# Índice

# 1. Resumen ejecutivo

En este proyecto se presentan datos de usuarios a quienes se les ha otorgado una Tarjeta de Crédito de un grupo financiero.

Entendemos por Tarjeta de Crédito el instrumento cuya presentación permite aplazar obligaciones de pago en determinadas transacciones. La exhibición de la tarjeta acreditará a su titular para disponer de bienes o servicios sin entrega inmediata de dinero en efectivo. Es la combinación de un crédito revolvente (crédito que se puede utilizar repetidamente) y un plástico que es utilizado para acceder a los fondos. El plástico puede ser utilizado en los comercios que aceptan este medio de pago, para comprar bienes o servicios sin hacer uso de dinero en efectivo. Las tarjetas de crédito son emitidas por Grupos Financieros, los cuales son agrupaciones integradas por una Sociedad Controladora y por entidades financieras tales como Casas de Cambio, Instituciones de Banca Múltiple, Sociedades Operadoras de Fondos de Inversión, Sociedades Financieras Populares, entre otras. La institución emisora del plástico, liquidará al comercio el importe de la compra a nombre de su cliente. Posteriormente la cantidad adeudada por el titular de la tarjeta deberá ser liquidada a la institución financiera, la cual puede ofrecer diferentes esquemas de pago diferido con y sin intereses por financiamiento.Cada mes la institución emisora de la tarjeta de crédito envía un Estado de Cuenta que resume las compras, disposiciones de efectivo, comisiones y pagos realizados al crédito desde la fecha de corte inicial hasta la fecha de corte final; por lo cual la institución necesita llevar un registro de estos movimientos junto con los datos personales de cada cliente.

Con alrededor de diez mil registros, la tabla obtenida de un grupo financiero nos proporciona información sobre los datos personales de sus clientes tales como su edad, ocupación, estado civil o sexo. Esta información no resulta relevante por sí sola, por suerte en la tabla también podemos encontrar medios para identificar las condiciones con el mercado, situación tecnológica y relación bancaria. Para los grupos financieros es muy importa la activación de TdC, ya que el costo global es de $ 690.00, es el costo desde la venta hasta la distribución (entrega). Por ejemplo, si no se activan 10 tarjetas, el costo de adquisición que ya no se recuperará es $ 6,900.00. Por otro lado, el ingreso neto que una tarjeta le deja al banco a los 12 meses es $ 2,400; este ingreso es financiero por varias cosas (anualidad, comisiones, intereses, intercambio, sobretasa, etcétera). Cada cuenta que activa en promedio alcanza un saldo a los 12 meses de $ 14,000.00, es decir, si se tienen 10 cuentas que activan, a los 12 meses estás 10 cuentas van a contribuir a la cartera un monto de $ 140,000.00. Si el cliente activa la tarjeta y luego la cancela, no hay mucha pérdida; una vez que la tarjeta se activa genera estado de cuenta y se le carga anualidad al cliente, con lo que se logra al menos que el gasto por adquisición sea $ 0.00. La cuenta cae a la unidad de retención y el costo asumido es la oferta de valor y la comisión que se le paga al ejecutivo, en promedio el costo de $ 120.00. La frecuencia de que un cliente active y luego cancele la tarjeta en el primer mes es del 0.5 % y al tercer mes solo el 0.8 %

Por lo anterior surge el interés de crear modelos que ayuden a incrementar la activación de TdC del portafolio.

Con la ayuda de herramientas computacionales es posible hacer un análisis sobre las diversas variables incluidas en la tabla y de esta forma agrupar a los clientes de acuerdo a la similitud que guardan respecto a su comportamiento financiero y características. También segmentaremos el portafolio en dos grupos, clasificandolos por lo que sean propensos a la activación y los que no. Resulta imperativo decidir qué variables son aquellas más útiles para la tarea de agrupar y decidir quien posiblemente activará.

Una vez obtenidos los grupos en los que clasificamos a los clientes podemos visualizar las características personales que se asocian más con cada grupo y de esta manera dirigir de una manera óptima futuros campañas de venta como se mostrará en el desarrollo de este proyecto, así como el desarrallo de estrategias para el incremento de la activación de TdC.

# 2. Objetivos

## 2.1. Objetivo general

Identificar clientes propensos a activar la TdC, con el fin de orientar esfuerzos, reducir costos e incrementar la activación

## 2.2. Objetivos específicos

1. Agrupar a los clientes de acuerdo a sus diferentes características.
2. Determinar qué clientes son potenciales a activar su tarjeta de crédito
3. Generar una Estrategia de Negocio para aumentar el índice de activación de TdC.

# 3. Estado del arte

## 3.1. Conceptos Preliminares

### 3.1.1. Análisis univariante

Es un análisis básico, primario, en el cuál las características o propiedades han de medirse una a una de modo univariado. Los tipos que se usan en el análisis univariado serán vistos a continuación.

*Distribución de frecuencias*

La distribución de frecuencias nos indica el número de casos que hay en cada categoría de la variable. A partir de dichos valores, en una tabla de frecuencias, se calcula el porcentaje (respecto del total de observaciones), porcentaje válido (excluido los valores perdidos) y el porcentaje acumulado (porcentaje de la primera categoría, luego éste más el de la segunda categoría y así sucesivamente). Se aplica para variables nominales, ordinales y en cierto tipo de variables intervalares (por ejemplo, en escalas Likert). Además de la tabla de frecuencias también es posible hacer representaciones gráficas.

*Medidas de tendencia central*

Las medidas de tendencia central son cálculos o evaluaciones que nos proporcionan idea del comportamiento del fenómeno en la parte céntrica de éste. En otras palabras las medidas de tendencia central se ocupan de medir el centro, el foco o el valor medio de un fenómeno.

Algunas medidas son las siguientes:

- Media

  La media o promedio corresponde a la suma de todas las puntuaciones de la variable dividida por el número total de casos.

- Mediana

  La mediana es el valor que divide por la mitad a las puntuaciones de la variable: los que están por debajo de éste y los que están por encima. Es decir, es el valor que divide en dos mitades a las observaciones.

- Moda

  La moda es el valor que más se repite del conjunto de observaciones, pudiendo haber más de una moda (bimodal o multimodal).

*Medidas de dispersión*

Las medidas de dispersión indican el grado variabilidad de los datos respecto de la media (promedio). Se debe tener presente que una propiedad de la media es que la suma de las diferencias de todos los valores de la variable respecto de la media es siempre "0". Es por ello que para el cálculo de la varianza y la desviación estándar se procede a elevar la suma de las diferencias al cuadrado.

La varianza es el valor promedio del cuadrado de las puntuaciones respecto de la media. Se utiliza mucho en pruebas de inferencia estadística (de la muestra al universo), pero su unidad de medida no es directamente interpretable (ya que está al cuadrado), razón por la cual se recurre a la desviación estándar.

La desviación estándar o típica es el promedio de desviación de los valores de las observaciones respecto de la media, expresada en los valores originales de la medición de la variable. Esto no es otra cosa que la raíz cuadrada de la varianza. Cuanto más se aleje el valor respecto de la media, mayor será la desviación estándar. Se aplica a variables medidas a nivel intervalar o de razón.

*Medidas de Posición Relativa*

Medidas de Posición Relativa o llamados también cuantiles, son aquellos valores de las variables que dividen una distribución de frecuencias o serie de números en 4, 10 ó 100 partes iguales, tomando la denominación de quartiles, deciles ó percentiles, respectivamente.

### 3.1.2. Análisis bivariante

El análisis bivariado de datos es una forma evolucionada de análisis estadístico en el cual se cuantifica a nivel descriptivo e inferencial el nivel de covarianza entre dos variables y de esta forma se da cuenta de la relación entre dos variables. La cuantificación de la covarianza consiste en la construcción de coeficientes que permitan integrar en un valor estimado, información con respecto a la varianza conjunta entre dos variables y tiene como objetivo fundamental definir la magnitud y el sentido de la relación entre las variables. De este modo, el análisis conjunto de las varianzas de dos variables (regularmente definidas como X y Y) permite identificar

la relación empírica entre éstas, entendiendo por relación el ajuste de los datos a una función lineal estocástica subyacente. A partir de un referente teórico pertinente, el análisis bivariado busca someter a contrastación la tesis de asociación y hasta causalidad entre dos variables definidas. En cualquier caso, el análisis bivariado se plantea con la intención de determinar el nivel de relación entre dos variables y la función estocástica que subyace a un conjunto de observaciones $\{(x, y)\}$. Pues si bien, la relación no es evidencia suficiente de causalidad no se puede hablar de causalidad en ausencia de relación entre las variables. El análisis bivariante de datos involucra una familia de estadísticos cuya pertinencia está condicionada por el nivel de medición (Stevens, 1946) de las variables involucradas. Esta familia de estadísticos se divide en dos grandes grupos, a saber: paramétricos y no paramétricos. (Siegel and Castellan, 1995). Los paramétricos agrupan el caso de las variables con nivel de medición de intervalo o superior, distribución normal bivariada y n>30. Los no paramétricos son el resto de las pruebas de correlación que no cumplen con los supuestos de las pruebas paramétricas; lo cual, les permite agrupar los estadísticos de contingencia y de correlación para variables con nivel de medición inferior a intervalos. En cualquier caso, el interés fundamental es construir un índice que permita determinar la magnitud y dirección de la relación entre las variables.

La prueba basada en el Coeficiente V de Cramer tiene como finalidad comparar grados de asociación entre variables medidas a nivel nominal. El Coeficiente V de Cramer asume valores entre 0 y 1, en donde, valores próximos a 0 indican una muy baja asociación entre las variables y valores próximos a 1 indicas una fuerte asociación.

Finalmente, la prueba de significación estadística basada en el Coeficiente de Correlación de Spearman tiene por objeto determinar la dirección y la intensidad de la asociación entre dos variables medidas a nivel ordinal. Dicho coeficiente toma valores entre -1 y +1. Los valores cercanos a -1 ó +1 indican fuerte asociación entre las variables mientras que los valores cercanos a 0 indican una muy baja asociación. Si el valor es positivo, la variables varían en la misma dirección, en tanto, si es negativo lo hacen en direcciones opuestas (a medida que aumenta una disminuye la otra). Se debe tener presente que Spearman está pensado para detectar relaciones de tipo lineal, pero no todas las relaciones son lineales (por ejemplo, las curvilíneas).
*Valores extremos*

Un valor más extremo (outlier) es un valor en un conjunto de datos que es muy diferente de los otros valores. Esto es, los outliers son valores excepcionalmente lejanos del centro.

En la mayoría de los casos, los outliers tienen influencia en la media , pero no en la mediana , o la moda . Por lo tanto, los outliers son importantes en su efecto en la media.

## 3.2. Modelación No Supervisada

La Modelación No Supervisada consiste en hacer clasificaciones (agrupaciones) en un conjunto de datos buscando patrones que ayuden a diferenciar a alguna parte de los datos de otra. No hay una variable objetivo.

### 3.2.1. Análisis de Conglomerados (Clustering)

El análisis de conglomerados (Clustering) es una técnica estadística multivariante para agrupar conjuntos de objetos buscando la máxima homogeneidad en los grupos, al igual que la mayor diferenciación entre cada uno de estos.

Se basa en criterios geométricos y se usa como técnica exploratoria y descriptiva, pero no explicativa. Sus principales aplicaciones se dan en los Estudios de Mercado, Psiquiatría, Clasificación del Clima, Arqueología, Bioinformática y Genética, entre muchos otros más.

Cabe recalcar que el Clustering no funciona como técnica inferencial, ya que dependiendo del modelo utilizado, se pueden producir clasificaciones diferentes. Además, la adición o sustracción de variables impactan de forma directa en los resultados del algoritmo.

Sustancialmente, para hacer clusters debemos preguntarnos qué tan cerca está un elemento de otro, y es aquí donde entra un concepto que atienden a ésta duda: similitud.

### 3.2.2. Similitud

El punto de partida es considerar una matriz de $n \times p$, dicha matriz tiene entradas las cuales se analizan a modo de conocer la proximidad entre cada uno de sus vectores, por ejemplo. Decimos que dos elementos son similares cuando su distancia es pequeña.

Tenemos diferentes tipos de medidas de proximidad, cuyos atributos dependerán de los tipos de datos. Para datos categóricos, comúnmente se suelen escalar los resultados al intervalo [0,1], tal como se hace con la posibilidad de un evento en la Teoría de la Probabilidad. Para datos binarios, que son los más comunes, o de más de dos niveles, suele contarse el número de casos que satisfacen cada uno de los niveles para posteriormente obtener proporciones, por ejemplo.

Para medir la similitud entre dos datos binarios, consideremos dos observaciones $i$, $j$ y supongamos que tenemos $p$ variables de estas; diremos que $a$ es el número de veces en que ambos datos reportan el valor de 1, $b$ es el número de veces que se reportan 0 y 1, $c$ el número de veces que se dan 1 y 0 y $d$ cuando ambos valores son 0. Evidentemente, $p = a+b+c+d$ y así se propone un coeficiente de similitud $S_{ij}$. A continuación, se presenta un modelo general para calcular dicho coeficiente.

$$S_{ij} = (a+d)/(a+r(b+c)+d)$$

Dependiendo del tipo de investigación se decidirá si considerar o no a los casos $d$ y el peso $r$ ($r = \frac{1}{2}$ ó $r = 2$) que se les dará a los casos $b$ y $c$.

Para datos continuos se mide la no similitud entre dos observaciones, es decir, el error que tienen una respecto a la otra. Éste tipo de medida debe cumplir la desigualdad del triángulo, es decir, para cuaquier terna de datos continuos $i$, $j$ y $k$,la medida de diferencia $d_{ij}$ entre estos debe cumplir:

$$d_{ij} + d_{ik} \geq d_{jk}$$

Consideremos una matriz $D$ de $n \times n$ cuyos elementos son $d_{ij}$ (medida de diferencia entre la observación $i$ y $j$). Además, $w_k$, $k = 1, ..., p$ denotará los diferentes pesos no negativos de las $p$ variables cuantitativas; y $x_{ik}$ denotará el valor de la $k$-ésima entrada de la observación $i$.

Se proponen los siguientes coeficientes de no similitud.

- Distancia Euclidiana

$$d_{ij} = \left[\sum_{k=1}^{p} w_k^2 (x_{ik} - x_{jk})^2\right]^{\frac{1}{2}}$$

- Distancia Manhattan

$$d_{ij} = \sum_{k=1}^{p} w_k |x_{ik} - x_{jk}|$$

- Distancia de Minkowski

$$d_{ij} = \left[\sum_{k=1}^{p} w_k^r (x_{ik} - x_{jk})^r\right]^{\frac{1}{r}} ; \; r \geq 1$$

- Distancia de Camberra

$$d_{ij} = \begin{cases} 0 & \text{para} \quad x_{ik} = x_{jk} = 0 \\ \displaystyle\sum_{k=1}^{p} w_k |x_{ik} - x_{jk}|/(|x_{ik}| + |x_{jk}|) & \text{para} \quad x_{ik} \neq 0 \quad \text{ó} \quad x_{jk} \neq 0 \end{cases}$$

La elección del peso $w_k$ implica que la importancia de una variable decrece cuando su variabilidad incrementa. De las anteriores distancias, la Euclidiana suele ser la más utilizada, aunque junto con la Manhattan son casos particulares de la distancia de Minkowski. La de Camberra es muy sensible cuando los valores de las $x_{ik}$ se aproximan a 0. Existen otros coeficientes de distancia de medidas que se basan en la correlación entre dos variables, pero no serán utilizadas en nuestro estudio.

Ahora, para saber cómo escoger un coeficiente de similitud, Gower y Legendre (1986) dicen que "un coeficiente tiene que ser considerado en el contexto del estudio estadístico descriptivo del cual es parte, incluyendo la naturaleza de los datos y el tipo de análisis destinado"

Hay quienes sugieren utilizar las métricas cuyo cálculo resulte más simple, ya que así se podría facilitar la interpretación de los resultados. En general, podemos decir que no hay una métrica absoluta, todo dependerá de la intuición del investigador y del tipo de datos y variables de los que se dispongan.

### 3.2.3. Representación Gráfica de los Datos

*Principales Componentes (PCA)*

En nuestro estudio, para hacer manifiesto el poder del sistema visual humano de detectar patrones lo primero a lo que se recurre es a histogramas o gráficos de dispersión. Posteriormente se puede recorrer a hacer un análisis de componentes principales, el cual es un método para transformar las variables en un conjunto de

datos multivariados en una nueva cantidad menor de variables no correlacionadas entre ellas y que explican la mayor parte de la varianza del conjunto inicial de datos.
En Clustering, provee una proyección de los datos a una dimensión menor, lo cual conlleva a una inspección visual más informativa.
El proceso es el siguiente:

- La primer componente $y_1$ se define como una combinación lineal de todas las $p$ variables originales $x_1$, $x_2$, ..., $x_p$, de tal modo que albergue la mayor cantidad posible de la varianza de los datos.
- Se repite lo anterior para una segunda componente pero ahora para tratar de albergar la varianza remanente y así sucesivamente. Se espera que las variables $y_j$ vayan siendo no correlacionadas.
- Se tiene así el siguiente sistema de ecuaciones lineales

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + ... + a_{1p}x_p \\ y_2 &= a_{21}x_1 + a_{22}x_2 + ... + a_{2p}x_p \\ &\vdots \\ y_q &= a_{q1}x_1 + a_{q2}x_2 + ... + a_{qp}x_p \end{aligned}$$

- Los coeficientes en las combinaciones lineales serán los eigenvectores de la matriz de correlación $R$ (cuando los datos están en escalas muy diferentes) o con la matriz de covarianza $S$.
- Las varianzas de las componentes estará dada por los eigenvalores de $R$ o $S$, donde las primeras componentes que expliquen una mayor proporción de la varianza de las variables observadas se asumen como un resumen de éstas últimas. Escalamiento multidimensional.

PCA intenta representar las similitudes o diferencias mediante un modelo geométrico de un espacio de dimensión p en uno de dimensión $q$ ($q < p$), tal que una medida de distancia (comúnmente la Euclidiana) entre dos puntos en el espacio represente lo mejor posible la proximidad observada. Para dos observaciones $i$, $j$, se busca que la distancia $d_{ij}$ (en el espacio de menor dimensión) coincida, en algún sentido, con la medida de similitud (diferencia) $s_{ij}(d_{ij})$. Para $x_i$, $x_j$ dos vectores de dimensión $q$, $f$ se asume como una relación funcional entre las similitudes (diferencias) y las distancias correspondientes. Donde $h$ denota la medida de distancia.

*Escalamiento Multidimensional*

Intenta representar las similitudes o diferencias mediante un modelo geométrico de un espacio de dimensión $p$ en uno de dimensión $q$ ($q < p$), tal que una medida de distancia (comúnmente la Euclidiana) entre dos puntos en el espacio represente lo mejor posible la proximidad observada. Para dos observaciones $i$, $j$, se busca que la distancia $d_{ij}$ (en el espacio de menor dimensión) coincida, en algún sentido, con la medida de similitud (diferencia) $s_{ij}$ $(d_{ij})$.

$$\begin{aligned} d_{ij} &= f(\delta_{ij}) \\ \delta_{ij} &= h(x_i, x_j) \end{aligned}$$

Para $x_i$, $x_j$ dos vectores de dimensión $q$, $f$ se asume como una relación funcional entre las similitudes (diferencias) y las distancias correspondientes. Donde $h$ denota la medida de distancia.

### 3.2.4. Clustering de Optimización y Clustering Jerárquico

Hay diferentes tipos de Clustering, pero diferenciamos entre dos principalmente, el Jerárquico y el de Optimización.

*Clustering Jerárquico*

En el Clustering Jerárquico los datos se dividen a su vez de dos formas: en la primera se realizan una serie de particiones que pueden ir de un número $n$ de clusters a uno en particular; en el segundo caso, las particiones se realizan a un clúster para generar n grupos más. Es decir, yendo de lo particular a lo general (Aglomerativo) o de lo general a lo particular (Divisivo).

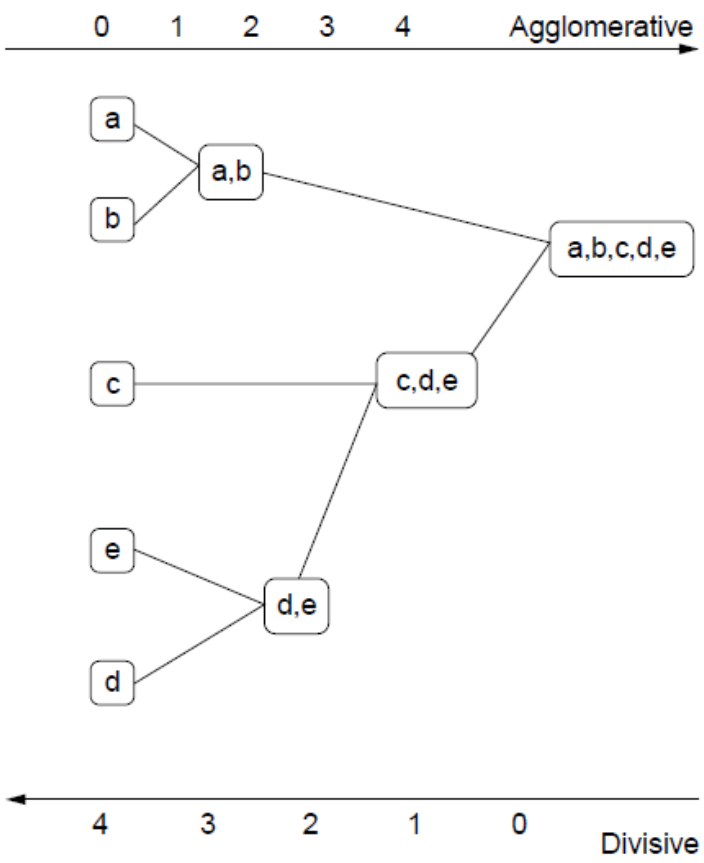



Figura 1: A la izquierda se lleva a cabo el Clustering Aglomerativo, mientras que a la derecha se realiza el Divisivo.

El investigador decide hasta qué momento parar las particiones.

La mayoría de las investigaciones se han concentrado en las técnicas aglomerativas, por lo que expondremos los principales métodos de clustering Jerárquico aplicados a éstas últimas. Las propiedades de lo aglomerativo se pueden aplicar a lo divisivo.

- Average
  Éste método se utiliza para medir similitud o distancia entre grupos de datos. La distancia entre clusters se define como la distancia promedio entre pares de observaciones $i$, $j$; $i$ en un cluster y $j$ en el otro. Tiende a unir clusters con varianzas pequeñas, toma en cuenta la estructura de cada grupo y es relativamente robusto.

- Centroid
  Requiere que los datos estén acomodados en una matriz de renglones e implica fusionar clústers cuyos vectores promedio sean más similares. La distancia entre clusters se define como la distancia euclidiana promedio a las medias de los vectores (centroides), asume que cada observación puede representarse en el espacio euclidiano y así tener una interpretación geométrica; del par de clústers fusionados, predominan las características de aquel con mayor cantidad de elementos.

- Método Ward
  También requiere una matriz de datos, pero aquí la distancia entre clusters se define con el incremento en la suma de errores cuadrados - respecto al vector promedio- dentro de los clusters, se trata de minimizar dicho incremento.
  El total de dichos errores $E$, está dado por:

$$E = \sum_{m=1}^{g} E_m,$$

donde

$$E_m = \sum_{l=1}^{n_m} \sum_{k=1}^{p_k} (x_{ml,k} - \bar{x}_{m,k})^2,$$

en el cual $\bar{x}_{m,k} = \frac{1}{n_m} \sum_{l=1}^{n_m} x_{ml,k}$ es la media del clúster $m$ para la variable $k$, y $x_{ml,k}$ es el resultado de la variable $k$ $(k = 1, ..., p)$ para el $l$-ésimo objeto $(l = 1, n_m)$ del clúster $m$ $(m = 1, ..., g)$.

Después de la fusión, sumando sobre todas las variables, tiende a crear clusters del mismo tamaño y esféricos. Es sensible a outliers.

*Clustering de Optimización*

En el Clustering de Optimización, se realizan particiones de los elementos iniciales en un número determinado de grupos optimizando algún criterio numérico. El número de clusters es fijado previamente (subjetivamente, auxiliándose de gráficos donde se contrasta el criterio numérico y el número de clúster).
Con la partición de los $n$ individuos en $g$ grupos se genera un índice $c(n, g)$ y se hace una medición de la calidad de dicho índice, éste se desarrollará con los conceptos de homogeneidad y separación. Los criterios para el análisis de cluster serán: minimizar la falta de homogeneidad o maximizar la separación de los grupos.
Considérese una matriz $D$, donde su elemento $d_{ql,kv}$ mide la diferencia entre el elemento $l$ del grupo $q$ y el elemento $k$ del grupo $k$, y sea $n_m$ el número de elementos del clúster $m$. Para $r = 1, 2$ , se muestran algunas medidas para la falta de homogeneidad $(h_k(m))$ y otras para la separación $(i_k(m))$ entre clústers.

- $h_1(m) = \sum_{l=1}^{n_m} \sum_{v=1, v\neq l}^{n_m} (d_{ml,mv})^r$
- $h_2(m) = \max_{\substack{l,v=1,...,n_m \\ v\neq l}} [(d_{ml,mv})^r]$
- $h_3(m) = \min_{v=1,...,n_m} [(d_{ml,mv})^r]$
- $i_1(m) = \sum_{l=1}^{n_m} \sum_{k\neq m} \sum_{v=1}^{n_k} (d_{ml,kv})^r$
- $i_2(m) = \min_{\substack{l=1,...,n_m \\ k\neq m \\ v=1,...,n_k}} [(d_{ml,kv})^r]$

Cuando $r = 1$, $h_2(m)$ puede verse como el diámetro del cluster, mientras que de $h_3(m)$ se dice que es el índice estrella del clúster.
Con lo anterior podemos crear alguno de los siguientes índices para medir la homogeneidad o separación en general de la técnica usada.

- $c_1(n, g) = E = \sum_{m=1}^{g} \frac{h_1(m)}{n_m}$
- $c_2(n, g) = \max_{m=1,...,g} h(m)$
- $c_3(n, g) = \min_{m=1,...,g} h(m)$

Por otra parte, para medir la variabilidad de los clusters, el criterio más utilizado es el siguiente. Considere una matriz $D$ de $n \times p$ con datos continuos, haremos uso de la siguiente matriz $T$ $p \times p$ de descomposición:

$$T = \sum_{m=1}^{g} \sum_{l=1, v\neq l}^{n_m} (x_{ml} - \overline{x})(x_{ml} - \overline{x})'$$

Donde $x_{ml}$ es un vector de dimensión $p$ que denota a la $l$-ésima observación en el grupo $m$ y $\overline{x}$ es el vector de promedios de cada una de las$p$variables.
$T$ se puede ver como la suma de la dispersión dentro del clúster $m$ $(W)$ y la dispersión entre los grupos $(B)$:

$$T = W + B$$

Donde $W = \sum_{m=1}^{g} \sum_{l=1}^{n_m} (x_{ml} - \overline{x}_m)(x_{ml} - \overline{x}_m)'$ y $B = \sum_{m=1}^{g} n_m (x_m - \overline{x}_m)(x_m - \overline{x}_m)'$

En el caso multivariado, es decir, cuando $p > 1$, para minimizar la suma de las sumas de cuadrados intraclusters sobre todas las variables, basta con minimizar la $Tr(W)$ (Lo cual es equivalente a maximizar $Tr(B)$). Lo anterior es equivalente a minimizar la distancia euclidiana cuadrada entre individuos y su media grupal, es decir,

$$E = \sum_{m=1}^{g} \sum_{l=1}^{n_m} (x_{ml} - \overline{x}_m)(x_{ml} - \overline{x}_m)' = \sum_{m=1}^{g} \sum_{l=1}^{n_m} d_{ml,m}^2$$

Donde $d_{ml,m}$ es la distancia euclidiana entre el individuo $l$ del grupo $m$ y su media grupal $x_m$.

### 3.2.5. Clustering Difuso

Hasta ahora, los métodos de Clustering que hemos visto resultan informales y subjetivos (a la hora de determinar el número de clusters, por ejemplo). El siguiente es un modelo estadístico formal aplicable a una "población"de datos.
Se asume a los datos como una población la cual se dividirá en subpoblaciones (Clusters) dentro de las cuales cada una de las variables tendrá una función de densidad de probabilidad diferente, lo que resultará en una Densidad Mixta Finita para la población completa.

*Modelos Gaussianos Mixtos*

Los Modelos Gaussianos Mixtos (MGM) son un modelo ejemplo de esta técnica. Estos suponen que todos los datos son generados por una cantidad finita de distribuciones gaussianas - por tanto, se tiene una distribución normal multivariante - con parámetros desconocidos. La estimación de dichos parámetros suele llevarse a cabo mediante máxima verosimilitud.
En nuestro estudio, en particular, los MGM implementan el algoritmo Expectation-Maximization (EM) para ajustar los datos al modelo. El principal problema para que un MGM aprenda con datos sin etiqueta es que no se suele conocer de qué componente vienen cada una de las observaciones, por lo que no se les puede asignar una distribución. EM es un algoritmo estadísticamente válido para atacar este problema y lo hace mediante iteraciones.
Primero, se asumen componentes aleatorias (utilizando K-Means, por ejemplo) y se calcula la probabilidad de que alguna observación haya sido generado por cada componente del modelo. Así se van variando los parámetros para maximizar las asignaciones sobre cada componente hasta que se converja a un óptimo local donde la probabilidad se maximice.
Éste algoritmo es muy rápido y al maximizar la verosimilitud no se sesgan ni la media hacia cero, ni los tamaños de los clusters para que tengan alguna estructura en particular, pero para estimar la matriz de covarianzas se requiere de suficientes datos, ya que el problema se complicaría de tal forma que el resultado podría divergir.

## 3.3. Modelación Supervisada

Mientras que en los modelos no supervisados realizábamos agrupaciones sin tener algún ejemplo previo, en la modelación supervisada buscamos que el modelo entrene con base en patrones identificados con anterioridad. Asumiendo que el futuro se comportará como el pasado reciente nuestro modelo realizará predicciones. La Modelación Supervisada permite buscar patrones de datos relacionando todos los campos y un campo en especial al cual llamaremos “Campo Objetivo” o “Variable Objetivo”.

Es importante señalar las diferencias entre las tareas de Clasificación y Regresión que podemos realizar con la modelación supervisada. La Clasificación tiene por objetivo la asignación de una clase, es decir predecir a que clase pertenece un conjunto de datos, mientras que la Regresión tiene el objetivo de predecir valores continuos, es decir valores numéricos.

### 3.3.1. Medidas de precisión de los modelos

**Misclassification rate**: Es conocido también como Error Rate, el cual es una métrica de error de predicción para un problema de clasificación binaria. Las métricas de tasa de error para un problema de clasificación de dos clases se calculan con la ayuda de una matriz de confusión. Una matriz de confusión es una tabla que se usa a menudo para describir el rendimiento de un modelo de clasificación en un conjunto de datos de prueba para los cuales se conocen los valores verdaderos.

**ROC Index:** son curvas en las que se presenta la sensibilidad en función de los falsos positivos (complementario de la especificidad) para distintos puntos de corte. Si la prueba fuera perfecta, es decir, sin solapamiento, hay una región en la que cualquier punto de corte tiene sensibilidad y especifidad iguales a 1: la curva sólo tiene el punto (0,1). Si la prueba fuera inútil: ambas coinciden y la sensibilidad (verdaderos positivos) es igual a la proporción de falsos positivos, la curva sería la diagonal de (0,0) a (1,1). Las pruebas habituales tienen curvas intermedias.

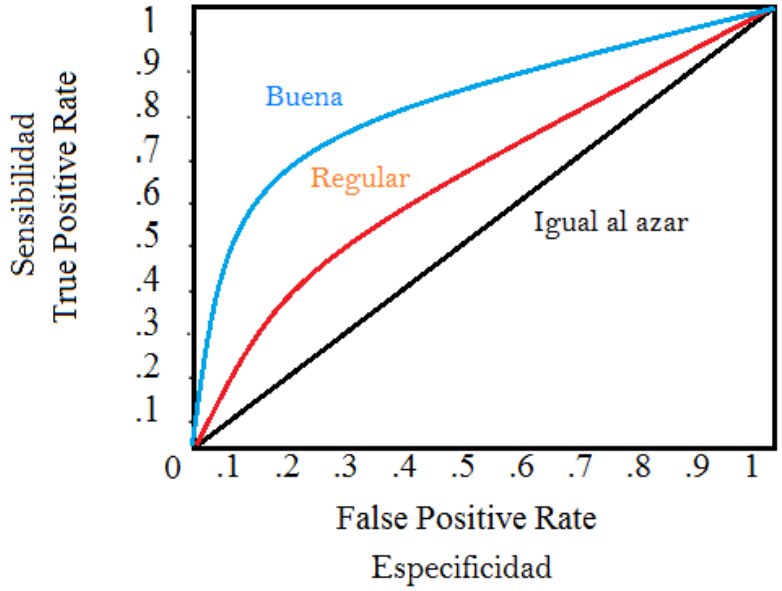



Figura 2: Un parámetro para evaluar la bondad de la prueba es el área bajo la curva que tomará valores entre 1 (prueba perfecta) y 0,5 (prueba inútil).

**Kolmogorov-Smirnov**: La prueba de Kolmogorov-Smirnov para una muestra es un procedimiento de "bondad de ajuste", que permite medir el grado de concordancia existente entre la distribución de un conjunto de datos y una distribución teórica específica. Su objetivo es señalar si los datos provienen de una población que tiene la distribución teórica especificada, es decir, contrasta si las observaciones podrían razonablemente proceder de la distribución especificada.

### 3.3.2. Regresión Logística

La regresión logística es un procedimiento cuantitativo de gran utilidad para problemas donde la variable dependiente toma valores en un conjunto finito. Su uso se impone de manera creciente desde la década de los 80 debido a las facilidades computacionales con que se cuenta desde entonces. A continuación, desarrollaremos el caso especial en que la variable dependiente o respuesta es dicotómica. Podemos decir que la variable dependiente Y toma valor 1 si ocurre el suceso, y valor 0 si no ocurre el suceso. Por otra parte nos interesa estudiar la relación entre una o más variables independientes o explicativas: $X1, X2, ..., Xp$ y la variable Y. El modelo logístico establece la siguiente relación entre la probabilidad de que ocurra el suceso, dado que el individuo presenta los valores $X1 = x1, X2 = x2, ..., Xp = xp$:

$$Pr(Y = 1 : x_1, x_2, ..., x_p) = \frac{1}{1 + exp(-\alpha - \beta_1 x_1 - ... - \beta_p x_p)} \tag{1}$$

Un problema importante es estimar los parámetros $\alpha, \beta_i's$ , a partir de un conjunto de observaciones. El procedimiento de estimación de estos parámetros se basa en el método de máxima verosimilitud Una vez que hayamos calculado los estimadores máximo-verosímiles (MV) de $\beta_i's$, puede interesarnos el cálculo de intervalos de confianza de estos parámetros, para ello podemos utilizar la estimación de la matriz de covarianza de los estimadores MV de los $\beta_i$.

### 3.3.3. Weight of Evidence

El valor del peso de la evidencia o WOE es una medida ampliamente utilizada de la "fortaleza"de una agrupación para separar el riesgo bueno y el malo (por defecto). Se calcula a partir del odds-ratio básico:

(Distribución de buenos resultados ) / (Distribución de los malos resultados)

O las proporciones de las entregas de los bienes de distribución / distritos para abreviar, donde Distr se refiere a la proporción de buenos o malos en el grupo respectivo, relativo a los totales de las columnas, es decir, expresado como proporciones relativas del número total de buenos y malos.

Específicamente, el valor del Peso de la Evidencia para un grupo que consiste en n observaciones se calcula como:

$$WoE = [ln(\frac{DistBuenos}{DistMalos})] * 100 \tag{2}$$

El valor de información (IV) de un predictor está relacionado con la suma de los valores (absolutos) de WoE en todos los grupos. Por lo tanto, expresa la cantidad de información de diagnóstico de una variable de predicción para separar los Bienes de los Bads. Específicamente, dado un predictor con n grupos, cada uno con una cierta Distribución de Bienes y Miedos, el Valor de Información (IV) para ese predictor se puede calcular como:

$$IV = \sum_{i-1}^{n} \left[ (DistBuenos_i - DistMalos_i) * ln \left( \frac{Distbuenos}{Distmalos} \right) \right] \tag{3}$$

Según Siddiqi (2006), por convención, los valores de la estadística IV pueden interpretarse de la siguiente manera. Si la estadística IV es:
- Menos de 0.02, entonces el predictor no es útil para modelar
- 0.02 a 0.1, entonces el predictor tiene solo una relación débil
- 0.1 a 0.3, entonces el predictor tiene una relación de fuerza media c
- 0.3 o superior, entonces el predictor tiene una fuerte relación

### 3.3.4. Árboles de Decisión

Los Árboles de Decisión de Clasificación y Regresión (CART, por sus siglas en inglés) es una técnica exploratoria de datos que tiene como objetivo fundamental encontrar reglas de clasificación y predicción. Dado un conjunto de datos $D = (X, Y)$, donde Y es la variable a explicar y $X = (X_1, ..., X_p)$ es un vector de p variables que describe a los individuos, el objetivo de CART es predecir los valores de Y a partir de los valores observados de las variables $X_i$ , i = 1,. . ., p. Tanto la variable dependiente $Y$, como cada una de las variables explicativas $X_i$ puede ser cuantitativa o cualitativa, esto dota a CART de una gran flexibilidad pues se puede aplicar en muchos contextos distintos. En el caso en que la variable dependiente Y sea cualitativa, se dice que CART es un árbol de clasificación, y el objetivo es predecir la clasificación que le correspondería a un individuo con cierto perfil de valores en las variables explicativas. Por otra parte, si Y es cuantitativa, CART es llamado árbol de regresión y el objetivo es idéntico al de un modelo lineal, obtener una estimación del valor de Y asociado a cada nicho o perfil de predictores. Además, esta técnica es utilizada para la selección de variables en el sentido que permite determinar cuál característica -o conjunto de características- es la que mejor define o discrimina a los grupos predeterminados.

Los CART, pueden verse como la estructura resultante de la partición recursiva del espacio de las variables explicativas (espacio de representación) a partir de un conjunto de reglas de decisión. La manera en que se construye cada partición es lo que distingue a los distintos tipos de árboles, éstas son determinadas por un conjunto de decisiones sobre las variables explicativas. En CART las reglas de decisión son desplegadas en forma de árbol binario. Determinan en cada momento dos alternativas posibles, las mismas se suceden hasta que el árbol llega a su construcción final. El procedimiento es recursivo y se traduce en una organización jerárquica del espacio de representación.

El objetivo es formar grupos homogéneos respecto a la variable que se desea discriminar y a su vez mantener el árbol razonablemente pequeño.

Para dividir los datos se requiere un criterio de particionamiento el cual se basa en una medida de impureza. Esta última establecerá el grado de heterogeneidad de la variable dependiente Y en cada nodo.

