

Maestría en Ciencia de Datos

Minería de Datos

Proyecto 02: Agrupamiento de bases de datos numéricas

Profesor:

Dr. Ángel Fernando Kuri Morales

Alumna:

Gabriela Flores Bracamontes

Clave única:

160124

México, D.F. 3 de diciembre de 2015.

Contenido

1. Antecedentes	3
2. Metas del problema asociado a resolver	3
3. Proyecto	3
3.1. Plataforma tecnológica	3
3.2. Descripción de los datos.....	4
3.3. Fases del proyecto.....	6
3.3.1. Descargar la base de datos.....	6
3.3.2. Análisis de los datos:	6
3.3.3. Preprocesamiento:	7
3.3.3.1. Creación de las bases de datos dbf	8
3.3.3.2. Convertir variables categóricas a pseudo-binarias	9
3.3.3.3. Escalar	11
3.3.3.4. Estabilizar	11
3.3.3.5. Correlacionar.....	12
3.3.3.6. Convertir los archivos en formato dbf a txt	13
3.3.4. Red Neuronal de Konohen:	14
3.3.4.1. Convertir los archivos en formato txt a dat	14
3.3.4.2. Determinamos el número de clusters.....	14
3.3.4.3. Crear la red neuronal de Konohen.....	16
3.3.4.4. Configurar los parámetros	16
3.3.4.5. Primeros Resultados.....	17
3.3.4.6. Proceso de etiquetamiento (Labeling)	18
4. Conclusiones.....	22
5. Bibliografía	22

1. Antecedentes

El agrupamiento o clustering es una tarea popular de la minería de datos que consiste en procesar un gran volumen de datos para obtener los grupos donde los elementos de cada exposición colectiva cuantificable (bajo algunas métricas) pequeñas diferencias entre ellos y, por el contrario, grandes diferencias entre los elementos de los diferentes grupos. Dada su gran importancia como una tarea de minería de datos, clustering ha sido objeto de múltiples esfuerzos de investigación y tiene demostrado ser útiles para muchos propósitos [1].

Para obtener un cluster, sin tener información a priori se utilizan los mapas auto organizados o SOM (Self-Organizing Map), también llamados redes de Kohonen [1995] son un tipo de red neuronal no supervisada competitiva, con capacidad para formar mapas de características bidimensionales a partir del principio de formación de mapas topológicos. Se orientan a descubrir la estructura subyacente de los datos ingresados a partir de establecer características comunes entre los vectores de información de entrada a la red. A lo largo del entrenamiento de la red; los vectores de datos son introducidos en cada neurona y se comparan con el vector de peso característico de la misma. La neurona que presenta menor diferencia entre su vector de peso y el vector de datos es la neurona ganadora (o BMU) y ella y sus vecinas verán modificados sus vectores de pesos.

2. Metas del problema asociado a resolver

Obtener el agrupamiento de una base de datos de carros según sus características a través de la utilización de mapas auto-organizados.

3. Proyecto

3.1. Plataforma tecnológica

Para realizar el presente proyecto se instaló una máquina virtual con Sistema Operativo Windows XP a 32 bits, ya que las herramientas utilizadas solamente funcionan en Sistemas Operativos Windows a 32 bits:

- PREPROC versión 9.2.- Se utilizó para realizar el preprocesamiento de los datos.
- DATA ENGINE versión 2.10.012.- Se utilizó para obtener generar el mapa auto organizado.
- R – Statistical Data Analysis.

3.2. Descripción de los datos.

La base de datos de evaluación de carros, se descargó del repositorio de Aprendizaje Máquina de la UCI en la siguiente ruta:

- <https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

1. Title: Car Evaluation Database

2. Sources:

- (a) Creator: Marko Bohanec
- (b) Donors: Marko Bohanec (marko.bohanec@ijs.si)
Blaz Zupan (blaz.zupan@ijs.si)
- (c) Date: June, 1997

3. Past Usage:

The hierarchical decision model, from which this dataset is derived, was first presented in

M. Bohanec and V. Rajkovic: Knowledge acquisition and explanation for multi-attribute decision making. In 8th Intl Workshop on Expert Systems and their Applications, Avignon, France. pages 59-78, 1988.

Within machine-learning, this dataset was used for the evaluation of HINT (Hierarchy INduction Tool), which was proved to be able to completely reconstruct the original hierarchical model. This, together with a comparison with C4.5, is presented in

B. Zupan, M. Bohanec, I. Bratko, J. Demsar: Machine learning by function decomposition. ICML-97, Nashville, TN. 1997 (to appear)

4. Relevant Information Paragraph:

Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX (M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure:

CAR	car acceptability
. PRICE	overall price
. . buying	buying price
. . maint	price of the maintenance
. TECH	technical characteristics
. . COMFORT	comfort
. . . doors	number of doors

```

... persons      capacity in terms of persons to carry
... lug_boot     the size of luggage boot
... safety       estimated safety of the car

```

Input attributes are printed in lowercase. Besides the target concept (CAR), the model includes three intermediate concepts: PRICE, TECH, COMFORT. Every concept is in the original model related to its lower level descendants by a set of examples (for these examples sets see <http://www-ai.ijs.si/BlazZupan/car.html>).

The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety.

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods.

