HarvardX: PH125.9x Capstone Choose Your Own Project

Andrea Blasio

March 14th, 2020

Contents

1 Introduction

The project described in this document is aimed at solving a machine learning challenge based on a freely chosen dataset available in the public domain as required by the *HarvardX PH125.9x Capstone Choose Your Own* exam; its purpose is to build a model to perform binary classification prediction on the *Biomechanical features of orthopedic patients* dataset distributed by University of California, School of Information and Computer Science (M. Lichman, *UCI Machine Learning Repository*, 2013) and published by Kaggle in a curated list of materials suitable for training in the data science field.

We will be focusing on the **column2Cweka.csv** set, containing **310** observations related to patients potentially affected by spinal diseases, **100** of which have been classified as *normal* and **210** as *abnormal* based on the features described in the measurements. Given a test subset, our predictive model should allow to accurately perform such binary classification.

The following script loads the dataset:

```
# Install and attach the required add-on packages
required_packages <- c("dplyr", "caret", "corrplot", "GGally", "kernlab",
                       "C50", "klaR", "knitr", "kableExtra", "grid", "gridExtra")

for (package in required_packages) {
  if (!require(package, character.only = TRUE)) {
    install.packages(package)
  }
  library(package, character.only = TRUE)
}
```

```r
# Load dataset
data <- read.csv("./biomechanical-features-of-orthopedic-patients/column_2C_weka.csv",
                 stringsAsFactors = FALSE)

# Remove temporary variables
rm(required_packages, package)
```

2 Analysis and data preparation

Each of the **310** observations contains **7 variables**, 6 of which are quantitative continuous values defining spinal features; the latter are then summarized in the *class* variable, containing a qualitative nominal category, that indicates their adherence to either *normal* or *abnormal* patients' subgroups:

```r
glimpse(data)
```

```
## Observations: 310
## Variables: 7
## $ pelvic_incidence         <dbl> 63.02782, 39.05695, 68.83202, 69.29701, 49.7…
## $ pelvic_tilt.numeric      <dbl> 22.552586, 10.060991, 22.218482, 24.652878, …
## $ lumbar_lordosis_angle    <dbl> 39.60912, 25.01538, 50.09219, 44.31124, 28.3…
## $ sacral_slope             <dbl> 40.47523, 28.99596, 46.61354, 44.64413, 40.0…
## $ pelvic_radius            <dbl> 98.67292, 114.40543, 105.98514, 101.86850, 1…
## $ degree_spondylolisthesis <dbl> -0.2544000, 4.5642586, -3.5303173, 11.211523…
## $ class                    <chr> "Abnormal", "Abnormal", "Abnormal", "Abnorma…
```

```r
# List of available classes
unique(data$class)
```

```
## [1] "Abnormal" "Normal"
```

```r
# Convert classes expressed as character strings to factors
data <- data %>% mutate(class = as.factor(class))
```

```r
summary(data)
```

