Analysis of Biomechanical Features of Orthopedic Patients

Nirmal Sai Swaroop Janapaneedi

25/05/2020

Contents

Chapter 1

Introduction

A patient's orthopedic health condition can be detected from his biomechanical features. Biomechanics is a branch of biophysics (1). It is the study of the structure, function and motion of the mechanical aspects of biological systems using the methods of mechanics (1). The condition when an injury occurs to the cushioning and connective tissue between vertebrae is termed disc herniation (2). Spondylolisthesis is a medical condition in which one of the vertebrae slips out of place onto the bone below it(3). Spondylolisthesis is different than a herniated disc, though the two can coexist. With a herniated disc, the soft interior of the spinal disc bulges through a tear in the outer layer of the disc, wheras with spondylolisthesis, the slippage is of the bony vertebra (4). Depending on the changes they make in the patient's biomechanical features, the disease can be predicted.

Machine learning algorithms in medical fields have widely been used in disease prediction as such approaches may be considered of great assistance in the decision making process of medical practitioners.

1.1 Objective

The aim of this project is to train machine learning models to predict whether a patient is normal, or has spondylolisthesis or disc herniation based on the biomechanical features provided. The goal is to find an appropriate algorithm with high accuracy combined with a high sensitivity and specificity.

1.2 Dataset

In this project, I will be using a dataset provided by UCI Machine Learning Repository(https://www.kaggle.com/uciml/biomechanical-features-of-orthopedic-patients). The original dataset was downloaded from UCI ML repository: Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

The dataset comprises 310 patients, of which each patient is represented by six biomechanical attributes derived from the shape and orientation of the pelvis and lumbar spine: pelvic incidence, pelvic tilt, lumbar lordosis angle, sacral slope, pelvic radius and grade of spondylolisthesis. The patients are classified into three classes: normal, disc, and spondylolisthesis.

```
dat$class <- as.factor(dat$class)
```

Chapter 2

Methods and Analysis

2.1 Data Analysis

I will start by examining the structure of our dataset:

```
str(dat)
```

```
'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",..: 1111111111...
```

```
head(dat)
```

```
  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
```

The dataset contains 7 variables and 310 observations.

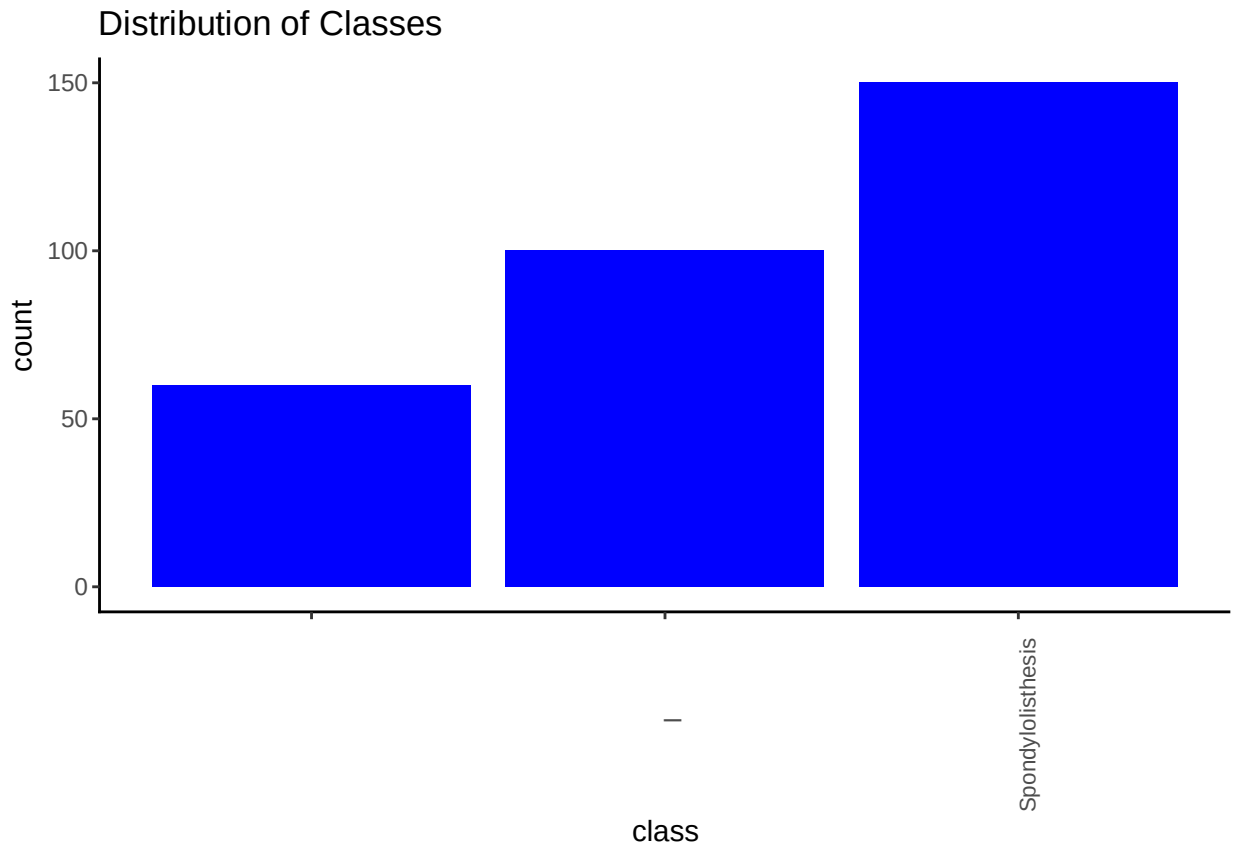
We have to check if the dataset contains any missing values

```
$pelvic_incidence
[1] 0

$pelvic_tilt
[1] 0

$lumbar_lordosis_angle
[1] 0

$sacral_slope
[1] 0

$pelvic_radius
[1] 0

$degree_spondylolisthesis
[1] 0

$class
[1] 0
```

It appears that there are no NA values.

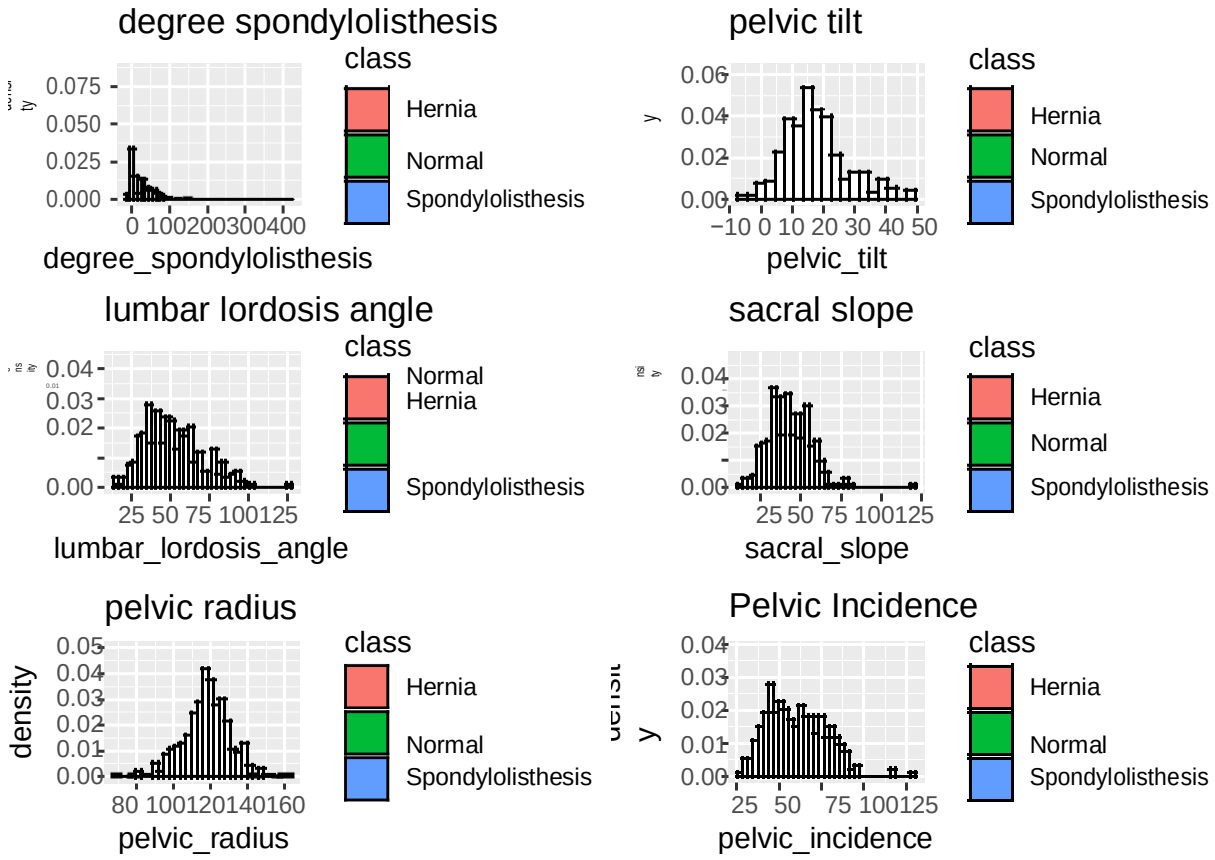
Now let's examine the distribution of the three classes

```
dat %>% ggplot(aes(class)) +
  geom_bar(stat="count", fill = "blue", alpha = 0.5) +
  theme_classic() + theme(axis.text.x = element_text(angle = 90)) + labs(title = "Distribution of Class
```

