Machine learning models for cancer predictive analysis

Natalia

28 May 2019

```
data <- read.table("C://Users//Natalia//Desktop//ITMO//R//R project//cancer data//lympha//lymphography.
colnames(data) <- c("class", "lymphatics", "blaff", "blc", "bls", "pass", "extravasats", "regeneration"
View(data)
```

Analyse the dataset and tidy it up.

```
# Analyse the data - checking for values, NAs, data type.
summary(data)
```

```
##      class         lymphatics        blaff            blc       
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000  
##  Median :2.000   Median :3.000   Median :2.000   Median :1.000  
##  Mean   :2.449   Mean   :2.735   Mean   :1.551   Mean   :1.177  
##  3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:1.000  
##  Max.   :4.000   Max.   :4.000   Max.   :2.000   Max.   :2.000  
##       bls             pass        extravasats    regeneration  
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000  
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:1.000  
##  Median :1.000   Median :1.000   Median :2.00   Median :1.000  
##  Mean   :1.048   Mean   :1.245   Mean   :1.51   Mean   :1.068  
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:2.00   3rd Qu.:1.000  
##  Max.   :2.000   Max.   :2.000   Max.   :2.00   Max.   :2.000  
##      uptake         nodesdim       nodesenlar        chlym       
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:2.000  
##  Median :2.000   Median :1.000   Median :2.000   Median :2.000  
##  Mean   :1.701   Mean   :1.061   Mean   :2.476   Mean   :2.401  
##  3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:3.000   3rd Qu.:3.000  
##  Max.   :2.000   Max.   :3.000   Max.   :4.000   Max.   :3.000  
##      defect          chnode          chstru          forms      
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.00  
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:4.000   1st Qu.:2.00  
##  Median :3.000   Median :3.000   Median :5.000   Median :3.00  
##  Mean   :2.973   Mean   :2.796   Mean   :5.197   Mean   :2.34  
##  3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:8.000   3rd Qu.:3.00  
##  Max.   :4.000   Max.   :4.000   Max.   :8.000   Max.   :3.00  
##      disloc        exclusion       numbnodes     
##  Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000  
##  Median :2.000   Median :2.000   Median :2.000  
##  Mean   :1.667   Mean   :1.789   Mean   :2.605  
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:3.000  
##  Max.   :2.000   Max.   :2.000   Max.   :8.000
```

```r
str(data)
```

```
## 'data.frame':    147 obs. of  19 variables:
##  $ class       : int  2 3 3 2 2 2 2 3 3 2 ...
##  $ lymphatics  : int  3 3 3 3 2 2 3 2 2 2 ...
##  $ blaff       : int  2 2 1 1 1 2 2 2 1 2 ...
##  $ blc         : int  1 2 1 1 1 1 1 1 1 2 ...
##  $ bls         : int  1 2 1 1 1 1 1 1 1 1 ...
##  $ pass        : int  2 2 1 1 1 1 1 1 1 2 ...
##  $ extravasats : int  2 2 2 1 1 1 2 1 1 2 ...
##  $ regeneration: int  1 2 1 1 1 1 1 1 1 1 ...
##  $ uptake      : int  2 2 2 1 2 2 2 2 2 2 ...
##  $ nodesdim    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ nodesenlar  : int  3 4 3 2 3 2 2 3 2 3 ...
##  $ chlym       : int  3 3 3 2 3 3 2 2 2 2 ...
##  $ defect      : int  2 3 4 4 3 2 2 2 3 4 ...
##  $ chnode      : int  3 4 4 3 3 3 2 2 3 3 ...
##  $ chstru      : int  4 8 4 5 6 8 1 8 5 5 ...
##  $ forms       : int  2 3 3 1 3 2 3 3 3 1 ...
##  $ disloc      : int  2 2 1 2 1 1 1 1 1 2 ...
##  $ exclusion   : int  2 2 2 2 2 1 1 2 1 2 ...
##  $ numbnodes   : int  2 7 6 1 4 1 1 5 2 3 ...
```

```r
#Comments for dataset:

writeLines(readLines("C://Users//Natalia//Desktop//ITMO//R//R project//cancer data//lympha//lymph.txt"))
```

```
## Warning in readLines("C://Users//Natalia//Desktop//ITMO//R//R project//
## cancer data//lympha//lymph.txt"): incomplete final line found on 'C://
## Users//Natalia//Desktop//ITMO//R//R project//cancer data//lympha//
## lymph.txt'
```

```
## --- NOTE: All attribute values in the database have been entered as numeric values corresponding to t
## 1. class: normal find, metastases, malign lymph, fibrosis
## 2. lymphatics: normal, arched, deformed, displaced
## 3. block of affere: no, yes
## 4. bl. of lymph. c: no, yes
## 5. bl. of lymph. s: no, yes
## 6. by pass: no, yes
## 7. extravasates: no, yes
## 8. regeneration of: no, yes
## 9. early uptake in: no, yes
## 10. lym.nodes dimin: 0-3
## 11. lym.nodes enlar: 1-4
## 12. changes in lym.: bean, oval, round
## 13. defect in node: no, lacunar, lac. marginal, lac. central
## 14. changes in node: no, lacunar, lac. margin, lac. central
## 15. changes in stru: no, grainy, drop-like, coarse, diluted, reticular, stripped, faint,
## 16. special forms: no, chalices, vesicles
## 17. dislocation of: no, yes
## 18. exclusion of no: no, yes
## 19. no. of nodes in: 0-9, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, >=70
```

```r
#Replace class "numeric" specification for strings abbreviation:

data$Class <- ifelse(data$class== 1, "normal", ifelse(data$class == 2, "metastases", ifelse(data$class =
```

```
data$class <- NULL
head(data)
```

```
##   lymphatics blaff blc bls pass extravasats regeneration uptake nodesdim
## 1          3     2   1   1    2           2            1      2        1
## 2          3     2   2   2    2           2            2      2        1
## 3          3     1   1   1    1           2            1      2        1
## 4          3     1   1   1    1           1            1      1        1
## 5          2     1   1   1    1           1            1      2        1
## 6          2     2   1   1    1           1            1      2        1
##   nodesenlar chlym defect chnode chstru forms disloc exclusion numbnodes
## 1          3     3      2      3      4     2      2         2         2
## 2          4     3      3      4      8     3      2         2         7
## 3          3     3      4      4      4     3      1         2         6
## 4          2     2      4      3      5     1      2         2         1
## 5          3     3      3      3      6     3      1         2         4
## 6          2     3      2      3      8     2      1         1         1
##        Class
## 1 metastases
## 2     malign
## 3     malign
## 4 metastases
## 5 metastases
## 6 metastases
```

```
dim(data)
```

```
## [1] 147  19
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------------- tidyverse 1.2
```

```
## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------- tidyverse_conflicts
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
map_int(data, function(.x) sum(is.na(.x)))
```

```
##   lymphatics        blaff          blc          bls         pass
##            0            0            0            0            0
##  extravasats regeneration       uptake     nodesdim   nodesenlar
##            0            0            0            0            0
##        chlym       defect       chnode       chstru        forms
##            0            0            0            0            0
##       disloc    exclusion    numbnodes        Class
##            0            0            0            0
```

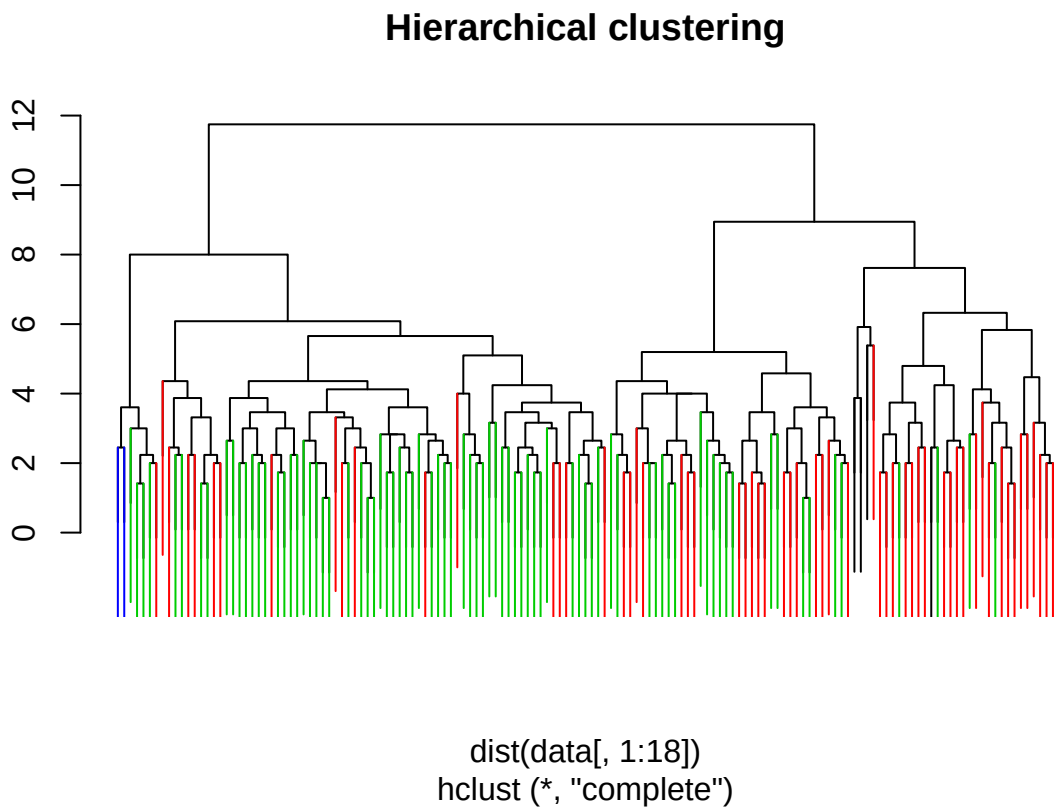
```
# Data type "Class" as factor:

data <- as.data.frame(data, stringsAsFactors=T)
data$Class <- as.factor(data$Class)
head(data)
```

```
##   lymphatics blaff blc bls pass extravasats regeneration uptake nodesdim
## 1          3     2   1   1    2           2            1      2        1
## 2          3     2   2   2    2           2            2      2        1
## 3          3     1   1   1    1           2            1      2        1
## 4          3     1   1   1    1           1            1      1        1
## 5          2     1   1   1    1           1            1      2        1
## 6          2     2   1   1    1           1            1      2        1
##   nodesenlar chlym defect chnode chstru forms disloc exclusion numbnodes
## 1          3     3      2      3      4     2      2         2          2
## 2          4     3      3      4      8     3      2         2          7
## 3          3     3      4      4      4     3      1         2          6
## 4          2     2      4      3      5     1      2         2          1
## 5          3     3      3      3      6     3      1         2          4
## 6          2     3      2      3      8     2      1         1          1
##        Class
## 1 metastases
## 2     malign
## 3     malign
## 4 metastases
## 5 metastases
## 6 metastases
```

DATA EXPLORATION

Hierarchical clustering

```
library(sparcl)
hc <- hclust(dist(data[,1:18]), method = "complete")
ColorDendrogram(hc,y=data$Class, main = "Hierarchical clustering", branchlength=5)
```



Most of the "green class"" and "red class"" samples are grouped together with two prevalent clusters.

