**Autor**: José Romualdo Villalobos Pérez

**ID**: 000294087

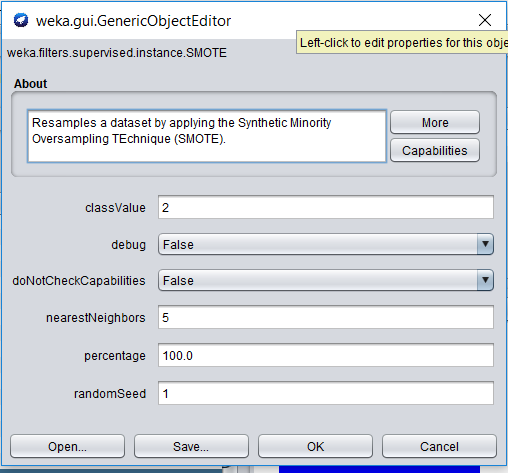
**El problema**:

Un banco alemán desea establecer políticas de mercadeo según el tipo de cliente. Para esto se desean identificar tipos de clientes de una base de datos de 1000 registros con 20 atributos cada uno.

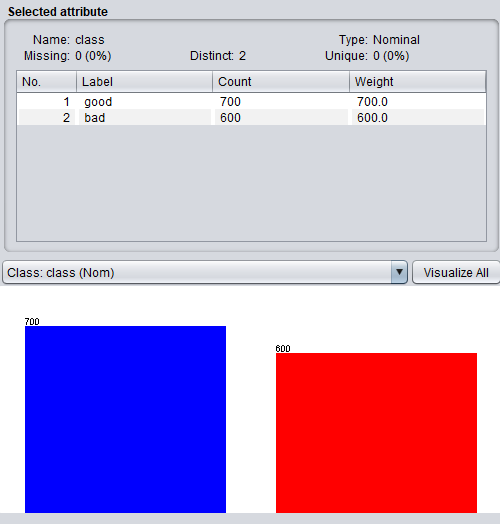
# Preparando los datos



Al observar los datos, se consideró que estos se encuentran desbalanceados, por lo tanto se procedió a balancearlos aplicando un SMOTE.



El SMOTE agregó 300 nuevas instancias al dataset dándonos un total de 1300 instancias en total.

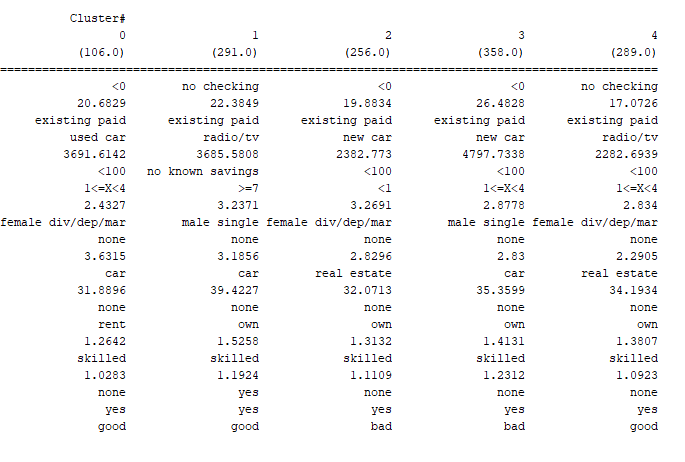


Como se puede evidenciar el dataset quedo significativamente más balanceado.

Luego, se reviso el dataset en busca de valores atípicos, pero no se identificó ninguno.

# Definiendo el número de clusters a detectar

Determinar el parámetro K es uno de los problemas de las técnicas no supervisadas, existen diversos métodos para hallar un valor optimo, como por ejemplo maximizar el Bayesian Information Criterion, sin embargo para nuestro caso se utilizó una técnica de prueba – error para determinar dicho parámetro, teniendo como criterio de selección **el primer valor de K para el cual la cantidad de centroides que pertenezcan al grupo de buenos clientes sea mayor al numero de centroides que pertenezcan al grupo de malos clientes** al aplicar el algoritmo de clustering **k-means**.