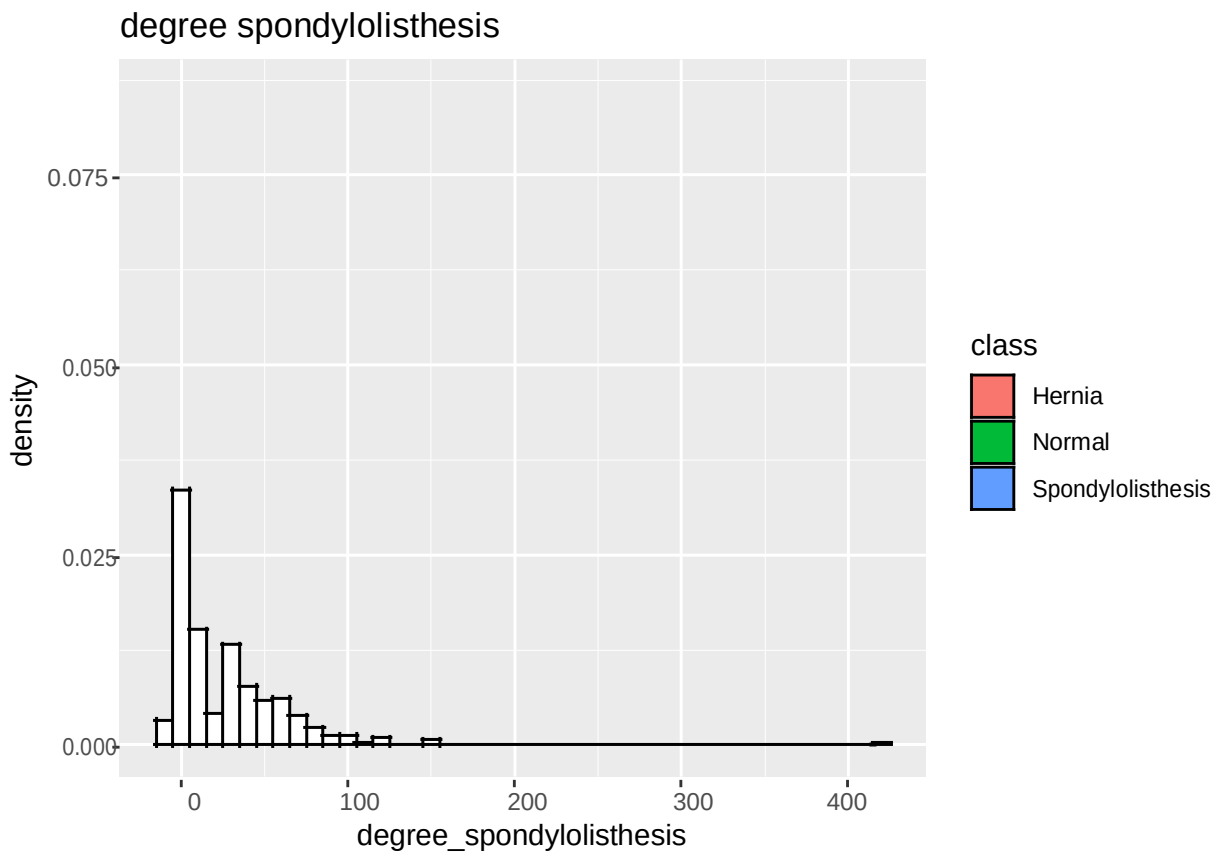


The variables in the dataset show a nearly normal distribution as presented in the plots below

```
fig1 <- dat %>% ggplot(aes(degree_spondylolisthesis, fill=class)) + geom_histogram(aes(y=..density..), fig2 <- dat
%>% ggplot(aes(pelvic_tilt, fill=class)) + geom_histogram(aes(y=..density..), binwidth = 3, fig3 <- dat %>%
ggplot(aes(x=lumbar_lordosis_angle, fill=class)) + geom_histogram(aes(y=..density..), b fig4 <- dat %>%
ggplot(aes(sacral_slope,fill=class)) + geom_histogram(aes(y=..density..), binwidth = 3, fig5 <- dat %>%
ggplot(aes(pelvic_radius, fill=class)) + geom_histogram(aes(y=..density..), binwidth = fig6 <- dat %>%
ggplot(aes(x=pelvic_incidence,fill=class)) + geom_histogram(aes(y=..density..), binwidt grid.arrange(fig1, fig2, fig3,
fig4, fig5, fig6)
```

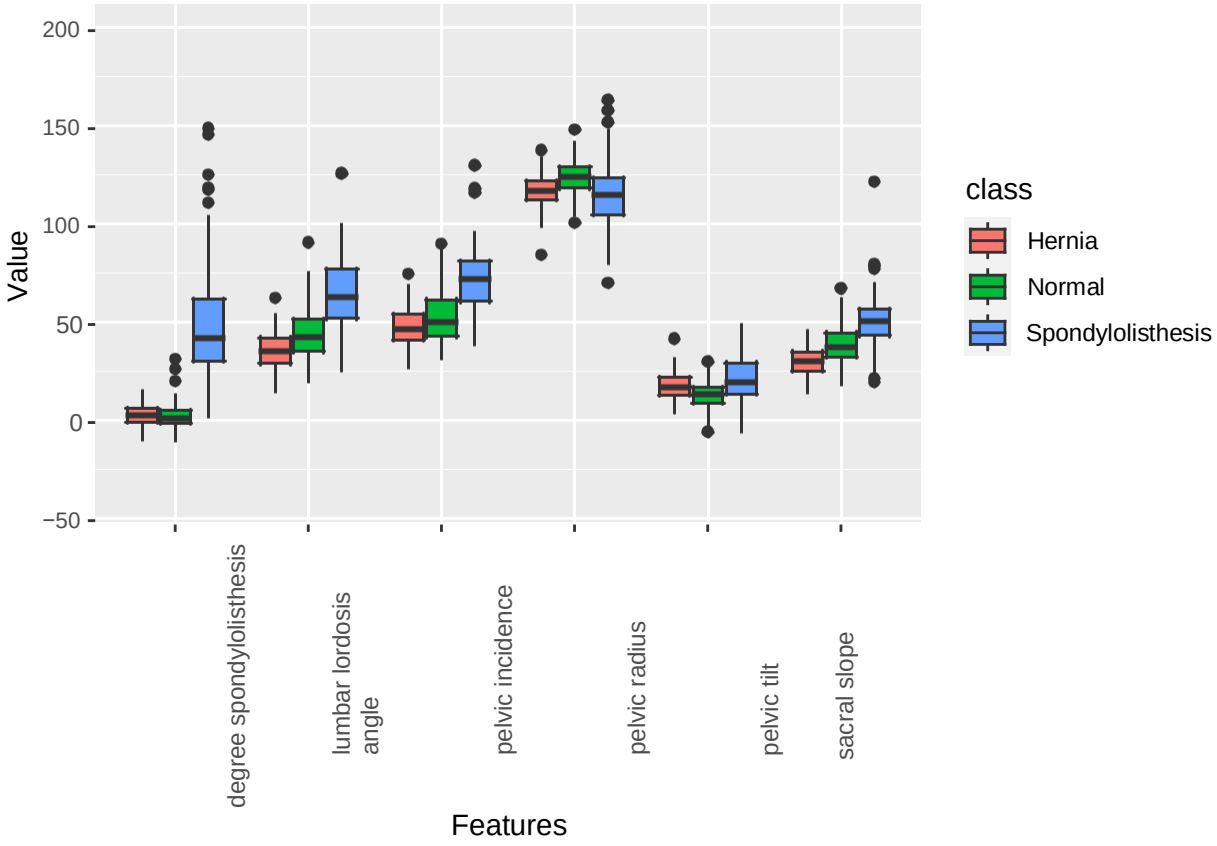


By examining the plots, the density plot of the degree spondylolisthesis variable caught my attention in that it shows that it may provide a good separation between patients with spondylolisthesis and ones that have hernia and normal patients. Let's keep this variant in mind for further analysis.



Here's a boxplot for all the variables

```
b1 <- dat %>% gather(Features, Value, -class) %>%
  ggplot(aes(Features, Value, fill = class)) +
  geom_boxplot() + theme(axis.text.x = element_text(angle = 90)) + xlab("Features") + ylim(-40, 200)

b1
```



Let's check the correlation between the variables by performing a correlation matrix since machine learning algorithms assume that the predictor variables are independent from one another. Before that, we need to convert our dataset into a matrix followed by scaling.