El análisis de árboles de clasificación y regresión generalmente consiste en tres pasos (Timofeev, 2004):

1. Construcción del árbol maximal (Construir todas las particiones hasta el final)

2. Poda del árbol (Eliminar las particiones que menos aportan a explicar la respuesta)

3. Selección del árbol óptimo mediante un procedimiento de validación

### 3.3.5. Análisis Discriminante

Predice una variable respuesta categórica con un plano discriminante generado ajustando densidades condicionales a los datos y usando la Regla de Bayes. Se ajusta una densidad gaussiana a cada clase, asumiendo que todas las clases comparten la misma matriz de covarianzas. El modelo ajustado puede utilizarse, además, para reducir la dimensión de los datos de entrada. Existe una extensión cuadrática de éste modelo, pero aquí sí son requeridas las matrices de covarianza de cada clase.

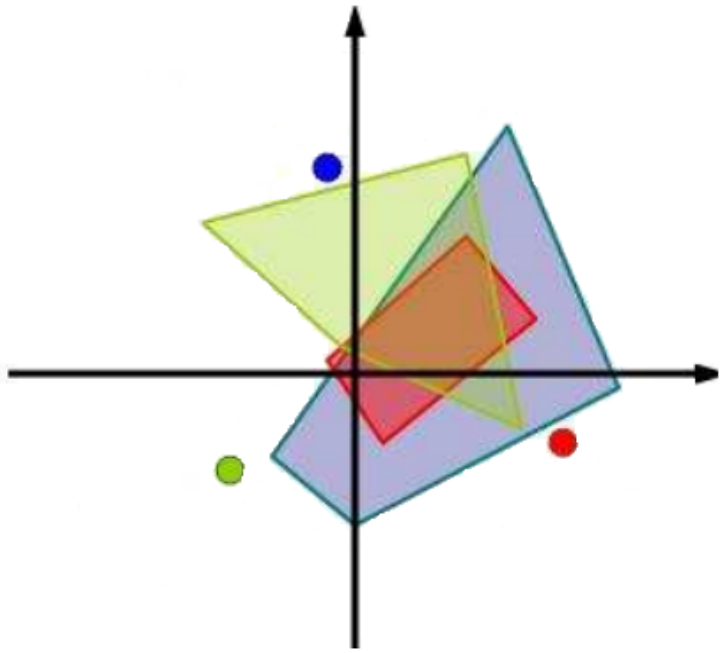


Figura 3: Diferente plano de separación representados en $R^2$.

Más específicamente, $P(X|y = k)$ se modela como una distribución normal multivariada para cada clase k. Esto conduce a superficies de decisión lineales entre clases.

Dado un conjunto de $p$ variables observadas en $k$ grupos, se buscarán $m$ funciones y de tal forma que Para algún $i \in 1, ..., m$

$$y_i = w_{i1}x_1 + ... + w_{ip}x_p \text{ de tal forma que } y_i \perp y_j; \forall i \neq j$$

Se suelen transformar los datos con componentes principales, con lo que las diferentes clases se conocen como discriminates canónicas. Así, la construcción de las $y_i$ se realiza un proceso similar al ya descrito en la subsección 3.1.3 de esta investigación, es decir:

- Se crea $y_1$ de tal forma que discrimine lo mayormente posible entre los grupos.
- Lo mismo para $y_2$ con la varianza remanente de los datos tal que la correlación entre $y_1$ y $y_2$ sea cero.

### 3.3.6. K-Vecinos más cercanos

Este algorimo consiste en clasificar a los vecinos más cercanos en un espacio de características multidimensionales a algún punto en particular, esto se hará con la ayuda de una cantidad fija K que estará dada por algún criterio de distancia (suele ser la Euclidiana o Manhattan).

Como tal, tiene bastantes desventajas ante conjuntos de datos grandes, no suele generalizar bien, se necesita mucho cuidado escogiendo las características de los datos, considera en el modelo muchos datos irrelevantes, etc.

La mejor elección de $K$ depende de qué tan grande sea nuestra población de estudio. Grandes valores de $K$ tienen a crear gran cantidad de clases en términos de los valores incluidos en éstas. Una elección adecuada puede ser estimada por una regla de decisión o usando un método de remuestreo (como Cross-Validation) para asignar el valor medio entre las muestras. La medidad de Accuracy no es buena referencia para calificar a este modelo.

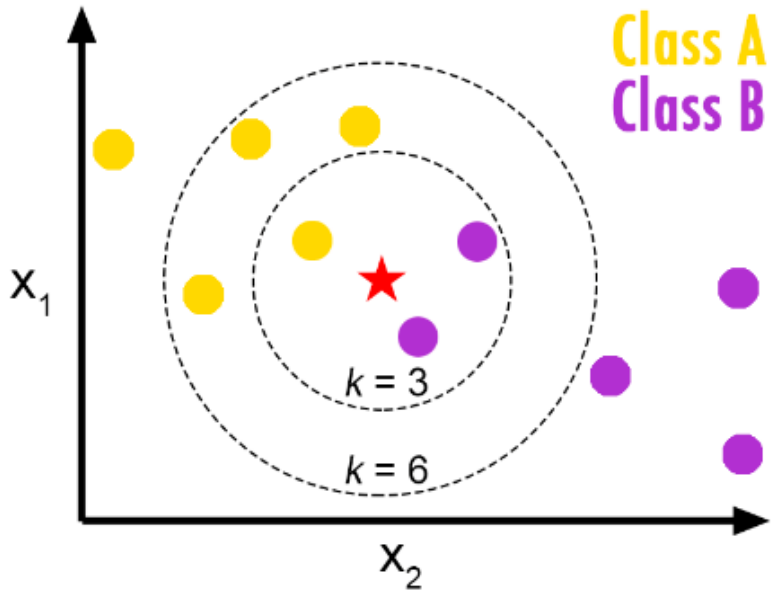



Figura 4: La estrella denota el atributo con el que queremos clasificar. Nótese que si K=3, la clasificación es con la clase B, pero si K=6, será la clase A la que predomine.

### 3.3.7. Clasificador Ingenuo de Bayes

EN este algoritmo se aplica el Teorema de Bayes (TB) con el supuesto ïngenuo"de que cada par de variables de entrada son independientes entre sí. Dado una variable y (objetivo) y un conjunto de n vectores característicos dependientes, el TB estipula:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1,...,x_n|y)}{P(x_1,...,x_n)}$$

En consecuencia a la independencia, se tiene que

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, x_n) = P(x_i|y), \forall i$$

Así, usando TB,

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,...,x_n)}$$

Y por tanto,

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

Con lo que se propone el siguiente estimador para y:

$$\hat{y} = arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y)$$

$P(y)$ y $P(x_i|y)$ se obtienen con estimación A Posteriori del Máximo (MAP) donde el precedente es la frecuencia relativa de la clase $y$ en el conjunto de entrenamiento.

Los clasificadores de Bayes requieren de un conjunto pequeño de variables de entrenamiento, por lo que resultan ser más rápidos que algoritmos más sofisticados. La distribución de $P(x_i|y)$ suele asumirse Normal o Bernoulli.

### 3.3.8. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (ANN, por sus siglas en inglés) buscan imitar la estructura y funciones del cerebro humano. La estructura es la siguiente:

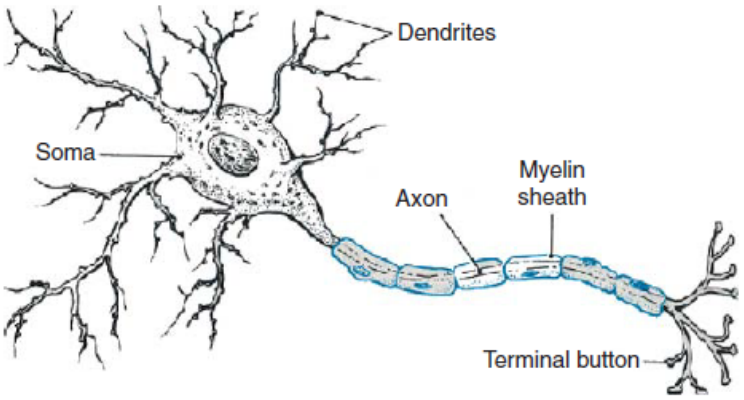



Figura 5: Estructura de la nuerona del ser humano.

Cada neurona recibe impulsos eléctricos de células vecinas y los acumula hasta que cierto nivel es excedido. Con ello, de dispara un impulso a la célula inmediata. La capacidad de cada celda para alacenar dichos impulsos y el límite son controlados por procesos bioquímicos que cambian sobre el tiempo. Éste cambio es generado gracias al sistema nervioso autónomo y es por ello que aprendemos a pensar o activar nuestro cuerpo.
El proceso de activación en las ANN está representado por una función, usualmente lineal (Regresión) o logística (Clasificación). El umbral del límite se muestra en el siguiente gráfico.

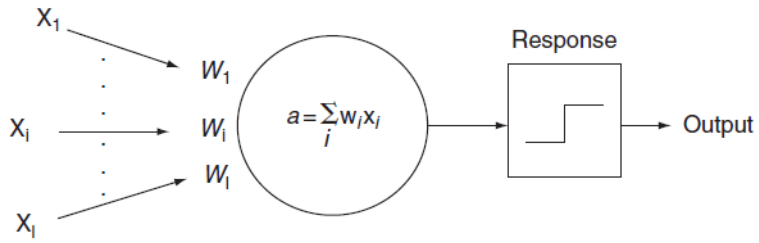



Figura 6: Cada entrada recibe un peso, lo cual se manifestará en los resultados.

Las $x_i$ son las variables de entrada, las $w_i$ son los pesos asociados a cada lazo con otra variable y son equivalentes a las interconecciones. Esto representa como tal la conexión entre dos células llamada sinapsis.
La propiedad más interesante de una red neural se manifiesta cuando se intercalan capas intermedias de neuronas (nodos) entre los nodos de entrada y de salida Entre más capas intermedias, el modelo es mejor, aunque el costo computacional incrementa considerablemente.

El aprendizaje de las neuronas humanas, se refleja realizando uno de varios procesos de ajuste de peso, el más común es llamado backpropagation. Se basa en la magnitud de los errores para reasignar los pesos, residiseña el modelo de forma iterativa y mejora su poder predictivo.
El proceso de backpropagation es el siguiente.

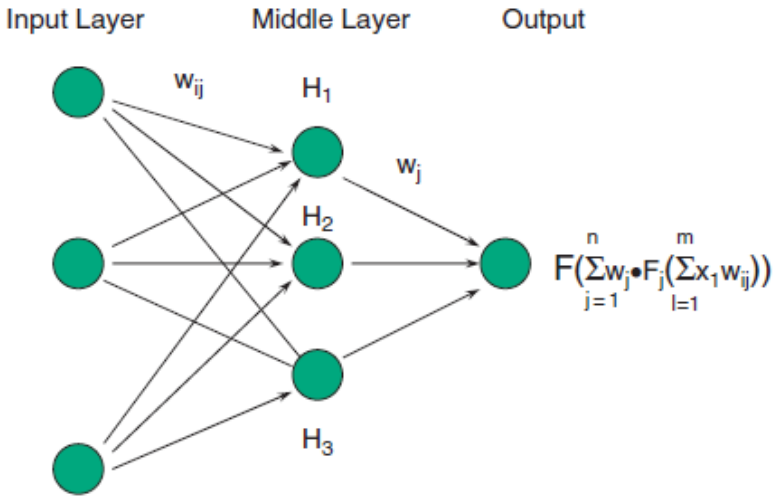



Figura 7: Arquitectura básica de una red neural.

- Se asignan pesos al azar a cada conección
- Tras leer la primer observación se calculan los valores para cada nodo como la suma de las entradas por sus pesos.
- Se especifica tanto un límite inferior como superior para que la salida sea calificada como 1 (interconección) o 0 (no interconección).
- Calcular el Error de Predicción: Predición esperada - Predicción Actual
- Ajuste de Pesos = Error de Predicción * Peso del resultado
- Calcular nuevo peso: Anterior peso de entrada + Ajuste de Pesos
- Se hace lo mismo para todas las demás entradas del modelo
- Relizar el proceso anterior hasta que se converja a un resultado o cuando los cambios no sean significativos entre una iteración y otra.

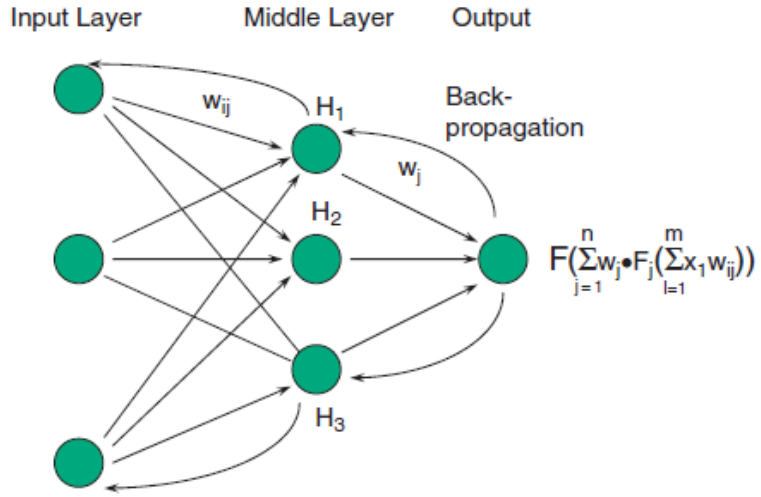



Figura 8: Hay un retorno a nodos visitados con anterioridad, lo que mejora el aprendizaje del algoritmo.

Una de las grandes desventajas de las ANN es que uno no puede saber la forma en específico en que está discriminando en cada nodo. Por eso tienen la reputación de ser çajas negras", aunque son un algoritmo muy

poderoso.

El tipo de red neural más utilizada, y la usada en nuestro modelo, es la Perceptron Multicapa (MLP).

*Perceptron Multicapa*

Este algoritmo supervisado aprende con una función $f(\cdot) : R^m \rightarrow R^s$, m es la dimensión de las entradas y s la de las salidas del modelo. Dado un conjunto de variables $\{X_{i=1,\ldots,m}\}$ y una variable target $y$, éste algoritmo clasifica o hace regresión de forma no lineal.
MLP entrena usando Gradientes Estocásticos Descendientse (SGD), Adam (un método de estimació estocástica) y con el algormitmo de memoria limitada L-BFGS (Limited-Memory of Broyden–Fletcher–Goldfarb–Shanno). SGD se trató en subsecciones anteriores. Adam trabaja de manera similar pero con un enfoque estocástica, además puede ajustar automáticamente el monto para actualizar los parámetros adaptando estimaciones de momentos de bajo orden. Por último, L-BFGS aproxima la matriz Hessiana que representa las derivadas de segundo orden de una función y después aproxima la inversa de dicha matriz para actualizar los parámetros.

Dado un conjunto de entrenamiento $\{(x_i, y_i)\}$, donde $x_i \in R^n$ y $y \in 0, 1$, para cada capa oculta y para cada neurona MLP entrena con la función $f(x) = W_2 g(W_1^T x + b_1) + b2$, donde $W_1 \in R^m$ y $W_1, b_1, b_2 \in R$ son parámetros del modelo. $W_1, W_2$ representan los pesos para las capas de entrada y las capas ocultas, respectivamente y $b_1, b_2$ su respectivo sesgo. $g(\cdot) : R \rightarrow R$ es una función de activación (Tangente hiperbólica, Función logística, entre otras)

MLP usa funciones de pérdida: Cross-Entropy (Clasificación) y Error Cuadrado (Regresión). El algoritmo para cuando se alcanza cierto número de iteraciones dadas o cuando la mejora está po debajo de cierto número pequeño.

### 3.3.9. Máquina Vector Soporte

Las Máquinas de Soporte Vectorial (SVM) realizan una búsqueda de hiperplanos con la función kernel para así segmentar los datos de entrada del modelo. A diferencia de otros algoritmos donde se minimiza el error empírico, éste ataca a otro conocido como riesgo estructural. La idea general es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para así obtener el margen máximo a cada lado del hiperplano. Sólo se consideran aquellas observaciones que caen justo en la frontera de dichos márgenes y se les conoce como vectores soportetete.

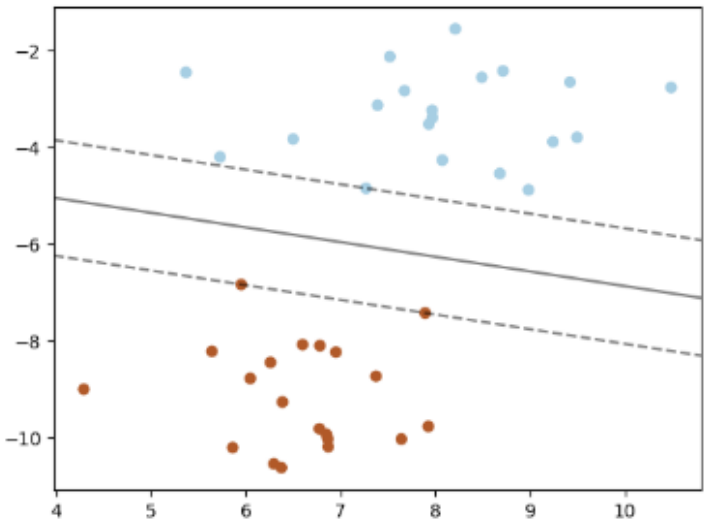



Figura 9: Aquellos vectores que aparecen sobre las líneas punteadas se conocen como *Vectores Soporte* .

Lo anterior representa un problema de optimización cuadrático con restricciones lineales que puede resolverse mediante programación cuadrática. Se exige convexidad para la resolución por lo que ésta última será única.

Dados $n$ vectores de dimensión $p$, en dos clases, y un vector $y \in 1, -1^n$, SVM resuelve el siguiente problema primal:

$$\begin{gathered}\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \\ \text{s.a.} \\ y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \\ \zeta_i \geq 0, i = 1, ..., n\end{gathered}$$

El planteamiento y solución del problema son los siguientes:

- Obtención de función langrangiana a optimizar
- Relacionar variables del problema primal con el dual
- Restriccciones adicionales a las variables duales
- Obtener problema dual a minimizar

Su problema dual estará dado por:

$$\begin{gathered}\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ \text{s.a.} \\ y^T \alpha = 0, \\ 0 \leq \alpha_i \leq C, i = 1, ..., n\end{gathered}$$

Donde $e$ es el vector de unos, $C > 0$ es la cota superior y $Q$ es una matriz $n \times n$ definida semipositiva, $Q_{ij} \equiv y_i y_j K(x_i, x_j), dondeK(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ es el kernel. Aquí, los vectores de entrenamiento se mapean a un espacio de dimensión mayor por la función $\phi$.
Para resolver el problema dual, se requiere de una función $K$ conocida como Kernel y se puede escoger entre las siguientes:

- Kernel Lineal: $K(x, x') =< x, x' >$
- Kernel Polinómico de grado p: $K(x, x')_p = [\gamma < x, x' > +r]^p$
- Kernel Gaussiano: $K(x, x') = exp(-\gamma \parallel x - x' \parallel^2), \gamma > 0$

### 3.3.10. Gradiente Estocástico Descendiente

Al igual que la Regresión Logística y las Máquinas de Vector Soporte, este algoritmo de aprendizaje discriminativo, clasifica linealmente sobre funciones de pérdida convexa.
Dado un conjunto de entrenamiento $(x_i, yi)_{i=1}^{i=n}$ donde $x_i \in R^m$ y $y_i$ es una variable dicotómica igual a 1 o -1. Se busca una función de scoring lineal $f(x) = w^T x + b$ donde $w \in R^m$ y $b \in R$.
Para encontrar los parámetros del modelo se suele minimizar el error de entrenamiento dado por:

$$E(w, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(w)$$

Donde $L$ es una función de pérdida que mide el ajuste del modelo y $R$ es un término de regularización (aka penalty) que penaliza la complejidad del modelo; $\alpha$ es un híper parámetro no negativo.
L puede ser:

- Hinge: Máquinas de Vectores Soporte
- Log: Regresión Logística
- Mínimos cuadrados: Regresión de cresta
- Insensibilidad de épsilon: Regresión de Vectores Soporte

Mientras que R:

- Norma $L_1$: $R(w) := \sum_{i=1}^{n} |w_i|$
- Norma $L_2$: $R(w) := \frac{1}{2} \sum_{i=1}^{n} w_i^2$
- Red Elástica: $\frac{\rho}{2} \sum_{i=1}^{n} w_i^2 + (1-\rho) \sum_{i=1}^{n} |w_i|$ (Combinación convexa de las Normas $L_1$ y $L_2$ )

El algoritmo de Grandiente Estocástico Descendiente es un método de optimización sin restricciones, aproxima el valor real de $E(w, b)$ iterando sobre ejemplos de entrenamiento actualizando los parámetros del modelo con la siguiente regla:
$w \leftarrow w - \eta(\alpha \frac{\partial R(w)}{\partial w} + \frac{\partial L(w^T x_i + b, y_i)}{\partial w})$
donde $\eta$ es la tasa de aprendizaje que controla el tamaño de cada paso en el espacio parametral. El sesgo b se acualiza similar, pero sin considerar el factor de regularización $R(w)$.
Considerando $n$ iteraciones, y $t$ alguna en particular, tenemos las expresiones siguientes para $\eta$:

- $\eta^{(t)} = \frac{1}{\alpha(t_0+t)}$ (para clasificación)
- $\eta^{(t)} = \frac{\eta_0}{t^{power_t}}$ (para regresión)

$\alpha$, $\eta_0$ y $t^{power_t}$ son hiperparámetros.

### 3.3.11. Ensambles

Usar un algoritmo es bueno, pero usar varios algoritmos a la vez es mejor. Un algoritmo da una "perspectiva"de los patrones en los datos, pero múltiples algoritmos te dan múltiples perspectivas. De cierto modo, se les permite votar sobre una correcta clasificación o regresión sobre los resultados.Una vez que se tienen los resultados de cada modelo se puede optar por algún método heurístico: la media de los resultados o el que se repite más.
Construir un Ensamble consiste en dos pasos: (1) Construir Modalos variados, (2) Combinar sus estimaciones. A cada modelo se le pueden asignar pesos, se le puede encomendar cierta parte de los datos o variables, se puede particionar el espacio, etc.

Sólo para ejemplificar la mejora que ofrecen los ensambles, los siguientes diagramas nos dan el Antes y el Después de la combinación de varios modelos con un mismo objetivo.

Antes de aplicar Ensambles

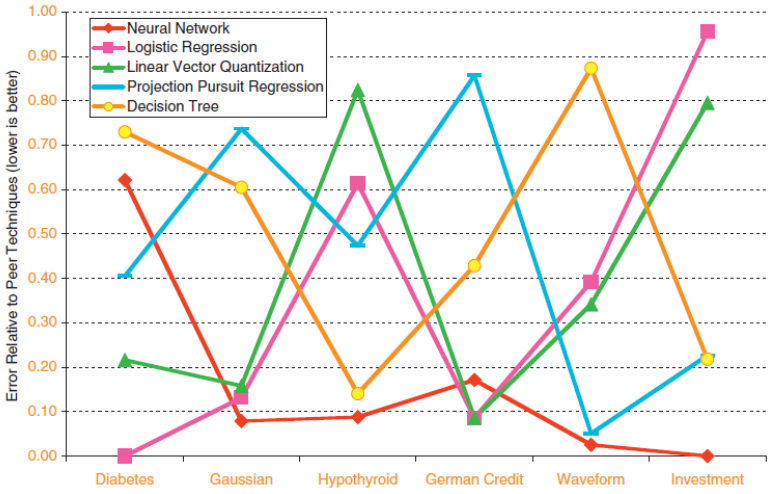



Figura 10: En este ejemplo sobre regresión, vemos el error relativo esperado de diferentes modelos sobre un mismo experimento .

Después de aplicar Ensambles

Como podemos ver, los ruidos que mantenían por separado los modelos, se omiten tras considerar sólo aquellas mejores clasificaciones o regresiones. Es por eso que actualmente no hay mejores modelos que los ensambles.

Dos de los principales tipos de Ensambles son: Bagging y Boosting.

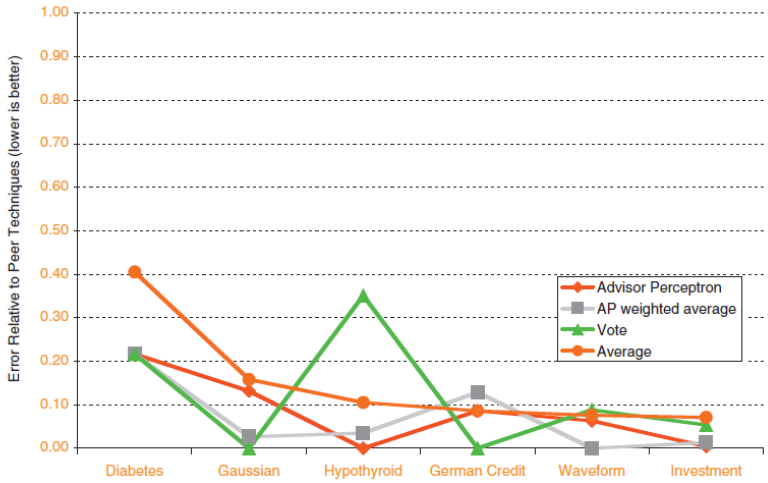



Figura 11: Aquí, el error relativo continúa, pero ya sólo se consideran debajo de cierto umbral. Es la mejora manifiesta que ofrecen los ensambles.

*Bagging (Bootstrap Aggregating)*
El término Boostraping suele utilizarse para hacer referencia a empezar algo sin recursos o con muy pocos recursos. Éste algoritmo crea $M$ réplicas del conjunto de datos, ajusta un modelo a cada réplica y obtiene el promedio de los resultados de cada modelo.

*Boosting*
En este algoritmo se crean modelos variados ponderando según para qué algoritmos fue más o menos difícil el modelar correctamente. Los casos más difíciles tienen un mayor peso, mientras que para los más fáciles es menor.
Primero se distribuye uniformemente el peso para después, para $j = 1, ...M$: Se ajusta un clasificador $f_j(x)$ usando lo pesos uniformes, luego se incrementa el peso para los casos de predicción pobres y al final se quita el peso a los casos bien predichos.
Después, se conbinan $f_1(x), f_2(x), ..., f_M(x)$ para general el clasificador potenciado dando mayor peso a los primeros modelos realizados.

De particular interés en nuestro estudio son los ensambles conocidos como Random Forest (Bosque Aleatorio) y AdaBoost.

*Random Forest*
Se trata de un ensamble de árboles de decisión, donde cada árbol es construido tras una muestra con reemplazo de los datos de entrenamiento. Como resultado de esta aleatoriedad, el sesgo del bosque aumenta ligeramente (respecto al sesgo de un solo árbol no aleatorizado) pero su varianza también disminuye compensando el aumento en el sesgo, lo que resulta en un mejor modelo.

*AdaBoost*

Este algoritmo entrenacon un conjunto de modelos débiles en diferentes versiones de los datos. Así, las predicciones de los modelos componentes se combina a través de un voto mayoritario ponderado para producir una predicción final. La iteración boosting consiste en ir cambiando los pesos $w_1, ..., w_n$ en cada una de las muestras de entrenamiento. Inicialmente, se distribuyen uniformemente los pesos, pero conforme el proceso aquellos modelos que predicen correctamente irán perdiendo peso y el recíproco aplica. De esta manera, los modelos débiles se concentran en lo no tratado por los otros modelos.

## 3.4. Situación Actual

*Un caso de estudio*
Indus Insights es una empresa fundada en 2009 que aplica Analytics para generar valor en negocios orientando a las compañías para que adopten estrategias conducidas en sus datos. La compañía mejoró la activación de tarjetas y KPI's relacionados de un 20 a un 50 % para uno de los mejores cinco bancos a nivel mundial usando Análisis de Comportamiento y programas específicos (por cuestiones de negocio, sólo se menciona a groso modo la labor de dicha empresa).

Lo anterior se logró identificando variables predictoras de actividad a largo plazo, para el corto plazo se desarrollan modelos, se califican y se redefinen con la experiencia. Usaron Árboles de Decisión para identificar variables predictoras y desarrollaron pruebas estadísticas para medir el impacto de particules intervenciones en el negocio de las tarjetas de crédito.

Los resultados que obtuvieron fueron que algunos comportamientos iniciales en particular influyen directamente en el uso a largo plazo de los productos. Se incentivaron, además, estrategias basadas en promociones, estrategias y mensajes a los clientes o posibles clientes. Con lo anterior se logró un incremento del 20 % en la activación de tarjetas de crédito y casi un 50 % de incremento de clientes que usan la Banca por Internet lo cual reduce considerablemente costos en sucursales.

# 4. Análisis exploratorio

## 4.1. Análisis univariante

Es un análisis básico, primario, en el cuál las carácteristícas o propiedades han de medirse una a una de modo univariado.

La sentencia .describe() nos proporciona las principales medidas de tendencia central, medidas de dispersion y de posición relativa de nuestras variables tales como la media, la desviación estándar, los cuántiles y los máximos y mínimos.

A continuación presentamos una muestra de los resultados de esta sentencia:

| | BCSCORE | MO_R__LC_MAX | MO_R__SALDO | V_MEDIA_B_D_TO_3 | V_MEDIA_CH_P_3 | V_SDO_DEBITO |
|---|---|---|---|---|---|---|
| **COUNT** | 9587.00 | 9617.00 | 9620.00 | 4947.00 | 7018.00 | 6899.00 |
| **MEAN** | 670.86 | 28051.18 | 12659.30 | 134.240 | 33442.28 | 29938.25 |
| **STD** | 121.38 | 43079.68 | 33855.04 | 7680.25 | 332175.52 | 419559.86 |
| **MIN** | -8.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **25 %** | 624.00 | 7500.00 | 0.00 | 0.00 | 1086.05 | 34.32 |
| **50 %** | 706.00 | 15500.00 | 3257.50 | 13.66 | 3691.52 | 1486.89 |
| **75 %** | 740.00 | 30000.00 | 13361.50 | 41.26 | 14277.09 | 8851.96 |
| **MAX** | 789.00 | 1005000.00 | 1742487.00 | 540209.88 | 22466157.64 | 30961052.20 |

Cuadro 1: Muestra de análisis univariante

A continuación mostramos un par de histogramas de las variables de la tabla:

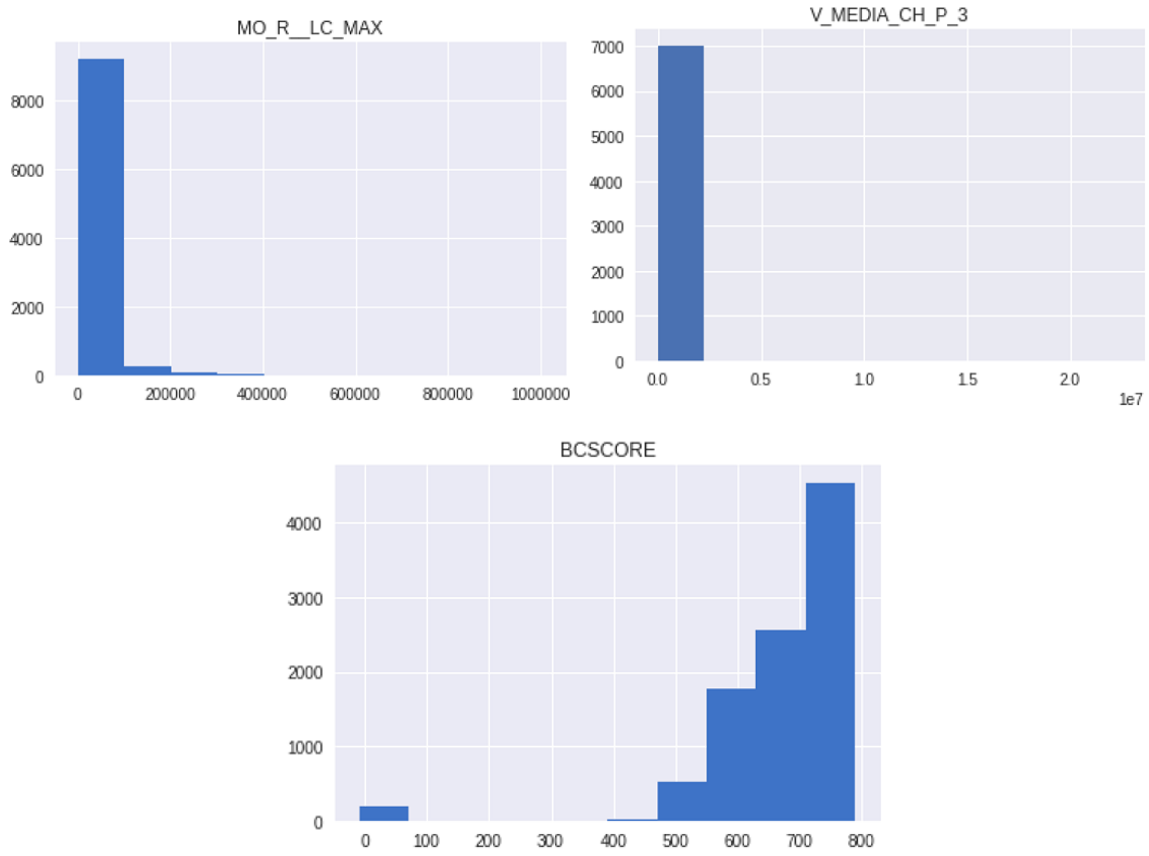



Figura 12: Ejemplos de histogramas de las variables de nuestra tabla

## 4.2. Análisis bivariante

Para el Análisis Bivariante obtuvimos la correlación de las variables continuas de nuestra tabla.

| | BCSCORE | MO_R__LC_MAX | MO_R__SALDO | V_MEDIA_B_D_TO_3 | V_MEDIA_CH_P_3 | V_SDO_DEBITO |
|---|---|---|---|---|---|---|
| **BCSCORE** | 1.00 | 0.141 | -0.024 | -0.006 | 0.022 | 0.022 |
| **MO_R__LC_MAX** | 0.141 | 1.00 | 0.539 | 0.009 | 0.324 | 0.187 |
| **MO_R__SALDO** | -0.024 | 0.539 | 1.00 | 0.005 | 0.471 | 0.178 |
| **V_MEDIA_B_D_TO_3** | -0.006 | 0.009 | 0.005 | 1.00 | -0.001 | -0.001 |
| **V_MEDIA_CH_P_3** | 0.022 | 0.324 | 0.471 | -0.001 | 1.00 | 0.727 |
| **V_SDO_DEBITO** | 0.022 | 0.187 | 0.178 | -0.001 | 0.727 | 1.00 |

Cuadro 2: Muestra de correlación entre variables

## 4.3. Valores extremos

Gracias al análisis Univariante y Bivariante pudimos notar que en efecto nuestra tabla tenía varios outliers en diferentes variables

Para evitar que estos valores extremos se volvieran un obstáculo para el desarrollo de los modelos eliminamos aquellos registros que eran menores al percentil de 0,025 y mayores al percentil de 0,975

## 4.4. Tramiento de missings

Para el tratamiento de los Missings decidimos no trabajar con aquellas variables que tuvieran como nulos más del 30 %. Para el resto de las variables imputamos con la mediana con el fin de optimizar nuestros modelos. A manera de comparativo mostraremos los histogramas resultantes despues del tratamiento de valores extremos y missings

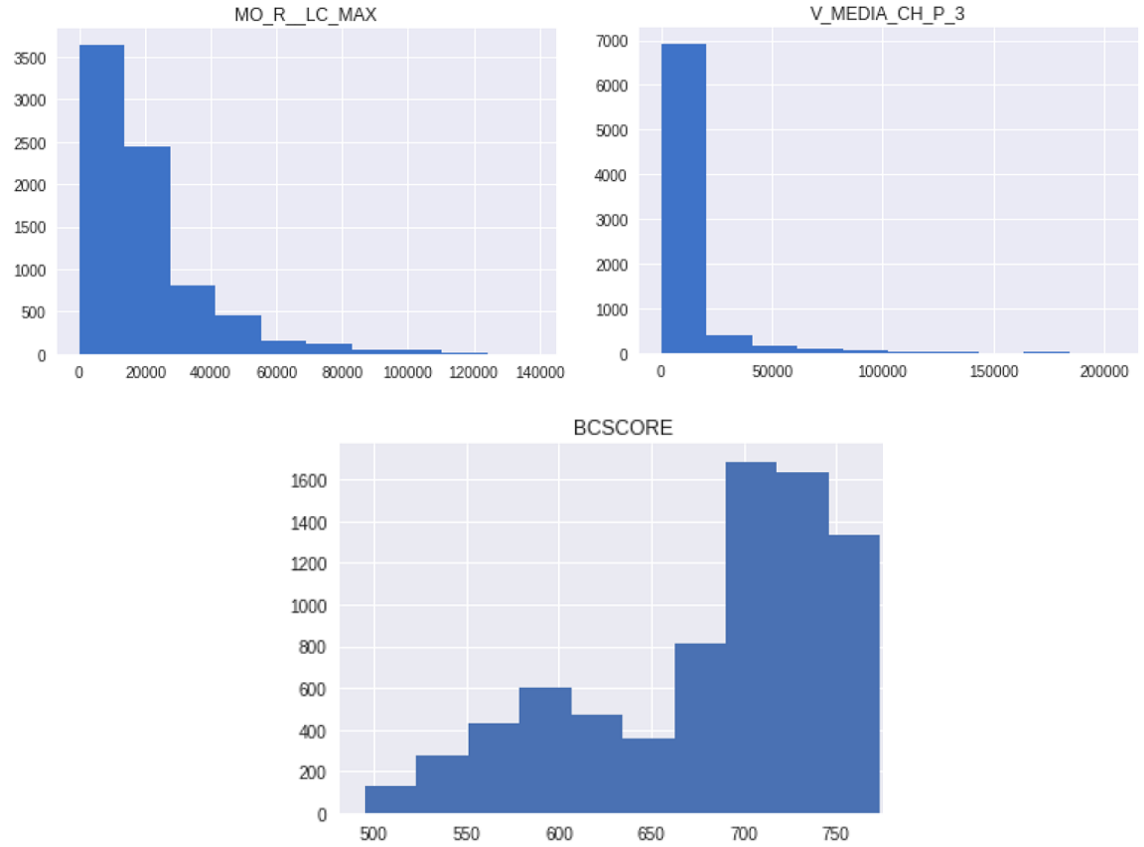



Figura 13: Histogramas despues del tratamiento de valores extremos y missings

## 4.5. Análisis de datos categóricos

Nuestra tabla presenta 300 variables discretas, pero no todas las variables son dicotómicas, por los que con ayuda de Python creamos una función para discretizarlas y aplicarles WOE.

Una vez que les aplicamos WOE las sustitumos en nuestra tabla por las variables originales, para así poderlas incluir en la construcción de algunos modelos.

# 5. Modelación no supervisada

## 5.1. Modelación no supervisada en Python

A continuación representamos en una gráfica lineal la inercia respecto del número de Clusters. En esta gráfica se aprecia un cambio brusco en la evolución de la inercia, obteniendo el Codo de Inercia. El punto en el que se observa ese cambio brusco en la inercia nos dirá el número óptimo de Clusters a seleccionar; para nuestro ejercicio son 4 Clusters.

