Report on Biomechanical features of orthopedic patients

Nathalie Dumont

5/30/2020

Contents

Introduction

This document is my report for the Capstone Project of The Data Science Professional Certificate I am pursuing.
We were supposed to choose a dataset and apply machine learning techniques. I chose a dataset summarizing biomechanical features of orthopedic patients, because my companion suffers from such disease (Hernia). I found adequate to do some research on the subject and establish if a diagnosis can be based on simple observation of verterbraes angles.

We begin by installing some packages :

```
library(formatR)
library(tidyverse)
library(caret)
library(data.table)
library(readr)
library(rpart)
library(ggplot2)
library(dplyr)
library(knitr)
```

Then, we upload the data, originally found on Kaggle from my personal GitHub Repo

```
address <- "https://raw.githubusercontent.com/nmdumont/orthopedic-patients/master/column_3C_weka.csv"
data <- read.csv(url(address))
```

We can now take a look at the data :

```
head(data)
```

```
##   pelvic_incidence pelvic_tilt lumbar_lordosis_angle sacral_slope pelvic_radius
## 1         63.02782   22.552586              39.60912     40.47523      98.67292
## 2         39.05695   10.060991              25.01538     28.99596     114.40543
## 3         68.83202   22.218482              50.09219     46.61354     105.98514
## 4         69.29701   24.652878              44.31124     44.64413     101.86850
## 5         49.71286    9.652075              28.31741     40.06078     108.16872
## 6         40.25020   13.921907              25.12495     26.32829     130.32787
##   degree_spondylolisthesis  class
## 1                -0.254400 Hernia
## 2                 4.564259 Hernia
## 3                -3.530317 Hernia
## 4                11.211523 Hernia
## 5                 7.918501 Hernia
## 6                 2.230652 Hernia
```

```
# We search for NA values
sum(is.na(data))
```

```
## [1] 0
```

```
# Let's look at the structure:
str(data)
```

```
## 'data.frame':    310 obs. of  7 variables:
##  $ pelvic_incidence        : num  63 39.1 68.8 69.3 49.7 ...
##  $ pelvic_tilt             : num  22.55 10.06 22.22 24.65 9.65 ...
##  $ lumbar_lordosis_angle   : num  39.6 25 50.1 44.3 28.3 ...
##  $ sacral_slope            : num  40.5 29 46.6 44.6 40.1 ...
##  $ pelvic_radius           : num  98.7 114.4 106 101.9 108.2 ...
##  $ degree_spondylolisthesis: num  -0.254 4.564 -3.53 11.212 7.919 ...
##  $ class                   : Factor w/ 3 levels "Hernia","Normal",..: 1 1 1 1 1 1 1 1 1 1 ...
```

We have 310 observations of 7 variables : **6 degrees** of various bones and the **indication of disease**.

The goal of this study is to predict the disease (spondylolisthesis, hernia, or no disease) based on the degrees given. We will first explore the data a bit more, look for correlation, and use several machine learning techniques to predict the disease.

Analysis

The data provided by Kaggle is already clean. Reading the .csv file gave us the degrees and the indication of the disease as a factor.
We have not yet check the factor class :

```
levels(data$class)
```

```
## [1] "Hernia"           "Normal"           "Spondylolisthesis"
```

```
str(unique(data))
```

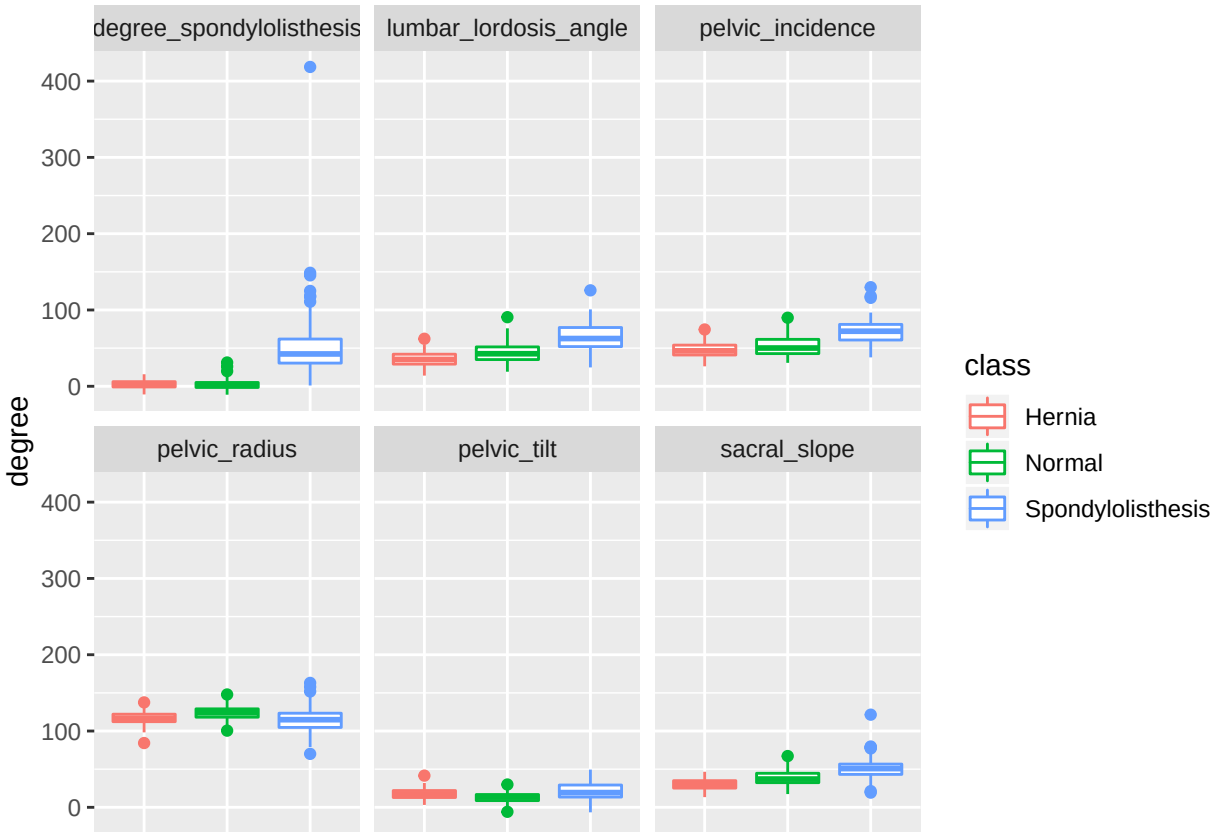
```
## 'data.frame':    310 obs. of  7 variables:
##  $ pelvic_incidence        : num  63 39.1 68.8 69.3 49.7 ...
##  $ pelvic_tilt             : num  22.55 10.06 22.22 24.65 9.65 ...
##  $ lumbar_lordosis_angle   : num  39.6 25 50.1 44.3 28.3 ...
##  $ sacral_slope            : num  40.5 29 46.6 44.6 40.1 ...
##  $ pelvic_radius           : num  98.7 114.4 106 101.9 108.2 ...
##  $ degree_spondylolisthesis: num  -0.254 4.564 -3.53 11.212 7.919 ...
##  $ class                   : Factor w/ 3 levels "Hernia","Normal",..: 1 1 1 1 1 1 1 1 1 1 ...
```

