Project Challenge

Franssila Fanni, Suihkonen Sini, Savolainen Outi

December 9, 2021

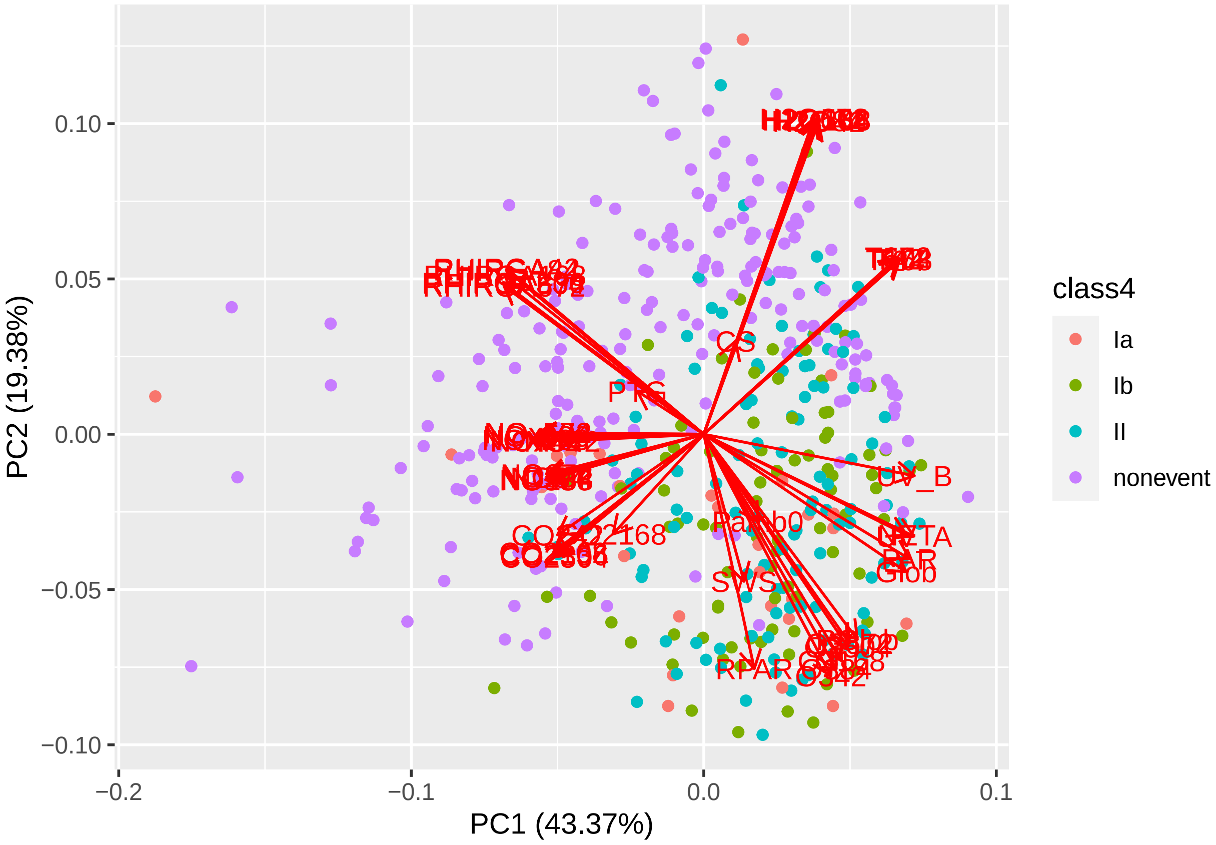
Used libraries

```
library(ggfortify)
library(randomForest)
library(caret)
```

Read data

```
set.seed(42)

npf <- read.csv("npf_train.csv")
npf_test <- read.csv("npf_test_hidden.csv")

rownames(npf) <- npf[,"date"]

vars <- colnames(npf)[sapply(colnames(npf),
                             function(s) nchar(s)>5 && substr(s,nchar(s)-4,nchar(s))==".mean")]
vars2 <- colnames(npf_test)[sapply(colnames(npf_test),
                             function(s) nchar(s)>5 && substr(s,nchar(s)-4,nchar(s))==".mean")]
npf <- npf[,c(vars,"class4")]
npf_test <- npf_test[,c(vars2)]
## strip the trailing ".mean" to make the variable names prettier
colnames(npf)[1:length(vars)] <- sapply(colnames(npf)[1:length(vars)],
                                        function(s) substr(s,1,nchar(s)-5))
vars <- colnames(npf)[1:length(vars)]

colnames(npf_test)[1:length(vars2)] <- sapply(colnames(npf_test)[1:length(vars2)],
                                        function(s) substr(s,1,nchar(s)-5))
vars2 <- colnames(npf_test)[1:length(vars2)]
```

```
npf.pcA2 <- prcomp(scale(npf[,vars]))
npf.pcA2.withtest <- prcomp(scale(rbind(npf[,vars],npf_test[,vars])))
autoplot(npf.pcA2, data=npf, colour="class4", loadings=T, loadings.label=T)
```