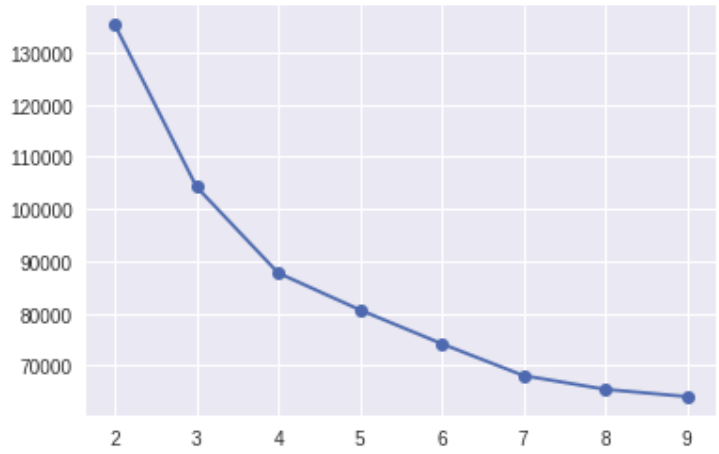



Figura 14: Codo de Inercia

Al realizar el Análisis de conglomerados obtenemos que los mejores resultados son los grupos obtenidos por el Modelo Gaussiano Mixto.

Utilizando la técnica de Escalamiento Multidimensional.

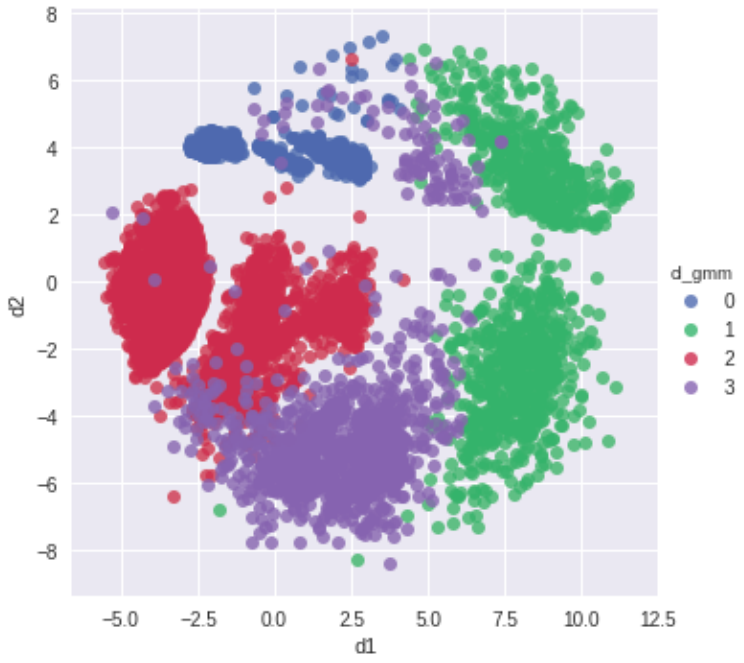



Figura 15: Gráfica de los grupos obtenidos bajo Análisis Clúster Y Esacalamiento Multidimensional

Utilizando Análisis de Componentes principales.


Figura 16: Gráfica de los grupos obtenidos bajo Análisis Clúster y Componentes principales.

La distribución de los clusters es la siguiente:


Figura 17: Gráfica de la distribución de los grupos.

### 5.1.1. Perfilamiento

Tomamos las variables más importantes para el modelo y para el Negocio para perfilar, se muestran en la siguiente tabla:

| *Perfilamiento* | Promedio de la cuenta en cheques en los útlimos 5 meses | Antigüedad en meses que el cliente lleva de relación con el Banco | Score de buró de crédito | Saldo del cliente con el Banco | Veces en las que se tuvo una línea de crédito mayor a $5,000.00 en los últimos 4 meses | Veces en las que se tuvo cuentas abiertas en los últimos 3 meses | Entrega de TdC en el domicilio del cliente |
|---|---|---|---|---|---|---|---|
| 0 | | | | | | | |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |

A partir del perfilamiento obtenemos las características de cada grupo que serán analizadas a continuación.

### Cliente Principiante

Tabla con las características principales del grupo Cliente Principiante.

| Grupo | Características |
|---|---|
| Cliente Principiante | Promedio bajo en la cuenta de cheques durante los últimos 5 meses<br>El cliente tiene muy poca antigüedad con el Banco<br>Score de Buró de Crédito medio<br>Saldo con el Banco muy bajo<br>Pocas veces tuvo una línea de crédito mayor a $ 5,000.00 en los últimos 4 meses<br>Pocas veces tuvo cuentas abiertas en los últimos 3 meses<br>Casi siempre se entregó la TdC en el domicilio del cliente |

Activación de TdC dentro del grupo:

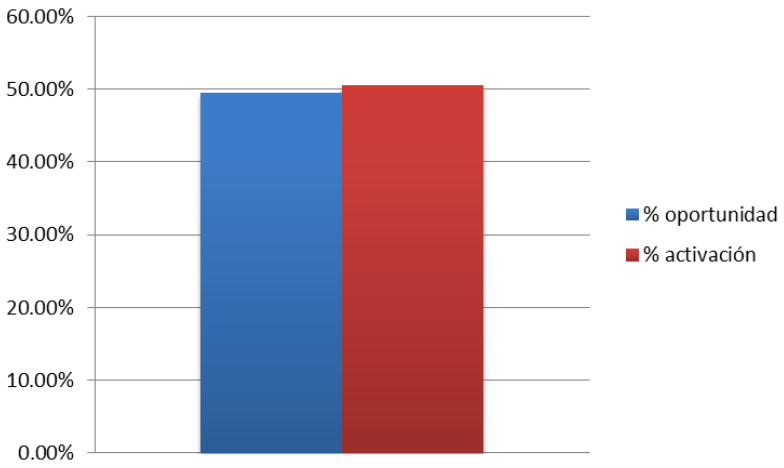



Figura 18: Gráfica de la distribución de la activación dentro de Clientes Principiantes.

### Cliente Ventajoso

Tabla con las características principales del grupo Cliente Ventajoso.

| Grupo | Características |
|---|---|
| Cliente Ventajoso | Promedio medio en la cuenta de cheques durante los últimos 5 meses<br>El cliente tiene poca antigüedad con el Banco<br>Score de Buró de Crédito alto<br>Saldo con el Banco alto<br>Muchas veces tuvo una línea de crédito mayor a $ 5,000.00 en los últimos 4 meses<br>Muchas veces tuvo cuentas abiertas en los últimos 3 meses<br>Casi siempre se entregó la TdC en el domicilio del cliente |

Activación de TdC dentro del grupo:

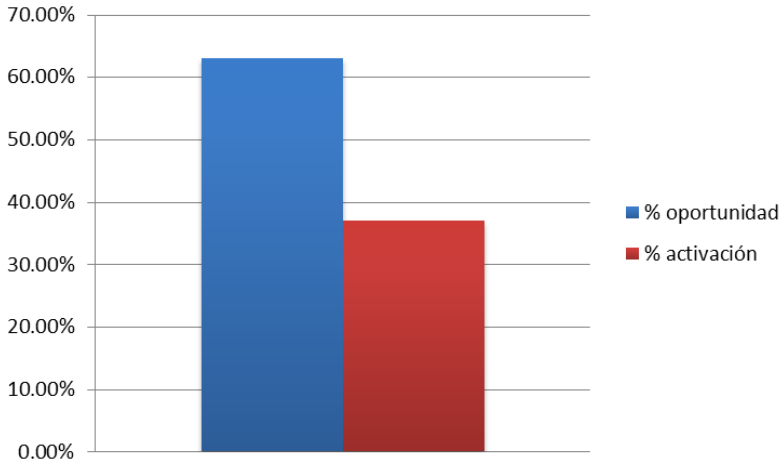



Figura 19: Gráfica de la distribución de la activación dentro de Clientes Ventajoso.

### Cliente Moderado

Tabla con las características principales del grupo Cliente moderado.

| Grupo | Características |
|---|---|
| Cliente moderado | Promedio alto en la cuenta de cheques durante los últimos 5 meses<br>El cliente tiene mucha antigüedad con el Banco<br>Score de Buró de Crédito bajo<br>Saldo con el Banco bajo<br>Pocas veces tuvo una línea de crédito mayor a $ 5,000.00 en los últimos 4 meses<br>Pocas veces tuvo cuentas abiertas en los últimos 3 meses<br>Casi nunca se entregó la TdC en el domicilio del cliente |

Activación de TdC dentro del grupo:

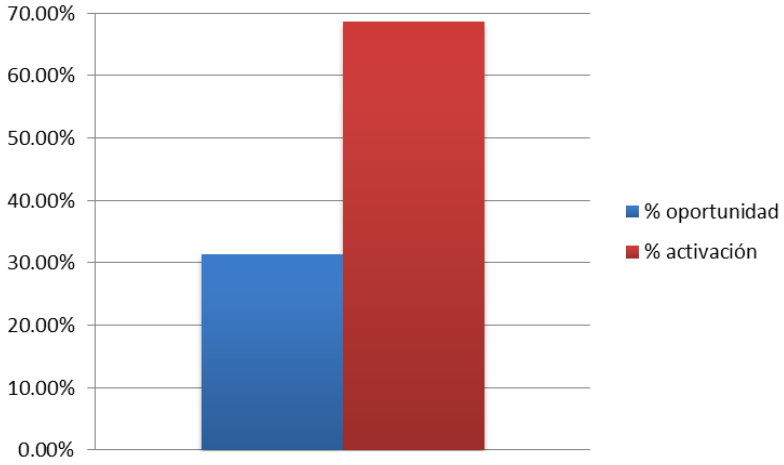



Figura 20: Gráfica de la distribución de la activación dentro de Clientes Moderados.

**Cliente Influyente**

Tabla con las características principales del grupo Cliente leal.

| Grupo | Características |
|---|---|
| Cliente Influyente | Promedio muy alto en la cuenta de cheques durante los últimos 5 meses<br>El cliente tiene gran antigüedad con el Banco<br>Score de Buró de Crédito medio<br>Saldo con el Banco alto<br>Muchas veces tuvo una línea de crédito mayor a $ 5,000.00 en los últimos 4 meses<br>Muchas veces tuvo cuentas abiertas en los últimos 3 meses<br>Casi nunca se entregó la TdC en el domicilio del cliente |

Activación de TdC dentro del grupo:

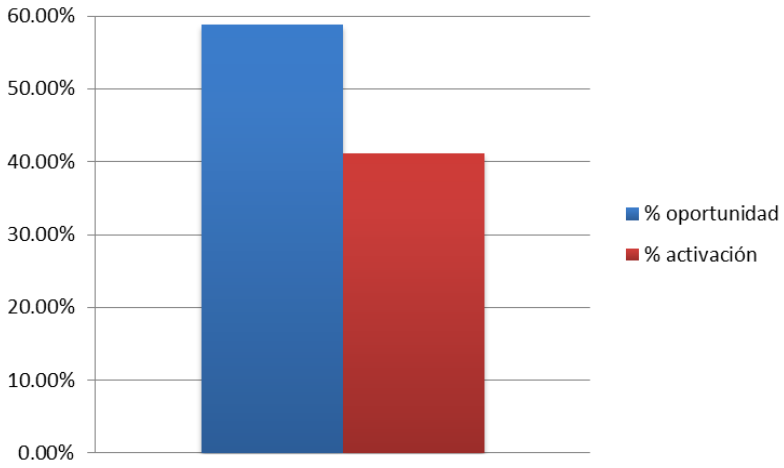



Figura 21: Gráfica de la distribución de la activación dentro de Clientes Influyentes.

## 5.2. Modelación no supervisada en SAS

Con la tabla lista para ser analizada, el siguiente paso es hacer el agrupamiento de clientes en base a características principales, primero hacemos un primer agrupamiento de variables para ver cuáles son las más significativas (Clustering de Variables).

En este primer agrupamiento el 90 % de la varianza se explica con 19 clusters, por lo tanto seleccionamos las variables de menor ratio.

Presentamos los resultados del perfilamiento que se realizó al comparar variables significativas utilizadas en el método K-means para 5 Clusters, todo esto se llevo a cabo después de realizar procedimiento ANOVA.

La distribución de los clusters es la siguiente:

### 5.2.1. Perfilamiento

Tomamos las variables más importantes para el modelo y para el Negocio para perfilar, se muestran en la siguiente tabla:

A partir del perfilamiento obtenemos las características de cada grupo que serán analizadas a continuación.

**Cliente frecuente**

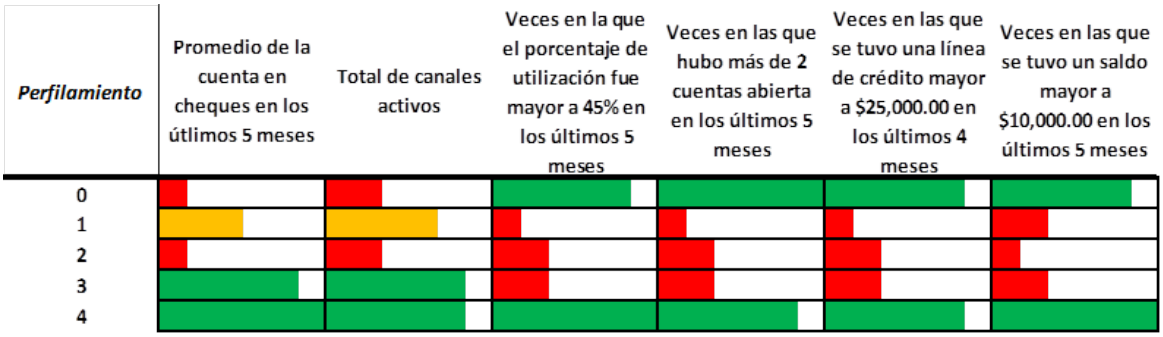
Tabla con las características principales del grupo Cliente frecuente.

Activación de TdC dentro del grupo:

**Cliente ausente**

Tabla con las características principales del grupo Cliente ausente.


Figura 22: Gráfica de la distribución de los grupos.




| Grupo | Características |
|---|---|
| Cliente frecuente | Promedio muy bajo en la cuenta de cheques durante los últimos 5 meses<br>Pocos canales activos<br>Su porcentaje de utilización fue mayor a 45 % casi siempre durante los últimos 5 meses<br>Siempre tuvo más de 2 cuentas abiertas en los últimos 5 meses<br>Casi siempre tuvo una línea de crédito mayor a $ 25,000.00 en los últimos 4 meses<br>Casi siempre tuvo un saldo mayor a $ 10,000.00 en los últimos 5 meses |

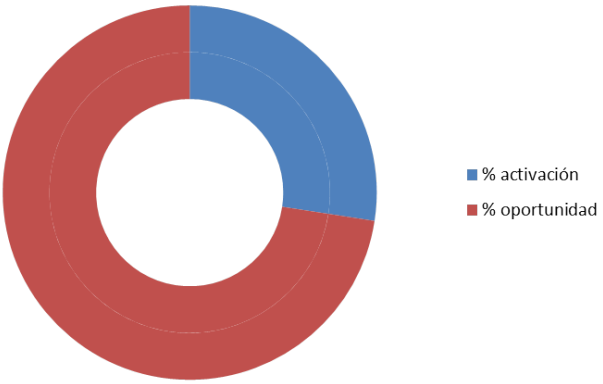



Figura 23: Gráfica de la distribución de la activación dentro de Clientes frecuentes.

Activación de TdC dentro del grupo:

### Cliente ocasional

Tabla con las características principales del grupo Cliente ocasional.
Activación de TdC dentro del grupo:

| Grupo | Características |
|---|---|
| Cliente ausente | Promedio medio bajo en la cuenta de cheques durante los últimos 5 meses<br>Número medio de canales activos<br>Su porcentaje de utilización fue mayor a 45 % casi nunca durante los últimos 5 meses<br>Casi nunca tuvo más de 2 cuentas abiertas en los últimos 5 meses<br>Casi nunca tuvo una línea de crédito mayor a $ 25,000.00 en los últimos 4 meses<br>Pocas veces tuvo un saldo mayor a $ 10,000.00 en los últimos 5 meses |

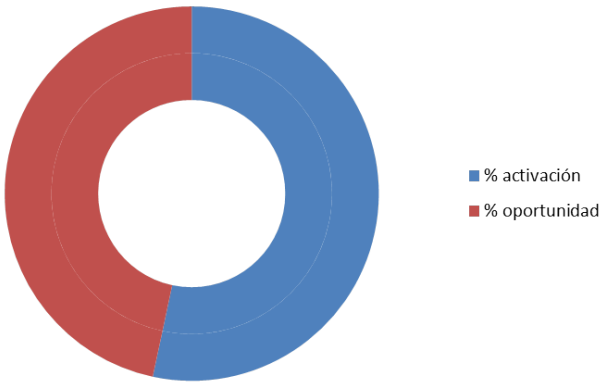



Figura 24: Gráfica de la distribución de la activación dentro de Clientes ausentes.

| Grupo | Características |
|---|---|
| Cliente ocasional | Promedio muy bajo en la cuenta de cheques durante los últimos 5 meses<br>Pocos canales activos<br>Su porcentaje de utilización fue mayor a 45 % pocas veces durante los últimos 5 meses<br>Pocas veces tuvo más de 2 cuentas abiertas en los últimos 5 meses<br>Pocas veces tuvo una línea de crédito mayor a $ 25,000.00 en los últimos 4 meses<br>Casi nunca tuvo un saldo mayor a $ 10,000.00 en los últimos 5 meses |

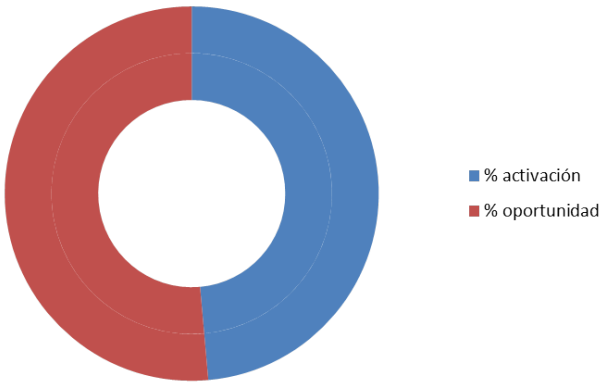



Figura 25: Gráfica de la distribución de la activación dentro de Clientes ocasionales.

### Cliente líquido

Tabla con las características principales del grupo Cliente líquido.
Activación de TdC dentro del grupo:

### Cliente experto

Tabla con las características principales del grupo Cliente experto.
Activación de TdC dentro del grupo:

| Grupo | Características |
|---|---|
| Cliente líquido | Promedio alto en la cuenta de cheques durante los últimos 5 meses<br>Muchos canales activos<br>Su porcentaje de utilización fue mayor a 45 % pocas veces durante los últimos 5 meses<br>Pocas veces tuvo más de 2 cuentas abiertas en los últimos 5 meses<br>Pocas veces tuvo una línea de crédito mayor a $ 25,000.00 en los últimos 4 meses<br>Pocas veces tuvo un saldo mayor a $ 10,000.00 en los últimos 5 meses |

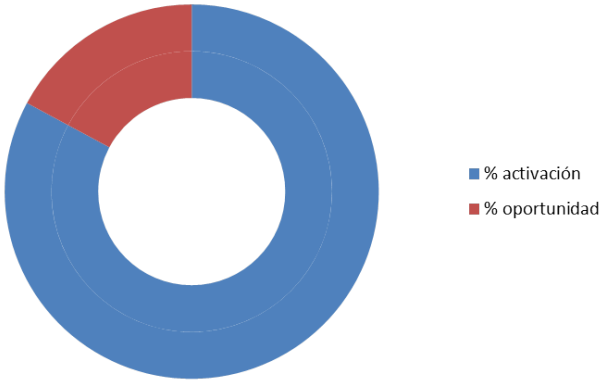



Figura 26: Gráfica de la distribución de la activación dentro de Clientes líquidos.

| Grupo | Características |
|---|---|
| Cliente experto | Promedio muy alto en la cuenta de cheques durante los últimos 5 meses<br>Muchos canales activos<br>Su porcentaje de utilización fue mayor a 45 % muchas veces durante los últimos 5 meses<br>Muchas veces tuvo más de 2 cuentas abiertas en los últimos 5 meses<br>Muchas veces tuvo una línea de crédito mayor a $ 25,000.00 en los últimos 4 meses<br>Muchas veces tuvo un saldo mayor a $ 10,000.00 en los últimos 5 meses |

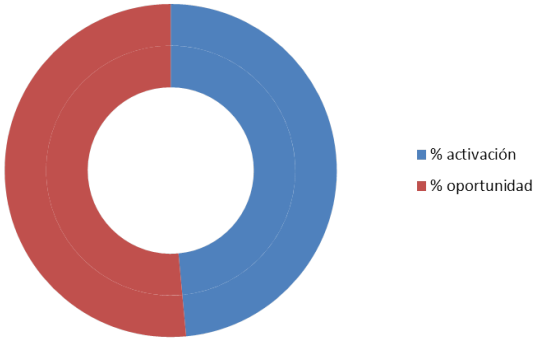



Figura 27: Gráfica de la distribución de la activación dentro de Clientes expertos.

## 5.3. Comparación de Modelos

Analizaremos las diferencias y similitudes que presentan los Modelos desarrollados en los diferentes softwares.

| Características | Python | SAS |
|---|---|---|
| Técnicas | Podemos observar de forma gráfica el agrupamiento de clústers.<br>Siempre elegimos el número de clústers que queremos formar.<br>Se utilizaron 288 variables para construir el Modelo, dichas variables fueron discretizadas bajo el procedimiento WOE.<br>Las mejores variables dadas por el Modelo se asemejaron mucho a las más representativas resultantes por el Clustering de variables. | Sólo podemos apreciar las distribución de los grupos.<br>Al utilizar el método Centride, éste asigna el número de Clústers a crear.<br>Se utilizaron 19 variables significativas para la construcción del Modelo. |
| De resultados | Resultó óptimo bajo el criterio de Codo de Inercia seleccionar 4 clústers.<br>Consideramos que a pesar de que los grupos resultantes del Modelo creado en el Software SAS discriminan bien, en éste software discriminan mejor.<br>La mitad de los grupos tiene más porcentaje de activación que de no activación, con una diferencia no muy grande.<br>Al perfilar los grupos notamos grandes similitudes con los grupos obtenidos por el otro Modelo. | Al comparar los resultados de los diferentes nodos con distintos números de grupos, el Método de K-Means arrojó un mejo resultado creando 5 Clústers.<br>Los grupos discriminan de forma óptima, aunque un grupo es notablemente más pequeño que los demás.<br>Al igual que en los resultados arrojados por el Modelo de Pyhton, aquí los mitad grupos también presentan mayor porcentaje de activación,con la diferencia que los porcentajes están más alejados. |

Al hacer una comparación profunda entre los resultados y las técnicas utilizadas en ambos Modelos, concluimos que los arrojan muy buenos resultados, sin embargo decidimos utilizar el Modelo creado en Python por la manera en que discrimina los datos.

# 6. Modelación supervisada

Con el propósito de cumplir con el objetivo de encontrar el modelo predictivo para determinar si un cliente nuevo activará o no su tarjeta, realizamos diferentes modelos, los cuales evaluamos con las métricas *Accuracy* y *ROC*.

Los modelos que desarrollamos fueron:
- *Regresión Logística*
- *Árboles de Decisión*
- *Redes Neuronales*
- *Análisis Discriminante*
- *Máquina Vector Soporte*
- *K-Vecinos*
- *Gradiante Estocástico Descendiente*
- *Naive Bayes*
- *Ensambles*

Algunos de ellos los desarrollamos en diferentes softwares, Python y SAS. En las próximas secciones revisaremos los resultudas obtenidos en cada Modelo.

## 6.1. Modelación supervisada en Python

En la siguiente tabla se muestran los valores de métrica para cada modelo:

| Modelo | ROC | | Accuracy | |
|---|---|---|---|---|
| | Train | Test | Train | Test |
| Regresión Logística | 80.10% | 81.00% | 72.40% | 73.90% |
| Árboles de Decisión | 84.50% | 83.20% | 73.50% | 73.90% |
| Análisis Discriminante | 84.00% | 84.40% | 76.60% | 76.90% |
| Máquina Vector Soporte | 83.80% | 85.20% | 76.90% | 78.00% |
| K-Vecinos | 83.00% | 81.40% | 74.80% | 73.60% |
| Gradiente Estocástico Descendiente | 83.90% | 83.30% | 75.20% | 74.40% |
| Naive Bayes | 77.20% | 77.20% | 68.60% | 67.40% |
| Ensambles | 91.10% | 90.40% | 84.00% | 83.30% |

Figura 6.1

Como se muestra en la *Figura 6.1*, el modelo de Ensambles fue el que arrojó un mejor valor de ROC sin sobreajustar.

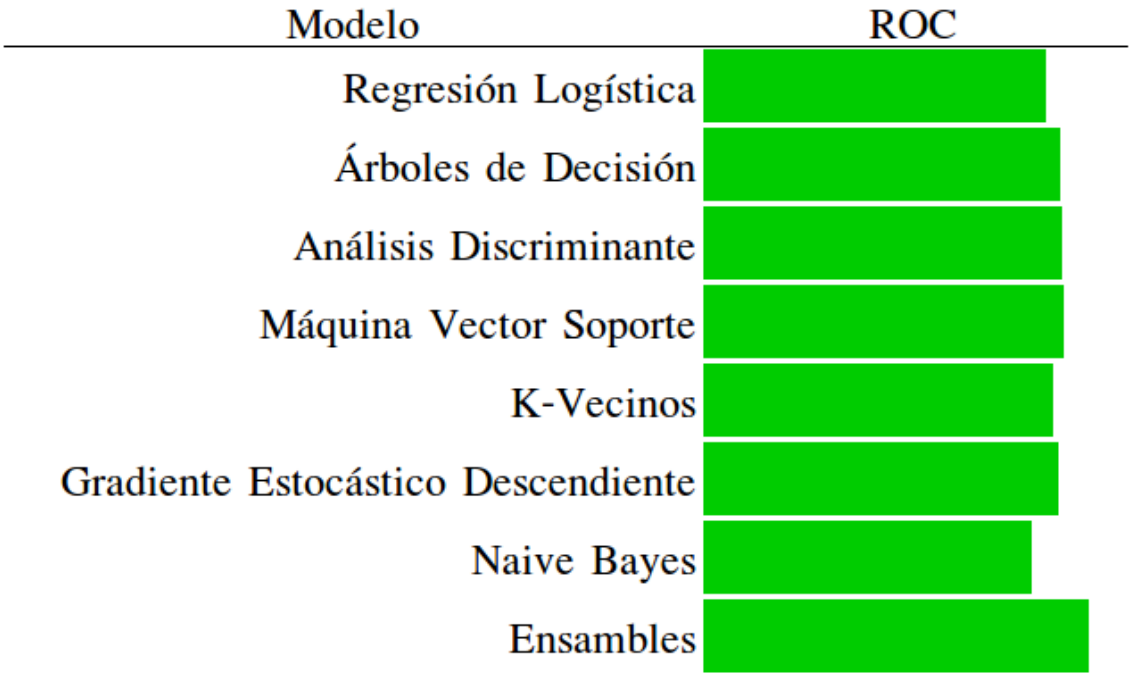

Modelo
ROC
Regresión Logística
Árboles de Decisión
Análisis Discriminante
Máquina Vector Soporte
K-Vecinos
Gradiente Estocástico Descendiente
Naive Bayes
Ensambles

Figura 6.1

Si analizamos ahora los valores de Accuracy, en la *Figura 6.2* para los diferentes modelos observamos que nuevamente es el modelo de Ensambles el que tiene los mejores valores:

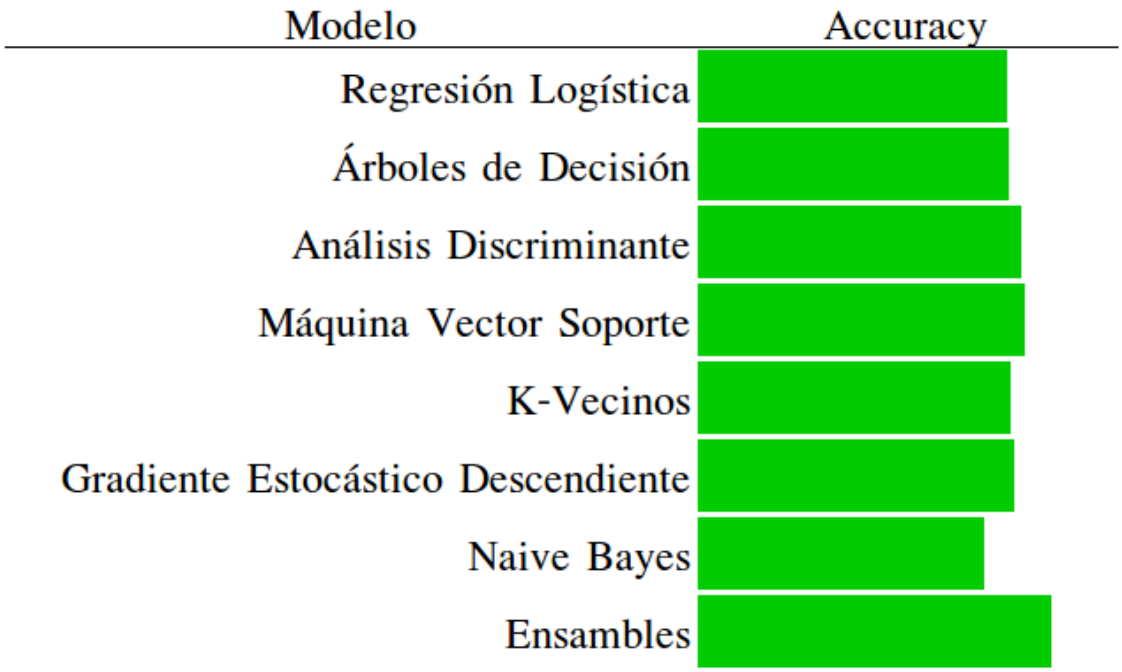



Figura 6.2

Por lo anterior podemos concluir que el modelo de Ensambles es el que mejor predice si un cliente nuevo activará o no su tarjeta.

### 6.1.1. Modelación supervisada en SAS

**Árbol de decisión**

El objetivo de hacer árboles de decisión es poder determinar qué variables son las que determinan que una proporción de la tabla completa o de los grupos de clientes tienen mayor probabilidad de activar su TdC. Los áboles también nos ayudan a saber de que tamaño es la proporción antes mencionada y qué porcentaje de probabilidad de activar la TdC existe.

Al comparar varior árboles, decidimos optar por un árbol de 3 ramas y de profundidad 4.

Obtenemos una Curva ROC de .80

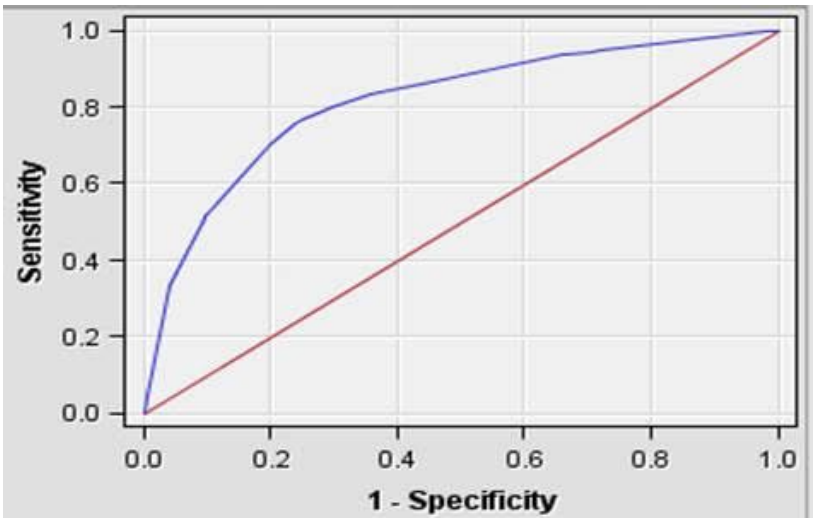



Figura 28: Curva ROC.

El árbol seleccionado es el siguiente:

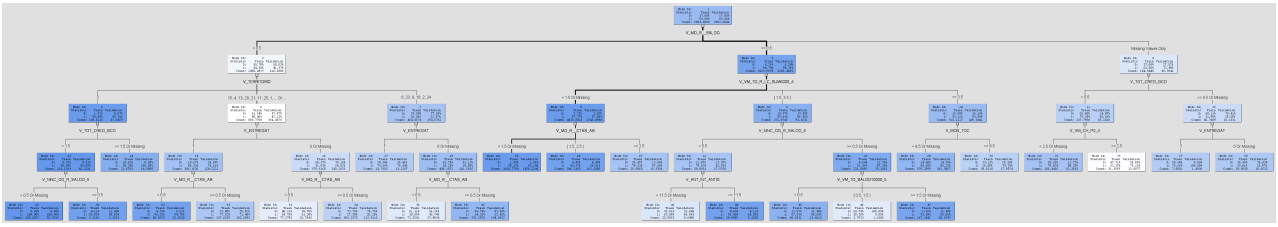


Figura 29: Árbol de 3 ramas y profundidad 4.

Observamos que la variable más importante para el Modelo es el Saldo del cliente con el Banco, las variables que siguen respecto a importancia son: el territorio en el que se asignó la TdC, el número de veces en el que la suma de Líneas de Crédito totales fueron mayor a $ 5,000.000 en los últimos 4 meses y el total de crédito que se tiene con el Banco.

- *Peor nodo*

Tiene una probabilidad de no activar la TdC de 85.75 % y una probabilidad de activación de 14.26 %.

Las reglas que sigue éste Nodo son las siguientes:

- Saldo del cliente con el Banco <0.5
- Territorio en el que se asignó la TdC 3
- Total de crédito que se tiene con el banco <1.5

-Número de incrementos con otro otorgante en el saldo de crédito revolvente en los último 5 meses <0.5 o missing.

- *Mejor nodo*

Tiene una probabilidad de no activar la TdC de 0 % y una probabilidad de activación de 100 %

Las reglas que sigue éste Nodo son las siguientes:

- Saldo del cliente con el Banco <0.5
- Territorio en el que se asignó la TdC 6, 23, 9, 16, 2, 24
- Entrega de TdC a domicilio 0 o missing.

-Número de cuentas abiertas con el Banco >= 0.5 o missing.

**Regresión logistica**

Como ya se ha mencionado, la regresión logística es una técnica estadística que nos permite predecir el resultado de una variable categórica la cual está en función de un conjunto de variables observables las cuales se conocen como independientes o predictoras. En esta ocasión atenderemos el problema de determinar qué clientes son más propensos a activar la TdC que se les otorgó.

Para éste modelo utilizamos el método de Stepwise.

Obtenemos una Curva ROC de .867

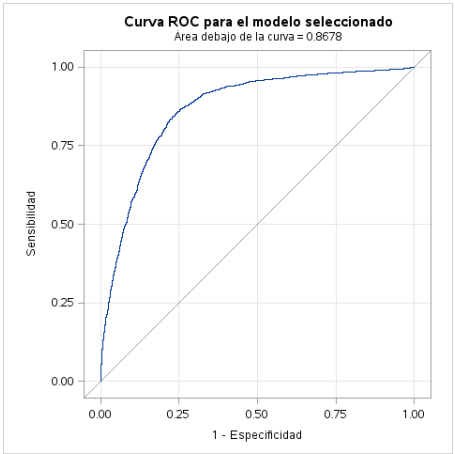



Figura 30: Curva ROC.

A continuación mostraremos los gráficos finales:

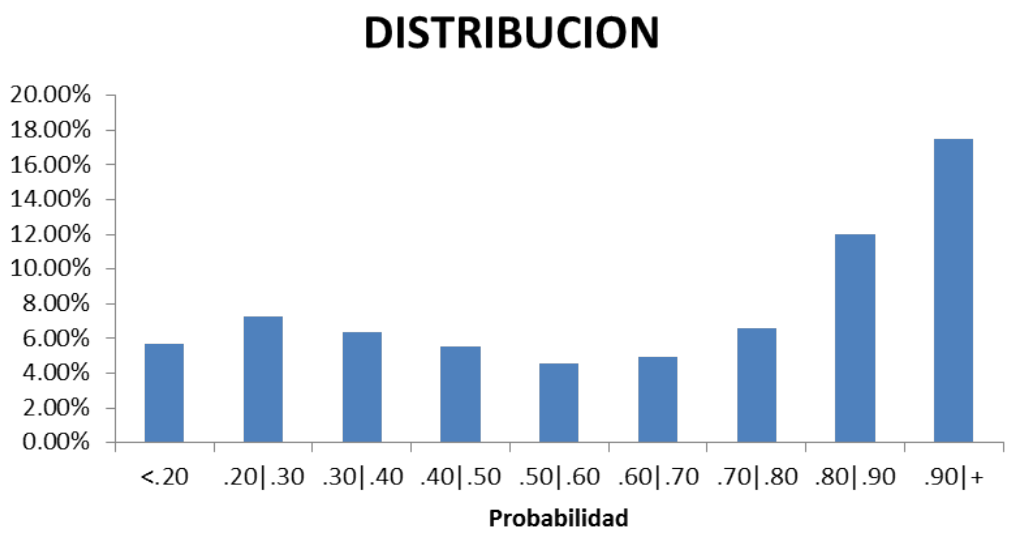



Figura 31: Distribución de los rangos de probabilidad.

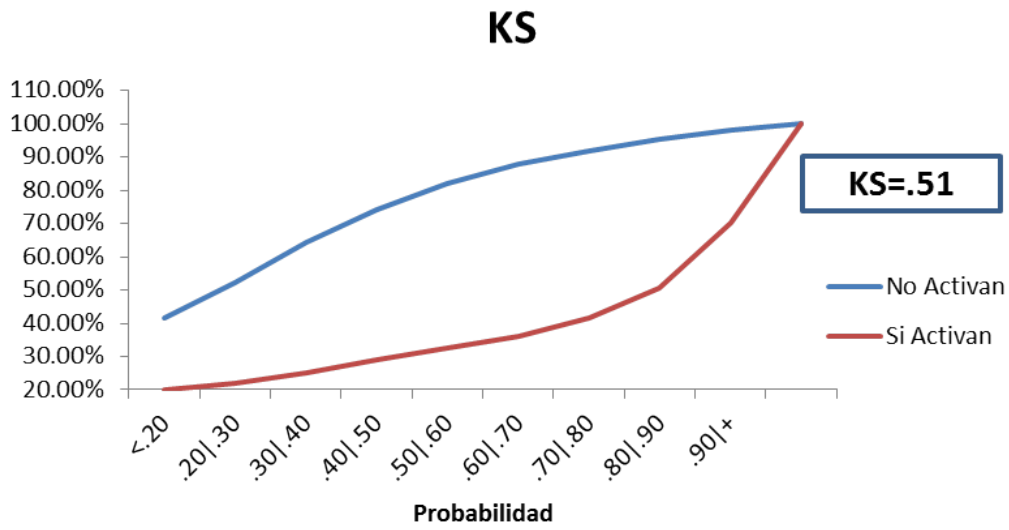



Figura 32: Estadigrafo KS.

Observamos que tenemos gran área de oportunidad para la activación de TdC, ya que los rangos de probabilidad más altos son los los más propensos a la activación.

El modelo resulta muy atractivo al presentar un valor ROC alto y al ajustarse en gran porcentaje a una curva exponencial.

