



TECNOLÓGICO NACIONAL DE MÉXICO



TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO

**TECNOLÓGICO DE TIJUANA
SUBDIRECCIÓN ACADÉMICA**

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE FEBRERO-JUNIO 2022

CARRERA

Ingeniería en informática e Ingeniería en Sistemas

Computacionales

MATERIA

Minería de datos

TÍTULO

Práctica evaluatoria, unidad #3

Integrantes:

Munguía Silva Edgar Geovanny #17212344

López Higuera Saúl Alfredo #18210493

NOMBRE DEL MAESTRO

Jose Christian Romero Hernadez

Tijuana Baja California 24 de mayo del 2022



Introducción.

En esta práctica, el objetivo es implementar el modelo de clasificación de Naive Bayes usando un conjunto de datos (proporcionado por el profesor) y también implementando la biblioteca llamada e1071 usando la función naive bayes(), una vez hecha la clasificación, haremos un análisis de los datos mencionados. Naive bayes es una fórmula matemática que proviene del análisis probabilístico, la implementaremos en el análisis de datos para mostrar una relación (en términos de probabilidad) de ciertos datos.

Código.

```
## Install necessary libraries.

install.packages ("e1071")
library(e1071)
library(caTools)
library(ElemStatLearn)

## Load the CSV file

csvdata = read.csv(file.choose())

## Define what columns to use
csvdata = csvdata[3:5]
csvdata

## Get "purchased" column from the data

csvdata$Purchased=factor(csvdata$Purchased, levels = c(0,1))

## Using caTools library
set.seed(123)
```



```
split <- sample.split(csvdata$Purchased, SplitRatio = 0.75)
training_set <- subset(csvdata, split == TRUE)
test_set <- subset(csvdata, split == FALSE)
training_set[-3]=scale(training_set[-3])
test_set[-3] = scale(test_set[-3])

##Using e1071 library

classifier = naiveBayes(formula = Purchased ~ .,
                        data=training_set,
                        type='c-classification',
                        kernel = 'linear')

naiveBayes

## Predictions

y_pred=predict(classifier, newdata = test_set[-3])
y_pred

## Creating confusion matrix
cm = table(test_set[, 3], y_pred)
cm

## Visualizing the training set results (using ElemStatLearn
library)

set=training_set
x1=seq(min(set[, 1]) - 1, max(set[, 1]) +1, by =0.01)
x2=seq(min(set[, 2]) - 1, max(set[, 2]) +1, by =0.01)
grid_set=expand.grid(x1,x2)
colnames(grid_set)=c('Age', 'Estimated salary')
y_grid=predict(classifier,newdata=grid_set)
plot(set[, -3],
      main='Naive Bayes (Training set)',
      xlab = 'Age', ylab = 'Estimated salary',
```



```
xlim = range(x1), ylim=range(x2))
contour(x1, x2, matrix(as.numeric(y_grid), length(x1),
length(x2)), add=TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1,
'springgreen3', 'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4',
'red3'))

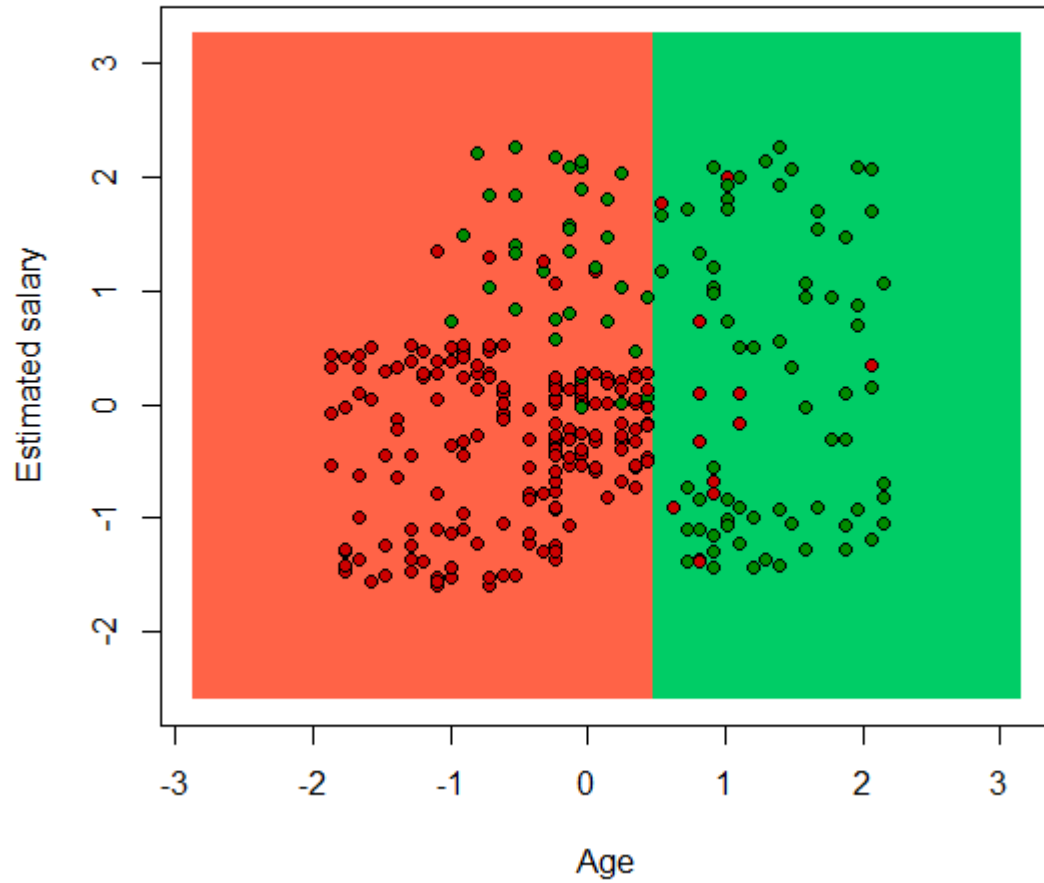
## Visualizing the test set results (using ElemStatLearn library)

set = test_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)
colnames(grid_set) = c('Age', 'EstimatedSalary')
y_grid = predict(classifier, newdata = grid_set)
plot(set[, -3], main = 'Naive Bayes (Test set)',
      xlab = 'Age', ylab = 'Estimated Salary',
      xlim = range(X1), ylim = range(X2))
contour(X1, X2, matrix(as.numeric(y_grid), length(X1),
length(X2)), add = TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1,
'springgreen3', 'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4',
'red3'))
```