Binary accuracy on random forest classifier with different sized training/validation sets: ## Binary accuracy (class 2)

Accuracy of the estimate of accuracy

Perplexity

Multi-class accuracy (class4)

Multiclass accuracy on the random forest classifier:

```
set.seed(42)
# Calculates the accuracy to each class and the total accuracy
npf.pcA2 <- prcomp(npf[,vars], center=T, scale=T)

idx <- sample.int(nrow(npf),229)
training_set <- npf[ idx,]
validation_set <- npf[-idx,]
train.pc <- data.frame(npf.pcA2$x[idx,1:14])
train.pc$class4 <- npf[idx,]$class4
validate.pc <- data.frame(npf.pcA2$x[-idx,1:14])
validate.pc$class4 <- npf[-idx,]$class4
ctrl <- trainControl(method = "repeatedcv",
                     number = 10,
                     repeats = 10,
                     classProbs = TRUE)
rFClass4 <- train(factor(class4) ~ .,
                    method="rf",
```

```r
                    data=train.pc,
                    trControl=ctrl)
probs4 <- predict(rFClass4, newdata = validate.pc, type = "prob")
pred <- predict(rFClass4, newdata = validate.pc)
confusionMatrix(pred, factor(validate.pc$class4))
```

```
## Confusion Matrix and Statistics
## 
##            Reference
## Prediction  Ia  Ib  II nonevent
##   Ia         2   0   0        0
##   Ib         5  21  13        2
##   II         3  11  27        3
##   nonevent   5   7  22      108
## 
## Overall Statistics
##                                           
##                Accuracy : 0.69            
##                  95% CI : (0.6257, 0.7492)
##     No Information Rate : 0.4934          
##     P-Value [Acc > NIR] : 1.362e-09       
##                                           
##                   Kappa : 0.4925          
##                                           
##  Mcnemar's Test P-Value : 3.322e-05       
## 
## Statistics by Class:
## 
##                      Class: Ia Class: Ib Class: II Class: nonevent
## Sensitivity           0.133333    0.5385    0.4355          0.9558
## Specificity           1.000000    0.8947    0.8982          0.7069
## Pos Pred Value        1.000000    0.5122    0.6136          0.7606
## Neg Pred Value        0.942731    0.9043    0.8108          0.9425
## Prevalence            0.065502    0.1703    0.2707          0.4934
## Detection Rate        0.008734    0.0917    0.1179          0.4716
## Detection Prevalence  0.008734    0.1790    0.1921          0.6201
## Balanced Accuracy     0.566667    0.7166    0.6668          0.8313
```

```r
train.pc <- data.frame(npf.pcA2.withtest$x[1:458,1:14])
train.pc$class4 <- npf$class4

test.pc <- data.frame(npf.pcA2.withtest$x[459:(458+nrow(npf_test)),1:14])

ctrl <- trainControl(method = "repeatedcv",
                     number = 10,
                     repeats = 10,
                     classProbs = TRUE)
rFClass4 <- train(factor(class4) ~ .,
                    method="rf",
                    data=train.pc,
                    trControl=ctrl)

pred_test4 <- predict(rFClass4, newdata = test.pc)
probs_test4 <- predict(rFClass4, newdata = test.pc, type = "prob")
```

```r
length(which(pred_test4=="nonevent"))
```

```
## [1] 589
```

```r
length(which(pred_test4=="Ia"))
```

```
## [1] 18
```

```r
length(which(pred_test4=="Ib"))
```

```
## [1] 168
```

```r
length(which(pred_test4=="II"))
```

```
## [1] 190
```

```r
# Code to produce the answer-csv
# The first column "class4" in the answers.csv file is our
# prediction for the day, where class4 is Ia, Ib, II, or nonevent.
# The second column "p" is our prediction for probability Pr(class2=event)


# Creates the csv and adds the first line
# Change the string here to our guess of the accuracy
write.table(0.75,
             file="./answers.csv",
             append = F,
             sep=',',
             row.names=F,
             col.names=F)

# Write column names to the file
write.table(data.frame("class4","p"),
            file="./answers.csv",
            append = T,
            sep=',',
            row.names=F,
            col.names=F)

# testing testing
#setwd()
#probs_test4 <- data.frame(c(0.1,0.2,0.5,0.1,0.5),c(0.6,0.05,0.1,0.2,0.05),c(0.1,0.7,0.2,0.2,0.15),c(0.
#colnames(probs_test4)<- c("Ia","Ib","II","nonevent")

# Assume the class probabilities for each row are in probs_test4
classes_test4 <- colnames(probs_test4)[max.col(probs_test4, ties.method = "first")]

# Write the class predictions and probabilities
write.table(data.frame(classes_test4, (probs_test4$Ia+probs_test4$Ib+probs_test4$II)),
            file="./answers.csv",
            append = T,
            sep=',',
            row.names=F,
            col.names=F)
```