Biomechanical features of orthopedic patients

Rajeev Kumar Rajesh

23/08/2020

Introduction

This project 'Biomechanical features of orthopedic patients'.In this project each patient is represented in the data set by six biomechanical attributes derived from the shape and orientation of the pelvis and lumbar spine (each one is a column).

i)Pelvic incidence ii)Pelvic tilt iii)lumbar lordosis angle iv)Sacral slope v)Pelvic radius vi)Grade of spondylolisthesis

We use these biomechanical features to classify patients according to their labels.And also find the correlation between above variable.

```
# Read the csv File
dat <- read.csv("column_2C_weka.csv", stringsAsFactors = F)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## corrplot 0.84 loaded

## Loading required package: lattice

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

##
## Attaching package: 'psych'

## The following object is masked from 'package:randomForest':
##
##     outlier

## The following objects are masked from 'package:scales':
##
##     alpha, rescale

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
head(dat)
```

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

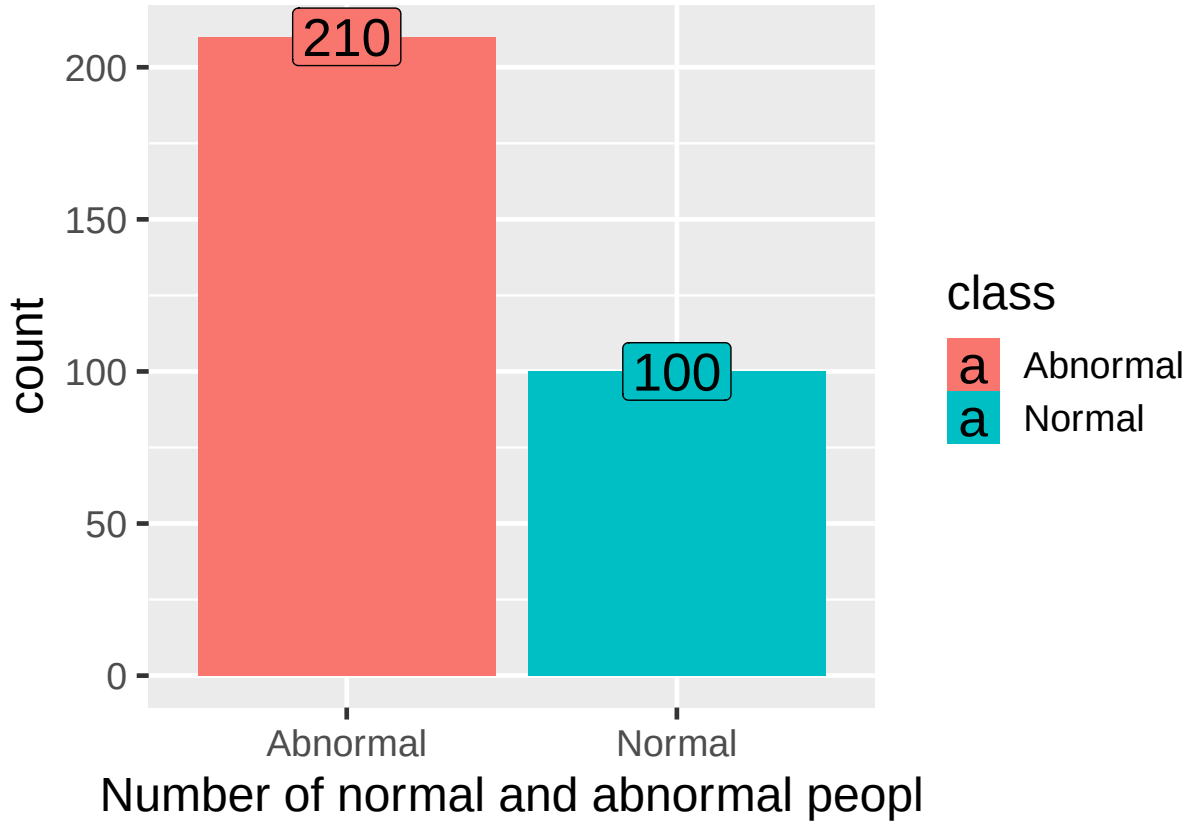
```r
dim(dat)
```

```
## [1] 310   7
```

```r
str(dat)
```

```
## 'data.frame':    310 obs. of  7 variables:
##  $ pelvic_incidence        : num  63 39.1 68.8 69.3 49.7 ...
##  $ pelvic_tilt.numeric     : 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                   : chr  "Abnormal" "Abnormal" "Abnormal" "Abnormal" ...
```

```r
#Total Normal and abnormal people
ggplot(dat,aes(x=class,fill=class))+geom_bar(stat = 'count')+labs(x = 'Number of normal and abnormal pe
  geom_label(stat='count',aes(label=..count..), size=7) +theme_grey(base_size = 18)
```




```
table(dat$class)
```