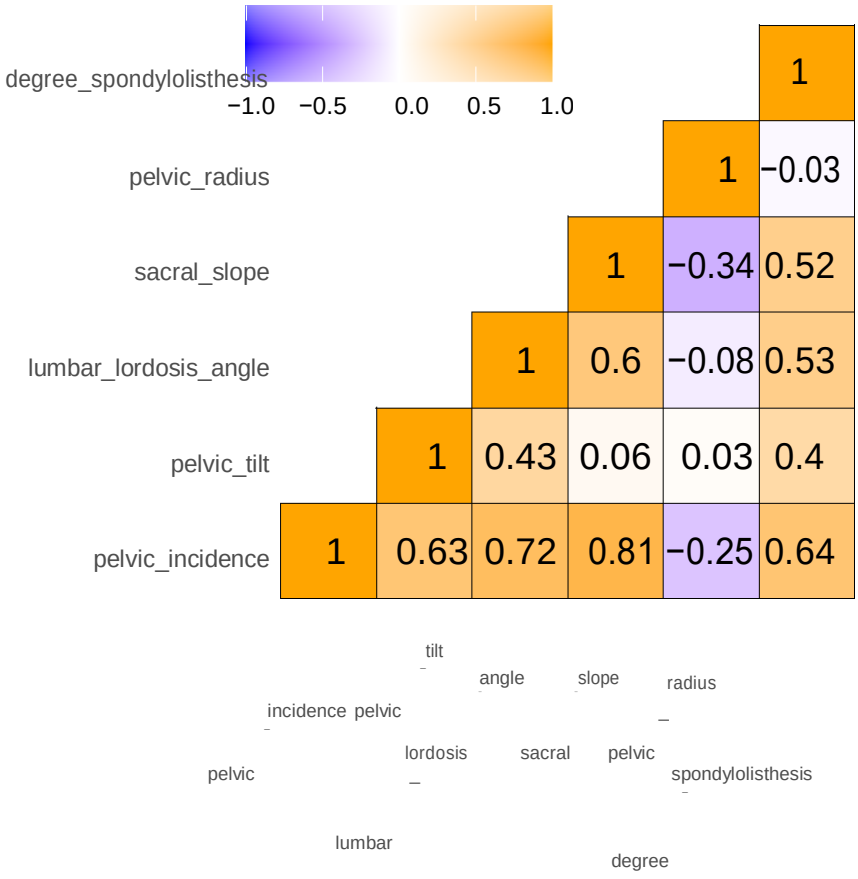
```
#Converting dat into a matrix so we can perform scaling

dat.matrix <- dat %>% select(-class) %>% as.matrix()

#Scaling of dat.matrix by substracting the columns' means and deviding by the column's starnard deviati dat_centered <-
sweep(dat.matrix, 2, colMeans(dat.matrix))
dat_scaled <- sweep(dat_centered, 2, colSds(dat.matrix), FUN = "/")
```

```
#Creating the correlation matrix
cor.matrix <- round(cor(dat.matrix), 2)

cor.matrix_metled <- melt(cor.matrix)
head(cor.matrix_metled)
```

```
                          Var1                 Var2 value
1              pelvic_incidence pelvic_incidence  1.00
2                    pelvic_tilt pelvic_incidence  0.63
3       lumbar_lordosis_angle pelvic_incidence  0.72
4                  sacral_slope pelvic_incidence  0.81
5                 pelvic_radius pelvic_incidence -0.25   6
degree_spondylolisthesis pelvic_incidence 0.64
```

```
# Get the lower triangle of the correlation matrix
get_lower_tri <- function(cor.matrix){
  cor.matrix[upper.tri(cormat.matrix)] <- NA
  return(cor.matrix)
}

# Get the upper triangle of the correlation matrix
get_upper_tri <- function(cor.matrix){
  cor.matrix[lower.tri(cor.matrix)]<- NA
  return(cor.matrix)
}

upper_tri <- get_upper_tri(cor.matrix)

melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Heatmap
# Use correlation between variables as distance

reorder_cormat <- function(cor.matrix){
  dd <- as.dist((1-cor.matrix)/2) hc <-
  hclust(dd)
  cor.matrix <-cor.matrix[hc$order, hc$order]
}

# Reorder the correlation matrix cormat <-
reorder_cormat(cor.matrix) upper_tri <-
get_upper_tri(cor.matrix)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "black")+
  scale_fill_gradient2(high = "orange", low = "blue", mid = "white", midpoint = 0,
                       limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +

  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 10,
                                   hjust = 1))+
  coord_fixed() +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 5) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    legend.justification = c(1, 0),
    legend.position = c(0.5, 0.8),
    legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 8, barheight = 2, title.position = "top",
                               title.hjust = 0.5))
```

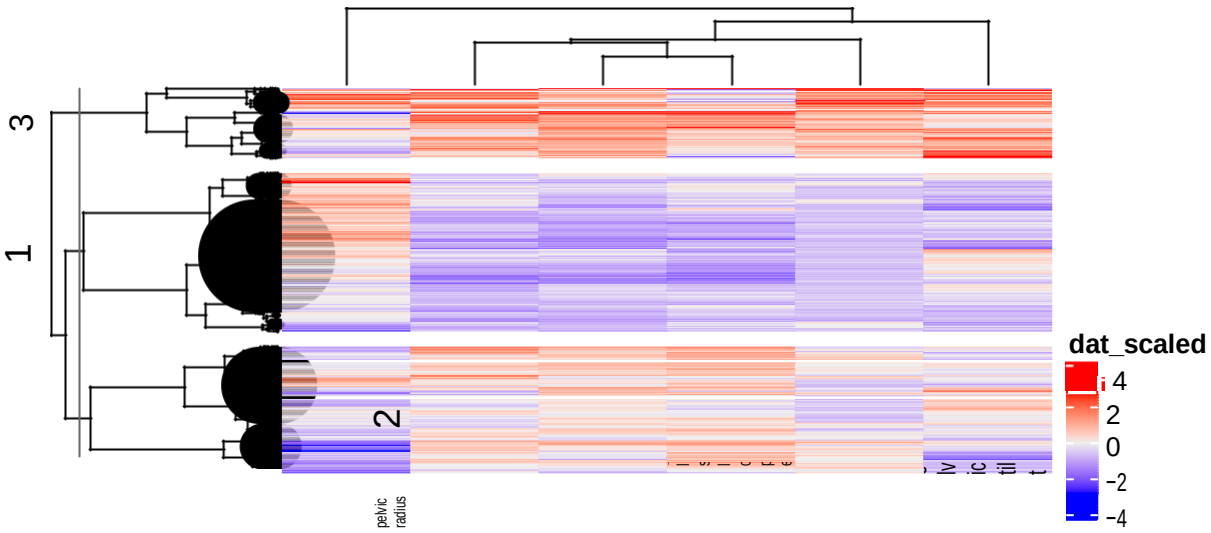
ggheatmap



As we can see from the correlation matrix, lumbar_lardosis_angle and sacral_slope are moderately correlated with degree_spondylolisthesis and, interestingly, pelvic_radius and degree_spondylolisthesis are not correlated.

We can see that the data can be clustered into three groups corresponding to the three classes from the following heatmap

```
Heatmap(dat_scaled, clustering_distance_rows = "maximum",
        clustering_method_rows = "ward.D", row_dend_width = unit(3, "cm"),
        show_row_names = TRUE, km = 3,   gap = unit(2, "mm"),       name = "dat_scaled")
```



2.2 Modelling Approach

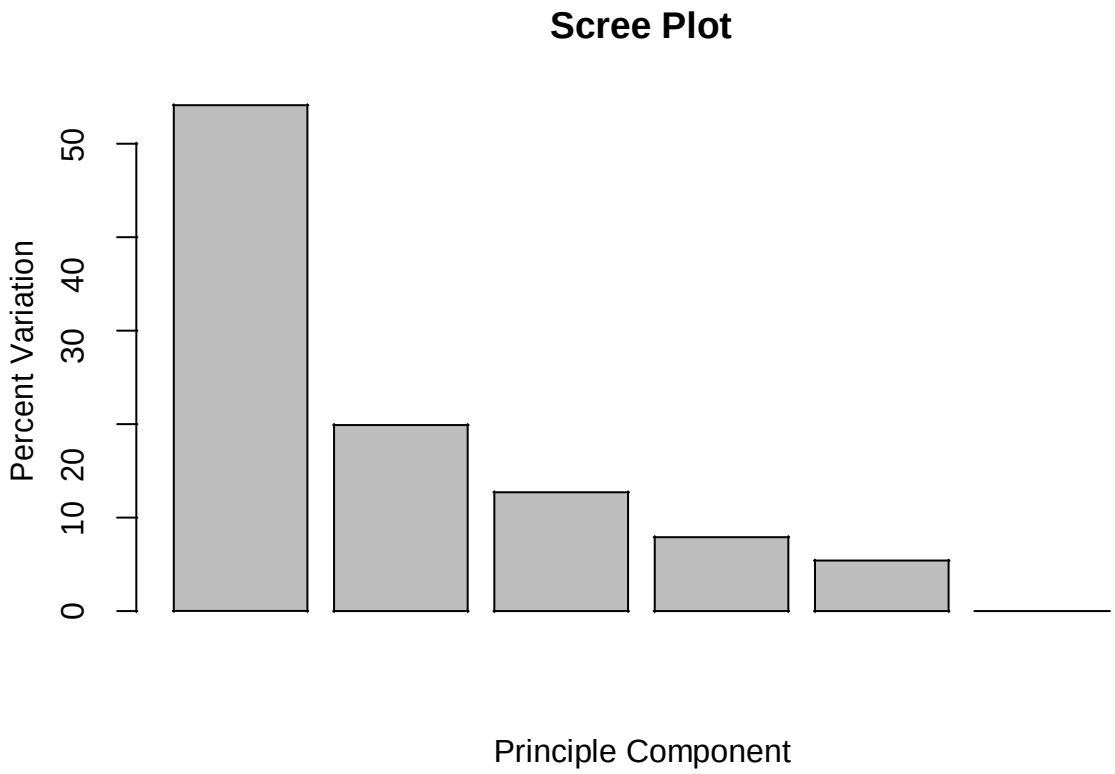
2.2.1 Modelling

Let's perform principle component analysis to examine the clustering of the data.

```
pca <- prcomp(dat_scaled)
summary(pca)
```