5. Number of Instances: 1728
(instances completely cover the attribute space)

6. Number of Attributes: 6

7. Attribute Values:

```

buying    v-high, high, med, low
maint     v-high, high, med, low
doors     2, 3, 4, 5-more
persons   2, 4, more
lug_boot  small, med, big
safety    low, med, high

```

8. Missing Attribute Values: none

9. Class Distribution (number of instances per class)

class	N	N[%]
unacc	1210	(70.023 %)
acc	384	(22.222 %)
good	69	(3.993 %)
v-good	65	(3.762 %)

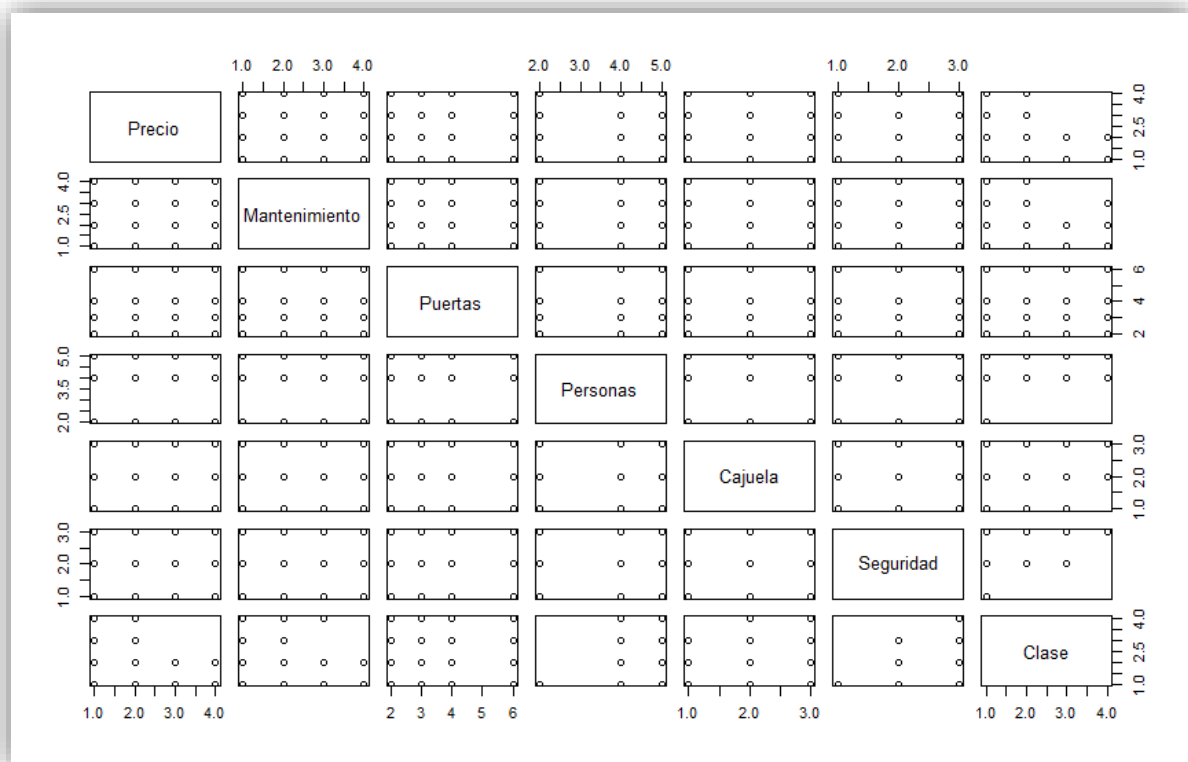
3.3. Fases del proyecto

3.3.1. Descargar la base de datos

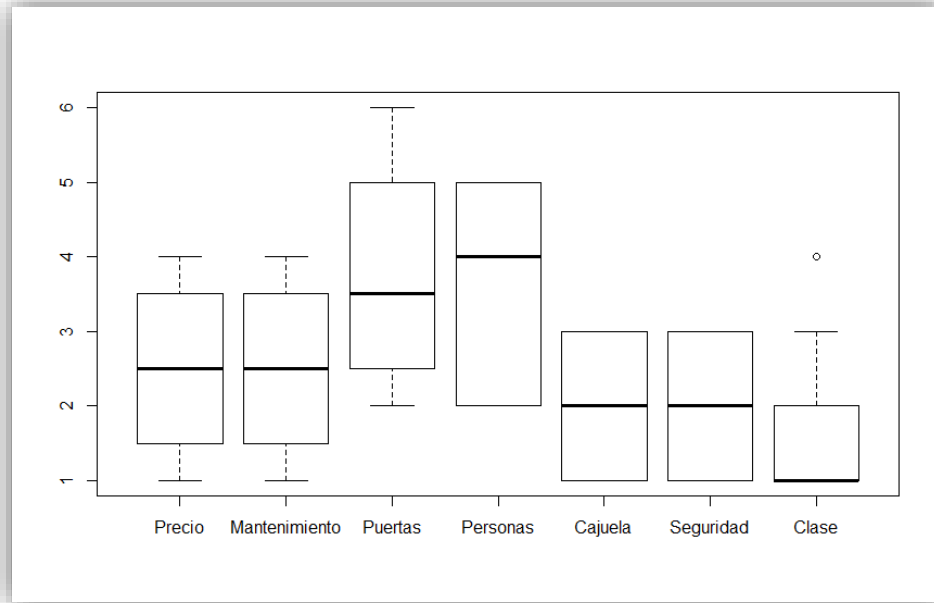
La base de datos se descargó de la página señalada en la sección “Descripción de los datos” y se guardó en un archivo con formato TXT.

3.3.2. Análisis de los datos:

- Los datos venían separados por comas.
- Los datos contienen 6 atributos que son: Precio, Mantenimiento, Puertas, Personas, Cajuela, Seguridad y una variable dependiente llamada **Clase**.
- Los atributos y la variable dependiente son variables categóricas.
- En la revisión de los datos, se realiza una primera gráfica para que se visualicen mejor los datos.