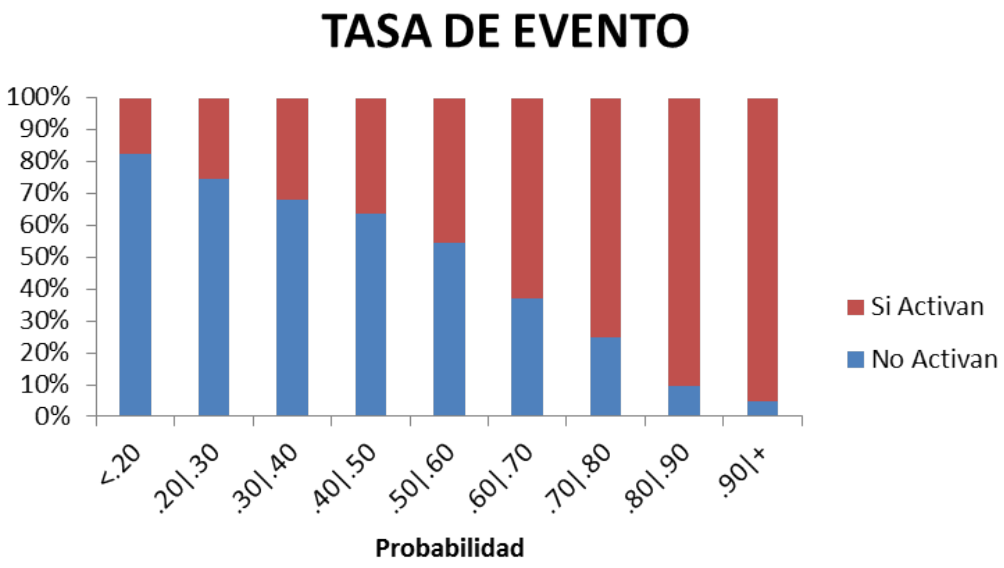



Figura 33: Proporción de evento.

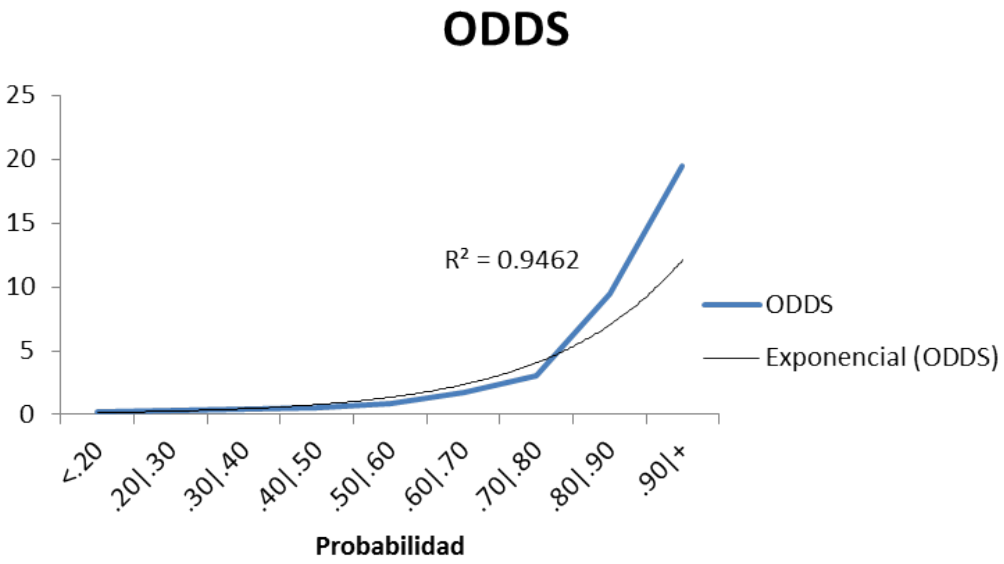



Figura 34: La curva se ajusta en un 94.6 % a una distribución exponencial.

## 6.2. Comparación de modelos

Analizaremos las diferencias y similitudes que presentan los Modelos desarrollados en los diferentes softwares.

| Características | Árbol de decisión en Python | Árbol de decisión en SAS |
|---|---|---|
| Técnicas | El código para realizar el modelo es secillo y breve<br>Al hiperparametrizar se pueden obtener mejores resultados. | EL proceso para crear el árbol a pesar de ser sencillo, resulta más largo que Python.<br>Al construir diferentes nodos, tenemos la opción de elegir el que más convenza según los resultados. |
| De resultados | Curva ROC de .832<br>No se cuenta con una visualización gráfica del árbol. | Curva ROC de .80<br>Tiene una representación visual muy fácil de interpretar los resultados de cada nodo y cual fue el flujo de ellos. |

Aunque el Árbol de decisión realizado en SAS sea muy fácil de interpretar, decidimos que es mejor el que se realizó en Python, ya que tiene un mejor nivel de predicción presentando un valor en Curva ROC mayor.

| Características | Regresión lógistica en Python | Regresión lógistica en SAS |
|---|---|---|
| Técnicas | El código para realizar el modelo es sencillo y breve.<br>Al hiperparametrizar se pueden obtener mejores resultados.<br>Al tener variables categóricas que no son dicotomicas se tuvo que programar una función que discretizara las variables con el WOE. | EL proceso para crear la regresión a pesar de ser sencillo, resulta más largo que Python.<br>Al construir diferentes nodos, tenemos la opción de elegir el que más convenza según los resultados.<br>Al ser SAS un softaware de explotación de datos y no un software de programación evita el realizar procedimiento WOE para éste modelo, sin embargo a veces si se tiene que hacer, como en modelos de Scoring. |
| De resultados | Curva ROC de .81<br>No se cuenta con una visualización gráfica de la regresión. | Curva ROC de .867<br>Tiene una representación visual muy fácil de interpretar los resultados de cada rango de probabilidad y su tasa de evento. |

Ambos modelos son buenos, pero tiene mayor cofianza de predicción el modelo elaborado en SAS al presentar una Curva ROC de .867

# 7. Estrategia de Negocio

## 7.1. Modelo para determinar si un cliente es propenso a activar la TdC

Elegimos un modelo de Regresión logistica, ya que tuvo el mayor valor en curva ROC,con un poder de predicción del 86.7 %, lo utilizamos para identificar cuáles serán los clientes con una alta probabilidad de activar la tarjeta de crédito y cuáles no, para de esta manera generar estrategias de negocio a fin de incrementar el índice de activación.

Al tener conocimiento de la propención del cliente a la activación, los pudimos segamentar en dos grupos, para implemetarles una estrategia distinta.

**Clientes potenciales a activar**
Este grupo de clientes tiene una gran probabilidad de activar la tarjeta, mayor al 50 %.

**Clientes no propensos a activar**
Este grupo de clientes tiene baja probabilidad de activar la tarjeta, de acuerdo a los resultados del modelo, con una prbabilidad menor al 50 %.

### 7.1.1. Orientación de esfuerzos

Esta estrategia va dirigida hacia los Clientes potenciales a activar,ya que este grupo de clientes tiene una gran probabilidad de activar la tarjeta por lo que para garantizar la activación basta con incentivarlos a través de canales múltiples como pueden ser la banca móvil, llamadas telefónicas o envío de correo.

**Impacto**

### 7.1.2. Venta cruzada

La venta cruzada es una estrategia de venta, para incrementar las compras, en nuestro ejercicio nos interesa aumentar el índice de activación de TdC, a través de la venta de productos diferentes complementarios o adicionales.

Dadas las pocas posibilidades de que los clientes nos propensos a activar, activen su tarjeta de crédito, proponemos incentivarlos por medio de Venta Cruzada, cabe señalar que esta es sólo una sugerencia que está sujeta al marketing de la empresa.

La Venta Cruzada será la estrategia sugerida para los clientes no propensos a activar y esta será diferente para cada perfil del cliente, según el modelo no supervisado que se realizó para la segmentación de clientes de acuerdo a sus características.

Actualmente sin implementar la estrategia estudiada, el costo de adquisición, es decir la pérdida es de $ 3 MM, los ingresos netos a los 12 meses son de $ 13 MM y la contribución a cartera a los 12 meses es de $ 77 MM. Implementando la estrategia sólo se pierden $ 1.2 MM, el ingreso neto aumenta a $ 17 MM y la contribución a carterar a $100 MM. **Cliente Moderado**

A este cliente sugerimos **ofrecerle un producto premium** con características más atractivas para persuadirlo a hacer uso de la tarjeta y por tanto activarla.

**Cliente Ventajoso**
Como este cliente tiene múltiples cuentas con otras instituciones financieras lo que sugerimos es ofrecerle la posibilidad de **transferir su saldo** de otras instituciones a la nuestra a fin de incentivarlo a la activación de nuestro producto.

**Cliente Influyente**
De este cliente pudimos notar que ya tenía previamente algún producto con nosotros, así que la propuesta es **homologar sus líneas de crédito**, esto quiere decir sumar la línea de crédito de la tarjeta no activada a la línea de crédito de la tarjeta que ya tiene activada.

**Cliente Principiante**
El cliente principiante puede incentivarse a activar su tarjeta si le **otorgamos crédito** haciendo ver a nuestro producto más atractivo.

### 7.2. Análisis de grupos

En nuestros grupos de clientes notamos que dos tipos clientes son poco propensos a activar la TdC, debido a que ya están penetrados por el mercado, estos son el Cliente influyente y el Cliente Ventajoso; mientras que los dos otros tipos de clientes al no manejar tantos créditos son más propensos a activar. Esas observaciones nos permiten decidir a un futuro qué perfiles son los que nos convendrían incluir en nuestro portafolio debido a su alta propensión a la activación y con cuáles al no incluirlos generaríamos un ahorro.

## 8. Conclusiones

Al crear varios modelos supervisados y no supervisados para competir, decidimos bajo criterio de eficiencia de modelo y análisis de Negocio optar por un modelo no supervisado arrojado por K-means con 4 grupos de clientes: Cliente principiante, cliente ventajoso, Cliente moderado y Cliente influyente.

Mientras que para la elección del modelo supervisada, optamos por una regresión logística, ya que además de resultar un buen modelo predictivo nos da la oportunidad de crear una estrategia de Negocio basada en los rangos de probabilidad más altos que además tienen mayores tasas de evento.

A continuación enunciaremos resultados más específicos.

1. Observamos que los grupos más propensos a la activación de TdC son: Cliente Principiante y Cliente Moderado. Decidimos que en un futuro se deben de otorgar nuevas TdC a clientes con perfiles similares, ya que éste tipo de clientes tienen una mayor propensión a la activación. Los clientes pertenecientes a los otros grupos (Cliente Ventajoso y Cliente Influyente), tienen como característica en común que tienen gran antigüedad con el Banco y varios créditos, por lo que deciden no activar un TdC más.

2. Al implementar la Estrategia de Negocio creada con ayuda de nuestro Modelo podemos reducir perdidas hasta un 37 %, aumentar ingresos en un 31 % e incrementar la cartera un 30 %.

3. Se decide observar la respuesta de venta cruzada, y en base a los resutados cambiar criterios.

# 9. Bibliografía general


1. Sánchez Carrión, Juan J. 1995. "Manual de Análisis de Datos". Alianza Editorial, Madrid.
2. Nisbet, R., Elder, J. y Miner, G. 2009. "Handbook of Statistical Analysis and Data Mining Applications". Academic Press, Canada.
3. Banks J., Carson J.S., Nelson B.L, 1996, "Discrete-Event System Simulation. Second Edition.", Prentice-Hall, New Jersey.
4. Fishman G.S., 1978, "Conceptos y métodos en la simulación digital de eventos discretos", Limusa, México.
5. Kelton W.D., Sadowski R.P., Sadowski D.A., 1998, "Simulation with Arena", Mc Graw Hill, Boston.
6. Ogunnaike B.A., Harmon Ray W., 1994, "Process Dynamics, Modeling and Control", Oxford, New York.
7. Shannon R.E., 1988, "Simulación de Sistemas. Diseño, desarrollo e implementación", Trillas, México.
8. Law A.M., Kelton W.D., 1991, "Simulation Modeling & Analysis", Second Edition, McGraw-Hill, New York.
9. Gibbans J.D., 2003, "Nonparametric Statatistical Inference", Fourth Edition, Marcel Dekker, New York.
10. Everitt B.S., 2011, "Cluster Analysis", Fifth Edition, John Wiley and Sons, Ltd, United Kingdom.
11. Carmona, E. 2014. "Tutorial sobre Máquinas de Vectores Soporte (SVM)". Depto. de Inteligencia Aritificial, España.
12. Indus Insights. 2013. 'Indusinsights: A case study'. India. Disponible en: http://www.indusinsights.com/case-studies/
13. Pedregosa, F. y otros (Desarrolladores de Scikit-learn). (2007-2017). "Scikit-learn: Machine Learning in Python". INRIA and others. Obtenido de: http://scikit-learn.org/stable/supervised_learning.html#supervised-learning

# 10. Anexo

## 10.1. Código fuente en Python

Listing 1: Código en Python

```
1
2  # coding: utf-8
3
4  # # Proyecto Final
5
6  # In[152]:
7
8  #!/usr/bin/env python
9
10 '''
11 Librerías que ocupamos en todo el código
12 '''
13 from __future__ import division
14 import pandas as pd
15 import numpy as np
16 import seaborn as sns
17 import matplotlib.pyplot as plt
18 pd.set_option('display.max_columns',500)
19 get_ipython().magic(u'matplotlib inline')
20 import pickle
21 from sklearn.preprocessing import MinMaxScaler, StandardScaler
22 from sklearn.decomposition import PCA
23 from sklearn.manifold import MDS
24 from sklearn.cluster import AgglomerativeClustering,KMeans
25 from sklearn.mixture import GaussianMixture
26 from sklearn.feature_selection import SelectKBest
27 from sklearn.model_selection import GridSearchCV,train_test_split
28
29
30 # In[2]:
31
32 '''
33 Carga de los datos
34 '''
35 ruta='/home/dbh/Documentos/AMV/Proyecto AMV/'
36
37 df = pd.read_excel('/home/dbh/Documentos/AMV/Proyecto AMV/SAMPLE_MDL_ACT (1).xlsx',encoding='utf-8').reset_inc
38
39 df.head()
40
41
42 # ## Funciones de ayuda
43
44 # In[3]:
45
46 '''
47 Funciones para la limpieza de datos
48 '''
49 def miss_u(df,x):
50     '''
51     df=dataframe
52     0<x<1
53     return lista de variables que tienen menos missings que x
54     '''
55
56     aux = df.describe().T[['count']]
57     aux/=len(df)
58     aux=1-aux
59     var_fin = list(aux[aux['count']<=x].reset_index()['index'])
60     return var_fin
```

```
def percent(df,i=0,o=.01):
    '''
    df=dataframe
    i=decimales de la o en porcentaje
    0<o<1'desde que que percentil se toman los outliers'

    return dataframe sin outliers
    '''

    c=1-o
    a='%.'+str(i)+'f'
    l_b=a%(o*100)+'%'
    u_b=a%(c*100)+'%'
    aux = df.describe(percentiles=[o,c]).T[[l_b,u_b]]
    aux.reset_index(inplace=True)
    for i,row in aux.iterrows():
        df['ol_%s'%row['index']] = ((df[row['index']]<row[l_b])|
         (df[row['index']]>row[u_b])).astype(int)
    var_ol = [x for x in df.columns if x[:2]=='ol']
    df ['ol'] = (df[var_ol].sum(axis=1)>=1).astype(int)
    af = df[df.ol!=1].copy()
    af = pd.DataFrame(af,columns=df.columns)
    return af
def imputar(df,estrategia='median'):
    '''
    df=dataframe
    estrategia=['mean','median','most_frequent'], por default median
    return datframe con estrategia imputada
    '''

    from sklearn.preprocessing import Imputer
    im = Imputer(strategy='median')
    im.fit(df)
    df_1 = pd.DataFrame(im.transform(df),columns=df.columns)
    return df_1
def unico(lista):
    '''
    lista=lista
    return= el valor unico que aparece o indica 'No es único en caso contrario'
    '''
    if len(list(set(lista)))==1:
        return 'Si es unico'
    else:
        return ('No es unico',list(set(lista)))
def woe_code(df,var_disc):
    for v in var_disc:
        aux = df[[v,'ID','TARGET']].copy()
        aux = aux.pivot_table(aggfunc='count',
                              columns='TARGET',
                              fill_value=0,
                              index=v)
        aux = aux['ID']
        aux.reset_index(inplace=True)
        aux

        for i in range(2):
            aux[i]/=aux[i].sum()

        aux['woe'] = np.log(aux[0]/aux[1])
        a=((aux[0]-aux[1])*aux['woe']).sum()
        print "IV variable %s = %.3f " %(v,a)
        if 0.02<=a<=.5:
            df = df.merge(aux[[v,'woe']],on=v,how='inner')
            #df.drop(v,axis=1,inplace=True)
            df.rename(columns={'woe':'W_'+v},inplace=True)
    return df
```

```
def metricas(model,Xt,Xv,yt,yv):
    from sklearn.metrics import accuracy_score,roc_auc_score
    print "ROC train:%.3f | ROC test:%.3f " %(roc_auc_score(y_true=yt,y_score=model.predict_proba(Xt)[:,1]),ro
    print "ACC train:%.3f | ACC test:%.3f " %(accuracy_score(y_true=yt,y_pred=model.predict(Xt)),accuracy_scor


# ## Variables continuas y discretas

# In[4]:

'''
Definición de variables continuas y discretas
'''
var_cont = ['V_RT_OO_R_LC_SUM_OO_R_CTAS_AB_3', 'V_RT_TO_R_LC_SUM_TO_CTAS_AB_3', 'V_RT_TO_R_SALDO_TO_SALDO_3',

var_disc = [str(x) for x in df.columns if x not in var_cont]


# ## Tabla Limpia

# In[5]:

var_miss=miss_u(df,.3)

var_cont_1=[str(x) for x in var_cont if x in var_miss]+['ID']

var_disc_1=[str(x) for x in var_disc if x in var_miss]

df_cont=df[var_cont_1].copy()
df_disc=df[var_disc_1].copy()

df_cont_1=percent(df_cont,1,.025)[var_cont_1]
df_cont_1=imputar(df_cont_1,'median')

df_cont_1.head()


# ## Grafica de variables continuas

# In[6]:

'''
Histogramas
'''
df_cont_1.hist()


# ## Dataframe final

# In[7]:

'''
DataFrame Final
'''
df_final=pd.merge(df_disc,df_cont_1,how='inner',on='ID')

df_final.head()


# In[8]:

df_final.shape


# ## Variables Continuas y Discretas

```

```
# In[9]:

'''
Variales continuas y discretas
'''
var_cont_f=[str(x) for x in df_cont_1.columns if x != 'ID']

var_disc_f=[str(x) for x in var_disc_1 if (x!='ID' and x!='TARGET')]


# ## WOE , IV

# In[10]:

'''
WOE, IV
'''
af=woe_code(df_final,var_disc_f)


# In[11]:

for v in var_cont_f:
    af.rename(columns={v:'x_%s'%v},inplace=True)
    af[v] = pd.cut(af['x_%s'%v],bins=5).astype(str)

af = woe_code(af,var_cont_f)

af.drop(var_cont_f,axis=1,inplace=True)

af.rename(columns=dict(zip(['x_%s'%v for v in var_cont_f],var_cont_f)),inplace=True)


# ## Dataframe con WOE en pickle

# In[12]:

'''
Guardamos el Dataframe con WOE
'''
pickle.dump(af,open('Sample_MDL_ACT_con_WOE','wb'))

'''
Empezamos con la modelación supervisada
'''


# ### -Regresion Logistica

# In[13]:

'''
Regresión Logistica
'''
from sklearn.linear_model import LogisticRegression


# In[14]:

model = LogisticRegression()

var_woe = [v for v in af.columns if v[:2]=='W_']

X = af[var_woe]
y = af['TARGET']

```

```
Xt,Xv,yt,yv = train_test_split(X,y,train_size=0.7)

model.fit(Xt,yt)


# #### -Resultado-

# In[15]:

metricas(model,Xt,Xv,yt,yv )


# ### -Arboles de decision

# In[16]:

'''
Arboles de decisión
'''
from sklearn.tree import DecisionTreeClassifier


# In[17]:

model = DecisionTreeClassifier()

df = pickle.load(open('Sample_MDL_ACT_con_WOE','rb'))



# In[18]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()

y = df['TARGET'].copy()

Xt,Xv,yt,yv = train_test_split(X,y,train_size=0.7)

param_grid = dict(criterion = ['gini','entropy'],
splitter = ['best','random'],
max_features = range(2,13),
max_depth = range(3,15))

grid = GridSearchCV(cv=3,
                    verbose=True,
                    estimator=model,
                    param_grid=param_grid,
                    n_jobs=-1,
                    scoring='roc_auc')

grid.fit(X,y)

model = grid.best_estimator_
model.fit(Xt,yt)


# #### -Resultados-

# In[19]:

metricas(model,Xt,Xv,yt,yv)


# ### -RedesNeuronales

# In[20]:
```

```
325
326 '''
327 Redes neuronales
328 '''
329 from sklearn.neural_network import MLPClassifier
330 from sklearn.preprocessing import MinMaxScaler,StandardScaler
331 from sklearn.decomposition import PCA
332 from sklearn.pipeline import make_pipeline
333 from sklearn.metrics import accuracy_score,roc_auc_score
334 from sklearn.model_selection import GridSearchCV,train_test_split,RandomizedSearchCV
335
336
337 # In[21]:
338
339 model = MLPClassifier()
340
341 df = pickle.load(open('Sample_MDL_ACT_con_WOE','rb'))
342
343
344 # In[22]:
345
346 X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()
347
348 y = df['TARGET'].copy()
349
350 pipe = make_pipeline(StandardScaler(),PCA(),MinMaxScaler())
351
352 pipe.fit(X)
353 Xs = pd.DataFrame(pipe.transform(X))
354
355 Xt,Xv,yt,yv = train_test_split(Xs,y,train_size=0.7)
356
357 param_grid = dict(activation = ['identity', 'logistic', 'tanh', 'relu'],
358 alpha = np.arange(0.0001,0.0005,0.0001),
359 learning_rate = ['constant', 'invscaling', 'adaptive'],
360 max_iter = [1000],
361 hidden_layer_sizes = [(a,b,c,) for a in range(20,35,5) for b in range(20,35,5) for c in range(20,35,5)]
362                  )
363 grid = RandomizedSearchCV(cv=3,
364                           verbose=True,
365                           scoring='roc_auc',
366                           estimator=model,
367                           n_iter=100,
368                           n_jobs=-1,
369                           param_distributions=param_grid)
370
371 grid.fit(Xs,y)
372
373 grid.best_estimator_
374
375 model = grid.best_estimator_
376 model.fit(Xt,yt)
377
378
379
380 # #### -Resultados-
381
382 # In[23]:
383
384 metricas(model,Xt,Xv,yt,yv)
385
386
387 # ### -Analisis Discriminante
388
389 # In[24]:
390
```

```
'''
Analisis discrimintante
'''
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.preprocessing import MinMaxScaler,StandardScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import make_pipeline
from sklearn.metrics import accuracy_score,roc_auc_score
from sklearn.model_selection import GridSearchCV,train_test_split,RandomizedSearchCV


# In[25]:

model = LinearDiscriminantAnalysis()

# df = pickle.load(open('Sample_MDL_ACT_con_WOE','rb'))


# In[26]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()

y = df['TARGET'].copy()

pipe = make_pipeline(StandardScaler(),PCA(),StandardScaler())
pipe.fit(X)
Xs = pd.DataFrame(pipe.transform(X))

Xt,Xv,yt,yv = train_test_split(Xs,y,train_size=0.7)

model.fit(Xt,yt)


# #### -Resultados-

# In[27]:

metricas(model,Xt,Xv,yt,yv)


# ### -Máquina Vector Soporte

# In[28]:

'''
Máquina Vector Soporte
'''
import pickle
from sklearn.svm import SVC
from sklearn.preprocessing import MinMaxScaler,StandardScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import make_pipeline
from sklearn.metrics import accuracy_score,roc_auc_score
from sklearn.model_selection import GridSearchCV,train_test_split,RandomizedSearchCV


# In[29]:

model = SVC(probability=True)

# df = pickle.load(open('Sample_MDL_ACT_con_WOE','rb'))


# In[30]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()
```

```

y = df['TARGET'].copy()

pipe = make_pipeline(MinMaxScaler())
pipe.fit(X)
Xs = pd.DataFrame(pipe.transform(X),columns=X.columns)

Xt,Xv,yt,yv = train_test_split(Xs,y,train_size=0.7)

param_grid = dict( C = np.arange(0.1,1,0.1),
                   kernel = ['linear', 'poly', 'rbf', 'sigmoid'],
                   degree = range(1,6),
                   probability = [True]
)

grid = GridSearchCV(cv=3,
                   verbose=True,
                   scoring='roc_auc',
                   estimator=model,
                   n_jobs=-1,
                   param_grid=param_grid)

grid.fit(Xs,y)

model = grid.best_estimator_
model.fit(Xt,yt)


# #### -Resultados-

# In[31]:

metricas(model,Xt,Xv,yt,yv)


# ### -K-Vecinos

# In[32]:

'''
K-Vecinos
'''
from sklearn.neighbors import KNeighborsClassifier


# In[33]:

model = KNeighborsClassifier()


# In[34]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()

y = df['TARGET'].copy()

pipe = make_pipeline(MinMaxScaler())
pipe.fit(X)
Xs = pd.DataFrame(pipe.transform(X),columns=X.columns)

Xt,Xv,yt,yv = train_test_split(Xs,y,train_size=0.7)

param_grid = dict(n_neighbors = range(4,90))

grid = GridSearchCV(cv=3,
                   verbose=True,
```

```
                    scoring='roc_auc',
                    estimator=model,
                    n_jobs=-1,
                    param_grid=param_grid)

grid.fit(Xs,y)

model = grid.best_estimator_
model.fit(Xt,yt)


# #### -Resultados-

# In[35]:

metricas(model,Xt,Xv,yt,yv)


# ### -Gradiente estocástico descendiente

# In[36]:

'''
Gradiente estocástico descendiente
'''
from sklearn.linear_model import SGDClassifier


# In[37]:

model = SGDClassifier(loss='log')


# In[38]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()

y = df['TARGET'].copy()

pipe = make_pipeline(MinMaxScaler())
pipe.fit(X)
Xs = pd.DataFrame(pipe.transform(X),columns=X.columns)

Xt,Xv,yt,yv = train_test_split(Xs,y,train_size=0.7)

param_grid = dict(loss = ['log'],
                    penalty = ['l1','l2','elasticnet'],
                    alpha = np.arange(0.0001,0.001,0.0001),
                    eta0 = np.arange(0.01,0.1,0.01),
                    learning_rate = ['constant','optimal','invscaling']
                    )
grid = GridSearchCV(cv=3,
                    verbose=True,
                    scoring='roc_auc',
                    estimator=model,
                    n_jobs=-1,
                    param_grid=param_grid)

grid.fit(Xs,y)

model = grid.best_estimator_
model.fit(Xt,yt)


# #### -Resultados-

```

```
# In[39]:

metricas(model,Xt,Xv,yt,yv)


# ### -Naive Bayes

# In[40]:

'''
Clasificador ingenuo de Bayes
'''
from sklearn.naive_bayes import GaussianNB


# In[41]:

model = GaussianNB()


# In[42]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()

y = df['TARGET'].copy()

pipe = make_pipeline(MinMaxScaler())
pipe.fit(X)
Xs = pd.DataFrame(pipe.transform(X),columns=X.columns)

Xt,Xv,yt,yv = train_test_split(Xs,y,train_size=0.7)

model.fit(Xt,yt)


# #### -Resultados-

# In[43]:

metricas(model,Xt,Xv,yt,yv)


# ### -Ensambles

# In[44]:

'''
Ensambles
'''
from sklearn.ensemble import AdaBoostClassifier,RandomForestClassifier


# In[45]:

modelos = [RandomForestClassifier(),AdaBoostClassifier()]


# In[46]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()

y = df['TARGET'].copy()
pipe = make_pipeline(MinMaxScaler())
pipe.fit(X)
Xs = pd.DataFrame(pipe.transform(X),columns=X.columns)

```

```
Xt,Xv,yt,yv = train_test_split(Xs,y,train_size=0.7)


param_grid = dict (n_estimators=range(10,250,10))

grid = RandomizedSearchCV(cv=3,
                          error_score='accuracy',
                          estimator=modelos[-1],
                          n_jobs=-1,
                          param_distributions=param_grid,verbose=True)

grid.fit(Xt,yt)

model = grid.best_estimator_
model.fit(Xt,yt)


# #### -Resultados-

# In[47]:

metricas(model,Xt,Xv,yt,yv)


# In[48]:

'''
Guardamos el modelo predictivo
'''
# save model to file
pickle.dump(model, open("modelito_ada1.pickle.dat", "wb"))


# In[49]:

'''
Cargamos el modelo predictivo
'''
# load model from file
loaded_model = pickle.load(open("modelito_ada1.pickle.dat", "rb"))


# In[50]:

metricas(loaded_model,Xt,Xv,yt,yv)


# In[51]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()

y = df['TARGET'].copy()

pipe = make_pipeline(MinMaxScaler())
pipe.fit(X)
Xs = pd.DataFrame(pipe.transform(X),columns=X.columns)

y_pred = loaded_model.predict(Xs)


# In[52]:

print "ROC:%.3f" %(roc_auc_score(y_true=y,y_score=loaded_model.predict_proba(Xs)[:,1]))
print "ACC:%.3f" %(accuracy_score(y_true=y,y_pred=loaded_model.predict(Xs)))

```

```
# In[ ]:

'''
Seleccionamos las mejores variables
'''
sk = SelectKBest(k=5)

sk.fit(Xs,y)

var_best= [x for x,y in zip(var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]],sk.get_sup

var_best


# In[53]:

'''
Modelación no supervisada, análisis de clustering
'''
var=var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]


# In[54]:

X = df[var_cont_f+[v for v in df.columns if v in ['W_'+x for x in var_disc_f]]].copy()
y = df['TARGET'].copy()


# In[55]:

'''
Estandarización de variables, escalamiento multidimencional y componentes principales
'''
sc = StandardScaler()
mm = MinMaxScaler()
mds= MDS(n_components=2)
pca = PCA(n_components=2)

sc.fit(X)
mm.fit(X)


Xs = pd.DataFrame(sc.transform(X),columns=X.columns)
Xmm = pd.DataFrame(mm.transform(X),columns=X.columns)


# In[56]:

'''
Gráfica de  las coponentes principales
'''
pca.fit(Xs)
print pca.explained_variance_ratio_.cumsum()
Xp = pd.DataFrame(pca.transform(Xs),columns=['p1','p2'])
Xmds = pd.DataFrame(mds.fit_transform(Xmm),columns=['d1','d2'])
sns.lmplot(data=Xp,x='p1',y='p2',fit_reg=False)


# In[57]:

'''
Gráfica de escalamiento multidimensional
'''
sns.lmplot(data=Xmds,x='d1',y='d2',fit_reg=False)


```

```
# In[58]:

'''
La Inercia
'''
lst_in = []
for k in range(2,10):
    km = KMeans(n_clusters=k)
    km.fit(Xmm)
    lst_in.append(km.inertia_)

plt.plot(range(2,10),lst_in,marker='o')



# In[59]:

'''
Modelos de Clustering
'''
agg = AgglomerativeClustering(n_clusters=4)
km = KMeans(n_clusters=4)
gmm = GaussianMixture(n_components=4)

km.fit(Xmm[var])
gmm.fit(Xmm[var])


# In[60]:

X.head()


# In[61]:

'''
Predicción con los diferentes modelos
'''
X['cl_agg'] =Xp['cl_agg'] =Xmds['cl_agg'] = Xmm['cl_agg']= Xs['cl_agg'] =agg.fit_predict(Xmm)

X['cl_km'] =Xp['cl_km'] =Xmds['cl_km'] = Xmm['cl_km']= Xs['cl_km'] =km.predict(Xmm[var])

X['cl_gmm'] =Xp['cl_gmm'] =Xmds['cl_gmm'] = Xmm['cl_gmm']= Xs['cl_gmm'] =gmm.predict(Xmm[var])

Xs[var+['cl_gmm']].groupby('cl_gmm').mean()

sk = SelectKBest(k=5)

sk.fit(Xmm[var],Xmm['cl_gmm'])

var_best= [x for x,y in zip(var,sk.get_support()) if y]

var_best


# In[62]:

'''
El mejor modelo es el gausseano y lo guardamos
'''
# save model to file
pickle.dump(gmm, open("modelito_gmm1.pickle.dat", "wb"))


# In[63]:

```

```
'''
Cargamos el modelo gausseano
'''
# load model from file
loaded_model_gmm = pickle.load(open("modelito_gmm1.pickle.dat", "rb"))


# In[64]:

'''
Predecimos con el modelo cargado
'''
y_gmm_predict=loaded_model_gmm.predict(Xmm[var])


# In[65]:

y_gmm_predict


# In[66]:

X['prueba_gmm'] =Xp['prueba_gmm'] =Xmds['prueba_gmm'] = Xmm['prueba_gmm']= Xs['prueba_gmm'] =loaded_model_gmm.

Xs[var+['prueba_gmm']].groupby('prueba_gmm').mean()


# In[67]:

Xs[var+['cl_gmm']].groupby('cl_gmm').mean()


# In[68]:

'''
Gráfica de los clusters con el modelo y componentes principales
'''
sns.lmplot(data=Xp,x='p1',y='p2',hue='prueba_gmm',fit_reg=False)


# In[69]:

'''
Gráfica de los clusters con el modelo y escalamiento multidimencional
'''
sns.lmplot(data=Xmds,x='d1',y='d2',hue='prueba_gmm',fit_reg=False)


# In[ ]:




# In[70]:

sns.lmplot(data=Xp,x='p1',y='p2',hue='cl_gmm',fit_reg=False)


# In[71]:

sns.lmplot(data=Xmds,x='d1',y='d2',hue='cl_gmm',fit_reg=False)


# In[72]:

'''
```

```
Comenzamos con el perfilamiento
'''
Xs[var+['cl_gmm']].groupby('cl_gmm').describe()


# In[73]:


df2 = df.copy()

df2.describe()


# In[160]:


gmm.fit(df2[var])
df2['cl_gmm']=gmm.predict(df2[var])


# In[165]:

aux = df2[['V_N_ACADEMICO','V_OCUPACION','V_TERRITORIO','V_ST_CIVIL','cl_gmm']].copy()

aux['n'] = 1
aux['contar'] = 1


piv = aux.pivot_table(index='contar',
                columns='cl_gmm',
                values='n',
                fill_value=0,
                aggfunc='count')


# In[166]:




# In[76]:


for i in range(4):
    piv[i]/=piv[i].sum()



piv.sort_values(3,ascending=0)


# In[77]:


piv = aux.pivot_table(index='V_TERRITORIO',
                columns='cl_gmm',
                values='n',
                fill_value=0,
                aggfunc='count')
for i in range(4):
    piv[i]/=piv[i].sum()
piv.sort_values(3,ascending=0)


# In[78]:

```

```
piv = aux.pivot_table(index='V_ST_CIVIL',
                columns='cl_gmm',
                values='n',
                fill_value=0,
                aggfunc='count')
for i in range(4):
    piv[i]/=piv[i].sum()


piv.sort_values(2,ascending=0)


# In[79]:


Xs[var+['cl_gmm']].groupby('cl_gmm').describe()

Xs[var+['cl_gmm']].groupby('cl_gmm').describe()

aux = df2[['TARGET','cl_gmm']].copy()

aux['n'] = 1

piv = aux.pivot_table(index='TARGET',
                columns='cl_gmm',
                values='n',
                fill_value=0,
                aggfunc='count')


# In[80]:

for i in range(4):
    piv[i]/=piv[i].sum()
piv.sort_values(3,ascending=0)


# In[159]:

piv


# In[157]:

'''
Distribución de los grupos respecto a la variable objetivo.
'''
num = np.arange(4)
plt.bar(num + 0.25, piv.T[0].tolist(), color = "r", width = 0.25)
plt.bar(num + 0.5, piv.T[1].tolist(), color = "b", width = 0.25)
plt.xticks(num+0.38, ["G0","G1","G2","G3"])
plt.legend(labels=['NO activada','Activada'])
plt.title('Indice de activacion')
plt.show();


# In[81]:

Xmm[['cl_gmm']+var].groupby('cl_gmm').mean()
```

## 10.2. Código en SAS

Listing 2: Análisis de Conglomerados en SAS

```
/*IMPORTAR LA TABLA*/
```

```
data AAOK.SAMPLE_MDL_ACT;
SET SAMPLE_MDL_ACT;
RUN;

/* CREAMOS UNA COPIA EN EL WORK*/

DATA SAMPLE_MDL_ACT;
SET AAOK.SAMPLE_MDL_ACT;
RUN;

/* FASTCLUS, PARA EXCLUIR OUTLIERS */

PROC FASTCLUS DATA=SAMPLE_MDL_ACT MAXCLUSTERS=8 OUT=XF;
VAR V_:;
RUN;

/*ELIMINAMOS LOS CLUSTERS CON POCAS OBSERVACONES,
ES DECIR, LOS OUTLIERS Y CREAMOS UNA NUEVA TABLA */

DATA XF_;
SET XF;
WHERE CLUSTER NOT IN (3, 4, 6, 8);
DROP CLUSTER DISTANCE;
RUN;

/*P1. ANÁLISIS EXPLORATORIO*/

PROC MEANS DATA=XF_ NMISS N;
VAR V_:;
RUN;

/* HISTOGRAMA CON VALORES EN BCSCORE POSITIVO*/

PROC UNIVARIATE DATA=XF_;
VAR V_BCSCORE;WHERE V_BCSCORE > 0;HIST;
RUN;

/*EN EXCEL FILTRAMOS LAS VARIABLES QUE TIENE MAS DE 30% DE MISSINGS
Y LAS DROPEAMOS*/

data XF_1;
SET XF_;
DROP V_RT_OO_R_LC_SUM_OO_R_CTAS_AB_3
V_RT_TO_R_LC_SUM_TO_CTAS_AB_3
V_RT_TO_R_SALDO_TO_SALDO_3
V_RT_TO_R_SALDO_TO_CTAS_AB_3
V_RT_OO_R_SALDO_OO_R_LC_SUM_3
V_MEDIA_ATM_3
V_MEDIA_B_D_OO_3
V_MEDIA_B_D_TO_3
V_MEDIA_OO_R_CTAS_AB_3
V_MEDIA_OO_R_LC_MAX_3
V_MEDIA_OO_R_LC_SUM_3
V_MEDIA_OO_R_SALDO_3
V_MEDIA_TO_CTAS_AB_3
V_MEDIA_TO_R_LC_MAX_3
V_MEDIA_TO_R_LC_SUM_3
V_MEDIA_TO_R_SALDO_3
V_MEDIA_TO_SALDO_3
```

