Classification

Dian Ramadhani

08/01/2019

Classification

Klasifikasi adalah suatu pengelompokan data dimana data yang digunakan tersebut mempunyai kelas label atau target. Model yang diperoleh digunakan untuk memprediksi kelas dari data baru. Model klasifikasi dapat dibuat dengan berbagai algoritma seperti Decision Tree, Naive Bayes, K-NN, dll.

Install Packages

```
# Menginstall package(s)
install.packages("readr") # membaca file
install.packages("here") # menampilkan direktori
install.packages("rpart") # algoritma decision tree
install.packages("rpart.plot") # visualisasi decision tree
install.packages("naivebayes") # algoritma naivebayes
```

Import Library

Setelah paket diinstal, paket tersebut tidak secara otomatis aktif. Dengan demikian, paket yang telah diinstal selanjutnya akan diaktifkan melalui perintah berikut.

```
# Mengaktifkan package(s)
library(readr)
library(here)
library(rpart)
library(rpart.plot)
library(naivebayes)
```

Menampilkan Direktori

```
# Mengetahui direktori proyek
here()
```

Import Data

```
# Mengimport data
df.insurance <- read_csv(here("data", "raw", "classification_insurance.csv"))
```

```
## Parsed with column specification:
## cols(
##   Age = col_double(),
##   Sex = col_character(),
##   Bmi = col_double(),
##   Children = col_double(),
##   Smoker = col_character(),
##   Region = col_character(),
##   Charges = col_double(),
##   Claim = col_character()
```

```
## )
```

Data yang digunakan yaitu data asuransi. Data ini berisi tentang profil calon konsumen perusahaan asuransi dan keputusan pengambilan asuransi yang diambil masing masing orang. Data ini kemudian akan digunakan untuk memprediksi kecenderungan keputusan yang diambil oleh calon konsumen lainnya berdasarkan profil yang Ia miliki.

Eksplorasi Data

Data yang telah diimpor selanjutnya dieksplorasi untuk mengetahui strukturnya.

```
# Melihat attribute dan struktur data
names(df.insurance) # menampilkan nama kolom
```

```
## [1] "Age"      "Sex"      "Bmi"      "Children" "Smoker"   "Region"
## [7] "Charges"  "Claim"
```

```
dim(df.insurance) # menampilkan dimensi tabel
```

```
## [1] 1338    8
```

```
head(df.insurance) # menampilkan beberap data teratas
```

```
## # A tibble: 6 x 8
##     Age Sex      Bmi Children Smoker     Region    Charges Claim
##   <dbl> <chr>  <dbl>    <dbl> <chr>      <chr>       <dbl> <chr>
## 1    19 Female  27.9        0 Smoker     Southwest  16885. Yes
## 2    18 Male    33.8        1 Non Smoker Southeast   1726. Yes
## 3    28 Male    33          3 Non Smoker Southeast   4449. No
## 4    33 Male    22.7        0 Non Smoker Northwest  21984. No
## 5    32 Male    28.9        0 Non Smoker Northwest   3867. Yes
## 6    31 Female  25.7        0 Non Smoker Southeast   3757. No
```

```
str(df.insurance) # menampilkan struktur data
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 1338 obs. of  8 variables:
##  $ Age     : num  19 18 28 33 32 31 46 37 37 60 ...
##  $ Sex     : chr  "Female" "Male" "Male" "Male" ...
##  $ Bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ Children: num  0 1 3 0 0 0 1 3 2 0 ...
##  $ Smoker  : chr  "Smoker" "Non Smoker" "Non Smoker" "Non Smoker" ...
##  $ Region  : chr  "Southwest" "Southeast" "Southeast" "Northwest" ...
##  $ Charges : num  16885 1726 4449 21984 3867 ...
##  $ Claim   : chr  "Yes" "Yes" "No" "No" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Age = col_double(),
##   ..   Sex = col_character(),
##   ..   Bmi = col_double(),
##   ..   Children = col_double(),
##   ..   Smoker = col_character(),
##   ..   Region = col_character(),
##   ..   Charges = col_double(),
##   ..   Claim = col_character()
##   .. )
```

```
summary(df.insurance) # menampilkan rangkuman data
```

```
##       Age            Sex                 Bmi           Children
##  Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   Median :30.40   Median :1.000
##  Mean   :39.21                      Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                      Max.   :53.13   Max.   :5.000
##     Smoker             Region             Charges         Claim
##  Length:1338        Length:1338        Min.   : 1122   Length:1338
##  Class :character   Class :character   1st Qu.: 4740   Class :character
##  Mode  :character   Mode  :character   Median : 9382   Mode  :character
##                                        Mean   :13270
##                                        3rd Qu.:16640
##                                        Max.   :63770
```

```r
# Mengetahui jumlah data kosong
sum(is.na(df.insurance))
```

```
## [1] 0
```

