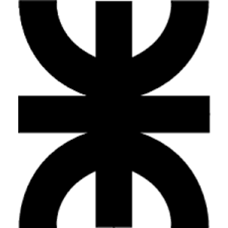
**Universidad Tecnológica Nacional**

Facultad Regional Resistencia



Ingeniería en Sistemas de Información

**Sistema de Inteligencia de Negocio y Soporte de las decisiones**

**Trabajo Práctico Final – Parte 4**

**Grupo Nro. 5**

**Alumnos:**

* Benitez Peressi, Gonzalo Facundo
* Tascón Gustavo Alejandro

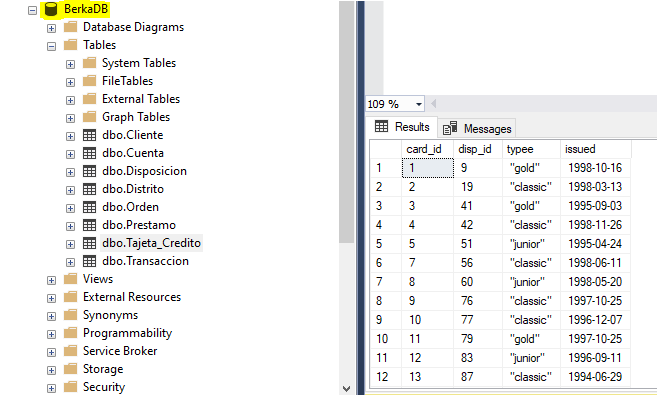
- Ciclo Lectivo 2021 –

**Trabajo Práctico Final – Parte 4: Dataminig**

**Paso 1: Conseguir los datos**

Los diferentes algoritmos de minería de datos requieren de un dataset, que contendrá un conjunto de valores, cada uno asociado con una variable y una observación. Crear este dataset no es una tarea sencilla, se describirán los pasos para poder crearlo.

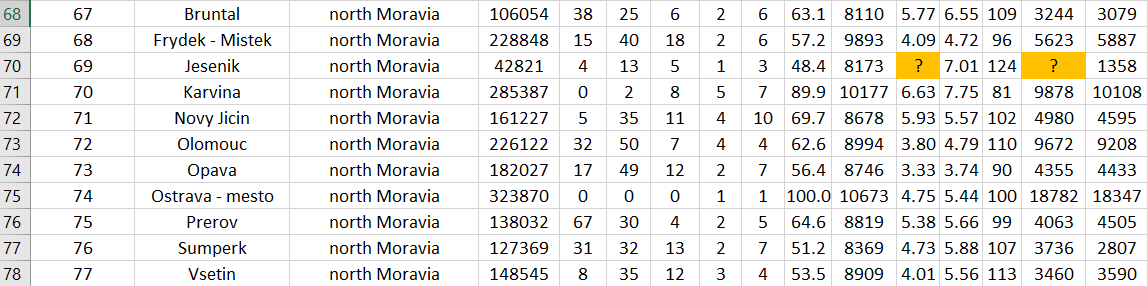
Para poder armar el dataset creamos una nueva base de datos transaccional según el modelo suministrado y realizando la carga de los archivos de entrada.



Para la creación del dataset se realizará por medio de la creación de una consulta cuyo resultado será el dataset a utilizar por el algoritmo de minería de datos. La decisión de cuáles son los variables a incluir en el dataset estará pensado en función de la variable a predecir que nos dará la respuesta a la consulta planteada en primer lugar. En general se tratará de brindar la mayor cantidad de variables para que sea el algoritmo de minería de datos, el que evalué la “importancia” de esa variable, que tal vez escape a la lógica del ser humano al no poder ver la combinatoria de relaciones entre las variables y como esto impacta en la variable a predecir.

**Paso 2: Preparar los datos**

Partimos analizando los archivos de entrada suministrados y leyendo el diccionario de datos provisto para cada uno de los atributos. En esta breve exploración nos damos cuenta de que los archivos de entrada no poseen valores nulos o valores faltantes, la excepción es district.txt el cual presenta en el registro 70, dos valores inválidos(?): tasa de desempleo '95 y No. de los crímenes cometidos en el 95:

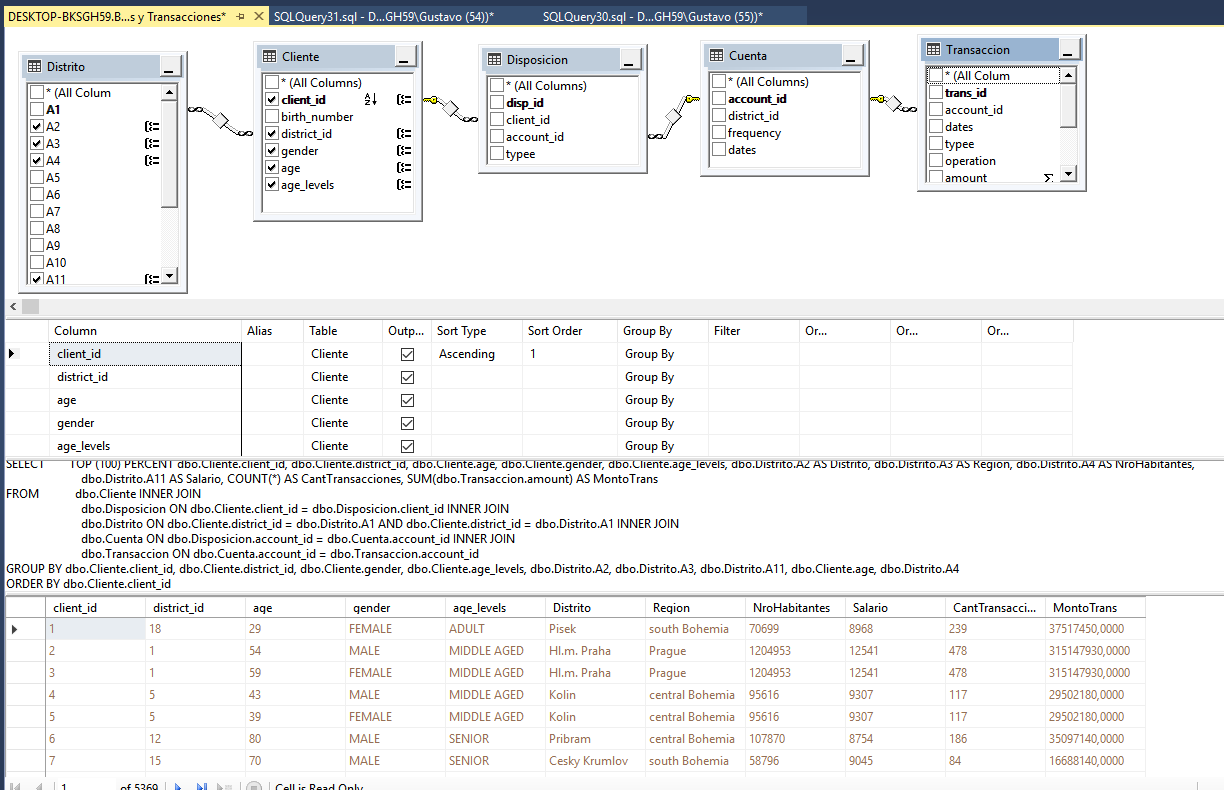


Se toma la decisión estimar esos valores inexistentes por los promedios de la tasa de desempleo '95 y No. de los crímenes cometidos en el 95 de las regiones de north Moravia que corresponden a la misma región de los valores faltantes.