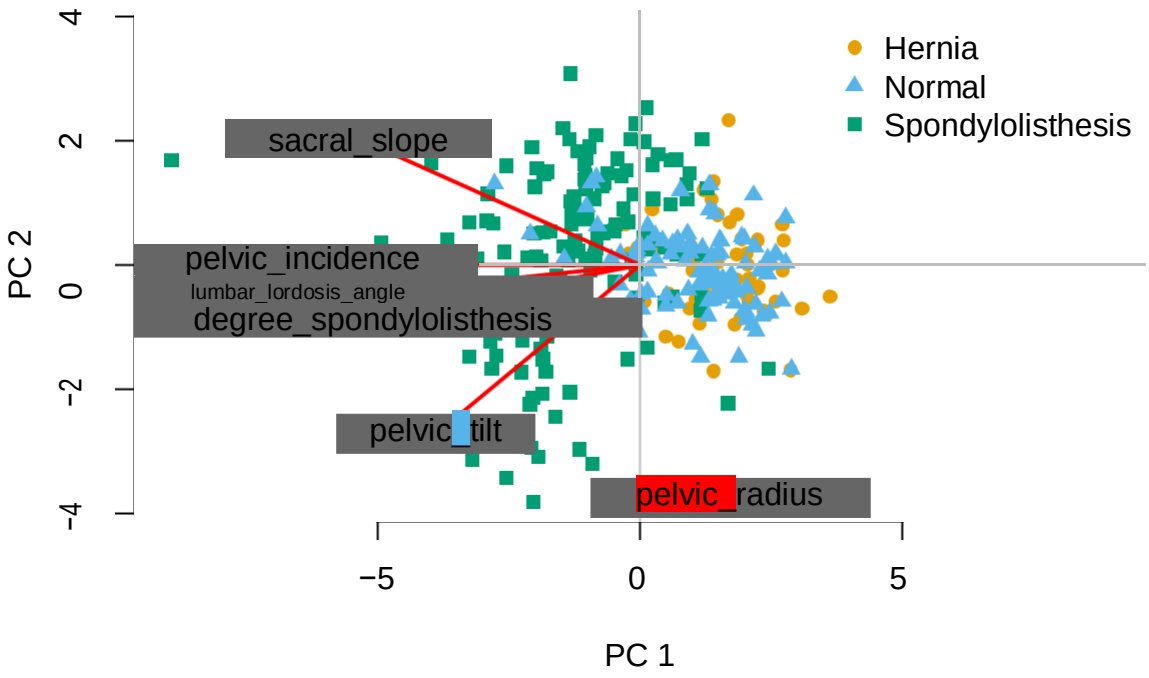
```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5      PC6
Standard deviation     1.802 1.0930 0.8724 0.68741 0.57098 1.935e-10
Proportion of Variance 0.541 0.1991 0.1268 0.07875 0.05434 0.000e+00
Cumulative Proportion  0.541 0.7401 0.8669 0.94566 1.00000 1.000e+00
```

```
#Scree plot

pca.var <- pca$sdev^2
pca.var.per <- round(pca.var/sum(pca.var)*100,1)
barplot(pca.var.per, main = "Scree Plot", xlab = "Principle Component", ylab = "Percent Variation")
```



We can see from the summary function and the scree plot that around 95% of the variance can be explained from the first four principle components. Let's plot the first two/three principle components with color representing disease class and plotting the variables.

```
gr <- dat$class
#Plotting the first two PCs
pca2d(pca, group=gr, biplot=TRUE, biplot.vars=4, legend="topright")
```




```
#Plotting the first three PCs
pca3d(pca, group = gr, biplot=TRUE, biplot.vars=4, legend="topright")
```

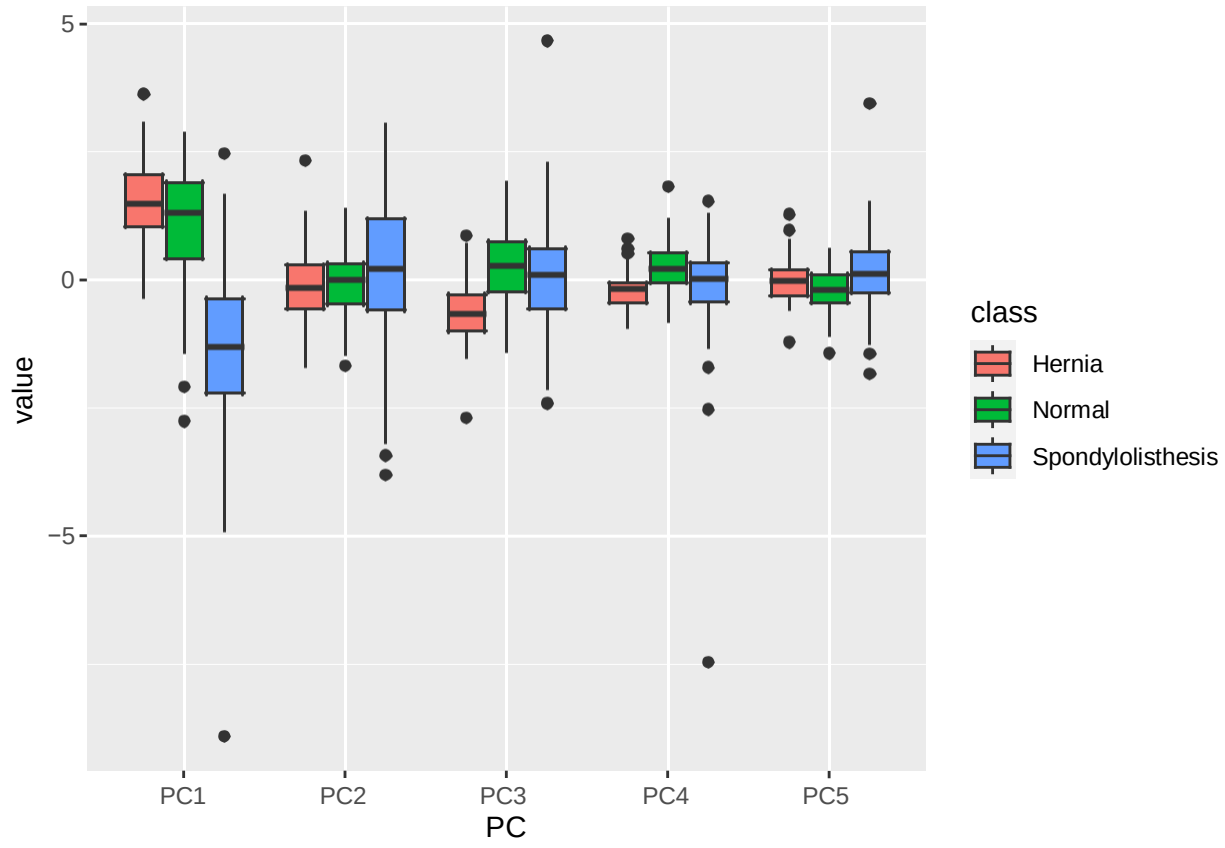
```
[1]    0.17829291 0.07626548 0.09330379
Creating new device
```

```
snapshotPCA3d(file="pca3d-plot.png")
```

We can assume from the above 2D and 3D plots that the variables degree_spondylolisthesis, pelvic incidence and lumbar_lardosis_angle which are correlated separate spondylolisthesis patients from normal and hernia patients, while sacral_sclope and pelvic_radius may separate normal from hernia patients.

Here's a boxplot of the first 5 PCs grouped by disease class

```
data.frame(class = dat$class, pca$x[,1:5]) %>%
   gather(key = "PC", value = "value", -class) %>%
   ggplot(aes(PC, value, fill = class)) +
   geom_boxplot()
```



In the first principle component, we observe a significant difference that there is no overlap in the interquartile ranges between spondylolisthesis class and Hernia and Normal classes. In the third principle component we see that between hernia and normal classes.