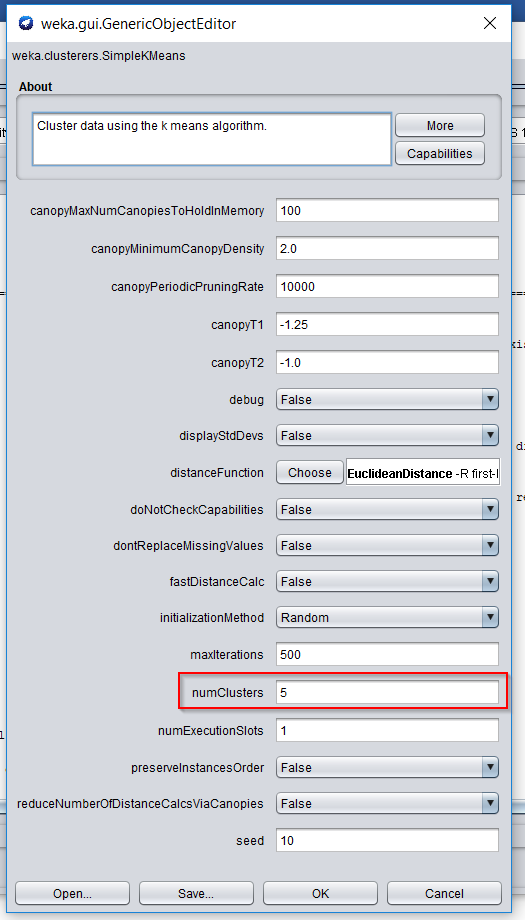


Se usó este criterio porque se consideró más importante segmentar a los buenos clientes que segmentar a los malos, esto con el objetivo de intentar brindarles una experiencia más personalizada.

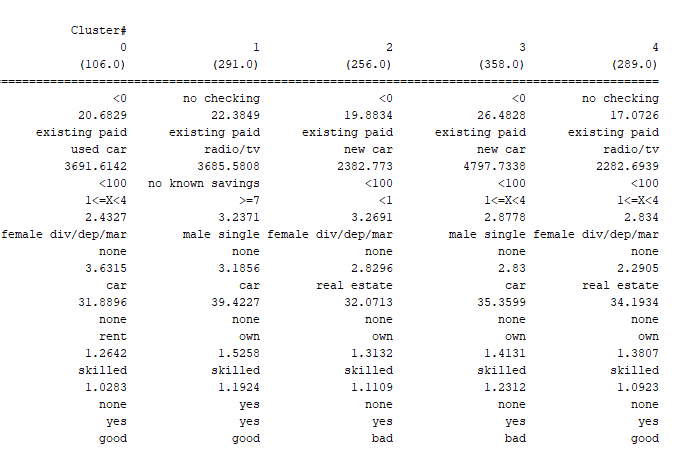
Luego de varias pruebas el primer valor de K que satisfizo este criterio fue **K=5**

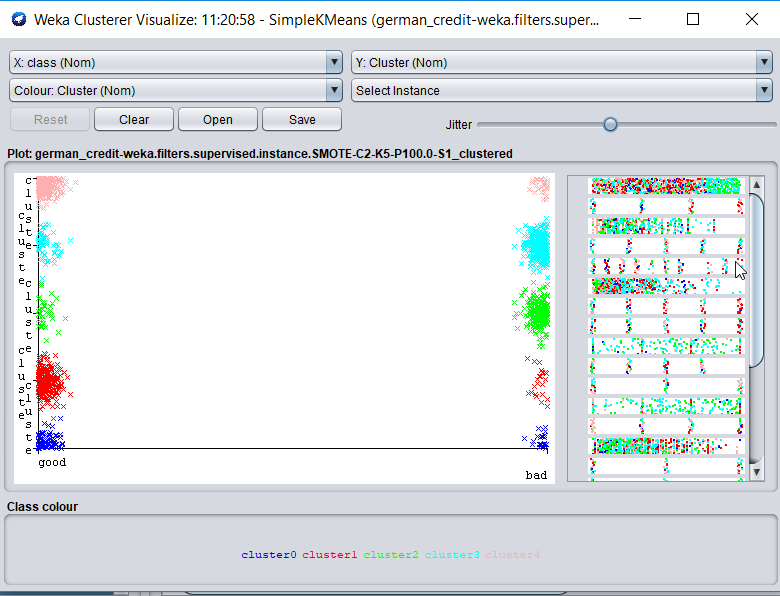
A continuación, vamos a probar varios algoritmos no supervisados de clustering.

# K-Means



Se obtuvieron los siguientes centroides para los 5 clusters:





Lamentablemente este método de clustering no separó los clientes buenos de los malos.

