que la Dimension 2 porte majoritairement l'information du *pelvic radius*. Cela confirme une fois encore l'indépendance de cette variable par rapport aux autres et donc le fait qu'on ne puisse pas l'expliquer par une combinaison des autres descripteurs. La Dimension 1 quant à elle regroupe l'information de l'ensemble des autres variables. On retrouve notamment la corrélation entre les variables *pelvic incidence*, *sacral slope* et *lumbar lordosis angle* relevée précédemment. L'information de cet axe est principalement portée par la variable *degree spondylolisthesis*, associée à la classe de même nom.

On peut s'interroger sur les conséquences d'une possible sur-représentation de cette classe suite à l'ACP, qui atténuerait les différences visibles entre Hernia et Normal lors de la représentation des clusters. Afin d'éprouver cette hypothèse, nous avons réitéré l'ACP en enlevant cette fois la variable *degree spondylolisthesis* du jeu de données. Après représentation selon les deux premiers axes factoriels (figure 8)- expliquant cette fois 82% de l'inertie - on constate que les individus de la classe spondylolisthesis sont moins regroupés qu'auparavant, mais que ceux des classes Hernia et Normal sont toujours mêlés. On abandonne donc cette hypothèse non concluante.

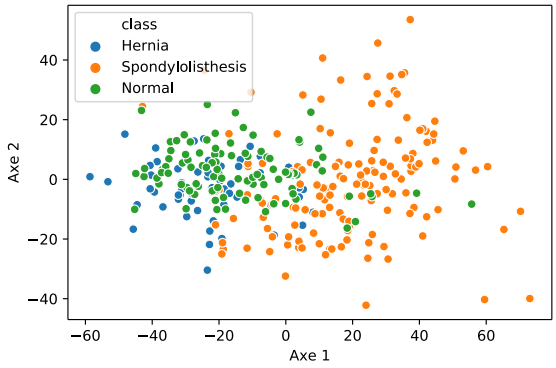

FIGURE 8 – Visualisation de l'ACP à deux composantes sans la variable degree spondylolisthesis

4 Méthodes non-supervisées

4.1 Formalisation du problème

Après exploration des données et mise en évidence de certaines corrélations entre les variables, on constate que dans les différentes représentations du jeu de données, la classe *Spondylolisthesis* se démarque tandis que *Normal* et *Hernia* se confondent. Le but de cette section est de trouver une méthode permettant de prédire de manière efficace l'appartenance d'un individu à l'une de ces classes.

On considère donc une population de 309 individus, chacun décrit par 6 caractéristiques qui correspondent à un vecteur forme. La classe correspond quant à elle à la variable à prédire. On souhaite ainsi déterminer la maladie d'un individu à partir de ses caractéristiques.

4.2 KMeans

Tout d'abord on applique la méthode non supervisée des centres-mobiles (ou K-means), possible grâce à la nature quantitative des variables à notre disposition. Notre connaissance du jeu de données facilite cette recherche de partition car on connaît déjà le nombre réel de classes du jeu de données. On applique donc les K-means successivement pour deux et trois clusters.

Pour K=2, on observe une classe réunissant Hernia et Normal, tandis que la plupart des individus de la classe Spondylolisthesis sont regroupés ensemble (figure 9).

Pour K=3, on constate que la partition obtenue est très différente de la partition réelle (alors que le nombre de classes correspond cette fois au nombre réel). En effet, la classe Spondylolisthesis est scindée en deux clusters tandis que les classes Normal et Hernia sont une

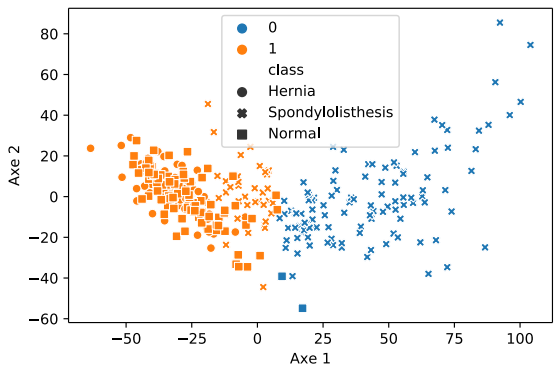

FIGURE 9 – Classification d'après la méthode des k-means avec K=2

fois encore regroupées en une unique classe (figure 10).

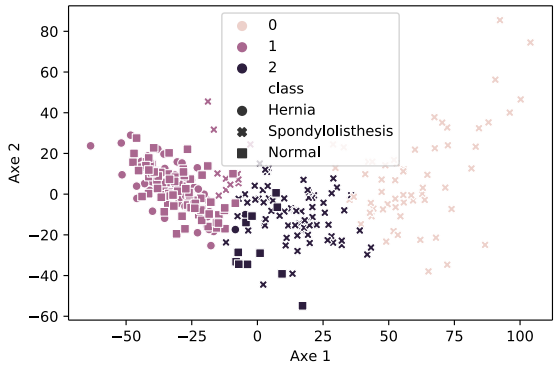

FIGURE 10 – Classification d'après la méthode des k-means avec K=3

Le calcul de l'indice de Rand ajusté confirme ces observations puisqu'on obtient ARI = 0.3085.

