TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

NOMBRE DE LOS ALUMNOS:

RAYMUNDO HIRALES LAZARENO (N. CONTROL: 17212339)

GALAVIZ LONA OSCAR EDUARDO (N.CONTROL: 17212993)

Carrera: Ingeniería Informática

Semestre: 9no

MATERIA: Minería de datos

PROFESOR: JOSE CHRISTIAN ROMERO HERNANDEZ

TRABAJOS: Practica 3

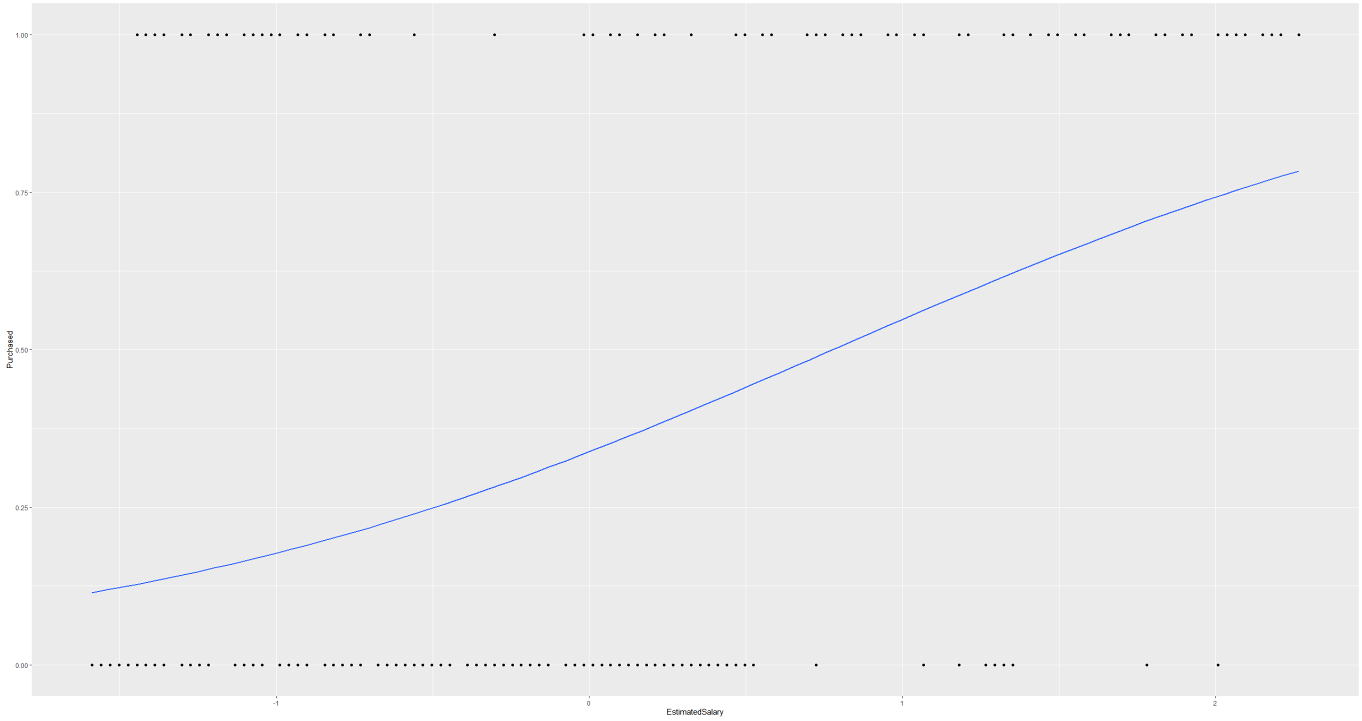
FECHA: 28/11/21

Visualización de datos de modelo de machine learning regresión lineal:

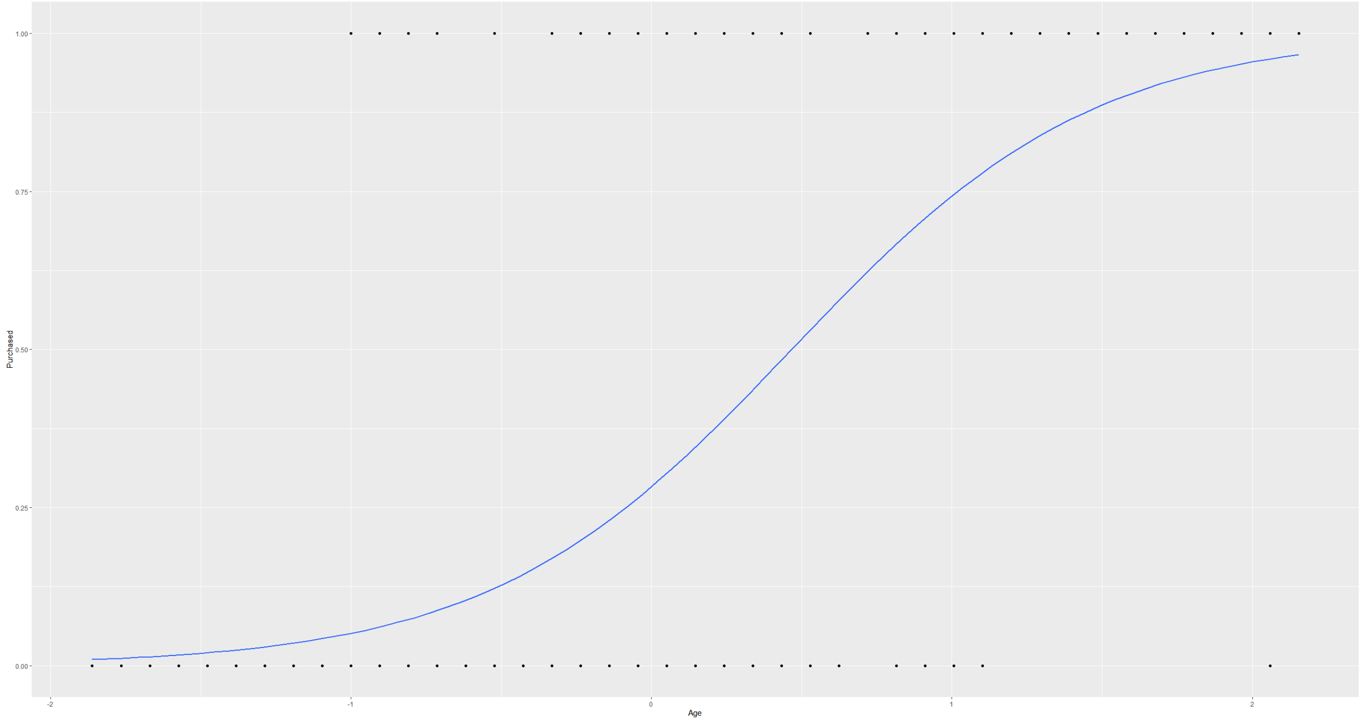
Usando una fuente de publicidad en redes sociales

Código:

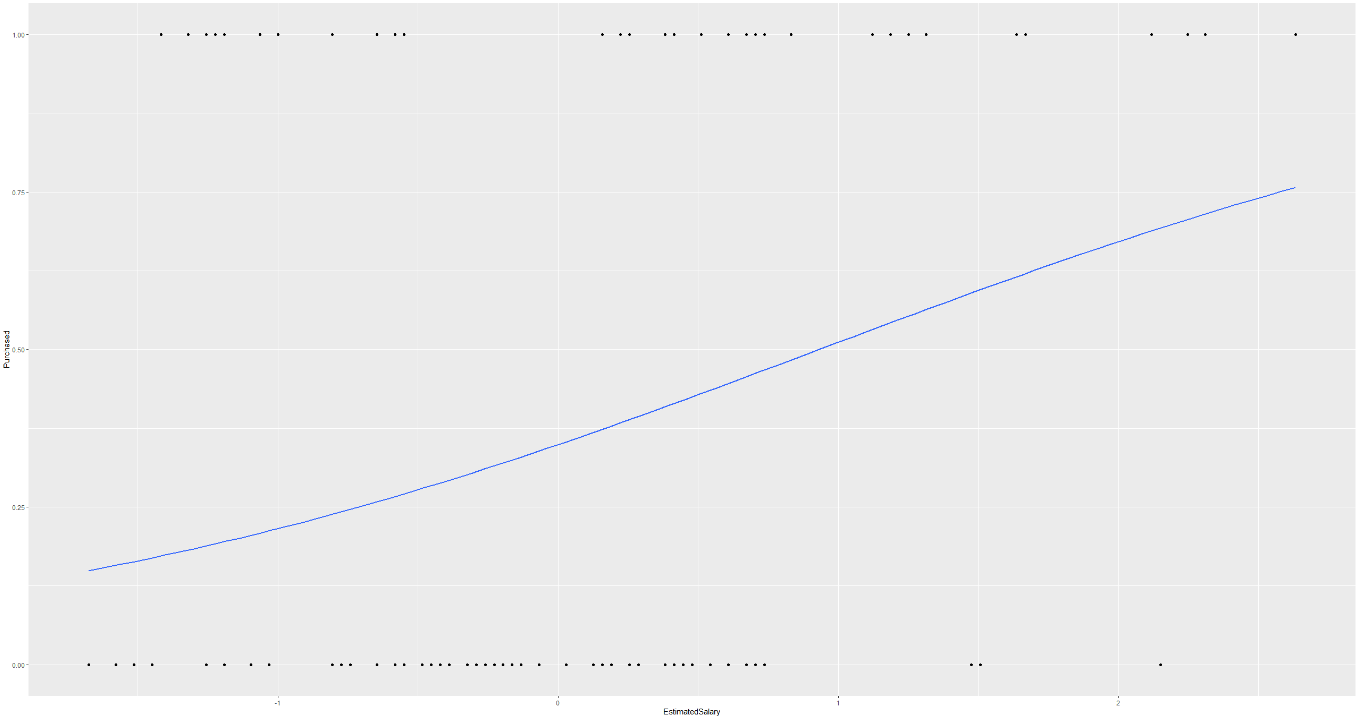
~~~
# Importando el dataset
dataset <- read.csv('Social_Network_Ads.csv')
dataset <- dataset[, 3:5]

# partiendo el dataset dentro del Training set y el 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)

# incluir el escalado
training_set[, 1:2] <- scale(training_set[, 1:2])
test_set[, 1:2] <- scale(test_set[, 1:2])

# acomodando la regresion logistica en el Training set
classifier = glm(formula = Purchased ~ .,
                 family = binomial,
                 data = training_set)

# prediccion de los resultados del Test set
prob_pred = predict(classifier, type = 'response', newdata = test_set[-3])