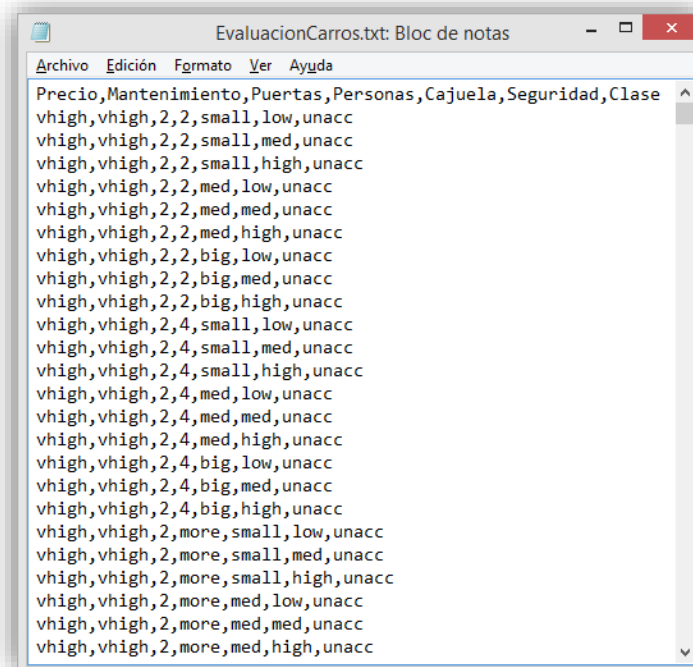


- También se realiza una gráfica de cajas y brazos



3.3.3. Preprocesamiento:

Una vez descargada la base de datos se guarda en un archivo txt llamado **EvaluacionCarros.txt**

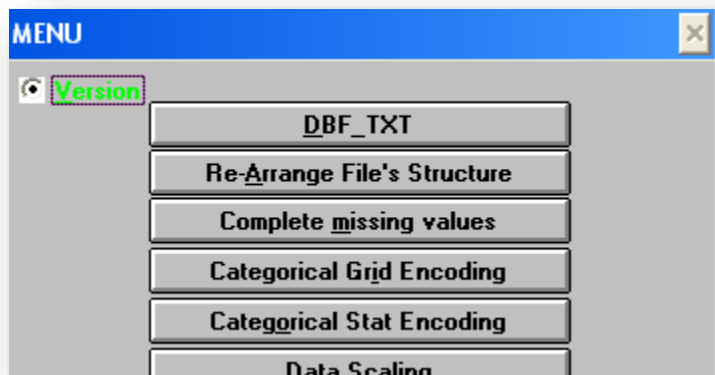


```

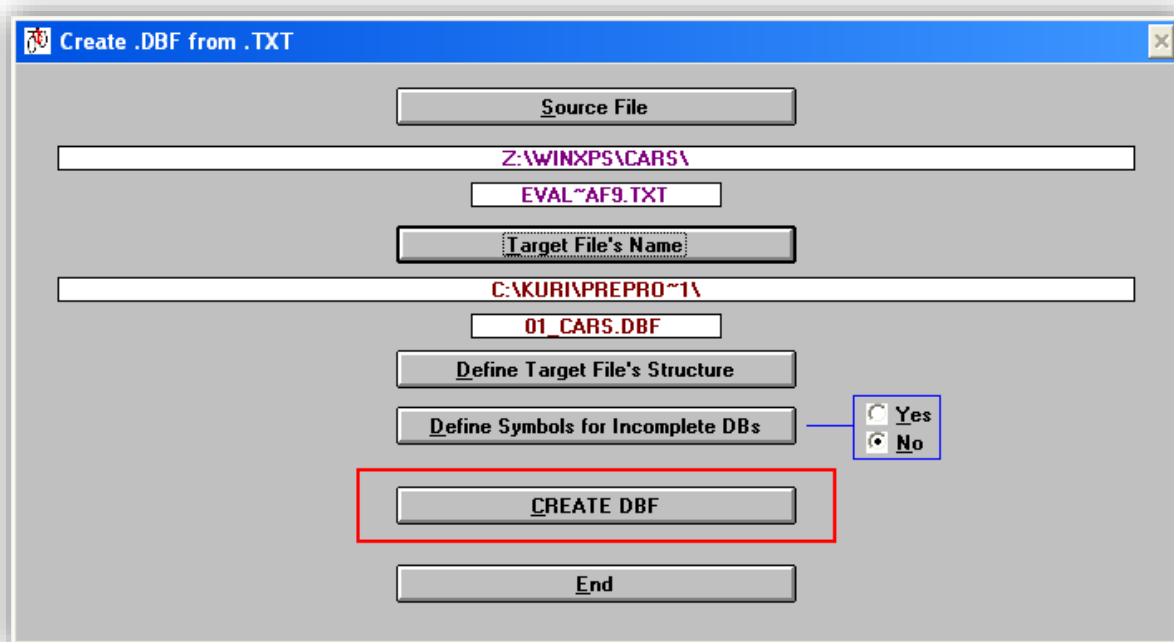
Archivo Edición Formato Ver Ayuda
Precio,Mantenimiento,Puertas,Personas,Cajuela,Seguridad,Clase
vhigh,vhigh,2,2,small,low,unacc
vhigh,vhigh,2,2,small,med,unacc
vhigh,vhigh,2,2,small,high,unacc
vhigh,vhigh,2,2,med,low,unacc
vhigh,vhigh,2,2,med,med,unacc
vhigh,vhigh,2,2,med,high,unacc
vhigh,vhigh,2,2,big,low,unacc
vhigh,vhigh,2,2,big,med,unacc
vhigh,vhigh,2,2,big,high,unacc
vhigh,vhigh,2,4,small,low,unacc
vhigh,vhigh,2,4,small,med,unacc
vhigh,vhigh,2,4,small,high,unacc
vhigh,vhigh,2,4,med,low,unacc
vhigh,vhigh,2,4,med,med,unacc
vhigh,vhigh,2,4,med,high,unacc
vhigh,vhigh,2,4,big,low,unacc
vhigh,vhigh,2,4,big,med,unacc
vhigh,vhigh,2,4,big,high,unacc
vhigh,vhigh,2,more,small,low,unacc
vhigh,vhigh,2,more,small,med,unacc
vhigh,vhigh,2,more,small,high,unacc
vhigh,vhigh,2,more,med,low,unacc
vhigh,vhigh,2,more,med,med,unacc
vhigh,vhigh,2,more,med,high,unacc
    
```