K-means clustering

```
fit <- kmeans(data[,c(1:18)], 2)
names(fit)
```

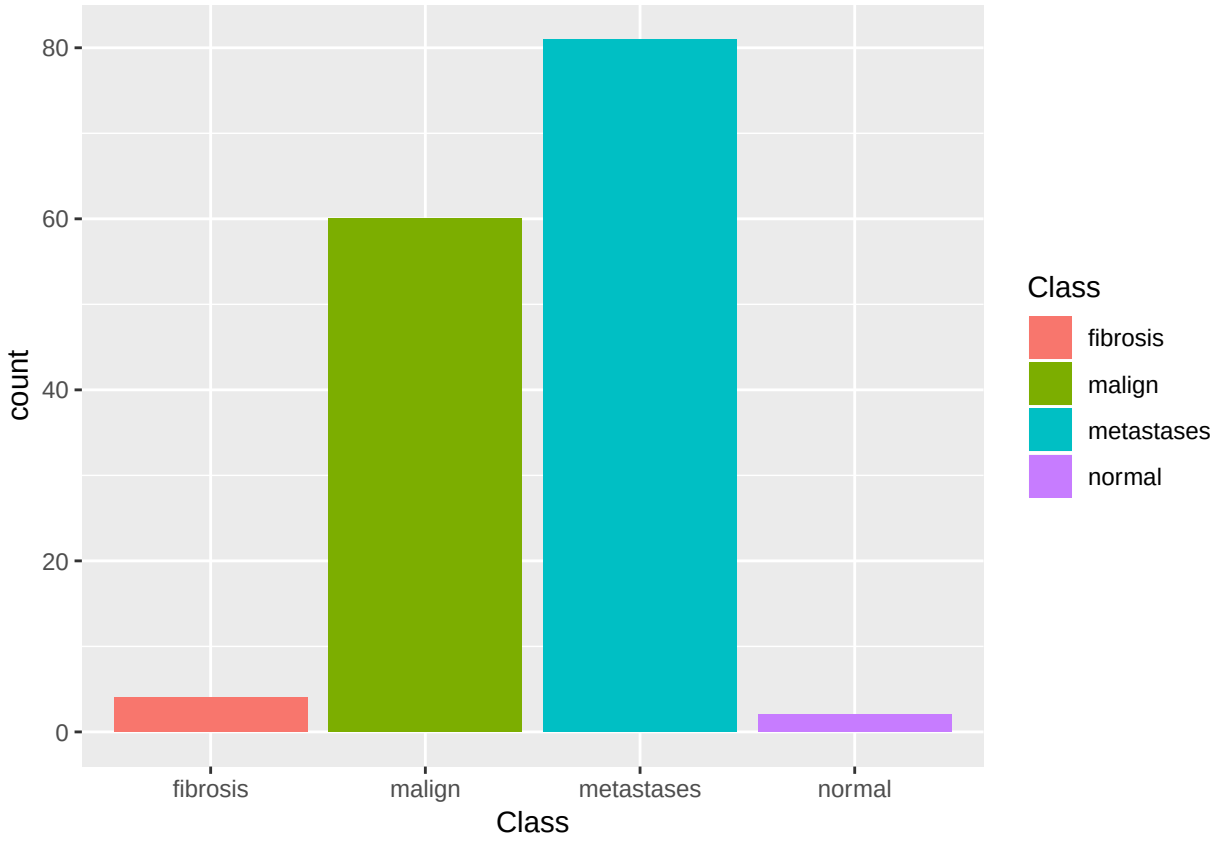
```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
#k-means did a fairly good job
table(data.frame(fit$cluster,data[,19]))
```

```
##             data...19.
## fit.cluster fibrosis malign metastases normal
##           1        4     45         24      0
##           2        0     15         57      2
```

Response variable for classification

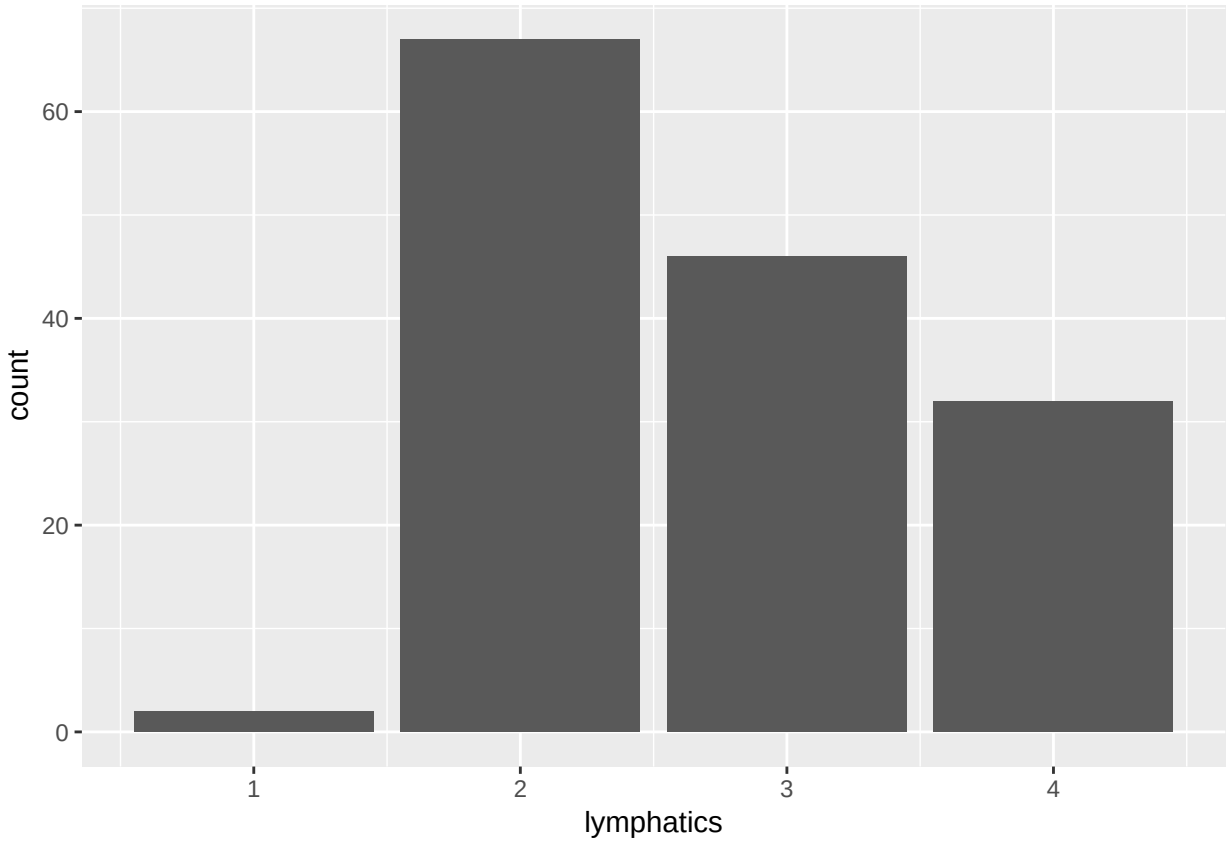
```
library(ggplot2)

ggplot(data, aes(x = Class, fill = Class)) +
  geom_bar()
```



Response variable for regression

```
ggplot(data, aes(x = lymphatics)) +
  geom_histogram(binwidth = 1, stat = "count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



Principal Component Analysis

```
library(pcaGoPromoter)
```

```
## Loading required package: ellipse
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
## Loading required package: Biostrings
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:dplyr':
```

```
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:tidyr':
##
##     expand

## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:purrr':
##
##     reduce

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: XVector

##
## Attaching package: 'XVector'

## The following object is masked from 'package:purrr':
##
##     compact
```

```
##
## Attaching package: 'Biostrings'
```

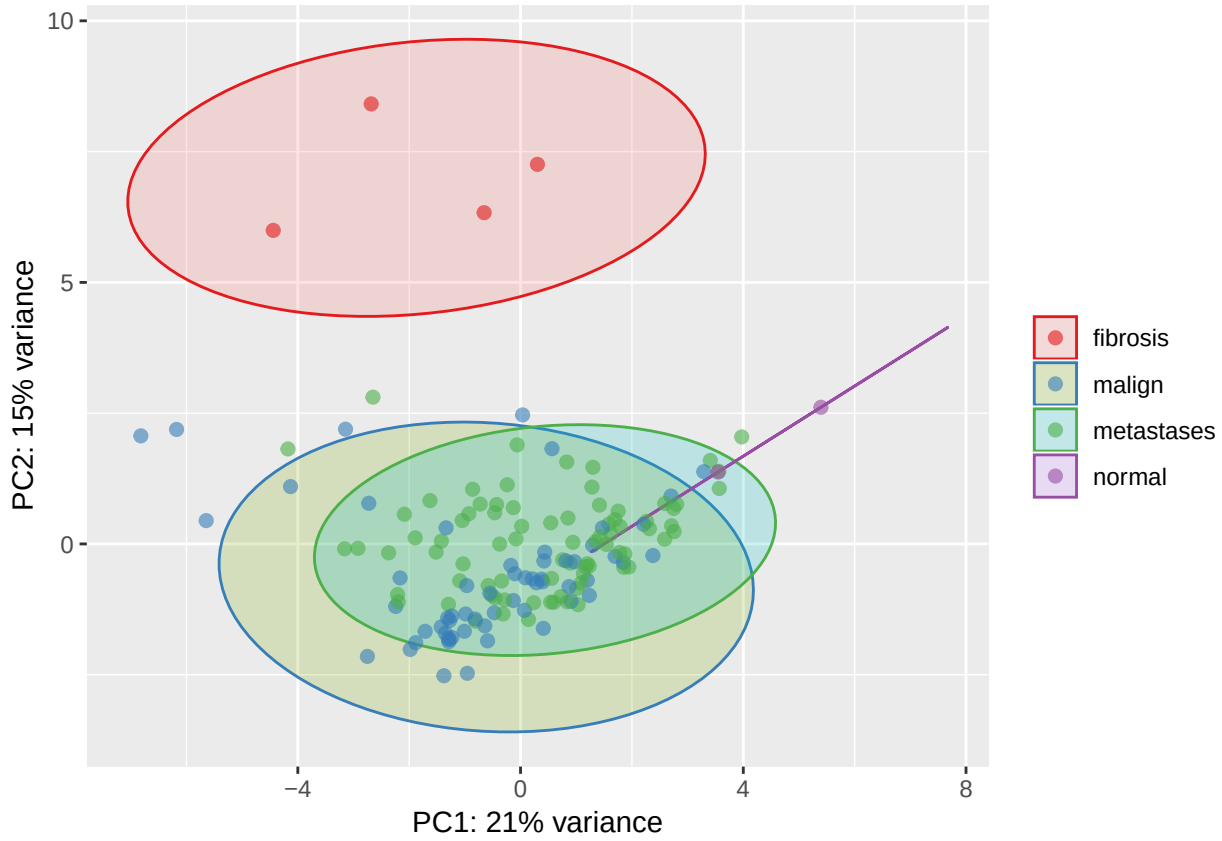
```
## The following object is masked from 'package:base':
##
##     strsplit
```