V_RT_OO_R_LC_SUM_OO_R_CTAS_AB_4
V_RT_TO_R_LC_SUM_TO_CTAS_AB_4
V_RT_TO_R_SALDO_TO_SALDO_4
V_RT_TO_R_SALDO_TO_CTAS_AB_4
V_RT_OO_R_SALDO_OO_R_LC_SUM_4
V_MEDIA_ATM_4
V_MEDIA_B_D_OO_4
V_MEDIA_B_D_TO_4
V_MEDIA_OO_R_CTAS_AB_4
V_MEDIA_OO_R_LC_MAX_4
V_MEDIA_OO_R_LC_SUM_4
V_MEDIA_OO_R_SALDO_4
V_MEDIA_TO_CTAS_AB_4
V_MEDIA_TO_R_LC_MAX_4
V_MEDIA_TO_R_LC_SUM_4
V_MEDIA_TO_R_SALDO_4
V_MEDIA_TO_SALDO_4
V_RT_OO_R_LC_SUM_OO_R_CTAS_AB_5
V_RT_TO_R_LC_SUM_TO_CTAS_AB_5
V_RT_TO_R_SALDO_TO_SALDO_5
V_RT_TO_R_SALDO_TO_CTAS_AB_5
V_RT_OO_R_SALDO_OO_R_LC_SUM_5
V_MEDIA_ATM_5
V_MEDIA_B_D_OO_5
V_MEDIA_B_D_TO_5
V_MEDIA_OO_R_CTAS_AB_5
V_MEDIA_OO_R_LC_MAX_5
V_MEDIA_OO_R_LC_SUM_5
V_MEDIA_OO_R_SALDO_5
V_MEDIA_TO_CTAS_AB_5
V_MEDIA_TO_R_LC_MAX_5
V_MEDIA_TO_R_LC_SUM_5
V_MEDIA_TO_R_SALDO_5
V_MEDIA_TO_SALDO_5
V_RT_OO_R_LC_SUM_OO_R_CTAS_AB_6
V_RT_TO_R_LC_SUM_TO_CTAS_AB_6
V_RT_TO_R_SALDO_TO_SALDO_6
V_RT_TO_R_SALDO_TO_CTAS_AB_6
V_RT_OO_R_SALDO_OO_R_LC_SUM_6
V_MEDIA_ATM_6
V_MEDIA_B_D_OO_6
V_MEDIA_B_D_TO_6
V_MEDIA_OO_R_CTAS_AB_6
V_MEDIA_OO_R_LC_MAX_6
V_MEDIA_OO_R_LC_SUM_6
V_MEDIA_OO_R_SALDO_6
V_MEDIA_TO_CTAS_AB_6
V_MEDIA_TO_R_LC_MAX_6
V_MEDIA_TO_R_LC_SUM_6
V_MEDIA_TO_R_SALDO_6
V_MEDIA_TO_SALDO_6
V_MEDIAN_ATM_7
V_MEDIAN_B_D_OO_7
V_MEDIAN_B_D_TO_7
V_MEDIAN_OO_R_CTAS_AB_7
V_MEDIAN_OO_R_LC_MAX_7
V_MEDIAN_OO_R_LC_SUM_7
V_MEDIAN_OO_R_SALDO_7
V_MEDIAN_TO_CTAS_AB_7

```
V_MEDIAN_TO_R_LC_MAX_7
V_MEDIAN_TO_R_LC_SUM_7
V_MEDIAN_TO_R_SALDO_7
V_MEDIAN_TO_SALDO_7
V_RAT_LC_MO_AVG_1
V_RAT_LC_MO_AVG_2
V_RAT_LC_MO_AVG_3
V_RAT_LC_MO_AVG_4
V_RAT_LC_MO_AVG_5
V_RAT_LC_MO_AVG_6
V_RAT_LC_MO_MAX_1
V_RAT_LC_MO_MAX_2
V_RAT_LC_MO_MAX_3
V_RAT_LC_MO_MAX_4
V_RAT_LC_MO_MAX_5
V_RAT_LC_MO_MAX_6
V_SDO_DEBITO;
RUN;

/*P2. ANÁLISIS FACTORIAL*/

PROC FACTOR DATA=XF_1 METHOD=PRINCIPAL PRIORS=ONE NORM=KAISER ROTATE=VARIMAX;
VAR V_:;
RUN;

DATA AAOK.XF_1;
SET XF_1;
RUN;

/*P3. CLUSTERING DE VARIABLES*/

PROC VARCLUS DATA= XF_1 PROPORTION=.9 SHORT SUMMARY;
VAR V_:;
RUN;

PROC VARCLUS DATA= XF_1 PROPORTION=.9 SHORT MAXCLUSTERS=62;
VAR V_:;
RUN;

/*IDENTIFICANDO LAS VARIABLES REPRESENTANTES*/

PROC SQL;
CREATE TABLE CUBO AS
SELECT CLUSTER, MIN(RATIO) AS RATIO
FROM MDL_CO
GROUP BY CLUSTER
;QUIT;

DATA AAOK.MDL_CLUST;
SET XF_1;
KEEP V_VM_OO_R_CTAS_AB0_5
V_VM_OO_R_CTAS_AB3_5
V_OO_R_BANCO_LC_MAX
V_TOT_CANALES_ACTI
V_NDEC_OO_R_LC_MAX_5
V_CANAL
V_EDAD_CLIENTE
V_CLASIF_CTE
V_WELCOME_CALL
```

```
V_NDEC_ATM_5
V_OCUPACION
V_NINC_OO_R_LC_SUM_4
V_N_ACADEMICO
V_TOT_CRED_BCO
V_ST_CIVIL
V_SEXO
V_BCSCORE
V_NDEC_TO_CTAS_AB_5
V_MOB_TDC
V_CELULAR
V_ANT_INT_ANTIG
V_ENTREGAT
V_VM_TO_R_LC_SUM5000_4
V_MO_R__SALDO
V_MO_R__CTAS_AB
V_VM_OO_R_LC_MAX25000_4
V_VM_TO_CTAS_AB2_5
V_NDEC_TO_R_LC_MAX_5
V_NINC_ATM_5
V_NINC_B_D_TO_5
V_VM_TO_R_SALDO1000_5
V_VM_CH_P0_4
V_NDEC_TO_R_SALDO_4
V_VM_ATM500_5
V_NINC_CH_P_3
V_VM_TO_R_LC_MAX25000_4
V_NDEC_OO_R_CTAS_AB_5
V_NINC_B_D_OO_4
V_NINC_OO_R_SALDO_5
V_NDEC_TO_R_LC_SUM_3
V_OO_R_BANCO_SALDO
V_TERRITORIO
V_SEGMENTO
V_VM_OO_R_SALDO25000_4
V_VM_CH_P100_4
V_ATM
V_NINC_TO_SALDO_5
V_VM_B_D_TO45_5
V_NINC_TO_R_LC_SUM_4
V_OO_R_BANCO_CTAS_AB
V_VM_OO_R_CTAS_AB1_4
V_NDEC_B_D_OO_4
V_VM_TO_R_SALDO25000_4
V_NINC_CH_P_6
V_VM_TO_SALDO10000_5
V_MEDIA_CH_P_5
V_NDEC_OO_R_LC_SUM_3
V_NDEC_ATM_3
V_NDEC_TO_SALDO_5
V_NDEC_CH_P_6
V_VM_CH_P1500_4
V_MO_R__LC_MAX
ID
TARGET;
RUN;

DATA X;
SET AAOK.MDL_CLUST;
```

```
*-------------------------------------------------------------*;
* EM SCORE CODE;
* EM Version: 13.2;
* SAS Release: 9.04.01M2P072314;
* Host: uxe25102;
* Encoding: latin1;
* Locale: en_US;
* Project Path: /herramientas/SAS/OKY;
* Project Name: SEG_OC;
* Diagram Id: EMWS1;
* Diagram Name: DIAG_OC;
* Generated by: A3725988;
* Date: 01APR2018:23:26:11;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Input Data Source;
* TYPE: SAMPLE;
* NODE: Ids;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Clustering;
* TYPE: EXPLORE;
* NODE: Clus;
*-------------------------------------------------------------*;
****************************************;
*** Begin Scoring Code from PROC DMVQ ***;
****************************************;


*** Begin Class Look-up, Standardization, Replacement ;
drop _dm_bad; _dm_bad = 0;

*** Standardize V_ANT_INT_ANTIG ;
drop T_V_ANT_INT_ANTIG ;
if missing( V_ANT_INT_ANTIG ) then T_V_ANT_INT_ANTIG = .;
else T_V_ANT_INT_ANTIG = (V_ANT_INT_ANTIG - 0) * 0.00327868852459;

*** Standardize V_BCSCORE ;
drop T_V_BCSCORE ;
if missing( V_BCSCORE ) then T_V_BCSCORE = .;
else T_V_BCSCORE = (V_BCSCORE - -8) * 0.00125470514429;

*** Standardize V_EDAD_CLIENTE ;
drop T_V_EDAD_CLIENTE ;
if missing( V_EDAD_CLIENTE ) then T_V_EDAD_CLIENTE = .;
else T_V_EDAD_CLIENTE = (V_EDAD_CLIENTE - 18) * 0.00124843945068;

*** Standardize V_MEDIA_CH_P_5 ;
drop T_V_MEDIA_CH_P_5 ;
if missing( V_MEDIA_CH_P_5 ) then T_V_MEDIA_CH_P_5 = .;
else T_V_MEDIA_CH_P_5 = (V_MEDIA_CH_P_5 - 0) * 5.3990347253701E-7;

*** Standardize V_MOB_TDC ;
drop T_V_MOB_TDC ;
if missing( V_MOB_TDC ) then T_V_MOB_TDC = .;
else T_V_MOB_TDC = (V_MOB_TDC - 3) * 0.02631578947368;

*** Standardize V_MO_R__CTAS_AB ;
drop T_V_MO_R__CTAS_AB ;
```

```
if missing( V_MO_R__CTAS_AB ) then T_V_MO_R__CTAS_AB = .;
else T_V_MO_R__CTAS_AB = (V_MO_R__CTAS_AB - 0) * 0.16666666666666;

*** Standardize V_MO_R__LC_MAX ;
drop T_V_MO_R__LC_MAX ;
if missing( V_MO_R__LC_MAX ) then T_V_MO_R__LC_MAX = .;
else T_V_MO_R__LC_MAX = (V_MO_R__LC_MAX - 0) * 1.547987616099E-6;

*** Standardize V_MO_R__SALDO ;
drop T_V_MO_R__SALDO ;
if missing( V_MO_R__SALDO ) then T_V_MO_R__SALDO = .;
else T_V_MO_R__SALDO = (V_MO_R__SALDO - 0) * 2.1531306519679E-6;

*** Standardize V_NDEC_ATM_3 ;
drop T_V_NDEC_ATM_3 ;
if missing( V_NDEC_ATM_3 ) then T_V_NDEC_ATM_3 = .;
else T_V_NDEC_ATM_3 = (V_NDEC_ATM_3 - 0) * 0.5;

*** Standardize V_NDEC_ATM_5 ;
drop T_V_NDEC_ATM_5 ;
if missing( V_NDEC_ATM_5 ) then T_V_NDEC_ATM_5 = .;
else T_V_NDEC_ATM_5 = (V_NDEC_ATM_5 - 0) * 0.25;

*** Standardize V_NDEC_B_D_OO_4 ;
drop T_V_NDEC_B_D_OO_4 ;
if missing( V_NDEC_B_D_OO_4 ) then T_V_NDEC_B_D_OO_4 = .;
else T_V_NDEC_B_D_OO_4 = (V_NDEC_B_D_OO_4 - 0) * 0.5;

*** Standardize V_NDEC_CH_P_6 ;
drop T_V_NDEC_CH_P_6 ;
if missing( V_NDEC_CH_P_6 ) then T_V_NDEC_CH_P_6 = .;
else T_V_NDEC_CH_P_6 = (V_NDEC_CH_P_6 - 0) * 0.2;

*** Standardize V_NDEC_OO_R_CTAS_AB_5 ;
drop T_V_NDEC_OO_R_CTAS_AB_5 ;
if missing( V_NDEC_OO_R_CTAS_AB_5 ) then T_V_NDEC_OO_R_CTAS_AB_5 = .;
else T_V_NDEC_OO_R_CTAS_AB_5 = (V_NDEC_OO_R_CTAS_AB_5 - 0) * 0.5;

*** Standardize V_NDEC_OO_R_LC_MAX_5 ;
drop T_V_NDEC_OO_R_LC_MAX_5 ;
if missing( V_NDEC_OO_R_LC_MAX_5 ) then T_V_NDEC_OO_R_LC_MAX_5 = .;
else T_V_NDEC_OO_R_LC_MAX_5 = (V_NDEC_OO_R_LC_MAX_5 - 0) * 0.5;

*** Standardize V_NDEC_OO_R_LC_SUM_3 ;
drop T_V_NDEC_OO_R_LC_SUM_3 ;
if missing( V_NDEC_OO_R_LC_SUM_3 ) then T_V_NDEC_OO_R_LC_SUM_3 = .;
else T_V_NDEC_OO_R_LC_SUM_3 = (V_NDEC_OO_R_LC_SUM_3 - 0) * 1;

*** Standardize V_NDEC_TO_CTAS_AB_5 ;
drop T_V_NDEC_TO_CTAS_AB_5 ;
if missing( V_NDEC_TO_CTAS_AB_5 ) then T_V_NDEC_TO_CTAS_AB_5 = .;
else T_V_NDEC_TO_CTAS_AB_5 = (V_NDEC_TO_CTAS_AB_5 - 0) * 0.5;

*** Standardize V_NDEC_TO_R_LC_MAX_5 ;
drop T_V_NDEC_TO_R_LC_MAX_5 ;
if missing( V_NDEC_TO_R_LC_MAX_5 ) then T_V_NDEC_TO_R_LC_MAX_5 = .;
else T_V_NDEC_TO_R_LC_MAX_5 = (V_NDEC_TO_R_LC_MAX_5 - 0) * 0.5;

*** Standardize V_NDEC_TO_R_LC_SUM_3 ;
```

```
drop T_V_NDEC_TO_R_LC_SUM_3 ;
if missing( V_NDEC_TO_R_LC_SUM_3 ) then T_V_NDEC_TO_R_LC_SUM_3 = .;
else T_V_NDEC_TO_R_LC_SUM_3 = (V_NDEC_TO_R_LC_SUM_3 - 0) * 1;

*** Standardize V_NDEC_TO_R_SALDO_4 ;
drop T_V_NDEC_TO_R_SALDO_4 ;
if missing( V_NDEC_TO_R_SALDO_4 ) then T_V_NDEC_TO_R_SALDO_4 = .;
else T_V_NDEC_TO_R_SALDO_4 = (V_NDEC_TO_R_SALDO_4 - 0) * 0.5;

*** Standardize V_NDEC_TO_SALDO_5 ;
drop T_V_NDEC_TO_SALDO_5 ;
if missing( V_NDEC_TO_SALDO_5 ) then T_V_NDEC_TO_SALDO_5 = .;
else T_V_NDEC_TO_SALDO_5 = (V_NDEC_TO_SALDO_5 - 0) * 0.5;

*** Standardize V_NINC_ATM_5 ;
drop T_V_NINC_ATM_5 ;
if missing( V_NINC_ATM_5 ) then T_V_NINC_ATM_5 = .;
else T_V_NINC_ATM_5 = (V_NINC_ATM_5 - 0) * 0.25;

*** Standardize V_NINC_B_D_OO_4 ;
drop T_V_NINC_B_D_OO_4 ;
if missing( V_NINC_B_D_OO_4 ) then T_V_NINC_B_D_OO_4 = .;
else T_V_NINC_B_D_OO_4 = (V_NINC_B_D_OO_4 - 0) * 0.5;

*** Standardize V_NINC_B_D_TO_5 ;
drop T_V_NINC_B_D_TO_5 ;
if missing( V_NINC_B_D_TO_5 ) then T_V_NINC_B_D_TO_5 = .;
else T_V_NINC_B_D_TO_5 = (V_NINC_B_D_TO_5 - 0) * 0.5;

*** Standardize V_NINC_CH_P_3 ;
drop T_V_NINC_CH_P_3 ;
if missing( V_NINC_CH_P_3 ) then T_V_NINC_CH_P_3 = .;
else T_V_NINC_CH_P_3 = (V_NINC_CH_P_3 - 0) * 0.5;

*** Standardize V_NINC_CH_P_6 ;
drop T_V_NINC_CH_P_6 ;
if missing( V_NINC_CH_P_6 ) then T_V_NINC_CH_P_6 = .;
else T_V_NINC_CH_P_6 = (V_NINC_CH_P_6 - 0) * 0.2;

*** Standardize V_NINC_OO_R_LC_SUM_4 ;
drop T_V_NINC_OO_R_LC_SUM_4 ;
if missing( V_NINC_OO_R_LC_SUM_4 ) then T_V_NINC_OO_R_LC_SUM_4 = .;
else T_V_NINC_OO_R_LC_SUM_4 = (V_NINC_OO_R_LC_SUM_4 - 0) * 0.5;

*** Standardize V_NINC_OO_R_SALDO_5 ;
drop T_V_NINC_OO_R_SALDO_5 ;
if missing( V_NINC_OO_R_SALDO_5 ) then T_V_NINC_OO_R_SALDO_5 = .;
else T_V_NINC_OO_R_SALDO_5 = (V_NINC_OO_R_SALDO_5 - 0) * 0.5;

*** Standardize V_NINC_TO_R_LC_SUM_4 ;
drop T_V_NINC_TO_R_LC_SUM_4 ;
if missing( V_NINC_TO_R_LC_SUM_4 ) then T_V_NINC_TO_R_LC_SUM_4 = .;
else T_V_NINC_TO_R_LC_SUM_4 = (V_NINC_TO_R_LC_SUM_4 - 0) * 0.5;

*** Standardize V_NINC_TO_SALDO_5 ;
drop T_V_NINC_TO_SALDO_5 ;
if missing( V_NINC_TO_SALDO_5 ) then T_V_NINC_TO_SALDO_5 = .;
else T_V_NINC_TO_SALDO_5 = (V_NINC_TO_SALDO_5 - 0) * 0.5;
```

```
*** Standardize V_OO_R_BANCO_CTAS_AB ;
drop T_V_OO_R_BANCO_CTAS_AB ;
if missing( V_OO_R_BANCO_CTAS_AB ) then T_V_OO_R_BANCO_CTAS_AB = .;
else T_V_OO_R_BANCO_CTAS_AB = (V_OO_R_BANCO_CTAS_AB - 0) * 0.07142857142857;

*** Standardize V_OO_R_BANCO_LC_MAX ;
drop T_V_OO_R_BANCO_LC_MAX ;
if missing( V_OO_R_BANCO_LC_MAX ) then T_V_OO_R_BANCO_LC_MAX = .;
else T_V_OO_R_BANCO_LC_MAX = (V_OO_R_BANCO_LC_MAX - 0) * 1.5384615384615E-6;

*** Standardize V_OO_R_BANCO_SALDO ;
drop T_V_OO_R_BANCO_SALDO ;
if missing( V_OO_R_BANCO_SALDO ) then T_V_OO_R_BANCO_SALDO = .;
else T_V_OO_R_BANCO_SALDO = (V_OO_R_BANCO_SALDO - 0) * 1.7543028663554E-6;

*** Standardize V_TOT_CRED_BCO ;
drop T_V_TOT_CRED_BCO ;
if missing( V_TOT_CRED_BCO ) then T_V_TOT_CRED_BCO = .;
else T_V_TOT_CRED_BCO = (V_TOT_CRED_BCO - 0) * 0.11111111111111;

*** Standardize V_VM_ATM500_5 ;
drop T_V_VM_ATM500_5 ;
if missing( V_VM_ATM500_5 ) then T_V_VM_ATM500_5 = .;
else T_V_VM_ATM500_5 = (V_VM_ATM500_5 - 0) * 0.2;

*** Standardize V_VM_B_D_TO45_5 ;
drop T_V_VM_B_D_TO45_5 ;
if missing( V_VM_B_D_TO45_5 ) then T_V_VM_B_D_TO45_5 = .;
else T_V_VM_B_D_TO45_5 = (V_VM_B_D_TO45_5 - 0) * 0.2;

*** Standardize V_VM_CH_P0_4 ;
drop T_V_VM_CH_P0_4 ;
if missing( V_VM_CH_P0_4 ) then T_V_VM_CH_P0_4 = .;
else T_V_VM_CH_P0_4 = (V_VM_CH_P0_4 - 0) * 0.25;

*** Standardize V_VM_CH_P100_4 ;
drop T_V_VM_CH_P100_4 ;
if missing( V_VM_CH_P100_4 ) then T_V_VM_CH_P100_4 = .;
else T_V_VM_CH_P100_4 = (V_VM_CH_P100_4 - 0) * 0.25;

*** Standardize V_VM_CH_P1500_4 ;
drop T_V_VM_CH_P1500_4 ;
if missing( V_VM_CH_P1500_4 ) then T_V_VM_CH_P1500_4 = .;
else T_V_VM_CH_P1500_4 = (V_VM_CH_P1500_4 - 0) * 0.25;

*** Standardize V_VM_OO_R_CTAS_AB0_5 ;
drop T_V_VM_OO_R_CTAS_AB0_5 ;
if missing( V_VM_OO_R_CTAS_AB0_5 ) then T_V_VM_OO_R_CTAS_AB0_5 = .;
else T_V_VM_OO_R_CTAS_AB0_5 = (V_VM_OO_R_CTAS_AB0_5 - 0) * 0.2;

*** Standardize V_VM_OO_R_CTAS_AB1_4 ;
drop T_V_VM_OO_R_CTAS_AB1_4 ;
if missing( V_VM_OO_R_CTAS_AB1_4 ) then T_V_VM_OO_R_CTAS_AB1_4 = .;
else T_V_VM_OO_R_CTAS_AB1_4 = (V_VM_OO_R_CTAS_AB1_4 - 0) * 0.25;

*** Standardize V_VM_OO_R_CTAS_AB3_5 ;
drop T_V_VM_OO_R_CTAS_AB3_5 ;
if missing( V_VM_OO_R_CTAS_AB3_5 ) then T_V_VM_OO_R_CTAS_AB3_5 = .;
else T_V_VM_OO_R_CTAS_AB3_5 = (V_VM_OO_R_CTAS_AB3_5 - 0) * 0.2;
```

```
*** Standardize V_VM_OO_R_LC_MAX25000_4 ;
drop T_V_VM_OO_R_LC_MAX25000_4 ;
if missing( V_VM_OO_R_LC_MAX25000_4 ) then T_V_VM_OO_R_LC_MAX25000_4 = .;
else T_V_VM_OO_R_LC_MAX25000_4 = (V_VM_OO_R_LC_MAX25000_4 - 0) * 0.25;

*** Standardize V_VM_OO_R_SALDO25000_4 ;
drop T_V_VM_OO_R_SALDO25000_4 ;
if missing( V_VM_OO_R_SALDO25000_4 ) then T_V_VM_OO_R_SALDO25000_4 = .;
else T_V_VM_OO_R_SALDO25000_4 = (V_VM_OO_R_SALDO25000_4 - 0) * 0.25;

*** Standardize V_VM_TO_CTAS_AB2_5 ;
drop T_V_VM_TO_CTAS_AB2_5 ;
if missing( V_VM_TO_CTAS_AB2_5 ) then T_V_VM_TO_CTAS_AB2_5 = .;
else T_V_VM_TO_CTAS_AB2_5 = (V_VM_TO_CTAS_AB2_5 - 0) * 0.2;

*** Standardize V_VM_TO_R_LC_MAX25000_4 ;
drop T_V_VM_TO_R_LC_MAX25000_4 ;
if missing( V_VM_TO_R_LC_MAX25000_4 ) then T_V_VM_TO_R_LC_MAX25000_4 = .;
else T_V_VM_TO_R_LC_MAX25000_4 = (V_VM_TO_R_LC_MAX25000_4 - 0) * 0.25;

*** Standardize V_VM_TO_R_LC_SUM5000_4 ;
drop T_V_VM_TO_R_LC_SUM5000_4 ;
if missing( V_VM_TO_R_LC_SUM5000_4 ) then T_V_VM_TO_R_LC_SUM5000_4 = .;
else T_V_VM_TO_R_LC_SUM5000_4 = (V_VM_TO_R_LC_SUM5000_4 - 0) * 0.25;

*** Standardize V_VM_TO_R_SALDO1000_5 ;
drop T_V_VM_TO_R_SALDO1000_5 ;
if missing( V_VM_TO_R_SALDO1000_5 ) then T_V_VM_TO_R_SALDO1000_5 = .;
else T_V_VM_TO_R_SALDO1000_5 = (V_VM_TO_R_SALDO1000_5 - 0) * 0.2;

*** Standardize V_VM_TO_R_SALDO25000_4 ;
drop T_V_VM_TO_R_SALDO25000_4 ;
if missing( V_VM_TO_R_SALDO25000_4 ) then T_V_VM_TO_R_SALDO25000_4 = .;
else T_V_VM_TO_R_SALDO25000_4 = (V_VM_TO_R_SALDO25000_4 - 0) * 0.25;

*** Standardize V_VM_TO_SALDO10000_5 ;
drop T_V_VM_TO_SALDO10000_5 ;
if missing( V_VM_TO_SALDO10000_5 ) then T_V_VM_TO_SALDO10000_5 = .;
else T_V_VM_TO_SALDO10000_5 = (V_VM_TO_SALDO10000_5 - 0) * 0.2;

*** Generate dummy variables for V_ATM ;
drop V_ATM0 V_ATM1 ;
if missing( V_ATM ) then do;
   V_ATM0 = .;
   V_ATM1 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_ATM , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '1'  then do;
      V_ATM0 = 0;
      V_ATM1 = 1;
   end;
   else if _dm12 = '0'  then do;
      V_ATM0 = 1;
      V_ATM1 = 0;
   end;
```

```
   else do;
      V_ATM0 = .;
      V_ATM1 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_CANAL ;
drop V_CANAL1 V_CANAL2 V_CANAL3 V_CANAL5 V_CANAL6 ;
*** encoding is sparse, initialize to zero;
V_CANAL1 = 0;
V_CANAL2 = 0;
V_CANAL3 = 0;
V_CANAL5 = 0;
V_CANAL6 = 0;
if missing( V_CANAL ) then do;
   V_CANAL1 = .;
   V_CANAL2 = .;
   V_CANAL3 = .;
   V_CANAL5 = .;
   V_CANAL6 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_CANAL , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '6'  then do;
      V_CANAL6 = 1;
   end;
   else if _dm12 = '1'  then do;
      V_CANAL1 = 1;
   end;
   else if _dm12 = '2'  then do;
      V_CANAL2 = 1;
   end;
   else if _dm12 = '5'  then do;
      V_CANAL5 = 1;
   end;
   else if _dm12 = '3'  then do;
      V_CANAL3 = 1;
   end;
   else do;
      V_CANAL1 = .;
      V_CANAL2 = .;
      V_CANAL3 = .;
      V_CANAL5 = .;
      V_CANAL6 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_CELULAR ;
drop V_CELULAR0 V_CELULAR1 ;
if missing( V_CELULAR ) then do;
   V_CELULAR0 = .;
   V_CELULAR1 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
```

```
   _dm12 = put( V_CELULAR , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '1' then do;
      V_CELULAR0 = 0;
      V_CELULAR1 = 1;
   end;
   else if _dm12 = '0' then do;
      V_CELULAR0 = 1;
      V_CELULAR1 = 0;
   end;
   else do;
      V_CELULAR0 = .;
      V_CELULAR1 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_CLASIF_CTE ;
drop V_CLASIF_CTE1 V_CLASIF_CTE2 V_CLASIF_CTE3 ;
*** encoding is sparse, initialize to zero;
V_CLASIF_CTE1 = 0;
V_CLASIF_CTE2 = 0;
V_CLASIF_CTE3 = 0;
if missing( V_CLASIF_CTE ) then do;
   V_CLASIF_CTE1 = .;
   V_CLASIF_CTE2 = .;
   V_CLASIF_CTE3 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_CLASIF_CTE , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '2' then do;
      V_CLASIF_CTE2 = 1;
   end;
   else if _dm12 = '1' then do;
      V_CLASIF_CTE1 = 1;
   end;
   else if _dm12 = '3' then do;
      V_CLASIF_CTE3 = 1;
   end;
   else do;
      V_CLASIF_CTE1 = .;
      V_CLASIF_CTE2 = .;
      V_CLASIF_CTE3 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_ENTREGAT ;
drop V_ENTREGAT0 V_ENTREGAT1 ;
if missing( V_ENTREGAT ) then do;
   V_ENTREGAT0 = .;
   V_ENTREGAT1 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_ENTREGAT , BEST12. );
   %DMNORMIP( _dm12 )
```

```
   if _dm12 = '0'  then do;
      V_ENTREGAT0 = 1;
      V_ENTREGAT1 = 0;
   end;
   else if _dm12 = '1'  then do;
      V_ENTREGAT0 = 0;
      V_ENTREGAT1 = 1;
   end;
   else do;
      V_ENTREGAT0 = .;
      V_ENTREGAT1 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_N_ACADEMICO ;
drop V_N_ACADEMICO1 V_N_ACADEMICO2 V_N_ACADEMICO3 V_N_ACADEMICO4
        V_N_ACADEMICO5 V_N_ACADEMICO6 V_N_ACADEMICO7 V_N_ACADEMICO8
        V_N_ACADEMICO9 ;
*** encoding is sparse, initialize to zero;
V_N_ACADEMICO1 = 0;
V_N_ACADEMICO2 = 0;
V_N_ACADEMICO3 = 0;
V_N_ACADEMICO4 = 0;
V_N_ACADEMICO5 = 0;
V_N_ACADEMICO6 = 0;
V_N_ACADEMICO7 = 0;
V_N_ACADEMICO8 = 0;
V_N_ACADEMICO9 = 0;
if missing( V_N_ACADEMICO ) then do;
   V_N_ACADEMICO1 = .;
   V_N_ACADEMICO2 = .;
   V_N_ACADEMICO3 = .;
   V_N_ACADEMICO4 = .;
   V_N_ACADEMICO5 = .;
   V_N_ACADEMICO6 = .;
   V_N_ACADEMICO7 = .;
   V_N_ACADEMICO8 = .;
   V_N_ACADEMICO9 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_N_ACADEMICO , BEST12. );
   %DMNORMIP( _dm12 )
   _dm_find = 0; drop _dm_find;
   if _dm12 <= '5'  then do;
      if _dm12 <= '3'  then do;
         if _dm12 <= '2'  then do;
            if _dm12 = '1'  then do;
               V_N_ACADEMICO1 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '2'  then do;
                  V_N_ACADEMICO2 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
```

```
         else do;
            if _dm12 = '3'  then do;
               V_N_ACADEMICO3 = 1;
               _dm_find = 1;
            end;
         end;
      end;
      else do;
         if _dm12 = '4'  then do;
            V_N_ACADEMICO4 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '5'  then do;
               V_N_ACADEMICO5 = 1;
               _dm_find = 1;
            end;
         end;
      end;
   end;
   else do;
      if _dm12 <= '7'  then do;
         if _dm12 = '6'  then do;
            V_N_ACADEMICO6 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '7'  then do;
               V_N_ACADEMICO7 = 1;
               _dm_find = 1;
            end;
         end;
      end;
      else do;
         if _dm12 = '8'  then do;
            V_N_ACADEMICO8 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '9'  then do;
               V_N_ACADEMICO9 = 1;
               _dm_find = 1;
            end;
         end;
      end;
   end;
   if not _dm_find then do;
      V_N_ACADEMICO1 = .;
      V_N_ACADEMICO2 = .;
      V_N_ACADEMICO3 = .;
      V_N_ACADEMICO4 = .;
      V_N_ACADEMICO5 = .;
      V_N_ACADEMICO6 = .;
      V_N_ACADEMICO7 = .;
      V_N_ACADEMICO8 = .;
      V_N_ACADEMICO9 = .;
      _DM_BAD = 1;
   end;
end;
```

```
*** Generate dummy variables for V_OCUPACION ;
drop V_OCUPACION1 V_OCUPACION2 V_OCUPACION3 V_OCUPACION5 V_OCUPACION6
        V_OCUPACION7 V_OCUPACION8 V_OCUPACION9 V_OCUPACION10 V_OCUPACION11
        V_OCUPACION12 V_OCUPACION13 V_OCUPACION14 V_OCUPACION16 V_OCUPACION17
        V_OCUPACION18 V_OCUPACION19 V_OCUPACION20 V_OCUPACION21 ;
*** encoding is sparse, initialize to zero;
V_OCUPACION1 = 0;
V_OCUPACION2 = 0;
V_OCUPACION3 = 0;
V_OCUPACION5 = 0;
V_OCUPACION6 = 0;
V_OCUPACION7 = 0;
V_OCUPACION8 = 0;
V_OCUPACION9 = 0;
V_OCUPACION10 = 0;
V_OCUPACION11 = 0;
V_OCUPACION12 = 0;
V_OCUPACION13 = 0;
V_OCUPACION14 = 0;
V_OCUPACION16 = 0;
V_OCUPACION17 = 0;
V_OCUPACION18 = 0;
V_OCUPACION19 = 0;
V_OCUPACION20 = 0;
V_OCUPACION21 = 0;
if missing( V_OCUPACION ) then do;
   V_OCUPACION1 = .;
   V_OCUPACION2 = .;
   V_OCUPACION3 = .;
   V_OCUPACION5 = .;
   V_OCUPACION6 = .;
   V_OCUPACION7 = .;
   V_OCUPACION8 = .;
   V_OCUPACION9 = .;
   V_OCUPACION10 = .;
   V_OCUPACION11 = .;
   V_OCUPACION12 = .;
   V_OCUPACION13 = .;
   V_OCUPACION14 = .;
   V_OCUPACION16 = .;
   V_OCUPACION17 = .;
   V_OCUPACION18 = .;
   V_OCUPACION19 = .;
   V_OCUPACION20 = .;
   V_OCUPACION21 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_OCUPACION , BEST12. );
   %DMNORMIP( _dm12 )
   _dm_find = 0; drop _dm_find;
   if _dm12 <= '19'  then do;
      if _dm12 <= '13'  then do;
         if _dm12 <= '11'  then do;
            if _dm12 <= '10'  then do;
               if _dm12 = '1'  then do;
                  V_OCUPACION1 = 1;
                  _dm_find = 1;
```

```
            end;
            else do;
               if _dm12 = '10'  then do;
                  V_OCUPACION10 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
         else do;
            if _dm12 = '11'  then do;
               V_OCUPACION11 = 1;
               _dm_find = 1;
            end;
         end;
      end;
      else do;
         if _dm12 = '12'  then do;
            V_OCUPACION12 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '13'  then do;
               V_OCUPACION13 = 1;
               _dm_find = 1;
            end;
         end;
      end;
   end;
   else do;
      if _dm12 <= '17'  then do;
         if _dm12 <= '16'  then do;
            if _dm12 = '14'  then do;
               V_OCUPACION14 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '16'  then do;
                  V_OCUPACION16 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
         else do;
            if _dm12 = '17'  then do;
               V_OCUPACION17 = 1;
               _dm_find = 1;
            end;
         end;
      end;
      else do;
         if _dm12 = '18'  then do;
            V_OCUPACION18 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '19'  then do;
               V_OCUPACION19 = 1;
               _dm_find = 1;
            end;
```

```
            end;
         end;
      end;
   end;
   else do;
      if _dm12 <= '5'  then do;
         if _dm12 <= '21'  then do;
            if _dm12 <= '20'  then do;
               if _dm12 = '2'  then do;
                  V_OCUPACION2 = 1;
                  _dm_find = 1;
               end;
               else do;
                  if _dm12 = '20'  then do;
                     V_OCUPACION20 = 1;
                     _dm_find = 1;
                  end;
               end;
            end;
            else do;
               if _dm12 = '21'  then do;
                  V_OCUPACION21 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
         else do;
            if _dm12 = '3'  then do;
               V_OCUPACION3 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '5'  then do;
                  V_OCUPACION5 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
      end;
      else do;
         if _dm12 <= '7'  then do;
            if _dm12 = '6'  then do;
               V_OCUPACION6 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '7'  then do;
                  V_OCUPACION7 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
         else do;
            if _dm12 = '8'  then do;
               V_OCUPACION8 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '9'  then do;
```

```
                    V_OCUPACION9 = 1;
                    _dm_find = 1;
                end;
            end;
        end;
    end;
    end;
    if not _dm_find then do;
        V_OCUPACION1 = .;
        V_OCUPACION2 = .;
        V_OCUPACION3 = .;
        V_OCUPACION5 = .;
        V_OCUPACION6 = .;
        V_OCUPACION7 = .;
        V_OCUPACION8 = .;
        V_OCUPACION9 = .;
        V_OCUPACION10 = .;
        V_OCUPACION11 = .;
        V_OCUPACION12 = .;
        V_OCUPACION13 = .;
        V_OCUPACION14 = .;
        V_OCUPACION16 = .;
        V_OCUPACION17 = .;
        V_OCUPACION18 = .;
        V_OCUPACION19 = .;
        V_OCUPACION20 = .;
        V_OCUPACION21 = .;
        _DM_BAD = 1;
    end;
end;

*** Generate dummy variables for V_SEGMENTO ;
drop V_SEGMENTO0 V_SEGMENTO1 ;
if missing( V_SEGMENTO ) then do;
    V_SEGMENTO0 = .;
    V_SEGMENTO1 = .;
end;
else do;
    length _dm12 $ 12; drop _dm12 ;
    _dm12 = put( V_SEGMENTO , BEST12. );
    %DMNORMIP( _dm12 )
    if _dm12 = '1'  then do;
        V_SEGMENTO0 = 0;
        V_SEGMENTO1 = 1;
    end;
    else if _dm12 = '0'  then do;
        V_SEGMENTO0 = 1;
        V_SEGMENTO1 = 0;
    end;
    else do;
        V_SEGMENTO0 = .;
        V_SEGMENTO1 = .;
        _DM_BAD = 1;
    end;
end;

*** Generate dummy variables for V_SEXO ;
drop V_SEXO1 V_SEXO2 ;
*** encoding is sparse, initialize to zero;
```

```
V_SEXO1 = 0;
V_SEXO2 = 0;
if missing( V_SEXO ) then do;
   V_SEXO1 = .;
   V_SEXO2 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_SEXO , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '2'  then do;
      V_SEXO2 = 1;
   end;
   else if _dm12 = '1'  then do;
      V_SEXO1 = 1;
   end;
   else do;
      V_SEXO1 = .;
      V_SEXO2 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_ST_CIVIL ;
drop V_ST_CIVIL1 V_ST_CIVIL2 V_ST_CIVIL3 V_ST_CIVIL4 V_ST_CIVIL5 V_ST_CIVIL6
        V_ST_CIVIL7 ;
*** encoding is sparse, initialize to zero;
V_ST_CIVIL1 = 0;
V_ST_CIVIL2 = 0;
V_ST_CIVIL3 = 0;
V_ST_CIVIL4 = 0;
V_ST_CIVIL5 = 0;
V_ST_CIVIL6 = 0;
V_ST_CIVIL7 = 0;
if missing( V_ST_CIVIL ) then do;
   V_ST_CIVIL1 = .;
   V_ST_CIVIL2 = .;
   V_ST_CIVIL3 = .;
   V_ST_CIVIL4 = .;
   V_ST_CIVIL5 = .;
   V_ST_CIVIL6 = .;
   V_ST_CIVIL7 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_ST_CIVIL , BEST12. );
   %DMNORMIP( _dm12 )
   _dm_find = 0; drop _dm_find;
   if _dm12 <= '4'  then do;
      if _dm12 <= '2'  then do;
         if _dm12 = '1'  then do;
            V_ST_CIVIL1 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '2'  then do;
               V_ST_CIVIL2 = 1;
               _dm_find = 1;
            end;
```

```
         end;
      end;
      else do;
         if _dm12 = '3'  then do;
            V_ST_CIVIL3 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '4'  then do;
               V_ST_CIVIL4 = 1;
               _dm_find = 1;
            end;
         end;
      end;
   end;
   else do;
      if _dm12 <= '6'  then do;
         if _dm12 = '5'  then do;
            V_ST_CIVIL5 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '6'  then do;
               V_ST_CIVIL6 = 1;
               _dm_find = 1;
            end;
         end;
      end;
      else do;
         if _dm12 = '7'  then do;
            V_ST_CIVIL7 = 1;
            _dm_find = 1;
         end;
      end;
   end;
   if not _dm_find then do;
      V_ST_CIVIL1 = .;
      V_ST_CIVIL2 = .;
      V_ST_CIVIL3 = .;
      V_ST_CIVIL4 = .;
      V_ST_CIVIL5 = .;
      V_ST_CIVIL6 = .;
      V_ST_CIVIL7 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_TERRITORIO ;
drop V_TERRITORIO1 V_TERRITORIO2 V_TERRITORIO3 V_TERRITORIO4 V_TERRITORIO6
        V_TERRITORIO8 V_TERRITORIO9 V_TERRITORIO10 V_TERRITORIO11
        V_TERRITORIO12 V_TERRITORIO13 V_TERRITORIO15 V_TERRITORIO16
        V_TERRITORIO17 V_TERRITORIO19 V_TERRITORIO20 V_TERRITORIO21
        V_TERRITORIO22 V_TERRITORIO23 V_TERRITORIO24 V_TERRITORIO25 ;
*** encoding is sparse, initialize to zero;
V_TERRITORIO1 = 0;
V_TERRITORIO2 = 0;
V_TERRITORIO3 = 0;
V_TERRITORIO4 = 0;
V_TERRITORIO6 = 0;
```

```
V_TERRITORIO8 = 0;
V_TERRITORIO9 = 0;
V_TERRITORIO10 = 0;
V_TERRITORIO11 = 0;
V_TERRITORIO12 = 0;
V_TERRITORIO13 = 0;
V_TERRITORIO15 = 0;
V_TERRITORIO16 = 0;
V_TERRITORIO17 = 0;
V_TERRITORIO19 = 0;
V_TERRITORIO20 = 0;
V_TERRITORIO21 = 0;
V_TERRITORIO22 = 0;
V_TERRITORIO23 = 0;
V_TERRITORIO24 = 0;
V_TERRITORIO25 = 0;
if missing( V_TERRITORIO ) then do;
   V_TERRITORIO1 = .;
   V_TERRITORIO2 = .;
   V_TERRITORIO3 = .;
   V_TERRITORIO4 = .;
   V_TERRITORIO6 = .;
   V_TERRITORIO8 = .;
   V_TERRITORIO9 = .;
   V_TERRITORIO10 = .;
   V_TERRITORIO11 = .;
   V_TERRITORIO12 = .;
   V_TERRITORIO13 = .;
   V_TERRITORIO15 = .;
   V_TERRITORIO16 = .;
   V_TERRITORIO17 = .;
   V_TERRITORIO19 = .;
   V_TERRITORIO20 = .;
   V_TERRITORIO21 = .;
   V_TERRITORIO22 = .;
   V_TERRITORIO23 = .;
   V_TERRITORIO24 = .;
   V_TERRITORIO25 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_TERRITORIO , BEST12. );
   %DMNORMIP( _dm12 )
   _dm_find = 0; drop _dm_find;
   if _dm12 <= '20'  then do;
      if _dm12 <= '15'  then do;
         if _dm12 <= '11'  then do;
            if _dm12 <= '10'  then do;
               if _dm12 = '1'  then do;
                  V_TERRITORIO1 = 1;
                  _dm_find = 1;
               end;
               else do;
                  if _dm12 = '10'  then do;
                     V_TERRITORIO10 = 1;
                     _dm_find = 1;
                  end;
               end;
            end;
```

```
        else do;
            if _dm12 = '11' then do;
               V_TERRITORIO11 = 1;
               _dm_find = 1;
            end;
        end;
    end;
    else do;
        if _dm12 <= '13' then do;
            if _dm12 = '12' then do;
               V_TERRITORIO12 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '13' then do;
                  V_TERRITORIO13 = 1;
                  _dm_find = 1;
               end;
            end;
        end;
        else do;
            if _dm12 = '15' then do;
               V_TERRITORIO15 = 1;
               _dm_find = 1;
            end;
        end;
    end;
end;
else do;
    if _dm12 <= '19' then do;
        if _dm12 <= '17' then do;
            if _dm12 = '16' then do;
               V_TERRITORIO16 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '17' then do;
                  V_TERRITORIO17 = 1;
                  _dm_find = 1;
               end;
            end;
        end;
        else do;
            if _dm12 = '19' then do;
               V_TERRITORIO19 = 1;
               _dm_find = 1;
            end;
        end;
    end;
    else do;
        if _dm12 = '2' then do;
           V_TERRITORIO2 = 1;
           _dm_find = 1;
        end;
        else do;
            if _dm12 = '20' then do;
               V_TERRITORIO20 = 1;
               _dm_find = 1;
            end;
```

```
            end;
         end;
      end;
   end;
   else do;
      if _dm12 <= '25' then do;
         if _dm12 <= '23' then do;
            if _dm12 <= '22' then do;
               if _dm12 = '21' then do;
                  V_TERRITORIO21 = 1;
                  _dm_find = 1;
               end;
               else do;
                  if _dm12 = '22' then do;
                     V_TERRITORIO22 = 1;
                     _dm_find = 1;
                  end;
               end;
            end;
            else do;
               if _dm12 = '23' then do;
                  V_TERRITORIO23 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
         else do;
            if _dm12 = '24' then do;
               V_TERRITORIO24 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '25' then do;
                  V_TERRITORIO25 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
      end;
      else do;
         if _dm12 <= '6' then do;
            if _dm12 <= '4' then do;
               if _dm12 = '3' then do;
                  V_TERRITORIO3 = 1;
                  _dm_find = 1;
               end;
               else do;
                  if _dm12 = '4' then do;
                     V_TERRITORIO4 = 1;
                     _dm_find = 1;
                  end;
               end;
            end;
            else do;
               if _dm12 = '6' then do;
                  V_TERRITORIO6 = 1;
                  _dm_find = 1;
               end;
            end;
```

```
         end;
         else do;
            if _dm12 = '8'  then do;
               V_TERRITORIO8 = 1;
               _dm_find = 1;
            end;
            else do;
               if _dm12 = '9'  then do;
                  V_TERRITORIO9 = 1;
                  _dm_find = 1;
               end;
            end;
         end;
      end;
   end;
   if not _dm_find then do;
      V_TERRITORIO1 = .;
      V_TERRITORIO2 = .;
      V_TERRITORIO3 = .;
      V_TERRITORIO4 = .;
      V_TERRITORIO6 = .;
      V_TERRITORIO8 = .;
      V_TERRITORIO9 = .;
      V_TERRITORIO10 = .;
      V_TERRITORIO11 = .;
      V_TERRITORIO12 = .;
      V_TERRITORIO13 = .;
      V_TERRITORIO15 = .;
      V_TERRITORIO16 = .;
      V_TERRITORIO17 = .;
      V_TERRITORIO19 = .;
      V_TERRITORIO20 = .;
      V_TERRITORIO21 = .;
      V_TERRITORIO22 = .;
      V_TERRITORIO23 = .;
      V_TERRITORIO24 = .;
      V_TERRITORIO25 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_TOT_CANALES_ACTI ;
drop V_TOT_CANALES_ACTI0 V_TOT_CANALES_ACTI1 V_TOT_CANALES_ACTI2
        V_TOT_CANALES_ACTI3 V_TOT_CANALES_ACTI4 ;
*** encoding is sparse, initialize to zero;
V_TOT_CANALES_ACTI0 = 0;
V_TOT_CANALES_ACTI1 = 0;
V_TOT_CANALES_ACTI2 = 0;
V_TOT_CANALES_ACTI3 = 0;
V_TOT_CANALES_ACTI4 = 0;
if missing( V_TOT_CANALES_ACTI ) then do;
   V_TOT_CANALES_ACTI0 = .;
   V_TOT_CANALES_ACTI1 = .;
   V_TOT_CANALES_ACTI2 = .;
   V_TOT_CANALES_ACTI3 = .;
   V_TOT_CANALES_ACTI4 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
```

```
   _dm12 = put( V_TOT_CANALES_ACTI , BEST12. );
   %DMNORMIP( _dm12 )
   _dm_find = 0; drop _dm_find;
   if _dm12 <= '2' then do;
      if _dm12 <= '1' then do;
         if _dm12 = '0' then do;
            V_TOT_CANALES_ACTI0 = 1;
            _dm_find = 1;
         end;
         else do;
            if _dm12 = '1' then do;
               V_TOT_CANALES_ACTI1 = 1;
               _dm_find = 1;
            end;
         end;
      end;
      else do;
         if _dm12 = '2' then do;
            V_TOT_CANALES_ACTI2 = 1;
            _dm_find = 1;
         end;
      end;
   end;
   else do;
      if _dm12 = '3' then do;
         V_TOT_CANALES_ACTI3 = 1;
         _dm_find = 1;
      end;
      else do;
         if _dm12 = '4' then do;
            V_TOT_CANALES_ACTI4 = 1;
            _dm_find = 1;
         end;
      end;
   end;
   if not _dm_find then do;
      V_TOT_CANALES_ACTI0 = .;
      V_TOT_CANALES_ACTI1 = .;
      V_TOT_CANALES_ACTI2 = .;
      V_TOT_CANALES_ACTI3 = .;
      V_TOT_CANALES_ACTI4 = .;
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for V_WELCOME_CALL ;
drop V_WELCOME_CALL0 V_WELCOME_CALL1 ;
if missing( V_WELCOME_CALL ) then do;
   V_WELCOME_CALL0 = .;
   V_WELCOME_CALL1 = .;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( V_WELCOME_CALL , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '0' then do;
      V_WELCOME_CALL0 = 1;
      V_WELCOME_CALL1 = 0;
   end;
```

```
   else if _dm12 = '1'  then do;
      V_WELCOME_CALL0 = 0;
      V_WELCOME_CALL1 = 1;
   end;
   else do;
      V_WELCOME_CALL0 = .;
      V_WELCOME_CALL1 = .;
      _DM_BAD = 1;
   end;
end;

*** End Class Look-up, Standardization, Replacement ;


*** Omitted Cases;
if _dm_bad then do;
   _SEGMENT_ = .; Distance = .;
   goto CLUSvlex ;
end; *** omitted;

*** Compute Distances and Cluster Membership;
label _SEGMENT_ = 'Segment Id' ;
label Distance = 'Distance' ;
array CLUSvads [5] _temporary_;
drop _vqclus _vqmvar _vqnvar;
_vqmvar = 0;
do _vqclus = 1 to 5; CLUSvads [_vqclus] = 0; end;
if not missing( T_V_ANT_INT_ANTIG ) then do;
   CLUSvads [1] + ( T_V_ANT_INT_ANTIG - 0.01353613484183 )**2;
   CLUSvads [2] + ( T_V_ANT_INT_ANTIG - 0.15084657024972 )**2;
   CLUSvads [3] + ( T_V_ANT_INT_ANTIG - 0.01585564718489 )**2;
   CLUSvads [4] + ( T_V_ANT_INT_ANTIG - 0.17512458329037 )**2;
   CLUSvads [5] + ( T_V_ANT_INT_ANTIG - 0.21316701316701 )**2;
end;
else _vqmvar + 0.03160787748279;
if not missing( T_V_BCSCORE ) then do;
   CLUSvads [1] + ( T_V_BCSCORE - 0.88350219015257 )**2;
   CLUSvads [2] + ( T_V_BCSCORE - 0.83791790081321 )**2;
   CLUSvads [3] + ( T_V_BCSCORE - 0.85648230403689 )**2;
   CLUSvads [4] + ( T_V_BCSCORE - 0.83453425905061 )**2;
   CLUSvads [5] + ( T_V_BCSCORE - 0.88176048020748 )**2;
end;
else _vqmvar + 0.02334625683847;
if not missing( T_V_EDAD_CLIENTE ) then do;
   CLUSvads [1] + ( T_V_EDAD_CLIENTE - 0.03078678060874 )**2;
   CLUSvads [2] + ( T_V_EDAD_CLIENTE - 0.02523032848259 )**2;
   CLUSvads [3] + ( T_V_EDAD_CLIENTE - 0.02884893679895 )**2;
   CLUSvads [4] + ( T_V_EDAD_CLIENTE - 0.02305556209873 )**2;
   CLUSvads [5] + ( T_V_EDAD_CLIENTE - 0.02948237779698 )**2;
end;
else _vqmvar + 0.00032344323796;
if not missing( T_V_MEDIA_CH_P_5 ) then do;
   CLUSvads [1] + ( T_V_MEDIA_CH_P_5 - 0.00104250605356 )**2;
   CLUSvads [2] + ( T_V_MEDIA_CH_P_5 - 0.00802165217219 )**2;
   CLUSvads [3] + ( T_V_MEDIA_CH_P_5 - 0.00129470722867 )**2;
   CLUSvads [4] + ( T_V_MEDIA_CH_P_5 - 0.01590764121953 )**2;
   CLUSvads [5] + ( T_V_MEDIA_CH_P_5 - 0.02034233278857 )**2;
end;
else _vqmvar + 0.00211596436915;
```

```
if not missing( T_V_MOB_TDC ) then do;
   CLUSvads [1] + ( T_V_MOB_TDC - 0.08653223568587 )**2;
   CLUSvads [2] + ( T_V_MOB_TDC - 0.06799451595703 )**2;
   CLUSvads [3] + ( T_V_MOB_TDC - 0.07323533165978 )**2;
   CLUSvads [4] + ( T_V_MOB_TDC - 0.08216998985217 )**2;
   CLUSvads [5] + ( T_V_MOB_TDC - 0.10464524571096 )**2;
end;
else _vqmvar + 0.00680665216513;
if not missing( T_V_MO_R__CTAS_AB ) then do;
   CLUSvads [1] + ( T_V_MO_R__CTAS_AB - 0.26482799525504 )**2;
   CLUSvads [2] + ( T_V_MO_R__CTAS_AB - 0.18215538847117 )**2;
   CLUSvads [3] + ( T_V_MO_R__CTAS_AB - 0.16739542340766 )**2;
   CLUSvads [4] + ( T_V_MO_R__CTAS_AB - 0.18399296394019 )**2;
   CLUSvads [5] + ( T_V_MO_R__CTAS_AB - 0.311165273909 )**2;
end;
else _vqmvar + 0.01056152355377;
if not missing( T_V_MO_R__LC_MAX ) then do;
   CLUSvads [1] + ( T_V_MO_R__LC_MAX - 0.05168841928979 )**2;
   CLUSvads [2] + ( T_V_MO_R__LC_MAX - 0.02958212528224 )**2;
   CLUSvads [3] + ( T_V_MO_R__LC_MAX - 0.03367273091548 )**2;
   CLUSvads [4] + ( T_V_MO_R__LC_MAX - 0.03467300253991 )**2;
   CLUSvads [5] + ( T_V_MO_R__LC_MAX - 0.08091977324947 )**2;
end;
else _vqmvar + 0.00313706701727;
if not missing( T_V_MO_R__SALDO ) then do;
   CLUSvads [1] + ( T_V_MO_R__SALDO - 0.028946489263 )**2;
   CLUSvads [2] + ( T_V_MO_R__SALDO - 0.01410281344563 )**2;
   CLUSvads [3] + ( T_V_MO_R__SALDO - 0.01623783528254 )**2;
   CLUSvads [4] + ( T_V_MO_R__SALDO - 0.02497069924366 )**2;
   CLUSvads [5] + ( T_V_MO_R__SALDO - 0.06783971603625 )**2;
end;
else _vqmvar + 0.00316007417442;
if not missing( T_V_NDEC_ATM_3 ) then do;
   CLUSvads [1] + ( T_V_NDEC_ATM_3 - 0.00263620386643 )**2;
   CLUSvads [2] + ( T_V_NDEC_ATM_3 - 0.28610709117221 )**2;
   CLUSvads [3] + ( T_V_NDEC_ATM_3 - 0.00626515763945 )**2;
   CLUSvads [4] + ( T_V_NDEC_ATM_3 - 0.2902553413236 )**2;
   CLUSvads [5] + ( T_V_NDEC_ATM_3 - 0.29765193370165 )**2;
end;
else _vqmvar + 0.08469255878939;
if not missing( T_V_NDEC_ATM_5 ) then do;
   CLUSvads [1] + ( T_V_NDEC_ATM_5 - 0.00659050966608 )**2;
   CLUSvads [2] + ( T_V_NDEC_ATM_5 - 0.34189580318379 )**2;
   CLUSvads [3] + ( T_V_NDEC_ATM_5 - 0.00747776879547 )**2;
   CLUSvads [4] + ( T_V_NDEC_ATM_5 - 0.3411933298593 )**2;
   CLUSvads [5] + ( T_V_NDEC_ATM_5 - 0.35583563535911 )**2;
end;
else _vqmvar + 0.0587176065859;
if not missing( T_V_NDEC_B_D_OO_4 ) then do;
   CLUSvads [1] + ( T_V_NDEC_B_D_OO_4 - 0.33391915641476 )**2;
   CLUSvads [2] + ( T_V_NDEC_B_D_OO_4 - 0.00101302460202 )**2;
   CLUSvads [3] + ( T_V_NDEC_B_D_OO_4 - 0.00161681487469 )**2;
   CLUSvads [4] + ( T_V_NDEC_B_D_OO_4 - 0.00234497133923 )**2;
   CLUSvads [5] + ( T_V_NDEC_B_D_OO_4 - 0.208908839779 )**2;
end;
else _vqmvar + 0.0304085229676;
if not missing( T_V_NDEC_CH_P_6 ) then do;
   CLUSvads [1] + ( T_V_NDEC_CH_P_6 - 0.02636203866432 )**2;
   CLUSvads [2] + ( T_V_NDEC_CH_P_6 - 0.44167872648335 )**2;
```

```
   CLUSvads [3] + ( T_V_NDEC_CH_P_6 - 0.01738075990299 )**2;
   CLUSvads [4] + ( T_V_NDEC_CH_P_6 - 0.44575299635226 )**2;
   CLUSvads [5] + ( T_V_NDEC_CH_P_6 - 0.48342541436464 )**2;
end;
else _vqmvar + 0.07400743204386;
if not missing( T_V_NDEC_OO_R_CTAS_AB_5 ) then do;
   CLUSvads [1] + ( T_V_NDEC_OO_R_CTAS_AB_5 - 0.13708260105448 )**2;
   CLUSvads [2] + ( T_V_NDEC_OO_R_CTAS_AB_5 - 0.00173661360347 )**2;
   CLUSvads [3] + ( T_V_NDEC_OO_R_CTAS_AB_5 - 0.00242522231204 )**2;
   CLUSvads [4] + ( T_V_NDEC_OO_R_CTAS_AB_5 - 0.00234497133923 )**2;
   CLUSvads [5] + ( T_V_NDEC_OO_R_CTAS_AB_5 - 0.05697513812154 )**2;
end;
else _vqmvar + 0.00930902545269;
if not missing( T_V_NDEC_OO_R_LC_MAX_5 ) then do;
   CLUSvads [1] + ( T_V_NDEC_OO_R_LC_MAX_5 - 0.1335676625659 )**2;
   CLUSvads [2] + ( T_V_NDEC_OO_R_LC_MAX_5 - 0.00246020260492 )**2;
   CLUSvads [3] + ( T_V_NDEC_OO_R_LC_MAX_5 - 0.00323362974939 )**2;
   CLUSvads [4] + ( T_V_NDEC_OO_R_LC_MAX_5 - 0.00286607608129 )**2;
   CLUSvads [5] + ( T_V_NDEC_OO_R_LC_MAX_5 - 0.05524861878453 )**2;
end;
else _vqmvar + 0.00966020288493;
if not missing( T_V_NDEC_OO_R_LC_SUM_3 ) then do;
   CLUSvads [1] + ( T_V_NDEC_OO_R_LC_SUM_3 - 0.21968365553602 )**2;
   CLUSvads [2] + ( T_V_NDEC_OO_R_LC_SUM_3 - 0.00144717800289 )**2;
   CLUSvads [3] + ( T_V_NDEC_OO_R_LC_SUM_3 - 0.00282942603071 )**2;
   CLUSvads [4] + ( T_V_NDEC_OO_R_LC_SUM_3 - 0.00260552371026 )**2;
   CLUSvads [5] + ( T_V_NDEC_OO_R_LC_SUM_3 - 0.08632596685082 )**2;
end;
else _vqmvar + 0.02633551731834;
if not missing( T_V_NDEC_TO_CTAS_AB_5 ) then do;
   CLUSvads [1] + ( T_V_NDEC_TO_CTAS_AB_5 - 0.15465729349736 )**2;
   CLUSvads [2] + ( T_V_NDEC_TO_CTAS_AB_5 - 0.00564399421128 )**2;
   CLUSvads [3] + ( T_V_NDEC_TO_CTAS_AB_5 - 0.00424413904607 )**2;
   CLUSvads [4] + ( T_V_NDEC_TO_CTAS_AB_5 - 0.00599270453361 )**2;
   CLUSvads [5] + ( T_V_NDEC_TO_CTAS_AB_5 - 0.07113259668508 )**2;
end;
else _vqmvar + 0.01214217360152;
if not missing( T_V_NDEC_TO_R_LC_MAX_5 ) then do;
   CLUSvads [1] + ( T_V_NDEC_TO_R_LC_MAX_5 - 0.13268892794376 )**2;
   CLUSvads [2] + ( T_V_NDEC_TO_R_LC_MAX_5 - 0.00593342981186 )**2;
   CLUSvads [3] + ( T_V_NDEC_TO_R_LC_MAX_5 - 0.00464834276475 )**2;
   CLUSvads [4] + ( T_V_NDEC_TO_R_LC_MAX_5 - 0.00677436164669 )**2;
   CLUSvads [5] + ( T_V_NDEC_TO_R_LC_MAX_5 - 0.05732044198895 )**2;
end;
else _vqmvar + 0.01090349791117;
if not missing( T_V_NDEC_TO_R_LC_SUM_3 ) then do;
   CLUSvads [1] + ( T_V_NDEC_TO_R_LC_SUM_3 - 0.27240773286467 )**2;
   CLUSvads [2] + ( T_V_NDEC_TO_R_LC_SUM_3 - 0.00318379160636 )**2;
   CLUSvads [3] + ( T_V_NDEC_TO_R_LC_SUM_3 - 0.00404203718674 )**2;
   CLUSvads [4] + ( T_V_NDEC_TO_R_LC_SUM_3 - 0.00573215216258 )**2;
   CLUSvads [5] + ( T_V_NDEC_TO_R_LC_SUM_3 - 0.10704419889502 )**2;
end;
else _vqmvar + 0.03346953951335;
if not missing( T_V_NDEC_TO_R_SALDO_4 ) then do;
   CLUSvads [1] + ( T_V_NDEC_TO_R_SALDO_4 - 0.35764499121265 )**2;
   CLUSvads [2] + ( T_V_NDEC_TO_R_SALDO_4 - 0.0068017366136 )**2;
   CLUSvads [3] + ( T_V_NDEC_TO_R_SALDO_4 - 0.00565885206143 )**2;
   CLUSvads [4] + ( T_V_NDEC_TO_R_SALDO_4 - 0.0166753517457 )**2;
   CLUSvads [5] + ( T_V_NDEC_TO_R_SALDO_4 - 0.34530386740331 )**2;
```

```
end;
else _vqmvar + 0.0456132802978;
if not missing( T_V_NDEC_TO_SALDO_5 ) then do;
   CLUSvads [1] + ( T_V_NDEC_TO_SALDO_5 - 0.46309314586994 )**2;
   CLUSvads [2] + ( T_V_NDEC_TO_SALDO_5 - 0.06266280752532 )**2;
   CLUSvads [3] + ( T_V_NDEC_TO_SALDO_5 - 0.01616814874696 )**2;
   CLUSvads [4] + ( T_V_NDEC_TO_SALDO_5 - 0.06096925482021 )**2;
   CLUSvads [5] + ( T_V_NDEC_TO_SALDO_5 - 0.49654696132596 )**2;
end;
else _vqmvar + 0.08380070324892;
if not missing( T_V_NINC_ATM_5 ) then do;
   CLUSvads [1] + ( T_V_NINC_ATM_5 - 0.00922671353251 )**2;
   CLUSvads [2] + ( T_V_NINC_ATM_5 - 0.53256150506512 )**2;
   CLUSvads [3] + ( T_V_NINC_ATM_5 - 0.00707356507679 )**2;
   CLUSvads [4] + ( T_V_NINC_ATM_5 - 0.514721208963 )**2;
   CLUSvads [5] + ( T_V_NINC_ATM_5 - 0.51691988950276 )**2;
end;
else _vqmvar + 0.09902111956259;
if not missing( T_V_NINC_B_D_OO_4 ) then do;
   CLUSvads [1] + ( T_V_NINC_B_D_OO_4 - 0.3804920913884 )**2;
   CLUSvads [2] + ( T_V_NINC_B_D_OO_4 - 0.02879884225759 )**2;
   CLUSvads [3] + ( T_V_NINC_B_D_OO_4 - 0.13561034761519 )**2;
   CLUSvads [4] + ( T_V_NINC_B_D_OO_4 - 0.05914538822303 )**2;
   CLUSvads [5] + ( T_V_NINC_B_D_OO_4 - 0.23653314917127 )**2;
end;
else _vqmvar + 0.05291117844345;
if not missing( T_V_NINC_B_D_TO_5 ) then do;
   CLUSvads [1] + ( T_V_NINC_B_D_TO_5 - 0.51054481546572 )**2;
   CLUSvads [2] + ( T_V_NINC_B_D_TO_5 - 0.13039073806078 )**2;
   CLUSvads [3] + ( T_V_NINC_B_D_TO_5 - 0.18350848827809 )**2;
   CLUSvads [4] + ( T_V_NINC_B_D_TO_5 - 0.23996873371547 )**2;
   CLUSvads [5] + ( T_V_NINC_B_D_TO_5 - 0.45096685082872 )**2;
end;
else _vqmvar + 0.08531426796217;
if not missing( T_V_NINC_CH_P_3 ) then do;
   CLUSvads [1] + ( T_V_NINC_CH_P_3 - 0.02021089630931 )**2;
   CLUSvads [2] + ( T_V_NINC_CH_P_3 - 0.52807525325615 )**2;
   CLUSvads [3] + ( T_V_NINC_CH_P_3 - 0.0109135004042 )**2;
   CLUSvads [4] + ( T_V_NINC_CH_P_3 - 0.48645127670661 )**2;
   CLUSvads [5] + ( T_V_NINC_CH_P_3 - 0.49620165745856 )**2;
end;
else _vqmvar + 0.13717251529056;
if not missing( T_V_NINC_CH_P_6 ) then do;
   CLUSvads [1] + ( T_V_NINC_CH_P_6 - 0.01616871704745 )**2;
   CLUSvads [2] + ( T_V_NINC_CH_P_6 - 0.50628075253256 )**2;
   CLUSvads [3] + ( T_V_NINC_CH_P_6 - 0.00913500404203 )**2;