Ahora, vamos a nombrar los clusters:

**Cluster 0:**

**Nombre:** Independiente

**Descripción Persona:** Mujer responsable y dedicada a su carrera, no le da miedo ir a otros países a trabajar, estas mujeres le huyen a los gastos grandes, prefiere tener un auto usado que uno nuevo, prefiere rentar su vivienda que comprar una no le preocupa lo que pueda pasar en el futuro.

**Cluster 1:**

**Nombre:** Libertad

**Descripción Persona:** Dedicado a su carrera, valora mucho los espacios de ocio, por eso tiene un TV gigante, vive en su propia casa y no tiene pareja.

**Cluster 2:**

**Nombre:** Imperial

**Descripción Persona:** Tiene su propia casa, no le gusta endeudarse a largo plazo, pero a veces no calcula bien sus verdaderas capacidades, le gusta movilizarse en automóvil y prefiere sentir que es la dueña.

**Cluster 3:**

**Nombre:** Orgullo personal

**Descripción Persona:** No tiene pareja, le gusta lucir su automóvil, dedicado a su carrera, se siente cómodo en proyectos de mediano plazo, a veces no calcula bien sus verdaderas capacidades.

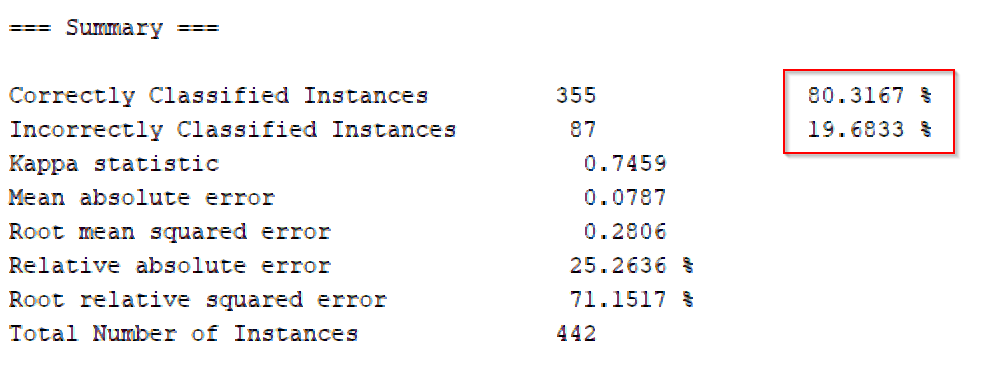
**Cluster 4:**

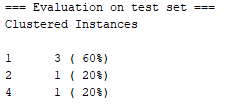
**Nombre:** Vive la vida

**Descripción Persona:** Dedicada a su carrera, valora su comodidad y los momentos de ocio, tiene su propia casa, despreocupada de lo que pueda pasar en el futuro.

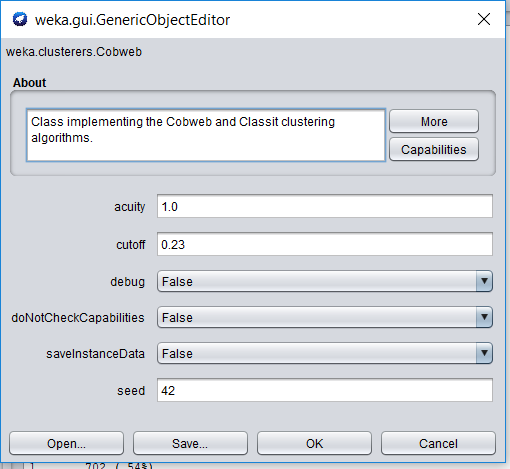
**Probando el modelo con 5 usuarios nuevos:**

Para predecir a que cluster pertenecen 5 usuarios nuevos, vamos a aplicar el clasificador Support Vector Machine donde la clase de cada instancia es el cluster al que pertenece, para este caso entrenamos una SVM con un kernel lineal.