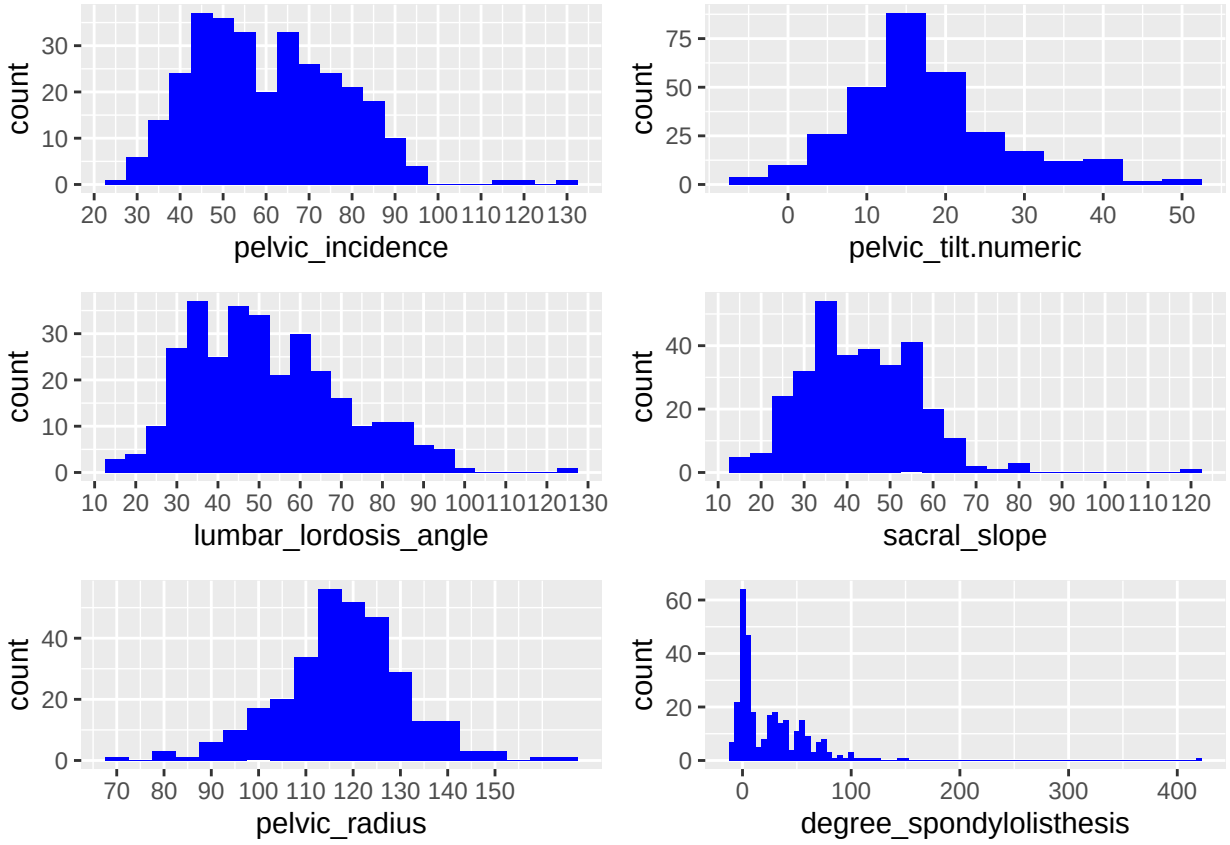
```
##
## Abnormal   Normal
##      210      100
```

Analysis of histogram Of all the explanatory variable

```
#Histogram explain all the detail of all six variable

H1<-ggplot(dat,aes(x=pelvic_incidence ))+
  geom_histogram(binwidth = 5, fill='blue')+ theme_grey() +
  scale_x_continuous(breaks= seq(0, 150, by=10))
H2<-ggplot(dat,aes(x=pelvic_tilt.numeric ))+
  geom_histogram(binwidth = 5, fill='blue') + theme_grey() +
  scale_x_continuous(breaks= seq(0, 150, by=10))
H3<-ggplot(dat,aes(x=lumbar_lordosis_angle ))+
  geom_histogram(binwidth = 5, fill='blue') + theme_grey() +
  scale_x_continuous(breaks= seq(0, 150, by=10))
H4<-ggplot(dat,aes(x=sacral_slope ))+
  geom_histogram(binwidth = 5, fill='blue') + theme_grey() +
  scale_x_continuous(breaks= seq(0, 150, by=10))
H5<-ggplot(dat,aes(x=pelvic_radius ))+
```

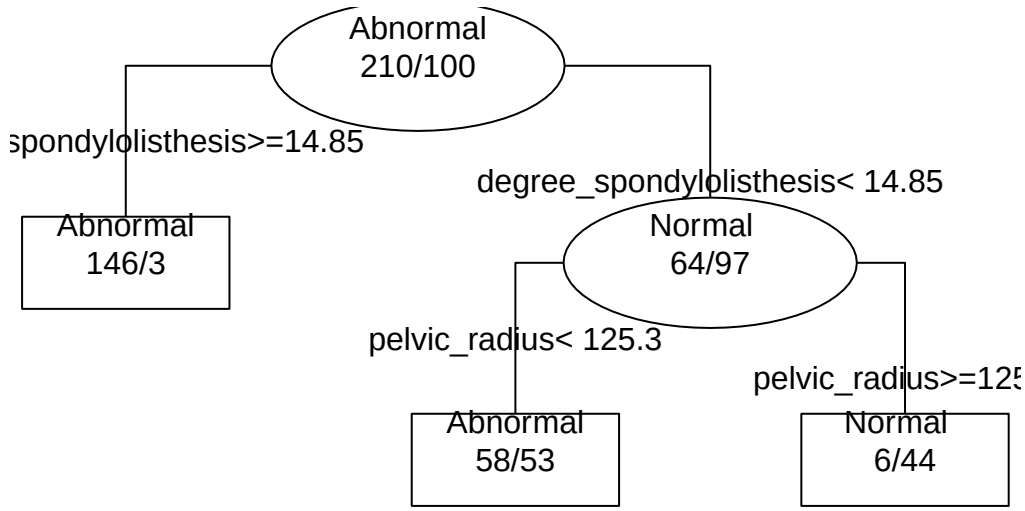
```
  geom_histogram(binwidth = 5, fill='blue') + theme_grey() +
  scale_x_continuous(breaks= seq(0, 150, by=10))
H6<-ggplot(dat,aes(x=degree_spondylolisthesis ))+
  geom_histogram(binwidth = 5, fill='blue') + theme_grey()
grid.arrange(H1,H2,H3,H4,H5,H6)
```




```
set.seed(1, sample.kind = 'Rounding')
```