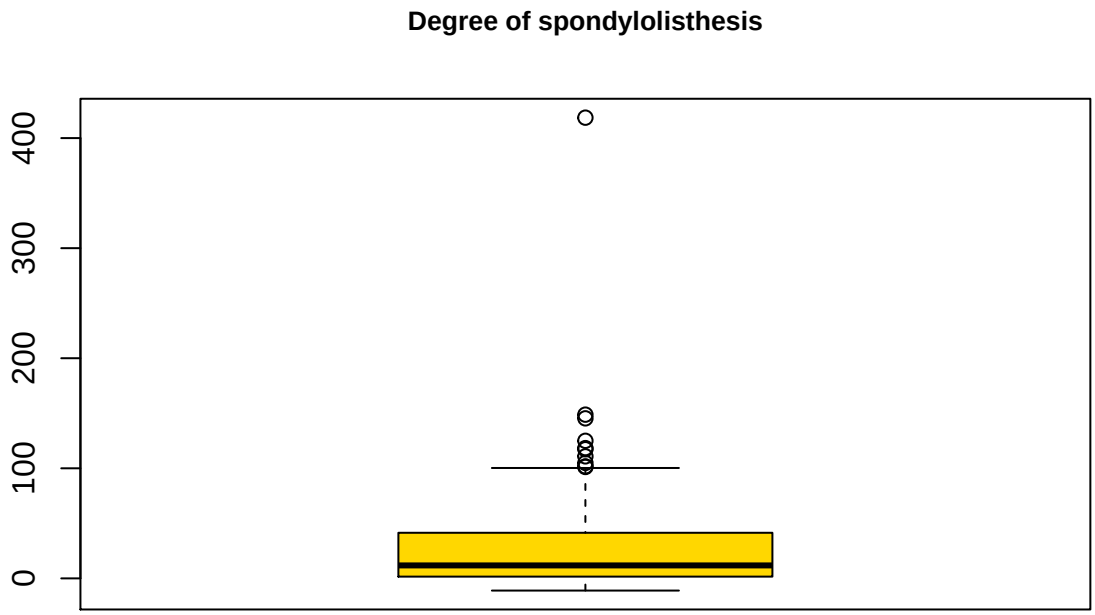
```
##  pelvic_incidence pelvic_tilt.numeric lumbar_lordosis_angle  sacral_slope
##  Min.   : 26.15   Min.   :-6.555      Min.   : 14.00        Min.   : 13.37
##  1st Qu.: 46.43   1st Qu.:10.667      1st Qu.: 37.00        1st Qu.: 33.35
##  Median : 58.69   Median :16.358      Median : 49.56        Median : 42.40
##  Mean   : 60.50   Mean   :17.543      Mean   : 51.93        Mean   : 42.95
##  3rd Qu.: 72.88   3rd Qu.:22.120      3rd Qu.: 63.00        3rd Qu.: 52.70
##  Max.   :129.83   Max.   :49.432      Max.   :125.74        Max.   :121.43
##  pelvic_radius    degree_spondylolisthesis      class
##  Min.   : 70.08   Min.   :-11.058          Abnormal:210
##  1st Qu.:110.71   1st Qu.:  1.604          Normal  :100
##  Median :118.27   Median : 11.768
##  Mean   :117.92   Mean   : 26.297
##  3rd Qu.:125.47   3rd Qu.: 41.287
##  Max.   :163.07   Max.   :418.543
```

```r
# Check for any not available variables
anyNA(data)
```

```
## [1] FALSE
```

The *degree_spondylolisthesis* variable is characterized by a mean value that is more than double than its median, highlighting the possible incidence of outliers:

```r
# Draw boxplot to highlight any possible outliers
# in the degree_spondylolisthesis variable distribution
boxplot(data$degree_spondylolisthesis,
        main="Degree of spondylolisthesis",
        cex.main = 0.8,
        col="gold")
```