On peut donc en conclure que la méthode des K-Means est peu concluante aux vues de ce jeu de données. Nous ne sommes pas en mesure de classer correctement les individus des classes Normal et Hernia. Étant donné la répartition des clusters, il ne nous semble pas intéressant d'approfondir une démarche non supervisée. Nous faisons donc le choix de ne pas appliquer la méthode des K-means adaptatifs et de nous concentrer d'avantage sur les méthodes supervisées.

5 Méthodes supervisées

5.1 Méthodologie générale

Ayant constaté des résultats peu concluants avec la méthode de classification des K-Means, le reste de cette étude se focalise sur des méthodes d'analyse supervisées. On sépare ainsi le jeu de données initial en deux sous-ensemble : l'ensemble d'apprentissage et l'ensemble de test (contenant respectivement 2/3 et 1/3 des individus). Il s'agit de faire l'apprentissage des données sur divers modèles puis d'en calculer *l'accuracy score* afin de comparer les performances obtenues.

Compte tenu de la nature des données à notre disposition, nous pourrions définir une fonction de coût qui rendrait plus importantes les conséquences d'une mauvaise classification des individus malades (par exemple classer un individu possédant un hernie dans la classe *Normal*). Toutefois, afin de faciliter la comparaison des performances des différents modèles et puisqu'il s'agit là d'un cas d'étude nous avons décidé de considérer des coûts 0,1. Le calcul du risque relatif à chaque classifieur considéré revient donc à calculer les performances de ces mêmes classifieurs.

Enfin, on peut noter que tout au long de cette analyse, le choix d'un meilleur modèle se fonde sur la méthode de validation croisée appliquée sur notre ensemble d'apprentissage initial (qui sera donc scindé entre un autre ensemble d'apprentissage et un ensemble de validation).

5.2 Analyse discriminante

L'une des méthodes supervisées que nous avons appliquées dans le cadre de ce projet est l'analyse discriminante, particulièrement populaire dans le domaine de la biologie. A travers cette méthode on compare 3 modèles d'analyse discriminante : l'analyse discriminante linéaire, quadratique et le classifieur bayésien naïf.

5.2.1 Hypothèse de normalité

Il est important de noter que ces modèles fournissent de meilleures solutions lorsque les données suivent une loi normale multidimensionnelle. Toutefois il est possible de les utiliser lorsque cette hypothèse n'est pas vérifiée.

Ainsi, on commence par tester l'hypothèse de normalité des données conditionnellement à chacune des classes. On utilise pour cela le test de Shapiro-Wilk. On considère un niveau de signification α^* égal à 0.05. On obtient des *p-values* très faibles (respectivement

1.5846e-21, 2.6126e-18 et 2.2085e-11). On rejette donc l'hypothèse par défaut de normalité conditionnellement à la classe. On peut conclure avec un niveau de confiance assez fort que les données ne suivent pas une loi normale.

En second recours et afin d'améliorer ces résultats, on cherche à normaliser les données avec un prétraitement de standardisation, par exemple *StandardScaler* (centrage et réduction). En renouvelant le test de Shapiro sur les données standardisées, on obtient des *p-values* plus élevées. Cependant, dans le cas des classes *Normal* et *Spondylolisthesis*, ellent restent inférieures à 0.05 ce qui nous amène une fois encore à rejeter l'hypothèse de normalité. A l'inverse pour *Hernia* on obtient *p-value = 0.123* : on ne peut pas rejeter l'hypothèse nulle de normalité.

Le tableau suivant répertorie les *p-values* pour chaque classe avant et après l'application du standardiseur.

TABLE 1 – p-values calculées par le test de Shapiro

Prétraitement	Normal	Hernia	Spondylolisthesis
Aucun	1.58e-21	2.61e-18	2.21e-11
StandardScaler	4.11e-9	0.12	3.06e-7

On déduit de ces résultats que les données ne vérifient pas l'hypothèse de normalité (mise à part pour la classe *Hernia* après prétraitement). Comme mentionné précédemment, cela n'interdit pas le recours à cette méthode mais retournera probablement des résultats moins satisfaisants.

5.2.2 Prétraitements

Il s'agit ici de déterminer les prétraitements à appliquer aux données d'apprentissage pour obtenir des résultats optimaux. On remarque tout d'abord que les différents prétraitements de standardidation n'influent pas sur les résultats. On choisit donc de ne pas appliquer une telle transformation.

Un autre élément à prendre en compte est la sensibilité de l'analyse discriminante à la colinéarité. Il faut en effet retirer la composante *pelvic-incidence* (qui, comme mentionné précédemment, est source de redondance) pour optimiser les résultats du modèle. Ce prétraitement est d'autant plus nécessaire que le classifieur bayésien naïf a pour hypothèse l'indépendance des variables conditionnellement à chacune des classes. En retirant la variable *pelvic-incidence*, on augmente les chances d'obtenir des résultats satisfaisants pour ce classifieur.

Par ailleurs, l'analyse quadratique étant sensible au nombre de composantes, on peut tenter de les faire varier grâce à une ACP sur les données. Les résultats de cette variation sont synthétisés dans le tableau 2.