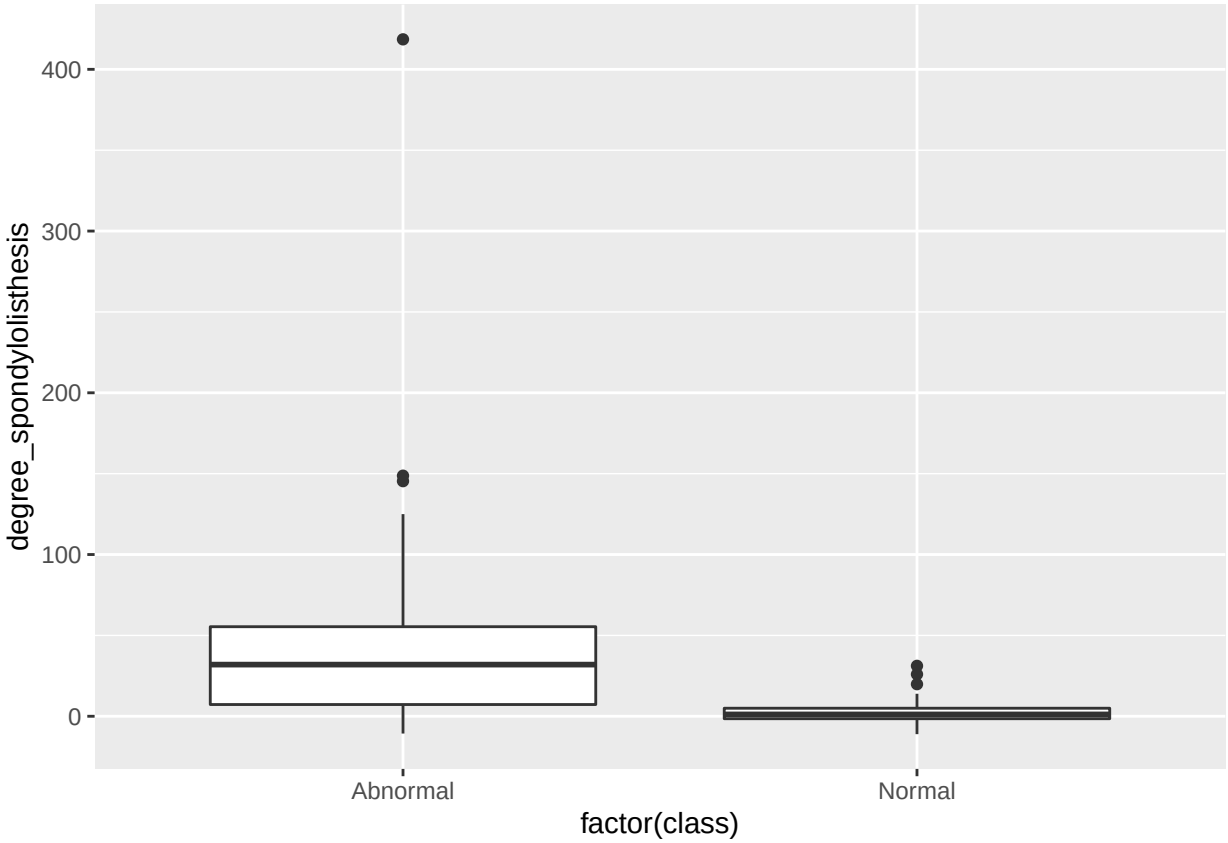
```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
library(rpart)
arbre <- rpart(dat$class~., method="class", minsplit=150, xval=10000, data=dat)
plot(arbre, uniform=TRUE, margin=0.1, main="Decision Tree")
text(arbre, fancy=TRUE, use.n=TRUE, pretty=0, all=TRUE)
```



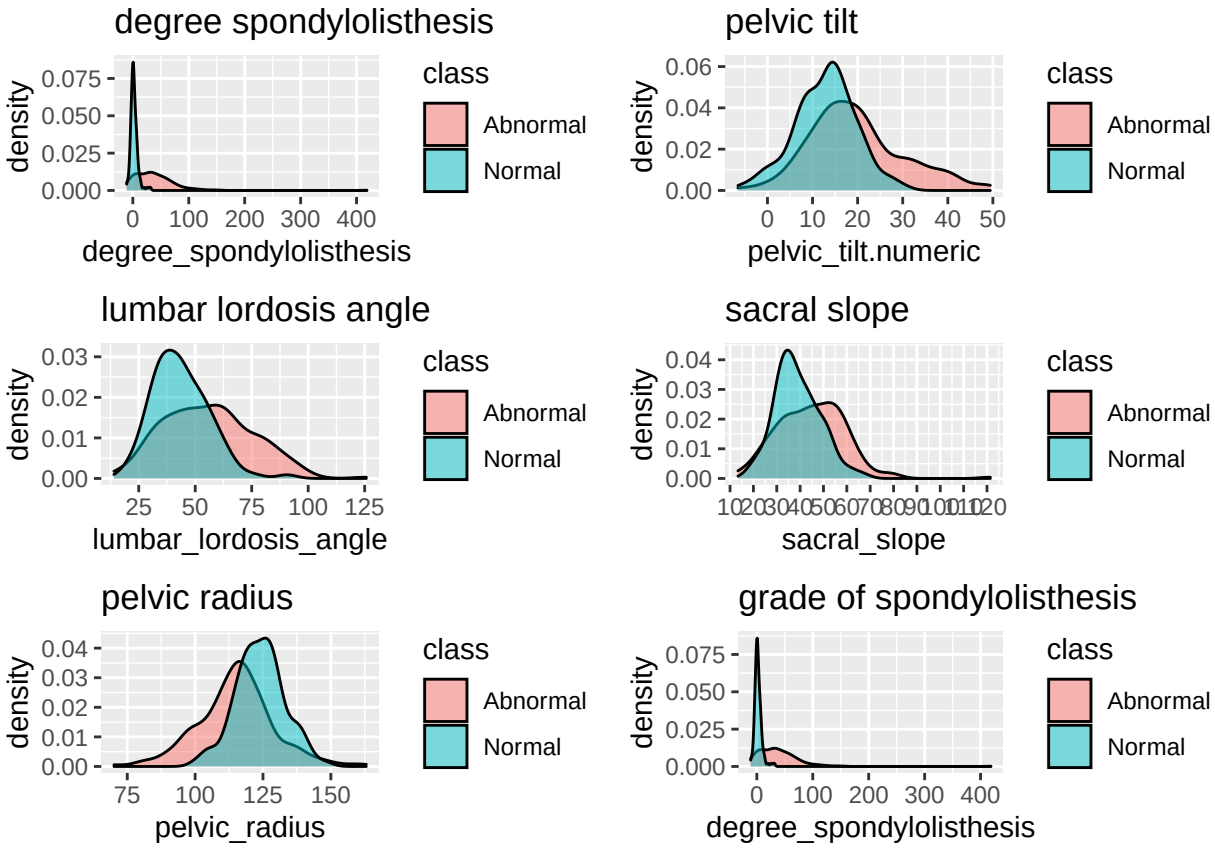
We find that those with inferior spondylolisthesis at 14.85 are those of patients who are fortunate enough to have normal class,but it should be noted that it is normal that when pelvic_radius is supperieur or equal to 125.

```
#We can also view it on the boxplot.
ggplot(dat,aes(x=factor(class),y=degree_spondylolisthesis))+geom_boxplot()
```