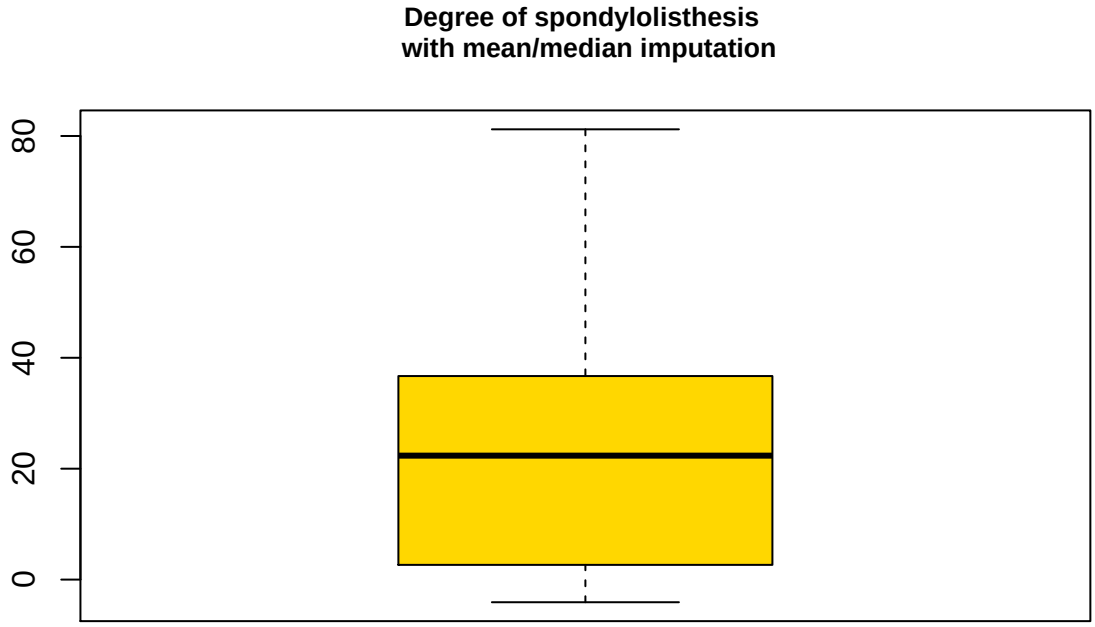


Outliers are then treated via *mean/median imputation*:

```r
# Reference: K. Ganguly, R Data Analysis Cookbook (2nd edition), Packt, 2017.
impute_outliers <- function(x, removeNA = TRUE){
    quantiles <- quantile(x, c(.05, .95), na.rm = removeNA)
    x[ x < quantiles[1] ] <- mean(x, na.rm = removeNA)
    x[ x > quantiles[2] ] <- median(x, na.rm = removeNA)
    x
}
```

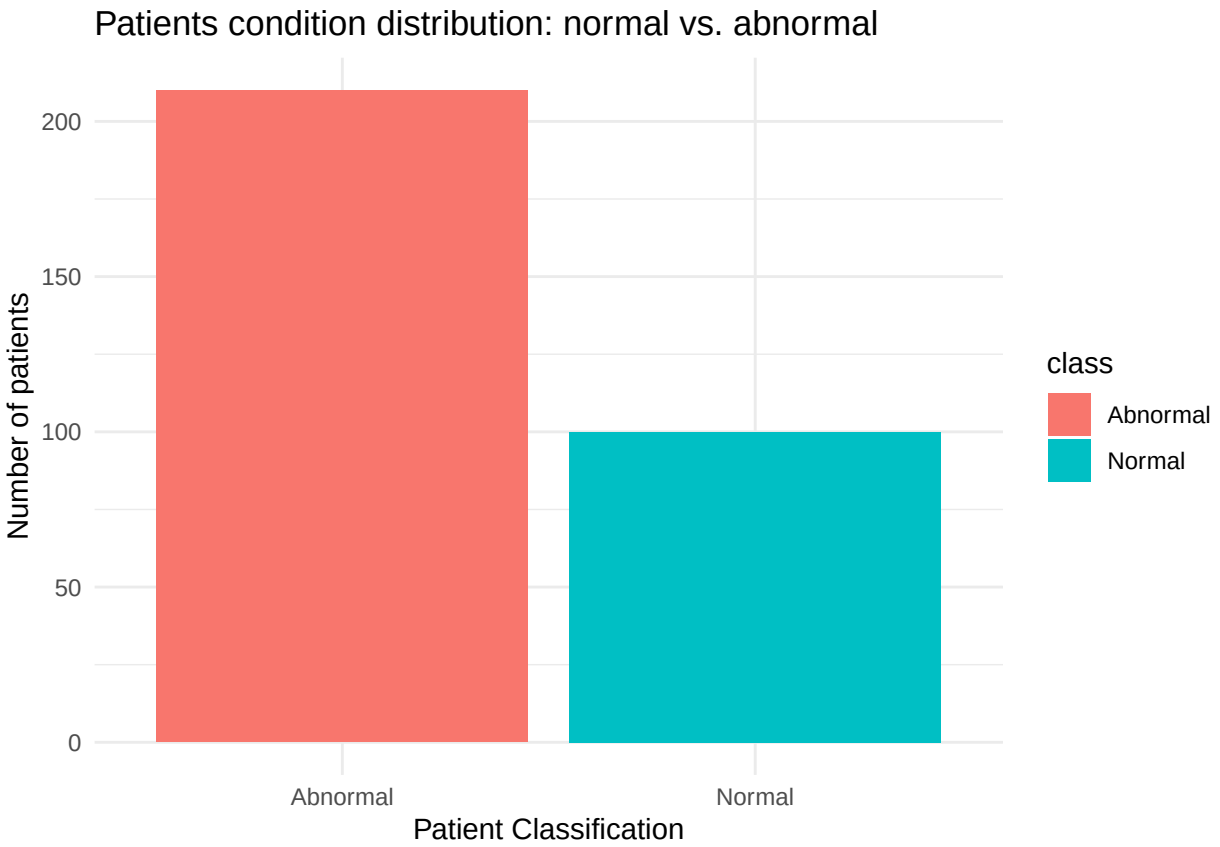
```r
# Apply mean/median imputation to degree_spondylolisthesis variable
data$degree_spondylolisthesis <- impute_outliers(data$degree_spondylolisthesis)

# Plot degree_spondylolisthesis variable distribution
# resulting from mean/median imputation
boxplot(data$degree_spondylolisthesis,
        main="Degree of spondylolisthesis \n with mean/median imputation",
        cex.main = 0.8,
        col="gold")
```



