**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 2 - Unidad 3**

**Unidad 3**

**Perez Ortega Victoria Valeria No.18210718 Díaz Ruiz Uriel No.18210839**

**JOSE CHRISTIAN ROMERO HERNANDEZ**

**Tijuana, B.C. a 18 de Mayo de 2022.**

Instalamos nuestra paqueteria de ggplot y catools para poder generar las nuestra práctica:

install.packages("ggplot2")

install.packages("caTools")

library(caTools)

library(ggplot2)

Creamos un dataset para que nos pueda buscar y leer nuestro archivo .CSV :

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

dataset <- read.csv('50\_Startups.csv')

Agregamos nuestros datos categóricos:

dataset$State = factor(dataset$State,

levels = c('New York', 'California', 'Florida'),

labels = c(1,2,3))

Cargamos nuestra librería de caTools y le colocamos una semilla aleatoria:

library(caTools)

set.seed(123)

Hacemos nuestro split con las variables de training y la de test:

split <- sample.split(dataset$Profit, SplitRatio = 0.8)

training\_set <- subset(dataset, split == TRUE)

test\_set <- subset(dataset, split == FALSE)

Colocamos nuestra regresión con los datos del training y mostramos resultado en summary además de agregar la predicción en Y:

regressor = lm(formula = Profit ~ .,

data = training\_set )

summary(regressor)

y\_pred = predict(regressor, newdata = test\_set)

y\_pred

Creamos un modelo para usar la backward elimination y al final usamos el summary para mostrarlo:

regressor = lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend + State, data = dataset)

summary(regressor)

regressor = lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend, data = dataset)

summary(regressor)

regressor = lm(formula = Profit ~ R.D.Spend + Marketing.Spend, data = dataset)

summary(regressor)

regressor = lm(formula = Profit ~ R.D.Spend + Marketing.Spend, data = dataset)

summary(regressor)

y\_pred = predict(regressor, newdata = test\_set)y\_pred

Utilizamos la función de eliminación paraque pueda escoger todasr las predicciones en el modelo. Es procedimiento ya que luego borrara lo menos importante hasta encontrar algún equilibrio entre todos para que así pueda tener una manera más óptima:

backwardElimination <- function(x, sl) {

numVars = length(x) for (i in c(1:numVars))

{ regressor = lm(formula = Profit ~ ., data = x)

maxVar = max(coef(summary(regressor))[c(2:numVars), "Pr(>|t|)"]) if (maxVar > sl){ j =

which(coef(summary(regressor))[c(2:numVars), "Pr(>|t|)"] == maxVar)

x = x[, -j] } numVars = numVars - 1 }

return(summary(regressor))}

SL = 0.05

Utilizamos un training para implementar los nuevos parámetros para poder sacar otro resultado:

training\_set

backwardElimination(training\_set, SL)

Datos Finales:







