Rapport du projet de SY09

Data mining

Margaux Peschiera, Tom Bourg, Yunfei ZHAO

14 juin 2020

1 Introduction

L'objectif de notre projet est d'utiliser un certain nombre d'algorithmes d'apprentissage sur un jeu de donnée afin d'être capable de classer des individus selon des classes et prédire la classe d'un individu nouveau. Ce projet a été réalisé au cours de l'UV SY09 en P20 à l'UTC. Notre rapport présente les résultats de nos analyses ainsi que nos conclusions. Notre jeu de données "Biomechanical features of orthopedic patients" est issu du site Kaggle.com. Les données sont représentées sous la forme de tableaux individus/ variables.

Au cours de notre projet, nous avons tenté de répondre au problème suivant : Comment et avec quelle précision l'analyse de données peut permettre de détecter qu'une personne est atteinte d'une pathologie ou non, en fonction de ses paramètres pelviens ? Pour répondre à cette question, nous commencerons par effectuer une analyse descriptive des données. Nous nous pencherons ensuite sur les méthodes d'analyse non supervisée. Pour finir, nous utiliserons différentes méthodes d'apprentissage supervisé afin de pouvoir prédire la classe de nouveaux individus.

2 Traitement des données

2.1 Présentation des données

Notre jeu de donnée est "Biomechanical features of orthopedic patients". Pour mener à bien ce projet, nous avons d'abord essayé de comprendre au mieux les données. Tout d'abord une définition formelle de l'orthopédie est : *l'orthopédie est la spécialité chirurgicale qui a pour objet la prévention et la correction des affections de l'appareil locomoteur. Le traitement chirurgical porte sur les membres supérieurs (épaule, coude et main) ainsi que les membres inférieurs (hanche, genou et pied).* On peut donc alors comprendre que les données représentent différentes caractéristiques prises sur des patients qui sont atteints ou non de maladie.
Notre jeu de donnée est composé de deux fichiers CSV qui se différencient par la colonne class. En effet, pour l'un des fichiers, les valeurs prises par "class" est "Normal" ou "Abnormal", tandis que pour le second fichier, les valeurs prises par class sont "Hernia", "Spondylolisthesis" et "Normal". On remarque également que toutes lignes correspondant à "Abnormal" dans le premier fichier sont des lignes correspondant à "Hernia" ou "Spondylolisthesis" dans le second fichier. Nous avons donc conclu que "Hernia" et "Spondylolisthesis" pourraient être des types de maladie de l'appareil locomoteur. Notre jeu de donnée est composé de 7 colonnes et 310 lignes. Il n'y a aucune valeur manquante. Pour chacune des colonnes, nous avons fait quelques recherches afin d'avoir une meilleure idée de la signification de chaque colonne. Il s'agit en réalité de paramètre pelvien.

```
RangeIndex: 310 entries, 0 to 309
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   pelvic_incidence          310 non-null    float64
 1   pelvic_tilt               310 non-null    float64
 2   lumbar_lordosis_angle     310 non-null    float64
 3   sacral_slope              310 non-null    float64
 4   pelvic_radius             310 non-null    float64
 5   degree_spondylolisthesis  310 non-null    float64
 6   class                     310 non-null    object
dtypes: float64(6), object(1)
```

Quelques recherches supplémentaires nous ont amenés à trouver les définitions suivantes :
Le spondylolisthésis lombaire correspond au glissement d'une vertèbre lombaire par rapport à la vertèbre située juste en dessous et entraînant avec elle tout le reste de la colonne vertébrale.
La hernie discale est un déplacement d'un disque intervertébral dans la colonne vertébrale. Ce disque sert à amortir les chocs.
Finalement, à ce stade nous avons deux fichiers CSV contenant 7 caractéristiques de différents patients étant atteint ou non de maladie de l'appareil locomoteur. De plus, le premier fichier CSV illustre uniquement des patients atteints de maladie ou non alors que le deuxième est plus précis et décrit de quel type de maladie le patient est atteint lorsqu'il est malade. Regardons à présent les autres colonnes des fichiers CSV (qui sont identiques dans chacun des fichiers).

- pelvic incidence(**PI**) : Incidence pelvienne. Il s'agit de l'angle qui représente grossièrement la fondation sur laquelle va reposer la colonne.
- pelvic tilt numeric(**PT**) : Inclinaison pelvienne.
- sacral slope(**SS**) : La pente sacrée est définie par l'orientation du plateau sacré (voir figure 1). 1

— pelvic radius(**PR**) : Rayon pelvien. Il s'agit de la distance entre le centre de l'axe de la hanche et le point de référence S1 (le point de mesure du SS).
— lumbar lordosis angle(**LLA**) : Angle de la lordose lombaire qui est une section de la colonne vertébrale 2
— degree spondylolisthesis(**DS**) : C'est un indicateur pour mesurer le niveau de spondylolisthésis.

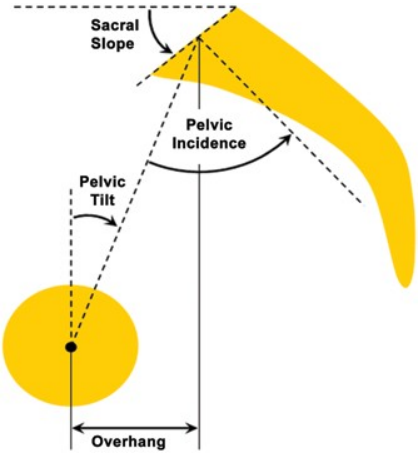

FIGURE 1 – Représentation des angles de pelvien. [4]

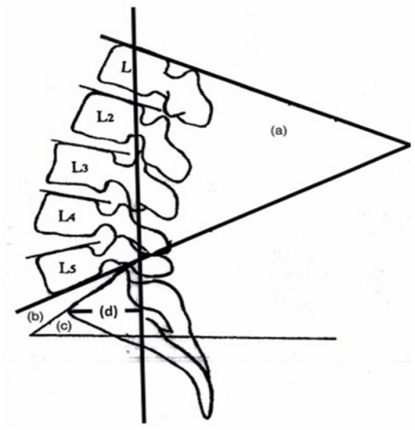

FIGURE 2 – Représentation des angle LLA. [1]