So our patients can't have both Spondylolisthesis *and* Hernia, and all observations are unique observations.

Data exploration

For a first approach, we can visualize how each angle reflects a disease, or not. If a certain angle takes certain value only for hernia or spondylolisthesis, it could be a good predictor. Let's see how each angle has an incidence over the defect :

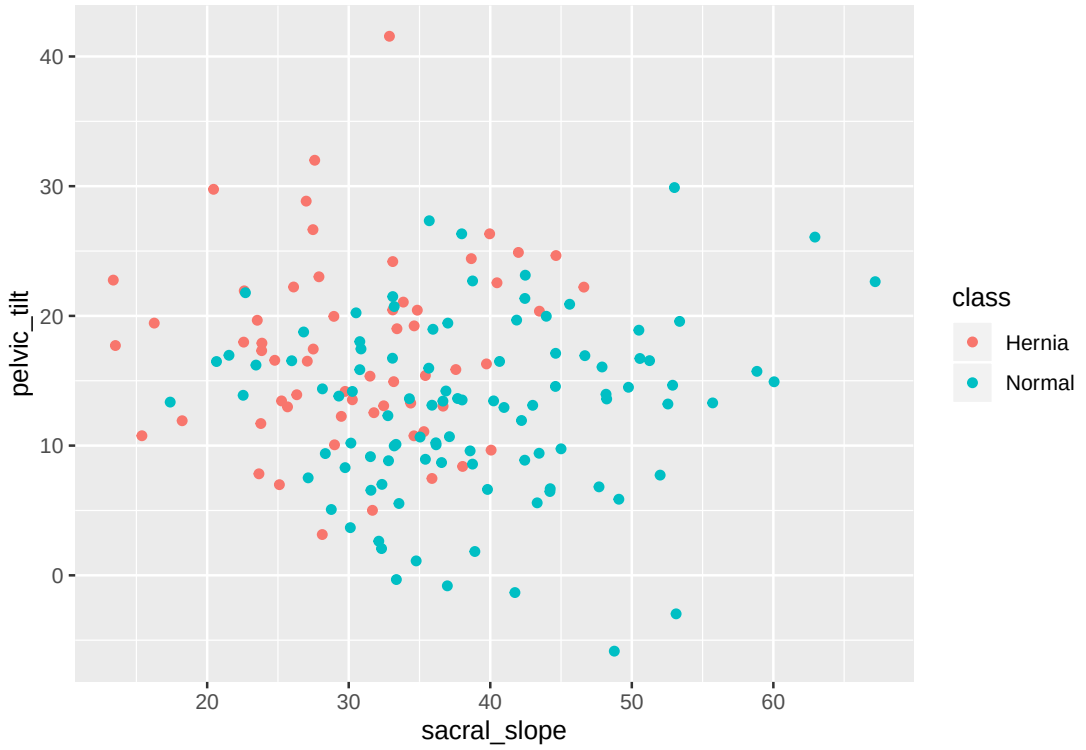
```
data %>% gather(angle, degree, -class) %>% ggplot(aes(class,
    degree, col = class)) + geom_boxplot() + facet_wrap(~angle) +
