Predicting low back pain symptoms with machine learning

Antonio Cappuccio

18 settembre 2016

Summary

Low back pain is major cause of morbidity in the developed world, affecting 40% of people at some point in their lives (ref) (https://en.wikipedia.org/wiki/Low_back_pain#cite_note-malhotra_2011-21). Recently, Kaggle released a dataset involving 310 patients prone to low back pain (ref) (https://www.kaggle.com/sammy123/lower-back-pain-symptoms-dataset). The spine of each patient was classified by practitioners as either normal (n=100) or abnormal (n=210). In addition, each spine was characterized by a set of 12 measured physical parameters. Here, I use the dataset to train and test a classifier discriminating normal Vs. abnormal spines. A basic random forest classifier gives reasonable sensitivity (82%) and specificity (72%). Upon further improvement of performance, the classifier may assist practitioners in the clinical diagnosis of abonormal spines prone to low back pain.

Data cleaning

```
#graphical options
source("setPowerPointStyle.R")
setPowerPointStyle()

#reading data
spine=read.csv2("Dataset_spine.csv",sep=",",stringsAsFactors = F)

str(spine)
```

```
## 'data.frame':    310 obs. of  14 variables:
##  $ Col1     : chr  "63.0278175" "39.05695098" "68.83202098" "69.29700807" ...
##  $ Col2     : chr  "22.55258597" "10.06099147" "22.21848205" "24.65287791" ...
##  $ Col3     : chr  "39.60911701" "25.01537822" "50.09219357" "44.31123813" ...
##  $ Col4     : chr  "40.47523153" "28.99595951" "46.61353893" "44.64413017" ...
##  $ Col5     : chr  "98.67291675" "114.4054254" "105.9851355" "101.8684951" ...
##  $ Col6     : chr  "-0.254399986" "4.564258645" "-3.530317314" "11.21152344" ...
##  $ Col7     : chr  "0.744503464" "0.415185678" "0.474889164" "0.369345264" ...
##  $ Col8     : chr  "12.5661" "12.8874" "26.8343" "23.5603" ...
##  $ Col9     : chr  "14.5386" "17.5323" "17.4861" "12.7074" ...
##  $ Col10    : chr  "15.30468" "16.78486" "16.65897" "11.42447" ...
##  $ Col11    : chr  "-28.658501" "-25.530607" "-29.031888" "-30.470246" ...
##  $ Col12    : chr  "43.5123" "16.1102" "19.2221" "18.8329" ...
##  $ Class_att: chr  "Abnormal" "Abnormal" "Abnormal" "Abnormal" ...
##  $ X        : chr  "" "" "Prediction is done by using binary classification." "" ...
```

The data requires some cleaning: the variable names are awkward and in the wrong place. They occupy some rows in column 14. Let’s modify them and move them to the right place, that is, the columns they refer to.