2.2 Analyse exploratoire

Une analyse exploratoire des données permet d'en apprendre davantage sur les données. L'objectif est de comparer les différentes variables entre elles, observer des corrélations s'il en existe et identifier des paramètres sur lesquels il serait intéressant de creuser pour nos futures analyses plus poussées.
Dans un premier temps, nous allons nous pencher sur le fichier *column_2C_weka.csv*. Une manière naturelle de procéder serait d'explorer les données classées en deux classes : "Normal" et "Abnormal", pour ensuite regarder ces mêmes données classées en trois classes : "Normal", "Hernia" et "Spondylolisthesis". Voici les boxplots des variables du fichier à deux classes :

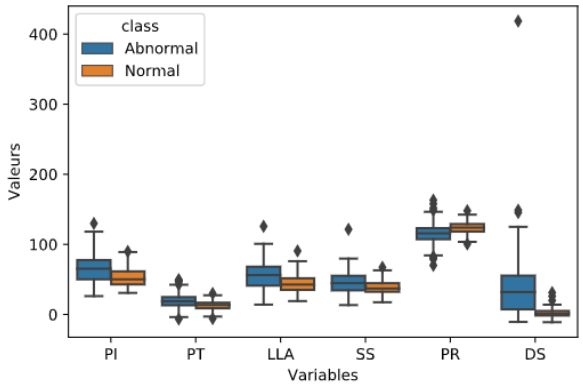

FIGURE 3 – Représentation boxplot pour les deux classes.

Ce diagramme nous permet d'observer que la classe "Abnormal" se distingue notamment avec la variable *dregree spondylolisthesis* qui a une médiane significativement supérieure à celle de la classe "Normal". On remarque également que la classe "Abnormal" a tendance à avoir des valeurs supérieures à la classe "Normal" sur les variables *PI*, *PT*, *LLA*, *SS* et inférieur sur la variable *PR*. retirant les valeurs aberrantes qui sont présentes.

Nous allons observer les boxplots des variables du fichier à trois classes.

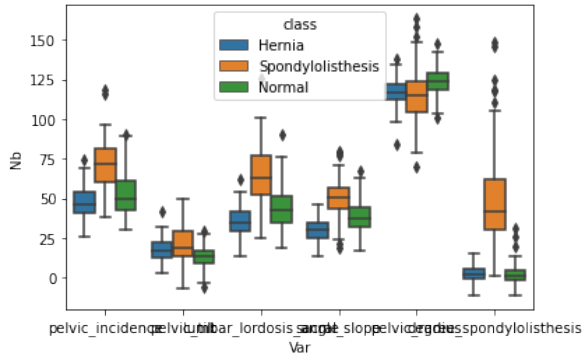

FIGURE 4 – Représentation boxplot pour les trois classes en retirant les valeurs aberrantes.

Cette analyse permet de mettre en évidence que les individus de la classe "Hernia" et "Spondylolisthesis" sont finalement significativement différents. En effet, la classe "Spondylolisthesis" a un DS avec une médiane élevée comparée aux classes "Normal" et "Hernia" qui ont un DS très faible. Sur les variables PI, LLA et SS, la classe "Spondylolisthesis" est également différente. Le fait de regrouper les classes "Hernia" et Spondylolisthesis" en une seule peut être trompeur.

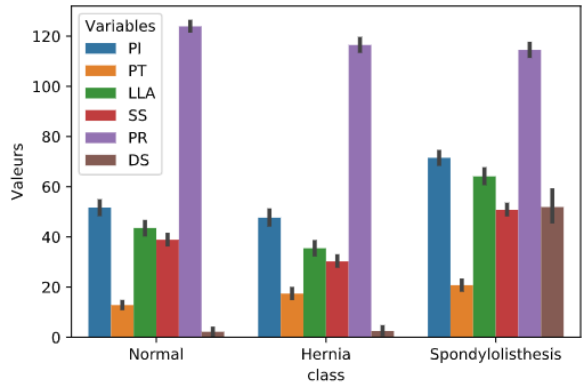

FIGURE 5 – Représentation barplot pour les trois classes.

On retrouve les mêmes écarts sur le barplot. Lorsqu'on regarde les classes "Normal" et "Hernia", on observe une différence entre les écarts des variables SS et PT. On sait également d'après nos recherches, que :

$$PI = PT + SS \tag{1}$$

Pour finir, il est intéressant de regarder la corrélation des variables.

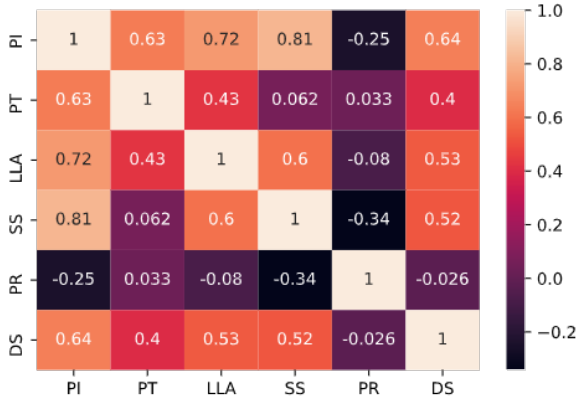

FIGURE 6 – Matrice de covariance.

On remarque que le PI est globalement corrélé avec les autres à l'exception du PR. En revanche le PR n'est corrélé avec aucune autre variable.

3 Modèle et Méthodologie

L'apprentissage se divise en 2 grandes sections. L'apprentissage non supervisé et l'apprentissage supervisé.

Dans le cas de l'apprentissage non supervisé, l'apprentissage du modèle se fait de façon totalement autonome. Les données communiquées au modèle sont donc non étiquetées. Nos données étant étiquetées, nous allons pouvoir utiliser les algorithmes d'apprentissage non supervisé et comparer les classes estimées avec les classes réelles. Nous pourrons ainsi mesurer l'efficacité de chaque méthode et observer quelles sont celles qui sont le plus performantes.

Pour l'apprentissage supervisé, un utilisateur "aide" l'algorithme en lui fournissant des exemples significatifs étiquetés. Le modèle apprend alors de chaque exemple en ajustant. La marge d'erreur se réduit donc grâce aux exemples. Le but étant d'être capable de généraliser son apprentissage à de nouveaux cas.