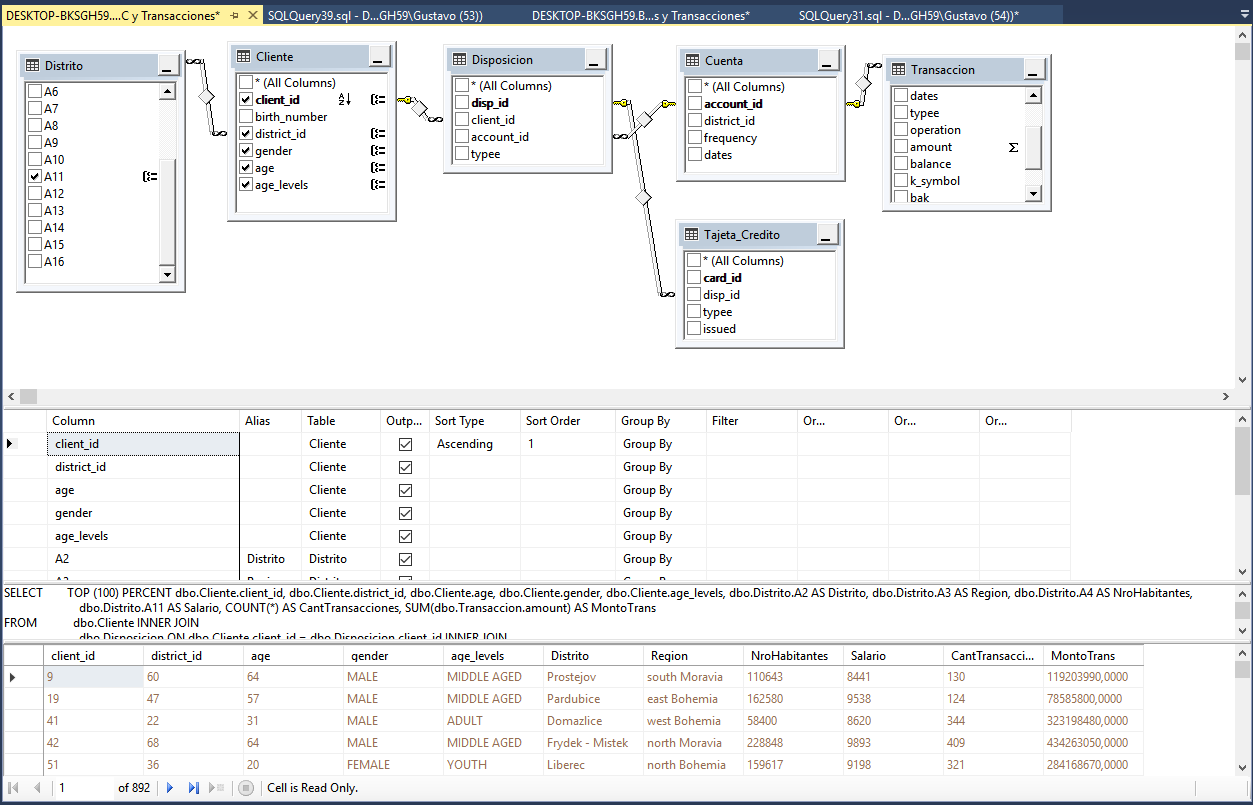
**Paso 3: Obtener el Dataset**

Para obtener el dataset hemos creado dos vistas:

La primera vista [Clientes y Transacciones] incluye las tablas Distrito, Cliente, Disposición, Cuenta y Transacciones. Con esta vista podremos obtener el listado de clientes con su información (age, age\_levels, distrito), información del distrito (nombre, región, número de habitantes, salario promedio) y, además, información de la cantidad de transacciones y monto total de las mimas.

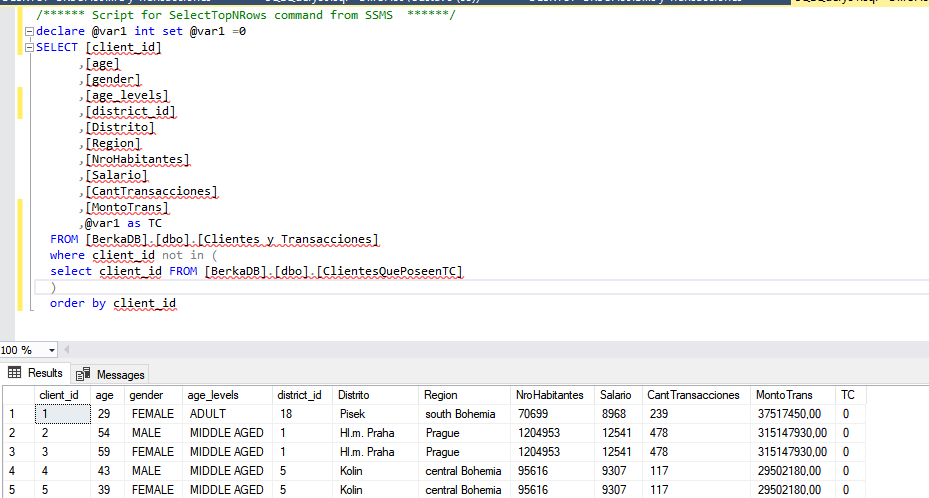


La segunda vista [ClientesQuePoseenTC] incluye todas las mimas tablas e información que la primera, con la particularidad que se agrega la tabla Tarjeta de Crédito. Básicamente, con esta vista podremos obtener un listado de los clientes que poseen tarjeta de crédito con la misma información que la primera vista.

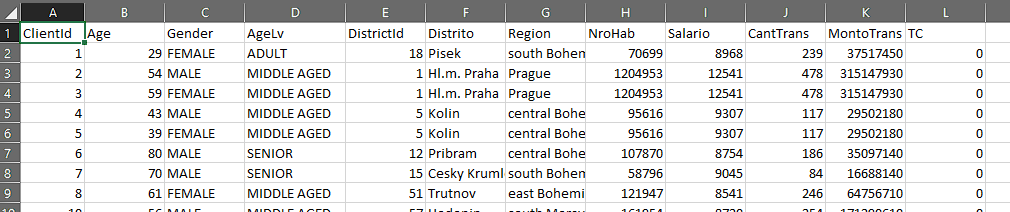


Para obtener el dataset final,

* Primero se realiza un select de la vista 2 [ClientesQuePoseenTC] y se agrega una variable TC=1 (indicando que el cliente posee tarjeta de crédito). Luego se exportan dicho listado.
* Segundo se extrae de la vista 1 [Clientes y Transacciones] el resultado de la vista 2 [ClientesQuePoseenTC], agregando una variable TC= 0 (indicando que el cliente no posee tarjeta de crédito).