    theme(axis.title.x = element_blank(), axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```



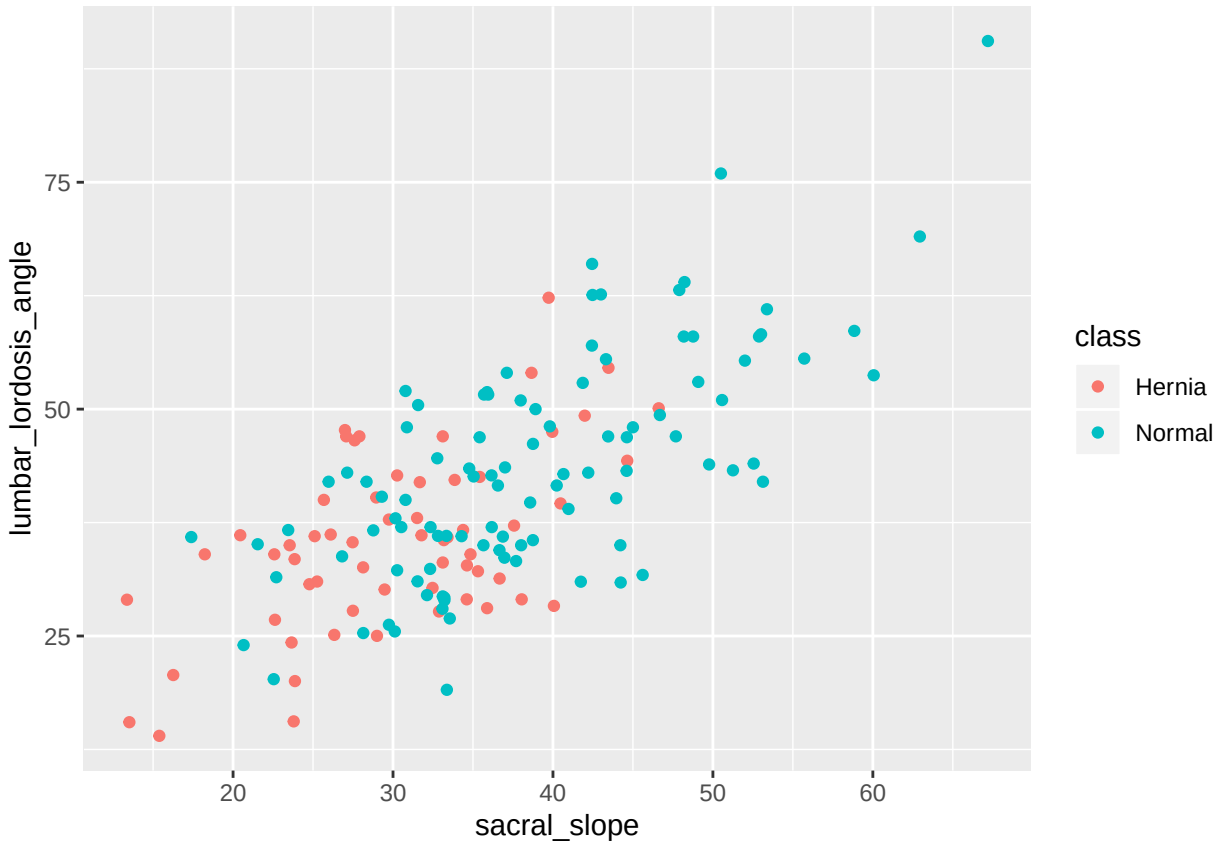
It seems clearly that spondylolisthesis angle is a good predictor for spondylosisthesis.

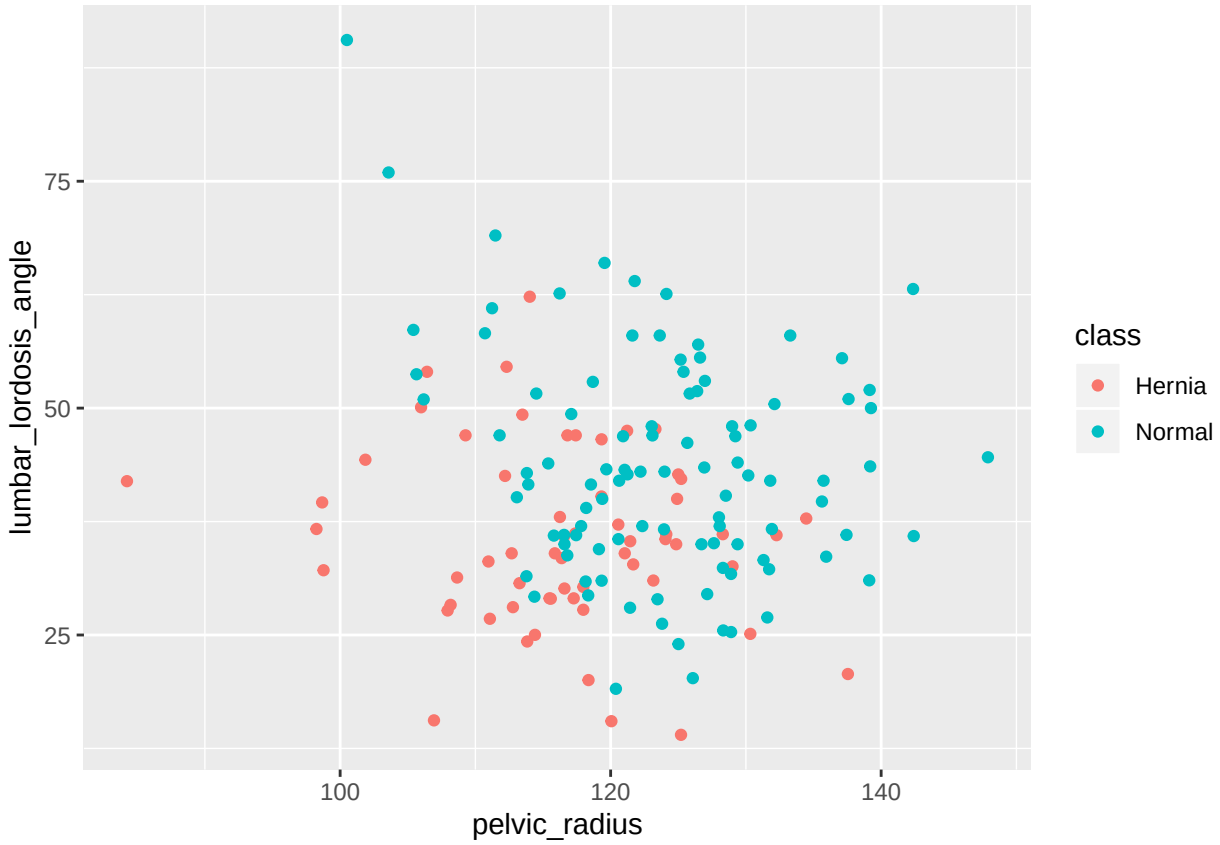
We do not have a clear dinstinction between hernia and normal for the various angles, except maybe for sacral slope. Could sacral slope and pelvic tilt be a good combination of predictors ? I want to make a plot to see if there is any. A bit arbitrarily I select those two variables to predict the "disease".