Nous allons utiliser premièrement les modèles d'apprentissage non supervisés pour illustrer notre jeu de donnée. Ensuite, nous allons appliquer les modèles d'apprentissage supervisé pour réaliser la mission de classification. Pour

évaluer les différents modèles, nous allons utiliser la *Nested Cross Validation*. Nous utilisons cette méthode, car le jeu de donnée à notre disposition est petit. Il y a au total seulement 310 individus. Il existe deux types de modèles que nous allons appliquer par la suite. Les modèles qui possèdent des hyper paramètres à déterminer, il faut alors faire de la validation croisée (Cross Validation) pour choisir les paramètres optimaux, puis évaluer le modèle avec les hyper paramètres trouvés. Il existe également d'autres types de modèles qui ne possèdent pas d'hyper paramètre à déterminer tel que le Naive Bayes. Nous pouvons alors appliquer un Nfold validation directement.

3.1 Nested Cross validation

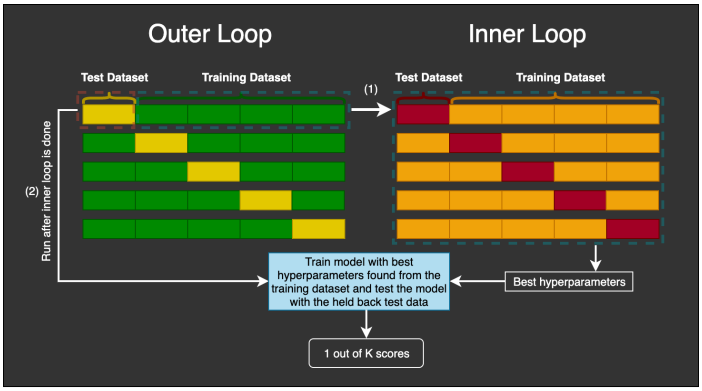

FIGURE 7 – Schéma de Nested Cross Validation. [2]

Comme la figure 7 illustre, pour les modèles possédant des hyper paramètres à choisir, nous séparons notre ensemble de données en N sous-ensemble, en mélangeant les données de façon à ce que chaque sous-ensemble possède les trois types d'individus. Nous avons d'abord une boucle extérieure, puis, pour chaque itération de boucle d'extérieure, nous séparons les données en un ensemble de tests et un ensemble d'entraînement. Dans l'ensemble d'entraînement, nous lançons une boucle intérieure. La boucle intérieure sert à réaliser un Kfold cross validation pour trouver les meilleurs hyper paramètres. Finalement, une fois que les hyper paramètres sont déterminés, nous appliquons les hyper paramètres trouvés sur l'ensemble de tests de boucle extérieure. Nous obtiendrons une estimation de ce modèle sans biais sur les données.

3.2 Évaluation des modèles

Après le nettoyage des données, nous avons compté le nombre d'individus dans chaque classe.

Hernia	Spondylolisthesis	Normal
60	149	100

Nous avons constaté que les tailles des classes ne sont pas équilibrées. Notre jeu de données représente des maladies, il est donc important d'éviter au maximum des faux négatifs. En effet, un faux négatif reviendrait à classifier un patient malade comme était non malade. Ceci pourrait avoir une conséquence importante pour les patients concernés (non-traitement de la maladie). Nous avons donc décidé de considérer comme métrique de comparaison de modèle le F1-score, afin de trouver un équilibre entre précision et rappel.

La précision représente l'efficacité du classificateur lors de la détection d'un patient malade. Par exemple, si la précision d'un classificateur est de 80%, et qu'il classifie un patient comme étant malade, alors il y a 80% de chance que le patient en effet réellement malade.

$$Precision = \frac{VraiPositif}{VraiPositif + FauxPositif}$$

Le rappel représente la capacité du classificateur à trouver tous les patients malades. Par exemple, si un classificateur possède 80% de rappel, sur un échantillon de 100 patients malades, il va en classifier 80 comme étant malade.

$$Rappel = \frac{VraiPositif}{VraiPositif + FauxNegatif}$$

Le $F1_score$ combine précision et rappel. Il est bon en lorsqu'il tend vers 1 et mauvais lorsqu'il tend vers 0.

$$F1_score = 2 * \frac{Precision * Rappel}{Precision + Rappel}$$

4 Apprentissage Non-Supervisé

Cette partie est consacrée à l'apprentissage non supervisé. Nous présentons les résultats obtenus avec différentes méthodes d'apprentissage. Nous avons réalisé une analyse en composantes principales (ACP), une classification as-

cendante hiérarchique (CAH) et appliqué l'algorithme des KMeans.

4.1 Analyse en composantes principales

Nous avons effectué une ACP pour essayer de voir si une séparation entre les classes était faisable.

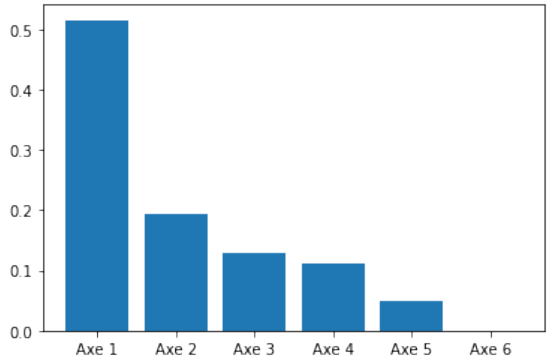

FIGURE 8 – taux de variance expliqué

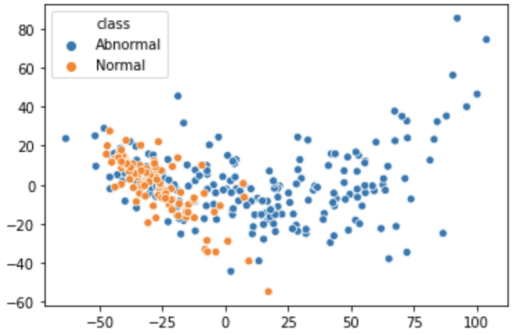

FIGURE 9 – ACP sur le fichier à 2 classes.