Membagi Data

Dalam klasifikasi, data dibagi menjadi dua yaitu data train untuk membuat model dan data test untuk menguji akurasi model. Biasanya data dibagi dengan proporsi 70% train dan 30% test.

```r
# Membangi data
split <- sample(1:nrow(df.insurance), 0.7 * nrow(df.insurance))

# Membuat tabel data train
df.train <- df.insurance[split, ]

# Membut tabel data test
df.test <- df.insurance[-split, ]
```

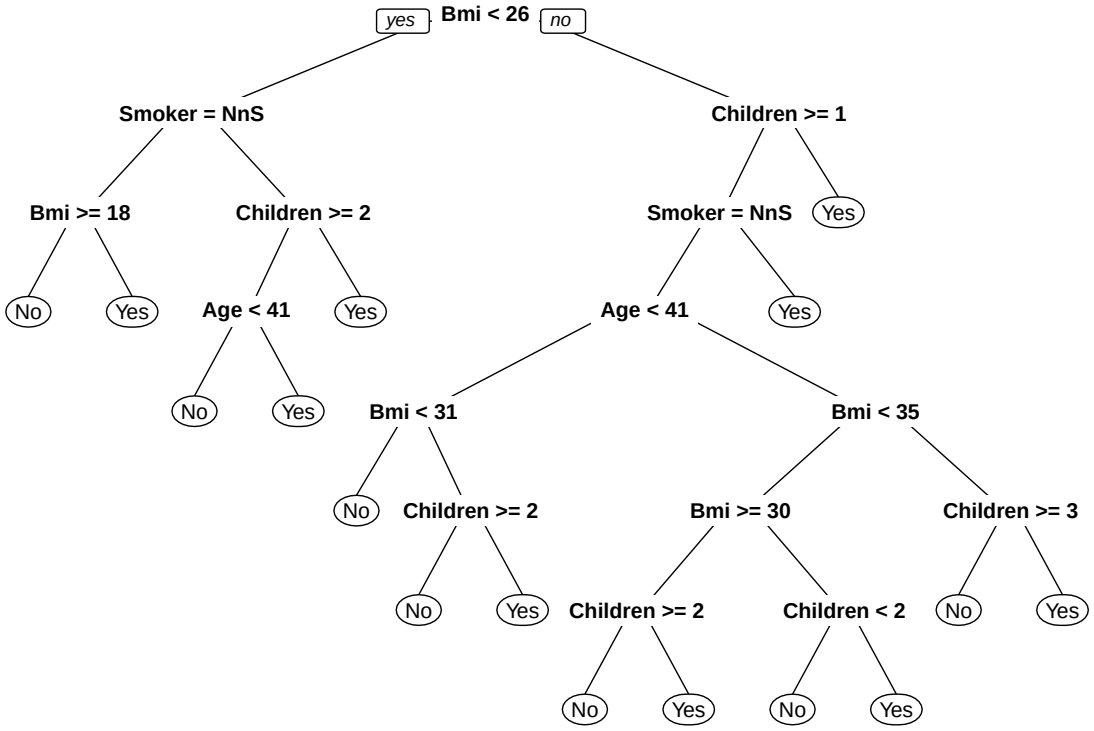
Decision Tree

Decision tree adalah salah satu metode klasifikasi yang paling populer, karena mudah untuk diinterpretasi oleh manusia. Decision tree adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. Konsep dari pohon keputusan adalah mengubah data menjadi decision tree dan aturan-aturan keputusan.