   CLUSvads [4] + ( T_V_NINC_CH_P_6 - 0.49786347055758 )**2;
   CLUSvads [5] + ( T_V_NINC_CH_P_6 - 0.50027624309392 )**2;
end;
else _vqmvar + 0.08168450233875;
if not missing( T_V_NINC_OO_R_LC_SUM_4 ) then do;
   CLUSvads [1] + ( T_V_NINC_OO_R_LC_SUM_4 - 0.19244288224956 )**2;
   CLUSvads [2] + ( T_V_NINC_OO_R_LC_SUM_4 - 0.12011577424023 )**2;
   CLUSvads [3] + ( T_V_NINC_OO_R_LC_SUM_4 - 0.17441390460792 )**2;
   CLUSvads [4] + ( T_V_NINC_OO_R_LC_SUM_4 - 0.20739968733715 )**2;
   CLUSvads [5] + ( T_V_NINC_OO_R_LC_SUM_4 - 0.09668508287292 )**2;
end;
else _vqmvar + 0.05466767192459;
if not missing( T_V_NINC_OO_R_SALDO_5 ) then do;
```

```
   CLUSvads [1] + ( T_V_NINC_OO_R_SALDO_5 - 0.46748681898066 )**2;
   CLUSvads [2] + ( T_V_NINC_OO_R_SALDO_5 - 0.12272069464544 )**2;
   CLUSvads [3] + ( T_V_NINC_OO_R_SALDO_5 - 0.17724333063864 )**2;
   CLUSvads [4] + ( T_V_NINC_OO_R_SALDO_5 - 0.21130797290255 )**2;
   CLUSvads [5] + ( T_V_NINC_OO_R_SALDO_5 - 0.29488950276243 )**2;
end;
else _vqmvar + 0.0761035485543;
if not missing( T_V_NINC_TO_R_LC_SUM_4 ) then do;
   CLUSvads [1] + ( T_V_NINC_TO_R_LC_SUM_4 - 0.27943760984182 )**2;
   CLUSvads [2] + ( T_V_NINC_TO_R_LC_SUM_4 - 0.13704775687409 )**2;
   CLUSvads [3] + ( T_V_NINC_TO_R_LC_SUM_4 - 0.18047696038803 )**2;
   CLUSvads [4] + ( T_V_NINC_TO_R_LC_SUM_4 - 0.22615945805106 )**2;
   CLUSvads [5] + ( T_V_NINC_TO_R_LC_SUM_4 - 0.30801104972375 )**2;
end;
else _vqmvar + 0.06817519584134;
if not missing( T_V_NINC_TO_SALDO_5 ) then do;
   CLUSvads [1] + ( T_V_NINC_TO_SALDO_5 - 0.50878734622144 )**2;
   CLUSvads [2] + ( T_V_NINC_TO_SALDO_5 - 0.15759768451519 )**2;
   CLUSvads [3] + ( T_V_NINC_TO_SALDO_5 - 0.18835893290218 )**2;
   CLUSvads [4] + ( T_V_NINC_TO_SALDO_5 - 0.24882751433038 )**2;
   CLUSvads [5] + ( T_V_NINC_TO_SALDO_5 - 0.48825966850828 )**2;
end;
else _vqmvar + 0.09154792558419;
if not missing( T_V_OO_R_BANCO_CTAS_AB ) then do;
   CLUSvads [1] + ( T_V_OO_R_BANCO_CTAS_AB - 0.15289781392984 )**2;
   CLUSvads [2] + ( T_V_OO_R_BANCO_CTAS_AB - 0.04264232008592 )**2;
   CLUSvads [3] + ( T_V_OO_R_BANCO_CTAS_AB - 0.1214629271035 )**2;
   CLUSvads [4] + ( T_V_OO_R_BANCO_CTAS_AB - 0.04003015454202 )**2;
   CLUSvads [5] + ( T_V_OO_R_BANCO_CTAS_AB - 0.09152407481098 )**2;
end;
else _vqmvar + 0.00976192748896;
if not missing( T_V_OO_R_BANCO_LC_MAX ) then do;
   CLUSvads [1] + ( T_V_OO_R_BANCO_LC_MAX - 0.07428072543115 )**2;
   CLUSvads [2] + ( T_V_OO_R_BANCO_LC_MAX - 0.01894526269519 )**2;
   CLUSvads [3] + ( T_V_OO_R_BANCO_LC_MAX - 0.05204504591167 )**2;
   CLUSvads [4] + ( T_V_OO_R_BANCO_LC_MAX - 0.01922640334836 )**2;
   CLUSvads [5] + ( T_V_OO_R_BANCO_LC_MAX - 0.05314628678638 )**2;
end;
else _vqmvar + 0.00480129602921;
if not missing( T_V_OO_R_BANCO_SALDO ) then do;
   CLUSvads [1] + ( T_V_OO_R_BANCO_SALDO - 0.05663752757119 )**2;
   CLUSvads [2] + ( T_V_OO_R_BANCO_SALDO - 0.0127559703642 )**2;
   CLUSvads [3] + ( T_V_OO_R_BANCO_SALDO - 0.03949396534733 )**2;
   CLUSvads [4] + ( T_V_OO_R_BANCO_SALDO - 0.01379683200026 )**2;
   CLUSvads [5] + ( T_V_OO_R_BANCO_SALDO - 0.03713395181842 )**2;
end;
else _vqmvar + 0.00401638718795;
if not missing( T_V_TOT_CRED_BCO ) then do;
   CLUSvads [1] + ( T_V_TOT_CRED_BCO - 0.004329004329 )**2;
   CLUSvads [2] + ( T_V_TOT_CRED_BCO - 0.10258390449528 )**2;
   CLUSvads [3] + ( T_V_TOT_CRED_BCO - 0.00224416517055 )**2;
   CLUSvads [4] + ( T_V_TOT_CRED_BCO - 0.07322309254424 )**2;
   CLUSvads [5] + ( T_V_TOT_CRED_BCO - 0.16628299570288 )**2;
end;
else _vqmvar + 0.01510342728204;
if not missing( T_V_VM_ATM500_5 ) then do;
   CLUSvads [1] + ( T_V_VM_ATM500_5 - 0.00421792618629 )**2;
   CLUSvads [2] + ( T_V_VM_ATM500_5 - 0.82709117221418 )**2;
   CLUSvads [3] + ( T_V_VM_ATM500_5 - 0.00606305578011 )**2;
```

```
   CLUSvads [4] + ( T_V_VM_ATM500_5 - 0.8001042209484 )**2;
   CLUSvads [5] + ( T_V_VM_ATM500_5 - 0.83080110497237 )**2;
end;
else _vqmvar + 0.21825355437086;
if not missing( T_V_VM_B_D_TO45_5 ) then do;
   CLUSvads [1] + ( T_V_VM_B_D_TO45_5 - 0.2530755711775 )**2;
   CLUSvads [2] + ( T_V_VM_B_D_TO45_5 - 0.00683068017366 )**2;
   CLUSvads [3] + ( T_V_VM_B_D_TO45_5 - 0.02134195634599 )**2;
   CLUSvads [4] + ( T_V_VM_B_D_TO45_5 - 0.02657634184471 )**2;
   CLUSvads [5] + ( T_V_VM_B_D_TO45_5 - 0.38563535911602 )**2;
end;
else _vqmvar + 0.05700739134849;
if not missing( T_V_VM_CH_P0_4 ) then do;
   CLUSvads [1] + ( T_V_VM_CH_P0_4 - 0.04130052724077 )**2;
   CLUSvads [2] + ( T_V_VM_CH_P0_4 - 0.95933429811866 )**2;
   CLUSvads [3] + ( T_V_VM_CH_P0_4 - 0.0218270008084 )**2;
   CLUSvads [4] + ( T_V_VM_CH_P0_4 - 0.95609692548202 )**2;
   CLUSvads [5] + ( T_V_VM_CH_P0_4 - 0.98843232044198 )**2;
end;
else _vqmvar + 0.2093917829876;
if not missing( T_V_VM_CH_P100_4 ) then do;
   CLUSvads [1] + ( T_V_VM_CH_P100_4 - 0.01801405975395 )**2;
   CLUSvads [2] + ( T_V_VM_CH_P100_4 - 0.91418234442836 )**2;
   CLUSvads [3] + ( T_V_VM_CH_P100_4 - 0.00939773645917 )**2;
   CLUSvads [4] + ( T_V_VM_CH_P100_4 - 0.91896821261073 )**2;
   CLUSvads [5] + ( T_V_VM_CH_P100_4 - 0.96685082872928 )**2;
end;
else _vqmvar + 0.20997932243437;
if not missing( T_V_VM_CH_P1500_4 ) then do;
   CLUSvads [1] + ( T_V_VM_CH_P1500_4 - 0.00659050966608 )**2;
   CLUSvads [2] + ( T_V_VM_CH_P1500_4 - 0.55289435600578 )**2;
   CLUSvads [3] + ( T_V_VM_CH_P1500_4 - 0.00373888439773 )**2;
   CLUSvads [4] + ( T_V_VM_CH_P1500_4 - 0.59914017717561 )**2;
   CLUSvads [5] + ( T_V_VM_CH_P1500_4 - 0.73342541436464 )**2;
end;
else _vqmvar + 0.18493045587244;
if not missing( T_V_VM_OO_R_CTAS_AB0_5 ) then do;
   CLUSvads [1] + ( T_V_VM_OO_R_CTAS_AB0_5 - 0.79437609841827 )**2;
   CLUSvads [2] + ( T_V_VM_OO_R_CTAS_AB0_5 - 0.01678726483357 )**2;
   CLUSvads [3] + ( T_V_VM_OO_R_CTAS_AB0_5 - 0.06523848019401 )**2;
   CLUSvads [4] + ( T_V_VM_OO_R_CTAS_AB0_5 - 0.03053673788431 )**2;
   CLUSvads [5] + ( T_V_VM_OO_R_CTAS_AB0_5 - 0.54875690607734 )**2;
end;
else _vqmvar + 0.10121031985329;
if not missing( T_V_VM_OO_R_CTAS_AB1_4 ) then do;
   CLUSvads [1] + ( T_V_VM_OO_R_CTAS_AB1_4 - 0.4354130052724 )**2;
   CLUSvads [2] + ( T_V_VM_OO_R_CTAS_AB1_4 - 0.00492040520984 )**2;
   CLUSvads [3] + ( T_V_VM_OO_R_CTAS_AB1_4 - 0.03385206143896 )**2;
   CLUSvads [4] + ( T_V_VM_OO_R_CTAS_AB1_4 - 0.01393955184992 )**2;
   CLUSvads [5] + ( T_V_VM_OO_R_CTAS_AB1_4 - 0.23756906077348 )**2;
end;
else _vqmvar + 0.05319824653298;
if not missing( T_V_VM_OO_R_CTAS_AB3_5 ) then do;
   CLUSvads [1] + ( T_V_VM_OO_R_CTAS_AB3_5 - 0.09771528998242 )**2;
   CLUSvads [2] + ( T_V_VM_OO_R_CTAS_AB3_5 - 0.00057887120115 )**2;
   CLUSvads [3] + ( T_V_VM_OO_R_CTAS_AB3_5 - 0.00533548908649 )**2;
   CLUSvads [4] + ( T_V_VM_OO_R_CTAS_AB3_5 - 0.00239708181344 )**2;
   CLUSvads [5] + ( T_V_VM_OO_R_CTAS_AB3_5 - 0.04668508287292 )**2;
end;
```

```
else _vqmvar + 0.01101045377807;
if not missing( T_V_VM_OO_R_LC_MAX25000_4 ) then do;
   CLUSvads [1] + ( T_V_VM_OO_R_LC_MAX25000_4 - 0.39806678383128 )**2;
   CLUSvads [2] + ( T_V_VM_OO_R_LC_MAX25000_4 - 0.00542691751085 )**2;
   CLUSvads [3] + ( T_V_VM_OO_R_LC_MAX25000_4 - 0.0269805982215 )**2;
   CLUSvads [4] + ( T_V_VM_OO_R_LC_MAX25000_4 - 0.01211568525273 )**2;
   CLUSvads [5] + ( T_V_VM_OO_R_LC_MAX25000_4 - 0.28884668508287 )**2;
end;
else _vqmvar + 0.05641188228842;
if not missing( T_V_VM_OO_R_SALDO25000_4 ) then do;
   CLUSvads [1] + ( T_V_VM_OO_R_SALDO25000_4 - 0.21704745166959 )**2;
   CLUSvads [2] + ( T_V_VM_OO_R_SALDO25000_4 - 0.00209840810419 )**2;
   CLUSvads [3] + ( T_V_VM_OO_R_SALDO25000_4 - 0.01485448666127 )**2;
   CLUSvads [4] + ( T_V_VM_OO_R_SALDO25000_4 - 0.00755601875977 )**2;
   CLUSvads [5] + ( T_V_VM_OO_R_SALDO25000_4 - 0.15538674033149 )**2;
end;
else _vqmvar + 0.02959837931704;
if not missing( T_V_VM_TO_CTAS_AB2_5 ) then do;
   CLUSvads [1] + ( T_V_VM_TO_CTAS_AB2_5 - 0.57961335676625 )**2;
   CLUSvads [2] + ( T_V_VM_TO_CTAS_AB2_5 - 0.01001447178002 )**2;
   CLUSvads [3] + ( T_V_VM_TO_CTAS_AB2_5 - 0.0415521422797 )**2;
   CLUSvads [4] + ( T_V_VM_TO_CTAS_AB2_5 - 0.02417926003126 )**2;
   CLUSvads [5] + ( T_V_VM_TO_CTAS_AB2_5 - 0.40994475138121 )**2;
end;
else _vqmvar + 0.07680182627679;
if not missing( T_V_VM_TO_R_LC_MAX25000_4 ) then do;
   CLUSvads [1] + ( T_V_VM_TO_R_LC_MAX25000_4 - 0.52636203866432 )**2;
   CLUSvads [2] + ( T_V_VM_TO_R_LC_MAX25000_4 - 0.01794500723589 )**2;
   CLUSvads [3] + ( T_V_VM_TO_R_LC_MAX25000_4 - 0.04001616814874 )**2;
   CLUSvads [4] + ( T_V_VM_TO_R_LC_MAX25000_4 - 0.04598749348619 )**2;
   CLUSvads [5] + ( T_V_VM_TO_R_LC_MAX25000_4 - 0.54627071823204 )**2;
end;
else _vqmvar + 0.09497801884647;
if not missing( T_V_VM_TO_R_LC_SUM5000_4 ) then do;
   CLUSvads [1] + ( T_V_VM_TO_R_LC_SUM5000_4 - 0.91520210896309 )**2;
   CLUSvads [2] + ( T_V_VM_TO_R_LC_SUM5000_4 - 0.09587554269175 )**2;
   CLUSvads [3] + ( T_V_VM_TO_R_LC_SUM5000_4 - 0.11004446240905 )**2;
   CLUSvads [4] + ( T_V_VM_TO_R_LC_SUM5000_4 - 0.16492965085982 )**2;
   CLUSvads [5] + ( T_V_VM_TO_R_LC_SUM5000_4 - 0.9381906077348 )**2;
end;
else _vqmvar + 0.15338699830107;
if not missing( T_V_VM_TO_R_SALDO1000_5 ) then do;
   CLUSvads [1] + ( T_V_VM_TO_R_SALDO1000_5 - 0.81757469244288 )**2;
   CLUSvads [2] + ( T_V_VM_TO_R_SALDO1000_5 - 0.02222865412445 )**2;
   CLUSvads [3] + ( T_V_VM_TO_R_SALDO1000_5 - 0.06790622473726 )**2;
   CLUSvads [4] + ( T_V_VM_TO_R_SALDO1000_5 - 0.06690984887962 )**2;
   CLUSvads [5] + ( T_V_VM_TO_R_SALDO1000_5 - 0.83370165745856 )**2;
end;
else _vqmvar + 0.13005532350971;
if not missing( T_V_VM_TO_R_SALDO25000_4 ) then do;
   CLUSvads [1] + ( T_V_VM_TO_R_SALDO25000_4 - 0.30667838312829 )**2;
   CLUSvads [2] + ( T_V_VM_TO_R_SALDO25000_4 - 0.00318379160636 )**2;
   CLUSvads [3] + ( T_V_VM_TO_R_SALDO25000_4 - 0.01909862570735 )**2;
   CLUSvads [4] + ( T_V_VM_TO_R_SALDO25000_4 - 0.01133402813965 )**2;
   CLUSvads [5] + ( T_V_VM_TO_R_SALDO25000_4 - 0.32061464088397 )**2;
end;
else _vqmvar + 0.05309158432019;
if not missing( T_V_VM_TO_SALDO10000_5 ) then do;
   CLUSvads [1] + ( T_V_VM_TO_SALDO10000_5 - 0.68681898066783 )**2;
```

```
   CLUSvads [2] + ( T_V_VM_TO_SALDO10000_5 - 0.10836468885672 )**2;
   CLUSvads [3] + ( T_V_VM_TO_SALDO10000_5 - 0.05699272433306 )**2;
   CLUSvads [4] + ( T_V_VM_TO_SALDO10000_5 - 0.11610213652944 )**2;
   CLUSvads [5] + ( T_V_VM_TO_SALDO10000_5 - 0.80027624309392 )**2;
end;
else _vqmvar + 0.13560771304747;
if not missing( V_ATM0 ) then do;
   CLUSvads [1] + ( V_ATM0 - 0.9244288224956 )**2;
   CLUSvads [2] + ( V_ATM0 - 0.01562952243125 )**2;
   CLUSvads [3] + ( V_ATM0 - 0.95836701697655 )**2;
   CLUSvads [4] + ( V_ATM0 - 0.0333507034914 )**2;
   CLUSvads [5] + ( V_ATM0 - 0.00552486187845 )**2;
end;
else _vqmvar + 0.2125548719555;
if not missing( V_ATM1 ) then do;
   CLUSvads [1] + ( V_ATM1 - 0.07557117750439 )**2;
   CLUSvads [2] + ( V_ATM1 - 0.98437047756874 )**2;
   CLUSvads [3] + ( V_ATM1 - 0.04163298302344 )**2;
   CLUSvads [4] + ( V_ATM1 - 0.96664929650859 )**2;
   CLUSvads [5] + ( V_ATM1 - 0.99447513812154 )**2;
end;
else _vqmvar + 0.2125548719555;
if not missing( V_CANAL1 ) then do;
   CLUSvads [1] + ( V_CANAL1 - 0.71177504393673 )**2;
   CLUSvads [2] + ( V_CANAL1 - 0.2341534008683 )**2;
   CLUSvads [3] + ( V_CANAL1 - 0.60145513338722 )**2;
   CLUSvads [4] + ( V_CANAL1 - 0.00312662845231 )**2;
   CLUSvads [5] + ( V_CANAL1 - 0.15825846579129 )**2;
end;
else _vqmvar + 0.20911589136128;
if not missing( V_CANAL2 ) then do;
   CLUSvads [1] + ( V_CANAL2 - 0.18804920913884 )**2;
   CLUSvads [2] + ( V_CANAL2 - 0.02547033285094 )**2;
   CLUSvads [3] + ( V_CANAL2 - 0.21948261924009 )**2;
   CLUSvads [4] + ( V_CANAL2 - 0.00364773319437 )**2;
   CLUSvads [5] + ( V_CANAL2 - 0.07878369039391 )**2;
end;
else _vqmvar + 0.07950872410252;
if not missing( V_CANAL3 ) then do;
   CLUSvads [1] + ( V_CANAL3 - 0 )**2;
   CLUSvads [2] + ( V_CANAL3 - 0.00028943560057 )**2;
   CLUSvads [3] + ( V_CANAL3 - 0 )**2;
   CLUSvads [4] + ( V_CANAL3 - 0 )**2;
   CLUSvads [5] + ( V_CANAL3 - 0.00207325501036 )**2;
end;
else _vqmvar + 0.00040539165973;
if not missing( V_CANAL5 ) then do;
   CLUSvads [1] + ( V_CANAL5 - 0.02987697715289 )**2;
   CLUSvads [2] + ( V_CANAL5 - 0 )**2;
   CLUSvads [3] + ( V_CANAL5 - 0.01131770412287 )**2;
   CLUSvads [4] + ( V_CANAL5 - 0 )**2;
   CLUSvads [5] + ( V_CANAL5 - 0.00414651002073 )**2;
end;
else _vqmvar + 0.00514410563404;
if not missing( V_CANAL6 ) then do;
   CLUSvads [1] + ( V_CANAL6 - 0.07029876977152 )**2;
   CLUSvads [2] + ( V_CANAL6 - 0.74008683068017 )**2;
   CLUSvads [3] + ( V_CANAL6 - 0.16774454324979 )**2;
   CLUSvads [4] + ( V_CANAL6 - 0.9932256383533 )**2;
```

```
   CLUSvads [5] + ( V_CANAL6 - 0.75673807878369 )**2;
end;
else _vqmvar + 0.23801406725505;
if not missing( V_CELULAR0 ) then do;
   CLUSvads [1] + ( V_CELULAR0 - 0.00351493848857 )**2;
   CLUSvads [2] + ( V_CELULAR0 - 0.00636758321273 )**2;
   CLUSvads [3] + ( V_CELULAR0 - 0.00889248181083 )**2;
   CLUSvads [4] + ( V_CELULAR0 - 0.00416883793642 )**2;
   CLUSvads [5] + ( V_CELULAR0 - 0.0110497237569 )**2;
end;
else _vqmvar + 0.00704615718526;
if not missing( V_CELULAR1 ) then do;
   CLUSvads [1] + ( V_CELULAR1 - 0.99648506151142 )**2;
   CLUSvads [2] + ( V_CELULAR1 - 0.99363241678726 )**2;
   CLUSvads [3] + ( V_CELULAR1 - 0.99110751818916 )**2;
   CLUSvads [4] + ( V_CELULAR1 - 0.99583116206357 )**2;
   CLUSvads [5] + ( V_CELULAR1 - 0.98895027624309 )**2;
end;
else _vqmvar + 0.00704615718526;
if not missing( V_CLASIF_CTE1 ) then do;
   CLUSvads [1] + ( V_CLASIF_CTE1 - 0.0088028169014 )**2;
   CLUSvads [2] + ( V_CLASIF_CTE1 - 0.41048930559624 )**2;
   CLUSvads [3] + ( V_CLASIF_CTE1 - 0.12464474218432 )**2;
   CLUSvads [4] + ( V_CLASIF_CTE1 - 0.38679245283018 )**2;
   CLUSvads [5] + ( V_CLASIF_CTE1 - 0.02217602217602 )**2;
end;
else _vqmvar + 0.18925543584257;
if not missing( V_CLASIF_CTE2 ) then do;
   CLUSvads [1] + ( V_CLASIF_CTE2 - 0.99119718309859 )**2;
   CLUSvads [2] + ( V_CLASIF_CTE2 - 0.32405508350424 )**2;
   CLUSvads [3] + ( V_CLASIF_CTE2 - 0.80308566788469 )**2;
   CLUSvads [4] + ( V_CLASIF_CTE2 - 0.2835429769392 )**2;
   CLUSvads [5] + ( V_CLASIF_CTE2 - 0.96812196812196 )**2;
end;
else _vqmvar + 0.2450985450875;
if not missing( V_CLASIF_CTE3 ) then do;
   CLUSvads [1] + ( V_CLASIF_CTE3 - 0 )**2;
   CLUSvads [2] + ( V_CLASIF_CTE3 - 0.2654556108995 )**2;
   CLUSvads [3] + ( V_CLASIF_CTE3 - 0.07226958993097 )**2;
   CLUSvads [4] + ( V_CLASIF_CTE3 - 0.3296645702306 )**2;
   CLUSvads [5] + ( V_CLASIF_CTE3 - 0.009702009702 )**2;
end;
else _vqmvar + 0.14524249045393;
if not missing( V_ENTREGAT0 ) then do;
   CLUSvads [1] + ( V_ENTREGAT0 - 0.90861159929701 )**2;
   CLUSvads [2] + ( V_ENTREGAT0 - 1 )**2;
   CLUSvads [3] + ( V_ENTREGAT0 - 0.85367825383993 )**2;
   CLUSvads [4] + ( V_ENTREGAT0 - 0 )**2;
   CLUSvads [5] + ( V_ENTREGAT0 - 0.72099447513812 )**2;
end;
else _vqmvar + 0.20048982430697;
if not missing( V_ENTREGAT1 ) then do;
   CLUSvads [1] + ( V_ENTREGAT1 - 0.09138840070298 )**2;
   CLUSvads [2] + ( V_ENTREGAT1 - 0 )**2;
   CLUSvads [3] + ( V_ENTREGAT1 - 0.14632174616006 )**2;
   CLUSvads [4] + ( V_ENTREGAT1 - 1 )**2;
   CLUSvads [5] + ( V_ENTREGAT1 - 0.27900552486187 )**2;
end;
else _vqmvar + 0.20048982430697;
```

```
if not missing( V_N_ACADEMICO1 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO1 - 0.07441860465116 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO1 - 0.08939526730937 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO1 - 0.08415841584158 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO1 - 0.08460325801366 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO1 - 0.0759052924791 )**2;
end;
else _vqmvar + 0.07760100755068;
if not missing( V_N_ACADEMICO2 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO2 - 0.00465116279069 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO2 - 0.00146070698217 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO2 - 0 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO2 - 0.00315291644771 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO2 - 0.00626740947075 )**2;
end;
else _vqmvar + 0.00269056431155;
if not missing( V_N_ACADEMICO3 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO3 - 0.41395348837209 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO3 - 0.302950628104 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO3 - 0.42821782178217 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO3 - 0.34419337887545 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO3 - 0.43245125348189 )**2;
end;
else _vqmvar + 0.228416375075;
if not missing( V_N_ACADEMICO4 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO4 - 0.26046511627906 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO4 - 0.36167104878761 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO4 - 0.27103960396039 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO4 - 0.34576983709931 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO4 - 0.31058495821727 )**2;
end;
else _vqmvar + 0.22318488726772;
if not missing( V_N_ACADEMICO5 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO5 - 0.01395348837209 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO5 - 0.00642711072158 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO5 - 0.00618811881188 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO5 - 0.00945874934314 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO5 - 0.01323119777158 )**2;
end;
else _vqmvar + 0.0085333219799;
if not missing( V_N_ACADEMICO6 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO6 - 0.10232558139534 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO6 - 0.10984516505988 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO6 - 0.11881188118811 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO6 - 0.11245401996847 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO6 - 0.09958217270194 )**2;
end;
else _vqmvar + 0.09737600505887;
if not missing( V_N_ACADEMICO7 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO7 - 0.02790697674418 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO7 - 0.02804557405784 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO7 - 0.02722772277227 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO7 - 0.02364687335785 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO7 - 0.00974930362116 )**2;
end;
else _vqmvar + 0.02295712562897;
if not missing( V_N_ACADEMICO8 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO8 - 0.06976744186046 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO8 - 0.07099035933391 )**2;
```

```
   CLUSvads [3] + ( V_N_ACADEMICO8 - 0.04702970297029 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO8 - 0.0525486074619 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO8 - 0.03272980501392 )**2;
end;
else _vqmvar + 0.05367309785532;
if not missing( V_N_ACADEMICO9 ) then do;
   CLUSvads [1] + ( V_N_ACADEMICO9 - 0.03255813953488 )**2;
   CLUSvads [2] + ( V_N_ACADEMICO9 - 0.02921413964358 )**2;
   CLUSvads [3] + ( V_N_ACADEMICO9 - 0.01732673267326 )**2;
   CLUSvads [4] + ( V_N_ACADEMICO9 - 0.02417235943247 )**2;
   CLUSvads [5] + ( V_N_ACADEMICO9 - 0.01949860724233 )**2;
end;
else _vqmvar + 0.02442389606069;
if not missing( V_OCUPACION1 ) then do;
   CLUSvads [1] + ( V_OCUPACION1 - 0.8268156424581 )**2;
   CLUSvads [2] + ( V_OCUPACION1 - 0.11820330969267 )**2;
   CLUSvads [3] + ( V_OCUPACION1 - 0.8268710550045 )**2;
   CLUSvads [4] + ( V_OCUPACION1 - 0.12167300380228 )**2;
   CLUSvads [5] + ( V_OCUPACION1 - 0.12019914651493 )**2;
end;
else _vqmvar + 0.22015996347272;
if not missing( V_OCUPACION2 ) then do;
   CLUSvads [1] + ( V_OCUPACION2 - 0 )**2;
   CLUSvads [2] + ( V_OCUPACION2 - 0.00325059101654 )**2;
   CLUSvads [3] + ( V_OCUPACION2 - 0.00180342651036 )**2;
   CLUSvads [4] + ( V_OCUPACION2 - 0.00325909831613 )**2;
   CLUSvads [5] + ( V_OCUPACION2 - 0.00568990042674 )**2;
end;
else _vqmvar + 0.00308048998037;
if not missing( V_OCUPACION3 ) then do;
   CLUSvads [1] + ( V_OCUPACION3 - 0.00372439478584 )**2;
   CLUSvads [2] + ( V_OCUPACION3 - 0.00413711583924 )**2;
   CLUSvads [3] + ( V_OCUPACION3 - 0.00135256988277 )**2;
   CLUSvads [4] + ( V_OCUPACION3 - 0.00325909831613 )**2;
   CLUSvads [5] + ( V_OCUPACION3 - 0.00071123755334 )**2;
end;
else _vqmvar + 0.00276270408517;
if not missing( V_OCUPACION5 ) then do;
   CLUSvads [1] + ( V_OCUPACION5 - 0.00558659217877 )**2;
   CLUSvads [2] + ( V_OCUPACION5 - 0.02393617021276 )**2;
   CLUSvads [3] + ( V_OCUPACION5 - 0.01036970243462 )**2;
   CLUSvads [4] + ( V_OCUPACION5 - 0.03585008147745 )**2;
   CLUSvads [5] + ( V_OCUPACION5 - 0.01422475106685 )**2;
end;
else _vqmvar + 0.02014186785178;
if not missing( V_OCUPACION6 ) then do;
   CLUSvads [1] + ( V_OCUPACION6 - 0.024208566108 )**2;
   CLUSvads [2] + ( V_OCUPACION6 - 0.09574468085106 )**2;
   CLUSvads [3] + ( V_OCUPACION6 - 0.02434625788999 )**2;
   CLUSvads [4] + ( V_OCUPACION6 - 0.1059206952743 )**2;
   CLUSvads [5] + ( V_OCUPACION6 - 0.13371266002844 )**2;
end;
else _vqmvar + 0.0756711187873;
if not missing( V_OCUPACION7 ) then do;
   CLUSvads [1] + ( V_OCUPACION7 - 0.00372439478584 )**2;
   CLUSvads [2] + ( V_OCUPACION7 - 0.02777777777777 )**2;
   CLUSvads [3] + ( V_OCUPACION7 - 0.00811541929666 )**2;
   CLUSvads [4] + ( V_OCUPACION7 - 0.02335687126561 )**2;
   CLUSvads [5] + ( V_OCUPACION7 - 0.01706970128022 )**2;
```

```
end;
else _vqmvar + 0.01891418100684;
if not missing( V_OCUPACION8 ) then do;
   CLUSvads [1] + ( V_OCUPACION8 - 0.00744878957169 )**2;
   CLUSvads [2] + ( V_OCUPACION8 - 0.02689125295508 )**2;
   CLUSvads [3] + ( V_OCUPACION8 - 0.00180342651036 )**2;
   CLUSvads [4] + ( V_OCUPACION8 - 0.02281368821292 )**2;
   CLUSvads [5] + ( V_OCUPACION8 - 0.03911806543385 )**2;
end;
else _vqmvar + 0.02044827870798;
if not missing( V_OCUPACION9 ) then do;
   CLUSvads [1] + ( V_OCUPACION9 - 0 )**2;
   CLUSvads [2] + ( V_OCUPACION9 - 0.00147754137115 )**2;
   CLUSvads [3] + ( V_OCUPACION9 - 0.00135256988277 )**2;
   CLUSvads [4] + ( V_OCUPACION9 - 0.00054318305268 )**2;
   CLUSvads [5] + ( V_OCUPACION9 - 0 )**2;
end;
else _vqmvar + 0.00095805755202;
if not missing( V_OCUPACION10 ) then do;
   CLUSvads [1] + ( V_OCUPACION10 - 0.03351955307262 )**2;
   CLUSvads [2] + ( V_OCUPACION10 - 0.23729314420803 )**2;
   CLUSvads [3] + ( V_OCUPACION10 - 0.03832281334535 )**2;
   CLUSvads [4] + ( V_OCUPACION10 - 0.24660510592069 )**2;
   CLUSvads [5] + ( V_OCUPACION10 - 0.16927453769559 )**2;
end;
else _vqmvar + 0.14128234379386;
if not missing( V_OCUPACION11 ) then do;
   CLUSvads [1] + ( V_OCUPACION11 - 0.00186219739292 )**2;
   CLUSvads [2] + ( V_OCUPACION11 - 0.00531914893617 )**2;
   CLUSvads [3] + ( V_OCUPACION11 - 0.00180342651036 )**2;
   CLUSvads [4] + ( V_OCUPACION11 - 0.00760456273764 )**2;
   CLUSvads [5] + ( V_OCUPACION11 - 0.00213371266002 )**2;
end;
else _vqmvar + 0.00424395666995;
if not missing( V_OCUPACION12 ) then do;
   CLUSvads [1] + ( V_OCUPACION12 - 0.04096834264432 )**2;
   CLUSvads [2] + ( V_OCUPACION12 - 0.14243498817966 )**2;
   CLUSvads [3] + ( V_OCUPACION12 - 0.03742110009017 )**2;
   CLUSvads [4] + ( V_OCUPACION12 - 0.14828897338403 )**2;
   CLUSvads [5] + ( V_OCUPACION12 - 0.17211948790896 )**2;
end;
else _vqmvar + 0.10363509692225;
if not missing( V_OCUPACION13 ) then do;
   CLUSvads [1] + ( V_OCUPACION13 - 0.00558659217877 )**2;
   CLUSvads [2] + ( V_OCUPACION13 - 0.01832151300236 )**2;
   CLUSvads [3] + ( V_OCUPACION13 - 0.00901713255184 )**2;
   CLUSvads [4] + ( V_OCUPACION13 - 0.03639326453014 )**2;
   CLUSvads [5] + ( V_OCUPACION13 - 0.03413940256045 )**2;
end;
else _vqmvar + 0.02085650865087;
if not missing( V_OCUPACION14 ) then do;
   CLUSvads [1] + ( V_OCUPACION14 - 0.00372439478584 )**2;
   CLUSvads [2] + ( V_OCUPACION14 - 0.02925531914893 )**2;
   CLUSvads [3] + ( V_OCUPACION14 - 0.00495942290351 )**2;
   CLUSvads [4] + ( V_OCUPACION14 - 0.02824551873981 )**2;
   CLUSvads [5] + ( V_OCUPACION14 - 0.02347083926031 )**2;
end;
else _vqmvar + 0.0205503702507;
if not missing( V_OCUPACION16 ) then do;
```

```
   CLUSvads [1] + ( V_OCUPACION16 - 0.03538175046554 )**2;
   CLUSvads [2] + ( V_OCUPACION16 - 0.14479905437352 )**2;
   CLUSvads [3] + ( V_OCUPACION16 - 0.02434625788999 )**2;
   CLUSvads [4] + ( V_OCUPACION16 - 0.14991852254209 )**2;
   CLUSvads [5] + ( V_OCUPACION16 - 0.17994310099573 )**2;
end;
else _vqmvar + 0.10281863703646;
if not missing( V_OCUPACION17 ) then do;
   CLUSvads [1] + ( V_OCUPACION17 - 0 )**2;
   CLUSvads [2] + ( V_OCUPACION17 - 0.0056146572104 )**2;
   CLUSvads [3] + ( V_OCUPACION17 - 0 )**2;
   CLUSvads [4] + ( V_OCUPACION17 - 0.00488864747419 )**2;
   CLUSvads [5] + ( V_OCUPACION17 - 0.00711237553342 )**2;
end;
else _vqmvar + 0.00403262161386;
if not missing( V_OCUPACION18 ) then do;
   CLUSvads [1] + ( V_OCUPACION18 - 0.00372439478584 )**2;
   CLUSvads [2] + ( V_OCUPACION18 - 0.05555555555555 )**2;
   CLUSvads [3] + ( V_OCUPACION18 - 0.00495942290351 )**2;
   CLUSvads [4] + ( V_OCUPACION18 - 0.03150461705594 )**2;
   CLUSvads [5] + ( V_OCUPACION18 - 0.04836415362731 )**2;
end;
else _vqmvar + 0.03362894055134;
if not missing( V_OCUPACION19 ) then do;
   CLUSvads [1] + ( V_OCUPACION19 - 0 )**2;
   CLUSvads [2] + ( V_OCUPACION19 - 0 )**2;
   CLUSvads [3] + ( V_OCUPACION19 - 0 )**2;
   CLUSvads [4] + ( V_OCUPACION19 - 0.00162954915806 )**2;
   CLUSvads [5] + ( V_OCUPACION19 - 0 )**2;
end;
else _vqmvar + 0.00031955685935;
if not missing( V_OCUPACION20 ) then do;
   CLUSvads [1] + ( V_OCUPACION20 - 0 )**2;
   CLUSvads [2] + ( V_OCUPACION20 - 0.00413711583924 )**2;
   CLUSvads [3] + ( V_OCUPACION20 - 0 )**2;
   CLUSvads [4] + ( V_OCUPACION20 - 0.00434546442151 )**2;
   CLUSvads [5] + ( V_OCUPACION20 - 0.00213371266002 )**2;
end;
else _vqmvar + 0.0026567300441;
if not missing( V_OCUPACION21 ) then do;
   CLUSvads [1] + ( V_OCUPACION21 - 0.00372439478584 )**2;
   CLUSvads [2] + ( V_OCUPACION21 - 0.05585106382978 )**2;
   CLUSvads [3] + ( V_OCUPACION21 - 0.00315599639314 )**2;
   CLUSvads [4] + ( V_OCUPACION21 - 0.0239000543183 )**2;
   CLUSvads [5] + ( V_OCUPACION21 - 0.03058321479374 )**2;
end;
else _vqmvar + 0.02944551452809;
if not missing( V_SEGMENTO0 ) then do;
   CLUSvads [1] + ( V_SEGMENTO0 - 0.07029876977152 )**2;
   CLUSvads [2] + ( V_SEGMENTO0 - 0.0286541244573 )**2;
   CLUSvads [3] + ( V_SEGMENTO0 - 0.0250606305578 )**2;
   CLUSvads [4] + ( V_SEGMENTO0 - 0.07139134966128 )**2;
   CLUSvads [5] + ( V_SEGMENTO0 - 0.22168508287292 )**2;
end;
else _vqmvar + 0.06234566074281;
if not missing( V_SEGMENTO1 ) then do;
   CLUSvads [1] + ( V_SEGMENTO1 - 0.92970123022847 )**2;
   CLUSvads [2] + ( V_SEGMENTO1 - 0.97134587554269 )**2;
   CLUSvads [3] + ( V_SEGMENTO1 - 0.97493936944219 )**2;
```

```
   CLUSvads [4] + ( V_SEGMENTO1 - 0.92860865033871 )**2;
   CLUSvads [5] + ( V_SEGMENTO1 - 0.77831491712707 )**2;
end;
else _vqmvar + 0.06234566074281;
if not missing( V_SEXO1 ) then do;
   CLUSvads [1] + ( V_SEXO1 - 0.05993690851735 )**2;
   CLUSvads [2] + ( V_SEXO1 - 0.32494969818913 )**2;
   CLUSvads [3] + ( V_SEXO1 - 0.11080523055746 )**2;
   CLUSvads [4] + ( V_SEXO1 - 0.43227513227513 )**2;
   CLUSvads [5] + ( V_SEXO1 - 0.3472770323599 )**2;
end;
else _vqmvar + 0.21169319599211;
if not missing( V_SEXO2 ) then do;
   CLUSvads [1] + ( V_SEXO2 - 0.94006309148265 )**2;
   CLUSvads [2] + ( V_SEXO2 - 0.67505030181086 )**2;
   CLUSvads [3] + ( V_SEXO2 - 0.88919476944253 )**2;
   CLUSvads [4] + ( V_SEXO2 - 0.56772486772486 )**2;
   CLUSvads [5] + ( V_SEXO2 - 0.65272296764009 )**2;
end;
else _vqmvar + 0.21169319599211;
if not missing( V_ST_CIVIL1 ) then do;
   CLUSvads [1] + ( V_ST_CIVIL1 - 0.2394366197183 )**2;
   CLUSvads [2] + ( V_ST_CIVIL1 - 0.45854087313214 )**2;
   CLUSvads [3] + ( V_ST_CIVIL1 - 0.301664636622 )**2;
   CLUSvads [4] + ( V_ST_CIVIL1 - 0.51415094339622 )**2;
   CLUSvads [5] + ( V_ST_CIVIL1 - 0.4060984060984 )**2;
end;
else _vqmvar + 0.24183346931474;
if not missing( V_ST_CIVIL2 ) then do;
   CLUSvads [1] + ( V_ST_CIVIL2 - 0.13028169014084 )**2;
   CLUSvads [2] + ( V_ST_CIVIL2 - 0.25783767946088 )**2;
   CLUSvads [3] + ( V_ST_CIVIL2 - 0.20056841250507 )**2;
   CLUSvads [4] + ( V_ST_CIVIL2 - 0.26624737945492 )**2;
   CLUSvads [5] + ( V_ST_CIVIL2 - 0.22106722106722 )**2;
end;
else _vqmvar + 0.1783342261511;
if not missing( V_ST_CIVIL3 ) then do;
   CLUSvads [1] + ( V_ST_CIVIL3 - 0.61619718309859 )**2;
   CLUSvads [2] + ( V_ST_CIVIL3 - 0.20773513038382 )**2;
   CLUSvads [3] + ( V_ST_CIVIL3 - 0.45432399512789 )**2;
   CLUSvads [4] + ( V_ST_CIVIL3 - 0.17295597484276 )**2;
   CLUSvads [5] + ( V_ST_CIVIL3 - 0.32085932085932 )**2;
end;
else _vqmvar + 0.21133777459358;
if not missing( V_ST_CIVIL4 ) then do;
   CLUSvads [1] + ( V_ST_CIVIL4 - 0.00352112676056 )**2;
   CLUSvads [2] + ( V_ST_CIVIL4 - 0.00761793143861 )**2;
   CLUSvads [3] + ( V_ST_CIVIL4 - 0.00730816077953 )**2;
   CLUSvads [4] + ( V_ST_CIVIL4 - 0.0062893081761 )**2;
   CLUSvads [5] + ( V_ST_CIVIL4 - 0.01386001386001 )**2;
end;
else _vqmvar + 0.00790063985794;
if not missing( V_ST_CIVIL5 ) then do;
   CLUSvads [1] + ( V_ST_CIVIL5 - 0.00176056338028 )**2;
   CLUSvads [2] + ( V_ST_CIVIL5 - 0.00673893934954 )**2;
   CLUSvads [3] + ( V_ST_CIVIL5 - 0.00406008932196 )**2;
   CLUSvads [4] + ( V_ST_CIVIL5 - 0.0041928721174 )**2;
   CLUSvads [5] + ( V_ST_CIVIL5 - 0.00693000693 )**2;
end;
```

```
else _vqmvar + 0.00528118652094;
if not missing( V_ST_CIVIL6 ) then do;
   CLUSvads [1] + ( V_ST_CIVIL6 - 0.0088028169014 )**2;
   CLUSvads [2] + ( V_ST_CIVIL6 - 0.03984764137122 )**2;
   CLUSvads [3] + ( V_ST_CIVIL6 - 0.02436053593179 )**2;
   CLUSvads [4] + ( V_ST_CIVIL6 - 0.03616352201257 )**2;
   CLUSvads [5] + ( V_ST_CIVIL6 - 0.02979902979902 )**2;
end;
else _vqmvar + 0.03093711049977;
if not missing( V_ST_CIVIL7 ) then do;
   CLUSvads [1] + ( V_ST_CIVIL7 - 0 )**2;
   CLUSvads [2] + ( V_ST_CIVIL7 - 0.02168180486375 )**2;
   CLUSvads [3] + ( V_ST_CIVIL7 - 0.00771416971173 )**2;
   CLUSvads [4] + ( V_ST_CIVIL7 - 0 )**2;
   CLUSvads [5] + ( V_ST_CIVIL7 - 0.001386001386 )**2;
end;
else _vqmvar + 0.00960573938498;
if not missing( V_TERRITORIO1 ) then do;
   CLUSvads [1] + ( V_TERRITORIO1 - 0.00527240773286 )**2;
   CLUSvads [2] + ( V_TERRITORIO1 - 0.00028943560057 )**2;
   CLUSvads [3] + ( V_TERRITORIO1 - 0.00282942603071 )**2;
   CLUSvads [4] + ( V_TERRITORIO1 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO1 - 0.00069108500345 )**2;
end;
else _vqmvar + 0.00121518822466;
if not missing( V_TERRITORIO2 ) then do;
   CLUSvads [1] + ( V_TERRITORIO2 - 0.71177504393673 )**2;
   CLUSvads [2] + ( V_TERRITORIO2 - 0.0437047756874 )**2;
   CLUSvads [3] + ( V_TERRITORIO2 - 0.52303961196443 )**2;
   CLUSvads [4] + ( V_TERRITORIO2 - 0.00312662845231 )**2;
   CLUSvads [5] + ( V_TERRITORIO2 - 0.12715964063579 )**2;
end;
else _vqmvar + 0.16405781098864;
if not missing( V_TERRITORIO3 ) then do;
   CLUSvads [1] + ( V_TERRITORIO3 - 0 )**2;
   CLUSvads [2] + ( V_TERRITORIO3 - 0.19044862518089 )**2;
   CLUSvads [3] + ( V_TERRITORIO3 - 0.07841552142279 )**2;
   CLUSvads [4] + ( V_TERRITORIO3 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO3 - 0.03109882515549 )**2;
end;
else _vqmvar + 0.08267563059166;
if not missing( V_TERRITORIO4 ) then do;
   CLUSvads [1] + ( V_TERRITORIO4 - 0.00527240773286 )**2;
   CLUSvads [2] + ( V_TERRITORIO4 - 0.12966714905933 )**2;
   CLUSvads [3] + ( V_TERRITORIO4 - 0.01859337105901 )**2;
   CLUSvads [4] + ( V_TERRITORIO4 - 0.14695153725898 )**2;
   CLUSvads [5] + ( V_TERRITORIO4 - 0.10919143054595 )**2;
end;
else _vqmvar + 0.08597714684692;
if not missing( V_TERRITORIO6 ) then do;
   CLUSvads [1] + ( V_TERRITORIO6 - 0.08084358523725 )**2;
   CLUSvads [2] + ( V_TERRITORIO6 - 0.00607814761215 )**2;
   CLUSvads [3] + ( V_TERRITORIO6 - 0.09175424413904 )**2;
   CLUSvads [4] + ( V_TERRITORIO6 - 0.00156331422615 )**2;
   CLUSvads [5] + ( V_TERRITORIO6 - 0.01935038009675 )**2;
end;
else _vqmvar + 0.03186574768571;
if not missing( V_TERRITORIO8 ) then do;
   CLUSvads [1] + ( V_TERRITORIO8 - 0 )**2;
```

```
   CLUSvads [2] + ( V_TERRITORIO8 - 0.00028943560057 )**2;
   CLUSvads [3] + ( V_TERRITORIO8 - 0 )**2;
   CLUSvads [4] + ( V_TERRITORIO8 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO8 - 0.00207325501036 )**2;
end;
else _vqmvar + 0.00040539165973;
if not missing( V_TERRITORIO9 ) then do;
   CLUSvads [1] + ( V_TERRITORIO9 - 0.01933216168717 )**2;
   CLUSvads [2] + ( V_TERRITORIO9 - 0.00173661360347 )**2;
   CLUSvads [3] + ( V_TERRITORIO9 - 0.0125303152789 )**2;
   CLUSvads [4] + ( V_TERRITORIO9 - 0.00052110474205 )**2;
   CLUSvads [5] + ( V_TERRITORIO9 - 0.00138217000691 )**2;
end;
else _vqmvar + 0.00514410563404;
if not missing( V_TERRITORIO10 ) then do;
   CLUSvads [1] + ( V_TERRITORIO10 - 0.00175746924428 )**2;
   CLUSvads [2] + ( V_TERRITORIO10 - 0.00028943560057 )**2;
   CLUSvads [3] + ( V_TERRITORIO10 - 0.00282942603071 )**2;
   CLUSvads [4] + ( V_TERRITORIO10 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO10 - 0.00207325501036 )**2;
end;
else _vqmvar + 0.00121518822466;
if not missing( V_TERRITORIO11 ) then do;
   CLUSvads [1] + ( V_TERRITORIO11 - 0.03690685413005 )**2;
   CLUSvads [2] + ( V_TERRITORIO11 - 0.00723589001447 )**2;
   CLUSvads [3] + ( V_TERRITORIO11 - 0.0626515763945 )**2;
   CLUSvads [4] + ( V_TERRITORIO11 - 0.00156331422615 )**2;
   CLUSvads [5] + ( V_TERRITORIO11 - 0.01865929509329 )**2;
end;
else _vqmvar + 0.02287238451456;
if not missing( V_TERRITORIO12 ) then do;
   CLUSvads [1] + ( V_TERRITORIO12 - 0 )**2;
   CLUSvads [2] + ( V_TERRITORIO12 - 0.00028943560057 )**2;
   CLUSvads [3] + ( V_TERRITORIO12 - 0 )**2;
   CLUSvads [4] + ( V_TERRITORIO12 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO12 - 0.00414651002073 )**2;
end;
else _vqmvar + 0.00070921955198;
if not missing( V_TERRITORIO13 ) then do;
   CLUSvads [1] + ( V_TERRITORIO13 - 0.01757469244288 )**2;
   CLUSvads [2] + ( V_TERRITORIO13 - 0.07091172214182 )**2;
   CLUSvads [3] + ( V_TERRITORIO13 - 0.02142279708973 )**2;
   CLUSvads [4] + ( V_TERRITORIO13 - 0.14643043251693 )**2;
   CLUSvads [5] + ( V_TERRITORIO13 - 0.09744298548721 )**2;
end;
else _vqmvar + 0.06853647608078;
if not missing( V_TERRITORIO15 ) then do;
   CLUSvads [1] + ( V_TERRITORIO15 - 0.0140597539543 )**2;
   CLUSvads [2] + ( V_TERRITORIO15 - 0.08683068017366 )**2;
   CLUSvads [3] + ( V_TERRITORIO15 - 0.03233629749393 )**2;
   CLUSvads [4] + ( V_TERRITORIO15 - 0.15059927045336 )**2;
   CLUSvads [5] + ( V_TERRITORIO15 - 0.12370421561852 )**2;
end;
else _vqmvar + 0.0792574408899;
if not missing( V_TERRITORIO16 ) then do;
   CLUSvads [1] + ( V_TERRITORIO16 - 0.02811950790861 )**2;
   CLUSvads [2] + ( V_TERRITORIO16 - 0.00839363241678 )**2;
   CLUSvads [3] + ( V_TERRITORIO16 - 0.03152789005658 )**2;
   CLUSvads [4] + ( V_TERRITORIO16 - 0 )**2;
```

```
   CLUSvads [5] + ( V_TERRITORIO16 - 0.02764340013821 )**2;
end;
else _vqmvar + 0.01625331724265;
if not missing( V_TERRITORIO17 ) then do;
   CLUSvads [1] + ( V_TERRITORIO17 - 0 )**2;
   CLUSvads [2] + ( V_TERRITORIO17 - 0 )**2;
   CLUSvads [3] + ( V_TERRITORIO17 - 0.00202101859337 )**2;
   CLUSvads [4] + ( V_TERRITORIO17 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO17 - 0.00069108500345 )**2;
end;
else _vqmvar + 0.00060796414528;
if not missing( V_TERRITORIO19 ) then do;
   CLUSvads [1] + ( V_TERRITORIO19 - 0.00351493848857 )**2;
   CLUSvads [2] + ( V_TERRITORIO19 - 0.07293777134587 )**2;
   CLUSvads [3] + ( V_TERRITORIO19 - 0.01131770412287 )**2;
   CLUSvads [4] + ( V_TERRITORIO19 - 0.11776967170401 )**2;
   CLUSvads [5] + ( V_TERRITORIO19 - 0.07463718037318 )**2;
end;
else _vqmvar + 0.05855533049328;
if not missing( V_TERRITORIO20 ) then do;
   CLUSvads [1] + ( V_TERRITORIO20 - 0.00351493848857 )**2;
   CLUSvads [2] + ( V_TERRITORIO20 - 0.21939218523878 )**2;
   CLUSvads [3] + ( V_TERRITORIO20 - 0.02142279708973 )**2;
   CLUSvads [4] + ( V_TERRITORIO20 - 0.11099531005732 )**2;
   CLUSvads [5] + ( V_TERRITORIO20 - 0.16033172080165 )**2;
end;
else _vqmvar + 0.11128070954888;
if not missing( V_TERRITORIO21 ) then do;
   CLUSvads [1] + ( V_TERRITORIO21 - 0.01230228471001 )**2;
   CLUSvads [2] + ( V_TERRITORIO21 - 0.05441389290882 )**2;
   CLUSvads [3] + ( V_TERRITORIO21 - 0.03314470493128 )**2;
   CLUSvads [4] + ( V_TERRITORIO21 - 0.14695153725898 )**2;
   CLUSvads [5] + ( V_TERRITORIO21 - 0.05597788527988 )**2;
end;
else _vqmvar + 0.06067882627849;
if not missing( V_TERRITORIO22 ) then do;
   CLUSvads [1] + ( V_TERRITORIO22 - 0.02108963093145 )**2;
   CLUSvads [2] + ( V_TERRITORIO22 - 0.06714905933429 )**2;
   CLUSvads [3] + ( V_TERRITORIO22 - 0.02223120452708 )**2;
   CLUSvads [4] + ( V_TERRITORIO22 - 0.06982803543512 )**2;
   CLUSvads [5] + ( V_TERRITORIO22 - 0.05805114029025 )**2;
end;
else _vqmvar + 0.04967074673225;
if not missing( V_TERRITORIO23 ) then do;
   CLUSvads [1] + ( V_TERRITORIO23 - 0 )**2;
   CLUSvads [2] + ( V_TERRITORIO23 - 0 )**2;
   CLUSvads [3] + ( V_TERRITORIO23 - 0.00565885206143 )**2;
   CLUSvads [4] + ( V_TERRITORIO23 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO23 - 0.00483759502418 )**2;
end;
else _vqmvar + 0.00212463672014;
if not missing( V_TERRITORIO24 ) then do;
   CLUSvads [1] + ( V_TERRITORIO24 - 0.02987697715289 )**2;
   CLUSvads [2] + ( V_TERRITORIO24 - 0 )**2;
   CLUSvads [3] + ( V_TERRITORIO24 - 0.01131770412287 )**2;
   CLUSvads [4] + ( V_TERRITORIO24 - 0 )**2;
   CLUSvads [5] + ( V_TERRITORIO24 - 0.00414651002073 )**2;
end;
else _vqmvar + 0.00514410563404;
```

```
if not missing( V_TERRITORIO25 ) then do;
   CLUSvads [1] + ( V_TERRITORIO25 - 0.00878734622144 )**2;
   CLUSvads [2] + ( V_TERRITORIO25 - 0.03994211287988 )**2;
   CLUSvads [3] + ( V_TERRITORIO25 - 0.01495553759094 )**2;
   CLUSvads [4] + ( V_TERRITORIO25 - 0.10369984366857 )**2;
   CLUSvads [5] + ( V_TERRITORIO25 - 0.07671043538355 )**2;
end;
else _vqmvar + 0.04721271031953;
if not missing( V_TOT_CANALES_ACTI0 ) then do;
   CLUSvads [1] + ( V_TOT_CANALES_ACTI0 - 0.00351493848857 )**2;
   CLUSvads [2] + ( V_TOT_CANALES_ACTI0 - 0 )**2;
   CLUSvads [3] + ( V_TOT_CANALES_ACTI0 - 0.00727566693613 )**2;
   CLUSvads [4] + ( V_TOT_CANALES_ACTI0 - 0 )**2;
   CLUSvads [5] + ( V_TOT_CANALES_ACTI0 - 0 )**2;
end;
else _vqmvar + 0.00202346437654;
if not missing( V_TOT_CANALES_ACTI1 ) then do;
   CLUSvads [1] + ( V_TOT_CANALES_ACTI1 - 0.86818980667838 )**2;
   CLUSvads [2] + ( V_TOT_CANALES_ACTI1 - 0.01331403762662 )**2;
   CLUSvads [3] + ( V_TOT_CANALES_ACTI1 - 0.92077607113985 )**2;
   CLUSvads [4] + ( V_TOT_CANALES_ACTI1 - 0.02292860865033 )**2;
   CLUSvads [5] + ( V_TOT_CANALES_ACTI1 - 0.00621546961325 )**2;
end;
else _vqmvar + 0.20635199277842;
if not missing( V_TOT_CANALES_ACTI2 ) then do;
   CLUSvads [1] + ( V_TOT_CANALES_ACTI2 - 0.09666080843585 )**2;
   CLUSvads [2] + ( V_TOT_CANALES_ACTI2 - 0.66657018813314 )**2;
   CLUSvads [3] + ( V_TOT_CANALES_ACTI2 - 0.04769603880355 )**2;
   CLUSvads [4] + ( V_TOT_CANALES_ACTI2 - 0.50547159979155 )**2;
   CLUSvads [5] + ( V_TOT_CANALES_ACTI2 - 0.4592541436464 )**2;
end;
else _vqmvar + 0.24309004899476;
if not missing( V_TOT_CANALES_ACTI3 ) then do;
   CLUSvads [1] + ( V_TOT_CANALES_ACTI3 - 0.02460456942003 )**2;
   CLUSvads [2] + ( V_TOT_CANALES_ACTI3 - 0.178002894356 )**2;
   CLUSvads [3] + ( V_TOT_CANALES_ACTI3 - 0.02061438965238 )**2;
   CLUSvads [4] + ( V_TOT_CANALES_ACTI3 - 0.22772277227722 )**2;
   CLUSvads [5] + ( V_TOT_CANALES_ACTI3 - 0.27831491712707 )**2;
end;
else _vqmvar + 0.13035262335116;
if not missing( V_TOT_CANALES_ACTI4 ) then do;
   CLUSvads [1] + ( V_TOT_CANALES_ACTI4 - 0.00702987697715 )**2;
   CLUSvads [2] + ( V_TOT_CANALES_ACTI4 - 0.14211287988422 )**2;
   CLUSvads [3] + ( V_TOT_CANALES_ACTI4 - 0.00363783346806 )**2;
   CLUSvads [4] + ( V_TOT_CANALES_ACTI4 - 0.24387701928087 )**2;
   CLUSvads [5] + ( V_TOT_CANALES_ACTI4 - 0.25621546961325 )**2;
end;
else _vqmvar + 0.11761626647494;
if not missing( V_WELCOME_CALL0 ) then do;
   CLUSvads [1] + ( V_WELCOME_CALL0 - 0.5360281195079 )**2;
   CLUSvads [2] + ( V_WELCOME_CALL0 - 0.56758321273516 )**2;
   CLUSvads [3] + ( V_WELCOME_CALL0 - 0.5626515763945 )**2;
   CLUSvads [4] + ( V_WELCOME_CALL0 - 0.85669619593538 )**2;
   CLUSvads [5] + ( V_WELCOME_CALL0 - 0.6042817679558 )**2;
end;
else _vqmvar + 0.23410913512466;
if not missing( V_WELCOME_CALL1 ) then do;
   CLUSvads [1] + ( V_WELCOME_CALL1 - 0.46397188049209 )**2;
   CLUSvads [2] + ( V_WELCOME_CALL1 - 0.43241678726483 )**2;
```

```
   CLUSvads [3] + ( V_WELCOME_CALL1 - 0.43734842360549 )**2;
   CLUSvads [4] + ( V_WELCOME_CALL1 - 0.14330380406461 )**2;
   CLUSvads [5] + ( V_WELCOME_CALL1 - 0.39571823204419 )**2;
end;
else _vqmvar + 0.23410913512466;
_vqnvar = 9.92816331750209 - _vqmvar;
if _vqnvar <= 1.4673119395357E-10 then do;
   _SEGMENT_ = .; Distance = .;
end;
else do;
   _SEGMENT_ = 1; Distance = CLUSvads [1];
   _vqfzdst = Distance * 0.99999999999988; drop _vqfzdst;
   do _vqclus = 2 to 5;
      if CLUSvads [_vqclus] < _vqfzdst then do;
         _SEGMENT_ = _vqclus; Distance = CLUSvads [_vqclus];
         _vqfzdst = Distance * 0.99999999999988;
      end;
   end;
   Distance = sqrt(Distance * (9.92816331750209 / _vqnvar));
end;
CLUSvlex :;

*****************************************;
*** End Scoring Code from PROC DMVQ ***;
*****************************************;
*-------------------------------------------------------------*;
* Clus: Creating Segment Label;
*-------------------------------------------------------------*;
length _SEGMENT_LABEL_ $80;
label _SEGMENT_LABEL_='Segment Description';
if _SEGMENT_ = 1 then _SEGMENT_LABEL_="Cluster1";
else
if _SEGMENT_ = 2 then _SEGMENT_LABEL_="Cluster2";
else
if _SEGMENT_ = 3 then _SEGMENT_LABEL_="Cluster3";
else
if _SEGMENT_ = 4 then _SEGMENT_LABEL_="Cluster4";
else
if _SEGMENT_ = 5 then _SEGMENT_LABEL_="Cluster5";
*-------------------------------------------------------------*;
* TOOL: Score Node;
* TYPE: ASSESS;
* NODE: Score;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* Score: Creating Fixed Names;
*-------------------------------------------------------------*;
LABEL EM_SEGMENT = 'Segment Variable';
EM_SEGMENT = _SEGMENT_;
;RUN;

DATA Y;
SET X;
KEEP V_VM_TO_R_SALDO1000_5
V_VM_OO_R_CTAS_AB0_5
V_VM_TO_R_LC_SUM5000_4
V_VM_CH_P0_4
V_VM_CH_P100_4
V_ATM
```

```
V_NINC_CH_P_6
V_TOT_CANALES_ACTI
V_NDEC_CH_P_6
V_ENTREGAT
V_NDEC_TO_R_SALDO_4
V_VM_B_D_TO45_5
V_MEDIA_CH_P_5
V_MOB_TDC
V_TERRITORIO
V_VM_TO_R_LC_MAX25000_4
V_VM_TO_SALDO10000_5
V_VM_TO_CTAS_AB2_5
V_NDEC_TO_SALDO_5
ID
TARGET
EM_SEGMENT;
RUN;

DATA Y1;
 SET Y;
RENAME V_VM_TO_R_SALDO1000_5=X1
 V_VM_OO_R_CTAS_AB0_5=X2
 V_VM_TO_R_LC_SUM5000_4=X3
 V_VM_CH_P0_4=X4
 V_VM_CH_P100_4=X5
 V_ATM=X6
 V_NINC_CH_P_6=X7
 V_TOT_CANALES_ACTI=X8
 V_NDEC_CH_P_6=X9
 V_ENTREGAT=X10
 V_NDEC_TO_R_SALDO_4=X11
 V_VM_B_D_TO45_5=X12
 V_MEDIA_CH_P_5=X13
 V_MOB_TDC=X14
 V_TERRITORIO=X15
 V_VM_TO_R_LC_MAX25000_4=X16
 V_VM_TO_SALDO10000_5=X17
 V_VM_TO_CTAS_AB2_5=X18
 V_NDEC_TO_SALDO_5=X19;
RUN;
 %MACRO AMV_CLUS;
 %DO i=1 %to 19;
 /* ---------------------------------------------------------------------
    Code generated by SAS Task

    Generated on: Sunday, April 01, 2018 at 11:55:59 PM
    By task: One-Way ANOVA (5)

    Input Data: SASApp:WORK.Y1
    Server:  SASApp
    --------------------------------------------------------------------- */
ODS GRAPHICS ON;

 %eg_conditional_dropds(WORK.TMP0TempTableInput,
                   WORK.TMP1TempTableTemporaryOutput,
                   WORK.TMPPlotDS);
 /* ---------------------------------------------------------------------
    Sort data set SASApp:WORK.Y1
    --------------------------------------------------------------------- */
```

```
PROC SQL;
        CREATE VIEW WORK.TMP0TempTableInput AS
                SELECT T.X&i., T.EM_SEGMENT
        FROM WORK.Y1 as T
;
QUIT;

TITLE;
TITLE1 "One-Way Analysis of Variance";
TITLE2 "Results";

FOOTNOTE;
FOOTNOTE1 "Generated by the SAS System (&_SASSERVERNAME, &SYSSCPL)
on %TRIM(%QSYSFUNC(DATE(), NLDATE20.)) at %TRIM(%SYSFUNC(TIME(), TIMEAMPM12.))";

ODS EXCLUDE BoxPlot;
/* ---------------------------------------------------------------------
   Run PROC ANOVA to perform the analysis.
   --------------------------------------------------------------------- */
PROC ANOVA DATA=WORK.TMP0TempTableInput ;
        CLASS EM_SEGMENT;
        MODEL X&i. = EM_SEGMENT ;

        MEANS EM_SEGMENT / DUNCAN ALPHA=0.05 ;
RUN; QUIT;
/* ---------------------------------------------------------------------
   End of task code
   --------------------------------------------------------------------- */
RUN; QUIT;
%_eg_conditional_dropds(WORK.TMP0TempTableInput,
                WORK.TMP1TempTableTemporaryOutput,
                WORK.TMPPlotDS);
TITLE; FOOTNOTE;
ODS GRAPHICS OFF;

%end;
%mend;
%AMV_CLUS;
```