Tout d'abord nous avons effectué une ACP sans traitement de données, en utilisant les données brutes (sans les valeurs aberrantes). Nous avons visualisé les deux axes les plus importants de notre ACP sur les données comportant deux classes. Le nuage de point obtenu ne possède pas de démarcation visible. Lorsqu'on affiche les classes réelles, on peut remarquer que la classe Normal se situe à gauche du plan, mais reste confondu avec la classe Abnormal. En affichant les 3 classes, on peut observer qu'a droite du plan, les individus appartiennent tous à la classe Spondylolisthesis. Cependant, à gauche, les classes Normal et Hernia sont confondus ce qui est problématique puisque la classe Hernia correspond a des personnes malades et la classe Normal à des personnes saines.

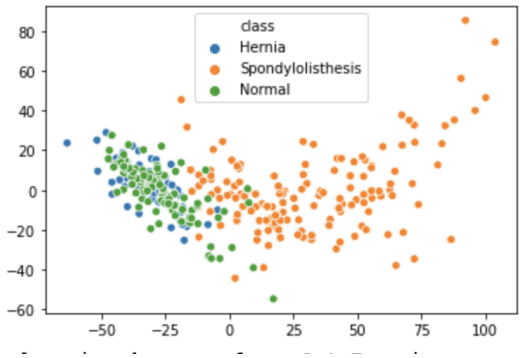

FIGURE 10 – ACP sur le fichier à 3 classes.

Nous avons ensuite essayé de trouver un moyen de faire apparaître 2 clusters différents. Pour cela, nous avons tout d'abord effectué de l'ACP en gardant uniquement les deux classes difficiles à différencier.

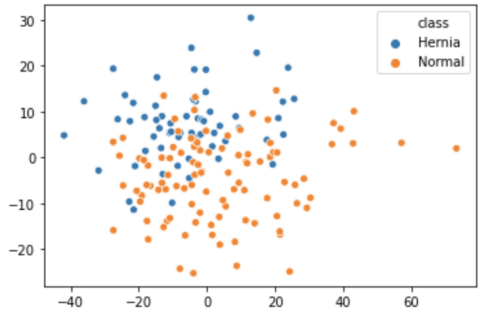

FIGURE 11 – ACP pour différencier Hernia et Normal.

D'après cette figure, nous pouvons voir apparaître deux clusters. Le cluster des patients non malades en orange est un petit peu plus bas que le cluster des patients atteints de Hernia. Cependant, on peut voir que de nombreux points sont "communs" aux deux clusters, ce qui pourrait expliquer que lors de l'ACP avec les 3 classes, on ne parvient pas à distinguer les 3 classes. Il est logique que la classe "Spondylolisthesis" se démarque d'après nos analyses exploratoires, en effet, nous avions remarqué à plusieurs reprises que le DS était significativement différent pour les individus appartenant à cette classe. Ensuite, nous avons essayé de normaliser les données afin de voir si cela pouvait améliorer la visualisation de clusters

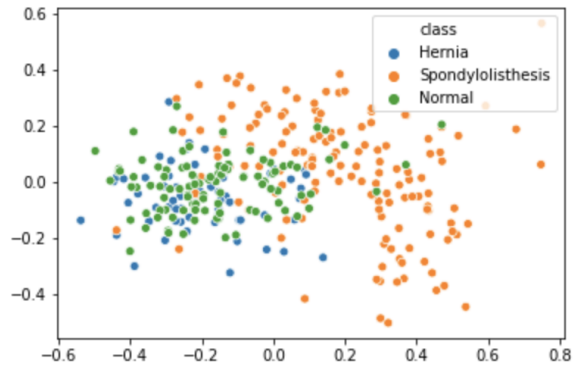

FIGURE 12 – ACP pour différencier Hernia et Normal.

Une fois encore, nous remarquons que les clusters correspondant à "Non malade" ainsi que "Hernia" sont difficilement discernables.

Nous avons également essayé de travailler avec les données en divisant les variables par le PI et en le retirant au vu de la relation entre PI, SS et PT mais également, car il s'agit de la variable la plus corrélée. Cependant, cela n'a pas considérablement changé les résultats de l'ACP.

L'affichage en trois dimensions des données selon les trois plus importants axes de l'ACP donne le résultat suivant.

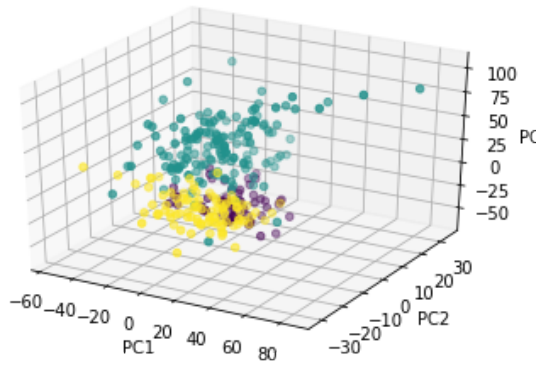

FIGURE 13 – Projection des points sur trois axes principaux de PCA

4.2 Classification ascendante hiérarchique

Nous avons également réalisé une classification ascendante hiérarchique. Nous avons utilisé le critère de Ward comme critère d'agglomération et la distance euclidienne pour le calculer. Nous avons également enlevé les valeurs aberrantes pour avoir de meilleurs résultats. Nous obtenons le dendrogramme suivant.

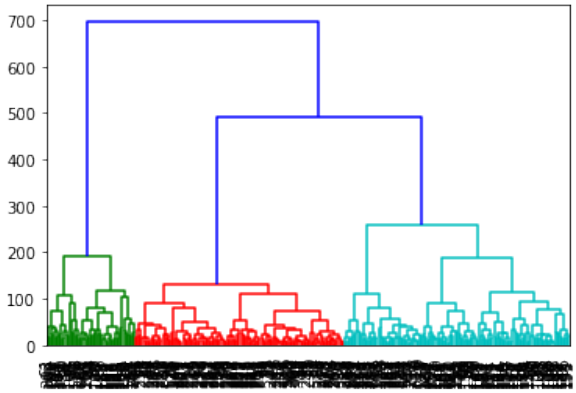

FIGURE 14 – Dendrogramme de la CAH avec la distance euclidienne

Trois clusters se distinguent dans le dendrogramme. Cependant lorsqu'on assigne des étiquettes pour chaque cluster et qu'on les compare avec les classes réelles, on remarque que la CAH n'est pas optimale. En effet, les formes