3.3.3.1. Creación de las bases de datos dbf

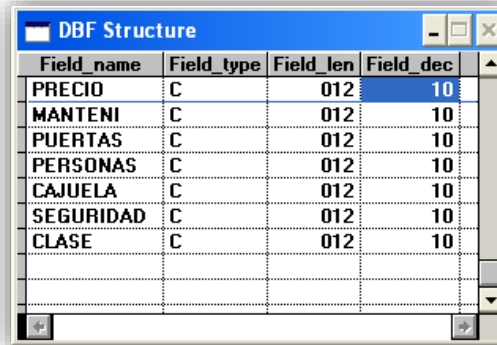
Utilizamos la opción **DBF_TXT** del programa PREPROC, para crear la base de datos inicial.



Seleccionamos el archivo fuente en este caso **EvaluacionCarros.txt** y el nombre de la base de datos, en este caso **01_CARS.DBF**.

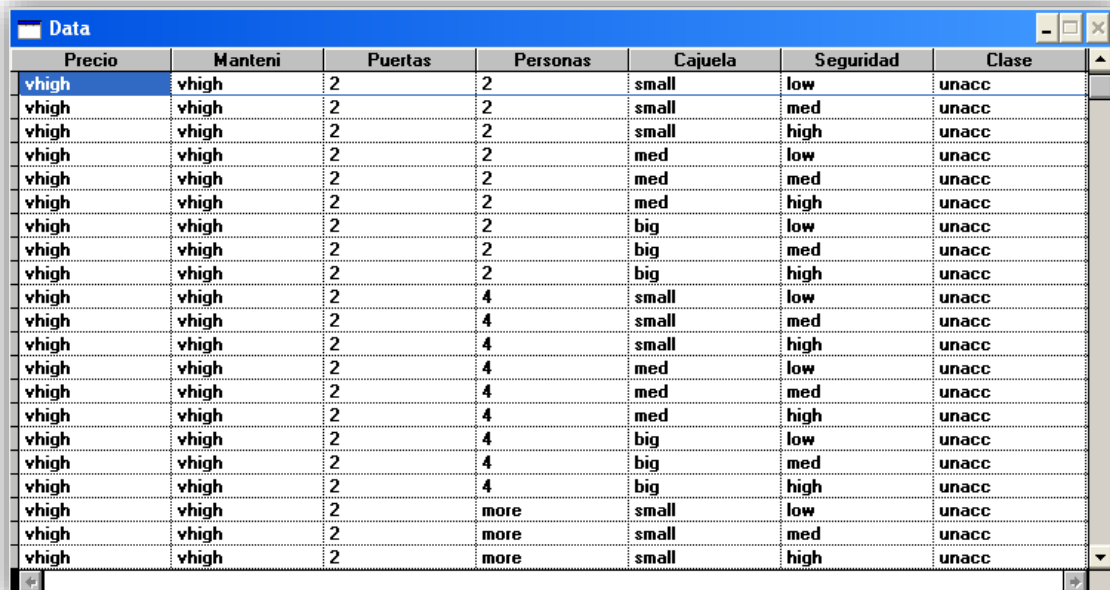


La estructura de datos utilizada fue:



Field_name	Field_type	Field_len	Field_dec
PRECIO	C	012	10
MANTENI	C	012	10
PUERTAS	C	012	10
PERSONAS	C	012	10
CAJUELA	C	012	10
SEGURIDAD	C	012	10
CLASE	C	012	10

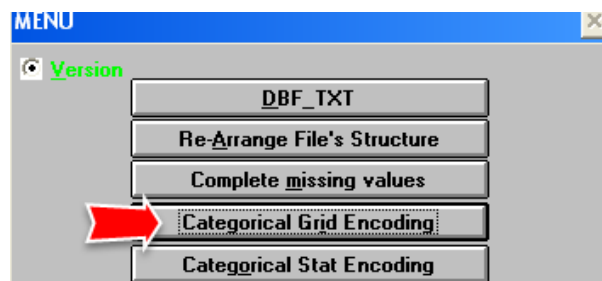
La base de datos queda de la siguiente manera:



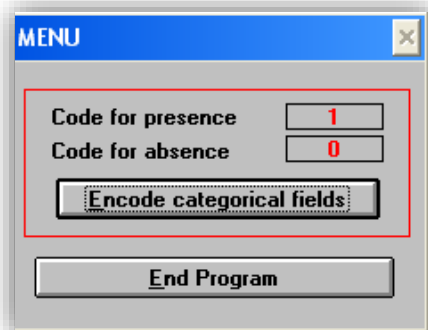
Precio	Manteni	Puertas	Personas	Cajuela	Seguridad	Clase
vhigh	vhigh	2	2	small	low	unacc
vhigh	vhigh	2	2	small	med	unacc
vhigh	vhigh	2	2	small	high	unacc
vhigh	vhigh	2	2	med	low	unacc
vhigh	vhigh	2	2	med	med	unacc
vhigh	vhigh	2	2	med	high	unacc
vhigh	vhigh	2	2	big	low	unacc
vhigh	vhigh	2	2	big	med	unacc
vhigh	vhigh	2	2	big	high	unacc
vhigh	vhigh	2	4	small	low	unacc
vhigh	vhigh	2	4	small	med	unacc
vhigh	vhigh	2	4	small	high	unacc
vhigh	vhigh	2	4	med	low	unacc
vhigh	vhigh	2	4	med	med	unacc
vhigh	vhigh	2	4	med	high	unacc
vhigh	vhigh	2	4	big	low	unacc
vhigh	vhigh	2	4	big	med	unacc
vhigh	vhigh	2	4	big	high	unacc
vhigh	vhigh	2	more	small	low	unacc
vhigh	vhigh	2	more	small	med	unacc
vhigh	vhigh	2	more	small	high	unacc

3.3.3.2. Convertir variables categóricas a pseudo-binarias

Utilizamos la opción Categorical Grid Encode en el Programa PREPROC

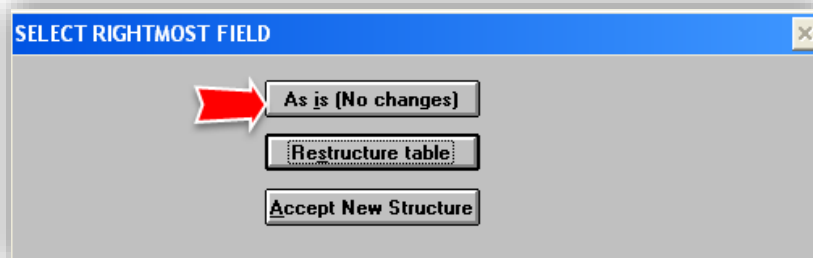


En la pantalla que se despliega dejamos los valores 1 si existe y 0 en caso de ausencia.



El sistema nos pide el archivo de entrada que se va a codificar, en este caso usamos el **01_CARS.DBF** y para la información de salida con las variables codificada se utilizamos el archivo **02_CARS.DBF**

A continuación, nos despliega una pantalla en la que seleccionamos la opción As is, esto quiere decir que no queremos realizar ningún cambio en la estructura de la base.

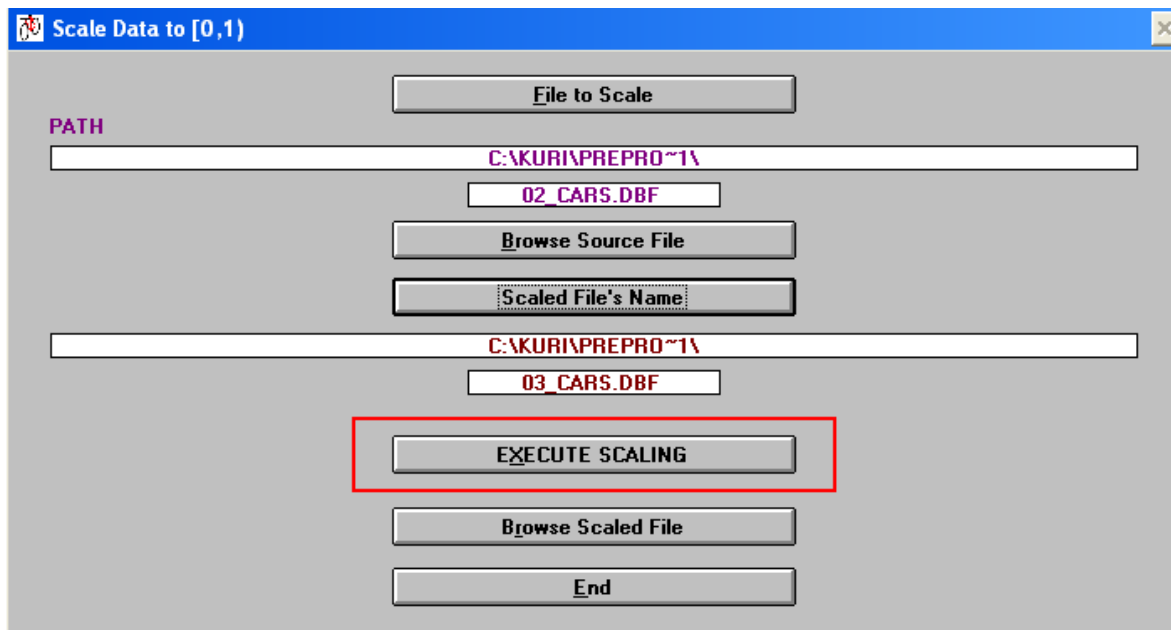


Las variables han sido codificadas, y el archivo **02_CARS.DBF** ahora contiene 25 variables pseudo-binarias.

