**Tecnológico Nacional de México**

**Instituto Tecnológico de Tijuana**

**Subdirección Académica**

**Departamento de Sistemas y Computación**

**SEMESTRE:**

Febrero-Julio 2021

**CARRERA:**

Ingeniería en Sistemas Computacionales e Ingeniería en Tecnologías de la Información y Comunicaciones

**MATERIA Y SERIE:**

Minería de Datos

BDD-1703 TI9A

**UNIDAD A EVALUAR:**

Unidad III

**NOMBRE DEL TRABAJO:**

**Practica #1**

¿Qué es una tabla de confusión?

**NOMBRE Y NÚMERO DE CONTROL DE LOS INTEGRANTES:**

Rodriguez Medrano Marco Antonio 17210635

**NOMBRE DEL DOCENTE:**

José Christian Romero Hernández

**Tijuana, Baja California Jueves 25 de Mayo de 2021**

**Instrucciones**

el docente le pidió al alumno que documente el siguiente código propiciado por el docente José Christian Romero Hernández.

primero para trabajar con esta parte del código anteriormente se trabajó con los datos de un archivo .csv que en total son 400 datos y después se dividieron los datos en 100 prueba y 300 de entrenamiento y se trabajó con esos datos de prueba y entrenamiento hasta la parte que estaré documentando.

la librería ElemStatLearn es una librería creada por los autores del siguiente libro: <https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12_toc.pdf>. Para resolver este tipo de problemas de logistic regression.

desues con el comando “set = training\_set” estaremos indicando que trabajaremos con los datos de entrenamiento solamente, con los comandos siguientes X1 y X2 generamos dos secuencias separadas utilizando para la variable X1 la columna 1 y la columna 2 para la variable X2. y después con el comando grid\_set = expand.grid(X1, X2) uniremos las dos secuencias en una solo tabla y con el comando colnames le otorgamos los nombres a las columnas de la tabla grid\_set. una vez hecho esto estaremos listos para hacer la predicción clasificatoria de tipo response con los datos del grid\_set y seran guardadas en la variable prob\_set.

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'))  
  
Bien ahora con el comando y\_grid = ifelse(prob\_set > 0.5, 1, 0) estaremos clasificando todos los datos guardados en prob\_set dependiendo de si el valor es mayor a 0.5 se le otorgara el valor de 1 de lo contrario se le asigna el 0.

después con el comando plot mostraremos una dispersión de puntos con los límites de las variables X1 y X2. a continuación se muestra el resultado.

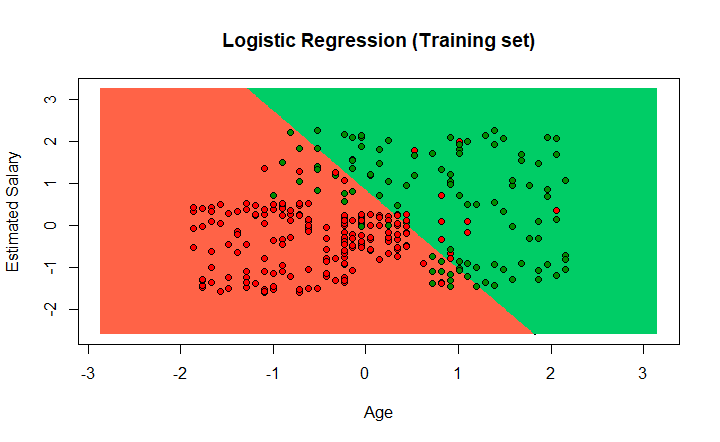


Ahora con el comando contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) crearemos una línea vertical que representa la matriz de los datos de las variables X1 yX2 como se muestra a continuación y ya para terminar le daremos formato a la dispersión de puntos para que se observe que conjunto de puntos pertenece a quien.



La modificación se hará utilizando los comandos siguientes y podemos observar las dos regiones y los puntos que están dentro de dicho regiones además claro ver a quién pertenecen

points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'springgreen3', 'tomato'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))



Para el resto del código se seguirán los mismos pasos anteriormente mencionados pero con la diferencia que ahora se trabajaría con los datos 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'))

Resultados del código que utiliza test\_set