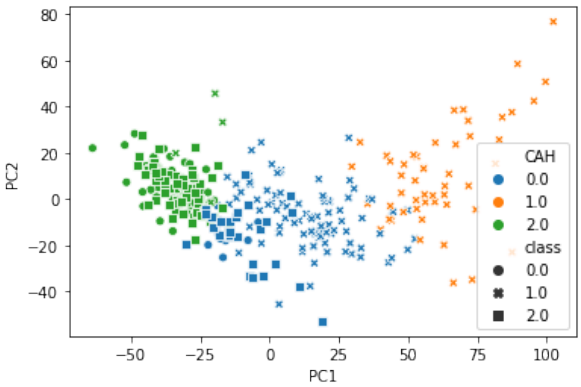

FIGURE 15 – Résultat de la CAH avec la distance euclidienne

de la figure 15 indiquent les classes réelles et les couleurs les résultats de la CAH. Nous obtenons la matrice de confusion suivante :

		Classe estimé		
		Hernia	Spondy	Normal
Classe réelle	Hernia	10	0	50
	Spondy	94	52	2
	Normal	29	0	71

F1-Score : 0,4768

La matrice de confusion et le F1-Score témoignent que l'estimation des classes avec la CAH n'est pas convaincante.

4.3 Kmeans

Nous avons appliqué l'algorithme des KMeans sur nos données en utilisant une initialisation des centres de manière aléatoire. Les résultats restent peu pertinents lorsqu'on compare la classification obtenue par les KMeans avec les classes réelles.

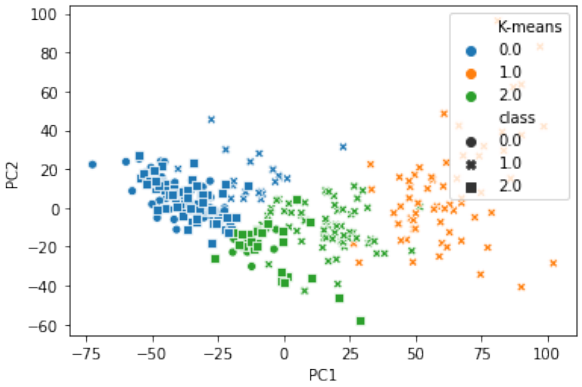

FIGURE 16 – Kmeans sur le fichier à 3 classes.

Nous pouvons observer que les résultats sont assez proches avec la classification ascendante hiérarchique. Nous obtenons la matrice de confusion suivante :

		Classe estimé		
		Hernia	Spondy	Normal
Classe réelle	Hernia	51	0	9
	Spondy	19	63	67
	Normal	73	0	27

F1-Score : 0,4702

La matrice de confusion et le F1-Score témoignent que l'estimation des classes avec l'algorithme des KMeans n'est pas convaincante. Le F1-Score est légèrement inférieur à celui de la CAH. De plus, le nombre de faux négatifs est bien plus important ce qui est problématique pour notre approche.

Nous remarquons que notre jeu de données ne se prête pas très bien à l'apprentissage non supervisé. En effets, les classes Hernia et Normal demeurent difficilement séparables avec les méthodes utilisées. Nous allons utiliser les algorithmes d'apprentissage supervisé dans l'espoir d'obtenir de meilleurs résultats.

5 Apprentissage Supervisé

L'apprentissage supervisé consiste à apprendre une fonction de prédiction à partir d'exemples étiquetés. Notre approche consiste à essayer un grand nombre de méthodes pour évaluer celles qui ont les meilleures performances. Nous avons utilisé la Nested Cross Validation comme nous l'avions évoqué dans la partie "Modèle et méthodologie". Pour faciliter la manipulation des données. Nous avons remplacé le nom des classes par 0 (Hernia), 1 (normal), 2 (Spondylolisthesis).

5.1 K Plus Proches Voisins

L'algorithme des K Plus Proches Voisins prédit la classe d'un individu en réalisant une comparaison de ses voisins plus proches avec une distance prédéfinie. Nous allons estimer l'hyper paramètre N qui est le nombre de voisins que nous allons évaluer autour de chaque point.

KNN sur les dimensions initiales :
Nous avons dans un premier temps, utilisé ce modèle sur les dimensions initiales et les dimensions de l'ACP en gardant les 5 dimensions les plus importantes. La figure 17 illustre les F1-Scores et les variances obtenues avec l'algorithme des KNN sur les 5 dimensions initiales.

Paramètre choisi :
— Nombre de voisins : 10
Le F1-Score moyen obtenu est de 0.8117.

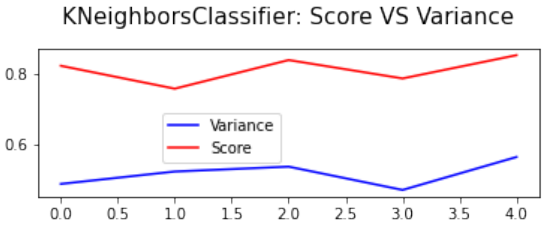

FIGURE 17 – Affichage des F1-Scores et Variances du KNN (dimensions initiales) pour 5 itérations de la boucle extérieur

KNN sur les dimensions de l'ACP :
Nous savons que les variables PI, PT et SS sont fortement corrélées. Par conséquent, nous avons décidé de diminuer le nombre de dimensions avec l'ACP pour obtenir des dimensions non dépendantes linéairement. La figure 19 montre les F1-Scores et les variances obtenues avec les nouvelles dimensions.

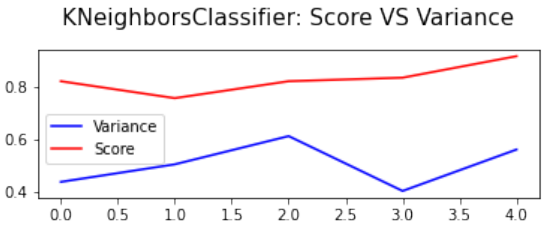

FIGURE 18 – Affichage des F1-Scores et Variances du KNN (dimensions ACP) pour 5 itérations de la boucle extérieur

Paramètre choisi :
— Nombre de voisins : 20
Le F1-Score moyen obtenu est de 0.8314. On constate alors une amélioration du F1-Score, cependant la variance a augmenté. On risque donc d'avoir un surapprentissage.

