



TECNOLÓGICO NACIONAL DE MÉXICO



**INSTITUTO TECNOLÓGICO DE TIJUANA**

**CARRERA**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**MATERIA**

**MINERÍA DE DATOS**

**TAREA**

**PRÁCTICA #3, UNIDAD #3**

**FECHA ENTREGA**

**19/05/2022**

**ALUMNO(A)**

**HOWARD HERRERA ERWIN #18210716**

**PÉREZ LÓPEZ ALICIA GUADALUPE #18210514**

**DOCENTE**

**JOSE CHRISTIAN ROMERO HERNANDEZ**





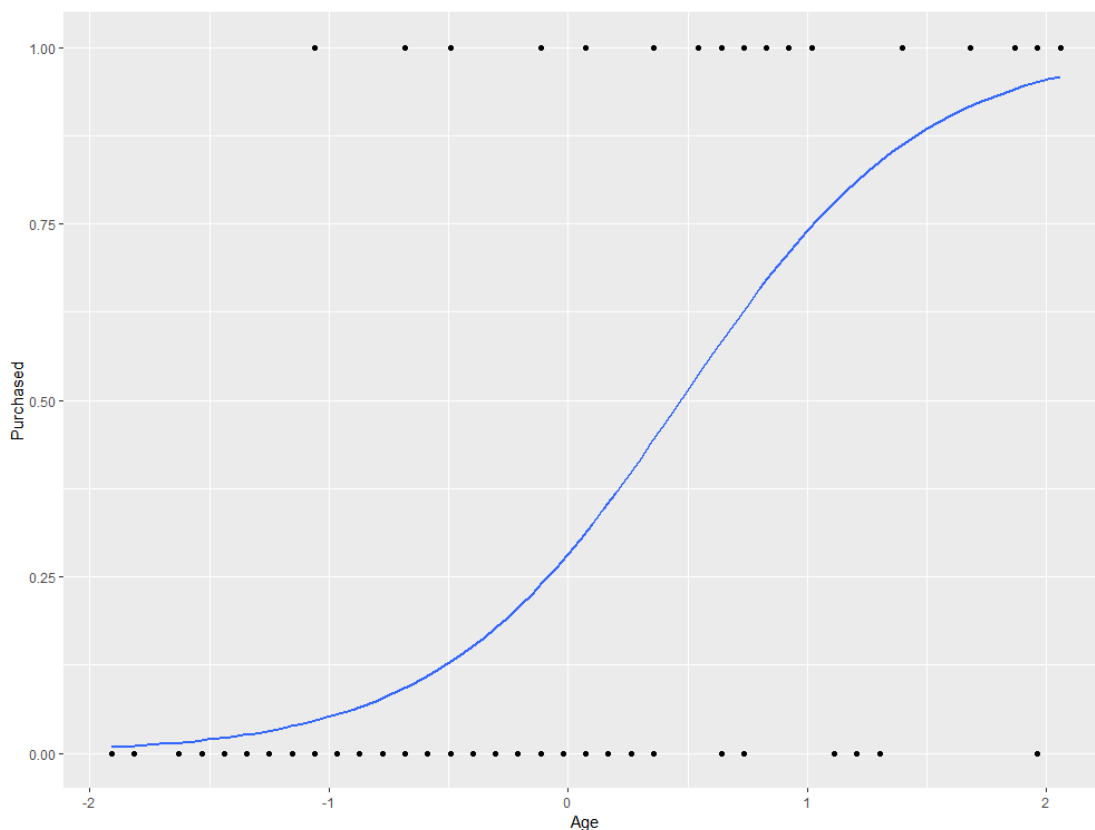
Luego hacemos la confusión matrix, generamos un pequeño vector, en el que solicitamos ciertos valores y le pedimos que lo muestre en forma de 3\*3.

```
> cm = table(test_set[, 3], y_pred)
> cm
```

|   | y_pred<br>0 | 1  |
|---|-------------|----|
| 0 | 57          | 7  |
| 1 | 10          | 26 |