2.2.2 Model creation

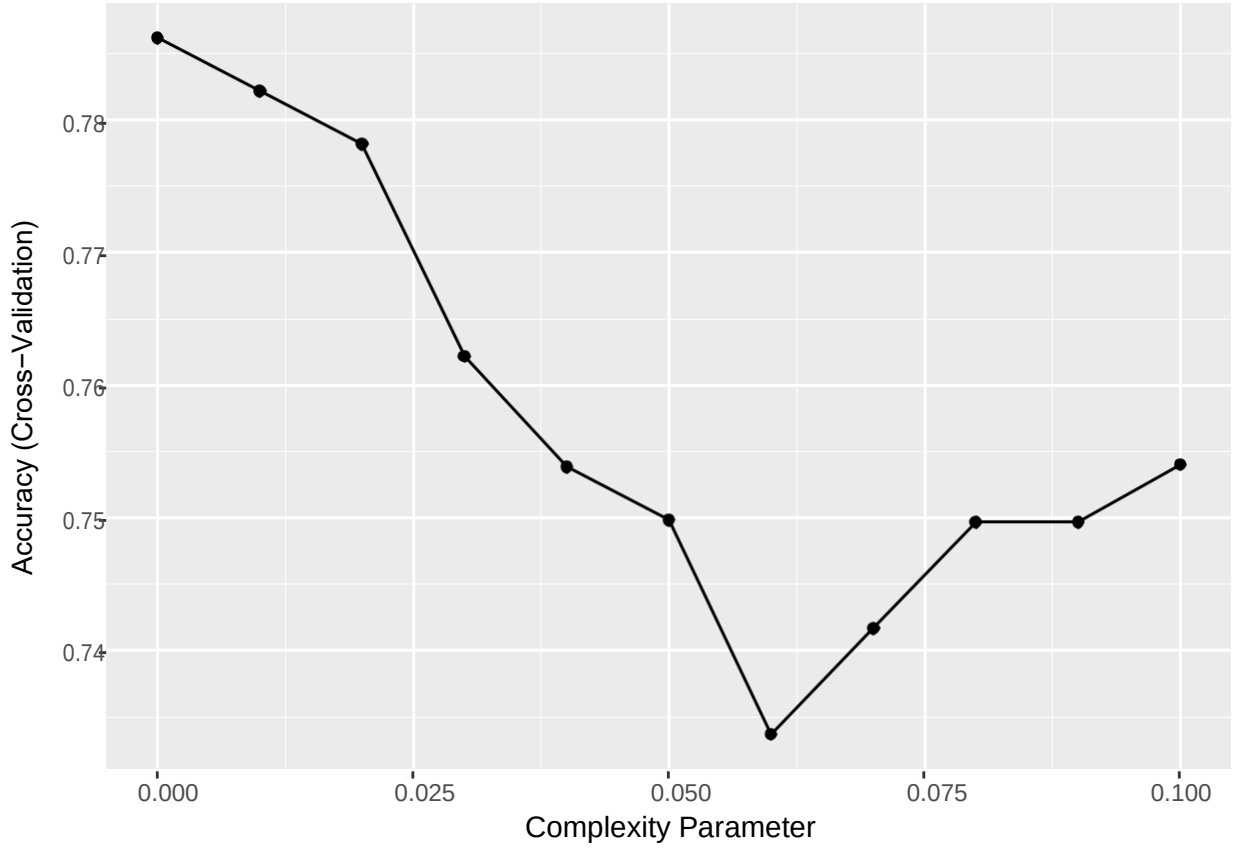
We are going to split the data into a training and a testing set to use when building some models. I split the modified dataset into Train (80%) and Test (20%), in order to predict disease class by building machine learning classification models.

```
set.seed(1815)
test_index <- createDataPartition(dat$class, times = 1, p = 0.2, list = FALSE)
test_x <- dat_scaled[test_index,]
test_y <- dat$class[test_index]
train_x <- dat_scaled[-test_index,]
train_y <- dat$class[-test_index]

control <- trainControl(method = "cv", number = 10, p = .9)
```

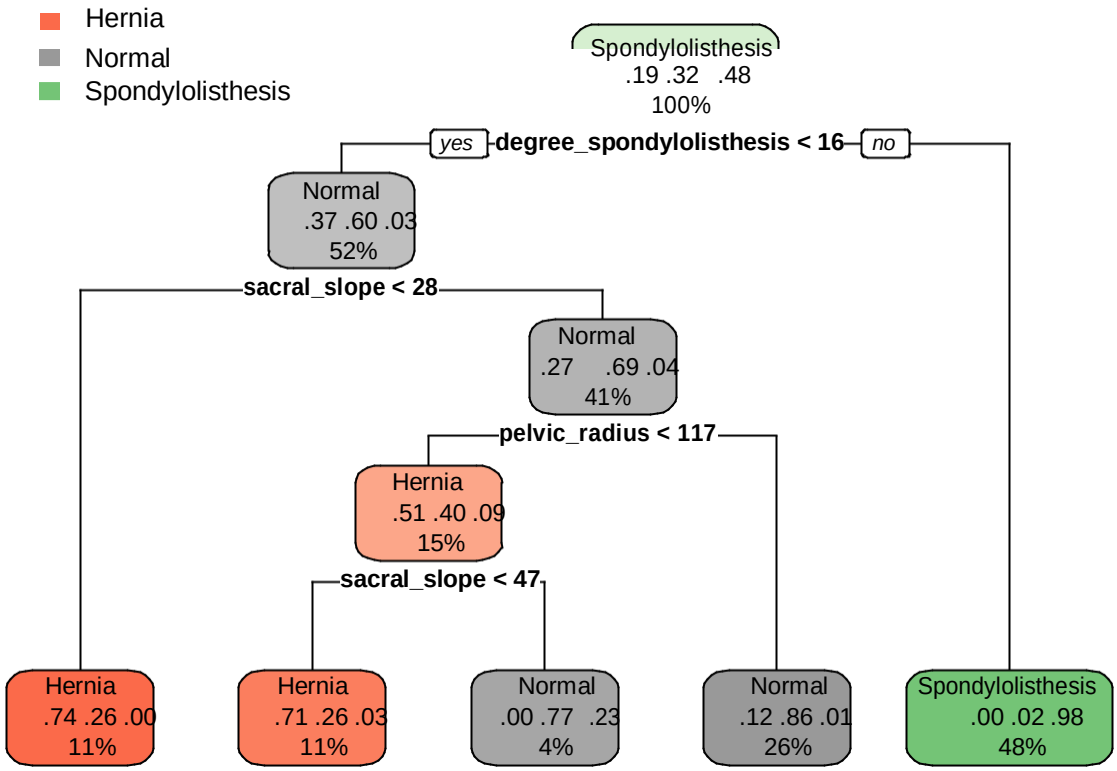
2.2.3 Recursive Partitioning and Regression Trees Model (rpart):

```
#Training the model
```

```
train_rpart <-      train(train_x, train_y, method = "rpart",
                          tuneGrid = data.frame(cp = seq(0, 0.1, 0.01)),
                          trControl = control)

#Finding the best cp

ggplot(train_rpart)
```




```
#Plotting rpart decision tree:

class.tree <- rpart( dat$class~., dat, control = rpart.control(cp = 0.03))
rpart.plot(class.tree)
```

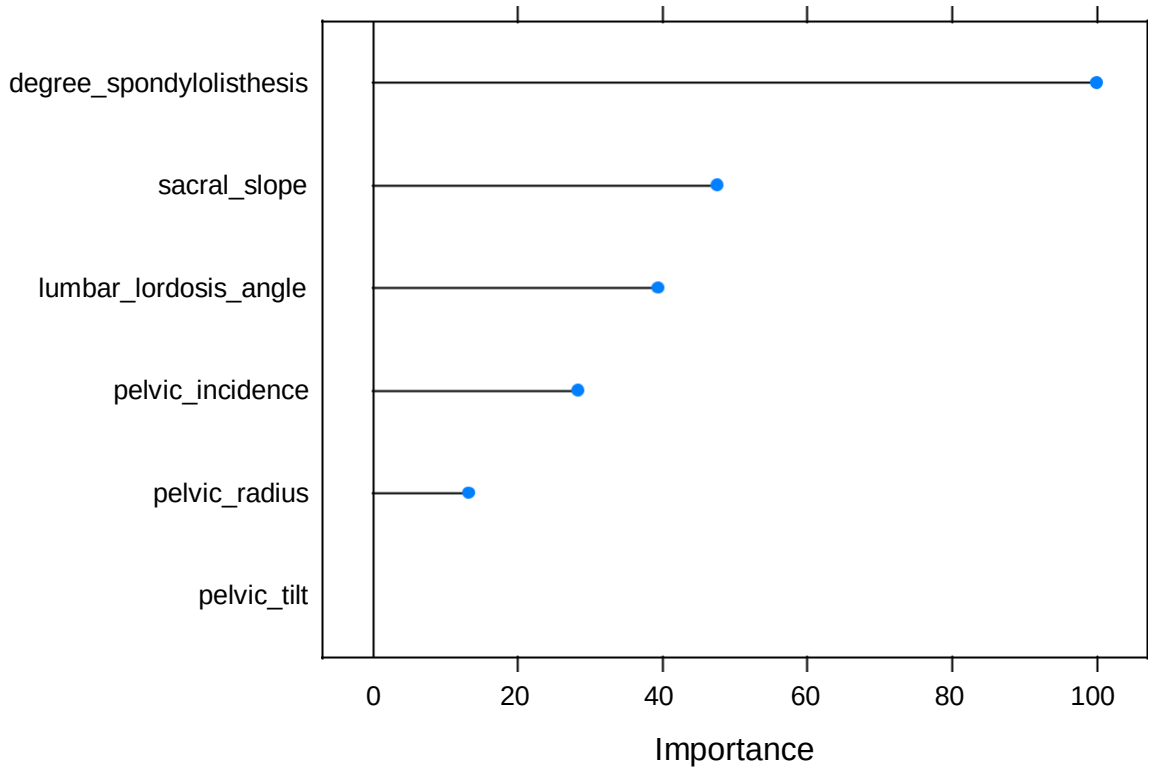


The above tree supports our previous observation that degree_spondylolisthesis separates spondylolisthesis class from hernia and normal classes while sacral_slope and pelvic_radius separate the normal from the hernia class. According to the tree, patients with degree_spondylolisthesis of more that 16 belong to the spondylolisthesis class while those with inferior spondylolisthesis at 16 are normal when sacral_slope and pelvic radius are superior than 28 and 117 resepectively. Let's move on to determine the accuracy of our model.

```
#Model prediction

rpart_preds <- predict(train_rpart,              test_x)

#Overall accuracy

mean(rpart_preds == test_y)
```

```
[1] 0.8709677
```

```
#Ranking the varaibles according to importance

plot(varImp(train_rpart))
```




```
#Confusion Matrix

cm_rpart <- confusionMatrix(rpart_preds, as.factor(test_y))
```

2.2.4 Random Forest Model (RF):

```
tuning <- data.frame(mtry = c(2, 20, 2))
train_rf <- train(train_x, train_y,
                  method = "rf",
                  tuneGrid = tuning,
                  importance = TRUE,
                  trControl = control)

train_rf$bestTune
```

```
  mtry
2   20
```

```
acc_rf <- rf_preds <- predict(train_rf, test_x)

mean(rf_preds == test_y)
```

```
[1] 0.8548387
```