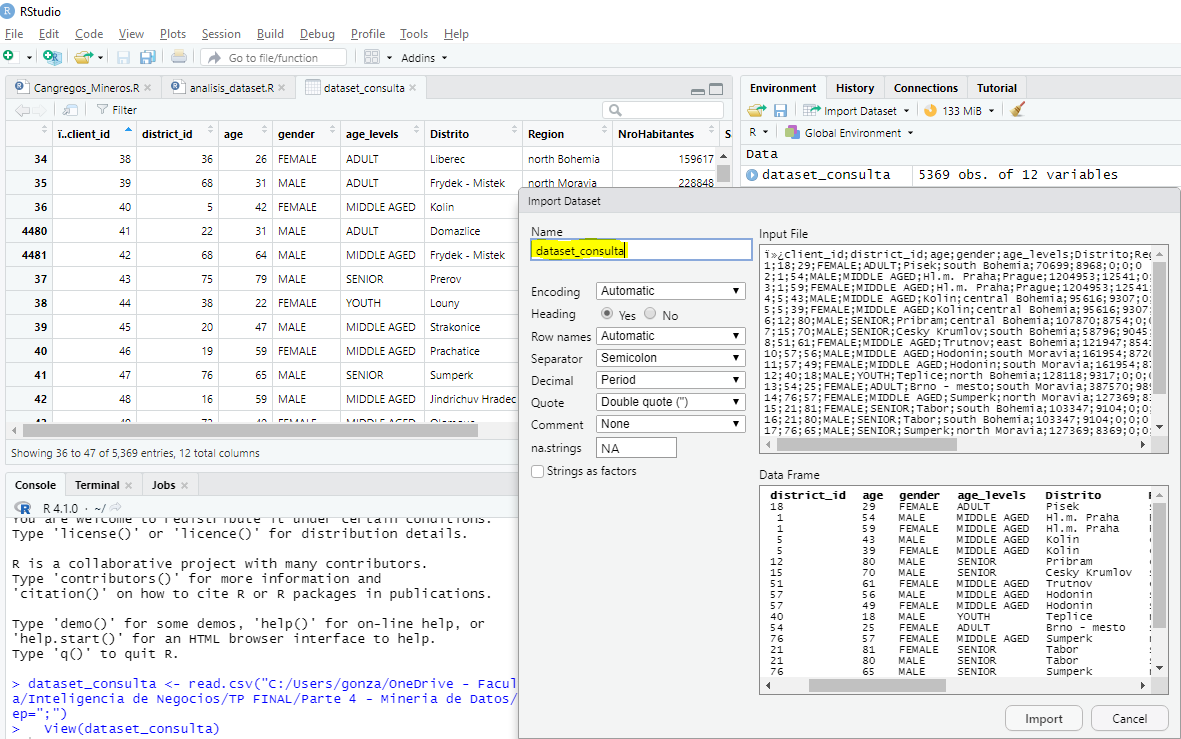
* Por ultimo unimos ambos listados y se conforma el dataset a utilizar.



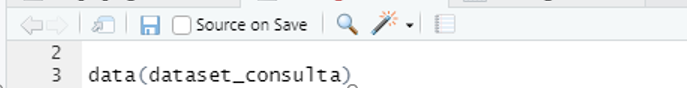
Como se ve en este dataset de ejemplo se listan los clientes que tienen y no tienen tarjeta de crédito, cual es el distrito donde viven, su edad, su rango etario, su género, cual es el salario medio del distrito donde viven, el nro. de habitantes del lugar donde viven, la cantidad de transacciones y monto de transacciones registradas hasta el momento.

**Se utilizará R para seguir realizando el análisis exploratorio sobre dataset obtenido y entender mejor las variables seleccionadas**, posteriormente aplicará un algoritmo de minería de datos para responder la consulta planteada y evaluar los resultados obtenidos:

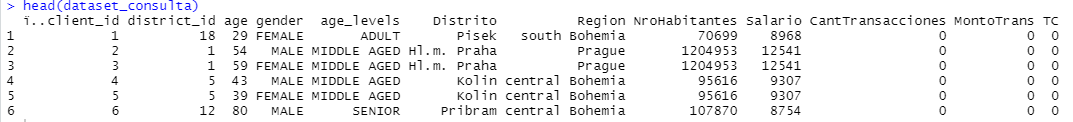
El primer paso es importar el dataset\_consulta a Rstudio:



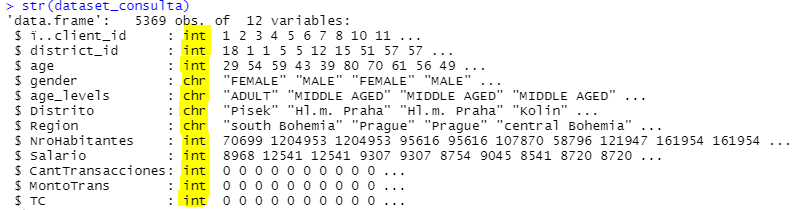
Cargamos el dataset\_consulta:



Evaluamos que los datos se hayan cargado correctamente:

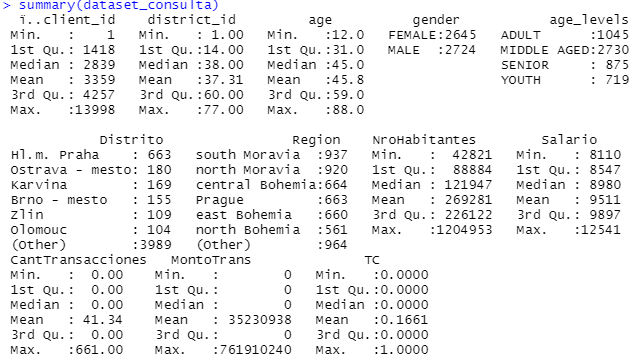


Evaluamos los tipos de datos presentes en nuestro dataset\_consulta:

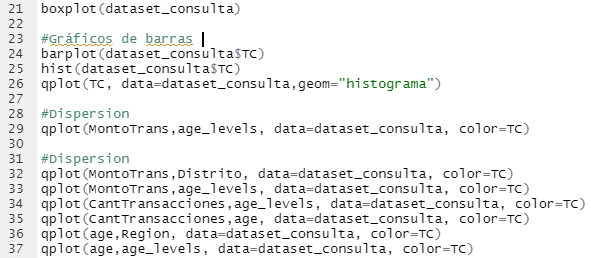


Comprobamos que los cambios se hayan realizado correctamente.

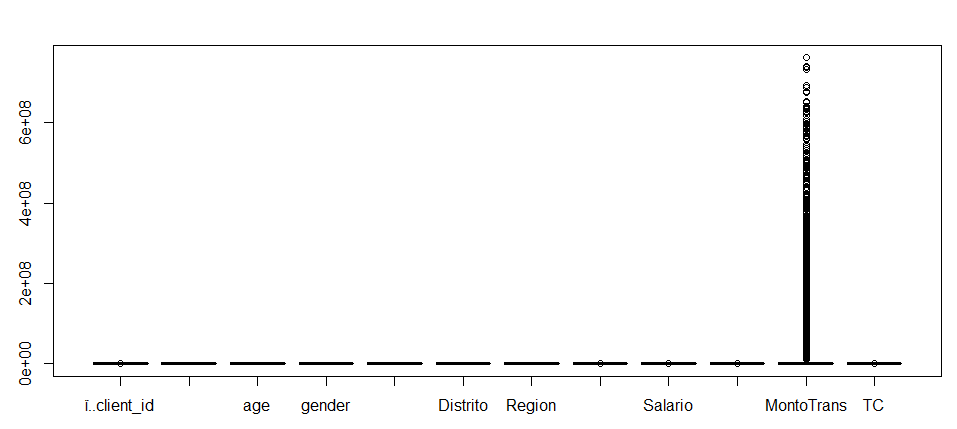
Se evalúan los valores máximos, mínimos, media de las variables tratando de entender mejor los datos y detectar anomalías, por ejemplo, por un error de carga (datos fuera de escala o proporción) que tal vez requieran un tratamiento particular:



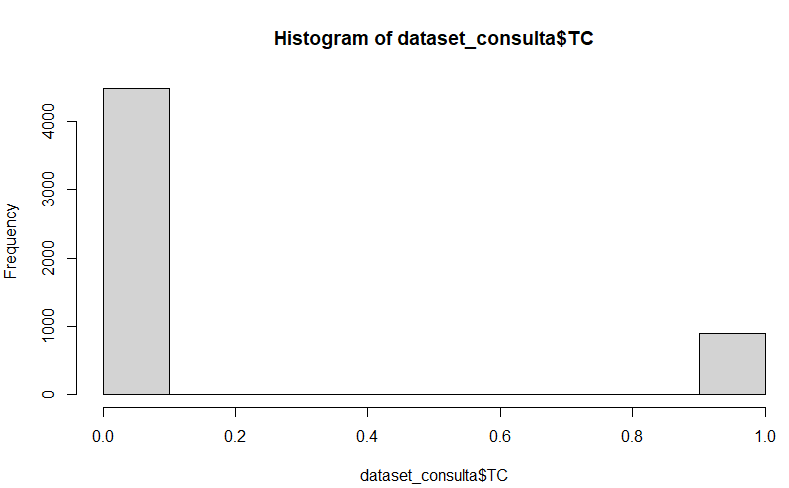
Se realizan diferentes gráficos para comprender mejor el dataset, se muestra el código utilizado:



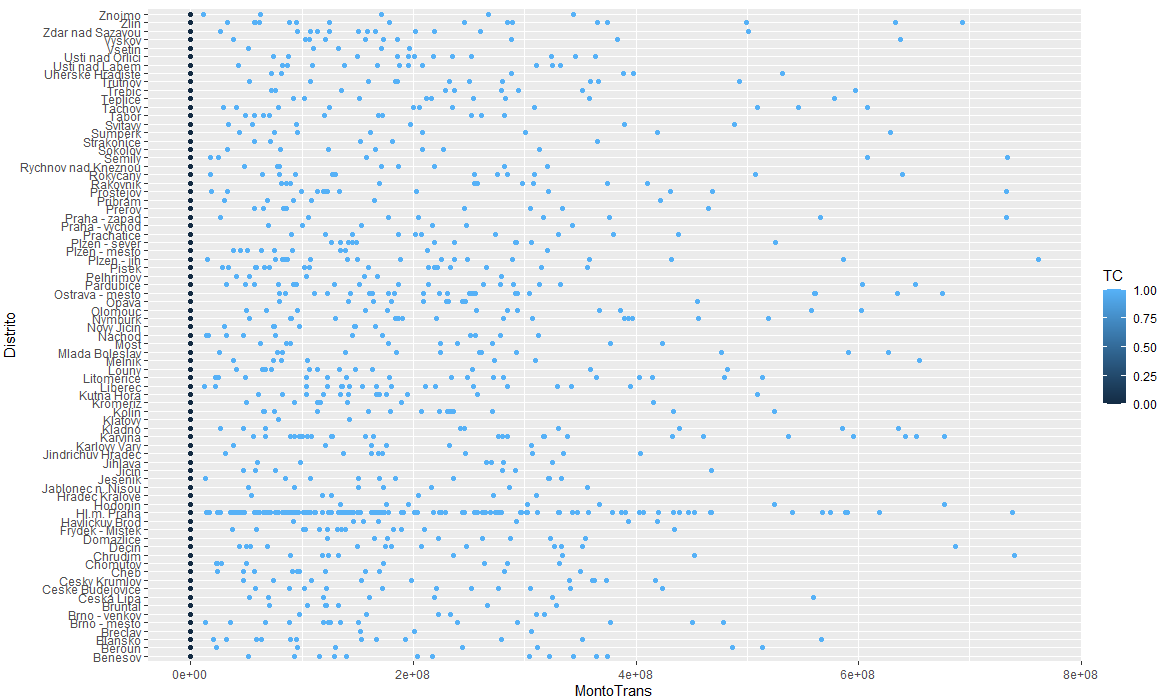
Se realiza un gráfico de boxplot para verificar la existencia de outliners sobre las variables numéricas. La única variable que presenta alguna dispersión en sus valores es Monto de transacción, la cual mirando en los registros vemos que existe una pequeña cantidad de clientes que tiene un monto elevado de transacciones, pero consideramos que es normal y posible que se de esta situación por lo que no se realiza ningún tratamiento sobre estos datos.



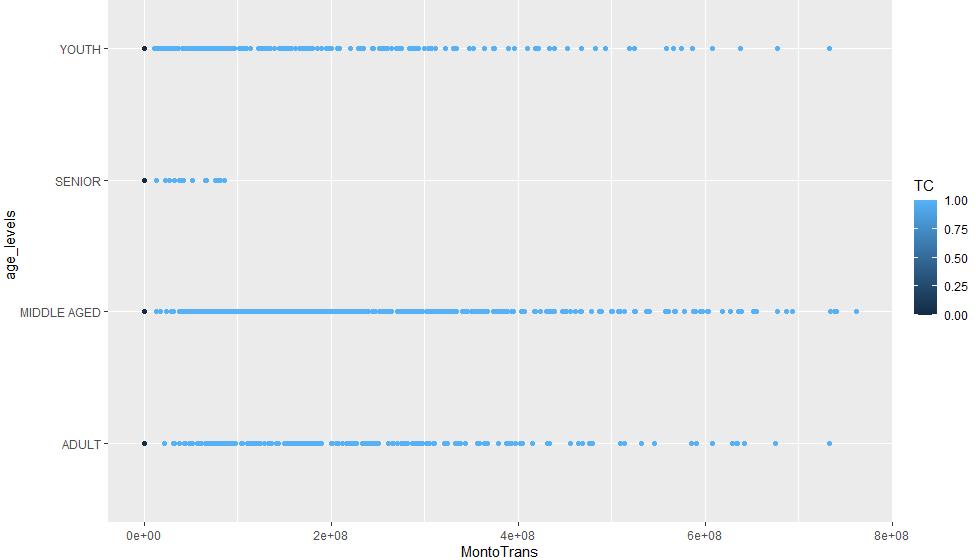
Vemos los clientes a través de un gráfico de barras los clientes que tienen y no tiene tarjeta de crédito:

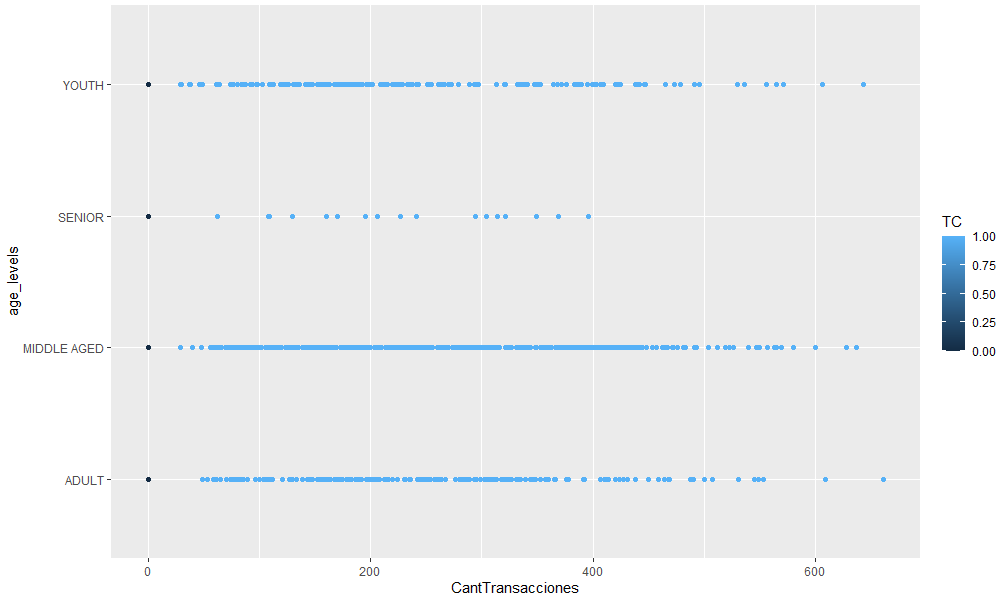


A través de los siguientes diagramas de dispersión trataremos de ver si existe alguna relación entre 2 variables de un cliente y si se otorga o no de una tarjeta de crédito.