Patients with spinal features classified as *abnormal* are more than double than *normal* cases:

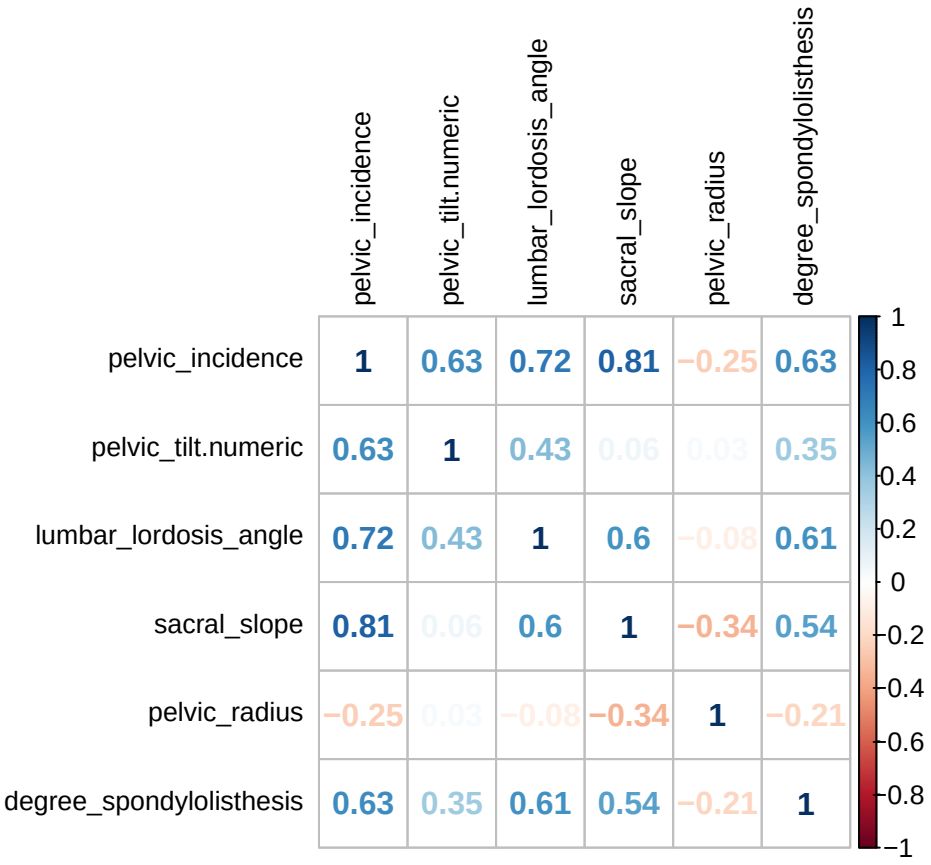
```r
# Patients condition distribution: normal vs. abnormal
data %>% ggplot(aes(class, fill = class)) +
  geom_bar(stat = "count") +
  labs(x = "Patient Classification", y = "Number of patients") +
  ggtitle("Patients condition distribution: normal vs. abnormal") +
  theme_minimal()
```