```
library(ellipse)
```

```
data <-na.omit(data)
```

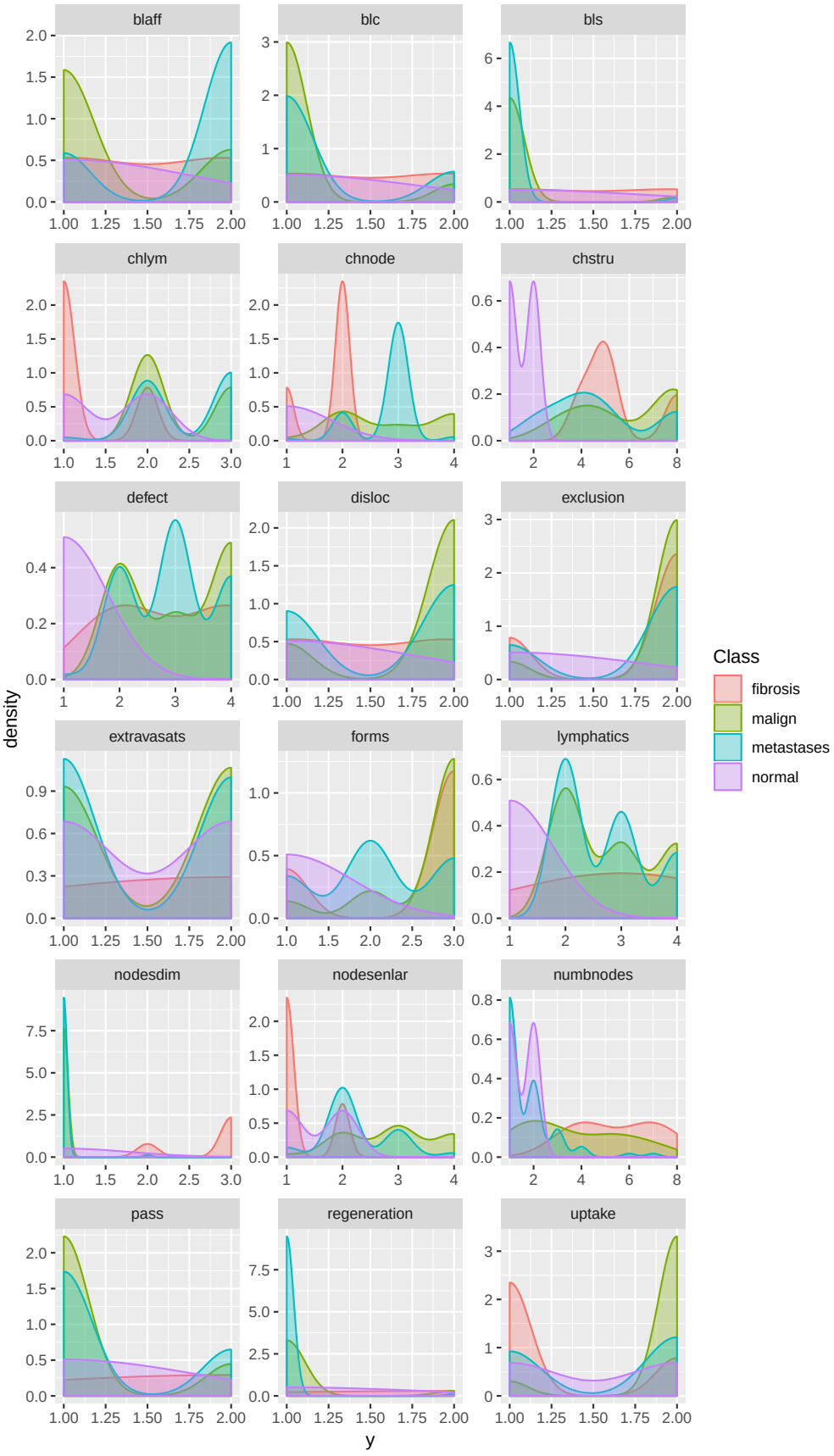
```
# perform pca and extract scores:

pcaOutput <- pca(t(data[,1:18]), printDropped = FALSE, scale = TRUE, center = TRUE)
pcaOutput2 <- as.data.frame(pcaOutput$scores)
```

```
# define groups for plotting:

pcaOutput2$groups <- data$Class

centroids <- aggregate(cbind(PC1, PC2) ~ groups, pcaOutput2, mean)

conf.rgn  <- do.call(rbind, lapply(unique(pcaOutput2$groups), function(t)
  data.frame(groups = as.character(t),
             ellipse(cov(pcaOutput2[pcaOutput2$groups == t, 1:2]),
                   centre = as.matrix(centroids[centroids$groups == t, 2:3]),
                   level = 0.95),
             stringsAsFactors = FALSE)))


#Plot PCA with variance %:

ggplot(data = pcaOutput2, aes(x = PC1, y = PC2, group = groups, color = groups)) +
    geom_polygon(data = conf.rgn, aes(fill = groups), alpha = 0.2) +
    geom_point(size = 2, alpha = 0.6) +
    scale_color_brewer(palette = "Set1") +
    labs(color = "",
         fill = "",
         x = paste0("PC1: ", round(pcaOutput$pov[1], digits = 2) * 100, "% variance"),
         y = paste0("PC2: ", round(pcaOutput$pov[2], digits = 2) * 100, "% variance"))
```



Features

```
library(tidyr)

gather(data, x, y, lymphatics:numbnodes) %>%
  ggplot(aes(x = y, color = Class, fill = Class)) +
    geom_density(alpha = 0.3) +
    facet_wrap( ~ x, scales = "free", ncol = 3)
```


blaff
blc
bls
chlym
chnode
chstru
defect
disloc
exclusion
extravasats
forms
lymphatics
nodesdim
nodesenlar
numbnodes
pass
regeneration
uptake
density
y
Class
fibrosis
malign
metastases
normal

MACHINE LEARNING PACKAGES FOR R

caret

```
# configure multicore
library(doParallel)
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loading required package: iterators
```

```
cl <- makeCluster(detectCores())
registerDoParallel(cl)

library(caret)
```

```
## Loading required package: lattice
```

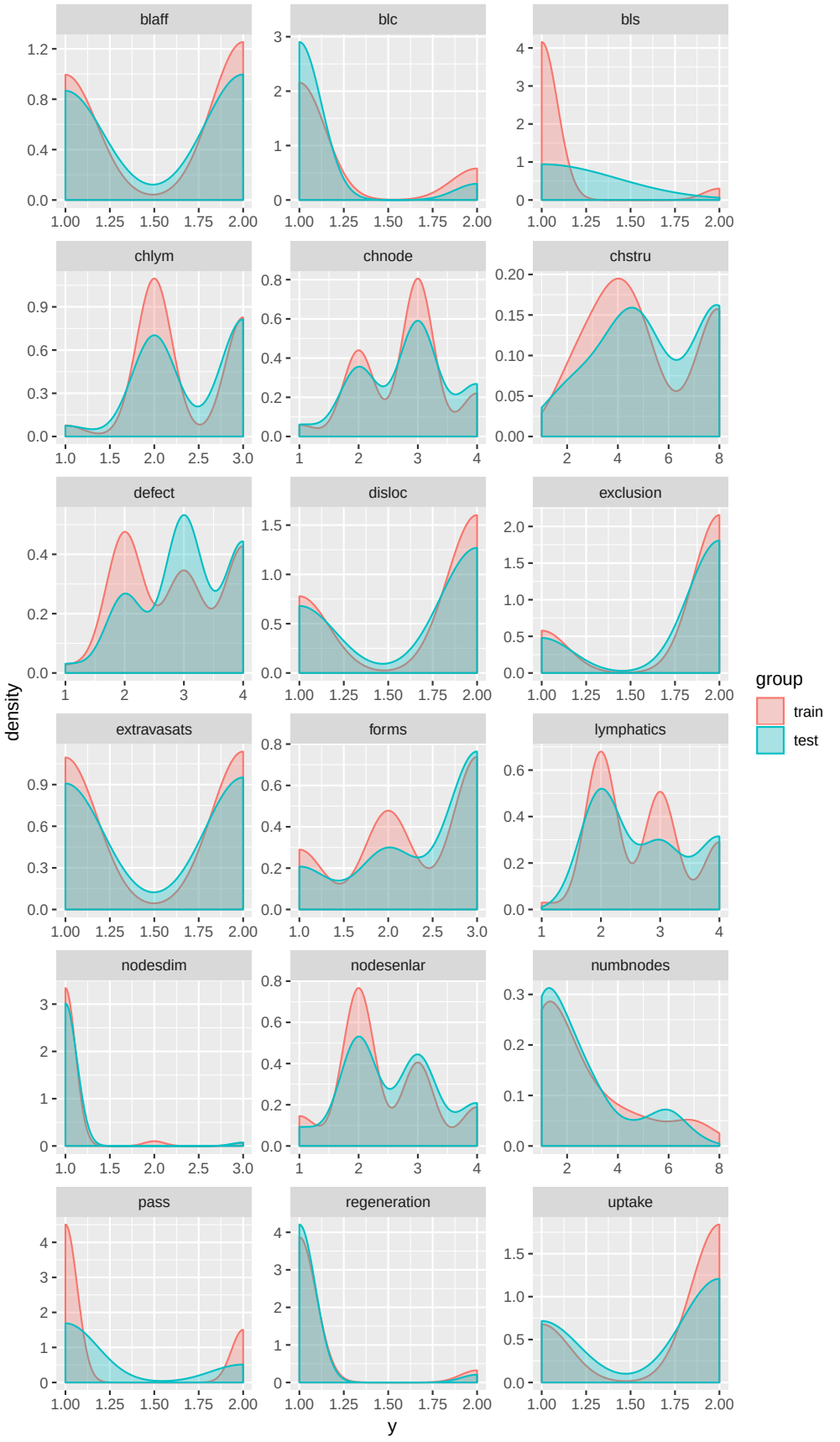
```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

Training, validation and test data

```
set.seed(42)
index <- createDataPartition(data$Class, p = 0.7, list = FALSE)
train_data <- data[index, ]
test_data  <- data[-index, ]
```

```
library(dplyr)

rbind(data.frame(group = "train", train_data),
      data.frame(group = "test", test_data)) %>%
  gather(x, y, lymphatics:numbnodes) %>%
  ggplot(aes(x = y, color = group, fill = group)) +
    geom_density(alpha = 0.3) +
    facet_wrap( ~ x, scales = "free", ncol = 3)
```


blaff
blc
bls
chlym
chnode
chstru
defect
disloc
exclusion
extravasats
forms
lymphatics
nodesdim
nodesenlar
numbnodes
pass
regeneration
uptake
density
y
group
train
test