Listing 3: Árbol de desición en SAS

```
DATA X;
SET AAOK.MDL_CLUST;
*-------------------------------------------------------------------*;
* EM SCORE CODE;
* EM Version: 13.2;
* SAS Release: 9.04.01M2P072314;
* Host: uxe25102;
* Encoding: latin1;
* Locale: en_US;
* Project Path: /herramientas/SAS/OKY;
* Project Name: SEG_OC;
* Diagram Id: EMWS2;
* Diagram Name: ARBOL_OC;
* Generated by: A3725988;
* Date: 26APR2018:00:29:03;
*-------------------------------------------------------------------*;
*-------------------------------------------------------------------*;
```

```
* TOOL: Input Data Source;
* TYPE: SAMPLE;
* NODE: Ids;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Partition Class;
* TYPE: SAMPLE;
* NODE: Part;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Extension Class;
* TYPE: MODEL;
* NODE: Tree2;
*-------------------------------------------------------------*;
************************************************************;
******                DECISION TREE SCORING CODE          ******;
************************************************************;

******          LENGTHS OF NEW CHARACTER VARIABLES        ******;
LENGTH I_TARGET  $    12;
LENGTH _WARN_  $     4;

******               LABELS FOR NEW VARIABLES             ******;
LABEL _NODE_  = 'Node' ;
LABEL _LEAF_  = 'Leaf' ;
LABEL P_TARGET1  = 'Predicted: TARGET=1' ;
LABEL P_TARGET0  = 'Predicted: TARGET=0' ;
LABEL Q_TARGET1  = 'Unadjusted P: TARGET=1' ;
LABEL Q_TARGET0  = 'Unadjusted P: TARGET=0' ;
LABEL V_TARGET1  = 'Validated: TARGET=1' ;
LABEL V_TARGET0  = 'Validated: TARGET=0' ;
LABEL I_TARGET  = 'Into: TARGET' ;
LABEL U_TARGET  = 'Unnormalized Into: TARGET' ;
LABEL _WARN_  = 'Warnings' ;


******      TEMPORARY VARIABLES FOR FORMATTED VALUES      ******;
LENGTH _ARBFMT_12 $      12; DROP _ARBFMT_12;
_ARBFMT_12 = ' '; /* Initialize to avoid warning. */


******               ASSIGN OBSERVATION TO NODE           ******;
IF  NOT MISSING(V_MO_R__SALDO ) AND
  V_MO_R__SALDO  <                    0.5 THEN DO;
  _ARBFMT_12 = PUT( V_TERRITORIO , BEST12.);
   %DMNORMIP( _ARBFMT_12);
  IF  _ARBFMT_12 IN ('3' ) THEN DO;
    IF  NOT MISSING(V_TOT_CRED_BCO ) AND
      V_TOT_CRED_BCO  <                    1.5 THEN DO;
      IF  NOT MISSING(V_NINC_OO_R_SALDO_5 ) AND
                          0.5 <= V_NINC_OO_R_SALDO_5  THEN DO;
        _NODE_  =                    30;
        _LEAF_  =                     2;
        P_TARGET1  =      0.95886777488937;
        P_TARGET0  =      0.04113222511062;
        Q_TARGET1  =      0.85714285714285;
        Q_TARGET0  =      0.14285714285714;
        V_TARGET1  =      0.88604054619884;
        V_TARGET0  =      0.11395945380115;
```

```
        I_TARGET  = '1' ;
        U_TARGET  =                    1;
        END;
      ELSE DO;
        _NODE_  =                   29;
        _LEAF_  =                    1;
        P_TARGET1  =                    1;
        P_TARGET0  =                    0;
        Q_TARGET1  =                    1;
        Q_TARGET0  =                    0;
        V_TARGET1  =                    1;
        V_TARGET0  =                    0;
        I_TARGET  = '1' ;
        U_TARGET  =                    1;
        END;
      END;
    ELSE DO;
      _NODE_  =                   14;
      _LEAF_  =                    3;
      P_TARGET1  =      0.89382219046027;
      P_TARGET0  =      0.10617780953972;
      Q_TARGET1  =      0.68421052631578;
      Q_TARGET0  =      0.31578947368421;
      V_TARGET1  =                    1;
      V_TARGET0  =                    0;
      I_TARGET  = '1' ;
      U_TARGET  =                    1;
      END;
    END;
  ELSE IF _ARBFMT_12 IN ('6' ,'23' ,'9' ,'16' ,'2' ,'24' ) THEN DO;
    _ARBFMT_12 = PUT( V_ENTREGAT , BEST12.);
     %DMNORMIP( _ARBFMT_12);
    IF _ARBFMT_12 IN ('1' ) THEN DO;
      _NODE_  =                   17;
      _LEAF_  =                    8;
      P_TARGET1  =      0.75658706366849;
      P_TARGET0  =       0.2434129363315;
      Q_TARGET1  =      0.44444444444444;
      Q_TARGET0  =      0.55555555555555;
      V_TARGET1  =       0.7953974834321;
      V_TARGET0  =      0.20460251656789;
      I_TARGET  = '1' ;
      U_TARGET  =                    1;
      END;
    ELSE DO;
      IF  NOT MISSING(V_MO_R__CTAS_AB ) AND
        V_MO_R__CTAS_AB  <                  0.5 THEN DO;
        _NODE_  =                   35;
        _LEAF_  =                    9;
        P_TARGET1  =      0.38889738497055;
        P_TARGET0  =      0.61110261502944;
        Q_TARGET1  =      0.14074074074074;
        Q_TARGET0  =      0.85925925925925;
        V_TARGET1  =      0.43741370452429;
        V_TARGET0  =       0.5625862954757;
        I_TARGET  = '0' ;
        U_TARGET  =                    0;
        END;
      ELSE DO;
```

```
        _NODE_ =                    36;
        _LEAF_ =                    10;
        P_TARGET1 =     0.14259879721601;
        P_TARGET0 =     0.85740120278398;
        Q_TARGET1 =     0.04104903078677;
        Q_TARGET0 =     0.95895096921322;
        V_TARGET1 =     0.12418545753069;
        V_TARGET0 =      0.8758145424693;
        I_TARGET = '0' ;
        U_TARGET =                     0;
       END;
     END;
   END;
ELSE DO;
   _ARBFMT_12 = PUT( V_ENTREGAT , BEST12.);
    %DMNORMIP( _ARBFMT_12);
   IF _ARBFMT_12 IN ('1' ) THEN DO;
     IF  NOT MISSING(V_MO_R__CTAS_AB ) AND
       V_MO_R__CTAS_AB <                  0.5 THEN DO;
        _NODE_ =                    62;
        _LEAF_ =                     4;
        P_TARGET1 =     0.95254835018477;
        P_TARGET0 =     0.04745164981522;
        Q_TARGET1 =     0.83783783783783;
        Q_TARGET0 =     0.16216216216216;
        V_TARGET1 =     0.94775826885865;
        V_TARGET0 =     0.05224173114134;
        I_TARGET = '1' ;
        U_TARGET =                     1;
       END;
     ELSE DO;
        _NODE_ =                    63;
        _LEAF_ =                     5;
        P_TARGET1 =     0.77517418305405;
        P_TARGET0 =     0.22482581694594;
        Q_TARGET1 =     0.47017543859649;
        Q_TARGET0 =      0.5298245614035;
        V_TARGET1 =     0.71480971969802;
        V_TARGET0 =     0.28519028030197;
        I_TARGET = '1' ;
        U_TARGET =                     1;
       END;
     END;
   ELSE DO;
     IF  NOT MISSING(V_MO_R__CTAS_AB ) AND
       V_MO_R__CTAS_AB <                  0.5 THEN DO;
        _NODE_ =                    33;
        _LEAF_ =                     6;
        P_TARGET1 =     0.59790413168646;
        P_TARGET0 =     0.40209586831353;
        Q_TARGET1 =     0.27678571428571;
        Q_TARGET0 =     0.72321428571428;
        V_TARGET1 =     0.61297413660861;
        V_TARGET0 =     0.38702586339138;
        I_TARGET = '1' ;
        U_TARGET =                     1;
       END;
     ELSE DO;
        _NODE_ =                    34;
```

```
          _LEAF_   =                    7;
          P_TARGET1  =     0.27297619651414;
          P_TARGET0  =     0.72702380348585;
          Q_TARGET1  =     0.08812260536398;
          Q_TARGET0  =     0.91187739463601;
          V_TARGET1  =     0.20249774412744;
          V_TARGET0  =     0.79750225587255;
          I_TARGET  = '0' ;
          U_TARGET  =                    0;
         END;
       END;
     END;
   END;
ELSE IF  NOT MISSING(V_MO_R__SALDO ) AND
                    0.5 <= V_MO_R__SALDO  THEN DO;
   IF  NOT MISSING(V_VM_TO_R_LC_SUM5000_4 ) AND
                      1.5 <= V_VM_TO_R_LC_SUM5000_4  AND
     V_VM_TO_R_LC_SUM5000_4  <                3.5 THEN DO;
     IF  NOT MISSING(V_NINC_OO_R_SALDO_5 ) AND
       V_NINC_OO_R_SALDO_5  <                0.5 THEN DO;
       IF  NOT MISSING(V_ANT_INT_ANTIG ) AND
                        11.5 <= V_ANT_INT_ANTIG  THEN DO;
         _NODE_   =                   44;
         _LEAF_   =                   15;
         P_TARGET1  =     0.95396294755114;
         P_TARGET0  =     0.04603705244885;
         Q_TARGET1  =     0.84210526315789;
         Q_TARGET0  =      0.1578947368421;
         V_TARGET1  =     0.95889028855841;
         V_TARGET0  =     0.04110971144158;
         I_TARGET  = '1' ;
         U_TARGET  =                    1;
         END;
       ELSE DO;
         _NODE_   =                   43;
         _LEAF_   =                   14;
         P_TARGET1  =      0.6309329554935;
         P_TARGET0  =     0.36906704450649;
         Q_TARGET1  =     0.30555555555555;
         Q_TARGET0  =     0.69444444444444;
         V_TARGET1  =     0.68958011348626;
         V_TARGET0  =     0.31041988651373;
         I_TARGET  = '1' ;
         U_TARGET  =                    1;
         END;
       END;
     ELSE DO;
       IF  NOT MISSING(V_VM_TO_SALDO10000_5 ) AND
         V_VM_TO_SALDO10000_5  <                0.5 THEN DO;
         _NODE_   =                   45;
         _LEAF_   =                   16;
         P_TARGET1  =      0.8792992028378;
         P_TARGET0  =     0.12070079716219;
         Q_TARGET1  =     0.65217391304347;
         Q_TARGET0  =     0.34782608695652;
         V_TARGET1  =     0.88604054619884;
         V_TARGET0  =     0.11395945380115;
         I_TARGET  = '1' ;
         U_TARGET  =                    1;
```

```
      END;
    ELSE IF  NOT MISSING(V_VM_TO_SALDO10000_5 ) AND
                          0.5 <= V_VM_TO_SALDO10000_5  AND
      V_VM_TO_SALDO10000_5  <                      1.5 THEN DO;
      _NODE_  =                    46;
      _LEAF_  =                    17;
      P_TARGET1  =     0.39303853804654;
      P_TARGET0  =     0.60696146195345;
      Q_TARGET1  =     0.14285714285714;
      Q_TARGET0  =     0.85714285714285;
      V_TARGET1  =                   0;
      V_TARGET0  =                   1;
      I_TARGET  = '0' ;
      U_TARGET  =                  0;
      END;
    ELSE DO;
      _NODE_  =                    47;
      _LEAF_  =                    18;
      P_TARGET1  =     0.93577350696475;
      P_TARGET0  =     0.06422649303524;
      Q_TARGET1  =     0.78947368421052;
      Q_TARGET0  =     0.21052631578947;
      V_TARGET1  =     0.88604054619884;
      V_TARGET0  =     0.11395945380115;
      I_TARGET  = '1' ;
      U_TARGET  =                  1;
      END;
    END;
  END;
ELSE IF  NOT MISSING(V_VM_TO_R_LC_SUM5000_4 ) AND
                      3.5 <= V_VM_TO_R_LC_SUM5000_4  THEN DO;
  IF  NOT MISSING(V_MOB_TDC ) AND
                        9.5 <= V_MOB_TDC  THEN DO;
    _NODE_  =                    70;
    _LEAF_  =                    20;
    P_TARGET1  =     0.26875407056159;
    P_TARGET0  =      0.7312459294384;
    Q_TARGET1  =     0.08641975308641;
    Q_TARGET0  =     0.91358024691358;
    V_TARGET1  =     0.70699807069829;
    V_TARGET0  =      0.2930019293017;
    I_TARGET  = '0' ;
    U_TARGET  =                  0;
    END;
  ELSE DO;
    _NODE_  =                    69;
    _LEAF_  =                    19;
    P_TARGET1  =     0.85602594957065;
    P_TARGET0  =     0.14397405042934;
    Q_TARGET1  =     0.60479041916167;
    Q_TARGET0  =     0.39520958083832;
    V_TARGET1  =     0.85218413637028;
    V_TARGET0  =     0.14781586362971;
    I_TARGET  = '1' ;
    U_TARGET  =                  1;
    END;
  END;
ELSE DO;
  IF  NOT MISSING(V_MO_R__CTAS_AB ) AND
```

```
                        1.5 <= V_MO_R__CTAS_AB  AND
      V_MO_R__CTAS_AB  <                    2.5 THEN DO;
      _NODE_  =                    66;
      _LEAF_  =                    12;
      P_TARGET1  =      0.9556805370259;
      P_TARGET0  =     0.04431946297409;
      Q_TARGET1  =      0.8473282442748;
      Q_TARGET0  =     0.15267175572519;
      V_TARGET1  =     0.95511485623788;
      V_TARGET0  =     0.04488514376211;
      I_TARGET  = '1' ;
      U_TARGET  =                    1;
      END;
    ELSE IF  NOT MISSING(V_MO_R__CTAS_AB ) AND
                        2.5 <= V_MO_R__CTAS_AB  THEN DO;
      _NODE_  =                    67;
      _LEAF_  =                    13;
      P_TARGET1  =     0.79530454666478;
      P_TARGET0  =     0.20469545333521;
      Q_TARGET1  =                  0.5;
      Q_TARGET0  =                  0.5;
      V_TARGET1  =     0.87496164057036;
      V_TARGET0  =     0.12503835942963;
      I_TARGET  = '1' ;
      U_TARGET  =                    1;
      END;
    ELSE DO;
      _NODE_  =                    65;
      _LEAF_  =                    11;
      P_TARGET1  =     0.98124198656967;
      P_TARGET0  =     0.01875801343032;
      Q_TARGET1  =     0.93086133642332;
      Q_TARGET0  =     0.06913866357667;
      V_TARGET1  =     0.98142632960674;
      V_TARGET0  =     0.01857367039325;
      I_TARGET  = '1' ;
      U_TARGET  =                    1;
      END;
    END;
  END;
ELSE DO;
  IF  NOT MISSING(V_TOT_CRED_BCO ) AND
    V_TOT_CRED_BCO  <                  0.5 THEN DO;
    IF  NOT MISSING(V_VM_CH_P0_4 ) AND
                        2.5 <= V_VM_CH_P0_4  THEN DO;
      _NODE_  =                    73;
      _LEAF_  =                    22;
      P_TARGET1  =     0.52286790132771;
      P_TARGET0  =     0.47713209867228;
      Q_TARGET1  =                 0.22;
      Q_TARGET0  =                 0.78;
      V_TARGET1  =     0.75146825592277;
      V_TARGET0  =     0.24853174407722;
      I_TARGET  = '1' ;
      U_TARGET  =                    1;
      END;
    ELSE DO;
      _NODE_  =                    72;
      _LEAF_  =                    21;
```

```
      P_TARGET1  =      0.79817089672697;
      P_TARGET0  =      0.20182910327302;
      Q_TARGET1  =      0.50442477876106;
      Q_TARGET0  =      0.49557522123893;
      V_TARGET1  =      0.88279529234934;
      V_TARGET0  =      0.11720470765065;
      I_TARGET  = '1' ;
      U_TARGET  =                     1;
     END;
   END;
  ELSE DO;
    _ARBFMT_12 = PUT( V_ENTREGAT , BEST12.);
     %DMNORMIP( _ARBFMT_12);
    IF _ARBFMT_12 IN ('1' ) THEN DO;
      _NODE_  =                    27;
      _LEAF_  =                    23;
      P_TARGET1  =      0.75658706366849;
      P_TARGET0  =       0.2434129363315;
      Q_TARGET1  =      0.44444444444444;
      Q_TARGET0  =      0.55555555555555;
      V_TARGET1  =                     1;
      V_TARGET0  =                     0;
      I_TARGET  = '1' ;
      U_TARGET  =                     1;
     END;
    ELSE DO;
      _NODE_  =                    28;
      _LEAF_  =                    24;
      P_TARGET1  =      0.23009984403051;
      P_TARGET0  =      0.76990015596948;
      Q_TARGET1  =      0.07142857142857;
      Q_TARGET0  =      0.92857142857142;
      V_TARGET1  =      0.23966212124988;
      V_TARGET0  =      0.76033787875011;
      I_TARGET  = '0' ;
      U_TARGET  =                     0;
     END;
    END;
  END;

*************************************************************;
******            END OF DECISION TREE SCORING CODE          ******;
*************************************************************;

drop _LEAF_;
*-------------------------------------------------------------*;
* TOOL: Score Node;
* TYPE: ASSESS;
* NODE: Score;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* Score: Creating Fixed Names;
*-------------------------------------------------------------*;
LABEL EM_SEGMENT = 'Node';
EM_SEGMENT = _NODE_;
LABEL EM_EVENTPROBABILITY = 'Probability for level 1 of TARGET';
EM_EVENTPROBABILITY = P_TARGET1;
LABEL EM_PROBABILITY = 'Probability of Classification';
EM_PROBABILITY =
```

```
max(
P_TARGET1
,
P_TARGET0
);
LENGTH EM_CLASSIFICATION $%dmnorlen;
LABEL EM_CLASSIFICATION = "Prediction for TARGET";
EM_CLASSIFICATION = I_TARGET;
;RUN;

PROC SQL;
CREATE TABLE CUB AS
SELECT _NODE_, TARGET, COUNT(*) AS CASOS
FROM X
GROUP BY 1, 2;
QUIT;
```