```
data %>% filter(class != "Spondylolisthesis") %>% ggplot(aes(sacral_slope,
    pelvic_tilt, col = class)) + geom_point()
```



All seem quite intricate for those two variables. The larger values of sacral slope reflect normal patients, whatever the pelvic tilt. We can visualise other couples of variables :





The areas for Hernia and Normal are overlapping. That motivates our study to establish a model to predict

Hernia for patients, based on the angles of their vertebraes.

It is time to separate the dataset into test and training set.

Test and training set

```
set.seed(1985)
test_index <- createDataPartition(data$class, times = 1, p = 0.2,
    list = FALSE)  # create a 20% test set
test_set <- data[test_index, ]
train_set <- data[-test_index, ]
```

Now that we have two different sets, one of 248 rows for training, and one of 62 for testing, we can try to predict the disease.

Modeling

We are trying to guess the disease from random :

```
guess <- sample(c("Hernia", "Normal", "Spondylolisthesis"), nrow(test_set),
    replace = TRUE)
guess_acc <- mean(guess == test_set$class)
```

Guessing is not a good method, as expected. We will save our results in a tibble to see how it improves, depending on the method used.

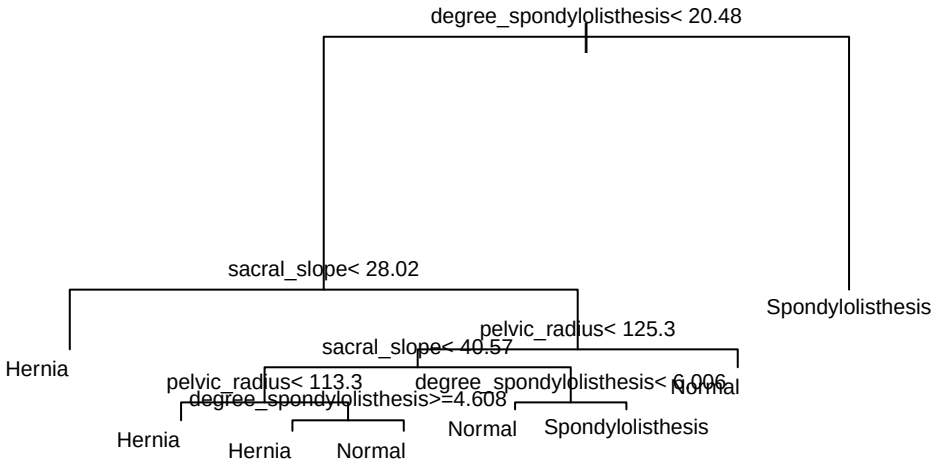
Method	Accuracy
Guess	0.3064516

We have a lot of predictors, 6, so I want to work classification and regression tree, as taught in the course.

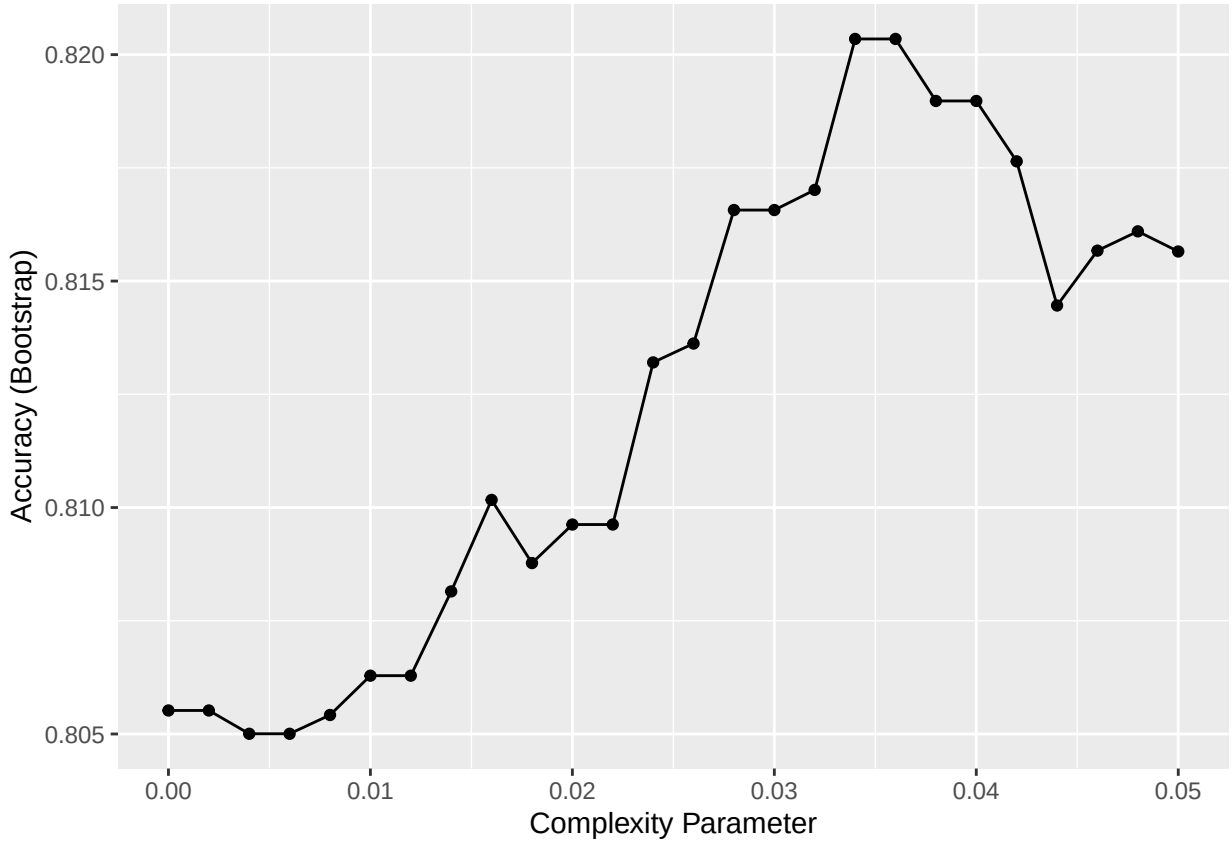
Regression tree

We have seen that the degree of spondylolisthesis is a good indicator of the disease. Using classification tree seems a good option to predict disease.