Model Building

```r
# Membuat model decision tree
tree <- rpart(Claim ~ ., data = df.train)

# Memvisualisasikan model decision tree
prp(tree)
```



Validasi

```
# Memprediksi data test
pred.tree <- predict(tree, df.test, type = "class")

# Melihat prediksi dalam bentuk tabel
df.pred.tree <- data.frame(df.test, pred.tree)
View(df.pred.tree)
```

```
# Confussion matrix
conf.tree <- table(df.test$Claim, pred.tree)
conf.tree
```

```
##      pred.tree
##        No Yes
##   No  164   6
##   Yes   3 229
```

```
# Megambil angka TP, FN, FP, TN
TP.tree <- conf.tree[1, 1]
FN.tree <- conf.tree[1, 2]
FP.tree <- conf.tree[2, 1]
TN.tree <- conf.tree[2, 2]

# Menghitung nilai akurasi
acc.tree <- (TP.tree + TN.tree) / (TP.tree + FN.tree + FP.tree + TN.tree)
acc.tree
```

```
## [1] 0.9776119
```

```
# Menghitung nilai presisi
prec.tree <- TP.tree / (TP.tree + FP.tree)
prec.tree
```

```
## [1] 0.9820359
```

```
# Menghitung Nilai Recall
rec.tree <- TP.tree / (TP.tree + FN.tree)
rec.tree
```

```
## [1] 0.9647059
```

Naive Bayes

Naive bayesian klasifikasi adalah suatu klasifikasi berpeluang sederhana berdasarkan aplikasi teorema Bayes dengan asumsi antar variabel penjelas saling bebas (independen). Dalam hal ini, diasumsikan bahwa kehadiran atau ketiadaan dari suatu kejadian tertentu dari suatu kelompok tidak berhubungan dengan kehadiran atau ketiadaan dari kejadian lainnya.