TABLE 2 – Score de validation de l'analyse discriminante selon le nombre de composante de l'ACP

Modèles	Linaire	Quadratique	Gaussien
Brute	83.88 %	84.48 %	83.90 %
ACP 4	83.38 %	84.35 %	83.38 %
ACP 3	80.90 %	83.88 %	80.00 %
ACP 2	72.30 %	70.38 %	69.45 %
ACP 1	72.21 %	72.21 %	70.29 %
Scaler	83.88 %	84.48 %	83.90 %
NCA	83.88 %	84.48 %	83.90 %

La standardisation des données a montrée via le test de Shapiro plus haute qu'elle permettait une certaine normalité du jeux de donnée. On pourrait alors penser qu'appliquer un *scaler* comme prétraitement pour une analyse discriminante permettrait d'obtenir des résultats plus satisfaisants. Cependant, la production des scores des discrimination annule tout effet de standardisation. En effet, la standardisation d'un ensemble de données ne modifie pas les valeurs propres de la matrice associée. La discrimination inter-classe qui se base sur ces valeurs propres reste alors inchangée. Il est donc ni utile ni pertinent d'appliquer une standardisation comme procédé de prétraitement. Il en va de même pour l'analyse de voisinnage.

5.2.3 Résultats

Au terme de ces diverses expérimentations, on remarque que l'on obtient de manière générale de meilleurs résultats à partir des données brutes, ce qui est intuitivement logique puisque l'ACP est susceptible de causer des pertes d'informations. C'est en particulier à partir des données brutes et en utilisant le modèle d'analyse discriminante quadratique qu'on obtient le meilleur score de validation croisée sur l'ensemble d'apprentissage. Nous avons évoqué précédemment le fait que l'analyse discriminante quadratique était sensible à la dimension de l'espace. En effet, plus le nombre de variables augmente, plus le nombre de paramètres à estimer croît. Dans notre cas, et malgré le peu d'individus d'apprentissage dont nous disposons, le nombre de variable est suffisamment faible pour limiter les erreurs d'estimation.

A partir de ce modèle optimal, on prédit les labels de l'ensemble de test que l'on compare aux classes réelles : on obtient un score de test égal à 79.61.

On peut enfin noter que pour l'analyse linéaire, c'est l'ACP à 4 composantes qui donne le meilleur résultat.

5.3 KNN

Nous poursuivons l'analyse avec la méthode des K-nearest-neighbors (ou KNN) qui consiste à affecter un individu x à la classe la plus représentée parmi celle de ses K plus proches voisins.

5.3.1 Détermination des meilleurs paramètres

Tout d'abord, il est important de déterminer le nombre de voisins K qui optimisera les prédictions (c'est-à-dire tel que le *accuracy score* soit maximisé). Pour cela on effectue un *GridSearchCV* sur un nombre de voisins variant de 1 à 100. On essaye également de choisir le meilleur paramètre *weight* du KNN entre *distance* et *uniform*. Le *GridSearchCv* utilise la validation croisée *StratifiedKFold*, ce qui permet de conserver la répartition d'origine des classes dans chaque Fold. La validation croisée est privilégiée ici par rapport à une validation simple répétée qui engendrerait un risque de sous ou sur-représentation de certains exemples dans l'ensemble d'apprentissage ou de validation.

On doit également déterminer si un prétraitement sur les données permet d'optimiser le partitionnement obtenu. On peut en effet imaginer appliquer la méthodes des KNN sur les données brutes ou transformées (normalisation, ACP, Analyse discriminante linéaire (ADL) Neighborhood Components Analysis(NCA)). Une autre hypothèse consiste en la suppression de la colonne *pelvic_ incidence* qui, étant la somme de *pelvic tilt* et *sacral slope* introduit de la redondance.

Pour chacun de ces ensembles initiaux de données, on applique la méthode des KNN puis on retourne le meilleur score de validatation obtenu (via la fonction cross-val-score) et la valeur k associée à ce score. Le tableau 1 présente la *validation accuracy* maximale obtenue avec différents prétraitements.

D'autres combinaisons de paramètres ont également été testées (telles qu'une variation dans le nombre de composantes pris en compte dans l'ACP) mais nous avons décidé de ne conserver que les valeurs intéressantes dans le cadre de notre étude.

On peut voir que c'est en utilisant les données transformées avec un *Neighborhood Components Analysis* à deux composantes, avec un *Standard Scaler* et avec *distance* comme *weights* que l'on obtient la meilleure *accuracy* (90.74%), pour *k=14* (figure **??**).

TABLE 3 – Meilleur score de validation par méthode de prétraitement

Méthode	Meilleure accuracy	K	weights
Données brutes	85.42 %	5	distance
MinMaxScaler	81.98 %	9	distance
StandardScaler	81.53 %	17	distance
ACP (2)	79.59 %	4	distance
NCA	86.85 %	1	uniform
NCA (2)	89.76 %	6	distance
ADL	86.85 %	1	uniform
ADL (2)	89.76 %	6	distance
Sans *pelvic_ incidence*	86.87 %	9	distance
Sans *pelvic_ incidence* + NCA (2)	89.77 %	1	uniform
StandardScaler + NCA (2)	**90.74 %**	**14**	**distance**
StandardScaler + ADL (2)	87.34 %	48	uniform
MinMaxScaler + NCA (2)	90.26 %	85	distance
MinMaxScaler + ADL (2)	87.34 %	48	uniform

