Tecnológico Nacional de México
Instituto Tecnológico de Tijuana

Subdirección Académica
Departamento de Sistemas y Computación
Ingeniería en Sistemas Computacionales
Semestre: AGOSTO-DICIEMBRE 2021

MINERÍA DE DATOS

BDD-1703SC9A

"Assessment 3"

Jiménez Ramírez Julio Fabián 17212147

Flores González Luis Diego C16211486

MC. JOSE CHRISTIAN ROMERO HERNANDEZ

Campus Tomas Aquino

"Por una juventud integrada al desarrollo de México"

Tijuana B.C. a 8 de Diciembre del 2021

Práctica evaluativa 3

Import and select dataset

```
getwd()
setwd("E:/Programas TEC/TEC/Mineria de
datos/Clone/DataMining/MachineLearning")
getwd()

dataset <- read.csv('Social_Network_Ads.csv')
dataset <- dataset[, 3:5]
```

Coding of the destination function as a factor, this will allow making the categorical data of the Purchased field to make the model.

```
dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))
```

Divide the dataset into training and test sets, creating the training and test datasets respectively.

```
library(caTools)
split <- sample.split(dataset$Purchased, SplitRatio = 0.75)
training_set <- subset(dataset, split == TRUE)
test_set <- subset(dataset, split == FALSE)
```

Feature scale, all fields are scaled relative to the Purchased field.

```
training_set[-3] = scale(training_set[-3])
test_set[-3] = scale(test_set[-3])
```

We use the Naive Bayes library to create the variable that contains the classification.

```
library(e1071)
classifier = naiveBayes(x = training_set[-3], y = training_set$Purchased)
```

Predicting the results of the test set, using the predict method and as parameters the classifier variable and the test dataset excluding the Purchased field.

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

Result:

```
> y_pred
  [1] 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
 [55] 0 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1
1 0 0 1 1 1 1 1 1 1
Levels: 0 1
```

The last step before seeing the plotted result is to make a confusion matrix. This aims to evaluate how accurate the prediction made in the previous step was. In this, the false positives and false negatives are added, thus obtaining the error percentage, that is, erroneous predictions made by the algorithm saying that a data was true or false when it was not.

```
cm = table(test_set[, 3], y_pred)
cm
```

Result:

```
> cm = table(test_set[, 3], y_pred)
> cm
   y_pred
     0  1
  0 62  2
  1  8 28
```

Viewing the results of the training set:

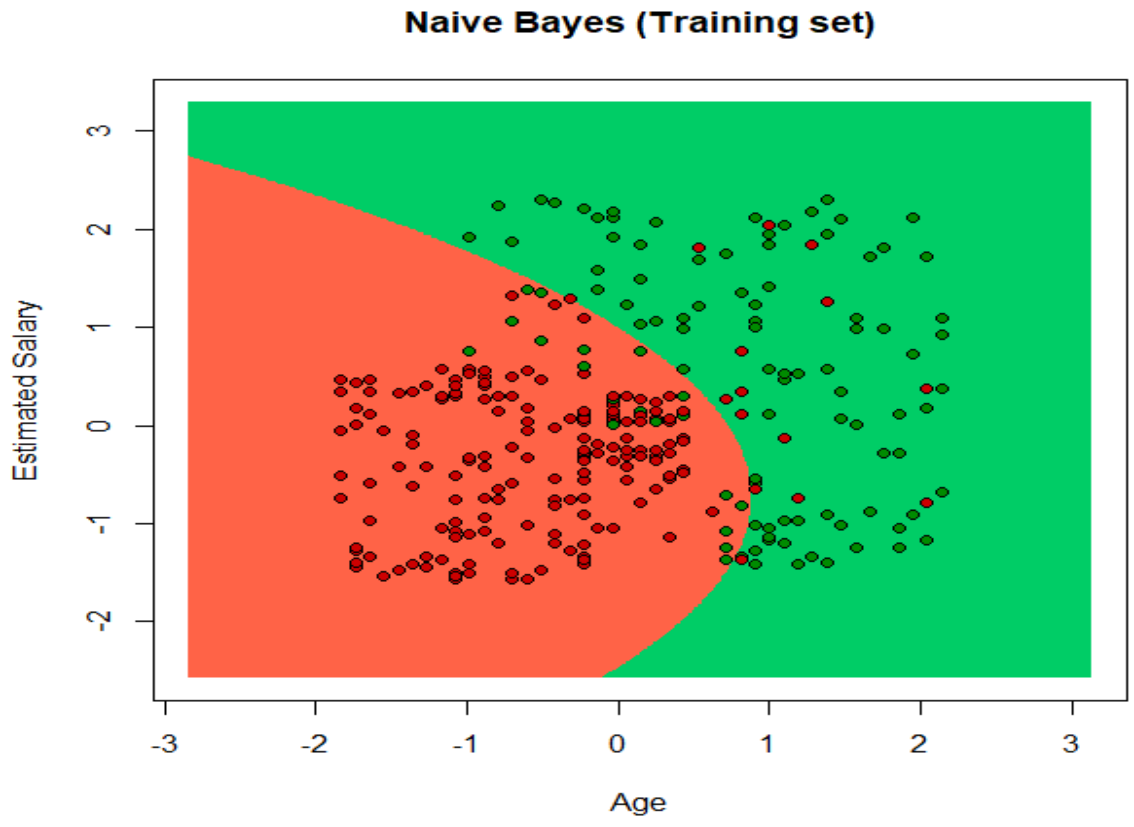
Finally we will show a graph using the "ElemStatLearn" library that allows a better visualization of the Naive Bayes model with the training data.

