Breast Cancer

B. Whillock

May 16, 2019

library(tidyverse)  
library(rsample)  
library(ROCR)  
library(caret)  
library(bnlearn)

## Breast Cancer Data

df <- read\_csv("Z:/data/breast cancer/bc.csv")  
df <- df %>% mutate\_all(as.factor)

summary(df)

## class age menopause tumor\_size  
## no-recurrence-events:201 20-29: 1 ge40 :129 30-34 :60   
## recurrence-events : 85 30-39:36 lt40 : 7 25-29 :54   
## 40-49:90 premeno:150 20-24 :50   
## 50-59:96 15-19 :30   
## 60-69:57 10-14 :28   
## 70-79: 6 40-44 :22   
## (Other):42   
## inv\_nodes node\_caps deg\_malig breast breast\_quad irradiat   
## 0-2 :213 ? : 8 1: 71 left :152 ? : 1 no :218   
## 12-14: 3 no :222 2:130 right:134 central : 21 yes: 68   
## 15-17: 6 yes: 56 3: 85 left\_low :110   
## 24-26: 1 left\_up : 97   
## 3-5 : 36 right\_low: 24   
## 6-8 : 17 right\_up : 33   
## 9-11 : 10

# Merge factor groups that only have a single point  
df$age <- df$age %>% fct\_collapse("20-39" = c("20-29", "30-39"))  
df$inv\_nodes <- df$inv\_nodes %>% fct\_collapse("15-26" = c("15-17","24-26"))  
df$breast\_quad <- df$breast\_quad %>% fct\_collapse("central" = c("central", "?"))

# Examine class distribution  
table(df$class) %>% prop.table()

##   
## no-recurrence-events recurrence-events   
## 0.7027972 0.2972028

## Training

# 80/20 Train-Test split with near equal class distributions  
set.seed(1)  
split <- initial\_split(df, prop=.8, strata="class")  
train <- training(split)  
test <- testing(split)

# set up 10-fold CV  
features <- setdiff(names(train), "class")  
x <- train[, features]  
y <- train$class  
train\_control <- trainControl(  
 method = "cv",  
 number = 10  
)

### Logistic Regression

# Logistic regression  
fit\_lr <- train(  
 x = x,  
 y = y,  
 method = "glm",  
 family = "binomial",  
 trControl = train\_control,  
 trace = FALSE  
)

summary(fit\_lr)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6555 -0.7615 -0.4797 0.8639 2.4166   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.84127 1.82655 -1.556 0.1198   
## age40-49 -0.35336 0.56043 -0.631 0.5284   
## age50-59 -0.47755 0.64515 -0.740 0.4592   
## age60-69 -0.04130 0.81973 -0.050 0.9598   
## age70-79 0.11351 1.62985 0.070 0.9445   
## menopauselt40 -0.25400 1.21521 -0.209 0.8344   
## menopausepremeno 0.37822 0.57194 0.661 0.5084   
## tumor\_size10-14 -2.26334 1.65886 -1.364 0.1724   
## tumor\_size15-19 -0.19424 1.32186 -0.147 0.8832   
## tumor\_size20-24 0.27821 1.26509 0.220 0.8259   
## tumor\_size25-29 0.07173 1.28736 0.056 0.9556   
## tumor\_size30-34 0.09630 1.27775 0.075 0.9399   
## tumor\_size35-39 -0.65828 1.39976 -0.470 0.6382   
## tumor\_size40-44 -0.80571 1.38373 -0.582 0.5604   
## tumor\_size45-49 0.34165 1.93488 0.177 0.8598   
## tumor\_size5-9 -14.36558 790.92071 -0.018 0.9855   
## tumor\_size50-54 0.16039 1.57655 0.102 0.9190   
## inv\_nodes12-14 0.79838 1.53195 0.521 0.6023   
## inv\_nodes15-26 1.11425 1.08995 1.022 0.3066   
## inv\_nodes3-5 0.61771 0.55966 1.104 0.2697   
## inv\_nodes6-8 0.53375 0.80248 0.665 0.5060   
## inv\_nodes9-11 1.68704 1.05561 1.598 0.1100   
## node\_capsno 0.91802 1.25527 0.731 0.4646   
## node\_capsyes 1.35083 1.27004 1.064 0.2875   
## deg\_malig2 -0.26704 0.50916 -0.524 0.5999   
## deg\_malig3 0.91273 0.52055 1.753 0.0795 .  
## breastright -0.47500 0.38241 -1.242 0.2142   
## breast\_quadleft\_low 0.98235 0.82863 1.186 0.2358   
## breast\_quadleft\_up 0.98846 0.84592 1.169 0.2426   
## breast\_quadright\_low 0.12211 1.01863 0.120 0.9046   
## breast\_quadright\_up 1.24022 0.90349 1.373 0.1698   
## irradiatyes 0.60130 0.41393 1.453 0.1463   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 281.00 on 229 degrees of freedom  
## Residual deviance: 224.77 on 198 degrees of freedom  
## AIC: 288.77  
##   
## Number of Fisher Scoring iterations: 14

