TECNOLÓGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

SEMESTRE FEBRERO-JUNIO 2022

MATERIA:

Minería de datos.

UNIDAD 3

Práctica #4

K Vecinos mas cercano K-NN

DOCENTE:

JOSE CHRISTIAN ROMERO HERNANDEZ

ALUMNO:

López Higuera Saúl Alfredo #18210493

Munguía silva Edgar Geovanny #17212344

Tijuana BC 21 de mayo del 2022

Introducción.

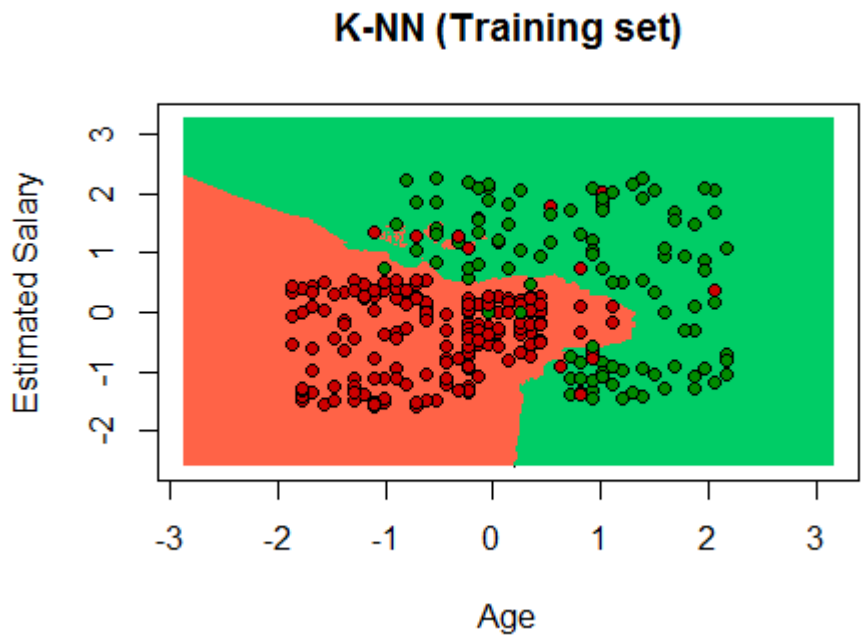
El algoritmo de k-vecinos más cercanos es un clasificador de aprendizaje supervisado no paramétrico, que usa la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, suponiendo que se pueden encontrar puntos similares cerca uno del otro.

Código.

```
# K-Nearest Neighbors (K-NN)

# Importing the dataset
dataset = read.csv(file.choose())
dataset = dataset[3:5]

# Encoding the target feature as factor
dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))

# Splitting the dataset into the Training set and 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)

# Feature Scaling
training_set[-3] = scale(training_set[-3])
test_set[-3] = scale(test_set[-3])

# Fitting K-NN to the Training set and Predicting the Test set
results
library(class)
y_pred = knn(train = training_set[, -3],
             test = test_set[, -3],
             cl = training_set[, 3],
             k = 5,
             prob = TRUE)

# Making the Confusion Matrix
cm = table(test_set[, 3], y_pred)
```

```
# Visualising the Training set results
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')
y_grid = knn(train = training_set[, -3], test = grid_set, cl =
training_set[, 3], k = 5)
plot(set[, -3],
     main = 'K-NN (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'))

# 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')
y_grid = knn(train = training_set[, -3], test = grid_set, cl =
training_set[, 3], k = 5)
plot(set[, -3],
     main = 'K-NN (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'))
```

Salida.



Conclusión.

En la siguiente imagen podemos ver el análisis correspondiente al conjunto de prueba, con este modelo podemos ver cual de los registros es el que más se acerca a un punto determinado, en este caso podemos ver que tan cerca están los salarios de los empleados con respecto a los demás y sus respectivas edades.