Regression

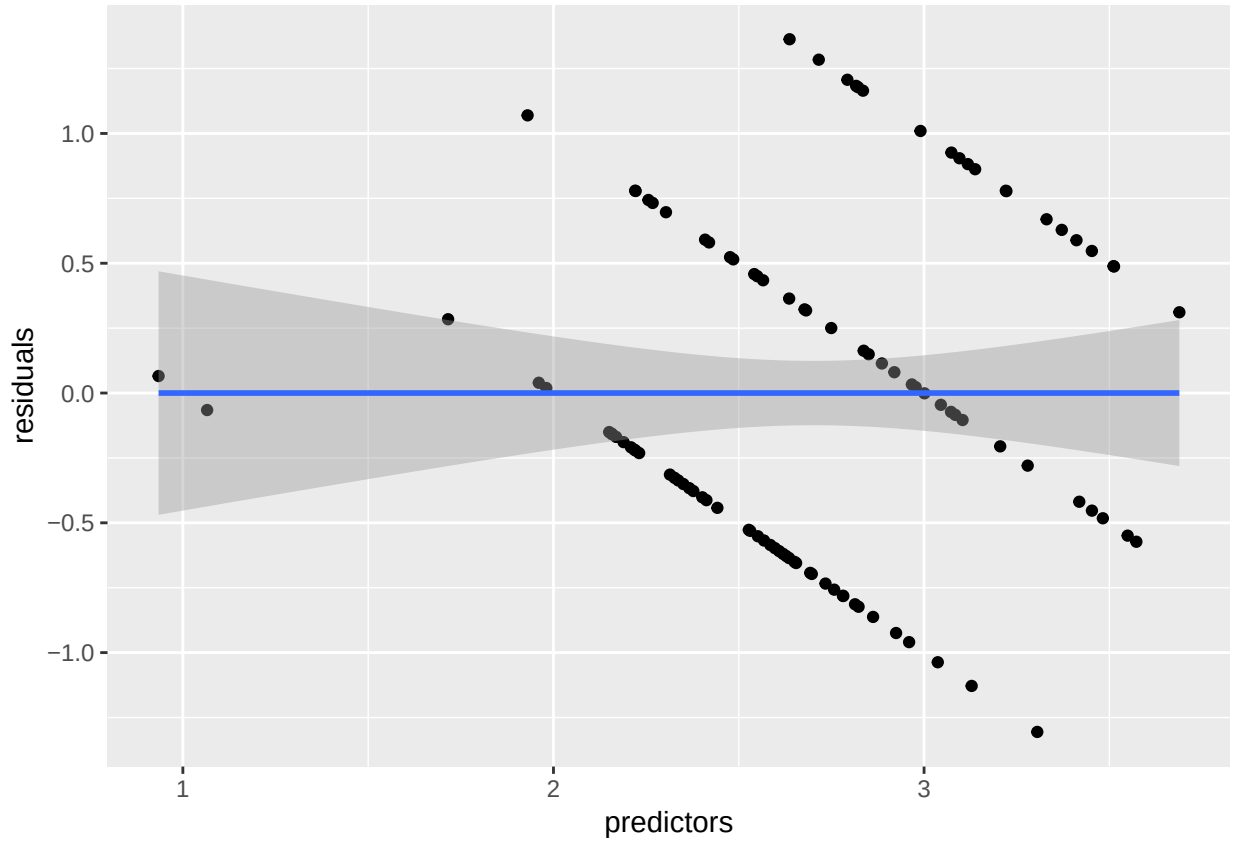
```
set.seed(42)
model_glm <- caret::train(lymphatics ~ .,
                          data = train_data,
                          method = "glm",
                          preProcess = c("scale", "center"),
                          trControl = trainControl(method = "repeatedcv",
                                                   number = 10,
                                                   repeats = 10,
                                                   savePredictions = TRUE,
                                                   verboseIter = FALSE))
```

```
model_glm
```

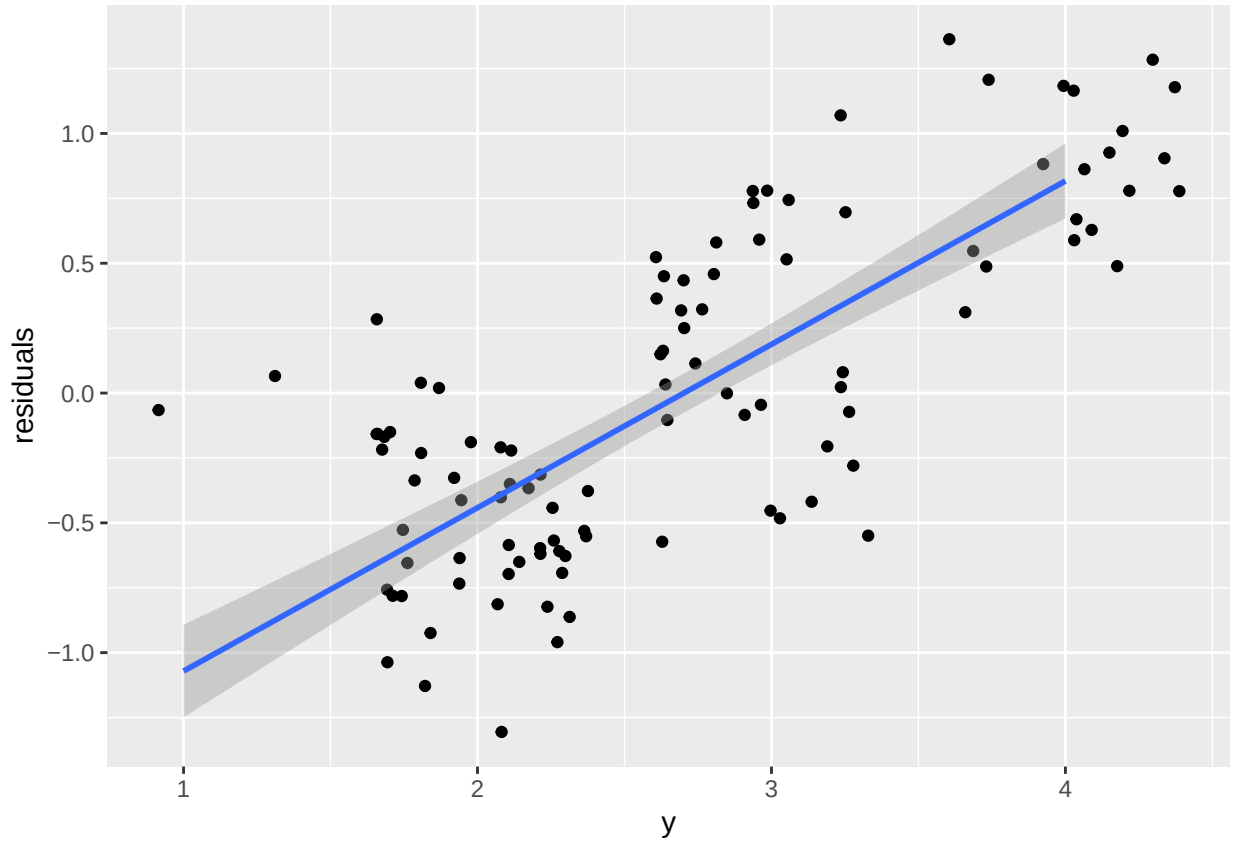
```
## Generalized Linear Model
##
## 104 samples
##  18 predictor
##
## Pre-processing: scaled (20), centered (20)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 95, 94, 93, 93, 94, 93, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.8024945  0.1863786  0.6800657
```

```
predictions <- predict(model_glm, test_data)
```

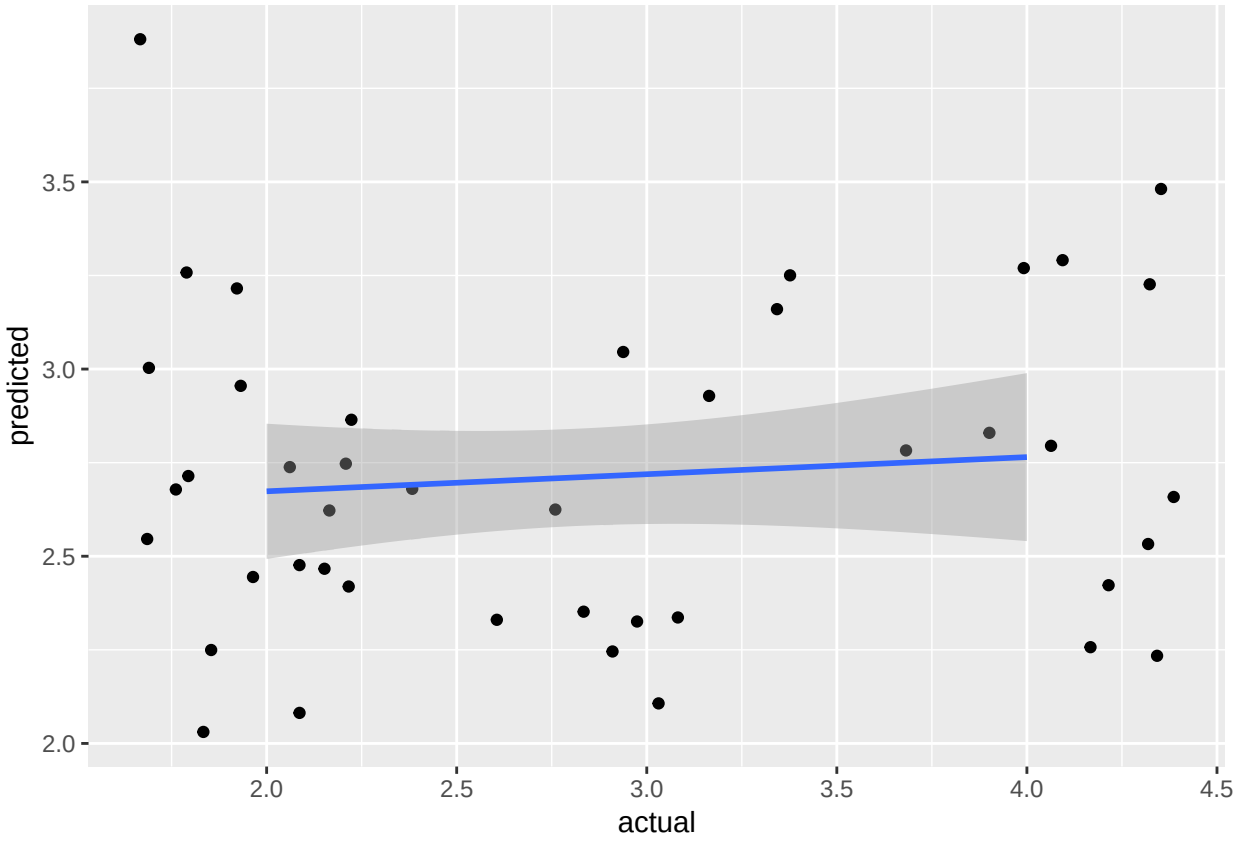
```
# model_glm$finalModel$linear.predictors == model_glm$finalModel$fitted.values
data.frame(residuals = resid(model_glm),
           predictors = model_glm$finalModel$linear.predictors) %>%
  ggplot(aes(x = predictors, y = residuals)) +
    geom_jitter() +
    geom_smooth(method = "lm")
```




```
y <- train_data$lymphatics
data.frame(residuals = resid(model_glm),
           y = model_glm$finalModel$y) %>%
  ggplot(aes(x = y, y = residuals)) +
    geom_jitter() +
    geom_smooth(method = "lm")
```