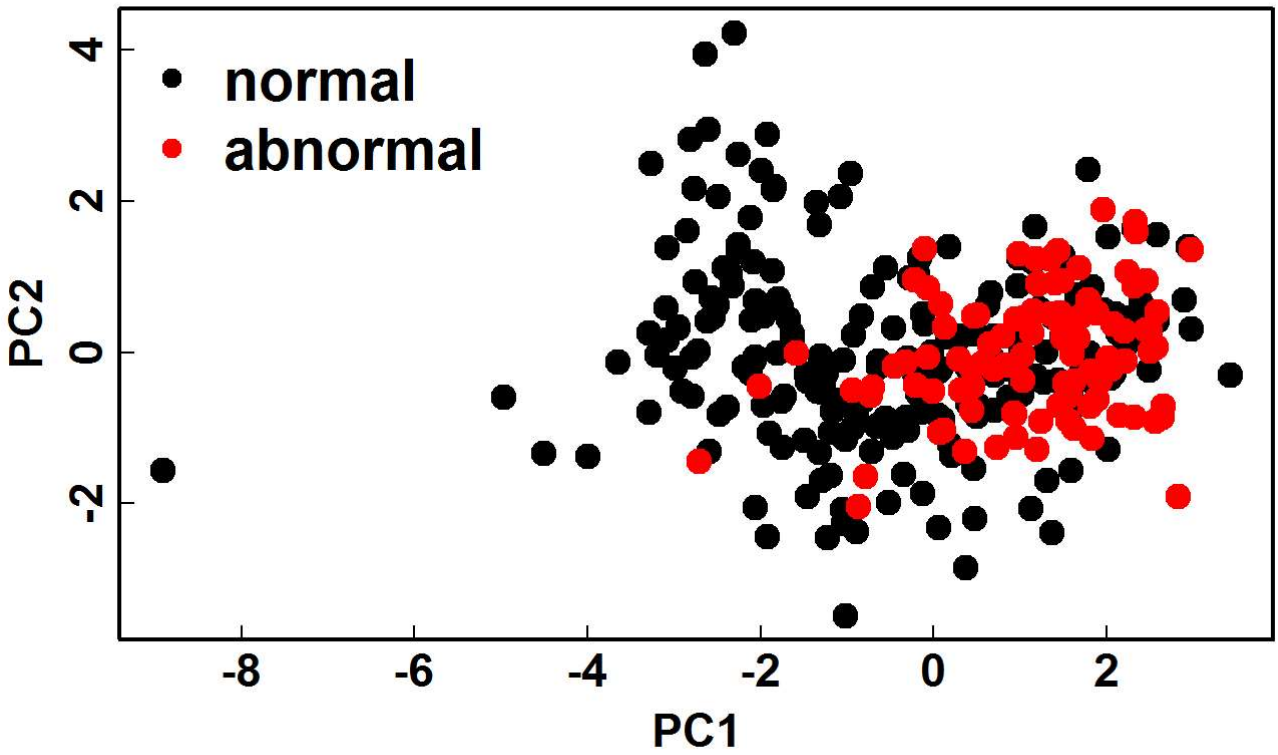
```
feature_names=spine[6:17,14]

feature_names_tidy=sapply(feature_names,
                          function(x) strsplit(x,"=")[[1]][2])

feature_names_tidy=sapply(feature_names_tidy,
                          function(x) trimws(gsub("\\(numeric\\)","",x)))

#let's move the variable names in the corresponding columns
colnames(spine)=c(feature_names_tidy,"predicted_class")

#remove useless column
spine=spine[,-14]

#convert numbers to numeric data
spine[,1:12]=data.matrix(spine[,1:12])

#last column is a factor
spine[,13]=as.factor(spine[,13])

head(spine)
```

```
##   pelvic_incidence pelvic_tilt 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 pelvic_slope Direct_tilt
## 1      98.67292                -0.254400    0.7445035     12.5661
## 2     114.40543                 4.564259    0.4151857     12.8874
## 3     105.98514                -3.530317    0.4748892     26.8343
## 4     101.86850                11.211523    0.3693453     23.5603
## 5     108.16872                 7.918501    0.5433605     35.4940
## 6     130.32787                 2.230652    0.7899929     29.3230
##   thoracic_slope cervical_tilt sacrum_angle scoliosis_slope
## 1        14.5386      15.30468   -28.658501         43.5123
## 2        17.5323      16.78486   -25.530607         16.1102
## 3        17.4861      16.65897   -29.031888         19.2221
## 4        12.7074      11.42447   -30.470246         18.8329
## 5        15.9546       8.87237   -16.378376         24.9171
## 6        12.0036      10.40462    -1.512209          9.6548
##   predicted_class
## 1        Abnormal
## 2        Abnormal
## 3        Abnormal
## 4        Abnormal
## 5        Abnormal
## 6        Abnormal
```

Looks like an honest dataset now!

Unsupervised analysis

```
setPowerPointStyle()

spine_pca=prcomp(spine[,1:12],scale. = T)
plot(spine_pca$x[,1], spine_pca$x[,2],col=spine$predicted_class,
     xlab="PC1",ylab="PC2",main="",pch=20,cex=2)

legend("topleft", pch = 20, col=c("black","red"),
       legend = c("normal","abnormal"), bty='n',cex=1.5)
```



There is some degree of separation in two groups along PC1, but it is not so clear. One donor on the let side might be an outlier. Before applying machine learning, let's see more in depth what really marks the difference between Abnormal and Normal spines.

Which variables discriminate Abnormal Vs. Normal spines?

