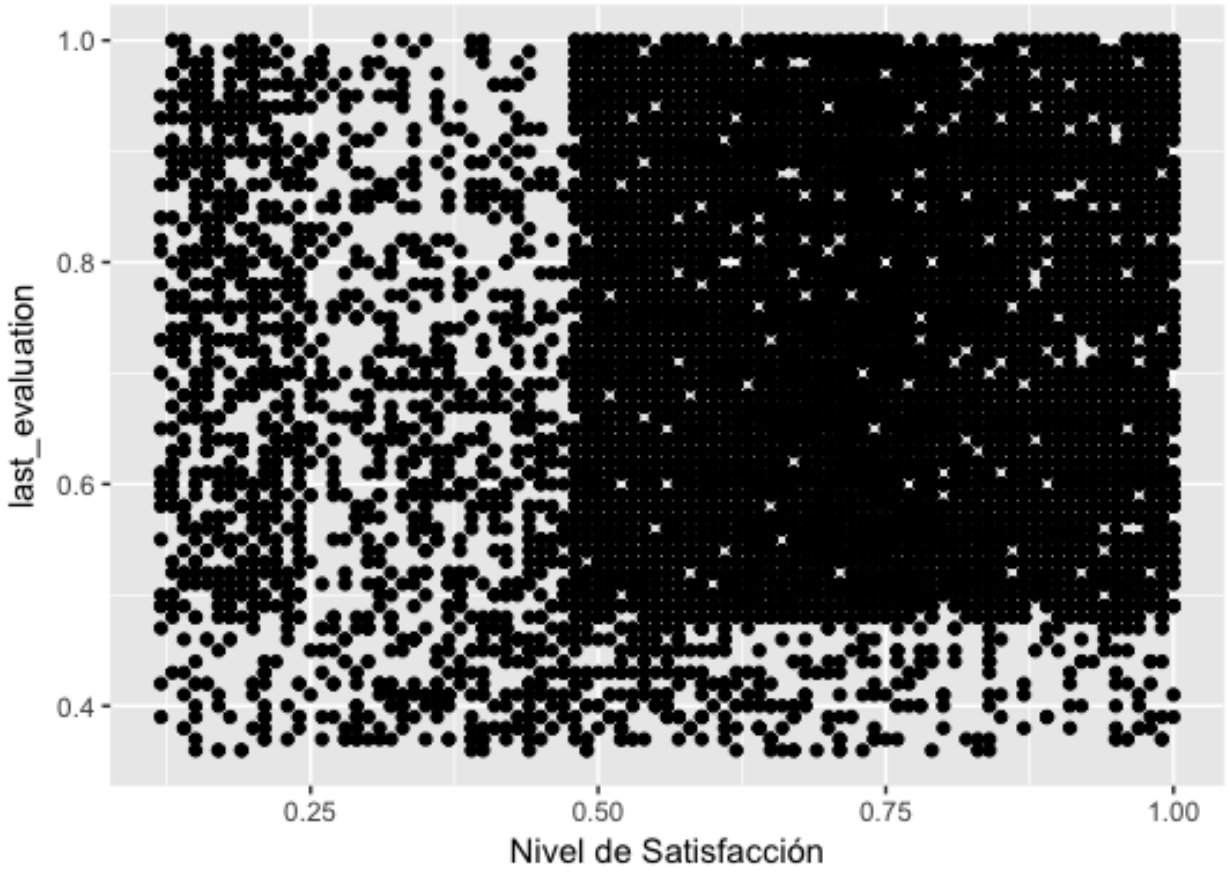
Data Set Recursos Humanos

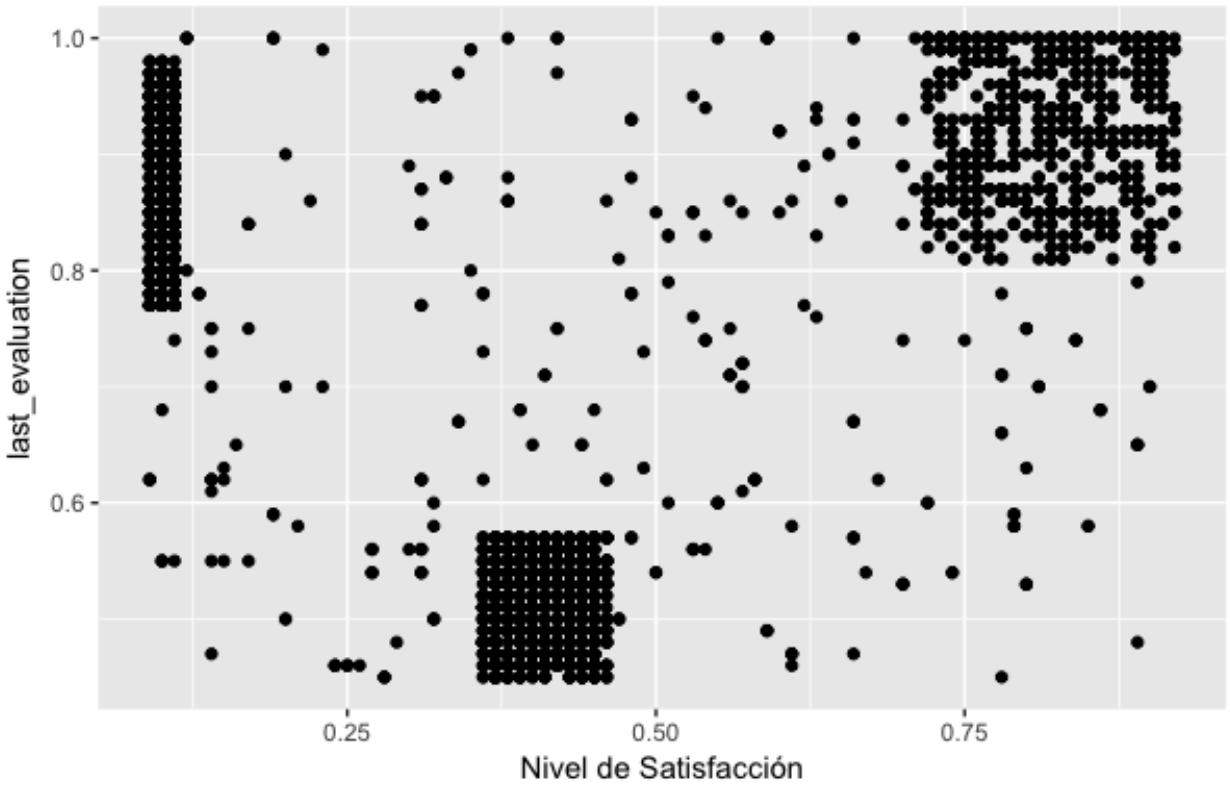
```
library(class)
library(gmodels)
library(formattable)
library(ggplot2)
HR_comma_sep <- read.csv('HR_comma_sep.csv', stringsAsFactors = FALSE)
```

Los valores del DataSet son simulados y estos se obtuvieron de la página de Kaggle.

El DataSet de Recursos Humanos contiene 14999 datos, las columnas se describen a continuación:

- Nivel de satisfació dentro de la empresa que esta entre 0 y 100%
- Calificación en la última evaluación que esta entrre 0 y 100
- Número de proyectos
- Hrs mensuales en promedio
- Antiguedad
- Accidentes de trabajo
- Promoción en los últimos 5 años
- Activo,Inactivo: los que actualmente laboral y los que ya renunciaron
- Departamento al que pertenecen laboralmete
- Salario, se representa por nivel bajo, medio y alto





Se pretende predecir a los empleados que son potenciales a renunciar.

KNN

Para identificar la relaciones entre los empleados y suponiendo que cada empleado sea diferente, observamos que algunas variables no pueden ser incluidas como vienen en el conjunto original, para ello se requiere normalizar los datos; en ente caso las variables salario y si el empleado es activo o no, son variables categoricas.

- La columna salary tiene tres etiquetas, salario alto, salario medio y salario bajo.
- La columna left tiene dos categoriass; el estado del empleado actualmente, que es activo e inactivo.

Normalización de los datos

Se quiere identificar la relación entre los empleados usando KNN, con la expectativa de predecir si un activo es activo o inanctivo de acuerdo a las variables del conjunto. Los valores de las variables deben ser homogeneas, pero se observa que las columnas

- Número de proyectos, que esta entre un rango [2, 7]
- Hrs mensuales en promedio, que esta entre un rango [96, 310]
- Antiguedad, que esta entre un rango [2, 10]

no son homogeneas y a estas se les aplica la función normalizar $fn = (x - min(x))/(max(x) - min(x))$.

Ya que el DataSet esta normalizado se le aplica KNN.

Para ello se obtinen valores de entrenamiento y de prueba. * Para los valores de Entrenamiento se toma un rango de [1, 10000] * Para los valores de Prueba se toma el rango de [10001, 14999]

Al modelo de KNN se le pasan los dos conjuntos, arroja un conjunto de predición.