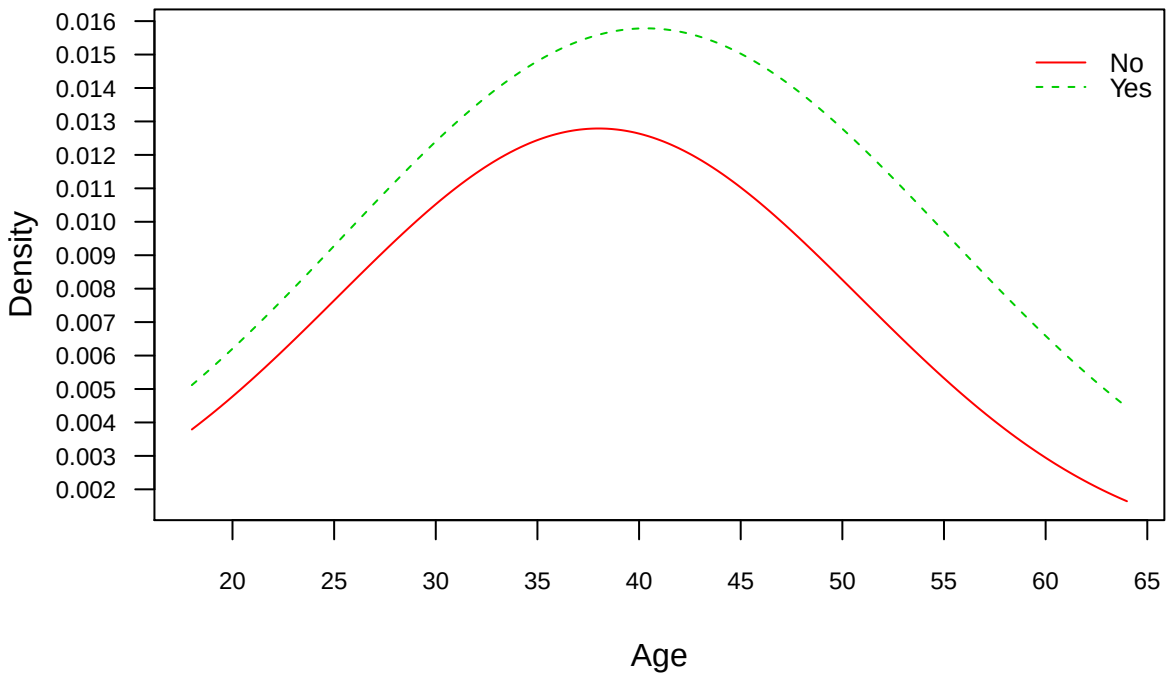
Model Building

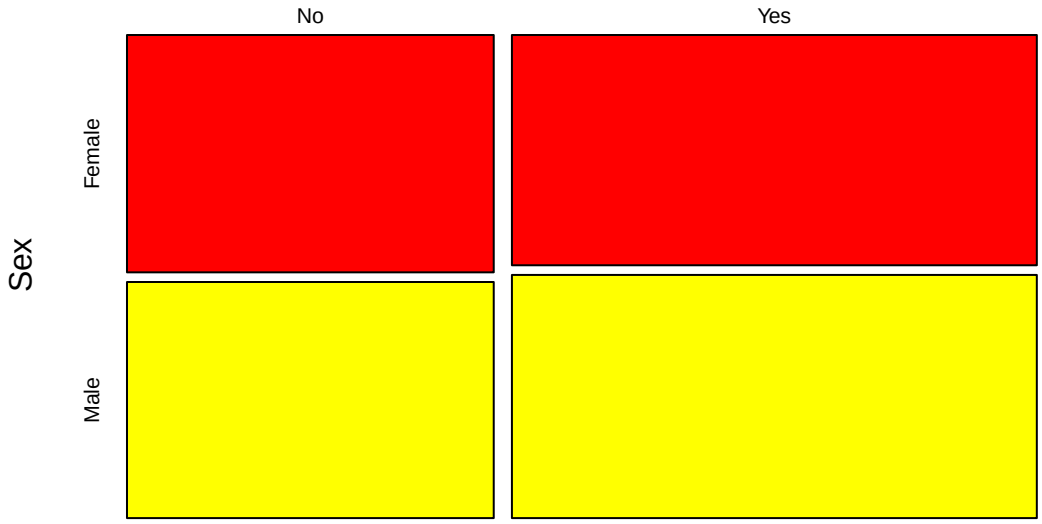
```
# Membuat model naive bayes
nb <- naive_bayes(Claim ~ ., data = df.train)
nb
```

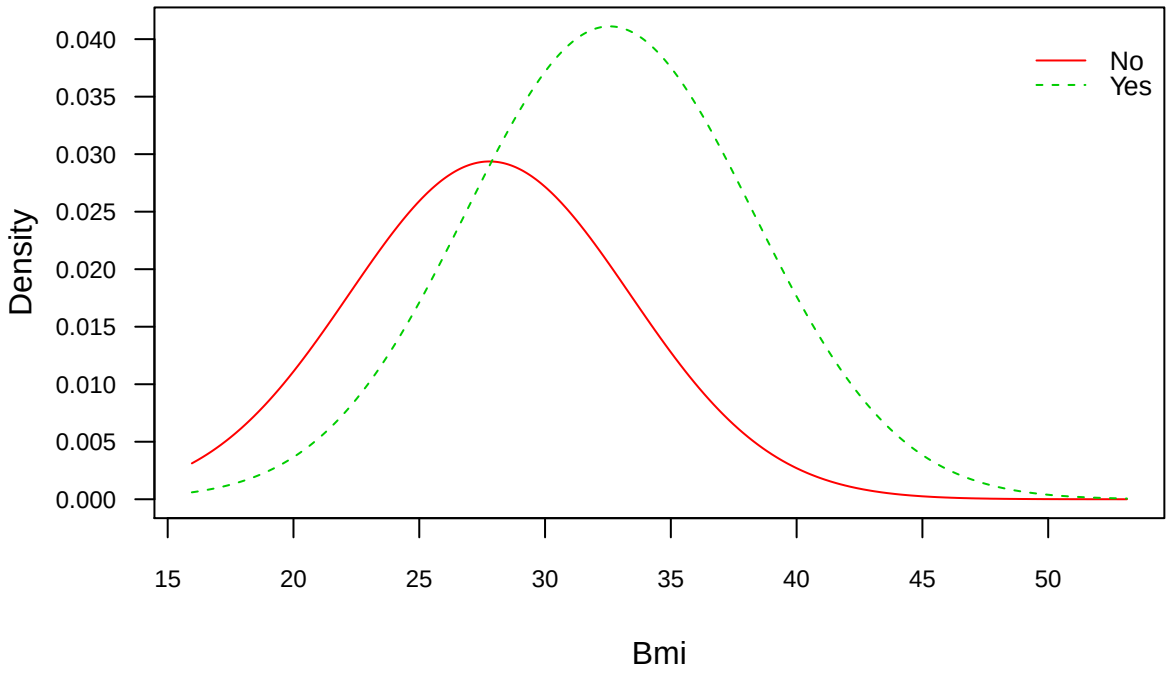
```
##
## ================================ Naive Bayes ================================
##
##  Call:
## naive_bayes.formula(formula = Claim ~ ., data = df.train)
##
## ---------------------------------------------------------------------------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------
##
##  A priori probabilities:
##
##        No       Yes
## 0.4113248 0.5886752
##
## ---------------------------------------------------------------------------
##
##  Tables:
##
## ---------------------------------------------------------------------------
##  ::: Age (Gaussian)
## ---------------------------------------------------------------------------
##
## Age          No      Yes
##   mean 38.01039 40.33394
##   sd   12.83062 14.87918
##
```