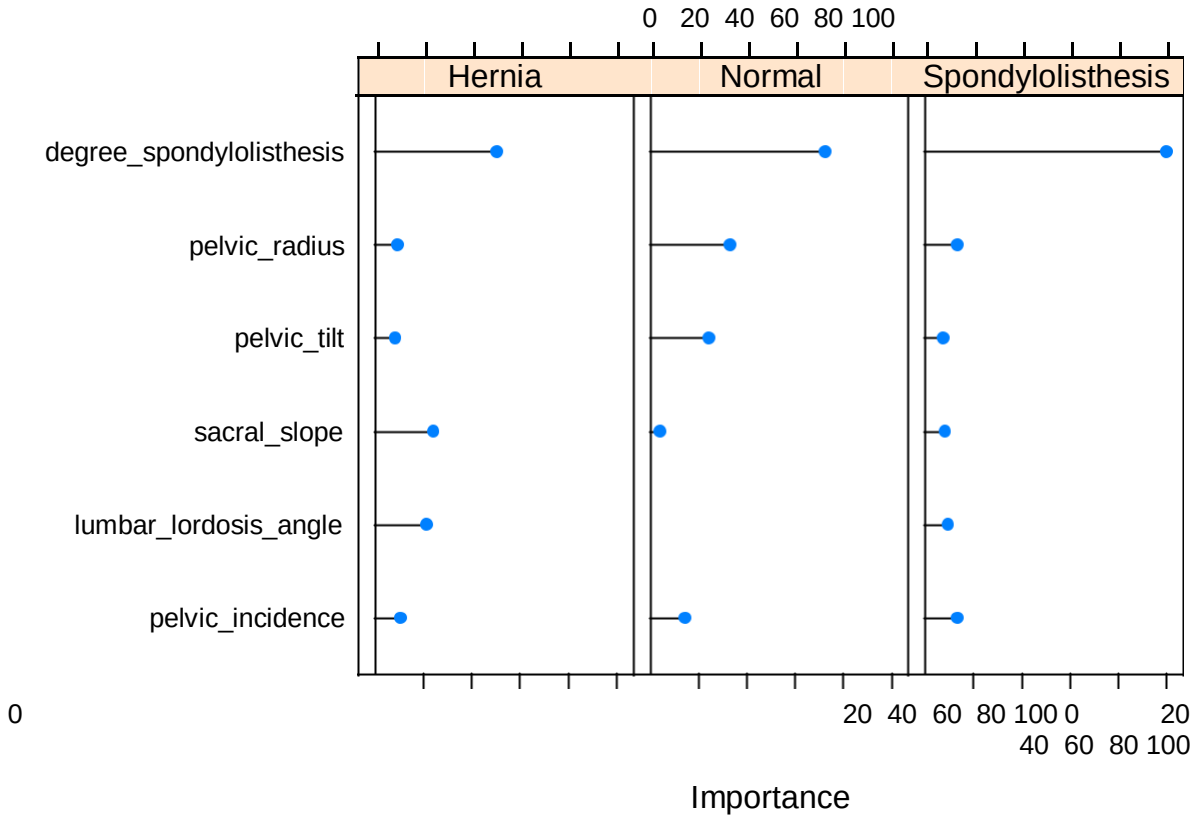
```
#Ranking the varaibles according to importance

varImp(train_rf)
```

rf variable importance

variables are sorted by maximum importance across the classes

	Hernia	Normal	Spondylolisthesis
degree_spondylolisthesis	50.514	72.588	100.000
pelvic_radius	9.231	33.252	13.475
pelvic_tilt	8.347	24.342	7.473
sacral_slope	24.153	4.235	8.205
lumbar_lordosis_angle	21.349	0.000	9.466
pelvic_incidence	10.611	14.543	13.523

```
plot(varImp(train_rf))
```



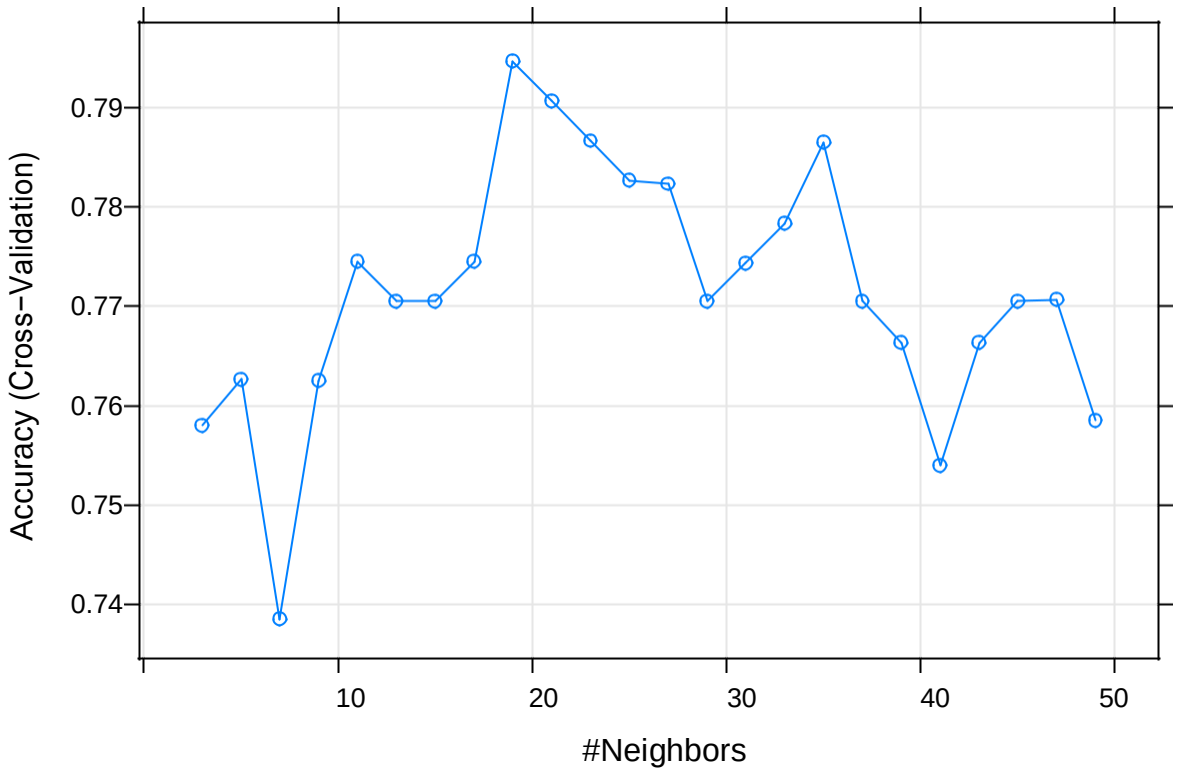

```
#Confusion Matrix

cm_rf <- confusionMatrix(rf_preds, as.factor(test_y))
```

2.2.5 K-Nearest Neighbors Model (KNN) Model:

```
tuning <- data.frame(k = seq(3, 50, 2))
train_knn <- train(train_x, train_y,
                                method = "knn",
                                tuneGrid = tuning,
                                trControl = control)
train_knn$bestTune
```

```
    k
9 19
```

```
#Finding the best k
plot(train_knn)
```




```
knn_preds <- predict(train_knn, test_x)

mean(knn_preds == test_y)
```

```
[1] 0.8225806
```

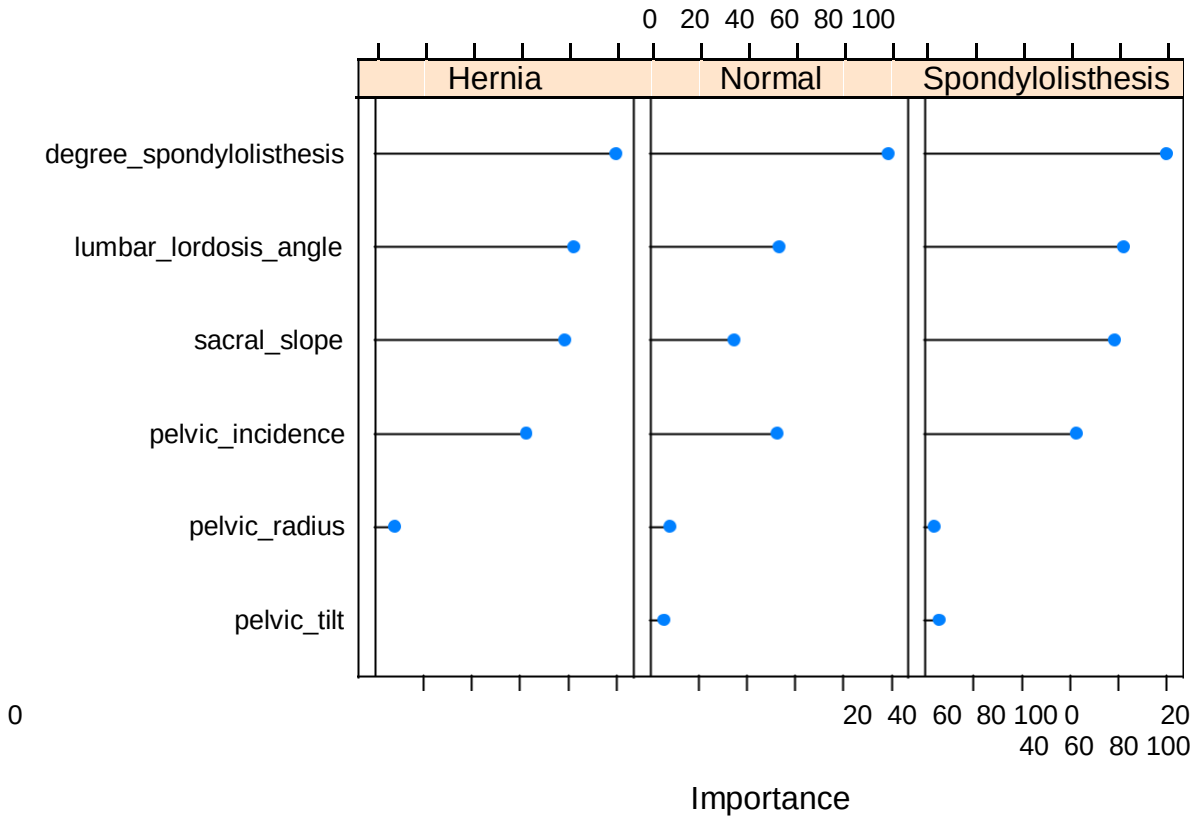
```
#Confusion Matrix

cm_knn <- confusionMatrix(knn_preds, as.factor(test_y))
```