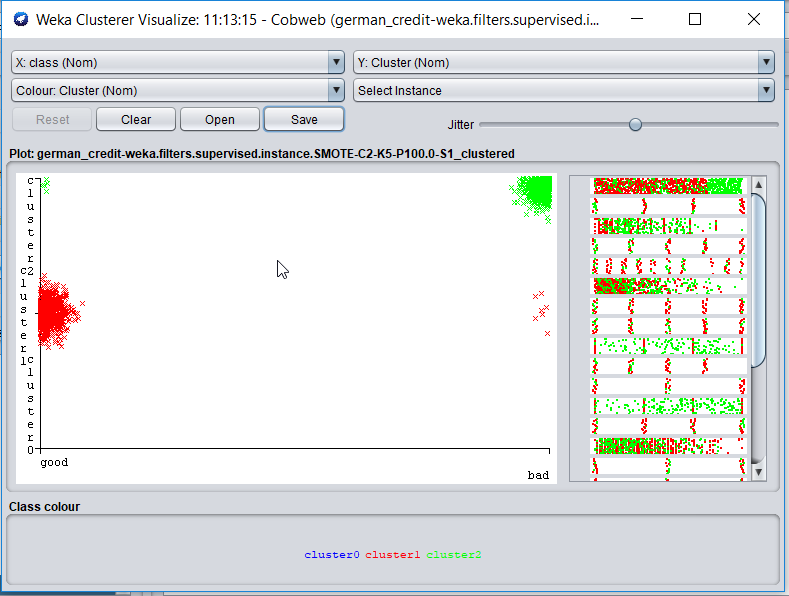




# Cobweb

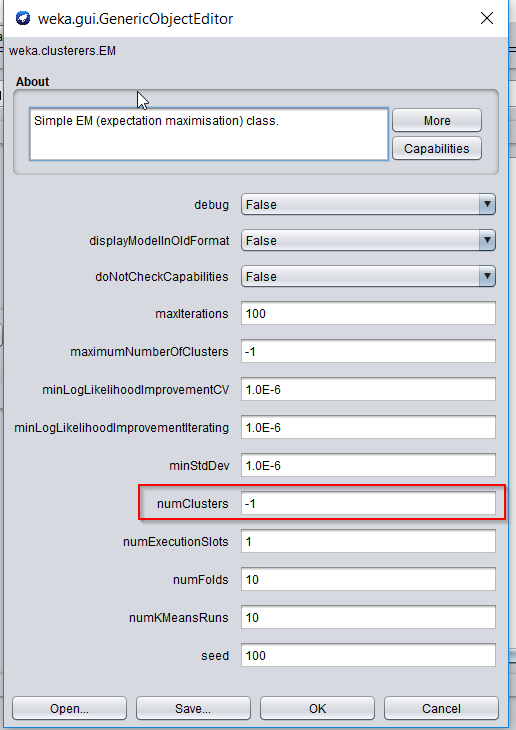


El valor de cutoff es 0.23 para evitar que el árbol crezca mucho, el resultado obtenido se muestra a continuación:

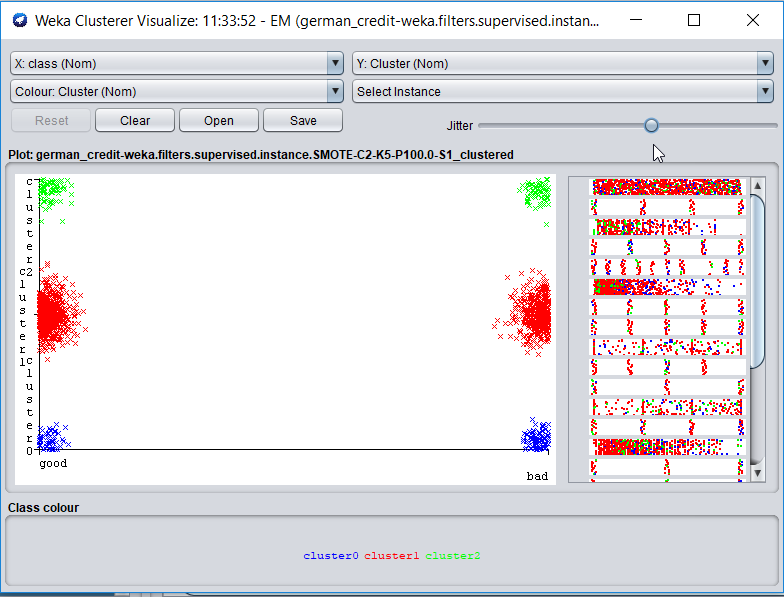


Afortunadamente para el banco, este método de clustering agrupo a los clientes buenos y a los clientes malos en clusters diferentes.

# Expectation Maxification



Cuando numClusters = -1 el algoritmo selecciona un valor automáticamente para K basado en la cross validation.