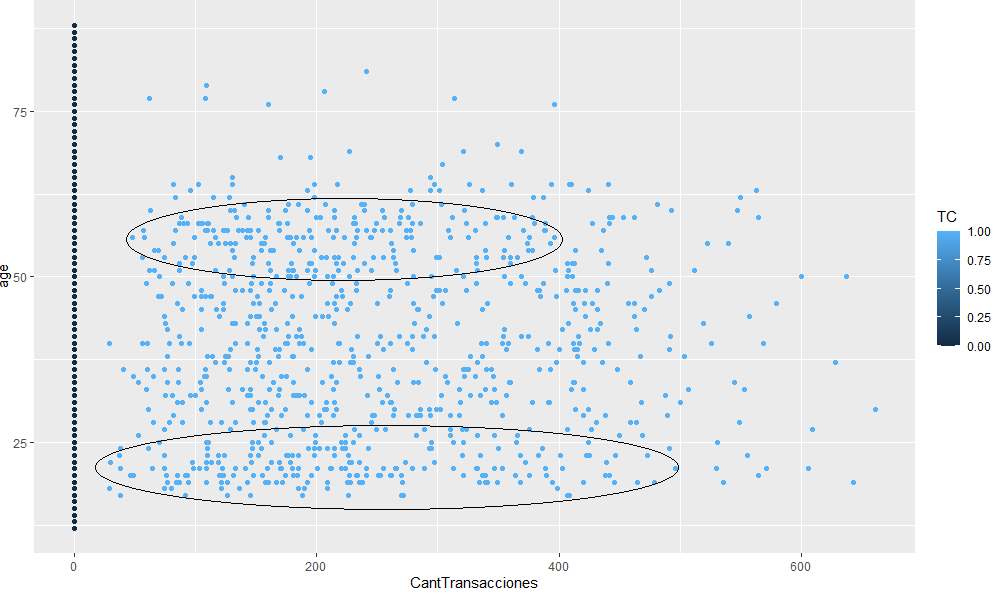


Vemos que todos los distritos tienen clientes que no realizaron aun transacciones. Además, vemos que la mayor cantidad de transacciones de clientes que tienen una tarjeta de crédito se realizan en Hl.m Prana.

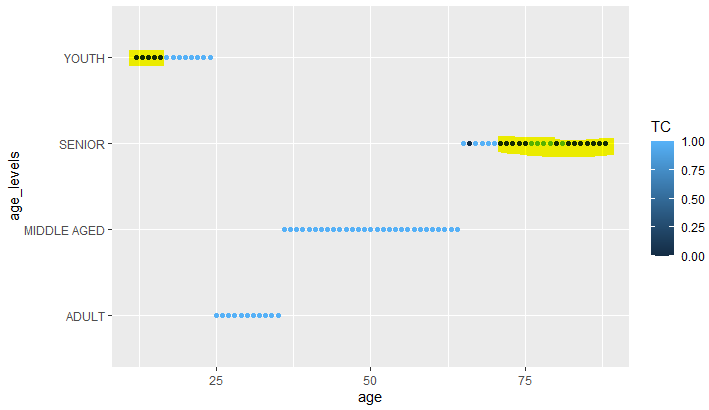
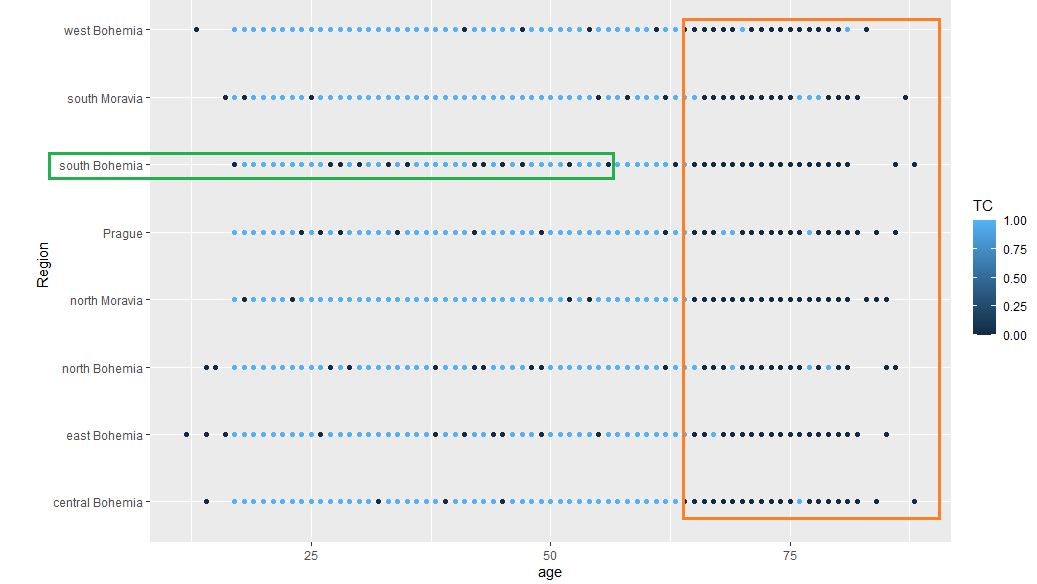




Vemos nuevamente que en todos los rangos etarios de los clientes que no aún no se ha realizado transacciones (monto transacción es cero). Además, observamos que el rango etario que mayor cantidad de transacciones y mayores montos registrados (línea casi continua y más larga) es el Middle Age.



Vemos que la cantidad de transaccione se tiene una mayor concentración en las franjas etarias que van aproximadamente desde los 20 a 25 años y de los 50 a los 60 años.



Se observa a partir de los últimos 2 gráficos que de los 65 años aproximadamente no se tiende a tener una tarjeta de crédito, en la mayoría de las regiones, por otro lado, tampoco se cuentan con tarjeta los clientes menores de 12 años o hasta los 18 años. Además, en la región se South Bohemnia se observa más edades con falta de tarjeta de crédito y que tienden a tener tarjetas de crédito en otras regiones.

**Primera tarea: Predecir un atributo discreto**

**Paso 4: Elegir el modelo**

Como deseamos predecir una variable a futuro (en este caso TC=otorgar o no una tarjeta de crédito a un cliente) que está presente en el pasado o sea forma parte de una de las variables del dataset, vamos a aplicar Aprendizaje supervisado para realizar una clasificación binaria sobre la variable a predecir TC, utilizando estos algoritmos:

* árbol de decisión
* redes neuronales

**Paso 5: Usar el modelo**

1. **Redes Neuronales**

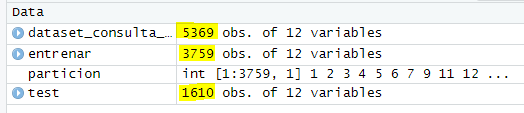
A partir del dataset cargado se realizan los siguientes pasos:

1. Realizamos un ordenamiento aleatorio de las filas del dataset (línea 15).



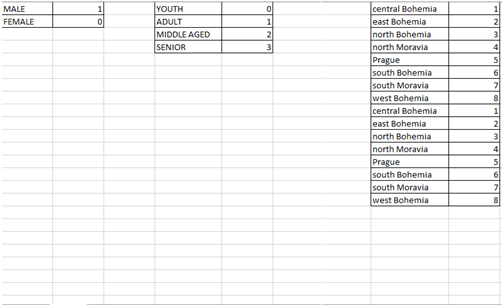
1. Se divide el dataset en dos, uno set de entrenamiento (con el 70% de los datos) y otro de prueba (con el 30% restante). En la línea 12 se realiza esta operación. Se establece una semilla fija para que la selección de los valores aleatorios al generar los dataset de entrenamiento y de prueba sean los mismos.