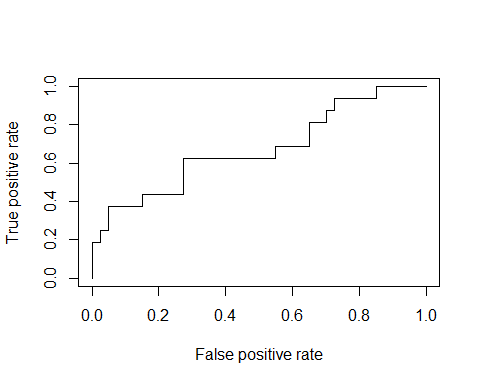
pred\_lr <- predict(fit\_lr$finalModel, test, type="response")  
pred\_lr <- factor(ifelse(pred\_lr<0.5, "no-recurrence-events", "recurrence-events"))  
confusionMatrix(pred\_lr, test$class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no-recurrence-events recurrence-events  
## no-recurrence-events 33 9  
## recurrence-events 7 7  
##   
## Accuracy : 0.7143   
## 95% CI : (0.5779, 0.827)   
## No Information Rate : 0.7143   
## P-Value [Acc > NIR] : 0.5669   
##   
## Kappa : 0.2727   
## Mcnemar's Test P-Value : 0.8026   
##   
## Sensitivity : 0.8250   
## Specificity : 0.4375   
## Pos Pred Value : 0.7857   
## Neg Pred Value : 0.5000   
## Prevalence : 0.7143   
## Detection Rate : 0.5893   
## Detection Prevalence : 0.7500   
## Balanced Accuracy : 0.6312   
##   
## 'Positive' Class : no-recurrence-events  
##

pred\_lr <- predict(fit\_lr$finalModel, test, type="response")  
pred\_lr <- prediction(pred\_lr, test$class)  
auc\_lr <- performance(pred\_lr,"auc"); auc\_lr <- as.numeric(auc\_lr@y.values);   
paste("AUC: ", auc\_lr)

## [1] "AUC: 0.6734375"

perf\_lr <- performance(pred\_lr, 'tpr', 'fpr'); plot(perf\_lr)



### Naive Bayes

# Naive Bayes  
fit\_nb <- train(  
 x = x,  
 y = y,  
 method = "nb",  
 trControl = train\_control,  
 trace = FALSE  
)

fit\_nb

## Naive Bayes   
##   
## 230 samples  
## 9 predictor  
## 2 classes: 'no-recurrence-events', 'recurrence-events'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 207, 207, 207, 207, 207, 207, ...   
## Resampling results across tuning parameters:  
##   
## usekernel Accuracy Kappa   
## FALSE 0.7220026 0.3102515  
## TRUE 0.7220026 0.3102515  
##   
## Tuning parameter 'fL' was held constant at a value of 0  
## Tuning  
## parameter 'adjust' was held constant at a value of 1  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were fL = 0, usekernel = FALSE  
## and adjust = 1.

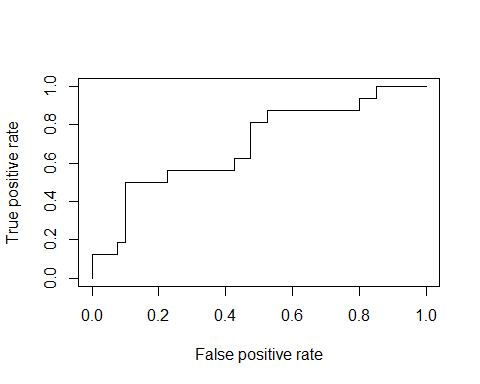
pred\_nb <- predict(fit\_nb$finalModel, test)  
confusionMatrix(pred\_nb$class, test$class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no-recurrence-events recurrence-events  
## no-recurrence-events 32 8  
## recurrence-events 8 8  
##   
## Accuracy : 0.7143   
## 95% CI : (0.5779, 0.827)   
## No Information Rate : 0.7143   
## P-Value [Acc > NIR] : 0.5669   
##   
## Kappa : 0.3   
## Mcnemar's Test P-Value : 1.0000   
##   
## Sensitivity : 0.8000   
## Specificity : 0.5000   
## Pos Pred Value : 0.8000   
## Neg Pred Value : 0.5000   
## Prevalence : 0.7143   
## Detection Rate : 0.5714   
## Detection Prevalence : 0.7143   
## Balanced Accuracy : 0.6500   
##   
## 'Positive' Class : no-recurrence-events  
##

pred\_nb <- prediction(pred\_nb$posterior[,2], test$class)  
auc\_nb <- performance(pred\_nb,"auc"); auc\_nb <- as.numeric(auc\_nb@y.values);   
paste("AUC :", auc\_nb)