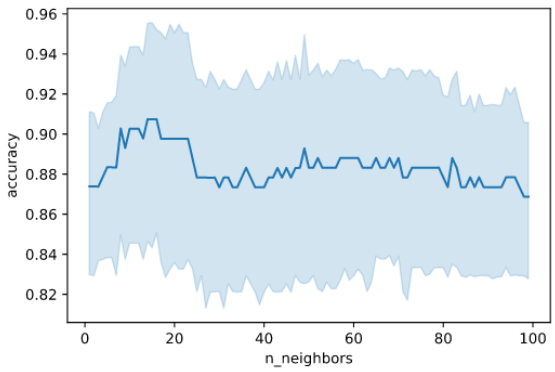

FIGURE 11 – Résultat de la validation avec différents nombres de voisins

5.3.2 Analyse des résultats

La plupart des algorithmes de Machine Learning nécessitent une mise à l'échelle des données visant à améliorer les performances et les résultats obtenus. Dans le cas du KNN, une normlisation des données est censée améliorer les résultats. En effet, cette méthode se base sur une mesure de la distance, ce qui peut aboutir à de mauvaises performances si on dispose de variables avec des unités et des ordres de grandeur différents. Nous avons donc choisi d'appliquer les techniques de MinMaxScaler et StandardScaler fournies par Scikit-Learn. La première consiste à transformer les caractéristiques en les adaptant à une plage donnée (ici [0, 1]). La deuxième transforme les données en enlevant la moyenne et en divisant par la variance. On constate néanmoins que les résultats obtenus après normalisation sont moins bons que ceux obtenus à partir des données brutes. Dans notre cas la plupart des variables sont des angles, leurs unités et ordres de grandeur sont donc semblables ce qui peut rendre une telle normalisation inutile. Néanmoins, cela n'explique pas l'obtention de moins bons résultats.

En revanche, il n'est pas surprenant de voir des résultats assez faibles pour l'ACP, puisque les axes selectionnés n'expliquent pas toute la répartition des données. Il serait intéressant d'utiliser l'ACP, et donc de sacrifier un peu de précision, si on avait un nombre de variables initiales important, et que l'algorithme prenait trop de temps à s'exécuter. Or dans notre cas on dispose de 6 variables uniquement ce qui rend l'exécution du KNN rapide. On abandonne donc ce prétraitement dans le cadre de cette méthode.

En plus de l'ACP, nous avons identifié deux autres méthodes de réduction de la dimension : l'Analyse discriminante linéaire (ADL) et la Neighborhood Components Analysis(NCA). Contrairement à l'ACP, ces deux méthodes sont des méthodes de prétraitement supervisées. La NCA permet d'apprendre une métrique de distance qui améliore les performances du KNN par rapport à la distance euclidienne classique. L'algorithme maximise le score d'un *Leave-one-out-knn*, (où K=N-1 avec N le nombre de points). Il permet ainsi de transformer les données et de réduire le nombre de dimensions. L'ADL quant à elle permet de réduire le nombre de dimensions en maximisant la séparation des classes (en se basant sur la variance entres celles-ci). On remarque que la NCA et la LDA sont plus performantes avec une réduction de dimension.On peut voir également que la NCA combinée au KNN est plus performante que la LDA, dans le cas où on applique une normalisation des données préalable.

Par ailleurs, notre intuition sur la redondance des données causée par *pelvic_incidence* s'avère être correcte puisque l'on obtient de meilleurs résultats en enlevant cette variable du jeu de données.

Nous avons également vérifié l'impact du paramètre weights du KNN. Par défaut, il prend la valeur *uniform*. Cependant, avec la valeur *distance*, le KNN donne un poids plus important aux points les plus proches. On a remarqué que dans certains cas la modification de la valeur par défaut permettait d'améliorer les performances.

5.3.3 Analyse du meilleur KNN

En appliquant le meilleur modèle à l'ensemble de test on obtient une *test accuracy* de 81.55 %.

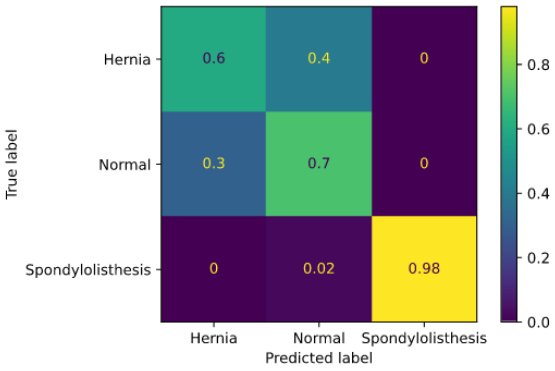

FIGURE 12 – Matrice de confusion pour le KNN

Il est également intéressant d'analyser le score par classe, puisqu'on pouvait voir avec l'ACP (figure 6) qu'il n'y avait qu'une seule classe qui était correctement séparée des autres. On calcule pour cela la matrice de confusion (figure **??**) relative au meilleur modèle appliqué à l'ensemble de test. On peut voir que la classe *Spondylolisthesis* est très bien prédite (98%), contrairement à *Hernia* (60%) qui est souvent (40%) confondue avec *Normal* (70%).