5.2 Naïve Bayes

La méthode d'apprentissage Naïve Bayes applique le théorème de Bayes sur plusieurs variables en supposant leur indépendance. La condition d'indépendances des variables n'est pas respectée dans nos données et dans la majorité des cas d'application de cet algorithme. Malgré cela, le Naïve Bayes est beaucoup utilisé, car il donne quand même de bons résultats. Nous avons pris le noyau gaussien.

Nous obtenons un F1-Score moyen de 0.8161 ce qui reste correct.

5.3 Analyses discriminantes linéaire et quadratique

Nous avons utilisé les analyses discriminantes linéaire et quadratique. La LDA (Analyse Discriminante Linéaire) et la QDA (Analyse Discriminante Quadratique) supposent que les données suivent une loi gaussienne. La LDA suppose que la matrice de variance est commune à toutes les classes, mais que la matrice d'espérance est différente. La QDA suppose que les matrices de variance et d'espérance sont différentes pour toutes les classes. Ensuite, nous avons appliqué la règle de Bayes pour classifier un individu.

On obtient un F1-Score moyen de 0.7903 avec la LDA et 0,8258 avec la QDA

Étant donnée que nous n'avons que 6 variables, la QDA a de meilleures performances par rapport à la LDA et à la méthode Bayes Naïve. De plus, nous pensons que les variables sont pas très séparables de manière linéaire, car la LDA a eu un score inférieur à la méthode Naïve Bayes et la QDA.

5.4 Régression Logistique

Dans la Régression Logistique, nous exprimons la probabilité par une fonction logistique de combinaison linéaire de $\beta^T X$, puis nous approchons le β vers β^* qui maximise la vraisemblance. L'hyper paramètre que nous allons tester est la fonction de pénalité. Nous avons testé la régression logistique sans pénalité et avec pénalité L2 (Ridge). Pour avoir une convergence plus rapide, nous standardisons d'abord les données.

Paramètre choisi :
— Pénalité : Ridge
F1-Score moyen obtenu : 0.8546

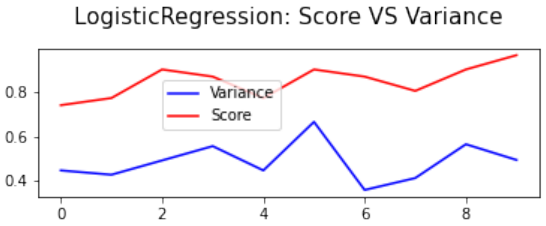

FIGURE 19 – Affichage des F1-Scores et Variances de la régression logistique pour 10 itérations de la boucle extérieur

5.5 Arbre de Décision

Arbre de décision nous aide de tracer le processus de décision. *Paramètre choisi :*

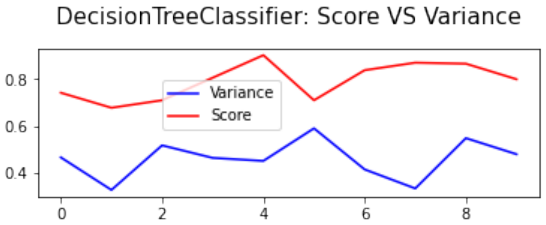

FIGURE 20 – Affichage des F1-Scores et Variances de l'arbre de décision pour 10 itérations de la boucle extérieur

— ccp (Pénalité de complexité) : 0.005

F1-Score moyen obtenu : 0.7903 Pour avoir une visualisation de l' arbre de décision, nous avons pris cpp = 0.008, c'est-à-dire, nous augmentons le nombre de nœuds élagués, pour avoir une vue plus claire. Mais pour cette méthode nous pouvons constater que pour cette méthode, la variance sur l'ensemble de tests est un peu plus haute par rapport aux autres méthodes bien que nous avons élagué l'arbre. Nous allons donc nous diriger vers les méthodes d'ensemble pour diminuer ce problème.

Le figure 21 nous avons montré que le DS a pris un poids très important pour séparer les Spondylolisthesis et les autres. Presque tous les individus ayant un DS supérieur à 16 appartiennent à la classe Spondylolisthesis. Puis dans les noeuds suivants, nous considérons principalement le PR et la SS.

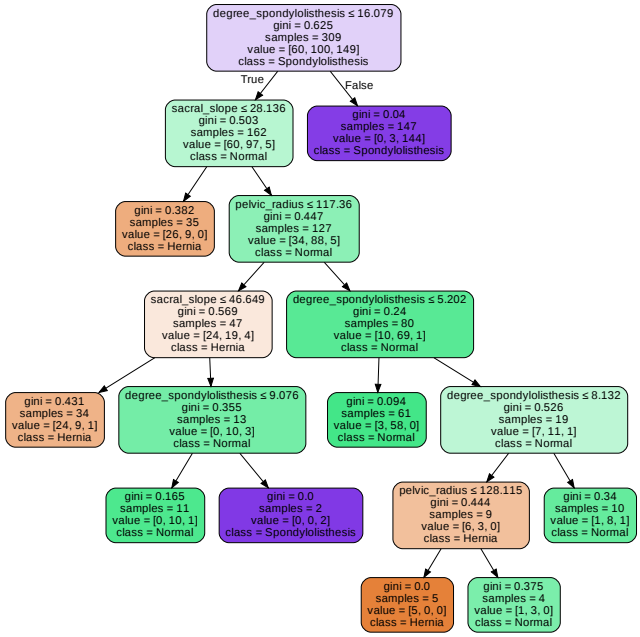

FIGURE 21 – arbre de décision avec $ccp - alpha$: 0.008

5.6 Forêt Aléatoire

À partir de nos données, nous avons effectué N tirages avec remise pour construire M nouveaux ensembles. Pour chaque ensemble, nous construisons un arbre complet afin d'en déduire ensemble le résultat.