2.2.6 Linear Discriminant Analysis (LDA):

```
train_lda <- train(train_x, train_y,
                   method = "lda", trControl = control)

lda_preds <- predict(train_lda,          test_x)

mean(lda_preds == test_y)
```

```
[1] 0.8225806
```

```
#Ranking the varaibles according to importance
plot(varImp(train_lda))
```




```
#Confusion Matrix

cm_lda <- confusionMatrix(lda_preds, factor(test_y))
```

2.2.7 Ensemble

Let's create an ensemble using the predictions from the four used models.

```
models <- c("lda", "knn", "rf", "rpart")

fits <- lapply(models, function(model){
  print(model)
  train(train_x, train_y, method = model)
})
```

```
[1]  "lda"
[1]  "knn"
[1]  "rf"
[1]  "rpart"
```

```
names(fits) <- models

ensemble_preds <- sapply(fits, function(object)
  predict(object, newdata = test_x))

accuracy <- colMeans(ensemble_preds == test_y)
accuracy
```

```
      lda       knn        rf     rpart
0.8225806 0.8548387 0.9032258 0.7419355
```

```
acc_ensemble <- mean(accuracy)
```

Chapter 3

Results

let's compare and evaluate the results obtained from the trained models. We will start by investigating the confusion matrices obtained from the different algorithms

```
acc_table <- data.frame(KNN = cm_knn$overall['Accuracy'], RF = cm_knn$overall['Accuracy'], LDA = cm_lda
  gather(key= model, value = overall_accuracy)
acc_table
```

```
        model overall_accuracy
1         KNN        0.8225806
2          RF        0.8225806
3         LDA        0.8225806
4       rpart        0.8709677
5  Ensemble        0.8306452
```

```
confusionmatrix.list <- list(
  LDA=cm_lda,
  rpart=cm_rpart,
  Random_forest=cm_rf,
  KNN=cm_rf
)
lapply(confusionmatrix.list, function(x) x$byClass)
```

```
$LDA
                                 Sensitivity Specificity Pos Pred Value Neg Pred Value
Class: Hernia                      0.6666667   0.9400000      0.7272727      0.9215686
Class: Normal                      0.7500000   0.9047619      0.7894737      0.8837209
Class: Spondylolisthesis           0.9333333   0.8750000      0.8750000      0.9333333
                                  Precision    Recall        F1  Prevalence
Class: Hernia                     0.7272727 0.6666667 0.6956522  0.1935484
Class: Normal                     0.7894737 0.7500000 0.7692308  0.3225806
Class: Spondylolisthesis          0.8750000 0.9333333 0.9032258  0.4838710
                                  Detection Rate Detection Prevalence Balanced Accuracy
Class: Hernia                          0.1290323            0.1774194         0.8033333
Class: Normal                          0.2419355            0.3064516         0.8273810
Class: Spondylolisthesis               0.4516129            0.5161290         0.9041667

$rpart
```

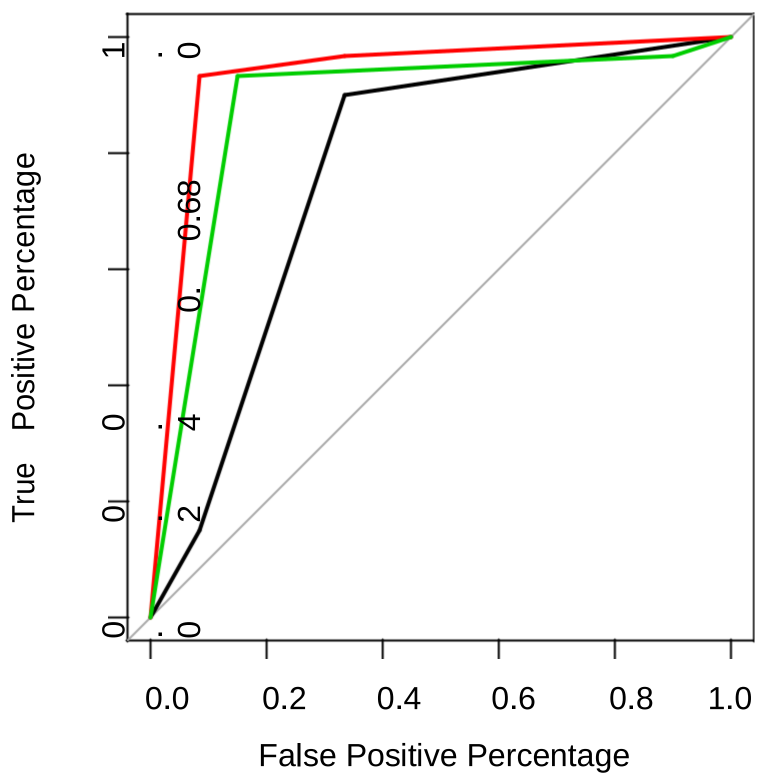
```
                     Sensitivity Specificity Pos Pred Value Neg Pred Value
Class: Hernia          0.6666667   0.9600000            0.80       0.9230769
Class: Normal          0.9000000   0.8571429            0.75       0.9473684
Class: Spondylolisthesis 0.9333333   1.0000000            1.00       0.9411765
                     Precision    Recall        F1  Prevalence
Class: Hernia            0.80 0.6666667 0.7272727  0.1935484
Class: Normal            0.75 0.9000000 0.8181818  0.3225806
Class: Spondylolisthesis 1.00 0.9333333 0.9655172  0.4838710
                     Detection Rate Detection Prevalence Balanced Accuracy
Class: Hernia             0.1290323            0.1612903         0.8133333
Class: Normal             0.2903226            0.3870968         0.8785714
Class: Spondylolisthesis  0.4516129            0.4516129         0.9666667

$Random_forest
                     Sensitivity Specificity Pos Pred Value Neg Pred Value
Class: Hernia          0.5833333   0.9600000       0.7777778       0.9056604
Class: Normal          0.9000000   0.8333333       0.7200000       0.9459459
Class: Spondylolisthesis 0.9333333   1.0000000       1.0000000       0.9411765
                     Precision    Recall        F1  Prevalence
Class: Hernia        0.7777778 0.5833333 0.6666667  0.1935484
Class: Normal        0.7200000 0.9000000 0.8000000  0.3225806
Class: Spondylolisthesis 1.0000000 0.9333333 0.9655172  0.4838710
                     Detection Rate Detection Prevalence Balanced Accuracy
Class: Hernia             0.1129032            0.1451613         0.7716667
Class: Normal             0.2903226            0.4032258         0.8666667
Class: Spondylolisthesis  0.4516129            0.4516129         0.9666667

$KNN
                     Sensitivity Specificity Pos Pred Value Neg Pred Value
Class: Hernia          0.5833333   0.9600000       0.7777778       0.9056604
Class: Normal          0.9000000   0.8333333       0.7200000       0.9459459
Class: Spondylolisthesis 0.9333333   1.0000000       1.0000000       0.9411765
                     Precision    Recall        F1  Prevalence
Class: Hernia        0.7777778 0.5833333 0.6666667  0.1935484
Class: Normal        0.7200000 0.9000000 0.8000000  0.3225806
Class: Spondylolisthesis 1.0000000 0.9333333 0.9655172  0.4838710
                     Detection Rate Detection Prevalence Balanced Accuracy
Class: Hernia             0.1129032            0.1451613         0.7716667
Class: Normal             0.2903226            0.4032258         0.8666667
Class: Spondylolisthesis  0.4516129            0.4516129         0.9666667
```

As we can see from the obtained results, the rpart model achieved the highest overall accuracy and the highest sensitivity (true positive rate) and specificity (true negative rate) for all three classes.