La frontière de décision pour le meilleur KNN est représentée dans la figure **??**. On remarque tout d'abord que la NCA permet de mieux séparer les classes, par rapport à l'ACP (figure 6) ce qui justifie notamment les meilleurs résultats obtenus suite à ce prétraitement. On peut également distinguer ce qui semble être de *l'overfitting*. On peut par exemple voir une région qui est considérée comme *Normal* à cause d'un seul point dans la région *Hernia*. Cela peut être expliqué par le fait que le paramètre$K$ n'est pas très élevé (5), mais surtout parce qu'on utilise *distance* pour le paramètre *weights*. Ainsi, si un point est excentré par rapport aux autres, il aura plus de poids dans la classification de la région

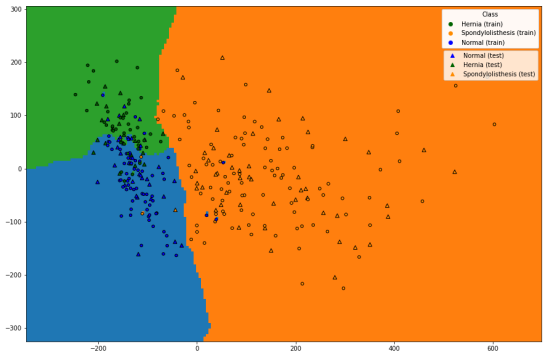

FIGURE 13 – Frontière de décision pour le KNN

autour de lui. Ce qui est bien le cas dans notre exemple.

Par ailleurs, on a représenté les points de l'ensemble de test (points en forme de triangle). Ils correspondent aux vraies classes, et non pas aux prédictions. Les afficher nous permet notamment de comprendre la différence constatée entre la test et la validation accuracy. On peut ainsi voir plusieurs points de *Hernia* de l'ensemble de test qui sont mélangés aux points de *Normal* et inversement. Ces points n'ont donc pas été pris en compte lors de l'apprentissage et ne sont donc pas du "'bon coté"' de la frontière de décision. Cette hypothèse est confirmée si on modifie la division entre les ensembles d'entraînement et de test, puisqu'on obtient une test accuracy différente en fonction du split initial.

6 Regression logistique

Une méthode de classification supervisée souvent utilisée pour la détection des maladies est la régression logistique. Dans notre cas il s'agit d'appliquer une régression logistique multinomiale (en raison des 3 classes à prédire) sur notre jeu de données.

6.1 Selection des prétraitements

Comme pour la méthode du KNN, il s'agit de sélectionner le prétraitement à appliquer sur l'ensemble d'apprentissage pour optimiser les performances du modèle. On peut par exemple étudier la méthode de prétraitement supervisée du Neighborhood Components Analysis détaillée dans la section relative au KNN. On procède ainsi à une réduction de la dimension à 2 composantes (qui s'était avérée concluante pour la méthode précédente). Une autre méthode de réduction envisagée est l'ACP à 2 composantes également.

On s'interroge aussi sur l'influence d'une standardisation des données via les fonction MinMaxScaler ou StandardScaler. Enfin une des conditions d'application de cette méthode est la non-colinéarité des variables. On calcule donc le score d'apprentissage après avoir supprimé la colonne *pelvic incidence*. Ces différents prétraitements sont appris successivement par le module de régression logistique au moyen d'un GridSearchCV et d'un pipeline. Les paramètres utilisés ici sont ceux par défaut.

TABLE 4 – Meilleur score de validation par méthode de prétraitement

Prétraitement	Score de validation
Aucun	83.47 %
Sans pelvic incidence	85.40%
StandardScaler	83.47%
MinMaxScaler	82.52%
NCA (2 composants)	84.42%
ACP2	73.73%

Le tableau 1 montre que la standardisation et la mise à l'échelle des données est inutile dans le cadre de cette méthode. Par ailleurs, le score d'apprentissage sur les données obtenues par ACP est nettement inférieur à celui obtenu à partir des données brutes (probablement car l'ACP n'explique pas toutes les informations portées par les données initiales). On rejette donc ce prétraitement. En revanche, et comme les hypothèses laissaient présager, la suppression de la variable *pelvic incidence* améliore les résultats. De même, on constate des résultats intéressants suite à une Neighborhood Components Analysis. On constate ainsi que parmi toutes les combinaisons de pré-traitement des données, le meilleur score d'apprentissage (85.90%) est obtenue à partir des données auxquelles on a appliqué successivement une suppression de la colonne *pelvic incidence*. et une NCA à 2 composantes.

6.2 Choix des paramètres de régression et des ensembles d'apprentissage

Un des paramètres du modèle intéressant dans le cadre d'une régression logistique est le "'solver"'. Il en existe 5 :

— *newton-cg* : la méthode de Newton calcule la matrice héssienne exacte. Cette exactitude dans les calculs peut néanmoins entraîner des coûts im-

portants pour les jeux de données de taille conséquente (ce qui n'est pas notre cas).
— *lbfgs* : cette méthode, similaire à la précédente consiste à approximer les dérivées secondes. Il s'agit de la méthode par défaut et donne des résultats satisfaisant dans la plupart des cas.
— *liblinear* : l'algorithme minimise une fonction multivariée en résolvant de manière itérative des problèmes d'optimisation univariés. En revanche cette méthode ne peut pas apprendre de modèle multinomial, se contentant de décomposer le problème en "'une classe VS le reste"'. Nous décidons donc d'ignorer ce solver.
— *sag* (Stochastic Average Gradient descent) : c'est une variation de l'algorithme du gradient qui utilise un échantillon aléatoire de gradients précédemment calculés.
— *saga* : ce solver est une extension de la méthode sag.