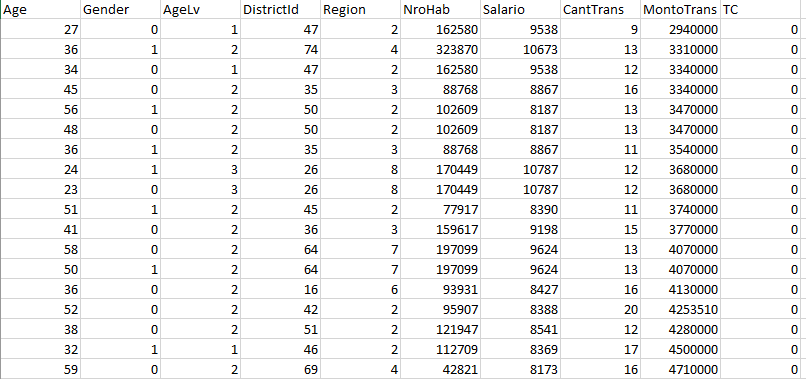




Se observa la división realizada del dataset original.

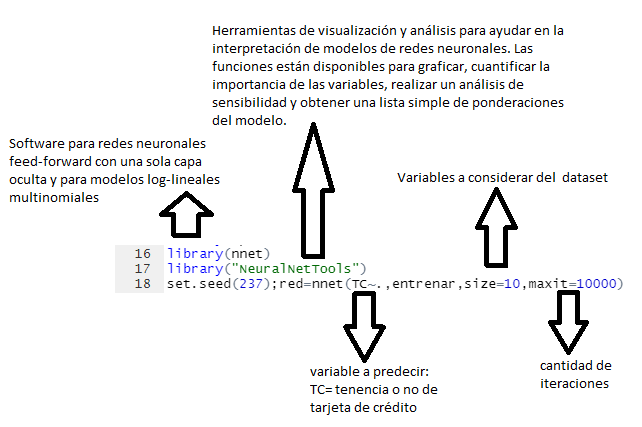
1. Antes de continuar se debe realizar una conversión de algunas variables char (región, sexo, distrito, genero) para que puedan servir de entrada al algoritmo de redes neuronales (que solo acepta números):





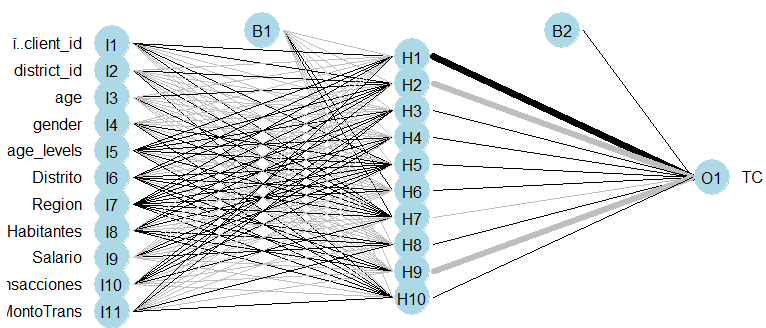
Dataset resultante luego de la conversión.

1. Se importan las librerías a utilizar y se aplica la función que crea la red neuronal



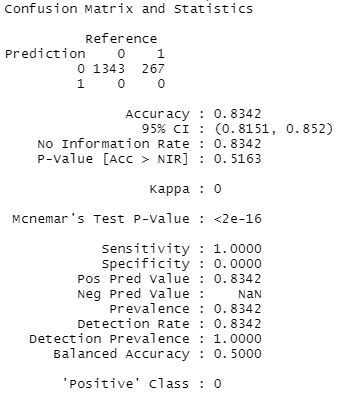
1. Se grafica la red y se obtiene la matriz de confusión





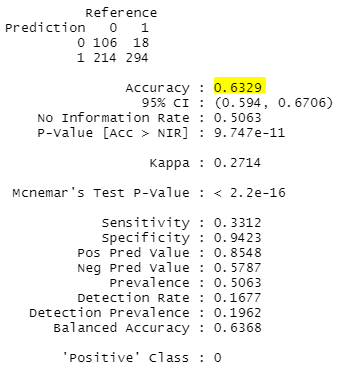
1. Testeamos el modelo:

En la línea 20 y 21 suministramos a la red neuronal el dataset de pruebas y solicitamos generamos una matriz de confusión para evaluar los resultados de las predicciones realizadas sobre el dataset de prueba sobre la variable aleatorio TC (otorgar tarjeta de crédito).



**Analicemos la matriz de confusión**: Esto nos indica que de los 1610 registros de clientes tomados a azar del dataset original, que conforman el dataset de testeo, la red neuronal entrenada fue capaz de predecir con una precisión del 83% aquellos clientes que no tenían tarjeta de crédito (verdaderos negativos) en 1343 registros de cliente del dataset de test. También se observa que se han registrado 267 falsos positivos o sea clientes a quienes se debía entregar la tarjeta, pero se denegó la misma.

Probando diferentes hiperparametros no sea ha logrado mejorar esta situación, según lo investigado puede que el data set este desbalanceado (ya que el dataset la mayoría de los clientes quienes no tienen tarjetas de crédito) produciendo que la red entrenada tienda a denegar siempre la entrega de Tarjetas de créditos a los clientes. Probando un dataset mas reducido y equilibrado se logra modificar este resultado y cambiar algunos hiperparametros. Se logra el siguiente resultado:

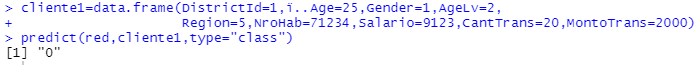


Como vemos se logra una precisión del 63%. Vemos que se tienen 106 verdades negativos, 294 verdades positivos. Además, se tienen 18 falsos negativos y 214 falsos positivos.

1. Predicción

Cargamos un cliente con datos al azar y vemos cual es el resultado de la predicción de la red neuronal.





En este caso no se otorga la tarjeta de crédito a este cliente.

A continuación se probará ahora utilizar un árbol de decisión para hacer la misma predicción y analizar los resultados.

