Instituto Tecnológico de Tijuana

Ingeniería en Sistemas Computacionales

Práctica #3

Materia: Minería de Datos

Unidad: Unidad III

Facilitador:

José Christian Romero Sánchez

Alumno:

Hernández Negrete Juan Carlos - **16212021**

Sifuentes Martinez Manuel Javier - **17212934**

Fecha:

Tijuana Baja California a 08 de Junio de 2021.

Practice #3

Analysis of data visualization in the model Decision Tree

Perform the analysis corresponding to the Decision Tree R script which must be documented in its repository by placing its visual results and your detailed description of your observations as well as the source code.

Importing the data set

The first line shown is used to load the csv, it is practical and fast compared to the option where the complete directory is specified, but it all depends on the context and need that arises.

```
dataset = read.csv ('Social_Network_Ads.csv')
dataset = dataset [3:5]
```

Encoding the target function as a factor

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

We divide the set dataset in training set and test set

```
library(caTools)
set.seed (123)
split = sample.split (dataset $ Purchased, SplitRatio = 0.75)
training_set = subset (dataset, split == TRUE)
test_set = subset (dataset, split == FALSE)
```

The Scale of functions is created

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

The adaptation of the classification of the Decision tree to the training set of our data

```
library(rpart)
classifier = rpart (formula = Purchased ~.,
                    data = training_set)
```

The results of the test set are

```
predicted y_pred = predict (classifier, newdata = test_set [-3], type =
'class')
y_pred
```

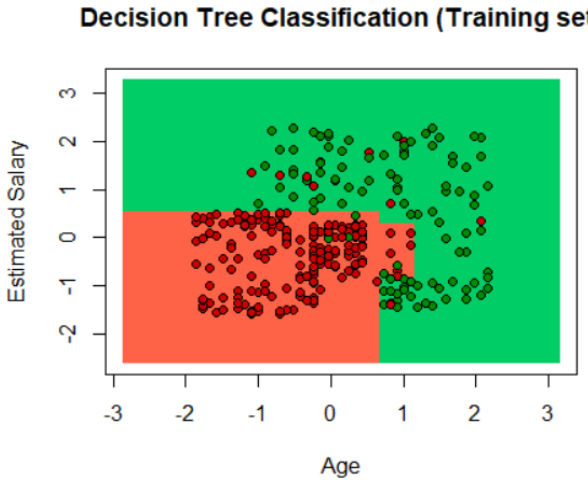
We create the confusion matrix

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

We visualize the results of the training, for this we use the elemenStatLearn library that helps us to color our graph

```
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, type = 'class')
plot (set [, -3],
     main = 'Decision Tree Classification (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'))
```

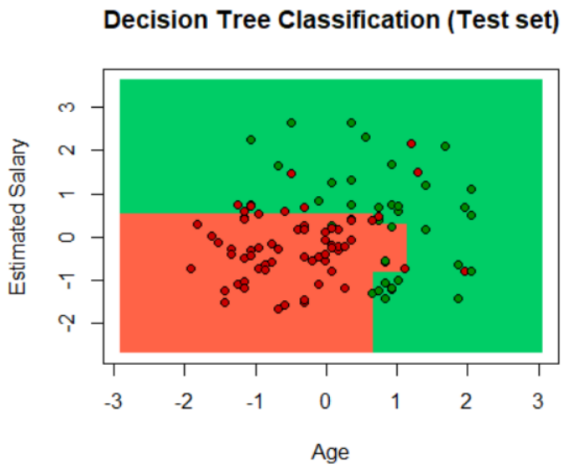
In the graph we can see that there are points and the color red and green, on the y axis we have the wage estimate and on the other we have the age for the data to be c The correct ones must be in the area of the same color, that is, the reds with the reds and the greens with the greens, otherwise they would be wrong data, we can see that in general most of the data is in its corresponding area although we have a small margin of error



Graph 1 (DTC Training Set)

We carry out the coding to make the diagram of the results of the test set

```
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, type = 'class')
plot (set [, -3], main = ' Decision Tree Classification (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'))
```



Graph 2 (DTC Test Set)