A l'aide du module GridSearchCV (qui effectue une validation croisée sur 5 plis), on teste les différents solver énumérés ci-dessus sur le prétraitement optimal introduit précédemment.

TABLE 5 – Meilleur score de validation par solver

Solver	Score de validation
newton-cg	85.90%
lbfgs	85.90 %
sag	76.19%
saga	76.19%

6.3 Analyse des résultats

Les meilleurs résultats sont obtenus avec les méthodes *lbfgs* et *newton-cg*. Cela n'est pas suprenant pour le *newton-cg* qui calcule la matrice hessienne exacte. La méthode *lbfgs* est également souvent utilisée car efficace et économe en mémoire, surtout pour les jeux de données de taille faible. De plus ces deux méthodes sont robustes dans le cas de données sans mise à l'échelle ce qui explique les bons résultats obtenus. En revanche, les méthodes *sag* et *saga* produisent généralement des meilleurs résultats sur des données formatées par centrage et réduction ce qui peut expliquer leur faible score.

Ainsi, en utilisant le solver textitnewton-cg pour une regression logistique appliquée aux données de test prétraitées, on obtient un score de test de 84.47%. La frontière de decision résultant de cet apprentissage est représentée dans la figure **??**

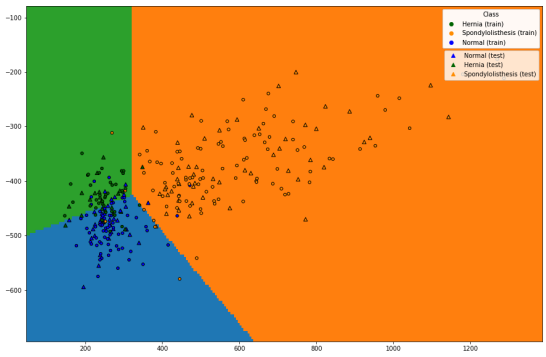

FIGURE 14 – Frontière de décision pour la régression logistique

6.4 Interprétation des coefficients

Il est par ailleurs intéressant d'analyser les coefficients obtenus suite à l'apprentissage d'un modèle afin de mettre en évidence la prépondérance de certaines variables dans la décision de classification. Or, étudier ces coefficients sur les données prétraitées n'a que peu d'intérêt puisque les axes considérés ne sont pas explicites aux vues de notre cas d'étude. Nous décidons ainsi de selectionner un autre modèle obtenu cette fois à partir des données auxquelles on a simplement retiré la variable *pelvic incidence*. En effet, nous avions pu observer un score d'apprentissage élevé, confirmé par un score de test satisfaisant (85.43%).

Grâce à la classe MNLogit du module *statsmodels*, on estime donc les coefficients de régression logistique multinomiale. On choisit comme groupe de référence la classe *Normal*. Cela signifie que les coefficients ainsi obtenus représentent les logit des variations des variables prédictives considérées par rapport aux valeurs de la classe de référence, en supposant les autres variables constantes. Ces coefficients sont obtenus par maximum de vraisemblance.

On applique le test de Wald pour tester la significativité des coefficients. Dans un soucis de simplification des résultats observés, nous avons décidé de ne représenter dans un tableau que les résultats obtenus pour la classe *Hernia* qui se rapproche d'avantage de la classe de réference que *Spondylolisthesis*.On regroupe ainsi les coefficients β*, le z-score et la p-value associé à cette statistique dans le tableau **??**.

La probabilité que la statistique de test soit aussi extrême que la valeur obtenue ici est définie par P>|z|. En

TABLE 6 – Estimation des coefficients de Régression Logistique pour la classe Hernia

Variables	Coef.	Z	P>|z|
const	21.2171	3.958	0.000
pelvic_tilt	0.0789	1.619	0.105
lumbar_lordosis_angle	-0.0291	-0.658	0.510
sacral_slope	-0.1915	-3.527	0.000
pelvic_radius	-0.1270	-3.554	0.000
degree_spondylolisthesis	0.0283	0.580	0.562

considérant un niveau de signification égal à 0.05, les p-values des statistiques de test z pour les descripteurs *sacral_slope* et *pelvic_radius* sont suffisamment faibles pour rejeter l'hypothèse nulle. On peut donc conclure que les coefficients relatifs à ces variables sont statistiquement différents de zéro et sont donc significatifs pour la classe *Hernia* comparée à la classe *Normal*. On peut donc s'appuyer sur ces variables afin de mieux distinguer les classes Hernia et Normal.

A titre indicatif, pour la classe *Spondylolisthesis*, le coefficient associé à la variable *degree_spondylolisthesis* est égal à 0.2803 ce qui renvoie z=5.032 et donc une probabilité P>|z| égale à 0.000. Le coefficient de cette variable est donc significatif. Cela conforte nos hypothèses sur l'importance de cette variable dans la classification des individus du groupe *Spondylolisthesis*.

7 Forêt aléatoire

La forêt aléatoire (Random Forest) est une autre méthode très populaire d'apprentissage supervisé. Afin d'approfondir notre analyse du jeu de données, nous décidons d'appliquer cette méthode à notre problème de classification.