1. **Árbol de Decisión**

A partir del dataset importado (data) se realizan los siguientes pasos:

1. Importamos las librerías a utilizar

|  |
| --- |
| library(lattice)  library(ggplot2)  library(caret)  library(rpart)  library(rpart.plot) |

1. Se factoriza el atributo TC

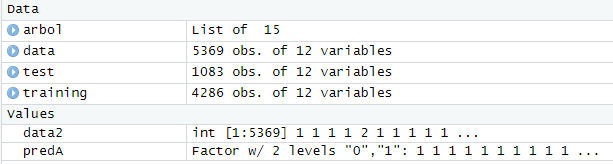
|  |
| --- |
| data$TC = factor(data$TC) |

1. Se reordena el dataset de manera aleatoria

|  |
| --- |
| data <- data[sample(1:nrow(data)),] |

1. Se particiona el dataset en dos grupos, el de entrenamiento (con el 80% de los datos) y el de prueba (con el 20% restante). En este caso usamos el dataset original, el árbol no requiere una conversión para que sus variables de entrada sean únicamente numéricas. El árbol al final decidirá que variables son las que se incluyen para tomar la decisión.

|  |
| --- |
| data2 <- sample(2,nrow(data),replace=TRUE,prob = c(0.8,0.2))  training <- data[data2==1,]  test <- data[data2==2,] |



1. Se crea el árbol. Para este ejemplo en particular, hemos decidido no tener en cuenta los atributos Nombre de Distrito y Región, ya que, a la hora de graficar, estos complicaban la lectura y grafica del árbol. En cambio, solo decidimos ocupar el ID de Distrito.

|  |
| --- |
| arbol<-rpart(TC~Age+Gender+AgeLv+DistrictId+NroHab+Salario+  CantTrans+MontoTrans, training,method="class",control=rpart.control  (minsplit=100,minbucket=1,cp=0)) |

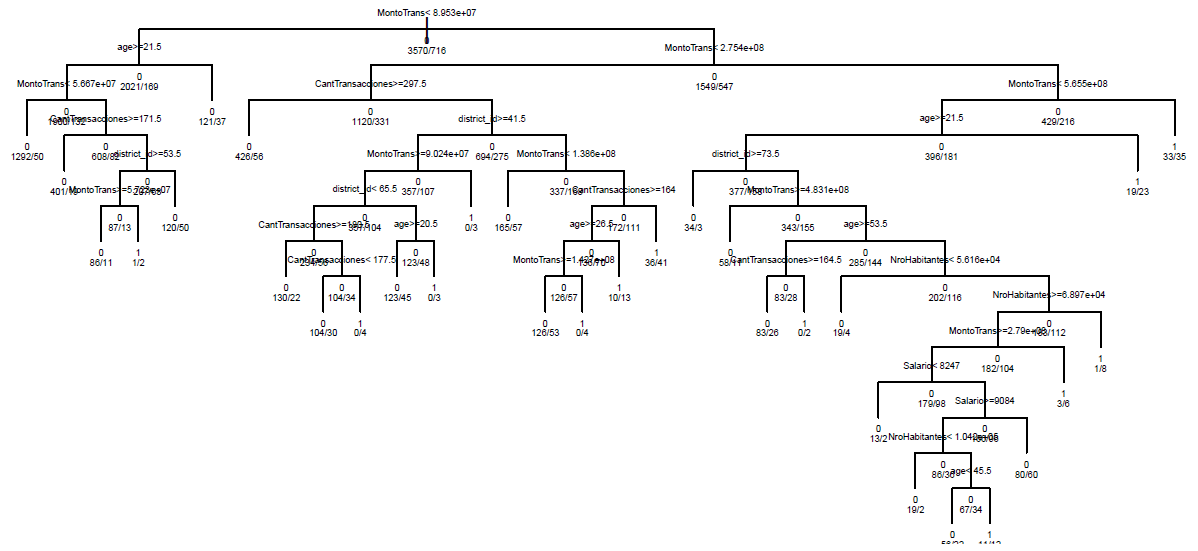
1. Se grafica el árbol

|  |
| --- |
| plot(arbol,uniform = TRUE,main="Arbol Decision")  text(arbol,use.n = TRUE,all = TRUE,cex=0.5) |

1. Se obtiene la matriz de confusión

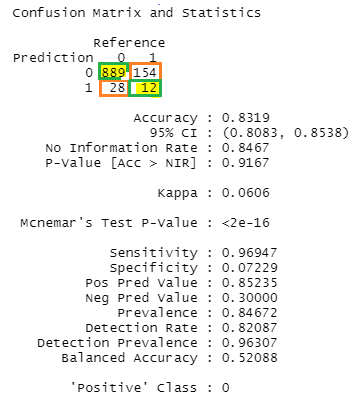
|  |
| --- |
| predA=predict(arbol,test,type="class")  confusionMatrix(predA,test$TC) |

1. Árbol de decisión resultante:



Vemos como el algoritmo decide a partir de las variables suministradas cuales son las más importante y en que rango de valores se decide se opta por tomar una rama u otra. Todo esto para decidir si la variable predictora TC termina en la última rama con lo decisión asociada.

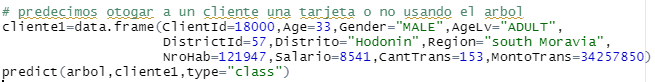
1. Testeamos el modelo:



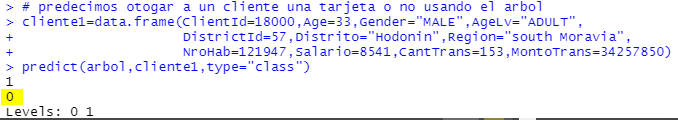
**Analicemos la matriz de confusión**: Esto nos indica que de los 1083 registros de clientes tomados a azar del dataset original que conforman el dataset de test, la red neuronal entrenada fue capaz de predecir con una precisión del 83% aquellos clientes a los que tenían no tenían tarjeta de crédito -> verdaderos negativos en 889 registros de cliente del dataset de test y cuáles de esto tenían una tarjeta de crédito -> verdaderos positivos en 12 registros de cliente del dataset de test. Además, existen 154 falsos negativos o sea la predicción es negativa (no otorgar tarjeta de crédito a ese cliente) cuando realmente el valor tendría que ser positivo (si otorgar tarjeta de crédito a ese cliente). Por último, se registran 28 falsos positivos o sea la predicción dio positiva (otorgar tarjeta de crédito a este cliente) pero realmente debiera ser negativo (no otorgar tarjeta a ese cliente). La tasa de no información es del 0.84 por lo que también es un bueno indicador.

1. Ejemplo de predicción:

Usamos el árbol para predecir si otorgar o no una tarjeta a un cliente:



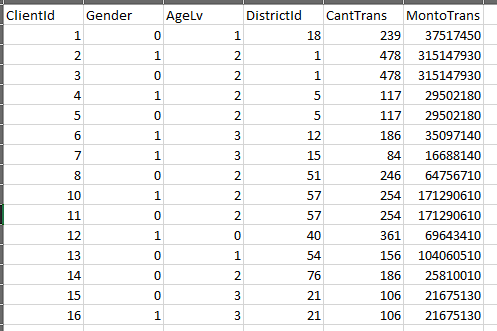
Vemos que la opción correcta seria no darle una tarjeta de crédito a este cliente.



**Segunda tarea: Buscar Grupo de elementos similares**

**Paso 4: Elegir el modelo**

Para completar esta tarea se hará uso de un algoritmo no supervisado, clustering. Se aplicará sobre el dataset original, pero con menos variables. La idea es ver que agrupamientos pueden formarse entre género, el rango etario y el distrito en donde viven los clientes, la cantidad de transacciones que realizaron y el monto total de transacciones. El objetivo es encontrar oportunidades de negocio para la gerencia a partir del análisis de los clusters de clientes que se formen. Ejemplo del dataset utilizado:



**Paso 5: Usar el modelo**

1. **K-Means**

A partir del dataset importado (data2) se realizan los siguientes pasos:

1. Importamos las librerías a utilizar

|  |
| --- |
| library(cluster)  library(factoextra) |

1. Convertimos el ClienteID en tipo carácter para ocuparlo posteriormente

|  |
| --- |
| data2[,1]<-as.character(data2[,1]) |

1. Quitamos el ClientId y almacenamos el resto en data1

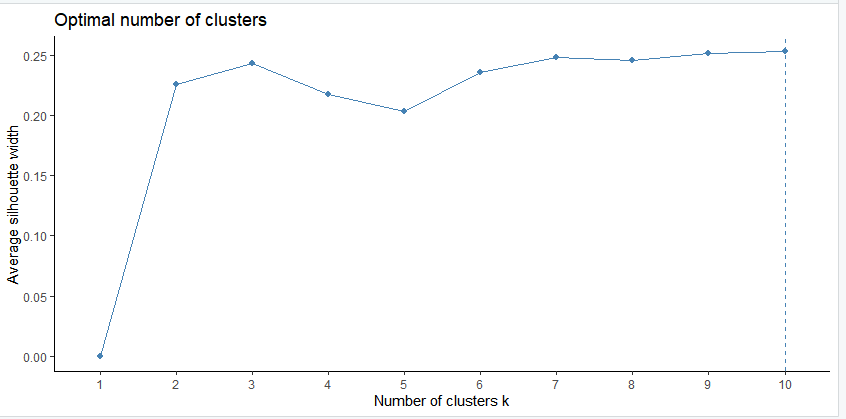
|  |
| --- |
| data1= data2[,2:6] |

1. Se estandariza el dataset para ser utilizado en este algoritmo

|  |
| --- |
| data1 <- scale(data1) |

1. Se estima el numero de clústers óptimos para este caso con el método silhouette

|  |
| --- |
| fviz\_nbclust(data1, FUNcluster=kmeans, method="silhouette",k.max = 10)+theme\_classic() |



Podemos observar que para este caso, la función recomienda el use de K=10, pero nosotros como ejemplo haremos uso de K=3.

1. Se ejecuta la función kmeans sobre el dataset y se lo almacena en la variable grupos3

|  |
| --- |
| set.seed(111)  grupos3 = kmeans(data1,3) |

1. Se agrupa el clúster con el data1

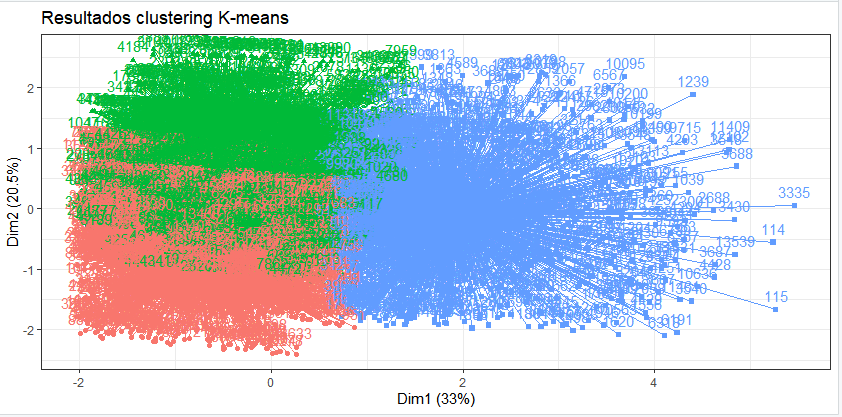
|  |
| --- |
| datakmeans=cbind(data1,grupos3$cluster) |

1. Se agrega la columa de ClientId del dataset original al data1, esto sirve para poder visualizar posteriormente que clientes caen en cada cluster.

|  |
| --- |
| rownames(data1) <- data2[,1] |

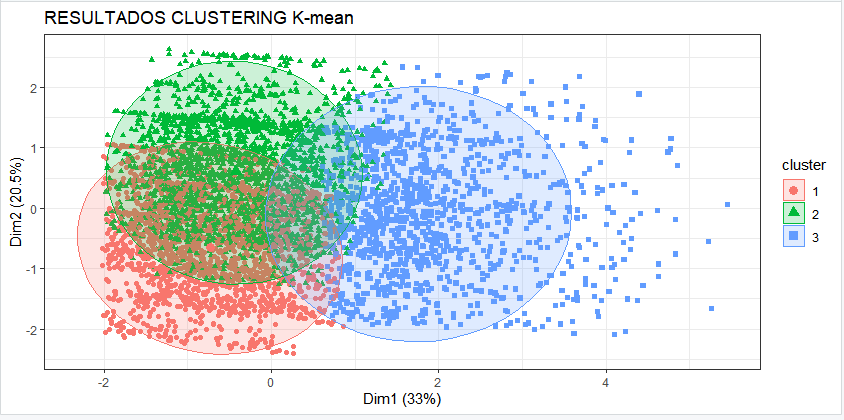
1. Se grafica con las etiquetas de ClientId

|  |
| --- |
| fviz\_cluster(object = grupos3, data = data1,show.clust.cent = TRUE,  ellipse.type = "euclid", star.plot = TRUE) + labs(title = "Resultados clustering K-means") + theme\_bw() + theme(legend.position = "none") |

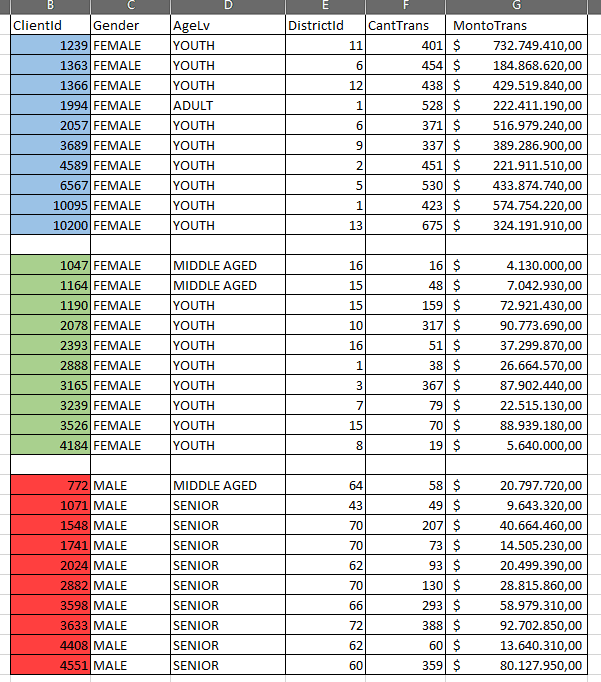


1. Se grafica sin las etiquetas de ClientId

|  |
| --- |
| fviz\_cluster(object = grupos3, data = data1,geom = "point", ellipse.type = "t",repel = TRUE) + theme\_bw() + labs(title = "RESULTADOS CLUSTERING K-mean") |



1. Análisis del resultado



Analizando los id\_cliente que aparecen en cada uno de los clusters se puede ver las siguientes relaciones:

* Podemos visualizar que en el clúster Azul se agrupan los clientes con las siguientes características: Gender=Female, AgeLevel=Youth, DisctrictId<=13, CantidadTransacciones>337 y MontoTransacciones > 221 Millones.
* En el cluster Verde se agrupan los clientes con las siguientes características: Gender=Female, AgeLevel=Youth, DisctrictId>10, CantidadTransacciones<367 y MontoTransacciones < 90 Millones.
* En el clúster Rojo predominan los clientes con las siguientes características: Gender=Male, AgeLevel=Senior y DisctrictId>60.

Como vemos el cluster celeste puede representar una oportunidad en la cual la gerencia puede enforcase, al registrarse históricamente montos y cantidades muy elevados de transacciones. Los clientes que registran estas características tienden a ser **mujeres** con rango etario **joven** que pertenecer id\_distritos menores o iguales a 13, o sea, viven en alguno de los siguientes distritos: **Hl.m. Praha, Benesov, Beroun, Kolin, Pribram, Kladno, Kutna Hora, Melnik, Mlada Boleslav, Nymburk, Praha – vychod, Pribram, Rakovnik**