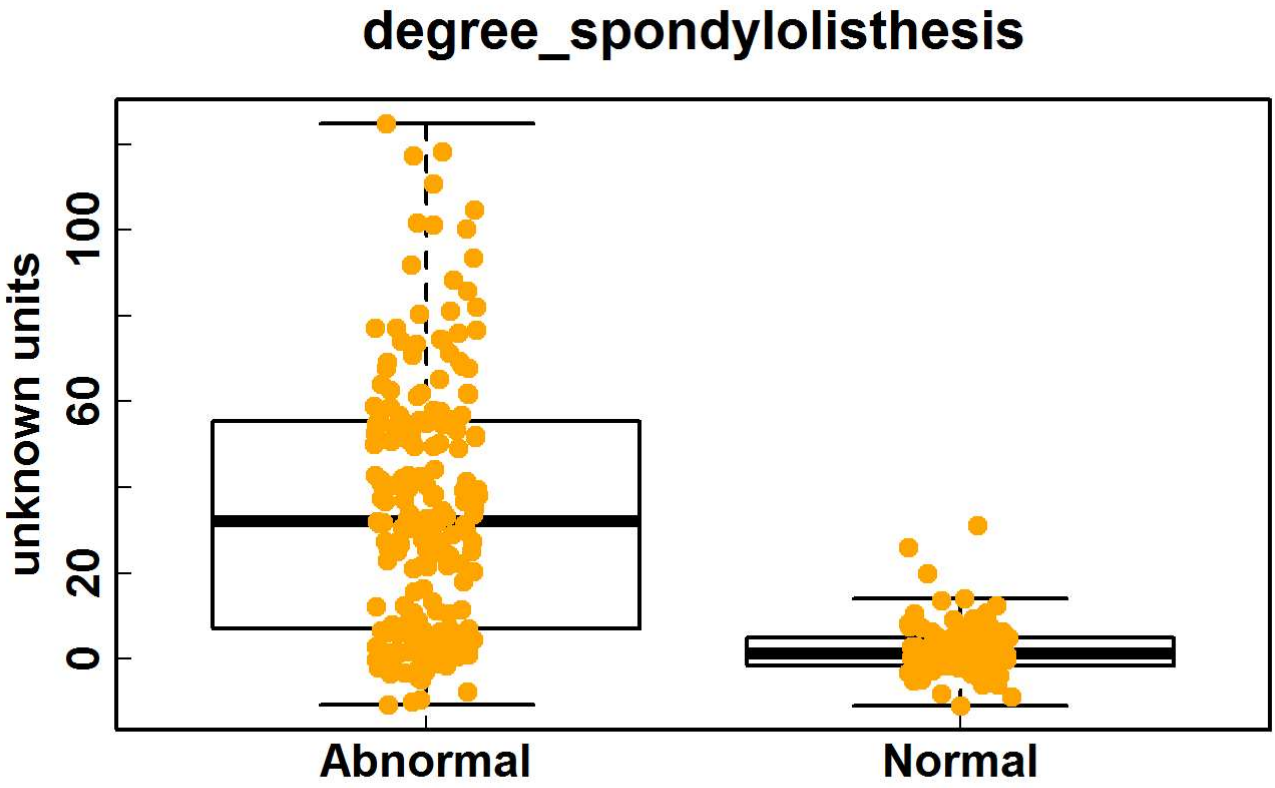
```
source("setPowerPointStyle.R")
setPowerPointStyle()

#class numerosity
summary(spine[,13])
```

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

```
#there is some class imbalance to bear in mind

#identifying variables with significant differences, as measured by t.test p-values

stat_test=vector(length = 12)
names(stat_test)=names(spine)[1:12]

for (k in 1:12){

  stat_test[k]=t.test(spine[spine["predicted_class"]=="Normal",k],
       spine[spine["predicted_class"]=="Abnormal",k])$p.value

  }

#correcting p-values for multiple testing
stat_test_adj=p.adjust(stat_test)

print(cbind(sort(stat_test_adj)))
```

```
##                                [,1]
## degree_spondylolisthesis 3.589907e-26
## pelvic_incidence         1.115125e-11
## pelvic_tilt              1.679709e-10
## lumbar_lordosis_angle    7.654428e-10
## pelvic_radius            1.233155e-09
## sacral_slope             1.362633e-04
## cervical_tilt            4.993720e-01
## pelvic_slope             1.000000e+00
## Direct_tilt              1.000000e+00
## thoracic_slope           1.000000e+00
## sacrum_angle             1.000000e+00
## scoliosis_slope          1.000000e+00
```

```
#the most significant variable is degree_spondylolisthesis
boxplot(spine[,"degree_spondylolisthesis"]~ spine[,"predicted_class"],
        ylab="unknown units",main="degree_spondylolisthesis",outline=FALSE)

stripchart(spine[,"degree_spondylolisthesis"]~ spine[,"predicted_class"], vertical = TRUE,
    method = "jitter", add = TRUE, pch = 20, col = 'orange',cex=1.5)
```



