Instituto Tecnológico de Tijuana

Ingeniería en Sistemas Computacionales

Práctica #2

Materia: Minería de Datos

Unidad: Unidad III

Facilitador:

José Christian Romero Sánchez

Alumno:

Hernández Negrete Juan Carlos - **16212021**

Sifuentes Martinez Manuel Javier - **17212934**

Fecha:

Tijuana Baja California a 01 de Junio de 2021.

Practice #2

Data visualization analysis in the KNN (K-Nearest Neighbors) model

Make the analysis corresponding to the R script of K-Nearest Neighbors (K-NN) which must be documented in its repository by putting in it its visual results and its detailed description of its observations as well as the source of the code.

Importing the dataset

To load the csv, the first line shown is used, it is practical and fast compared to the option where the complete directory is specified, but it all depends on the context and need presented.

```
dataset = read.csv('Social_Network_Ads.csv')
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

```
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)
```

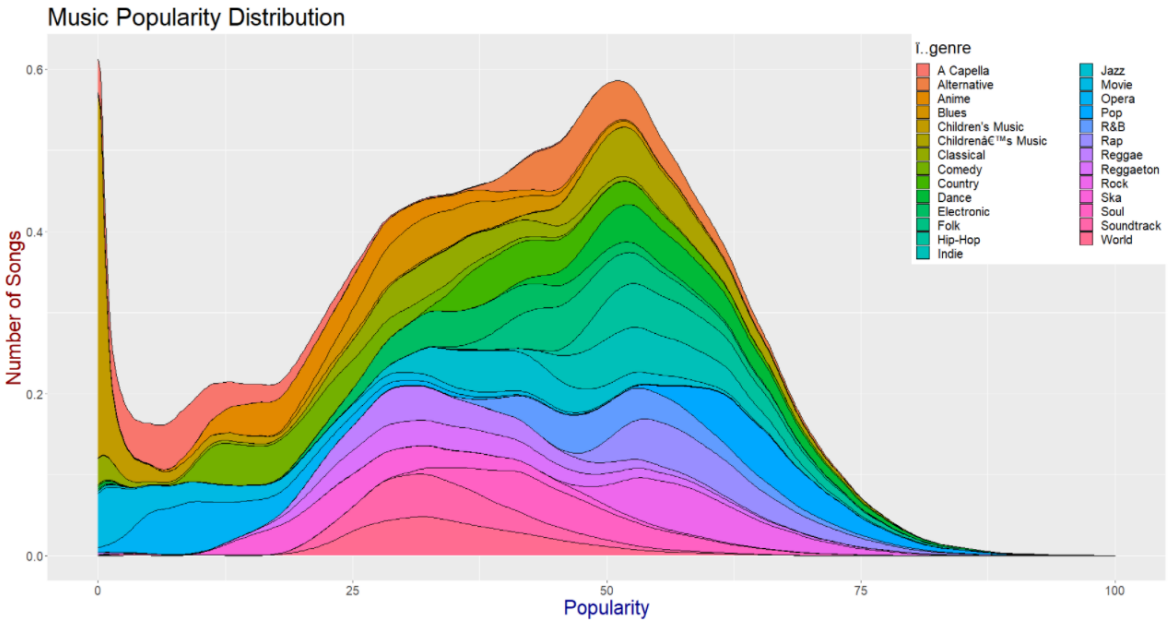
Making the Confusion Matrix

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

Styles are applied to the graph

```
h +
  xlab("Popularity") +
  ylab("Number of Songs") +
  ggtitle("Music Popularity Distribution") +
  theme(axis.title.x = element_text(color = "DarkBlue", size=25),
        axis.title.y = element_text(color = "DarkRed", size=25),
        axis.text.x = element_text(size = 15),
        axis.text.y = element_text(size = 15),
        legend.title = element_text(size = 20),
        legend.text = element_text(size = 15),
        legend.position = c(1,1),
        legend.justification = c(1,1),
        plot.title = element_text(color = "Black",
                                  size = 30,
                                  family = "Courier"))
```

Density Chart Image

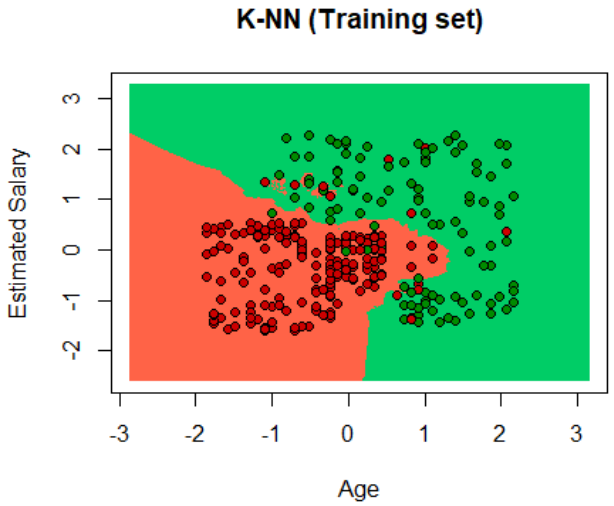


Graph-1

We visualize the results of the training sessions, for this we use the elemenStatLearn library that helps us to color our graph

```
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'))
```

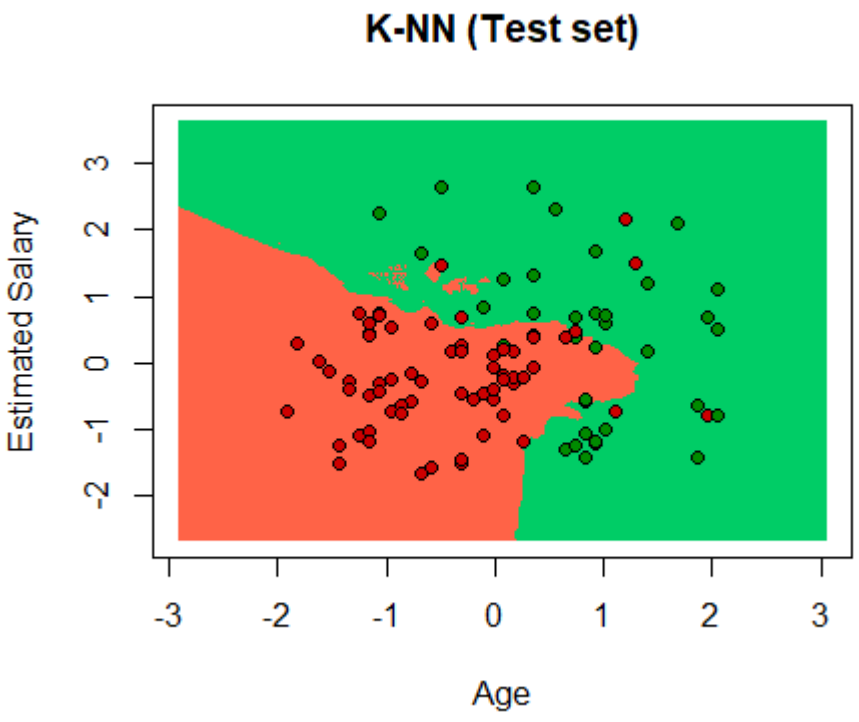
In the graph we can see that there are points and the color red and green, on the y axis we have the estimate of wages and on the other we have the age for the data to be correct they must be in the area of the same color, that is, the red ones with the reds and the greens with the greens otherwise they would be erroneous data, we can see that in general most of the data is in its corresponding area although we have a small margin of error



Graph-2

We carry out the coding to make the diagram of the results of the 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')
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'))
```



Graph-3