Resultados.

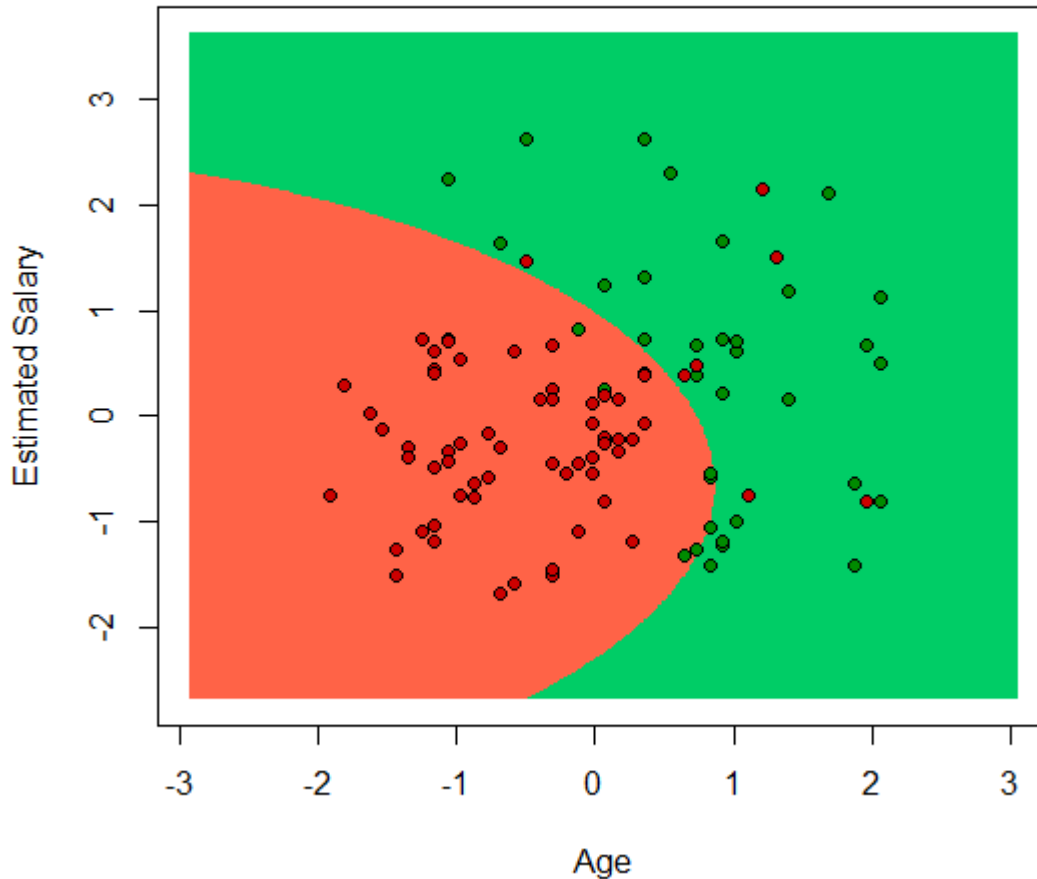
Training set.

Naive Bayes (Training set)



Test set.

Naive Bayes (Test set)



Conclusiones.

Edgar Munguia: Podemos ver algunas similitudes entre los dos gráficos, y también puedo concluir que cuantos menos datos tengamos, mejores resultados tendremos (usando este modelo). También puedo decir que la estadística es muy útil en este tipo de campo de datos, porque podemos obtener mucha información útil usando este tipo de algoritmos. Este análisis nos permitirá determinar qué decisiones tomar o cuáles no.

Saúl López:

Esta vez tocó aprender sobre Naive Bayes, requiere una pequeña cantidad de datos de entrenamiento, el clasificador aprende de los datos de entrenamiento y luego predice la clase de la instancia de prueba con la mayor probabilidad posterior, me parece un clasificador bastante preciso, ya que en las gráficas se ven ciertas características similares.



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO



Link del video (Youtube): <https://www.youtube.com/watch?v=aJg0fxQn150>

Link del repositorio de GitHub: <https://github.com/Saul12344/mineria-de-datos>