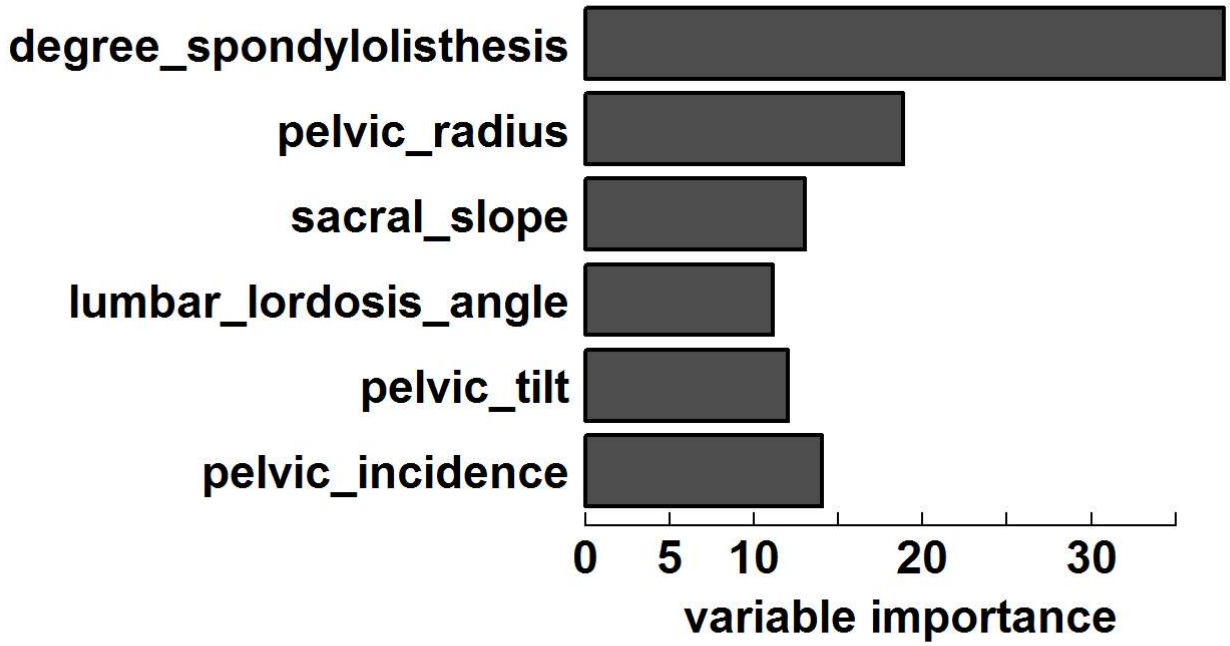

```
#for further analysis I only keep significant variables (p<0.05)
spine_red=spine[,c(which(stat_test_adj<0.05),13)]
```

I checked out on google what the degree of spondylolisthesis is (https://en.wikipedia.org/wiki/Spondylolisthesis (https://en.wikipedia.org/wiki/Spondylolisthesis)). Spondylolisthesis is the forward displacement of one vertrebra over the other in lumbar region.

Training a binary classifier to discriminating Abnormal Vs. Normal spines

Let's see if it is possible to classify Abnormal Vs. Normal spines using the significant variables identified above. I will train a random forest classifier.

```
library(randomForest)

source("setPowerPointStyle.R")
setPowerPointStyle()

n=nrow(spine_red)
#splitting dataset in 2: 80% for training, 20% for testing
set.seed(1)
train_ind = sample(1:n, size = round(0.8*n), replace=FALSE)

train = spine_red[train_ind,]
test = spine_red[-train_ind,]

rf_model=randomForest(predicted_class~.,data=train)

#variable importance
par(mar=c(5.1,13.1,4.1,2.1))
barplot(t(importance(rf_model)),las=1,horiz = T,xlab="variable importance")
```




```
#not surprisingly, the most important variable is still the degree of spondylolisthesis
```

Assessing the classifier performance

```
confusion=table(test[,"predicted_class"],predict(rf_model, newdata=test, type="class"))
print(confusion)
```

```
##
##            Abnormal Normal
##   Abnormal       33      7
##   Normal          6     16
```

```
#considering as positive an abnormal spine, we have sensitivity=TP/(TP+FP)
sensitivity=confusion[1,1]/sum(confusion[1,])
print(sensitivity)
```