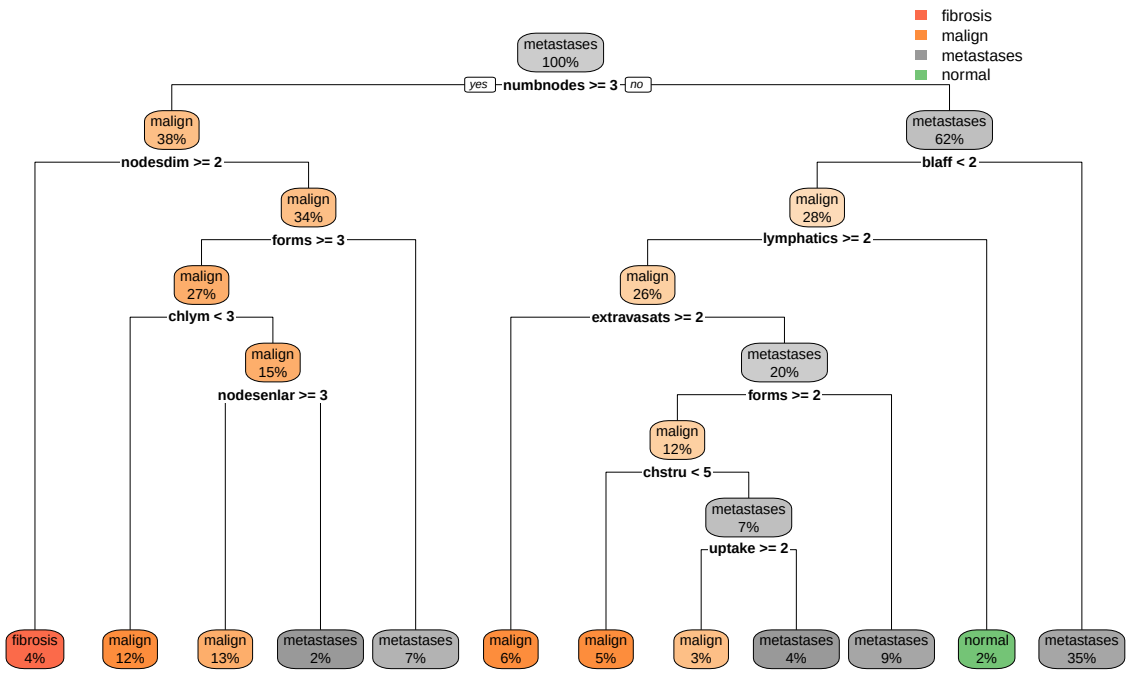

```
data.frame(actual = test_data$lymphatics,
           predicted = predictions) %>%
  ggplot(aes(x = actual, y = predicted)) +
    geom_jitter() +
    geom_smooth(method = "lm")
```



CLASSIFICATION

Decision trees

```
library(rpart)
library(rpart.plot)

set.seed(42)
fit <- rpart(Class ~ .,
             data = train_data,
             method = "class",
             control = rpart.control(xval = 10,
                                     minbucket = 2,
                                     cp = 0),
              parms = list(split = "information"))

rpart.plot(fit, extra = 100)
```



RANDOM FORESTS

```
#Random Forests predictions are based on the generation of
#multiple classification trees.
#They can be used for both, classification and regression tasks.
#Here, it is classification task.
```

```
set.seed(42)
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:BiocGenerics':
##
##     combine
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
model_rf <- caret::train(Class ~ .,
                         data = train_data,
                         method = "rf",
```

```
                         preProcess = c("scale", "center"),
                         trControl = trainControl(method = "repeatedcv",
                                                  number = 10,
                                                  repeats = 10,
                                                  savePredictions = TRUE,
                                                  verboseIter = FALSE))
```

```
#When savePredictions = TRUE is specified,
#can access the cross-validation resuls with model_rf$pred.

model_rf$finalModel$confusion
```

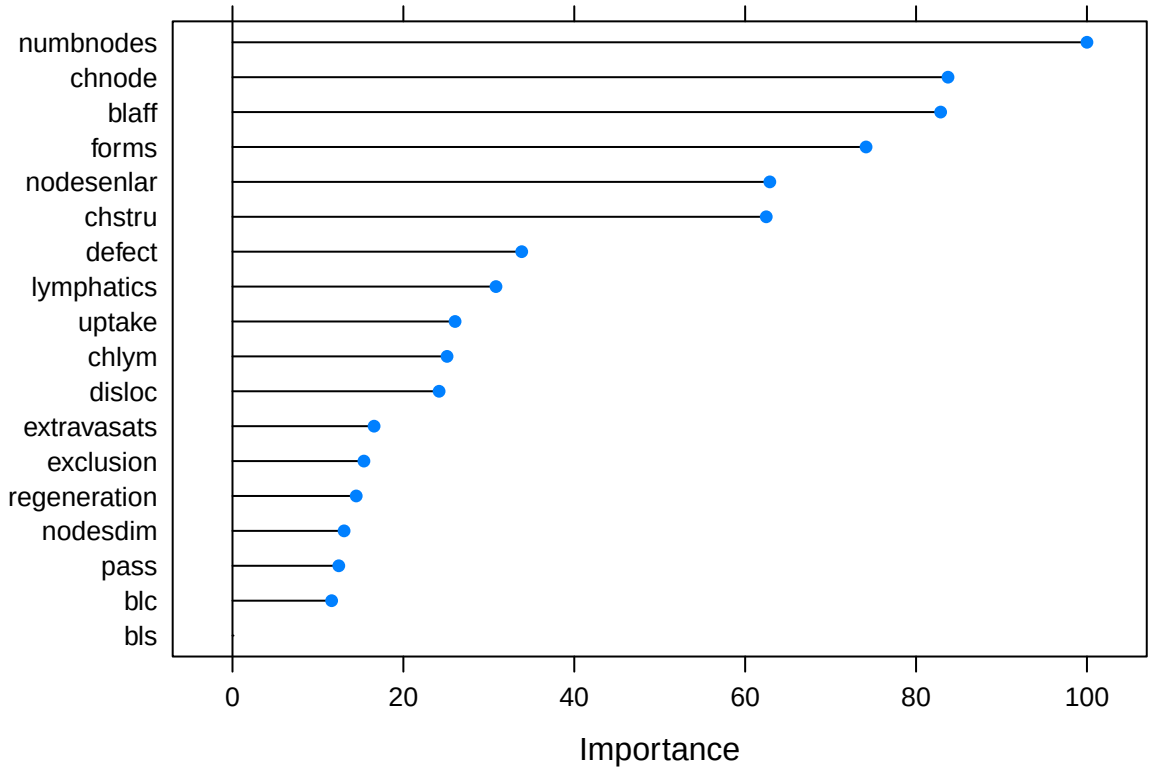
```
##            fibrosis malign metastases normal class.error
## fibrosis          0      2          1      0  1.00000000
## malign            0     30         12      0  0.28571429
## metastases        0      3         54      0  0.05263158
## normal            0      0          2      0  1.00000000
```

Feature Importance

```
imp <- model_rf$finalModel$importance
imp[order(imp, decreasing = TRUE), ]
```

```
##    numbnodes       chnode        blaff        forms   nodesenlar
##    5.7377787    4.8842513    4.8391096    4.3808429    3.7895828
##       chstru       defect   lymphatics       uptake        chlym
##    3.7677608    2.2654762    2.1074857    1.8560781    1.8068611
##       disloc  extravasats    exclusion regeneration     nodesdim
##    1.7579941    1.3585474    1.2961876    1.2488820    1.1743472
##         pass          blc          bls
##    1.1416776    1.0977760    0.4887381
```

```
# estimate variable importance
importance <- varImp(model_rf, scale = TRUE)
plot(importance)
```



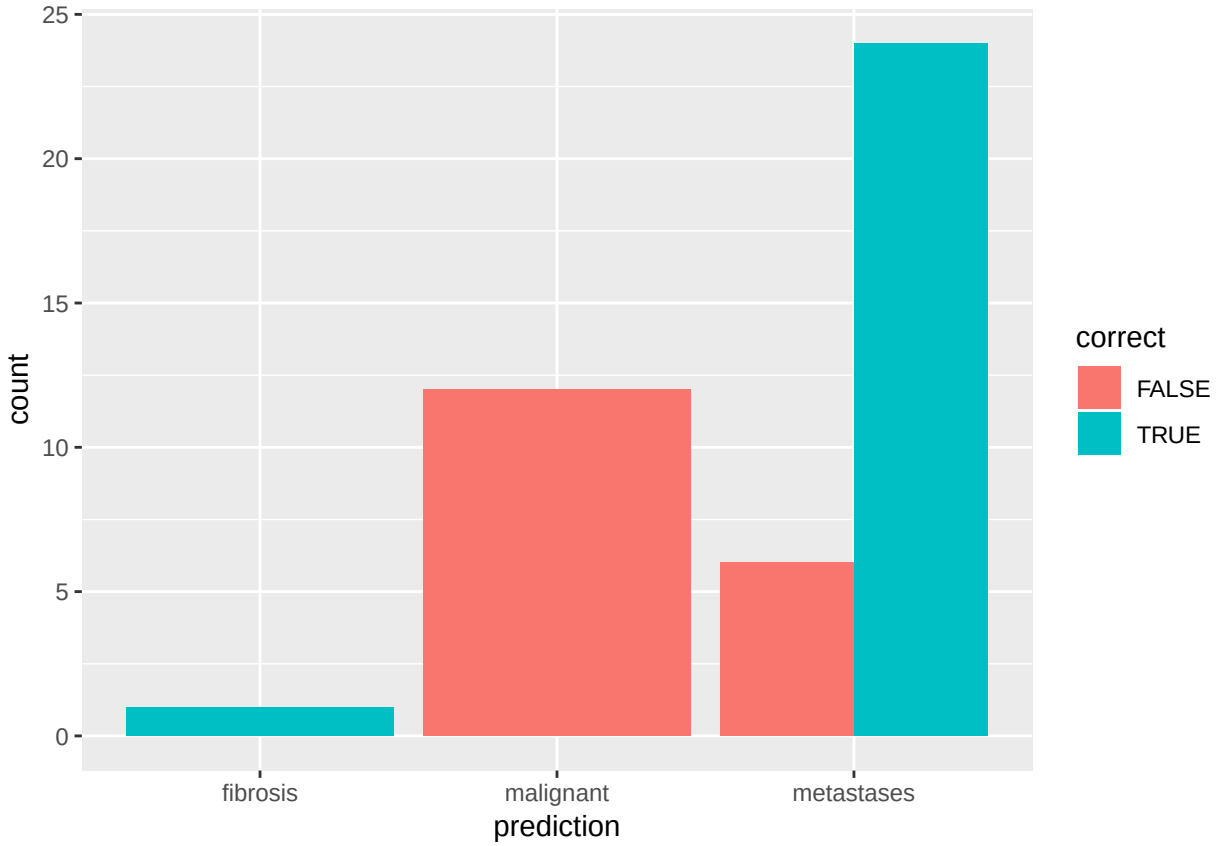
Predicting test data

```
confusionMatrix(predict(model_rf, test_data), test_data$Class)
```