Visualization of correlation between quantitative values highlights the following:

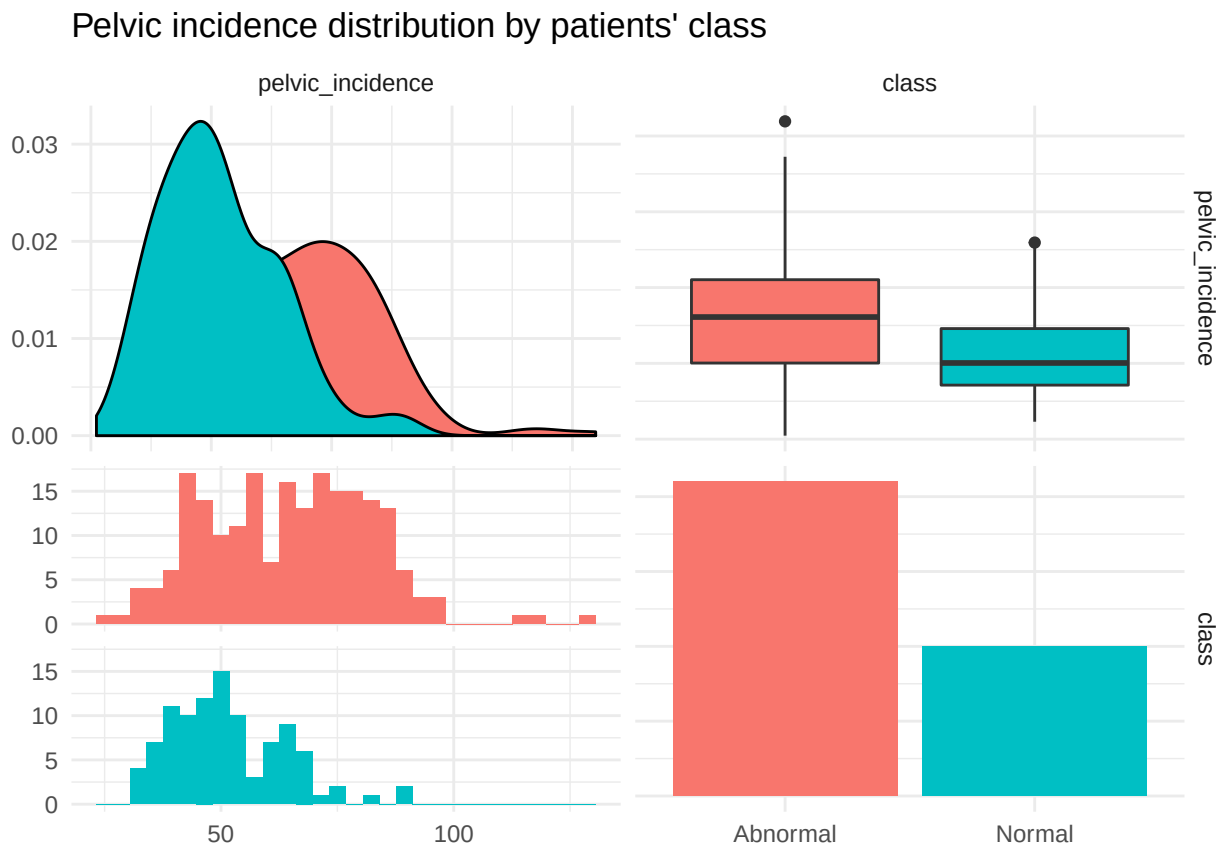
- *pelvic_radius* has the lowest correlation;
- highest correlation ratios are respectively found in *pelvic_incidence* in relation to *sacral_slope*, *degree_spondyloisthesis*, *lumbar_lordosis_angle* and *pelvic_tilt.numeric*.

```
# Plot variables correlation
M <- cor(data[,1:6])
corrplot(M, method = "number", tl.cex = 0.8, tl.col = "black")
```