## [1] "AUC : 0.6984375"

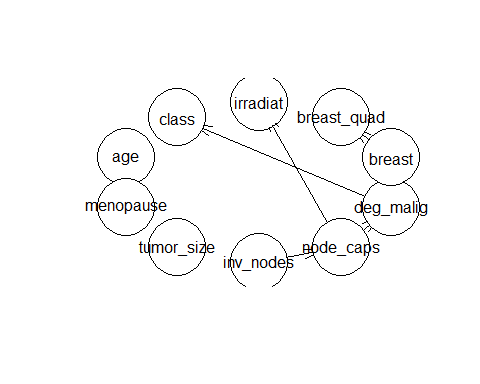
perf\_nb <- performance(pred\_nb, 'tpr', 'fpr'); plot(perf\_nb)



## Bayesian Network

# Bayesian Network  
fit\_bn <- hc(as.data.frame(train))

plot(fit\_bn)



fit\_bn <- bn.fit(fit\_bn, as.data.frame(train))

fit\_bn

##   
## Bayesian network parameters  
##   
## Parameters of node class (multinomial distribution)  
##   
## Conditional probability table:  
##   
## deg\_malig  
## class 1 2 3  
## no-recurrence-events 0.8305085 0.7641509 0.4769231  
## recurrence-events 0.1694915 0.2358491 0.5230769  
##   
## Parameters of node age (multinomial distribution)  
##   
## Conditional probability table:  
## 20-39 40-49 50-59 60-69 70-79   
## 0.12608696 0.31304348 0.31739130 0.22173913 0.02173913   
##   
## Parameters of node menopause (multinomial distribution)  
##   
## Conditional probability table:  
##   
## age  
## menopause 20-39 40-49 50-59 60-69 70-79  
## ge40 0.00000000 0.08333333 0.60273973 0.96078431 1.00000000  
## lt40 0.03448276 0.00000000 0.04109589 0.03921569 0.00000000  
## premeno 0.96551724 0.91666667 0.35616438 0.00000000 0.00000000  
##   
## Parameters of node tumor\_size (multinomial distribution)  
##   
## Conditional probability table:  
## 0-4 10-14 15-19 20-24 25-29 30-34   
## 0.034782609 0.104347826 0.104347826 0.169565217 0.182608696 0.208695652   
## 35-39 40-44 45-49 5-9 50-54   
## 0.069565217 0.078260870 0.008695652 0.013043478 0.026086957   
##   
## Parameters of node inv\_nodes (multinomial distribution)  
##   
## Conditional probability table:  
## 0-2 12-14 15-26 3-5 6-8 9-11   
## 0.75217391 0.01304348 0.02608696 0.12608696 0.05217391 0.03043478   
##   
## Parameters of node node\_caps (multinomial distribution)  
##   
## Conditional probability table:  
##   
## inv\_nodes  
## node\_caps 0-2 12-14 15-26 3-5 6-8  
## ? 0.01156069 0.00000000 0.00000000 0.06896552 0.00000000  
## no 0.94219653 0.33333333 0.16666667 0.44827586 0.16666667  
## yes 0.04624277 0.66666667 0.83333333 0.48275862 0.83333333  
## inv\_nodes  
## node\_caps 9-11  
## ? 0.28571429  
## no 0.14285714  
## yes 0.57142857  
##   
## Parameters of node deg\_malig (multinomial distribution)  
##   
## Conditional probability table:  
##   
## node\_caps  
## deg\_malig ? no yes  
## 1 0.6666667 0.3038674 0.0000000  
## 2 0.1666667 0.4640884 0.4883721  
## 3 0.1666667 0.2320442 0.5116279  
##   
## Parameters of node breast (multinomial distribution)  
##   
## Conditional probability table:  
## left right   
## 0.5217391 0.4782609   
##   
## Parameters of node breast\_quad (multinomial distribution)  
##   
## Conditional probability table:  
##   
## breast  
## breast\_quad left right  
## central 0.09166667 0.08181818  
## left\_low 0.48333333 0.21818182  
## left\_up 0.23333333 0.44545455  
## right\_low 0.12500000 0.05454545  
## right\_up 0.06666667 0.20000000  
##   
## Parameters of node irradiat (multinomial distribution)  
##   
## Conditional probability table:  
##   
## node\_caps  
## irradiat ? no yes  
## no 0.1666667 0.8453039 0.4651163  
## yes 0.8333333 0.1546961 0.5348837

pred\_bn <- predict(fit\_bn, node="class", as.data.frame(test))  
bn\_test\_acc <- sum(pred\_bn==test$class)/length(pred\_bn)  
paste("Test Accuracy: ", bn\_test\_acc)

## [1] "Test Accuracy: 0.75"