Naive Baye from Scratch with the Titanic Data

Karen Mazidi

We will use the same data and data cleaning as in the first notebook in this chapter, so we repeate those steps first with no commentary.

Load the data

```
df <- read.csv("data/titanic3.csv", header=TRUE, stringsAsFactors = FALSE)

# subset to just columns survived, pclass, sex, and age
df <- df[,c(1,2,4,5)]
# pclass and survived  and sex should be factors
df$pclass <- factor(df$pclass)
df$survived <- factor(df$survived)
df$sex <- factor(df$sex, levels=c("female", "male"))

# remove NAs
df <- df[!is.na(df$pclass),]
df <- df[!is.na(df$survived),]
df$age[is.na(df$age)] <- median(df$age,na.rm=T)

# divide into train and test
set.seed(1234)
i <- sample(1:nrow(df), 0.75*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]

# perform Naive Bayes
library(e1071)
nb1 <- naiveBayes(df[,-2], df[,2], data=train)
pred <- predict(nb1, newdata=test[,-2], type="raw")

# look at first 5 (actual: 0 1 1 1 0)
pred[1:5,]
```

```
##               0         1
## [1,] 0.07032876 0.9296712
## [2,] 0.13421950 0.8657805
## [3,] 0.64596090 0.3540391
## [4,] 0.11063661 0.8893634
## [5,] 0.11954430 0.8804557
```

Calculate priors

Using the training data we calculate prior probability fo survived the percentage for each category.

```
apriori <- c(
  nrow(df[df$survived=="0",])/nrow(df),
  nrow(df[df$survived=="1",])/nrow(df)
  )
print("Prior probability, survived=no, survived=yes:")
```

```
## [1] "Prior probability, survived=no, survived=yes:"
```

```
apriori
```

```
## [1] 0.618029 0.381971
```

Calculate likelihoods for qualitative data

The likelihood for qualitative data is calculated as follows:

- for each class
- for each factor i
- likelihood (class=i|survived=yes) = count(factor = i and survived=yes) / count(survived=yes)
- likelihood (class=i|survived=no) = count(factor = i and survived=no) / count(survived=np)

we will use nrow() for our counts

```
# get survived counts for no and yes
count_survived <- c(
  length(df$survived[df$survived=="0"]),
  length(df$survived[df$survived=="1"])
)
# likelihood for pclass
lh_pclass <- matrix(rep(0,6), ncol=3)
for (sv in c("0", "1")){
  for (pc in c("1","2","3")) {
    lh_pclass[as.integer(sv)+1, as.integer(pc)] <-
      nrow(df[df$pclass==pc & df$survived==sv,]) / count_survived[as.integer(sv)+1]
  }
}

# likelihood for sex
lh_sex <- matrix(rep(0,4), ncol=2)
for (sv in c("0", "1")){
  for (sx in c(1, 2)) {
    lh_sex[as.integer(sv)+1, sx] <-
      nrow(df[as.integer(df$sex)==sx & df$survived==sv,]) /
       count_survived[as.integer(sv)+1]
  }
}
```

likelihood p(survived|pclass)

```
print("Likelihood values for p(pclass|survived):")
```

```
## [1] "Likelihood values for p(pclass|survived):"
```

```
lh_pclass
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.1520396 0.1953028 0.6526576
## [2,] 0.4000000 0.2380000 0.3620000
```

likelihood p(survived|sex)

```
print("Likelihood values for p(sex|survived):")
```

```
## [1] "Likelihood values for p(sex|survived):"
```

```
lh_sex
```

```
##           [,1]      [,2]
## [1,] 0.1569839 0.8430161
## [2,] 0.6780000 0.3220000
```

Calculate likelihoods for quantitative data

Age is quantitative. We need to compute the mean and variance.

```
age_mean <- c(0, 0)
age_var <- c(0, 0)
for (sv in c("0", "1")){
    age_mean[as.integer(sv)+1] <-
      mean(df$age[df$survived==sv])
    age_var[as.integer(sv)+1] <-
      var(df$age[df$survived==sv])
  }
```

Probability density for quantitative data

For the qualitative variable we can calculate probailities by dividing but for the age variable we need a function that will calculate its probability.

```
calc_age_lh <- function(v, mean_v, var_v){
  # run like this: calc_age_lh(6, 25.9, 138)
  1 / sqrt(2 * pi * var_v) * exp(-((v-mean_v)^2)/(2 * var_v))
}
```

Function for scratch model

Write a function to calculate raw probabilities given pclass, sex, and age.

```
calc_raw_prob <- function(pclass, sex, age) {
  # pclass=1,2,3  sex=1,2   age=numeric
  num_s <- lh_pclass[2, pclass] * lh_sex[2, sex] * apriori[2] *
      calc_age_lh(age, age_mean[2], age_var[2])
  num_p <- lh_pclass[1, pclass] * lh_sex[1, sex] * apriori[1] *
      calc_age_lh(age, age_mean[1], age_var[1])
  denominator <- lh_pclass[2, pclass]  * lh_sex[2, sex] * calc_age_lh(age, age_mean[2], age_var[2]) * ap
      lh_pclass[1, pclass]  * lh_sex[1, sex] * calc_age_lh(age, age_mean[1], age_var[1]) * apriori[1]
  return (list(prob_survived <- num_s / denominator, prob_perished <- num_p / denominator))
}
```

Apply to the first 5 test observations

Let's look at just the first 5 test observations.

Note getting the right numbers, need to rethink likelihood values, flip table. Works for first 3=female, not for male.

```
for (i in 1:5){
  raw <- calc_raw_prob(test[i,1], as.integer(test[i,3]), test[i,4])
  print(paste(raw[2], raw[1]))
}
```

```
## [1] "0.0703287599020377 0.929671240097962"
## [1] "0.134219499226771 0.865780500773229"
## [1] "0.64596090198233 0.35403909801767"
## [1] "0.110636612039489 0.889363387960511"
## [1] "0.119544295476936 0.880455704523064"
```

```
pred[1:5,]
```

```
##               0         1
## [1,] 0.07032876 0.9296712
## [2,] 0.13421950 0.8657805
## [3,] 0.64596090 0.3540391
## [4,] 0.11063661 0.8893634
## [5,] 0.11954430 0.8804557
```