1. We import the library
2. In a new variable we define the data of the training dataset
3. We create 2 new variables with the transformation of the values of the "Age" field for "X1" and the "Estimated Salary" field for "X2"
4. We create a data frame "grid_set" from all the combinations of the vectors or factors provided previously
5. The names of the fields are changed to their previous ones
6. We generate a new variable "y_grid" for the prediction, where the "classifier" method is used using the values from the previous example as a parameter, in this case changing the test parameter to grid_set
7. We start the visualization of the data, defining the main name as well as the X axis and the Y axis, marking the limits of both with the values obtained from the variables X1 and X2
8. We add a contour line to the existing graph, using the "contour" method with the parameters of the sequences X1 and X2, as well as the values obtained in the prediction "y_grid"
9. Ending with the "points" methods that allow to make a change in the graphic design to visually show the separation of the categories by sections and the points as such in different colors

```
library(ElemStatLearn)
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', 'EstimatedSalary')
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'))
```

Result:
It is shown that the older the age and the purchasing power, the more likely they are to make the purchase through the advertisement, otherwise if they are young and have no resources they will not make the purchase, which makes sense in general contexts.



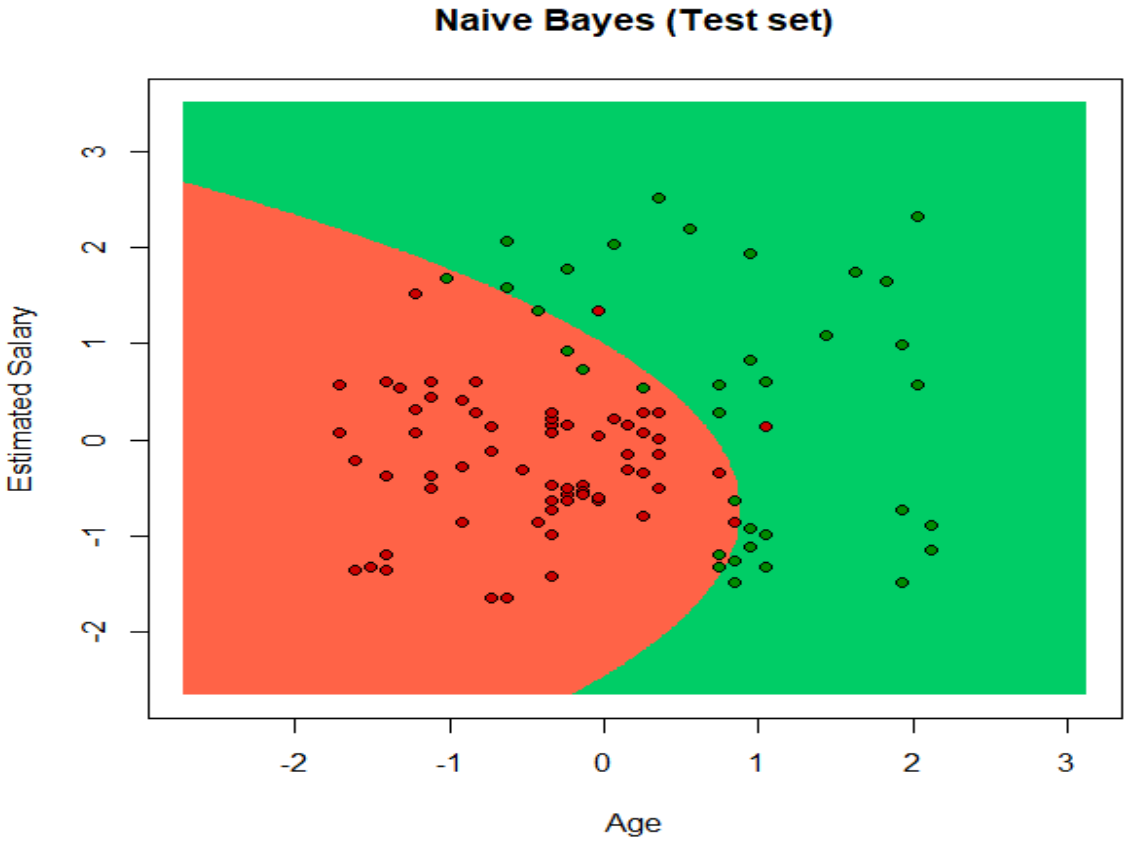
Viewing Test Set Results:
Finally we will show a graph using the "ElemStatLearn" library that allows a better visualization of the KNN model with the test data.