```
fit <- rpart(class ~ ., data = train_set)
plot(fit, margin = 0.1)
text(fit, cex = 0.65)
```




```
train_rpart <- train(class ~ ., method = "rpart", tuneGrid = data.frame(cp = seq(0,
    0.05, 0.002)), data = train_set)
ggplot(train_rpart, hightlight = TRUE)
```




```
rpart_preds <- predict(train_rpart, test_set)
confusionMatrix(rpart_preds, test_set$class)
```

```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Hernia Normal Spondylolisthesis
##   Hernia                 8     10                 0
```

```
##   Normal                 4     10                 2
##   Spondylolisthesis      0      0                28
##
## Overall Statistics
##
##                Accuracy : 0.7419
##                  95% CI : (0.615, 0.8447)
##     No Information Rate : 0.4839
##     P-Value [Acc > NIR] : 3.151e-05
##
##                   Kappa : 0.5981
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Hernia Class: Normal Class: Spondylolisthesis
## Sensitivity                 0.6667        0.5000                    0.9333
## Specificity                 0.8000        0.8571                    1.0000
## Pos Pred Value              0.4444        0.6250                    1.0000
## Neg Pred Value              0.9091        0.7826                    0.9412
## Prevalence                  0.1935        0.3226                    0.4839
## Detection Rate              0.1290        0.1613                    0.4516
## Detection Prevalence        0.2903        0.2581                    0.4516
## Balanced Accuracy           0.7333        0.6786                    0.9667
```

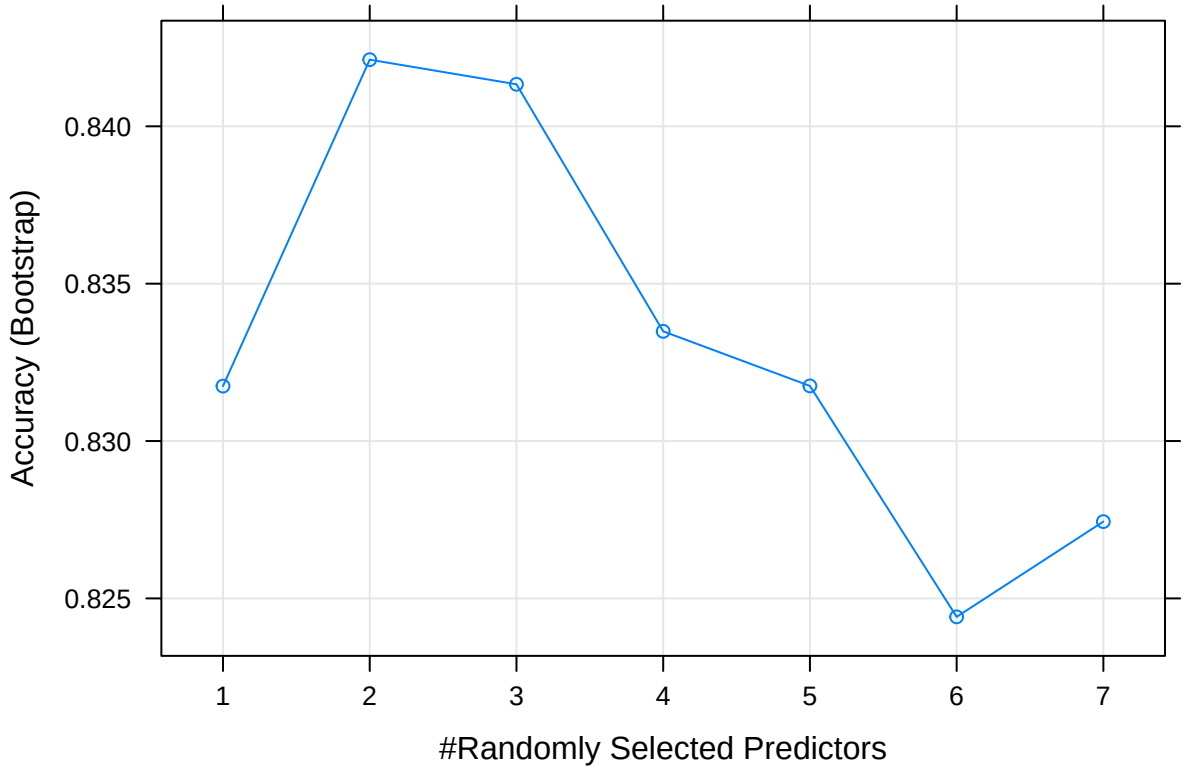
We have a really good sensitivity *and* specificity for predicting Spondylolisthesis. We can see how a random forest algorithm manages.

Method	Accuracy
Guess	0.3064516
Regression tree	0.7419355

Random forest

We try to compute a random forest algorithm over our training set.

```
train_rf <- train(class ~ ., data = train_set, method = "rf",
    ntree = 100, tuneGrid = data.frame(mtry = seq(1:7)))
plot(train_rf)
```