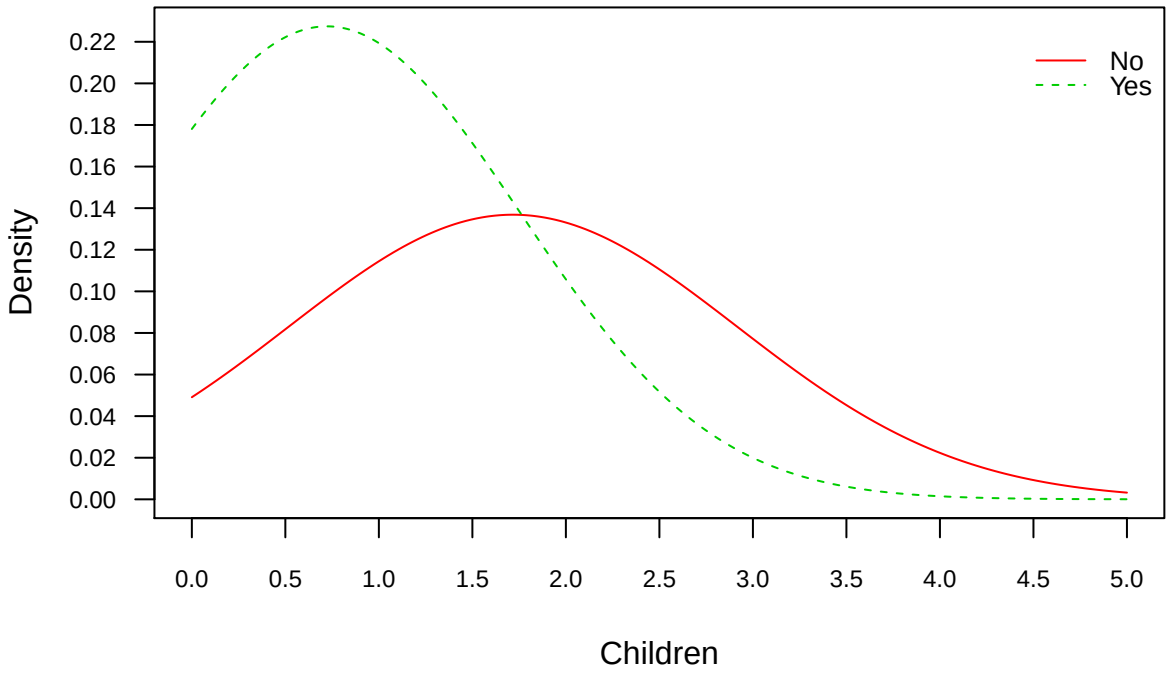
```
## ------------------------------------------------------------------------------
##  ::: Sex (Bernoulli)
## ------------------------------------------------------------------------------
##
## Sex             No       Yes
##   Female 0.5012987 0.4863884
##   Male   0.4987013 0.5136116
##
## ------------------------------------------------------------------------------
##  ::: Bmi (Gaussian)
## ------------------------------------------------------------------------------
##
## Bmi           No       Yes
##   mean 27.792104 32.564601
##   sd    5.587820  5.712381
##
## ------------------------------------------------------------------------------
##  ::: Children (Gaussian)
## ------------------------------------------------------------------------------
##
## Children       No      Yes
##     mean 1.716883 0.722323
##     sd   1.199110 1.032662
##
## ------------------------------------------------------------------------------
##  ::: Smoker (Bernoulli)
## ------------------------------------------------------------------------------
##
## Smoker               No        Yes
##   Non Smoker 0.94805195 0.67513612
##   Smoker     0.05194805 0.32486388
##
## ------------------------------------------------------------------------------
##
## # ... and 2 more tables
##
## ------------------------------------------------------------------------------
```

