**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*

*Práctica 4*

Landa Alvarez Ariel Nicolas #17211531

Ceron Uribe Arturo #17211506

## 

**MC. JOSE CHRISTIAN ROMERO HERNANDEZ**

Campus Tomas Aquino

Lo primero que se debe de hacer es establecer el directorio de trabajo para el proyecto en R. Utilizando “setwd” se realiza la modificación.

| *# K-Nearest Neighbors (K-NN)*  setwd("C:/Users/CORSAIR/DataminingRomero/DataMining/MachineLearning/MultipleLinearRegression") getwd() |
| --- |

Se importa el dataset y se especifica utilizar unicamente algunas columnas (3 a 5)

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

Se cambia el valor de la columna “Purchased” como tipo factor para poder utilizarse posteriormente

| *# Encoding the target feature as factor* dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1)) |
| --- |

Se divide el dataset en 2 partes, entrenamiento y prueba con un ratio de 0.75, además se especifica el seed para tener una aleatoriedad.

| *# Splitting the dataset into the Training set and Test set* *# install.packages('caTools')* 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) |
| --- |

Se escalan los valores de entrenamiento y de prueba para modelarlos y poder usarlos en el algoritmo KNN.

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

Se envian los datos de prueba y entrenamiento al algoritmo de KNN, además se establece la variable k como 5 para buscar obtener una mejor clasificación.

| *# Fitting K-NN to the Training set and Predicting the Test set results* library(class) y\_pred = knn(train = training\_set[, -3],  test = test\_set[, -3],  cl = training\_set[, 3],  k = 5,  prob = TRUE) |
| --- |

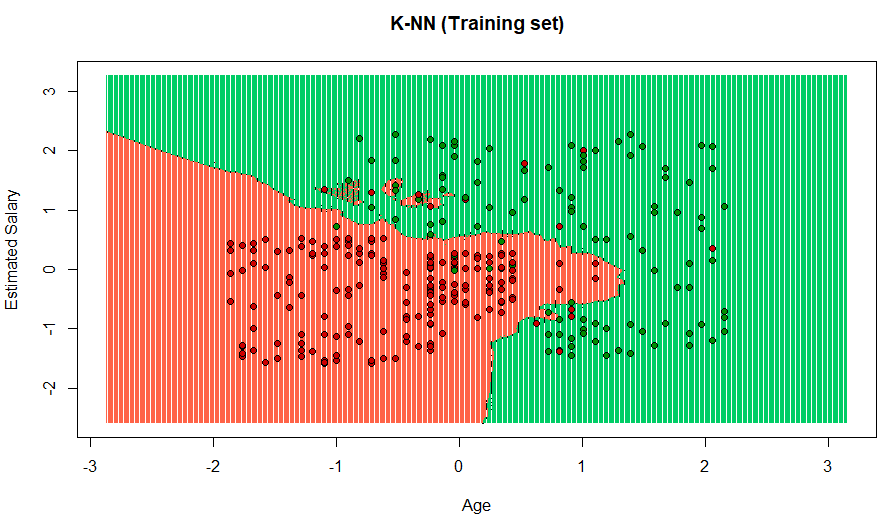
Creación de la matriz de confusión utilizando la variable de predicción creada a partir del algoritmo knn y enviando los datos de prueba.

| *# Making the Confusion Matrix* cm = table(test\_set[, 3], y\_pred) |
| --- |

Visualización de los resultados utilizando los valores de entrenamiento del algoritmo.

| *# Visualising the Training set results* install.packages('ElemStatLearn') 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')) |
| --- |

Resultados de valores de entrenamiento en algoritmo KNN (k = 5)



Visualización de los resultados utilizando los valores de prueba del algoritmo.

| *# Visualising the Test set results* 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 de valores de prueba en algoritmo KNN (k = 5)