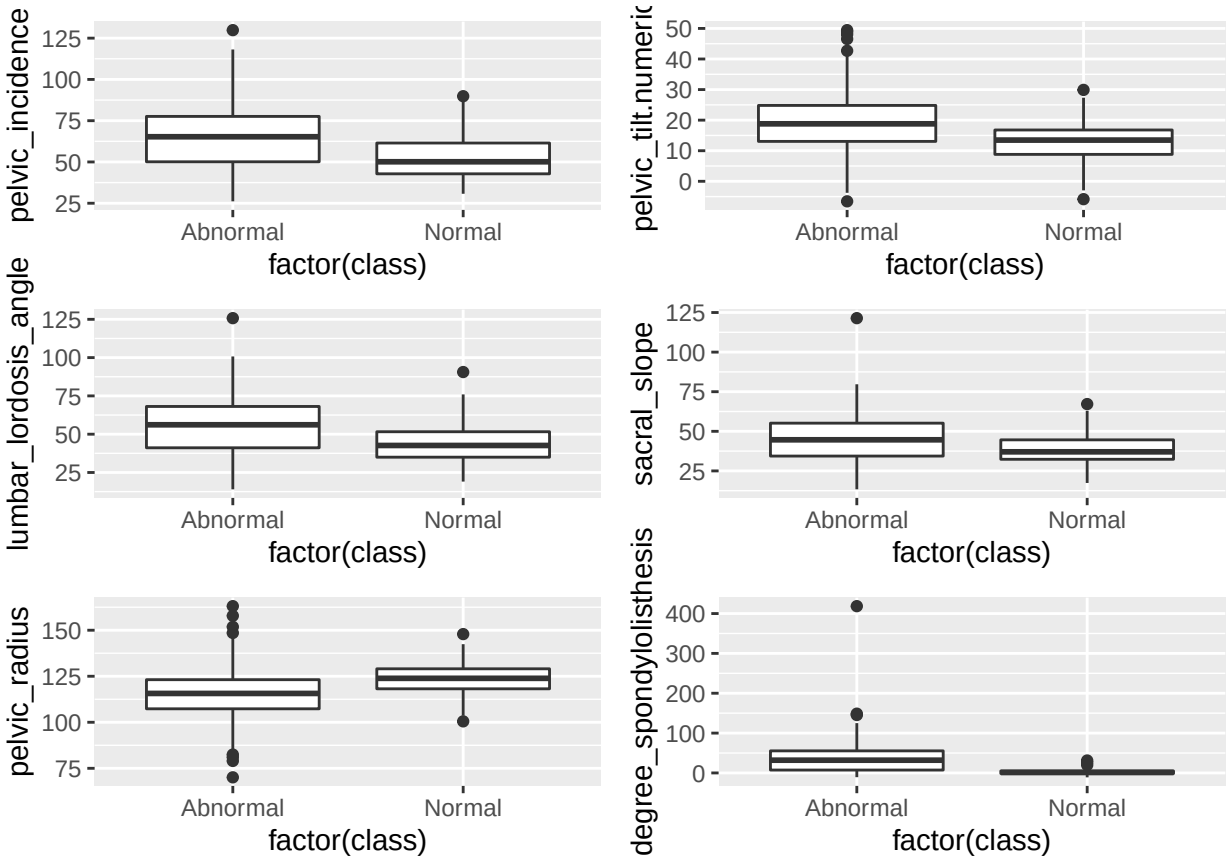


We can also visualize the density of the set of variables of importance that explains the class of patients.

```
D1<-ggplot(dat,aes(x=degree_spondylolisthesis,fill=class))+
  geom_density(alpha=0.5, aes(fill=factor(class))) +
  labs(title="degree spondylolisthesis")  + theme_grey()
D2<-ggplot(dat,aes(x=pelvic_tilt.numeric,fill=class))+
  geom_density(alpha=0.5, aes(fill=factor(class))) +
  labs(title="pelvic tilt")  + theme_grey()
D3<-ggplot(dat,aes(x=lumbar_lordosis_angle,fill=class))+
  geom_density(alpha=0.5, aes(fill=factor(class))) +
  labs(title="lumbar lordosis angle") + theme_grey()
D4<-ggplot(dat,aes(x=sacral_slope,fill=class))+
  geom_density(alpha=0.5, aes(fill=factor(class))) + labs(title="sacral slope") +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) + theme_grey()
D5<-ggplot(dat,aes(x=pelvic_radius,fill=class))+
  geom_density(alpha=0.5, aes(fill=factor(class))) +
  labs(title="pelvic radius") + theme_grey()
D6<-ggplot(dat,aes(x=degree_spondylolisthesis,fill=class))+
  geom_density(alpha=0.5, aes(fill=factor(class))) +
  labs(title="grade of spondylolisthesis") + theme_grey()
grid.arrange(D1,D2,D3,D4,D5,D6)
```