NCV résultat :

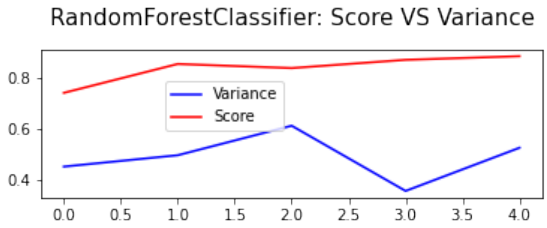

FIGURE 22 – Affichage des F1-Scores et Variances de la forêt aléatoire pour 5 itérations de la boucle extérieur

Paramètre choisi :

— Criterion : gini

— Minimum individuel pour générer un nouveau noeud : 8

— Nombre d'arbres : 200

Finalement, le F1-Score moyen obtenu est 0.8286. Nous pouvons constater que la variance a diminué par rapport l'arbre de décision. En utilisant le modèle de Forêt Aléatoire, nous pouvons étudier l'importance des variables, c'est-à-dire la fréquence des variables utilisée pour construire les arbres.

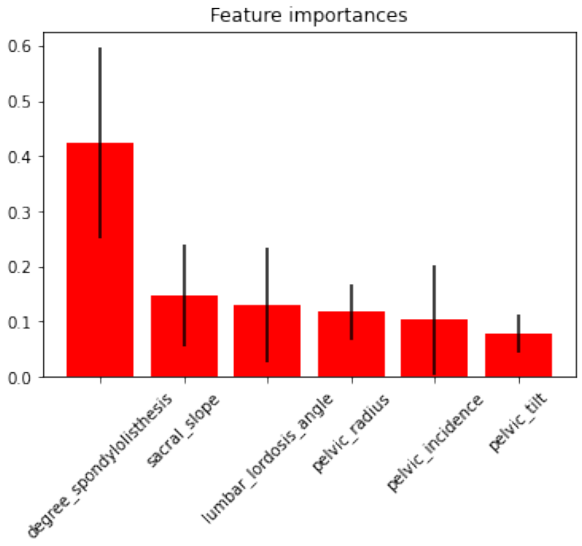

FIGURE 23 – L'importance des axes

L'affichage des données selon les axes sacral slope, degree spondylolisthesis et lumbar lordosis.

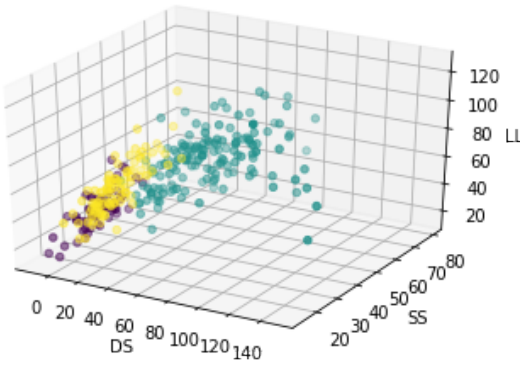

FIGURE 24 – Projection des points sur trois axes importants

5.7 Machine à vecteurs de support (SVM)

Enfin, nous avons appliqué le support machine vector (SVM). L'objectif de cette méthode est de trouver la frontière des différentes classes en maximisant la distance entre les points les plus proches de ce plan et les points qui sont "vecteur de support". Dans le cas où les données ne sont pas linéairement séparables, nous introduisons une constante C pour permettre de pénaliser les points mal classés. La constante C est un hyper paramètre. Pour la marge de frontière, plus C est grand, plus la marge est petite et plus on se focalise sur les points proches de la frontière.

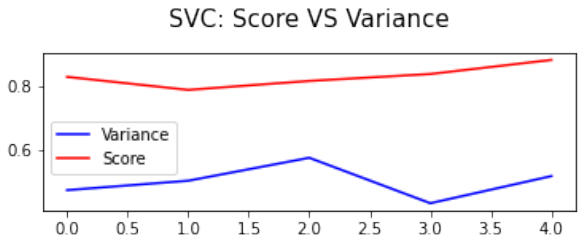

FIGURE 25 – Affichage des F1-Scores et Variances de la SVM pour 5 itérations de la boucle extérieur

Paramètrs choisi :

— C : 2.7826

F1-Score moyen : 0.8452

5.8 NCA : Analyse des Composantes du Voisin

NCA(Neighborhood Components Analysis) est une méthode d'apprentissage supervisée pour réduire la dimension. Nous avons référencé l'article de la NCA [3] pour l'utiliser dans notre étude. La NCA est une méthode d'apprentissage supervisé qui utilise la distance Mahalanobis.

Pour la distance Mahalanobis, nous avons :

$$d(x,y)^\top = (x-y)^\top Q(x-y) = (Ax-Ay)^\top (Ax-Ay) \quad (2)$$

On estime ensuite cette distance par une probabilité.

Pour un point i, la probabilité que le point j soit son voisin est :

$$p_{ij} = \frac{exp(-\|Ax_i - Ay_j\|^2)}{\sum_{k \neq i} exp(-\|Ax_i - Ay_j\|^2)}, p_{ii} = 0 \quad (3)$$

La probabilité que le point i soit correctement classifié est : pour ensemble Ci = {j|ci = cj}

$$p_i = \sum_{j \in C_i} p_{ij} \quad (4)$$

Finalement, l'algorithme cherche à trouver le A qui maximise la fonction de vraisemblance 5 par les méthodes de gradients :

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i \quad (5)$$

5.8.1 Visualisation par NCA

Ax est une réduction de A, et les nouvelles dimensions sont la transformation linéaire des anciennes dimensions. Nous avons appliqué de nouveau le classifier de régression logistique et le classificateur forêt aléatoire sur ce nouveau A. Nous avons constaté une amélioration du F1-Score. Le score est supérieur à 85% bien qu'on garde que deux dimensions. Donc la prédiction sur les deux nouvelles dimensions est généralement fiable et cela peut nous aider à visualiser notre jeu de données en trois clusters séparés. Nos prédictions seront également meilleures.