```
train_rf$bestTune
```

```
##   mtry
## 2    2
```

```
varImp(train_rf$finalModel)
```

```
##                            Overall
## pelvic_incidence          19.59884
## pelvic_tilt               11.23378
## lumbar_lordosis_angle     22.66449
## sacral_slope              22.29822
## pelvic_radius             19.08677
## degree_spondylolisthesis 58.78831
```

Sacral slope and Lumbar Lordosis angle are the two most important variables (after Spondylolisthesis degree, obvioulsy) but not by far. Checking the results :

Method	Accuracy
Guess	0.3064516
Regression tree	0.7419355
Random forest	0.7580645

K-Nearest-Neighbours

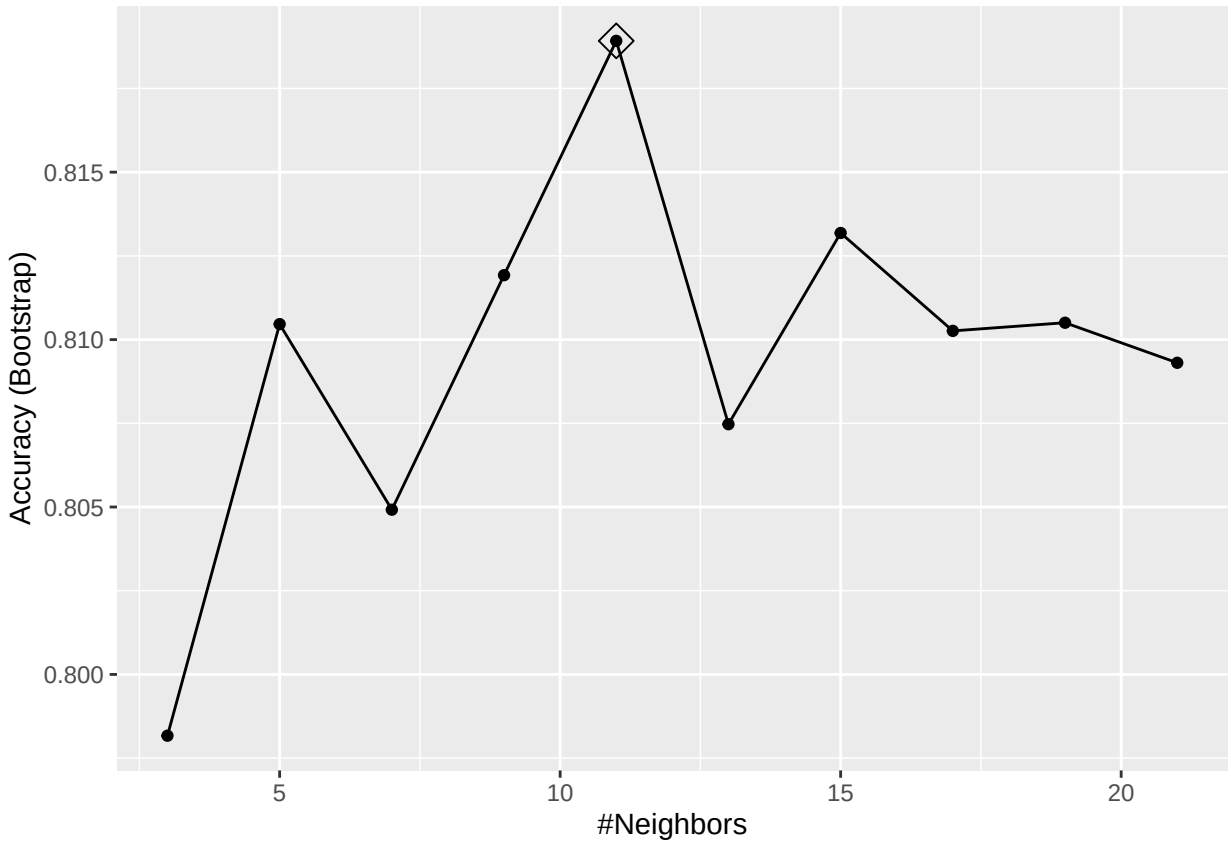
We can try to train a KNN regression.

```
knn_fit <- knn3(class ~ ., data = train_set, k = 5)
confusionMatrix(predict(knn_fit, test_set, type = "class"), test_set$class)$overall["Accuracy"]
```

```
##  Accuracy
## 0.7903226
```

We can do better by choosing a better number of neighbours :

```
train_knn <- train(class ~ ., data = train_set, method = "knn",
    tuneGrid = data.frame(k = seq(3, 21, 2)))
ggplot(train_knn, highlight = TRUE)
```




```
train_knn$bestTune
```

```
##    k
## 5 11
```

Method	Accuracy
Guess	0.3064516
Regression tree	0.7419355
Random forest	0.7580645
KNN	0.7903226

KNN using cross-validation

We can optimize a bit using cross-validation :

```
train_knn_cv <- train(class ~ ., method = "knn", data = train_set,
    tuneGrid = data.frame(k = seq(3, 51, 2)), trControl = trainControl(method = "cv",
        number = 10, p = 0.9))
```

```
train_knn_cv$bestTune
```

```
##   k
## 3 7
```

```
knn_cv_preds <- predict(train_knn_cv, test_set)
```

Method	Accuracy
Guess	0.3064516
Regression tree	0.7419355
Random forest	0.7580645
KNN	0.7903226
KNN Cross-validation	0.8064516

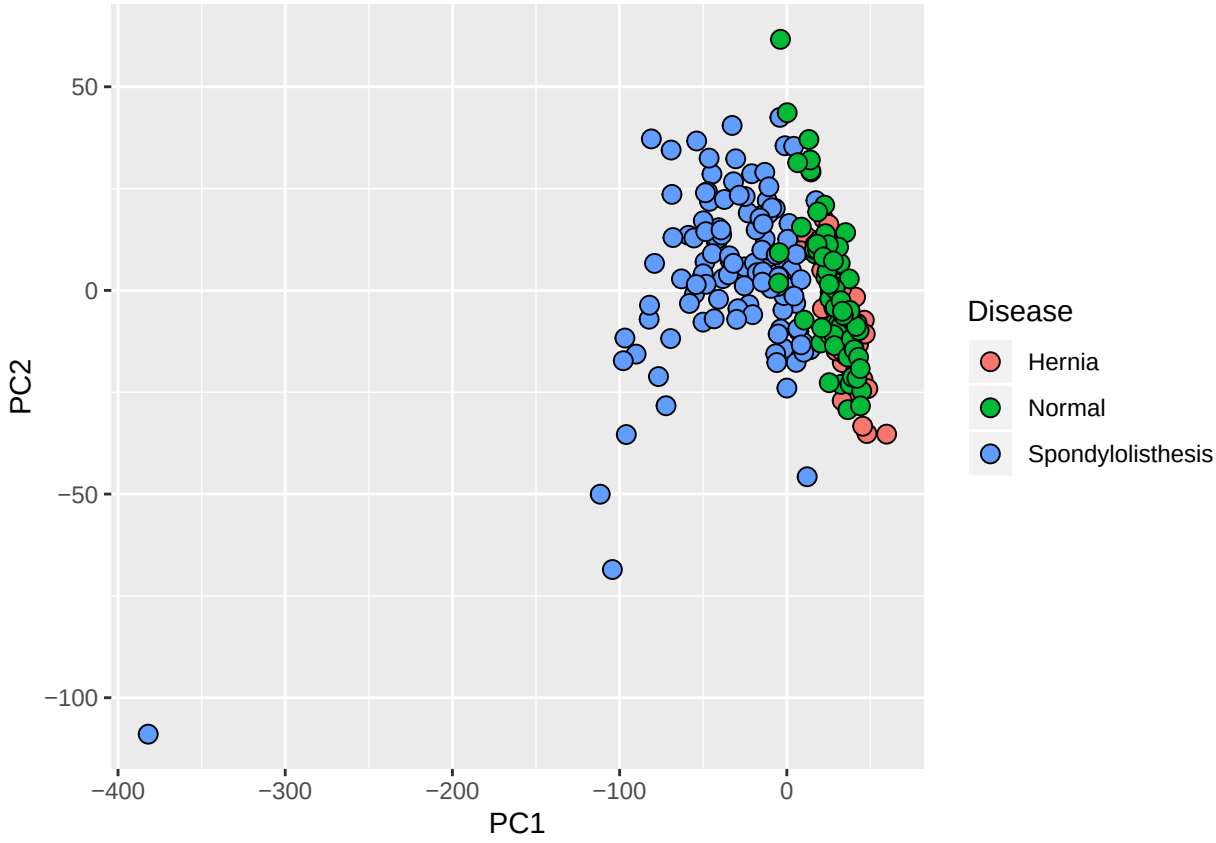
Results section

Here are the results obtained from the several models :

Method	Accuracy
Guess	0.3064516
Regression tree	0.7419355
Random forest	0.7580645
KNN	0.7903226
KNN Cross-validation	0.8064516

Maybe trying to run a principal component analysis could have helped us.

PCA