[illegible]

3.3.3.3. Escalar

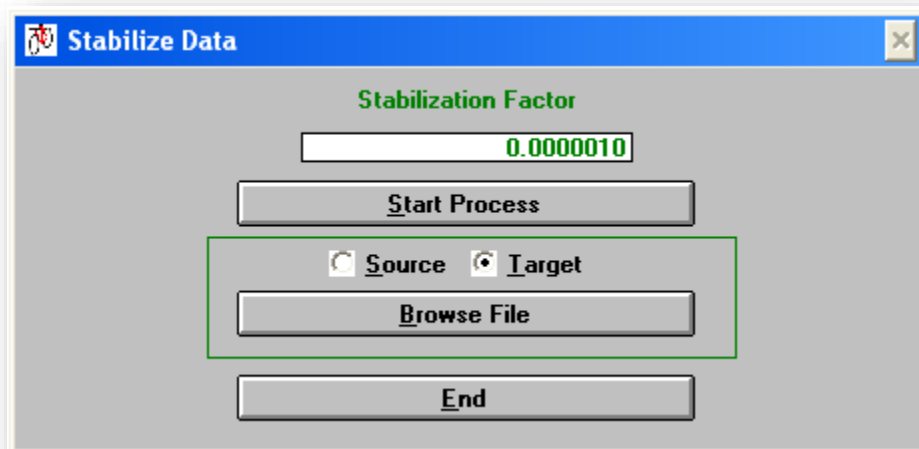
Los datos son escalados entre 0,1:



De este procedimiento se generan la base de datos:

- 03_CARS.DBF

3.3.3.4. Estabilizar

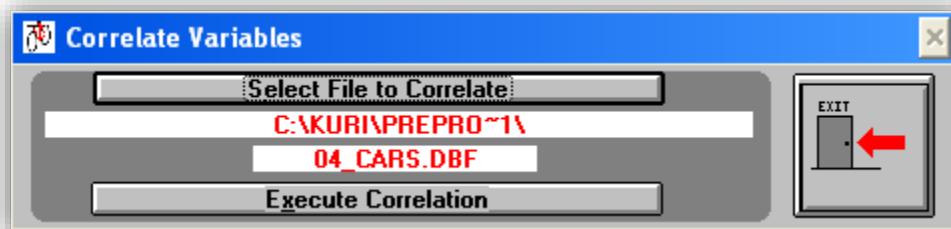


De este procedimiento se generan las siguientes bases de datos:

- 04_CARS.DBF

Data				
F06_03	F07_01	F07_02	F07_03	F07_04
0.000000518055	0.000000396430	0.000000388198	1.000000104615	0.000000724912
1.000000927823	0.000000947544	0.00000040682	1.000000938998	0.000000243696
0.000000357797	0.000000371479	0.000000375270	1.000000711534	0.000000379320
0.000000173991	0.000000789188	0.000000262040	1.000000654597	0.000000197650
1.00000053726	0.000000946978	0.000000873479	1.000000282333	0.000000363672
0.000000135068	0.000000566377	0.000000956740	1.000000610052	0.000000700382
0.000000077480	0.000000222448	0.000000669336	1.000000469456	0.000000319287
1.000000736370	0.000000972735	0.000000911884	1.000000067232	0.000000862795
0.000000273543	0.000000665944	0.000000516623	1.000000769331	0.000000566808
0.000000555038	0.000000464197	0.000000236589	1.000000912448	0.000000177418
1.000000400353	0.000000990166	0.000000548505	1.000000369225	0.000000021077
0.000000798621	0.000000683016	0.000000661510	1.000000317122	0.000000982251
0.000000233794	0.000000171160	0.000000404683	1.000000122930	0.000000827029
1.000000708973	0.000000959616	0.000000234988	1.000000806478	0.000000754458
0.000000059448	0.000000283076	0.000000406717	1.000000267555	0.000000421814
0.000000467258	0.000000955870	0.000000258222	1.000000890699	0.000000881988
1.000000832656	0.000000612479	0.000000510167	1.000000728379	0.000000819161
0.000000309673	0.000000877674	0.000000662231	1.000000222515	0.000000710591
0.000000778481	0.000000469821	0.000000290488	1.000000577409	0.000000448142
1.000000111312	0.000000931737	0.000000866284	1.000000250122	0.000000352214
0.000000649465	0.000000296340	0.000000693298	1.000000380774	0.000000172427
0.000000220060	0.000000643894	0.000000033425	1.000000846965	0.000000030752
1.000000270198	0.000000634376	0.000000011030	1.000000764638	0.000000980399
0.000000385827	0.000000606456	0.000000616308	1.000000232997	0.000000195285
0.000000166686	0.000000350386	0.000000349780	1.000000269579	0.000000583542
1.000000295520	0.000000992596	0.000000728120	1.000000967781	0.000000311541
0.000000817739	0.000000735296	0.000000034699	1.000000587795	0.000000602922

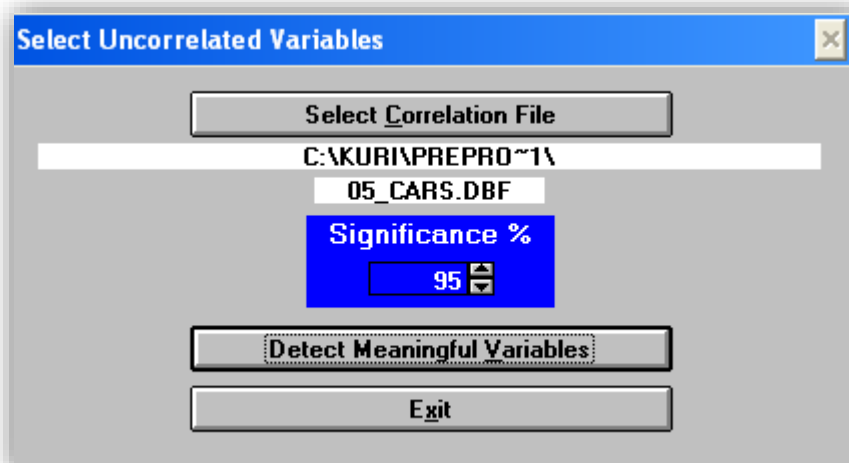
3.3.3.5. Correlacionar



De este procedimiento se generan las siguientes bases de datos:

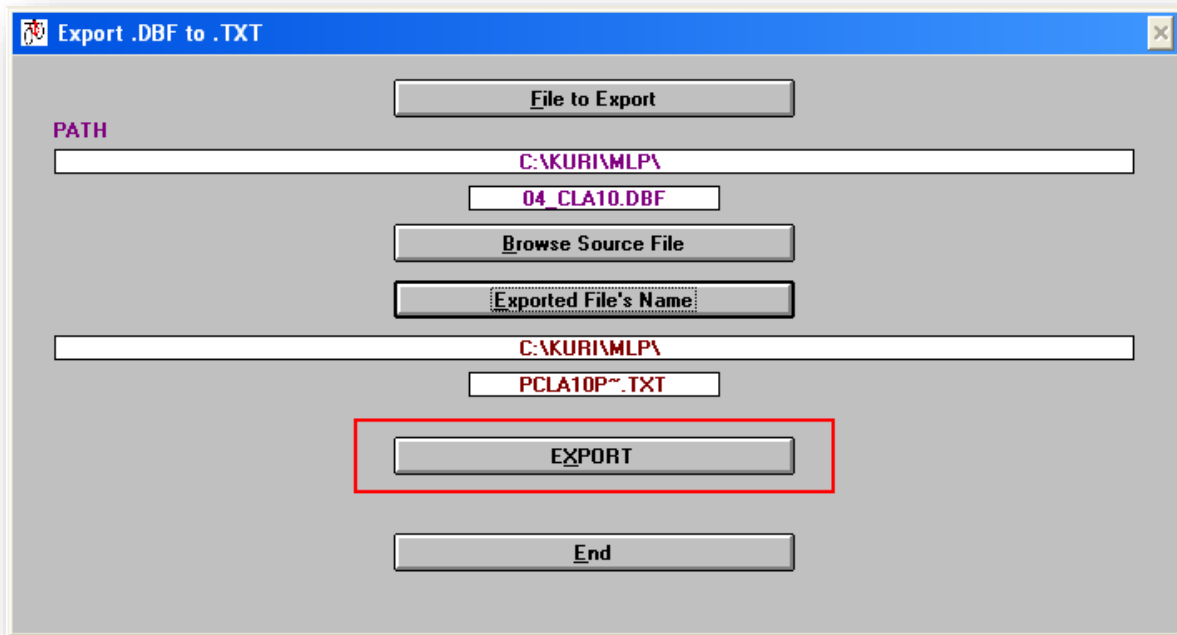
- 05_CARS.DBF

No se encontró ninguna correlación de variables, se realizó el procedimiento de detección en un rango del 50 al 95 %.



3.3.3.6. Convertir los archivos en formato dbf a txt

Por último exportamos la base de datos a un archivo en formato TXT.