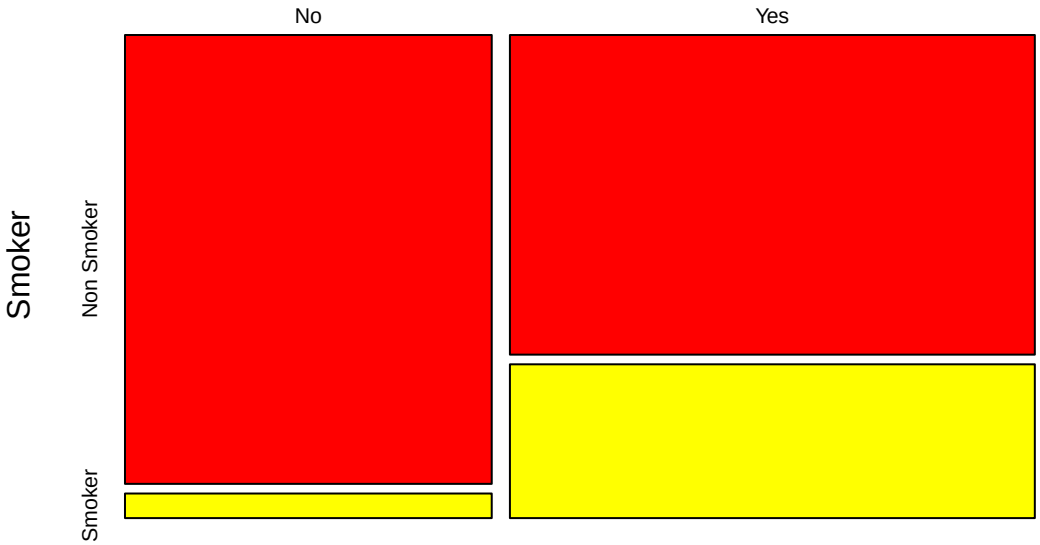
```r
# Memvisualisasikan model naive bayes
plot(nb)
```

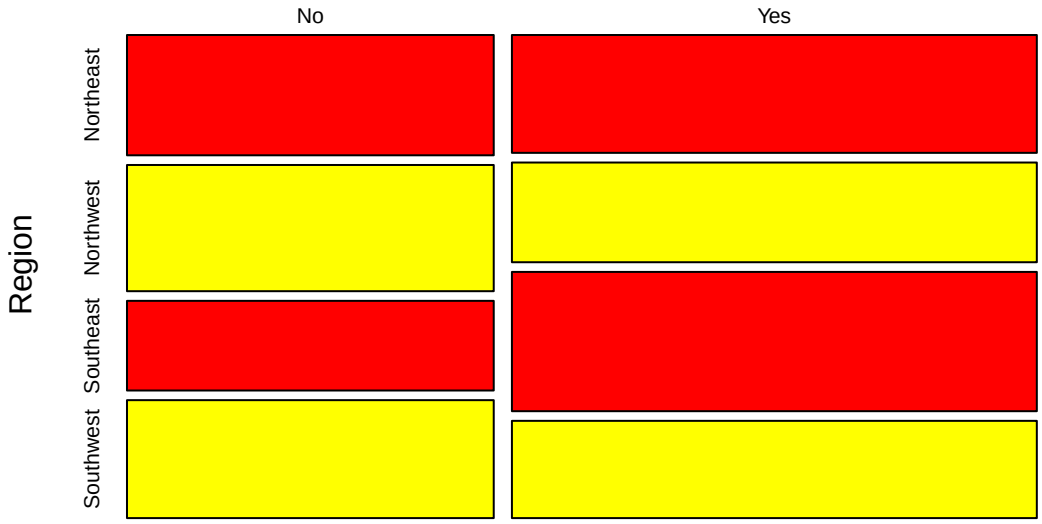

0.016
0.015
0.014
0.013
0.012
0.011
0.010
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
Density
20
25
30
35
40
45
50
55
60
65
Age
No
Yes