A continuación generamos los siguientes gráficos, en los que solicitamos los datos de prueba y datos de entrenamiento

```
> library(ggplot2)
> ggplot(training_set, aes(x=EstimatedSalary, y=Purchased)) + geom_point() +
+   stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
`geom_smooth()` using formula 'y ~ x'
>
> ggplot(training_set, aes(x=Age, y=Purchased)) + geom_point() +
+   stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
`geom_smooth()` using formula 'y ~ x'
>
> ggplot(test_set, aes(x=EstimatedSalary, y=Purchased)) + geom_point() +
+   stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
`geom_smooth()` using formula 'y ~ x'
>
> ggplot(test_set, aes(x=Age, y=Purchased)) + geom_point() +
+   stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE)
`geom_smooth()` using formula 'y ~ x'
```

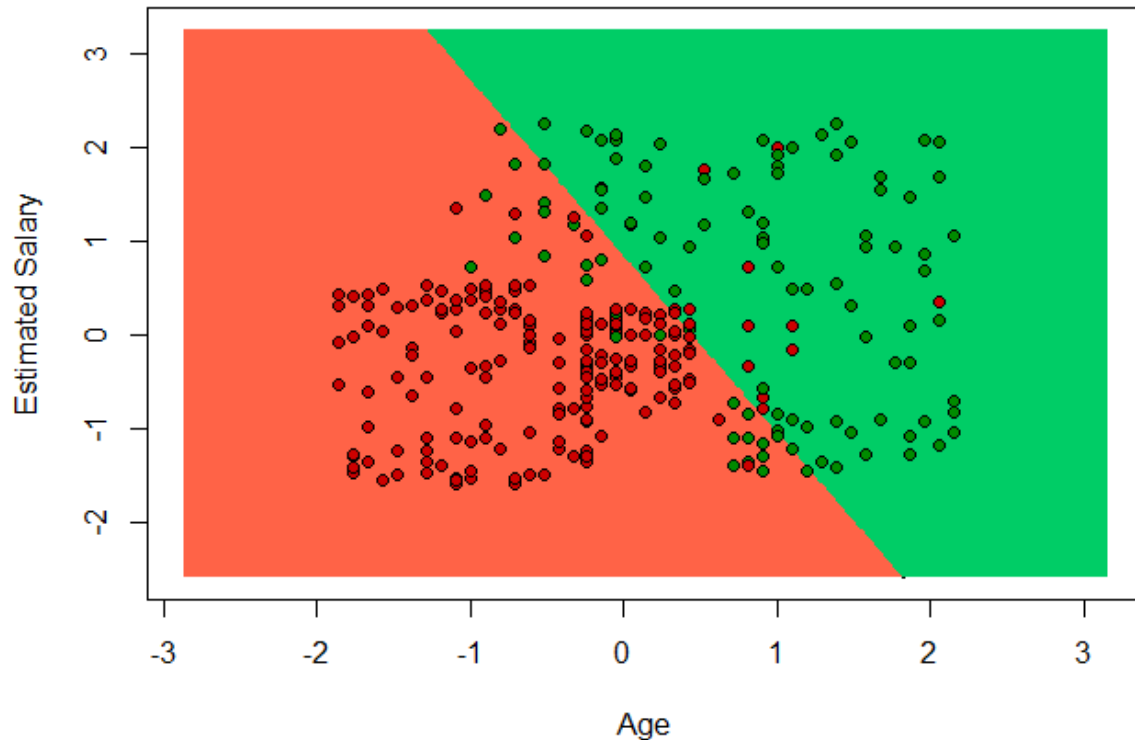


Dentro del gráfico solicitamos la edad, las compras y el número de compras con el salario estimado. Como podemos ver el punto más alto es aquel que es el mayor número de compras realizadas.

A continuación hacemos el siguiente gráfico en el que generamos lo que ya es la regresión logística

```
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'))
```

### Logistic Regression (Training set)



En este gráfico hacemos una comparativa de los datos de formación teniendo en cuenta el salario y la edad estimados, todos valores que son menores que los que están en rojo y más en verde, ya que por ejemplo puedes tener una edad adulta pero puede ser que tu situación económica no esté muy bien estarás en rojo, pero si tienes una edad temprana como mínimo de 19 años y tu estabilidad financiera está bien estarás en verde.

En este gráfico que generamos hacemos la misma comparación que la anterior pero con los valores de prueba

```
# 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')
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'))
```

**Logistic Regression (Test set)**