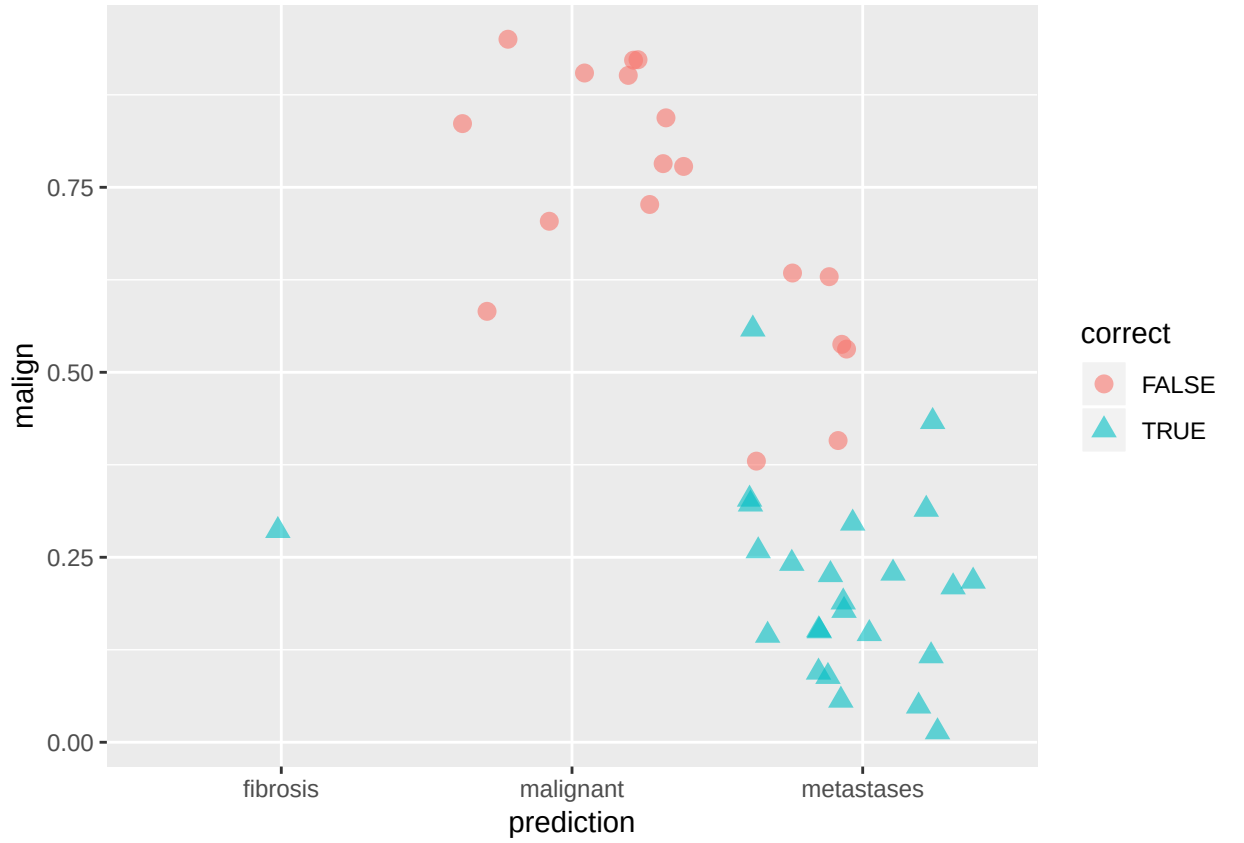
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   fibrosis malign metastases normal
##   fibrosis          1      0          0      0
##   malign            0     16          1      0
##   metastases        0      2         23      0
##   normal            0      0          0      0
##
## Overall Statistics
##
##                Accuracy : 0.9302
##                  95% CI : (0.8094, 0.9854)
##     No Information Rate : 0.5581
##     P-Value [Acc > NIR] : 8.661e-08
##
##                   Kappa : 0.8631
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: fibrosis Class: malign Class: metastases
## Sensitivity                  1.00000        0.8889            0.9583
## Specificity                  1.00000        0.9600            0.8947
## Pos Pred Value               1.00000        0.9412            0.9200
## Neg Pred Value               1.00000        0.9231            0.9444
## Prevalence                   0.02326        0.4186            0.5581
## Detection Rate               0.02326        0.3721            0.5349
## Detection Prevalence         0.02326        0.3953            0.5814
## Balanced Accuracy            1.00000        0.9244            0.9265
##                      Class: normal
## Sensitivity                     NA
## Specificity                      1
## Pos Pred Value                  NA
## Neg Pred Value                  NA
## Prevalence                       0
## Detection Rate                   0
## Detection Prevalence             0
## Balanced Accuracy               NA
```

```r
results <- data.frame(actual = test_data$Class,
                      predict(model_rf, test_data, type = "prob"))

results$prediction <- ifelse(results$metastases > 0.3, "metastases",
                             ifelse(results$malign > 0.3, "malignant",
                                    ifelse(results$fibrosis > 0.3, "fibrosis", NA)))

results$correct <- ifelse(results$actual == results$prediction, TRUE, FALSE)

ggplot(results, aes(x = prediction, fill = correct)) +
  geom_bar(position = "dodge")
```




```
ggplot(results, aes(x = prediction, y = malign, color = correct, shape = correct)) +
  geom_jitter(size = 3, alpha = 0.6)
```



EXTREME GRADIENT BOOSTING.

Extreme gradient boosting (XGBoost) is a faster and improved implementation of gradient boosting for supervised learning.

```
#XGBoost is a tree ensemble model, which means the sum of predictions
#from a set of classification and regression trees (CART).
#In that, XGBoost is similar to Random Forests but it uses a different approach
#to model training: it uses a combination of "weak" functions during iteration process,
#for each next iteration step, the model learns using the "mistakes" data of previous steps.
```

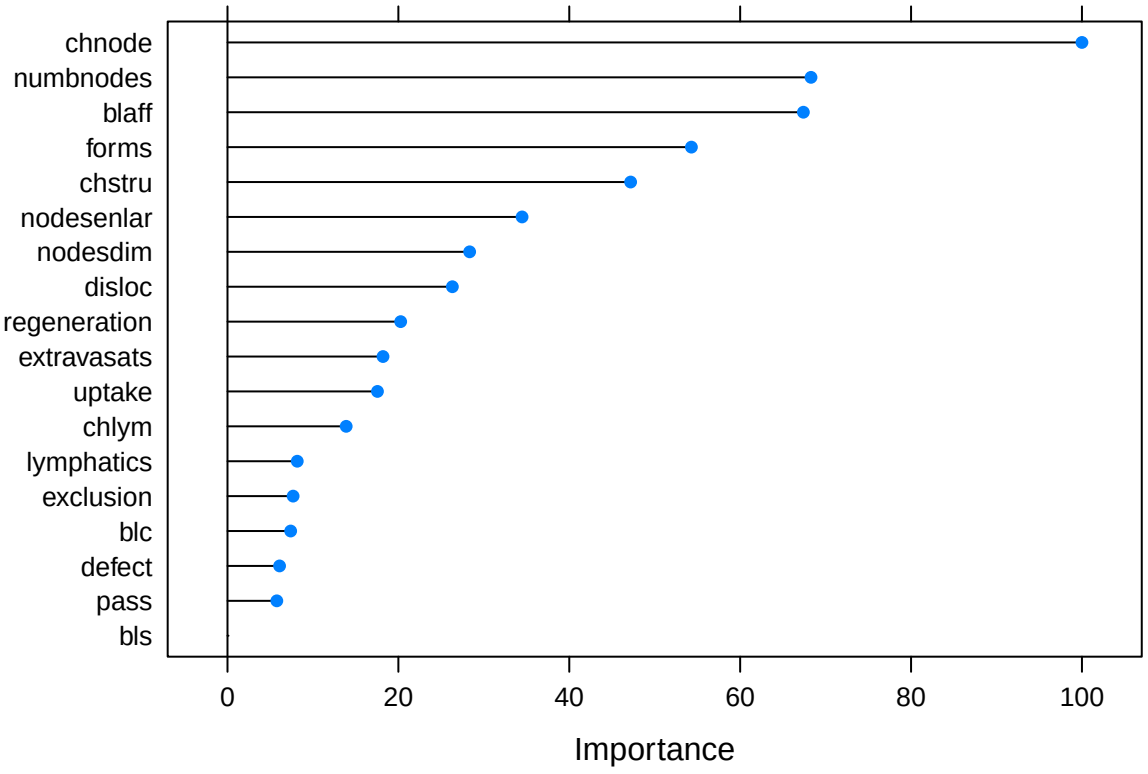
```
set.seed(42)
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
## The following object is masked from 'package:XVector':
##
##     slice
## The following object is masked from 'package:IRanges':
##
##     slice
## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
model_xgb <- caret::train(Class ~ .,
                          data = train_data,
                          method = "xgbTree",
                          preProcess = c("scale", "center"),
                          trControl = trainControl(method = "repeatedcv",
                                                   number = 10,
                                                   repeats = 10,
                                                   savePredictions = TRUE,
                                                   verboseIter = FALSE))
```

Feature Importance