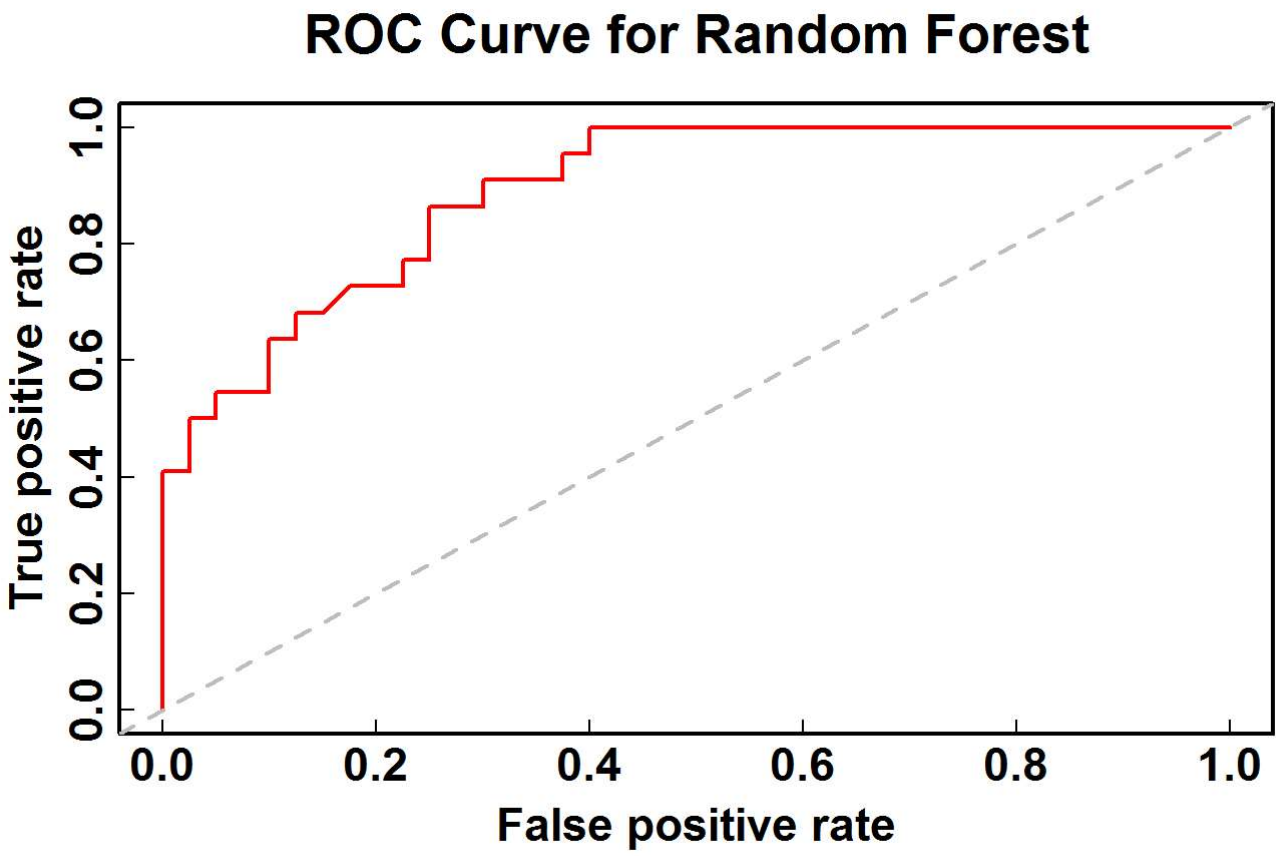
```
## [1] 0.825
```

```
specificity=confusion[2,2]/sum(confusion[2,])
print(specificity)
```

```
## [1] 0.7272727
```

ROC curve

```
library(ROCR)

source("setPowerPointStyle.R")
setPowerPointStyle()

spine.rf.pr = predict(rf_model,type="prob",newdata=test)[,2]
spine.rf.pred = prediction(spine.rf.pr, test$predicted_class)
spine.rf.perf = performance(spine.rf.pred,"tpr","fpr")
plot(spine.rf.perf,main="ROC Curve for Random Forest",col=2,lwd=2)
abline(a=0,b=1,lwd=2,lty=2,col="gray")
```


ROC Curve for Random Forest
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0