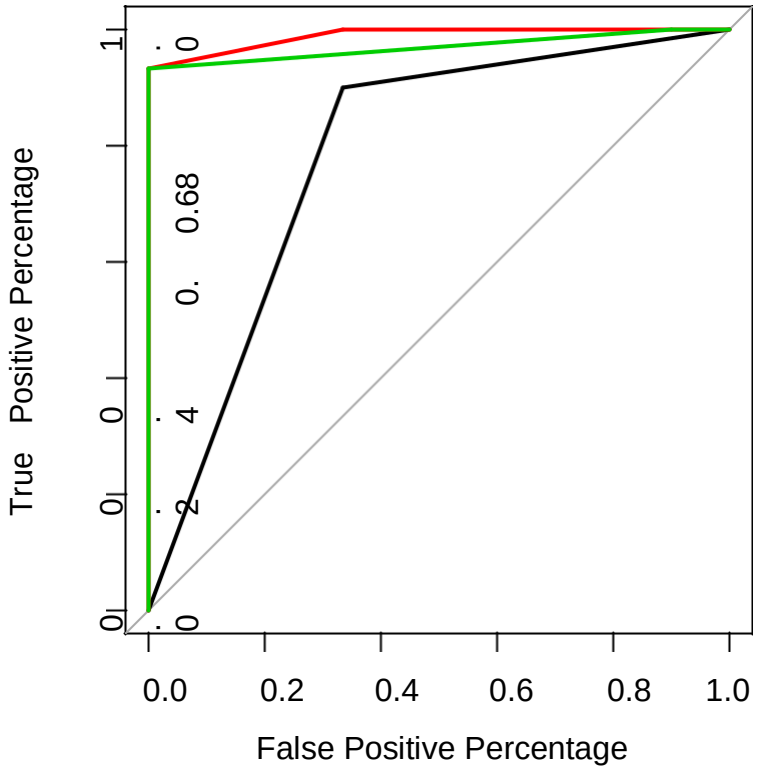
Now let's plot a multiclass ROC curve (Receiver Operating characteristic Curve) and compute the area under the ROC curve for the models

```
#Converting the y_hats to numeric values so the multiclass.roc function accepts the arguments

test_y.n <- as.numeric(test_y)
lda_preds.n <- as.numeric(lda_preds)
knn_preds.n <- as.numeric(knn_preds)
rf_preds.n <- as.numeric(rf_preds)
rpart_preds.n <- as.numeric(rpart_preds)
```

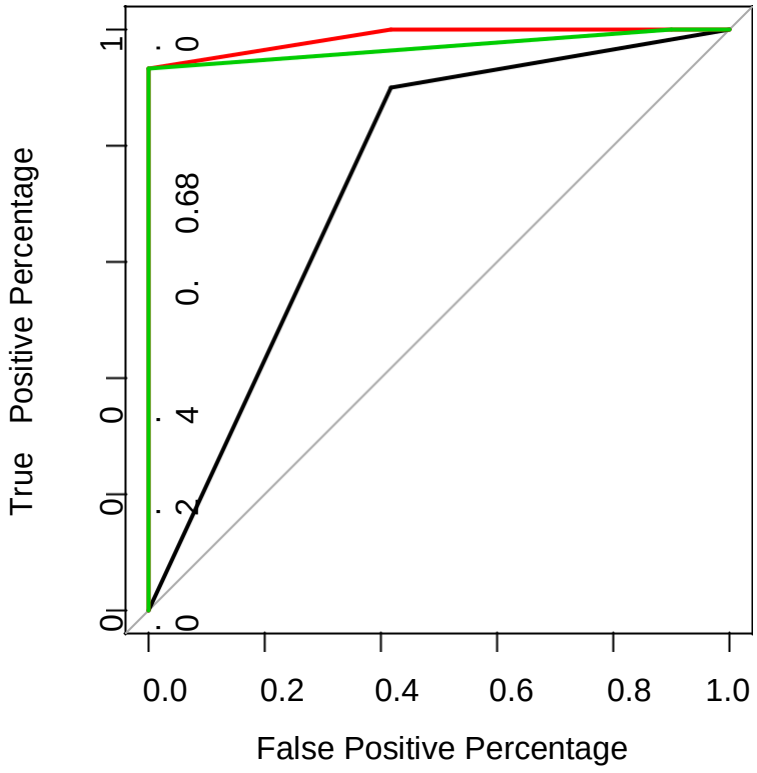
```
#ROC for LDA model:
par(pty = "s")
roc.multi.lda <- multiclass.roc(test_y.n, lda_preds.n)
rs1 <- roc.multi.lda[['rocs']]
plot.roc(rs1[[1]], legacy.axes = TRUE, percent = TRUE, xlab = "False Positive Percentage", ylab = "True
sapply(2:length(rs1),function(i) lines.roc(rs1[[i]],col=i))
```



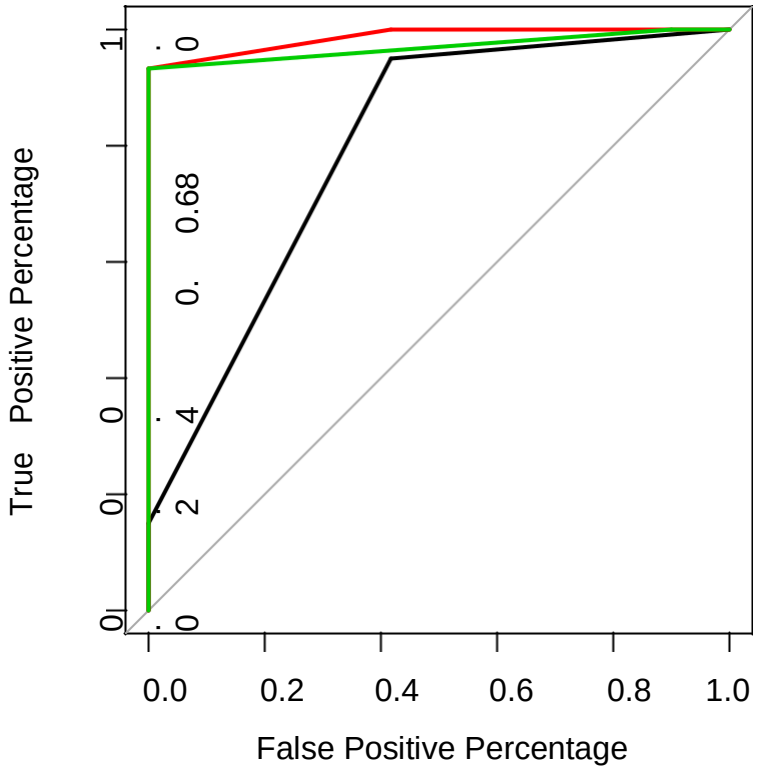

```
#ROC for rpart model:

roc.multi.rpart <- multiclass.roc(test_y.n, rpart_preds.n)
rs2 <- roc.multi.rpart[['rocs']]
plot.roc(rs2[[1]], legacy.axes = TRUE, percent = TRUE, xlab = "False Positive Percentage", ylab = "True
sapply(2:length(rs2),function(i) lines.roc(rs2[[i]],col=i))
```




```
#ROC for rf model:


roc.multi.rf <- multiclass.roc(test_y.n, rf_preds.n)
rs3 <- roc.multi.rf[['rocs']]
plot.roc(rs3[[1]], legacy.axes = TRUE, percent = TRUE, xlab = "False Positive Percentage", ylab = "True
sapply(2:length(rs3),function(i) lines.roc(rs3[[i]],col=i))
```




```
#ROC for knn model:


roc.multi.knn <- multiclass.roc(test_y.n, knn_preds.n)
rs4 <- roc.multi.knn[['rocs']]
plot.roc(rs4[[1]], legacy.axes = TRUE, percent = TRUE, xlab = "False Positive Percentage", ylab = "True
sapply(2:length(rs4),function(i) lines.roc(rs3[[i]],col=i))
```




```
#AUC for LDA model
roc.multi.lda$auc
```

Multi-class area under the curve: 0.8612

```
#AUC for rpart model
roc.multi.rpart$auc
```

Multi-class area under the curve: 0.9141

```
#AUC for RF model
roc.multi.rf$auc
```

Multi-class area under the curve: 0.8993

```
#AUC for KNN model
roc.multi.knn$auc
```

Multi-class area under the curve: 0.8822

The rpart model achieved the highest AUC value of 0.9141

Chapter 4

Conclusion

In our dataset, six biomechanical features have been used as predictors for three classes of patients: spondy-lolisthesis, hernia and normal patients. Through a hierarchical clustering heatmap, principle component analysis and data visualization of our dataset, we observed clustering of the data based on the biomechanical features.

We assumed that degree_spondylolisthesis may play a major role in separating_spondylolisthesis patients from normal and hernia patients, along with pelvic_incidence and lumbar_lardosis_angle variables which were moderately correlated with degree_spondylolisthesis, while pelvic redius and sacral slope played a role in separating normal from hernia patients.

Four machine learning models have been implemented in order to categorize patients based on their ortho-pedic condition. All four models agreed that the variable of most importance was degree_spondylolisthesis. This and the rpart decision tree investigation agree with our previous assumption. Among all the four algo-rithms, the Recursive Partitioning and Regression Trees Model (rpart) has provided the highest accuracy of 0.871, sensitivity, specificity and F1 scores across all classes for our dataset, also achieving the highest AUC value of 0.9141.

One drawback of our modeling results is the relatively low sensitivity of the hernia class (0.667) which has dropped the overall accuracy of our models. Perhaps this work can be extended to other machine learning algorithms such as deep learning and neural networks.

Chapter 5

References

1. Alexander, R.M., 2005. Mechanics of animal movement. Current biology, 15(16), pp.R616-R619.
2. Robert, E., Windsor (2006)." Frequency of asymptomatic cervical disc protrusions". Cervical Disc Injuries. eMedicine. Retrieved 2008-02-27.
3. Herman, M.J., Pizzutillo, P.D. and Cavalier, R., 2003. Spondylolysis and spondylolisthesis in the child and adolescent athlete. Orthopedic Clinics, 34(3), pp.461-467.
4. Foreman, P., Griessenauer, C.J., Watanabe, K., Conklin, M., Shoja, M.M., Rozzelle, C.J., Loukas, M. and Tubbs, R.S., 2013. L5 spondylolysis/spondylolisthesis: a comprehensive review with an anatomic focus. Child's Nervous System, 29(2), pp.209-216.