Pelvic incidence appears to also have a slightly higher mean in *abnormal* subjects:

```
# Plot the pelvic incidence distribution by patients' class
data %>%
ggpairs(columns = c(1, ncol(data)), aes(fill = class)) +
  ggtitle("Pelvic incidence distribution by patients' class") +
  theme_minimal()
```



Data is split in two subsets suitable for respectively training (80%) and testing (20%) the binary classification predictive models:

```
# Split dataset in two subsets for training and validation
set.seed(100)
test_index <- createDataPartition(y = data$class, p = 0.2, list = FALSE)
training_set <- data[-test_index,]
validation_set <- data[test_index,]

# Remove temporary variables
rm(test_index)
```

3 Results

Four different algorithms appropriate for classification tasks are employed and estimated in order to build an efficient predictive model: *support vector machines with polynomial kernel* (**svmPoly**), *decision tree* (**C5.0**), *naïve Bayes* (**nb**) and *neural network* (**nnet**); computational nuances of each model are checked via 10-fold cross validation with three repeats.

```
# Check the computational outcome of each model via 10-fold cross validation
# with three repeats
control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

# Model training via support vector machines with polynomial kernel
```

```r
svm_model <- train(class~., data = training_set,
                   method = "svmPoly",
                   trControl= control,
                   tuneGrid = data.frame(degree = 1,
                                         scale = 1,
                                         C = 1),
                   preProcess = c("pca","scale","center")
)

# Predictions outcome
svm_predictions <- predict(svm_model, validation_set)

# Create confusion matrix
svm_confusion_matrix <- confusionMatrix(svm_predictions, validation_set$class)

# Store accuracy
accuracy_summary = tibble(Model = "Support vector machines with polynomial kernel",
                          Accuracy = svm_confusion_matrix$overall["Accuracy"])

# Print confusion matrix
svm_confusion_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal       38      5
##   Normal          4     15
##
##                Accuracy : 0.8548
##                  95% CI : (0.7422, 0.9314)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 0.001243
##
##                   Kappa : 0.6634
##
##  Mcnemar's Test P-Value : 1.000000
##
##             Sensitivity : 0.9048
##             Specificity : 0.7500
##          Pos Pred Value : 0.8837
##          Neg Pred Value : 0.7895
##              Prevalence : 0.6774
##          Detection Rate : 0.6129
##    Detection Prevalence : 0.6935
##       Balanced Accuracy : 0.8274
##
##        'Positive' Class : Abnormal
##
```

```r
# Model training via C5.0 (decision tree)
decision_tree_model <- train(class~., data = training_set,
                             method = "C5.0",
```

```
                            preProcess=c("scale", "center"),
                            trControl= control,
                            na.action = na.omit,
                            trace = FALSE
)

# Predictions outcome
decision_tree_predictions <- predict(decision_tree_model, validation_set)

# Create confusion matrix
C50_confusion_matrix <- confusionMatrix(decision_tree_predictions, validation_set$class)

# Store accuracy
accuracy_summary <- bind_rows(
  accuracy_summary,
  tibble(Model = "C5.0 (decision tree)",
  Accuracy = C50_confusion_matrix$overall["Accuracy"])
)

# Print confusion matrix
C50_confusion_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal       38      2
##   Normal          4     18
##
##                Accuracy : 0.9032
##                  95% CI : (0.8012, 0.9637)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 2.961e-05
##
##                   Kappa : 0.7842
##
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.9048
##             Specificity : 0.9000
##          Pos Pred Value : 0.9500
##          Neg Pred Value : 0.8182
##              Prevalence : 0.6774
##          Detection Rate : 0.6129
##    Detection Prevalence : 0.6452
##       Balanced Accuracy : 0.9024
##
##        'Positive' Class : Abnormal
##
```

```
# Naïve Bayes algorithm
naive_model <- train(class~., data = training_set,
                     method = "nb",
```

```
                    preProcess=c("scale","center"),
                    trControl= control
)

# Predictions outcome
naive_predictions <- predict(naive_model, validation_set, na.action = na.pass)

# Create confusion matrix
naive_bayes_confusion_matrix <- confusionMatrix(naive_predictions, validation_set$class)

accuracy_summary <- bind_rows(
  accuracy_summary,
  tibble(Model = "Naïve Bayes",
  Accuracy = naive_bayes_confusion_matrix$overall["Accuracy"])
)

naive_bayes_confusion_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal       33      4
##   Normal          9     16
##
##                Accuracy : 0.7903
##                  95% CI : (0.6682, 0.8834)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 0.03518
##
##                   Kappa : 0.5497
##
##  Mcnemar's Test P-Value : 0.26726
##
##             Sensitivity : 0.7857
##             Specificity : 0.8000
##          Pos Pred Value : 0.8919
##          Neg Pred Value : 0.6400
##              Prevalence : 0.6774
##          Detection Rate : 0.5323
##    Detection Prevalence : 0.5968
##       Balanced Accuracy : 0.7929
##
##        'Positive' Class : Abnormal
##
```

```
# Train model with neural network
neural_network_model <- train(class~., data = training_set,
                              method = "nnet",
                              trControl = control,
                              preProcess = c("scale","center"),
                              trace = FALSE
)
```

```r
neural_network_predictions <- predict(neural_network_model, validation_set)

# Create confusion matrix
neural_network_confusion_matrix <- confusionMatrix(neural_network_predictions,
                                                   validation_set$class)

# Store accuracy
accuracy_summary <- bind_rows(
  accuracy_summary,
  tibble(Model = "Neural network",
  Accuracy = neural_network_confusion_matrix$overall["Accuracy"])
)

# Print confusion matrix
neural_network_confusion_matrix
```