No separó a los clientes buenos de los clientes malos.

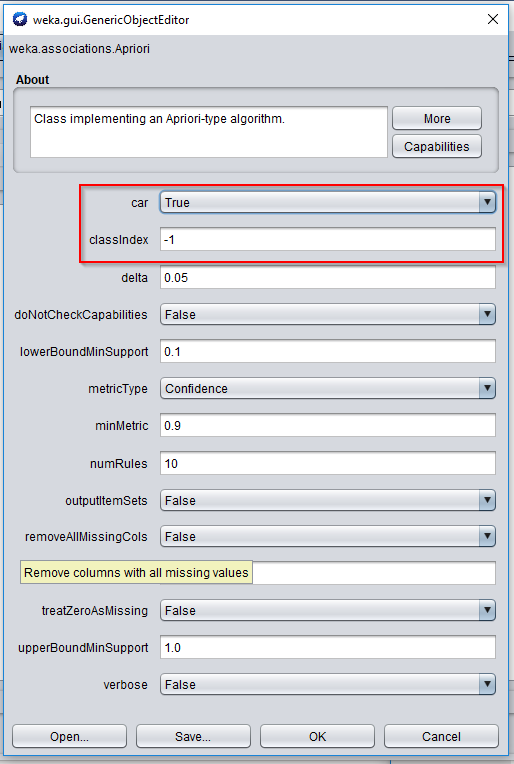
# Self-Organizing Maps

Debigo a que Weka no tiene una clase para aplicar este algoritmo, se implementara en Python y en R únicamente.

*“Self-organizing map is not in weka catalogue the only neural networks do not compress features to viewable 2d map only regular nn's named multilayerperceptron and rbfnetwork which have different (opposite) activation functions”*

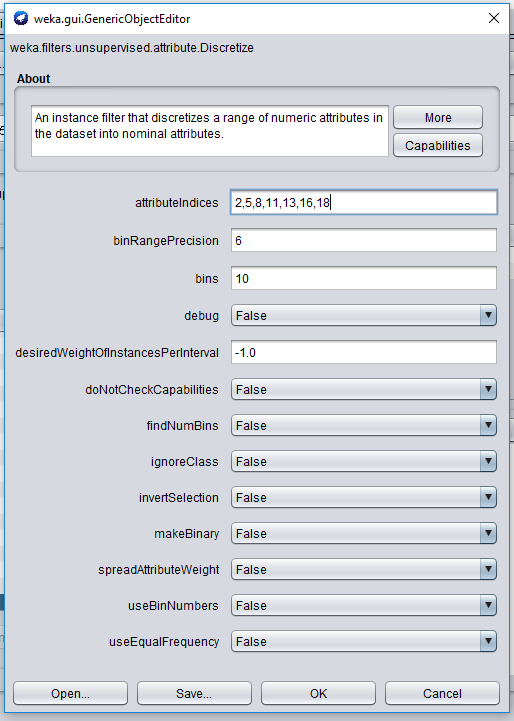
* *Harry M.T. Saarikoski, Weka mail list*

# Association, A priori

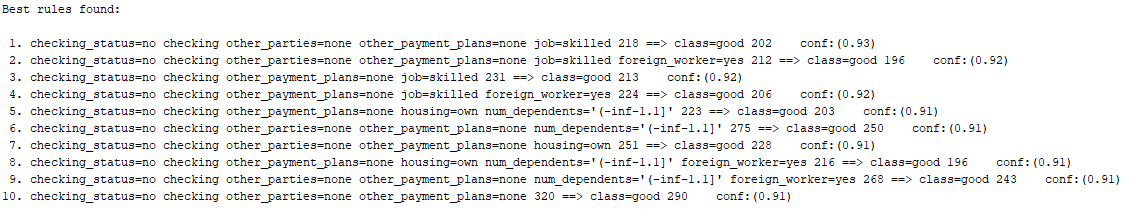


**Car = true** porque las instancias tienen un atributo clase, y **classIndex=-1** porque la clase se encuentra en la última columna.

Debido a que el método apriori solo funciona con variables categóricas, vamos a proceder discretizando los atributos numéricos.



Como resultado, ahora todos los atributos son de tipo nominal, ahora si podemos aplicar las reglas de asociación.



Podemos concluir que este algoritmo brinda al banco información muy valiosa, que podrá permitir sacar mas valor de sus clientes.