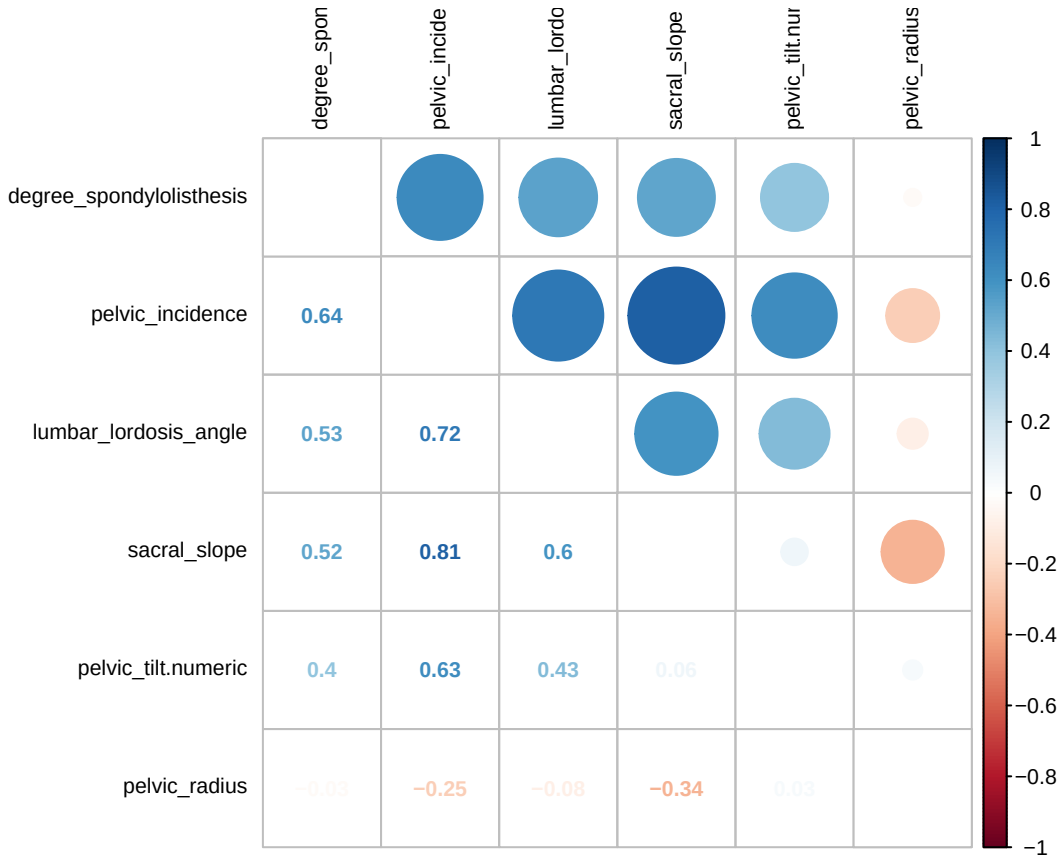

```
#Below is the boxplot of all variables
B1<-ggplot(dat,aes(x=factor(class),y=pelvic_incidence ))+geom_boxplot()
B2<-ggplot(dat,aes(x=factor(class),y=pelvic_tilt.numeric))+geom_boxplot()
B3<-ggplot(dat,aes(x=factor(class),y=lumbar_lordosis_angle))+geom_boxplot()
B4<-ggplot(dat,aes(x=factor(class),y=sacral_slope))+geom_boxplot()
B5<-ggplot(dat,aes(x=factor(class),y=pelvic_radius))+geom_boxplot()
B6<-ggplot(dat,aes(x=factor(class),y=degree_spondylolisthesis))+geom_boxplot()
grid.arrange(B1,B2,B3,B4,B5,B6)
```



Correlation between variables

For this part we are interested in the correlation of numeric variables to better understand their evolution between, we will take degree spondylolisthesis as target. Altogether, there all numeric variables with a correlation greater than zero with degree spondylolisthesis .

```
library(corrplot)
numericVars <- which(sapply(dat, is.numeric)) #index vector numeric variables
dat_numVar <- dat[, numericVars]
cor_numVar <- cor(dat_numVar, use="pairwise.complete.obs") #correlations of all numeric variables
#sort on decreasing correlations with degree_spondylolisthesis
cor_sorted <- as.matrix(sort(cor_numVar[,'degree_spondylolisthesis'], decreasing = TRUE))
#select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt", tl.cex = 0.7,cl.cex = .7, number.cex=.7)
```



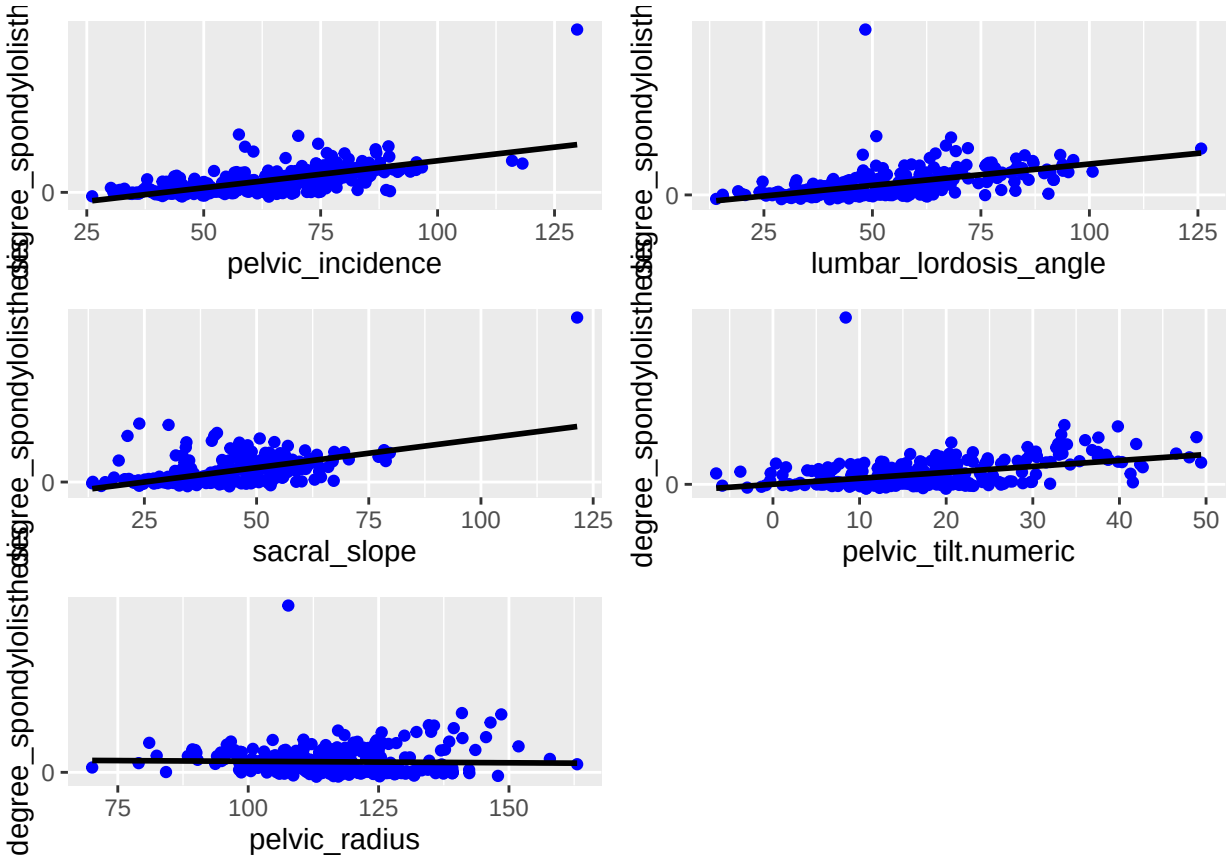
We notice that,there are 3 numeric variables with a correlation of at least 0.5 with degree_spondylolisthesis,and there is no correlation between pelvic_radius and degree_spondylolisthesis.It also becomes clear the multicollinearity is an issue. For example: the correlation between sacral_slope and pelvic_incidence is very high (0.81), and both have similar (high) correlations with degree_spondylolisthesis. Now let us visualize the correlation of these variables with respect to our target before visualizing that of sacral_slope and pelvic_incidence.