1. We import the library

2. In a new variable we define the data of the test dataset
3. We create 2 new variables with the transformation of the values of the "Age" field for "X1" and the "Estimated Salary" field for "X2"
4. We create a data frame "grid_set" from all the combinations of the vectors or factors provided previously
5. The names of the fields are changed to their previous ones
6. We generate a new variable "y_grid" for the prediction, where the "classifier" method is used using the values from the previous example as a parameter, in this case changing the test parameter to grid_set
7. We start the visualization of the data, defining the main name as well as the X axis and the Y axis, marking the limits of both with the values obtained from the variables X1 and X2
8. We add a contour line to the existing graph, using the "contour" method with the parameters of the sequences X1 and X2, as well as the values obtained in the prediction "y_grid"
9. Ending with the "points" methods that allow to make a change in the graphic design to visually show the separation of the categories by sections and the points as such in different colors

```
library(ElemStatLearn)
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'))
```

Resultado:
Se muestra que a mayor edad y poder adquisitivo es más propenso realizar la compra por el anuncio, en caso contrario el si es joven y no tiene recursos no realizará la compra, lo cual tiene sentido en contextos generales.

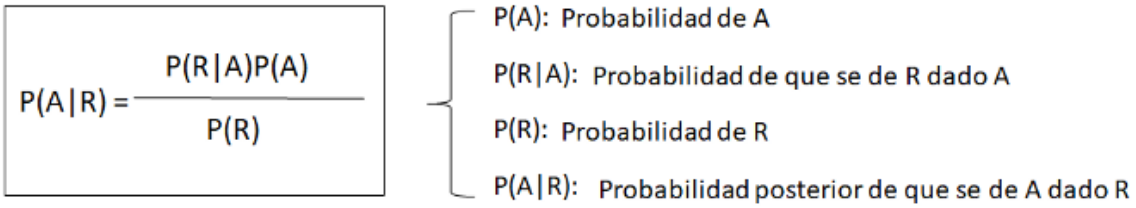

Naive Bayes (Test set)
Estimated Salary
Age
-2
-1
0
1
2
3

Naive Bayes classification model

In a broad sense, Naive Bayes models are a special class of classification algorithms for Machine Learning, or Machine Learning, as we will refer to from now on. They are based on a statistical classification technique called "Bayes's theorem."

These models are called "Naive" algorithms, or "Innocents" in Spanish. They assume that the predictor variables are independent of each other. In other words, that the presence of a certain feature in a data set is not at all related to the presence of any other feature.
They provide an easy way to build very well behaved models due to their simplicity.

They do this by providing a way to calculate the 'later' probability of a certain event A occurring, given some 'earlier' event probabilities.



Referencia:

- Roman, V. (2019, 29 abril). *Algoritmos Naive Bayes: Fundamentos e Implementación*. Medium. https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f#:%7E:text=En%20un%20sentido%20amplio%2C%20los,llamada%20%E2%80%9Cteorema%20de%20Bayes%E2%80%9D.