7.1 Choix des prétraitements et des paramètres

Nous commençons encore une fois par déterminer le meilleur prétraitement à appliquer à nos données. Nous avons pour cela exploré plusieurs méthodes : enlever *pelvic_incidence*, transformer les données avec un StandardScaler, un MinMaxScaler mais aussi avec un NCA. Le NCA est classiquement utilisé avec un KNN. Cependant, aux vues de ses bons résultats pour séparer les classes (figure **??**), nous tentons tout de même de l'appliquer à la méthode de la forêt aléatoire, d'autant qu'il était satisfaisant dans le cadre d'une régression logistique.

La deuxième étape est la détermination des meilleurs paramètres. Nous avons identifié les paramètres suivants qui nous semblent intéressants :

- *criterion* : le critère utilisé pour évaluer la qualité d'une séparation. (*gini*, *entropy*)
- *max_features* : le nombre de caractéristiques à retenir de manière aléatoire pour la recherche de la meilleure séparation. (2, 3, 4, 5, None(toutes les caractéristiques))
- *n_estimators* : le nombre d'arbres utilisés dans la forêt. (10, 100, 200, 300)
- *bootstrap* : s'il faut effectuer l'entraînement sur toutes les données ou sur un échantillon aléatoire (True, False)

Afin de tester ces paramètres nous effectuons un *GridSearchCV*, sur toutes les méthodes de prétraitement à chaque fois.

7.2 Résultats

Le tableau **??** indique les meilleurs résultats obtenus pour chacune des méthodes détaillées précédemment.

TABLE 7 – Meilleur score d'apprentissage par méthode de prétraitement

Méthode	Données brutes	Sans pelvic incidence
Aucun prétraitement	85.40%	86.36 %
StandardScaler+NCA	86.88 %	87.85 %
StandardScaler	85.40%	86.36%
MinMaxScaler	85.40 %	86.36%
NCA	**88.81** %	85.40%
StandardScaler+NCA(2)	86.88 %	87.85%

On obtient donc les meilleurs résultats avec une transformation NCA sans réduction de dimensions sur les données originales. Ce résultat est obtenu pour les paramètres suivants : *bootstrap* : False, *criterion* : gini, *max_features* : 4, *n_estimators* : 100.

On remarque une légère amélioration de la *validation accuracy* quand on enlève *pelvic incidence*. Néanmoins ce n'est pas le cas pour la NCA. En effet comme on a dispose de caractéristiques différentes retenues à chaque arbre afin de minimiser la variance, il est possible que *pelvic incidence* soit naturellement non considérée lors de la prise de décision. C'est d'autant plus possible que *max_features* dans le cas du meilleur prétraitement est différent de None, c'est-à-dire qu'on ne prend pas en compte toutes les caractéristiques.

Une normalisation seule n'a également pas beaucoup

d'impact sans NCA. Cela n'est pas surprenant sachant que le Random Forest ne calcule pas la distance lors de la prise de décision, donc une normalisation seule n'est pas intéressante.

La figure **??** montre l'évolution de l'accuracy en fonction des différents paramètres avec le prétraitement NCA appliqué sur les données originales.

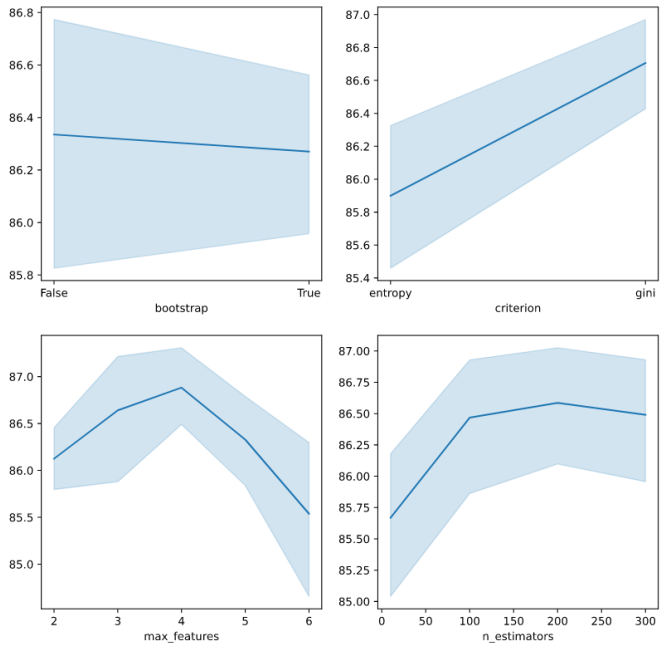

FIGURE 15 – Résultats en fonction des paramètres

On sait que le bootstrap est censé améliorer les résultats en diminuant la variance. Or le meilleur résultat est obtenu sans bootstrap. En réalité, on remarque que l'accuracy varie peu si on utilise ou pas le *bootstrap*.

On distingue également une légère amélioration des résultats avec un *n_estimators* plus élevé, ce qui est logique puisque l'algorithme prend une décision à partir d'un nombre d'arbres plus élevé. Par contre cette amélioration s'arrête après un certain seuil.

Concernant le nombre de caractéristiques utilisées, on peut voir qu'on obtient une meilleur accuracy avec un *max_features* autour de 4. Cela indique qu'il y a des caractéristiques qui permettent de mieux séparer le jeu de données que d'autres.

