**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 3**

**Unidad 3  
Díaz Ruiz Uriel No.18210839**

**Perez Ortega Victoria Valeria No.18210718**

**JOSE CHRISTIAN ROMERO HERNANDEZ**

**Tijuana, B.C. a 19 de Mayo de 2022.**

**Se agrega la ruta en donde se encuentra el archivo que es donde lo va a buscar:**

getwd()

setwd("C:/Users/valer/Documents/9no Semestre/Minería de Datos/Unit3")

getwd()

**Abrimos nuestro archivo csv y nos lo va a importar:**

dataset <- read.csv(file.choose())

dataset <- dataset[, 3:5]

**Se divide el conjunto de datos en el conjunto de entrenamiento y de prueba**

**Se instala el paquete de datos de 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)

**Escala de las características**

training\_set[, 1:2] <- scale(training\_set[, 1:2])

test\_set[, 1:2] <- scale(test\_set[, 1:2])

**Ajustes de la regresión logística al conjunto de entrenamiento:**

classifier = glm(formula = Purchased ~ .,

family = binomial,

data = training\_set)

**Se predicen los resultados del conjunto de pruebas:**

prob\_pred = predict(classifier, type = 'response', newdata = test\_set[-3])

prob\_pred

y\_pred = ifelse(prob\_pred > 0.5, 1, 0)

y\_pred

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

cm

library(ggplot2)

ggplot(training\_set, aes(x=EstimatedSalary, y=Purchased)) + geom\_point() +

stat\_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)

ggplot(training\_set, aes(x=Age, y=Purchased)) + geom\_point() +

stat\_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)

ggplot(test\_set, aes(x=EstimatedSalary, y=Purchased)) + geom\_point() +

stat\_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)

ggplot(test\_set, aes(x=Age, y=Purchased)) + geom\_point() +

stat\_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)

**Se muestra una visualización de los resultados obtenidos del conjunto de 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')

prob\_set = predict(classifier, type = 'response', newdata = grid\_set)

y\_grid = ifelse(prob\_set > 0.5, 1, 0)

plot(set[, -3],

main = 'Logistic Regression (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'))

**Se muestra una visualización de los resultados obtenidos del conjunto de 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')

prob\_set = predict(classifier, type = 'response', newdata = grid\_set)

y\_grid = ifelse(prob\_set > 0.5, 1, 0)

plot(set[, -3],

main = 'Logistic Regression (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'))