```
## Importance of components:
##                            PC1     PC2      PC3      PC4    PC5       PC6
## Standard deviation     44.0443 19.0952 13.66702 10.16952 9.6578 2.658e-09
## Proportion of Variance  0.7217  0.1356  0.06949  0.03847 0.0347 0.000e+00
## Cumulative Proportion   0.7217  0.8573  0.92683  0.96530 1.0000 1.000e+00
```

When running a Principal Component Analysis, we see that the two principal components can be held responsible for more that 85% of the variance. But it seems quite intricate between Hernia and Normal. Having ongly 6 predictors does not seem enough for PCA. But we can try to do something with only the 3 first components.

```
x_train <- pca$x[, 1:3]
y <- factor(train_set$class)
fit <- knn3(x_train, y)
z <- test_set[, -7] %>% as.matrix()
x_test <- sweep(z, 2, colMeans(z, na.rm = TRUE)) %*% pca$rotation
x_test <- x_test[, 1:3]
y_hat <- predict(fit, x_test, type = "class")
confusionMatrix(y_hat, factor(test_set$class))$overall["Accuracy"]
```

```
##  Accuracy
## 0.7580645
```

```
accuracy_results <- bind_rows(accuracy_results, tibble(Method = "PCA",
    Accuracy = confusionMatrix(y_hat, factor(test_set$class))$overall["Accuracy"]))
```

Using the 3 principal component does not give us more accuracy in our model. To be honest, I am not

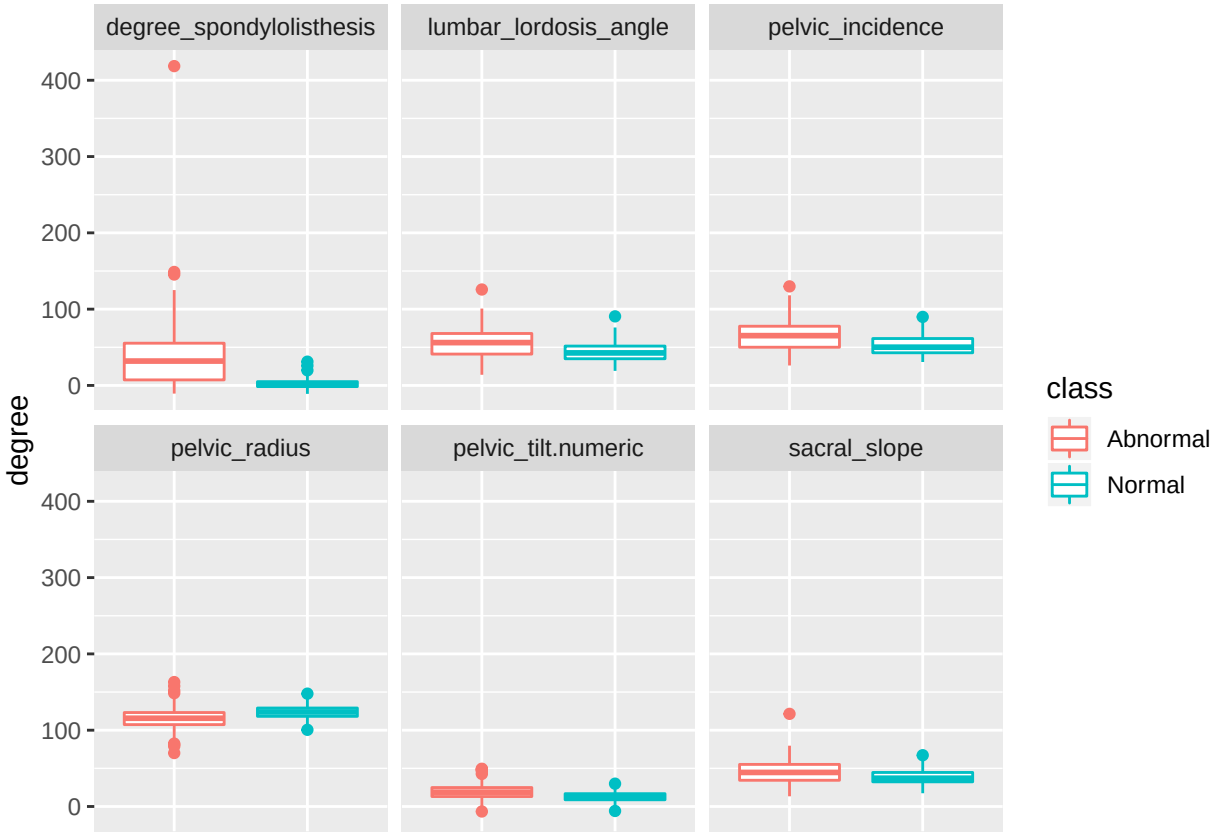
completely sure if I am using PCA with the right purpose.

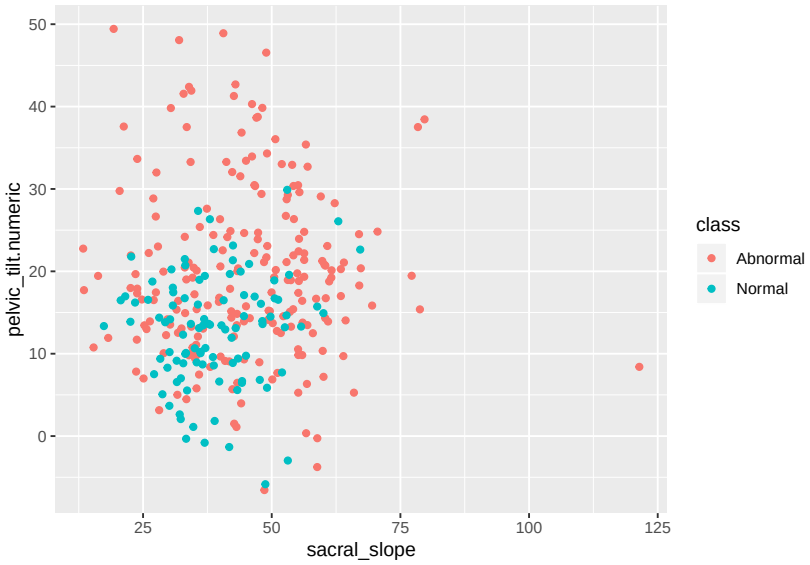
I acknowledge that my final model, KNN with cross-validation, obtaining merely 87% good results, could be improved. Perhaps it would have been better to use it on the other dataset provided by Kaggle, without distinction between the diseases (Spondylolisthesis and Hernia being both "disease", whereas normal stays "normal").

The other dataset provided

I try to do so in the following section :