Pour finir, on remarque validation accuracy légèrement plus élevée pour *gini*.

Il est également intéressant d'analyser (figure **??**) l'importance de chaque caractéristique dans la prise de décision. En utilisant l'attribut *feature_importances_* on remarque que *degree_spondylolisthesis* permet de classer presque la moitié des points. Cela confirme l'importance de cette variable dans la classification des individus. Par ailleurs *pelvic_radius* et *pelvic_tilt* permettent d'en séparer respectivement 20% et 15%. Cela confirme ce qu'on pouvait voir dans la section exploration des données où la variable *pelvic_radius* semblait assez indépendante des autres, de sorte qu'elle expliquait de manière importante l'axe 2 dans l'ACP à deux dimensions.

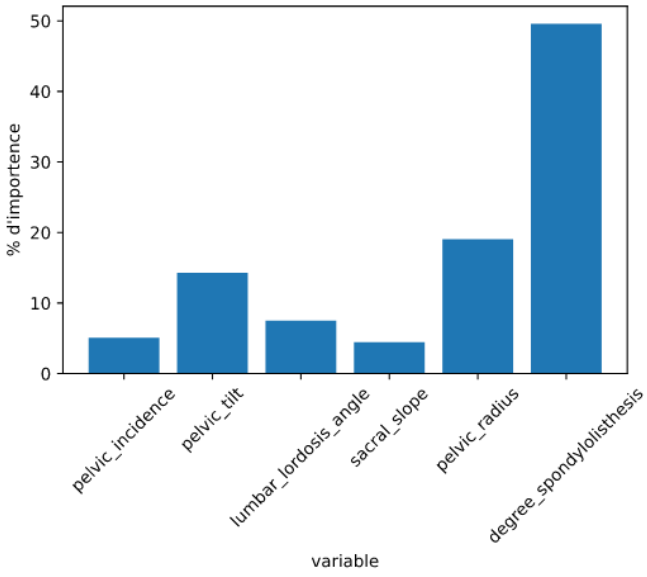

FIGURE 16 – Importance des variables

En appliquant le meilleur modèle à l'ensemble de test on obtient un score d'accuracy de 83.50 %

7.3 Extra Tree

Il existe un algorithme qui pousse la randomisation encore plus loin : le *Extremely randomized Trees*. Il suit le même principe que le Random Forest. La seule différence est dans la manière de prendre une décision au niveau d'une feuille. Au lieu de choisir le meilleur seuil pour chaque caractéristique, c'est un seuil aléatoire qui est choisi. Le meilleur seuil entre toutes les caractéristiques est ensuite choisi par le critère de séparation. De cette manière la variance diminue encore plus, avec néanmoins un risque d'augmenter le biais.

Nous avons donc suivi pour cet algorithme la même procédure de GridSearchCV que pour le Random Forest, avec les mêmes prétraitements et paramètres. On remarque alors une amélioration des résultats de la validation pour quelques méthodes de prétraitement, notamment pour le MinMaxScaler + NCA à deux compo-

sanates et sans *pelvic_ incidence*. On obtient une accuracy de 90.27 % avec les paramètres suivants : *bootstrap* : True, *criterion* : gini, *max_ features* : 2, *n_ estimators* : 200.

Néanmoins la test accuracy baisse à 80.58 %. En en déduit que cette méthode s'adapte mieux aux données d'apprentissage mais parvient moins bien à prédire les classes des données de test.

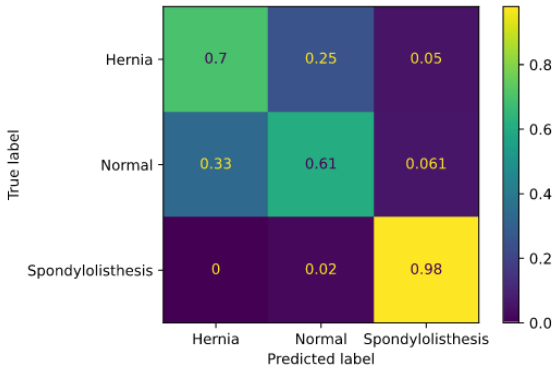

FIGURE 17 – Matrice de confusion pour le Extra tree

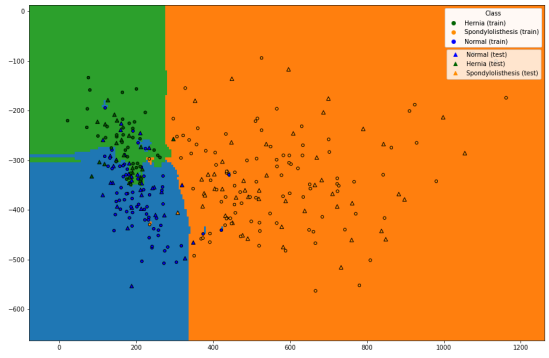

FIGURE 18 – Frontière de décision pour le Extra tree

TABLE 8 – Meilleur score de test par modèle

Modèle	Prétraitement	Test accu
QDA	Aucun	79.61 %
KNN	StandardScaler + NCA(2)	81.55 %
Régression logistique	NCA(2) sans *pelvic_ incidence*	84.47 %
Random forest	NCA	83.50 %
Extra tree	MinMaxScaler +NCA(2)	80.58 %

8 Conclusion