Now we analyze the relationship between degree spondylolisthesis And rest of the other Variables

```
library(gridExtra)
s1<-ggplot(data=dat, aes(x=pelvic_incidence, y=degree_spondylolisthesis))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000))
s2<-ggplot(data=dat, aes(x=lumbar_lordosis_angle, y=degree_spondylolisthesis))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000))
s3<-ggplot(data=dat, aes(x=sacral_slope, y=degree_spondylolisthesis))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000))
s4<-ggplot(data=dat, aes(x=pelvic_tilt.numeric, y=degree_spondylolisthesis))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000))
```

```
s5<-ggplot(data=dat, aes(x=pelvic_radius, y=degree_spondylolisthesis))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000))
grid.arrange(s1,s2,s3,s4,s5)
```

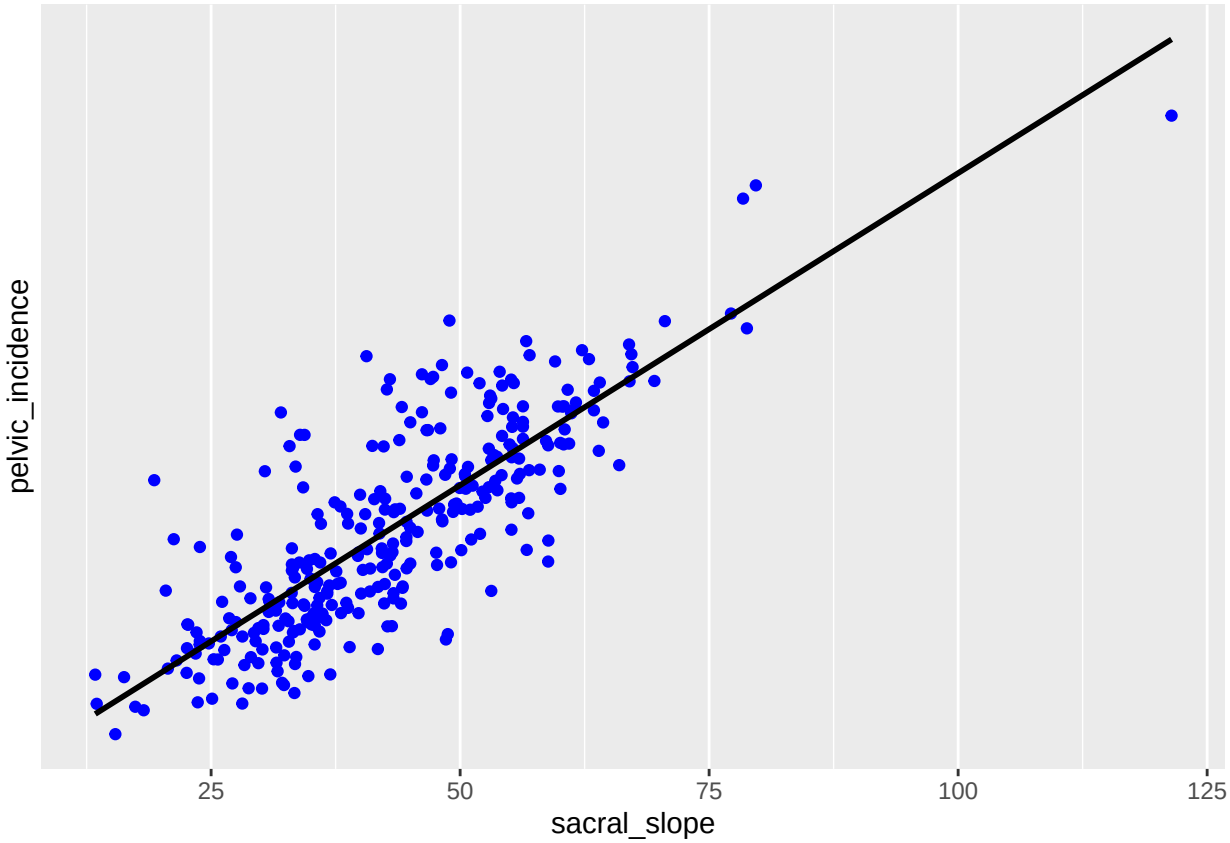
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Visualize us sacral_slope and pelvic_incidence

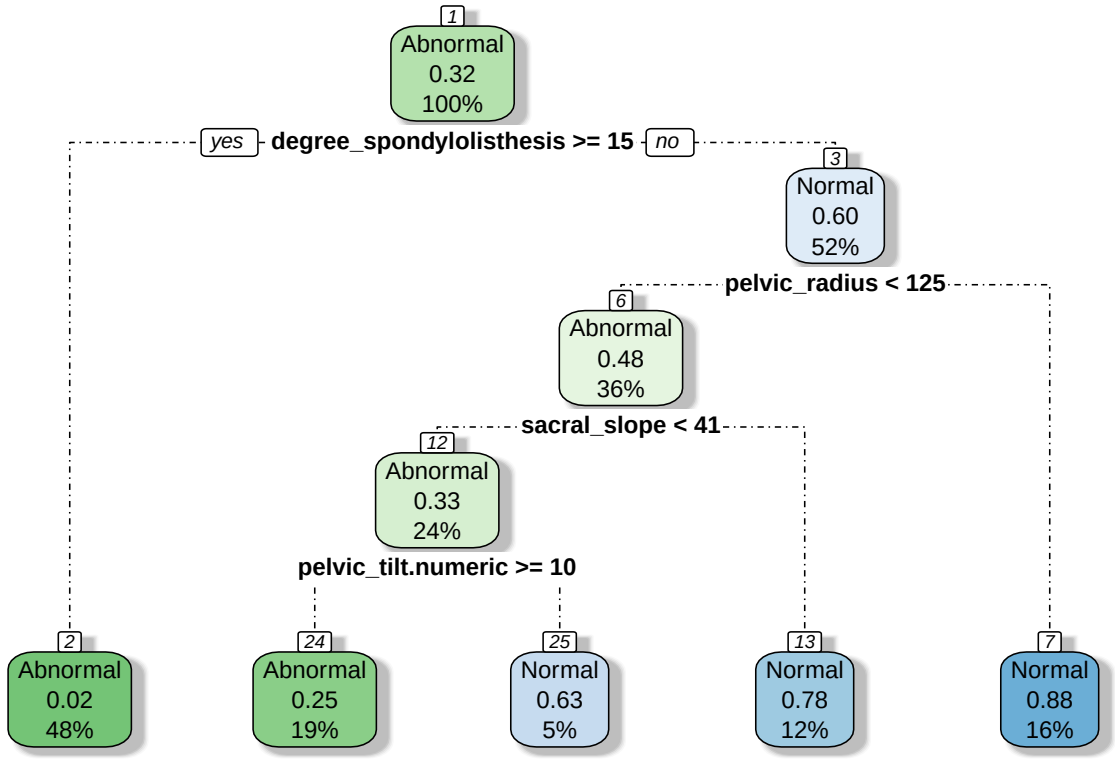
```
ggplot(data=dat, aes(x=sacral_slope, y=pelvic_incidence))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Decision tree Analysis