prob_pred
y_pred = ifelse(prob_pred > 0.5, 1, 0)
y_pred

# creando la metrica de confusion
cm = table(test_set[, 3], y_pred)
cm

#
library(ggplot2)
```R
ggplot(training_set, aes(x=EstimatedSalary, y=Purchased)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
~~~



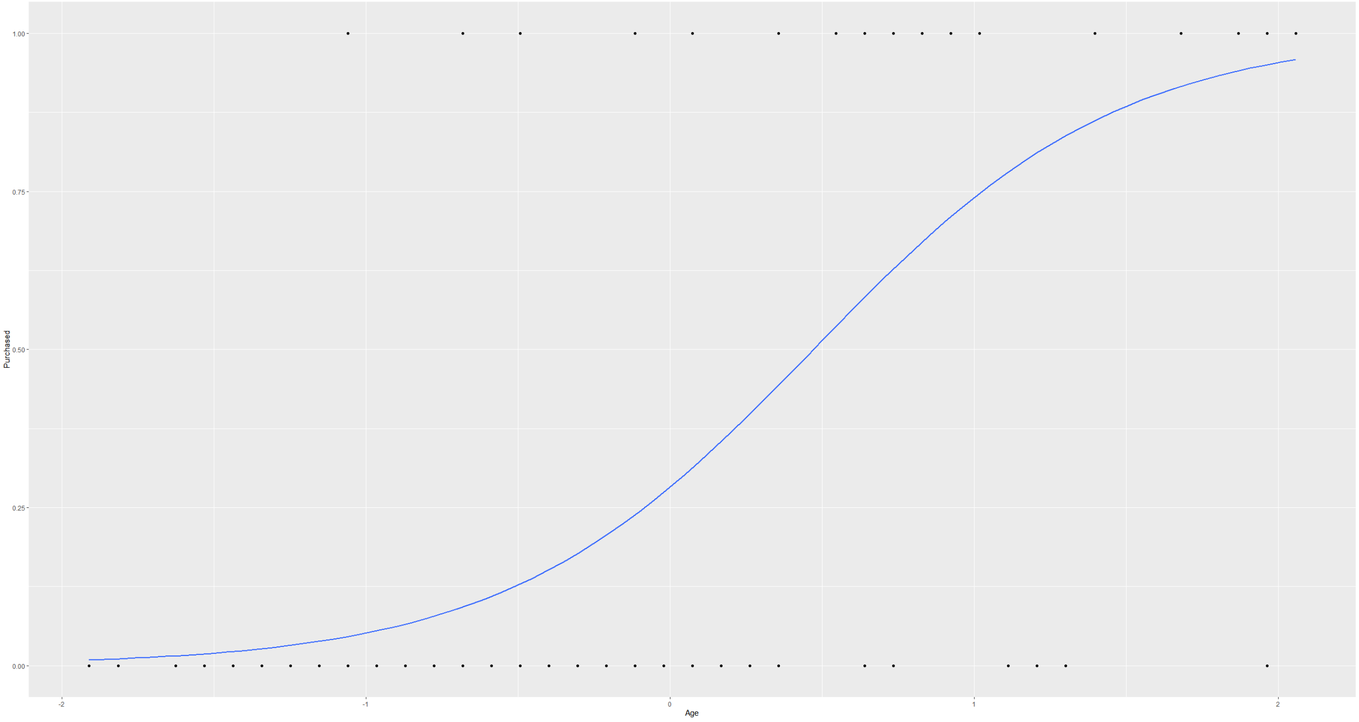
```r
ggplot(training_set, aes(x=Age, y=Purchased)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```



```r
ggplot(test_set, aes(x=EstimatedSalary, y=Purchased)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```

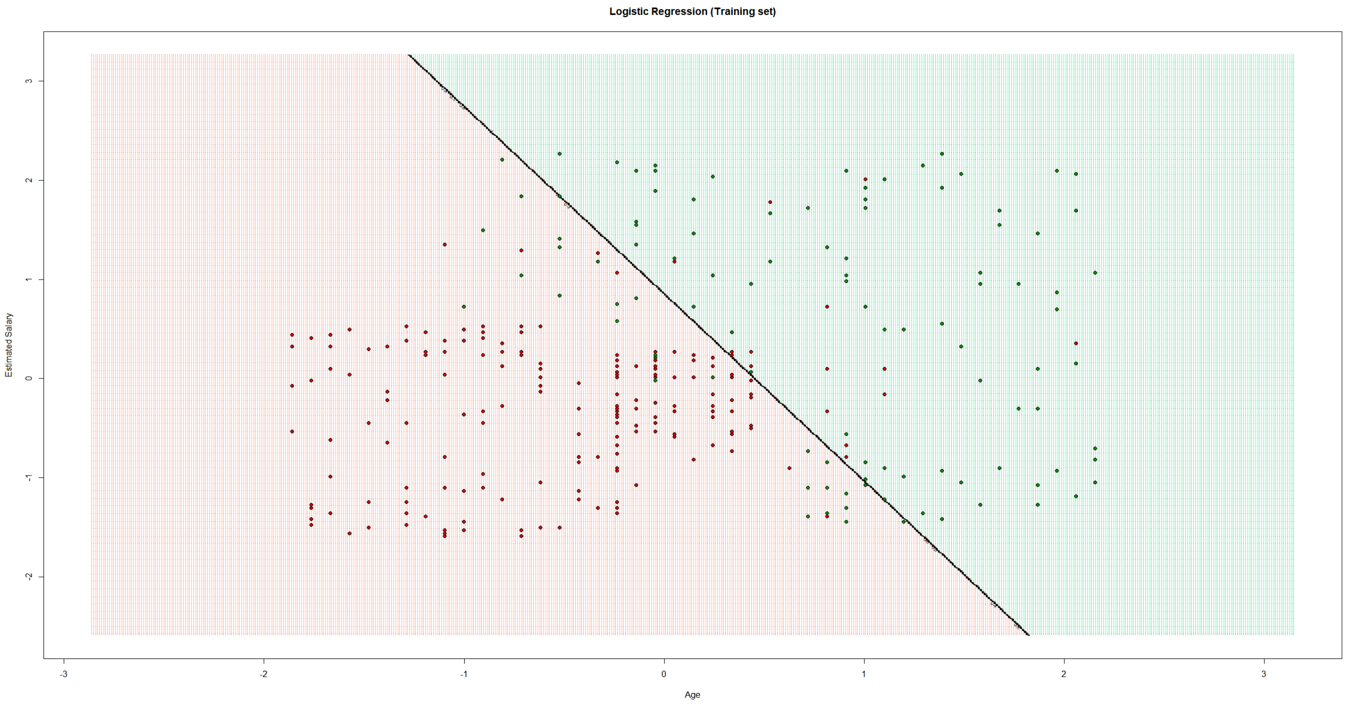