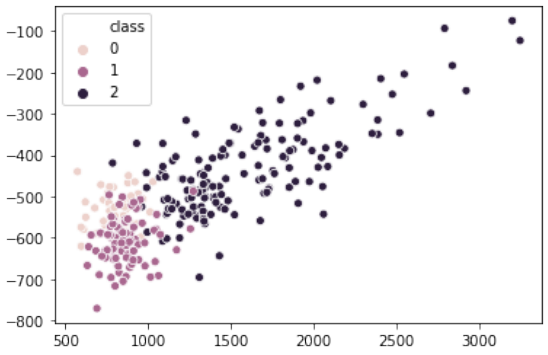

FIGURE 26 – Nuage de point après la transformation NCA

La figure 26 est la visualisation de notre jeu de données sur les nouvelles dimensions. La matrice de transformation :

$$\begin{pmatrix} 2.187 & 2.492 & 0.979 & -0.306 & 5.211 & 14.862 \\ -0.838 & 2.7465 & 0.619 & -3.585 & -4.007 & 3.360 \end{pmatrix}$$

Cela montre que le **degree spondylolisthesis** a pris un poids plus important pour faire la classification.

5.8.2 NCA et Foret Aléatoire

Nous avons donc testé la forêt aléatoire dans le plan principal obtenu par la NCA.

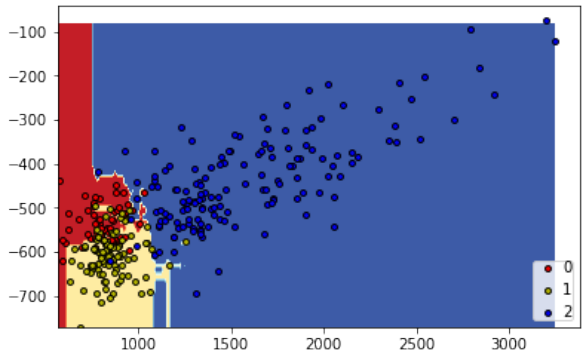

FIGURE 27 – Visualisation des frontières de la forêt aléatoire

On obtient un F1-Score moyen de 0.8608.

5.8.3 NCA et Régression Logistique

Nous avons également testé la régression logistique dans le plan obtenu par la NCA .

On obtient un F1-Score moyen de 0.8644.

Nous pouvons constater que la NCA nous a permis de bien séparer les données visuellement et d'améliorer la performance des classifieurs grâce à sa transformation.

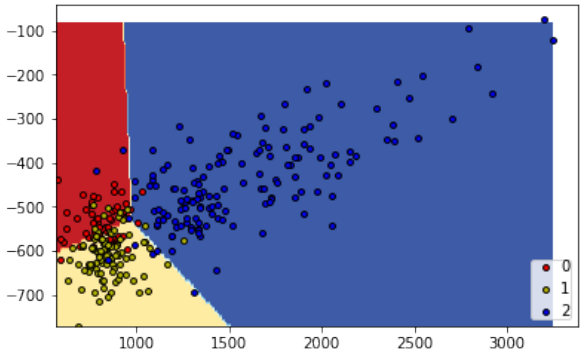

FIGURE 28 – Visualisation des frontières de la régression logistique

6 Conclusion

Nos différentes analyses nous ont permis d'obtenir plusieurs résultats concluants. Dans un premier temps, les méthodes d'analyse non supervisée ont révélé qu'il était difficile de visualiser les classes de manière distinctes. En effet, l'ACP et les F1-Scores obtenus par les analyses ne sont pas pertinents. Ces résultats peuvent venir du fait que les données se prêtent mal à l'apprentissage non supervisé où qu'il ait peut-être fallu effectuer un prétraitement spécifique des données.

Finalement, les méthodes d'analyse supervisées se sont montrées bien plus concluantes. Les données semblent, en effet plus se prêter à l'apprentissage supervisé. Nous avons utilisé la Nested Cross Validation ce qui nous a permis de comparer les scores de chaque méthode supervisée. Nous avons utilisé un grand nombre de méthodes qui utilisent différents principes d'apprentissage. Les scores obtenus avec ces méthodes sont dans l'ensemble assez pertinents. Étant donné que le nombre d'individus présent dans les données est très limité, nous n'avons pas réussi à obtenir une précision aussi haute que nous aurions espérée. La figure 30 montre les scores moyens de nos méthodes d'apprentissages supervisées. Pour la majorité des méthodes utilisées, nous avons obtenu un F1-Score supérieur à 0,80 ce qui est encourageant au vu de notre faible échantillon de données.

Ce projet nous a permis de mettre en pratique ce que nous avons appris au cours de l'UV SY09. Nous avons également dû pousser nos recherches dans le domaine médical afin de mieux comprendre nos données. Il s'agit d'un aspect important à ne pas négliger pour orienter notre travail dans la bonne direction. Nous avons également testé des méthodes d'apprentissages qui n'ont pas été abordés en cours comme la machine à vecteurs de support (SVM) ou encore l'analyse des composantes du voisin (NCA) et qui ont montré de très bons résultats.

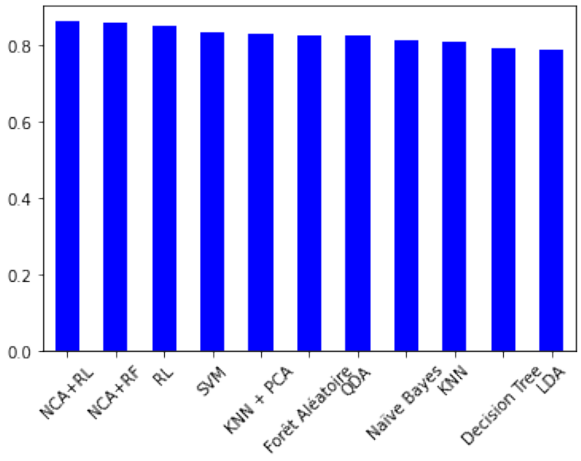

FIGURE 29 – F1-Score moyen des différents classifieurs

Références

[1] Ali Akbar Esmailiejah, Mohamad Qoreishy, Ali Keipourfard, and Shahrza Babaei. Changes in Lumbosacral Angles in Patients with Chronic Low Back Pain : A Prospective Study, 2017.

[2] Casper Hansen. Nested Cross-Validation Python Code.

[3] Geoff Hinton Ruslan Salakhutdinov Jacob Goldberger, Sam Roweis. Neighbourhood Components Analysis. "https://cs.nyu.edu/~roweis/papers/ncanips.pdf".

[4] DR ARUN PAL SINGH. Sacral slope, pelvic tilt and pelvic incidence. https://boneandspine.com/sacral-slope-pelvic-tilt-and-pelvic-incidence/.