```
library(rpart.plot)
class.tree <- rpart(dat$class~.,data = dat,control = rpart.control(cp = 0.01))
rpart.plot(class.tree,
           box.palette="GnBu",
           branch.lty=10, shadow.col="gray", nn=TRUE)
```



Predictions (RandomForest vs Support Vector Machine (SVM) model vs Gradient Boosting Machine (GBM) model)

RandomForest

```
caret_matrix <- train(x=dat[,1:6], y=dat[,7], data=dat, method='rf', trControl=trainControl(method="cv"
caret_matrix
```

```
## Random Forest
##
## 310 samples
##   6 predictor
##   2 classes: 'Abnormal', 'Normal'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 248, 248, 248, 248, 248
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.8193548  0.5826725
```

```
##   4     0.8225806  0.5945959
##   6     0.8225806  0.5996912
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

Support Vector Machine (SVM) model

```
caret_svm <- train(x=dat[,1:6], y=dat[,7], data=dat, method='svmRadial', trControl=trainControl(method=
caret_svm
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 310 samples
##   6 predictor
##   2 classes: 'Abnormal', 'Normal'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 248, 248, 248, 248, 248
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.8483871  0.6422330
##   0.50  0.8516129  0.6537262
##   1.00  0.8612903  0.6752598
##
## Tuning parameter 'sigma' was held constant at a value of 0.2651415
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.2651415 and C = 1.
```

Gradient Boosting Machine (GBM) model

```
caret_boost <- train(class~pelvic_incidence+pelvic_tilt.numeric+lumbar_lordosis_angle+sacral_slope+
pelvic_radius+
  degree_spondylolisthesis, data=dat, method='gbm', preProcess= c('center', 'scale'), trControl=trainCo
print(caret_boost)
```

```
## Stochastic Gradient Boosting
##
## 310 samples
##   6 predictor
##   2 classes: 'Abnormal', 'Normal'
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (7 fold)
## Summary of sample sizes: 266, 266, 266, 265, 266, 265, ...
## Resampling results across tuning parameters:
```

```
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.8161616  0.5751694
##   1                  100      0.8388167  0.6329559
##   1                  150      0.8387446  0.6255215
##   2                   50      0.8292208  0.6131845
##   2                  100      0.8387446  0.6296751
##   2                  150      0.8321789  0.6197502
##   3                   50      0.8613276  0.6823848
##   3                  100      0.8451659  0.6462887
##   3                  150      0.8259019  0.6003576
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 50, interaction.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

Correlation between algorithm

We are interested in correlation because uncorrelated models do better when they are assembled than correlated presentations.

Combining Models

Ensembling is a technique of combining two or more algorithms of similar or dissimilar types called base learners. This is done to make a more robust system which incorporates the predictions from all the base learners

Now we can take all of these predictions into account while making the final decision. This will make our final decision more robust, accurate and less likely to be biased. The final decision would have been opposite if one of these traders would have made this decision alone.

Majority vote ensemble for all the three models

Majority vote:It's defined as taking the prediction with maximum vote / recommendation from multiple models predictions while predicting the outcomes of a classification problem.

```
#recodify  our value
qualite<-c('Abnormal'=0,'Normal'=1)
dat$class<-as.factor(revalue(dat$class,qualite))
#Spliting training set into two parts based on outcome: 70% and 30%

index <- sample(2,nrow(dat),replace= TRUE,prob=c(0.7,0.3))
trainClean <- dat[index==1,]
testClean <- dat[index==2,]


# Random Forest model
```

```
caret_matrix <- train(x=trainClean[,1:6], y=trainClean[,7], data=trainClean, method='rf', trControl=tra
caret_matrix
```

```
## Random Forest
##
## 205 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 163, 164, 165, 164, 164
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.8490999  0.6529273
##   4     0.8535075  0.6688956
##   6     0.8632636  0.6883563
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 6.
```

```
solution_rf <- predict(caret_matrix, testClean)
```

```
# Support Vector Machine (SVM) model
caret_svm <- train(x=trainClean[,1:6], y=trainClean[,7], data=trainClean, method='svmRadial', trControl=
caret_svm
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 205 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 165, 164, 164, 164, 163
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.8488676  0.6515177
##   0.50  0.8539779  0.6663382
##   1.00  0.8637340  0.6896225
##
## Tuning parameter 'sigma' was held constant at a value of 0.2710532
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.2710532 and C = 1.
```

```
solution_svm <- predict(caret_svm, testClean)
```

```
## Warning in method$predict(modelFit = modelFit, newdata = newdata, submodels =
## param): kernlab class prediction calculations failed; returning NAs
```

```
#  Gradient Boosting Machine (GBM) model
caret_boost <- train(class~pelvic_incidence+pelvic_tilt.numeric+lumbar_lordosis_angle+sacral_slope+pelvi
print(caret_boost)
```

```
## Stochastic Gradient Boosting
##
## 205 samples
##   6 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (7 fold)
## Summary of sample sizes: 176, 176, 175, 177, 176, 175, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.8393620  0.6344864
##   1                  100      0.8582336  0.6731291
##   1                  150      0.8728478  0.7075420
##   2                   50      0.8733638  0.7124520
##   2                  100      0.8783017  0.7265360
##   2                  150      0.8735398  0.7146736
##   3                   50      0.8978419  0.7708901
##   3                  100      0.8878255  0.7525972
##   3                  150      0.8881539  0.7504804
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 50, interaction.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

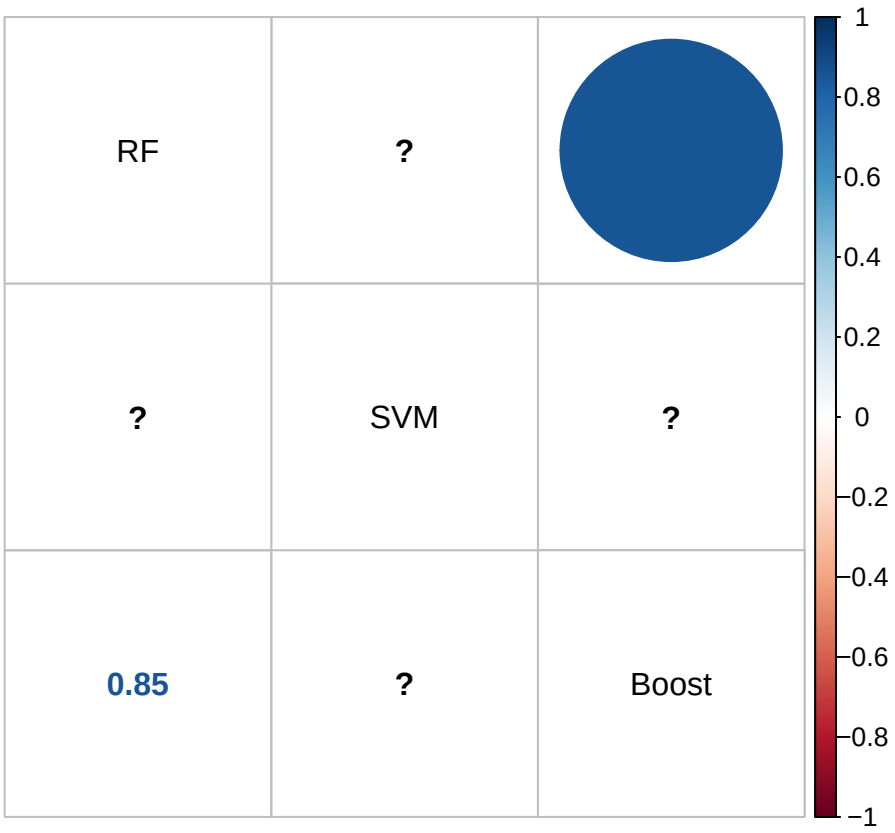
```
solution_boost <- predict(caret_boost, testClean)
```

Correlation between models

We are interested in correlation because uncorrelated models do better when they are assembled than correlated presentations.

```
#adding model predictions to test dataframe
testClean$RF <- as.numeric(solution_rf)-1
testClean$SVM <- as.numeric(solution_svm)-1
testClean$Boost <- as.numeric(solution_boost)-1

#compose correlations plot
corrplot.mixed(cor(testClean[, c('RF', 'SVM', 'Boost')]), tl.col="black")
```



We only observe a correlation, and a very strong correlation between RF and GBM, this is probably not surprising because these two algorithms are of very similar nature but on the other hand the SVM is very different algorithm of these two of or can be the lack of ability to provide the degree of correlation between the two previous models. We will now move to the majority vote, let us note

If 0 or 1 model predicts 'Normal', the overall prediction will be 'Abnormal'

If 2 or 3 models predict 'Normal', the overall prediction will be 'Normal'

```
testClean$Sum <- testClean$RF + testClean$SVM + testClean$Boost
testClean$Majority <- ifelse(testClean$Sum<=1, 0,1)
```

KNN Model Analysis

```
caret_knn <- train(class~., data=dat, method='knn', trControl=trainControl(method="cv", number=5),tuneL
caret_knn
```

```
## k-Nearest Neighbors
##
## 310 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
```

```
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 248, 248, 248, 248, 248
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.8451613  0.6478670
##    7  0.8451613  0.6491898
##    9  0.8516129  0.6683433
##   11  0.8419355  0.6506181
##   13  0.8516129  0.6699149
##   15  0.8322581  0.6280133
##   17  0.8419355  0.6502643
##   19  0.8419355  0.6487686
##   21  0.8548387  0.6744509
##   23  0.8548387  0.6781268
##   25  0.8548387  0.6781268
##   27  0.8451613  0.6575559
##   29  0.8483871  0.6665305
##   31  0.8354839  0.6422849
##   33  0.8322581  0.6349675
##   35  0.8258065  0.6243601
##   37  0.8258065  0.6238018
##   39  0.8225806  0.6200497
##   41  0.8161290  0.6070226
##   43  0.8161290  0.6095299
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 25.
```

```
#train
caret_knn <- train(class~., data=trainClean, method='knn', trControl=trainControl(method="cv", number=5)
caret_knn
```

```
## k-Nearest Neighbors
##
## 205 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 164, 164, 165, 163, 164
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.8687398  0.7111145
##    7  0.8539895  0.6843666
##    9  0.8492218  0.6742264
##   11  0.8345877  0.6471420
##   13  0.8348258  0.6552973
##   15  0.8250697  0.6341978
##   17  0.8297154  0.6504472
##   19  0.8298316  0.6512623
##   21  0.8347096  0.6622626
```

```
##   23  0.8347096  0.6622626
##   25  0.8201916  0.6342302
##   27  0.8250697  0.6433236
##   29  0.8250697  0.6433236
##   31  0.8201916  0.6331091
##   33  0.8201916  0.6328806
##   35  0.8250697  0.6438809
##   37  0.8250697  0.6438809
##   39  0.8201916  0.6342302
##   41  0.8055517  0.6061465
##   43  0.8005517  0.5941161
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

Conclusion

We analyze the Normal and Abnormal patient for different six biomechanical different biomechanicla desease.Found that 210 people are Abnormal and 100 people are normal.We have noticed that degree spondylolisthesis is the most important of the variables to explain the normal and abnormal of the patients.we analyze with decision tree ,boxplot.Also found the relationship between all the six different variables.There is correlation between degree spondylolisthesis and other variable.Also Noticed that high correlation between sacral_slope and pelvic_incidence. Also analyze the different algorithm like RandomForest,Support Vector Machine (SVM) model,Gradient Boosting Machine (GBM) modeland KNN model.