

**Tecnológico Nacional De México**

**Instituto Tecnológico De Tijuana**

**Subdirección Académica**

**Departamento de Sistemas y Computación**

**Semestre Enero - Junio 2022**

**Ingeniería Informática**

**Mineria De Datos**

**Práctica 4 - Unidad 3**

**Unidad 3**

**Perez Ortega Victoria Valeria No.18210718**

**Díaz Ruiz Uriel No.18210839**

**JOSE CHRISTIAN ROMERO HERNANDEZ**

**Tijuana, B.C. a 23 de Mayo de 2022.**

**Instalamos las lirberias ggplot2, caTools y ElemStatLearn:**

| **install.packages("ggplot2") install.packages("caTools") install.packages("ElemStatLearn")** |
| --- |

**Importamos las librerias ggplot2, caTools y ElemStatLearn:**

| **library(caTools) library(ggplot2)** |
| --- |

**Importamos nuestro set de datos:**

| **dataset = read.csv('Social\_Network\_Ads.csv')dataset = dataset[3:5]** |
| --- |

**Debemos de codificar la función de destino como factor:**

| **dataset$Purchased = factor(dataset$Purchased, levels = c(0,1))** |
| --- |

**Debemos de establecer nuestra semilla de aleatoriedad:**

| **set.seed(123)** |
| --- |

**Debemos de separar nuestro set de datos en el de entrenamiento y los de prueba:**

| **split = sample.split(dataset$Purchased, SplitRatio = 0.75) training\_set = subset(dataset, split == TRUE) test\_set = subset(dataset, split == FALSE)** |
| --- |

**Se debe de Normalizar los elementos con el siguiente comando:**

| **training\_set[-3] = scale(training\_set[-3]) test\_set[-3] = scale(test\_set[-3])** |
| --- |

**Debemos de adaptar nuestro K-NN al conjunto de entrenamiento y predicción de los resultados del conjunto de prueba:**

| **library(class) y\_pred = knn(train = training\_set[, -3], test = test\_set[, -3], cl = training\_set[, 3], k = 5, prob = TRUE)** |
| --- |

**Debemos de crear nuesta matriz de confusión:**

| **cm = table(test\_set[, 3], y\_pred)** |
| --- |

**Debemos de visualizar los resultados del set de entrenamiento y prueba:**

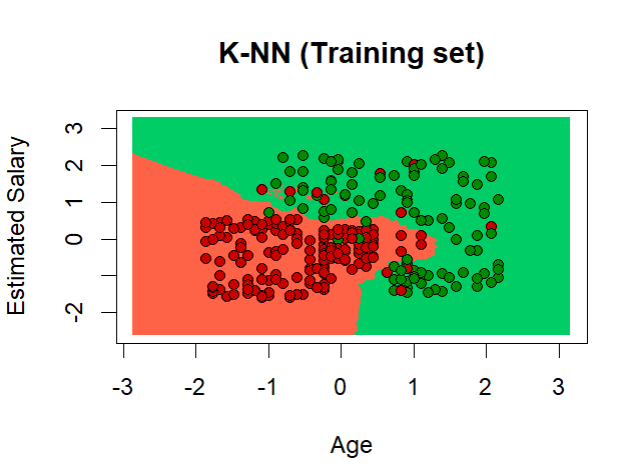
**Entrenamiento:**

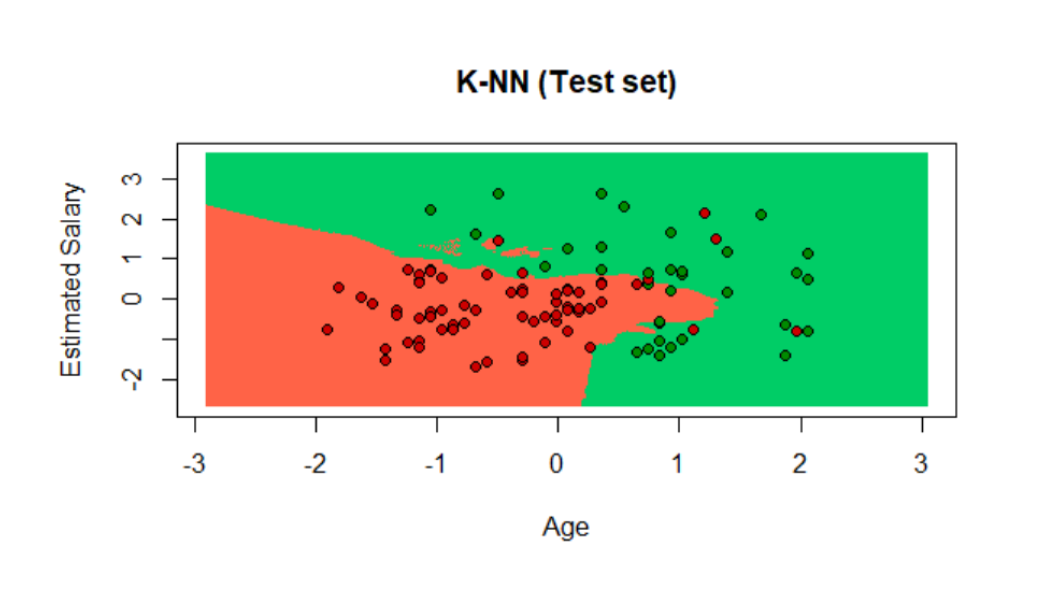
| **library(ElemStatLearn) set = training\_set X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01) X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = knn(train = training\_set[, -3], test = grid\_set, cl = training\_set[, 3], k = 5) plot(set[, -3], main = 'K-NN (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'))** |
| --- |

**Pruebas:**

| **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 = knn(train = training\_set[, -3], test = grid\_set, cl = training\_set[, 3], k = 5) plot(set[, -3], main = 'K-NN (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:**

****

****