```r
importance <- varImp(model_xgb, scale = TRUE)
plot(importance)
```



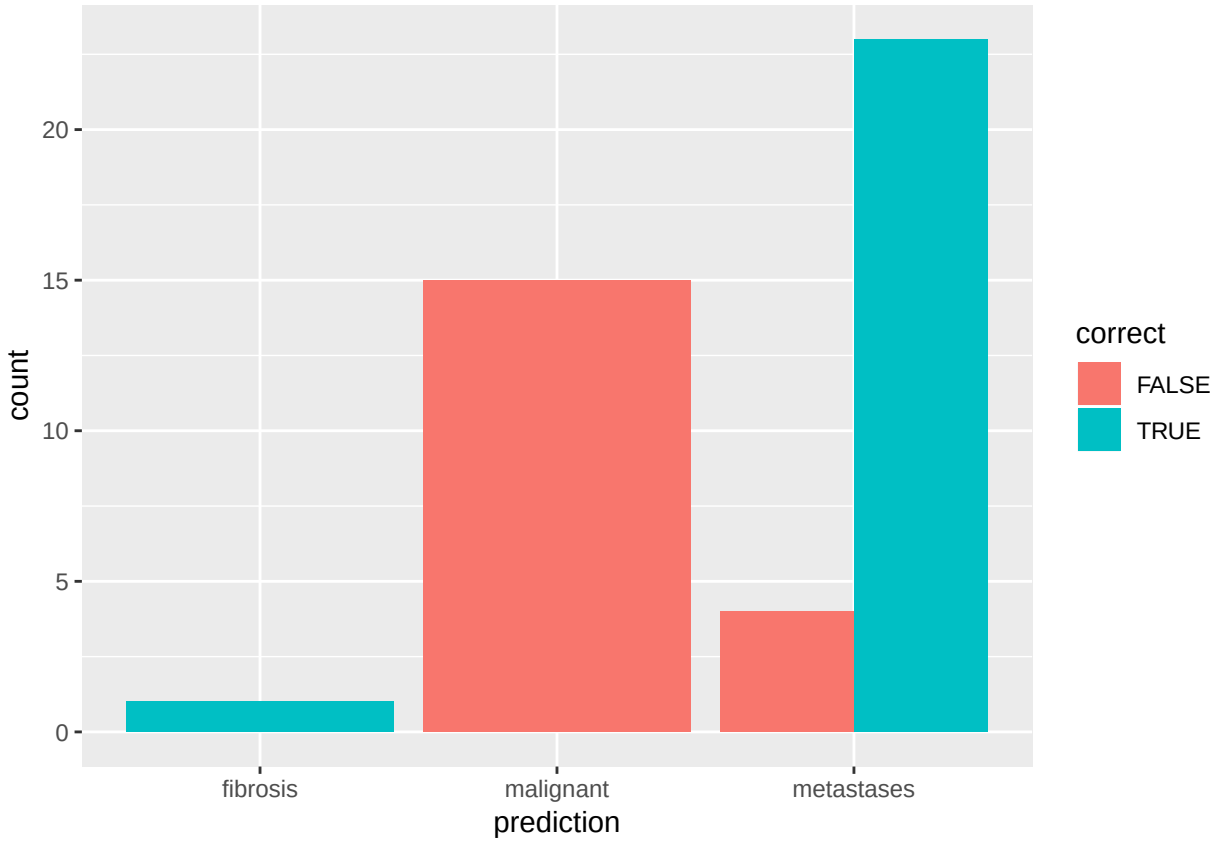
#Predicting test data

```r
confusionMatrix(predict(model_xgb, test_data), test_data$Class)
```

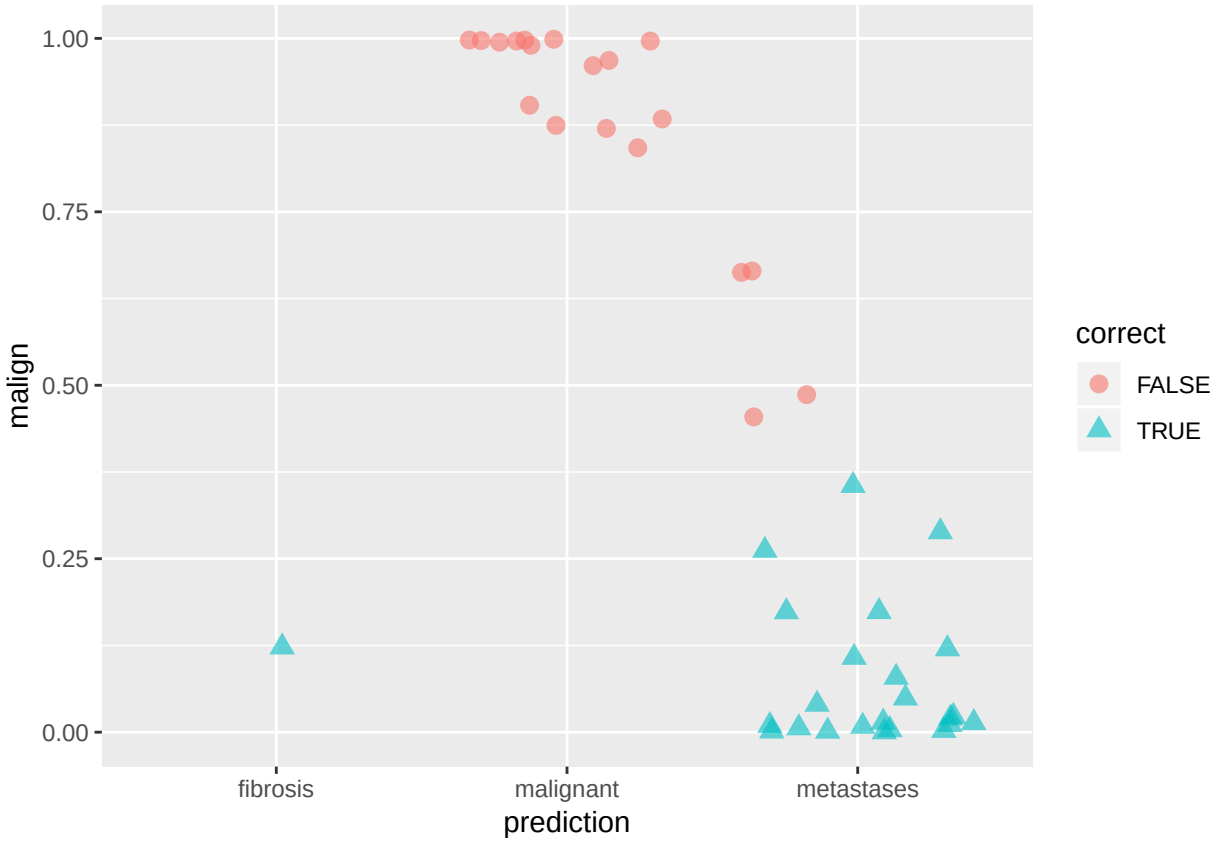
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   fibrosis malign metastases normal
##   fibrosis          1      0          0      0
##   malign            0     17          1      0
##   metastases        0      1         23      0
```

```
##   normal            0      0          0      0
##
## Overall Statistics
##
##                Accuracy : 0.9535
##                  95% CI : (0.8419, 0.9943)
##     No Information Rate : 0.5581
##     P-Value [Acc > NIR] : 7.741e-09
##
##                   Kappa : 0.9093
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: fibrosis Class: malign Class: metastases
## Sensitivity                  1.00000        0.9444            0.9583
## Specificity                  1.00000        0.9600            0.9474
## Pos Pred Value               1.00000        0.9444            0.9583
## Neg Pred Value               1.00000        0.9600            0.9474
## Prevalence                   0.02326        0.4186            0.5581
## Detection Rate               0.02326        0.3953            0.5349
## Detection Prevalence         0.02326        0.4186            0.5581
## Balanced Accuracy            1.00000        0.9522            0.9529
##                      Class: normal
## Sensitivity                     NA
## Specificity                      1
## Pos Pred Value                  NA
## Neg Pred Value                  NA
## Prevalence                       0
## Detection Rate                   0
## Detection Prevalence             0
## Balanced Accuracy               NA
```

```r
results <- data.frame(actual = test_data$Class,
                      predict(model_xgb, test_data, type = "prob"))

results$prediction <- ifelse(results$metastases > 0.3, "metastases",
                             ifelse(results$malign > 0.3, "malignant",
                                    ifelse(results$fibrosis > 0.3, "fibrosis", NA)))

results$correct <- ifelse(results$actual == results$prediction, TRUE, FALSE)

ggplot(results, aes(x = prediction, fill = correct)) +
  geom_bar(position = "dodge")
```




```
ggplot(results, aes(x = prediction, y = malign, color = correct, shape = correct)) +
  geom_jitter(size = 3, alpha = 0.6)
```



#FEATURE SELECTION

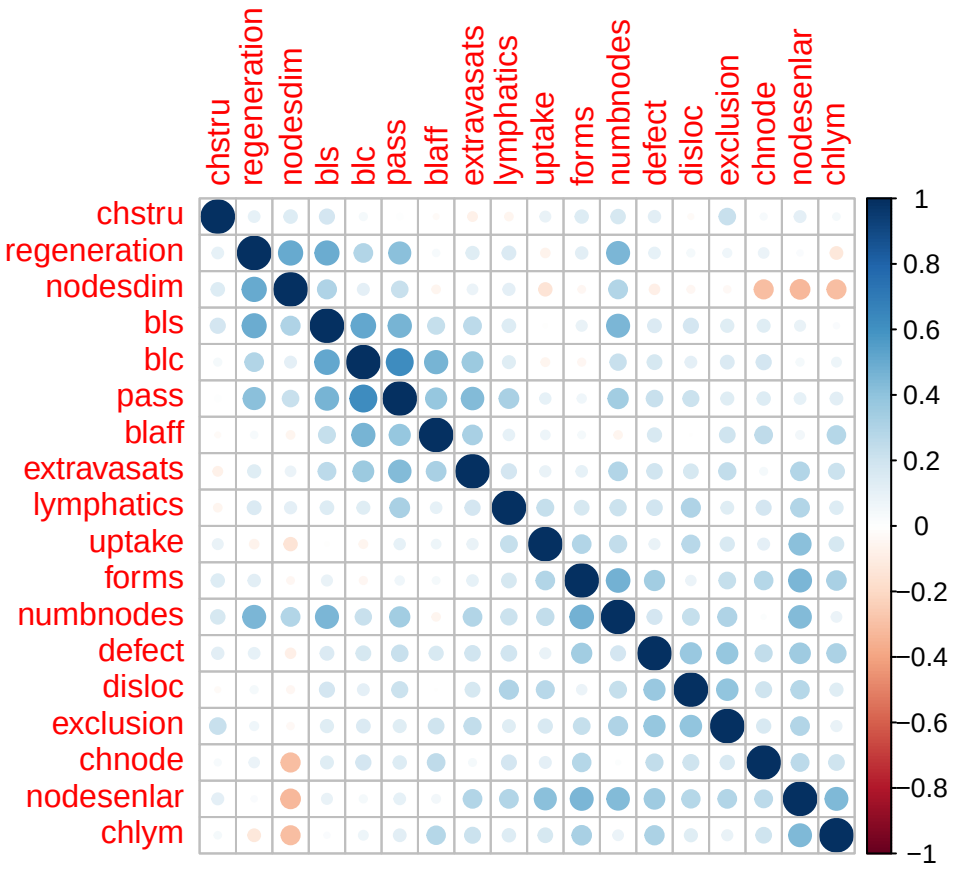
Performing feature selection on the whole dataset would lead to prediction bias, we therefore need to run the whole modeling process on the training data alone!