Tabla de validación cruzada

Se toma el conjunto de predición y el conjunto de prueba, haciendo una comparación del modelo de KNN vs conjunto de prueba.

```
formattable(head(HR_comma_sep))
```

satisfaction_level

last_evaluation

number_project

average_montly_hours

time_spend_company

Work_accident

left

promotion_last_5years

sales

salary

0.38

0.53

2

157

3

0

1

0

sales

low

0.80

0.86

5

262

6

0

1

0

sales

medium

0.11

0.88

7
272
4
0
1
0
sales
medium
0.72
0.87
5
223
5
0
1
0
sales
low
0.37
0.52
2
159
3
0
1
0
sales
low
0.41
0.50
2
153
3
0
1
0

sales

low

```
# La columna Salario y Departamento son categoricas convertimos Salario a factor
HR_comma_sep$salary <- factor(HR_comma_sep$salary, levels=c("low","medium","high"))

# La columna Left se categoriza como avtivo con 0 e inanctivo con 1
HR_comma_sep$left <- factor(HR_comma_sep$left,levels=c(0,1),labels=c("activo","inactivo"))
```

```
# Partición de datos por la columna left
#renunciaron <- filter(HR_comma_sep, left > 0)
#activos <- filter(HR_comma_sep, left < 1)
#head(renunciaron)

#colnames(HR_comma_sep)
#table(HR_comma_sep$sales)
#table(HR_comma_sep$salary)

#table(HR_comma_sep$sales,HR_comma_sep$salary) #todos
#table(renunciaron$sales,renunciaron$salary)
#table(activos$sales,activos$salary)

#summary(HR_comma_sep[c("satisfaction_level","last_evaluation","number_project","average_montly_hours",
```

```
# Normalizar las columas
normalizar <- function(x){
  return ((x-min(x))/(max(x)-min(x)))
}
```

```
columnas_normal <- as.data.frame(lapply(HR_comma_sep[1:5],normalizar))
head(columnas_normal)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1         0.31868132        0.265625            0.0            0.2850467
## 2         0.78021978        0.781250            0.6            0.7757009
## 3         0.02197802        0.812500            1.0            0.8224299
## 4         0.69230769        0.796875            0.6            0.5934579
## 5         0.30769231        0.250000            0.0            0.2943925
## 6         0.35164835        0.218750            0.0            0.2663551
##   time_spend_company
## 1              0.125
## 2              0.500
## 3              0.250
## 4              0.375
## 5              0.125
## 6              0.125
```

```
etiquetas = HR_comma_sep$left
```

```
# Partición de los conjunto de entrenamiento y el de prueba tomando los datos normalizados
entrenamiento <- columnas_normal[1:10000,]

prueba <- columnas_normal[10001:14999,]
```

```
# Asignación de etiquetas, con el DataSet original
etiquetas_entrenamiento <- etiquetas[1:10000]
```

```r
etiquetas_prueba <- etiquetas[10001:14999]
```

```r
# función KNN
modelo_prediccion <- knn(entrenamiento, prueba, etiquetas_entrenamiento,k=25)

CrossTable(x=etiquetas_prueba, y=modelo_prediccion, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  4999
##
##
##                  | modelo_prediccion
## etiquetas_prueba |    activo |  inactivo | Row Total |
## -----------------|-----------|-----------|-----------|
##           activo |      3282 |       146 |      3428 |
##                  |     0.957 |     0.043 |     0.686 |
##                  |     0.959 |     0.093 |           |
##                  |     0.657 |     0.029 |           |
## -----------------|-----------|-----------|-----------|
##         inactivo |       141 |      1430 |      1571 |
##                  |     0.090 |     0.910 |     0.314 |
##                  |     0.041 |     0.907 |           |
##                  |     0.028 |     0.286 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |      3423 |      1576 |      4999 |
##                  |     0.685 |     0.315 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```r
# Tabla de valores distintos en K
k_values = c(1,5,10,15,20,25)
Falsos_Positivos = c(117,82,102,121,134,146)
Falsos_Negativos = c(0,137,141,141,141,141)

tabla_K = data.frame(k_values,Falsos_Positivos,Falsos_Negativos)
formattable(tabla_K)
```

k_values

Falsos_Positivos

Falsos_Negativos

1

117

0

5

82

137

10

102

141

15

121

141

20

134

141

25

146

141