```
address <- "https://raw.githubusercontent.com/nmdumont/orthopedic-patients/master/column_2C_weka.csv"
data_2 <- read.csv(url(address))
```





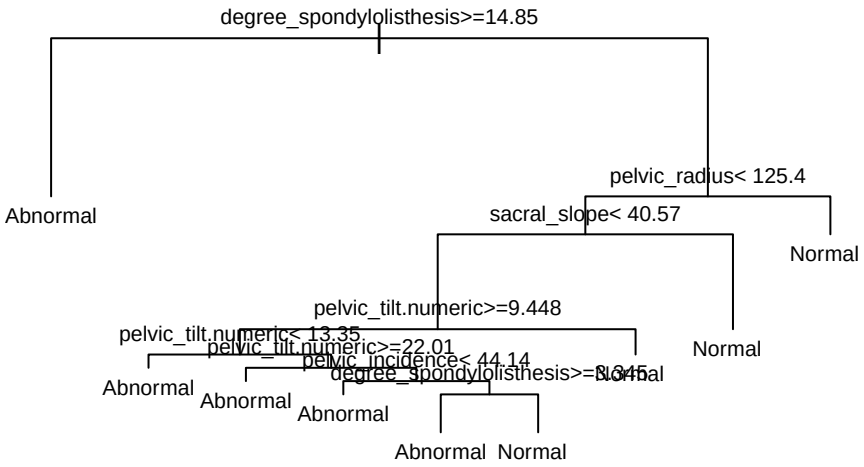
All seem quite intricate for those two variables, even more than before. We can visualise other couples of variables :

The areas for Hernia and Normal are overlapping.

Modeling

I use the same models on this data set :

Regression tree



Method	Accuracy
Regression tree	0.8225806

The tree looks a bit different, but we have the same accuracy.

Random forest

We try to compute a random forest algorithm over our training set.

```
train_rf_2 <- train(class ~ ., data = train_set_2, method = "rf",
    ntree = 100, tuneGrid = data.frame(mtry = seq(1:7)))
varImp(train_rf_2$finalModel)
```

```
##                            Overall
## pelvic_incidence          12.92088
## pelvic_tilt.numeric       13.57203
## lumbar_lordosis_angle     13.17216
## sacral_slope              11.32253
## pelvic_radius             19.27200
## degree_spondylolisthesis 36.88257
```

Sacral slope and Lumbar Lordosis angle are no loger the two most important variables (after Spondylolisthesis degree). Checking the results :

Method	Accuracy
Regression tree	0.8225806
Random forest	0.8548387

K-Nearest-Neighbours

We can try to train a KNN regression.

Method	Accuracy

Method	Accuracy
Regression tree	0.8225806
Random forest	0.8548387
KNN	0.8709677
KNN Cross-validation	0.8709677

Method	Accuracy
Regression tree	0.8225806
Random forest	0.8548387
KNN	0.8709677
KNN Cross-validation	0.8709677
PCA	0.7741935

So we have the results from the first data set

Method	Accuracy
Guess	0.3064516
Regression tree	0.7419355
Random forest	0.7580645
KNN	0.7903226
KNN Cross-validation	0.8064516
PCA	0.7580645

and the second data set

Method	Accuracy
Regression tree	0.8225806
Random forest	0.8548387
KNN	0.8709677
KNN Cross-validation	0.8709677
PCA	0.7741935

The KNN algorithm on the second dataset seems already optimized, since cross-validation does not give something better. I am not sure why the second dataset is best, since I used the same algorithms. Except from the way the class of the disease is categorised, there is no other difference.

I have also observed that the results are highly influenced by the number I use in the **set.seed** function. I do not have any input on this subject.

Conclusion

I tried to use the various models taught in the Data Science course. I did not include some clustering techniques, as I was not able to achieve them. Nevertheless, I can predict with almost 90% accuracy the presence of Hernia or Spondylolisthesis, based on a few parameters.

Maybe this model could be used to help prevent patients from suffering from Hernia or Spondylolisthesis, if only I could have drown a map based on the 6 parameters. I have no idea how that could be done, since I should develop a model showing tendencies. I wish I were able to compute an algorithm containing several parts, the first one being using tree classification to detect Sponylolisthesis and then using KNN to predict Hernia. Sadly, I was not able to do that.

I am glad to have done this analysis, even if it took me some time, having to view some contents again, searching for examples in the coursebook, on Stackoverflow, and on various resources such as knitr's website.