Listing 4: Regresión logística en SAS

```
DATA X;
SET AAOK.MDL_CLUST;
RUN;

PROC LOGISTIC DATA=WORK.SORTTEMPTABLESORTED_0000
                    PLOTS(ONLY)=ALL
          ;
          MODEL TARGET (Event = '0')=V_NINC_CH_P_3 V_NDEC_ATM_3
     V_NDEC_OO_R_LC_SUM_3 V_NDEC_TO_R_LC_SUM_3 V_NINC_B_D_OO_4
     V_NINC_OO_R_LC_SUM_4 V_NINC_TO_R_LC_SUM_4 V_NDEC_B_D_OO_4
     V_NDEC_TO_R_SALDO_4 V_VM_CH_P0_4 V_VM_CH_P100_4
     V_VM_OO_R_CTAS_AB1_4 V_VM_TO_R_LC_SUM5000_4 V_VM_CH_P1500_4
     V_VM_OO_R_LC_MAX25000_4 V_VM_OO_R_SALDO25000_4
     V_VM_TO_R_LC_MAX25000_4 V_VM_TO_R_SALDO25000_4 V_NINC_ATM_5
     V_NINC_B_D_TO_5 V_NINC_OO_R_SALDO_5 V_NINC_TO_SALDO_5 V_NDEC_ATM_5
     V_NDEC_OO_R_CTAS_AB_5 V_NDEC_OO_R_LC_MAX_5 V_NDEC_TO_CTAS_AB_5
     V_NDEC_TO_R_LC_MAX_5 V_NDEC_TO_SALDO_5 V_VM_OO_R_CTAS_AB0_5
     V_VM_TO_R_SALDO1000_5 V_VM_ATM500_5 V_VM_TO_CTAS_AB2_5
     V_VM_TO_SALDO10000_5 V_VM_B_D_TO45_5 V_VM_OO_R_CTAS_AB3_5
     V_MEDIA_CH_P_5 V_NINC_CH_P_6 V_NDEC_CH_P_6 V_ANT_INT_ANTIG
     V_EDAD_CLIENTE V_MOB_TDC V_TOT_CRED_BCO V_N_ACADEMICO V_OCUPACION
     V_ST_CIVIL V_SEXO V_CLASIF_CTE V_CANAL V_TERRITORIO V_SEGMENTO
     V_WELCOME_CALL V_ATM V_CELULAR V_TOT_CANALES_ACTI V_ENTREGAT
     V_BCSCORE V_MO_R__CTAS_AB V_OO_R_BANCO_CTAS_AB V_MO_R__LC_MAX
     V_OO_R_BANCO_LC_MAX V_MO_R__SALDO V_OO_R_BANCO_SALDO                    /
                    SELECTION=STEPWISE
                    SLE=0.05
                    SLS=0.05
                    INCLUDE=0
                    LINK=LOGIT
          ;
RUN;
QUIT;



PROC LOGISTIC DATA=X
PLOTS(ONLY)=ROC
;
MODEL TARGET (Event = '0')=V_NINC_CH_P_3 V_NDEC_ATM_3
```

```
V_NDEC_OO_R_LC_SUM_3 V_NDEC_TO_R_LC_SUM_3 V_NINC_B_D_OO_4
V_NINC_OO_R_LC_SUM_4 V_NINC_TO_R_LC_SUM_4 V_NDEC_B_D_OO_4
V_NDEC_TO_R_SALDO_4 V_VM_CH_P0_4 V_VM_CH_P100_4 V_VM_OO_R_CTAS_AB1_4
V_VM_TO_R_LC_SUM5000_4 V_VM_CH_P1500_4 V_VM_OO_R_LC_MAX25000_4
V_VM_OO_R_SALDO25000_4 V_VM_TO_R_LC_MAX25000_4 V_VM_TO_R_SALDO25000_4
V_NINC_ATM_5 V_NINC_B_D_TO_5 V_NINC_OO_R_SALDO_5 V_NINC_TO_SALDO_5
V_NDEC_ATM_5 V_NDEC_OO_R_CTAS_AB_5 V_NDEC_OO_R_LC_MAX_5
V_NDEC_TO_CTAS_AB_5 V_NDEC_TO_R_LC_MAX_5 V_NDEC_TO_SALDO_5
V_VM_OO_R_CTAS_AB0_5 V_VM_TO_R_SALDO1000_5 V_VM_ATM500_5
V_VM_TO_CTAS_AB2_5 V_VM_TO_SALDO10000_5 V_VM_B_D_TO45_5
V_VM_OO_R_CTAS_AB3_5 V_MEDIA_CH_P_5 V_NINC_CH_P_6 V_NDEC_CH_P_6
V_ANT_INT_ANTIG V_EDAD_CLIENTE V_MOB_TDC V_TOT_CRED_BCO V_N_ACADEMICO
V_OCUPACION V_ST_CIVIL V_SEXO V_CLASIF_CTE V_CANAL V_TERRITORIO
V_SEGMENTO V_WELCOME_CALL V_ATM V_CELULAR V_TOT_CANALES_ACTI V_ENTREGAT
V_BCSCORE V_MO_R__CTAS_AB V_OO_R_BANCO_CTAS_AB V_MO_R__LC_MAX
V_OO_R_BANCO_LC_MAX V_MO_R__SALDO V_OO_R_BANCO_SALDO                    /
SELECTION=STEPWISE
SLE=0.05
SLS=0.05
INCLUDE=0
LINK=LOGIT
ALPHA=0.05
CLPARM=WALD
TECHNIQUE=NEWTON
;
OUTPUT OUT=SALIDA
PREDPROBS=INDIVIDUAL;
RUN;
QUIT;
ODS GRAPHICS OFF;

DATA Y;
SET SALIDA;
FORMAT R_PROBABILIDAD $20. ;
IF IP_1 <= .1 THEN R_PROBABILIDAD = '01.(0|10]';
ELSE IF IP_1 > .1 AND IP_1 <= .2 THEN R_PROBABILIDAD = '02.(10|20]';
ELSE IF IP_1 > .2 AND IP_1 <= .3 THEN R_PROBABILIDAD = '03.(20|30]';
ELSE IF IP_1 > .3 AND IP_1 <= .4 THEN R_PROBABILIDAD = '04.(30|40]';
ELSE IF IP_1 > .4 AND IP_1 <= .5 THEN R_PROBABILIDAD = '05.(40|50]';
ELSE IF IP_1 > .5 AND IP_1 <= .6 THEN R_PROBABILIDAD = '06.(50|60]';
ELSE IF IP_1 > .6 AND IP_1 <= .7 THEN R_PROBABILIDAD = '07.(60|70]';
ELSE IF IP_1 > .7 AND IP_1 <= .8 THEN R_PROBABILIDAD = '08.(70|80]';
ELSE IF IP_1 > .8 AND IP_1 <= .9 THEN R_PROBABILIDAD = '09.(80|90]';
ELSE IF IP_1 > .9 THEN R_PROBABILIDAD = '10.(90|+)';
RUN;

PROC SQL;
CREATE TABLE CUB AS
SELECT TARGET, R_PROBABILIDAD, COUNT(*) AS CASOS
FROM Y
GROUP BY R_PROBABILIDAD, TARGET
; QUIT;
```