Correlation

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
# calculate correlation matrix
corMatMy <- cor(train_data[,1:18])
corrplot(corMatMy, order = "hclust")
```




```
#Apply correlation filter at 0.70:
highlyCor <- colnames(train_data[, -1])[findCorrelation(corMatMy, cutoff = 0.6, verbose = TRUE)]
```

```
## Compare row 5  and column  3 with corr  0.625
##   Means:  0.255 vs 0.196 so flagging column 5
## All correlations <= 0.6
```

```
# which variables are flagged for removal?
highlyCor
```

```
## [1] "extravasats"
```

```
#then we remove these variables
train_data_cor <- train_data[, which(!colnames(train_data) %in% highlyCor)]
```

GRID SEARCH WITH CARET

Automatic Grid

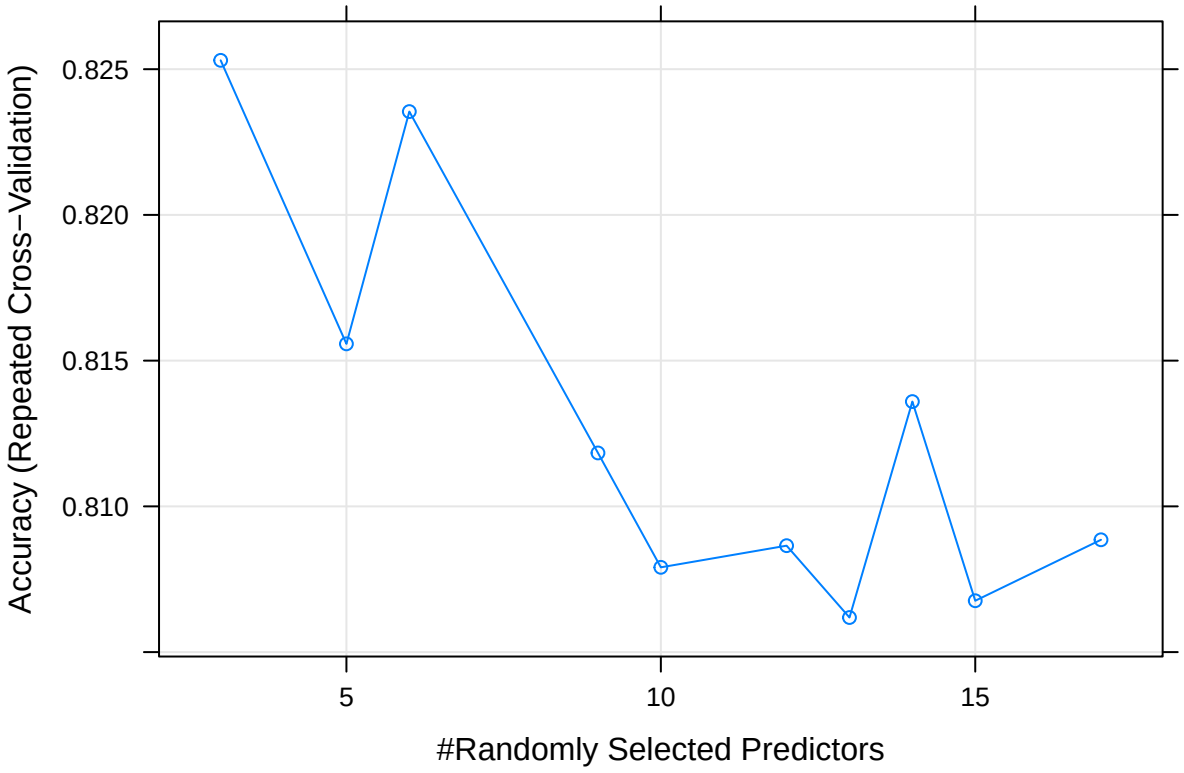
```
set.seed(42)
model_rf_tune_auto <- caret::train(Class ~ .,
                         data = train_data,
                         method = "rf",
                         preProcess = c("scale", "center"),
                         trControl = trainControl(method = "repeatedcv",
                                                  number = 10,
```

```
                                                    repeats = 10,
                                                    savePredictions = TRUE,
                                                    verboseIter = FALSE,
                                                    search = "random"),
                          tuneLength = 15)
```

```
model_rf_tune_auto
```

```
## Random Forest
##
## 104 samples
##  18 predictor
##   4 classes: 'fibrosis', 'malign', 'metastases', 'normal'
##
## Pre-processing: scaled (18), centered (18)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 94, 93, 95, 93, 94, 94, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    3    0.8253030  0.6592262
##    5    0.8155758  0.6406758
##    6    0.8235455  0.6580657
##    9    0.8118333  0.6353422
##   10    0.8079091  0.6273998
##   12    0.8086515  0.6300472
##   13    0.8061869  0.6253362
##   14    0.8135960  0.6397678
##   15    0.8067626  0.6267725
##   17    0.8088535  0.6307790
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

```
plot(model_rf_tune_auto)
```



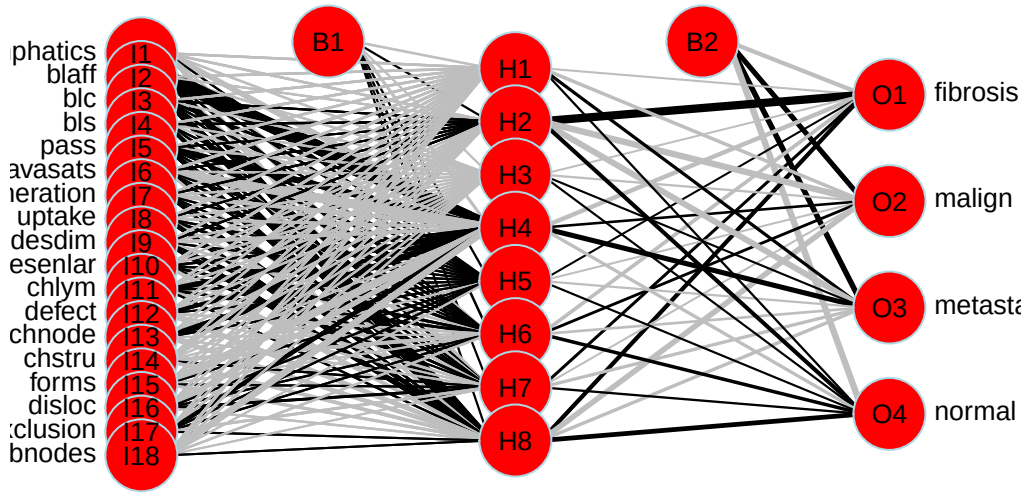
NEURAL NETWORK MODEL

```
library(nnet)
model_nnet<-nnet(Class ~. ,
                 data= train_data,
                 size=8
)
```

```
## # weights:  188
## initial  value 198.851838
## iter  10 value 52.494621
## iter  20 value 29.598474
## iter  30 value 22.748819
## iter  40 value 21.021300
## iter  50 value 20.435859
## iter  60 value 14.798132
## iter  70 value 3.550605
## iter  80 value 3.303038
## iter  90 value 3.099452
## iter 100 value 1.949187
## final  value 1.949187
## stopped after 100 iterations
```

```
library(NeuralNetTools)
# Plot a neural interpretation diagram for a neural network object
```

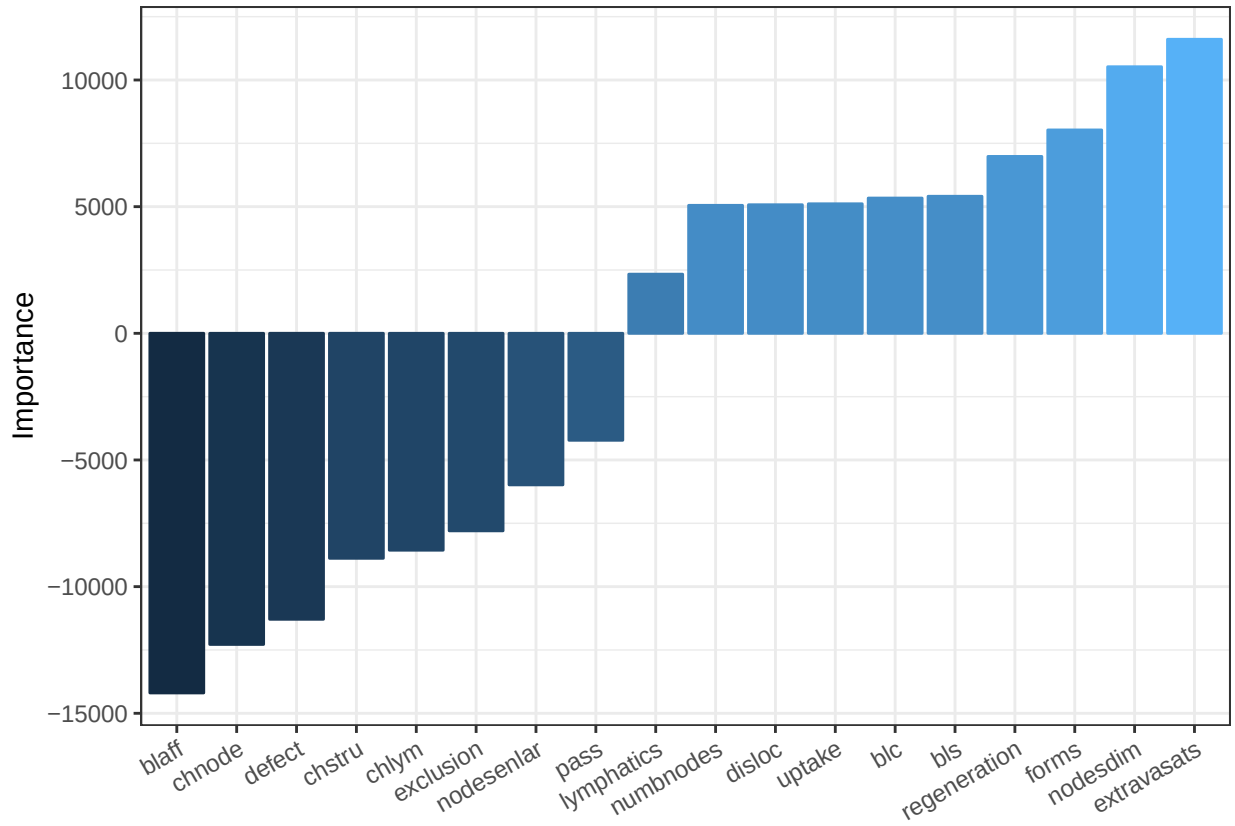
```
plotnet(model_nnet, cex_val =.8,max_sp=T,circle_cex=5,circle_col = 'red')
```




```
#Relative importance of input variables in neural networks using Garson's algorithm
#garson(model_nnet)
```

```
olden(model_nnet) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```

```
## Warning in olden.default(wts_in, x_names, y_names, skip_wts = skip_wts, :
## Results for first response variable only, use out_var argument to change
```



Here both the positve and negative value represents relative contibutions of each connection weight among the variables

```
#Predict
predict_nnet <- predict(model_nnet,test_data, type = "class")
```

```
#Draw the crosstable

library(gmodels)
CrossTable(test_data$Class,predict_nnet,prop.chisq = F,prop.r = F,prop.c = F,dnn =c("Actual Diagnosis",
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  43
##
##
##                  | Predict Diagnosis
## Actual Diagnosis |   fibrosis |     malign | metastases |     normal |  Row Total |
## -----------------|------------|------------|------------|------------|------------|
##         fibrosis |          0 |          1 |          0 |          0 |          1 |
##                  |      0.000 |      0.023 |      0.000 |      0.000 |            |
```

```
## -----------------|------------|------------|------------|------------|------------|
##           malign |          0 |         12 |          5 |          1 |         18 |
##                  |      0.000 |      0.279 |      0.116 |      0.023 |            |
## -----------------|------------|------------|------------|------------|------------|
##       metastases |          1 |          6 |         16 |          1 |         24 |
##                  |      0.023 |      0.140 |      0.372 |      0.023 |            |
## -----------------|------------|------------|------------|------------|------------|
##     Column Total |          1 |         19 |         21 |          2 |         43 |
## -----------------|------------|------------|------------|------------|------------|
##
##
```