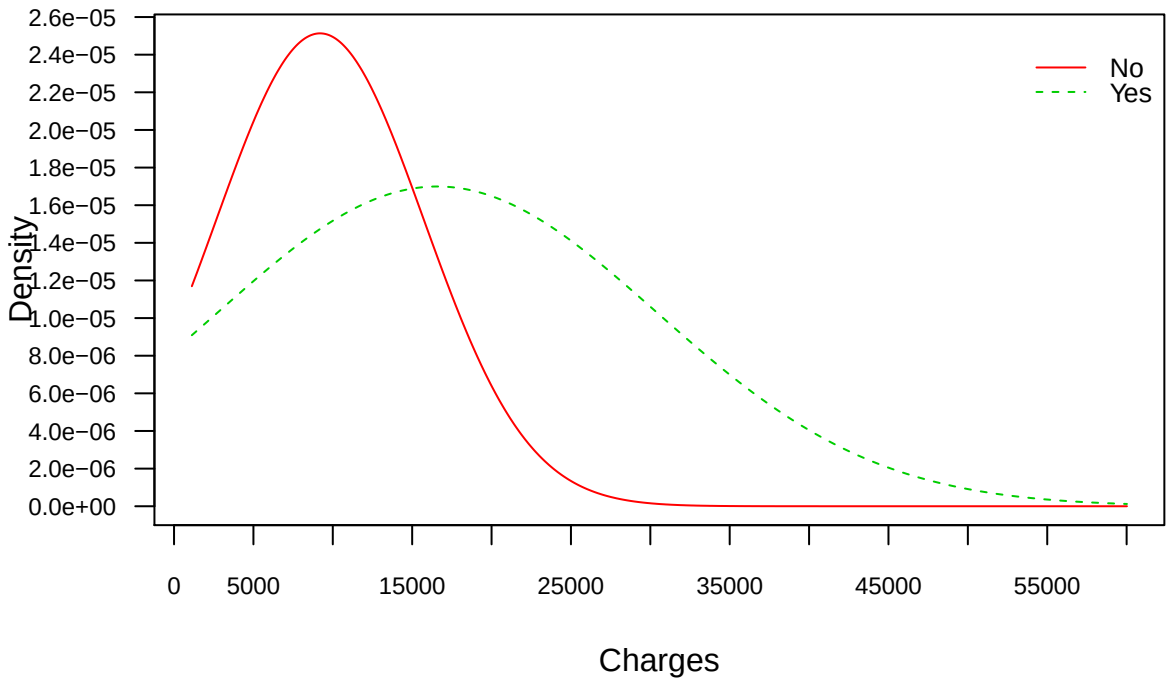

No
Yes
Female
Male
Sex


0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
15
20
25
30
35
40
45
50
Bmi
No
Yes


Density
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Children
No
Yes


No
Yes
Smoker
Non Smoker
Smoker


No
Yes
Region
Northeast
Northwest
Southeast
Southwest



Validasi

```
# Memprediksi data test
pred.nb <- predict(nb, df.test)
```

```
## Warning: predict.naive_bayes(): More features in the newdata are provided
## as there are probability tables in the object. Calculation is performed
## based on features to be found in the tables.
```

```
# Melihat prediksi dalam bentuk tabel
df.pred.nb <- data.frame(df.test, pred.nb)
View(df.pred.nb)
```

```
# Confussion matrix
conf.nb <- table(df.test$Claim, pred.nb)
conf.nb
```

```
##      pred.nb
##        No Yes
##   No  148  22
##   Yes  57 175
```

```
# Megambil angka TP, FN, FP, TN
TP.nb <- conf.nb[1, 1]
FN.nb <- conf.nb[1, 2]
FP.nb <- conf.nb[2, 1]
TN.nb <- conf.nb[2, 2]
```

```r
# Menghitung nilai akurasi
acc.nb <- (TP.nb + TN.nb) / (TP.nb + FN.nb + FP.nb + TN.nb)
acc.nb
```

```
## [1] 0.8034826
```

```r
# Menghitung nilai presisi
prec.nb <- TP.nb / (TP.nb + FP.nb)
prec.nb
```

```
## [1] 0.7219512
```

```r
# Menghitung Nilai Recall
rec.nb <- TP.nb / (TP.nb + FN.nb)
rec.nb
```

```
## [1] 0.8705882
```