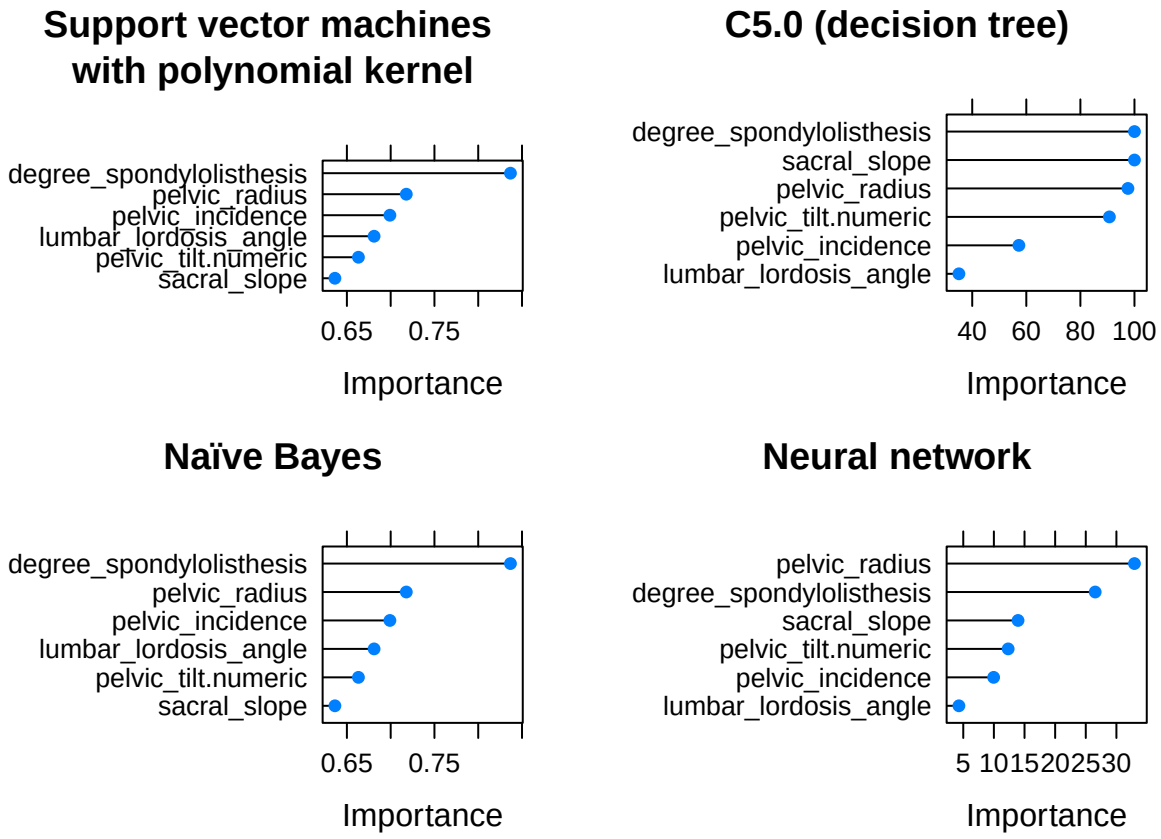
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Abnormal Normal
##   Abnormal       38      6
##   Normal          4     14
##
##                Accuracy : 0.8387
##                  95% CI : (0.7233, 0.9198)
##     No Information Rate : 0.6774
##     P-Value [Acc > NIR] : 0.003344
##
##                   Kappa : 0.621
##
##  Mcnemar's Test P-Value : 0.751830
##
##             Sensitivity : 0.9048
##             Specificity : 0.7000
##          Pos Pred Value : 0.8636
##          Neg Pred Value : 0.7778
##              Prevalence : 0.6774
##          Detection Rate : 0.6129
##    Detection Prevalence : 0.7097
##       Balanced Accuracy : 0.8024
##
##        'Positive' Class : Abnormal
##
```

The *degree_spondylolisthesis* appears to be a relatively good predictor of a patient's class, with *pelvic radius* having a prominent role in the *C5.0* and *neural network* models:

```r
# Compute the variables importance in each predictive model
svm_model_importance <- varImp(svm_model, scale = FALSE)
decision_tree_model_importance <- varImp(decision_tree_model, scale = FALSE)
naive_model_importance <- varImp(naive_model, scale = FALSE)
neural_network_importance <- varImp(neural_network_model, scale = FALSE)
```

```
# Plot the variables importance in each predictive model
p1 <- plot(svm_model_importance, main="Support vector machines \n with polynomial kernel ")
p2 <- plot(decision_tree_model_importance, main="C5.0 (decision tree)")
p3 <- plot(naive_model_importance, main="Naïve Bayes")
p4 <- plot(neural_network_importance, main="Neural network")

grid.arrange(p1, p2, p3, p4, ncol = 2)
```



4 Conclusion

The **C5.0 (decision tree)** model is by far the most accurate:

```
accuracy_summary %>%
  arrange(desc(Accuracy)) %>%
  knitr::kable() %>%
  kable_styling()
```

Model	Accuracy
C5.0 (decision tree)	0.9032258
Support vector machines with polynomial kernel	0.8548387
Neural network	0.8387097
Naïve Bayes	0.7903226

Model evaluation could be further sharpened by leveraging *ROC* curves in order to fine tune the binary classification threshold selection and reduce false negative outcomes, which are particularly undesiderable in the medical field.

5 Appendix: system configuration and R version

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          6.1
## year           2019
## month          07
## day            05
## svn rev        76782
## language       R
## version.string R version 3.6.1 (2019-07-05)
## nickname       Action of the Toes
```