```r
ggplot(test_set, aes(x=Age, y=Purchased)) + geom_point() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
```

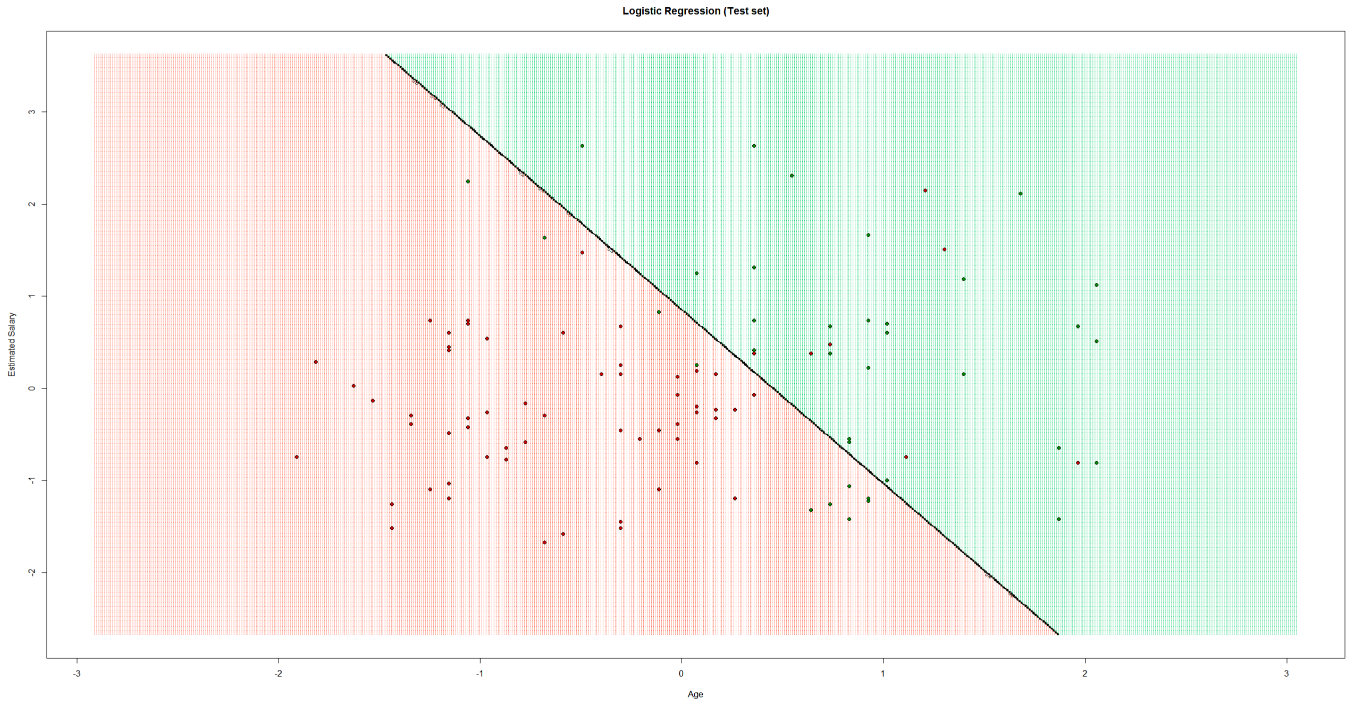


```r
# Visualization the Training set result
library(ElemStatLearn)
set = training_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)
colnames(grid_set) = c('Age', 'EstimatedSalary')
```

```
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'))
```



```
# Visualizando los ressultados del 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')
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'))
```



Analisis y conclusion

Se puede denotar la regresion lineal en el ejercicio a partir de las imagenes presentadas lo que demuestra el uso de este tema para el analisis de mineria de datos.