De este procedimiento se generó el archivo

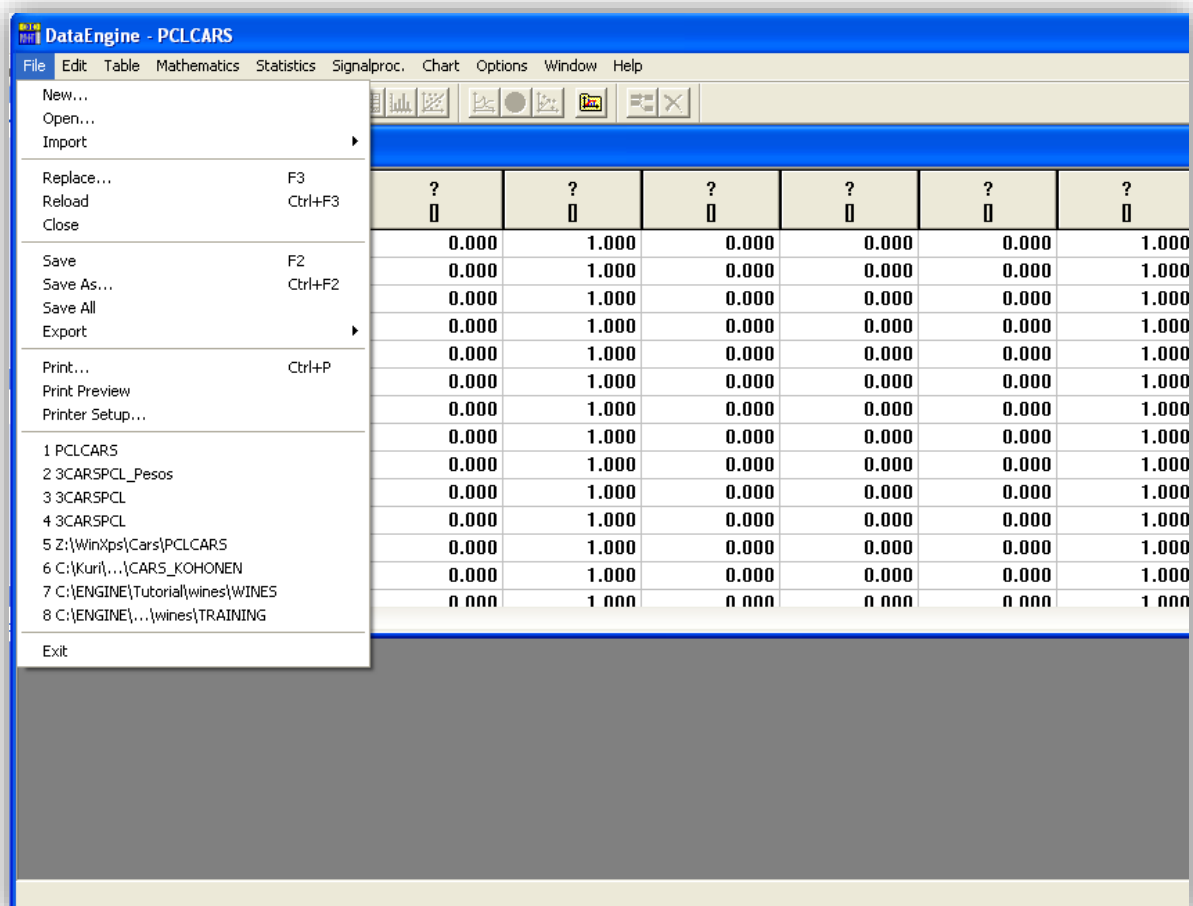
- PCLCARS.txt

3.3.4. Red Neuronal de Konohen:

3.3.4.1. Convertir los archivos en formato txt a dat

El programa Data Engine utiliza archivos en formato .dat por lo que es necesario seleccionar la opción Import del Menú File y seleccionar la opción ASCII.

Una vez que el programa nos muestra el contenido del archivo, seleccionamos la opción Save As y guardamos el archivo con el nuevo formato.

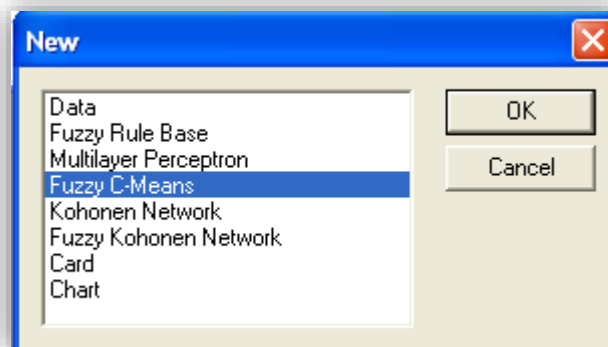


De este procedimiento se genera el archivo

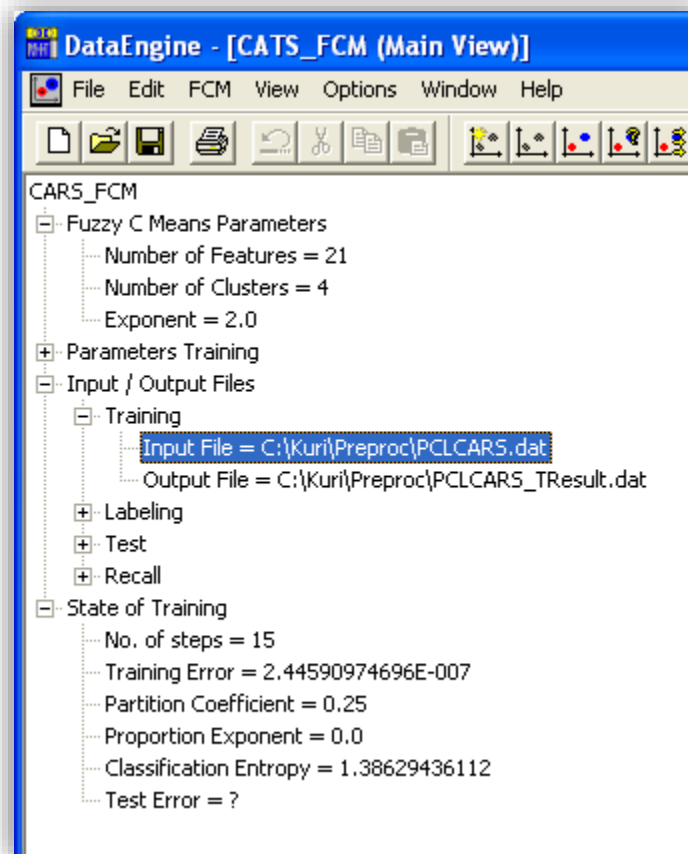
- PCLCARS.dat

3.3.4.2. Determinamos el número de clusters

Para determinar el número de clusters utilizamos el algoritmo Fuzzy C-Means, para lo cuál creamos un nuevo archivo de tipo Fuzzy C-Means en el DATAENGINE.



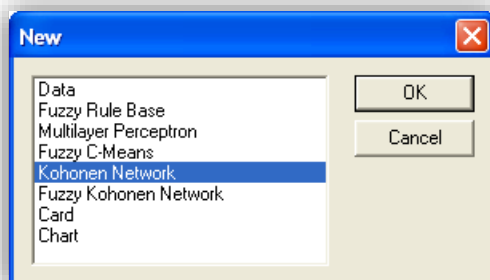
Para encontrar el número de clusters, se hicieron pruebas con 2,4,6,8 y 10 números de clusters mismos que se fueron modificando en el parámetro Number of Clusters.



Por la naturaleza de la base de datos, este proceso no nos fue de gran utilidad para determinar el número de clusters, por lo que se trató de obtener los números directamente de los pesos de las neuronas dentro de la red neuronal de Kohonen que se muestran a continuación.

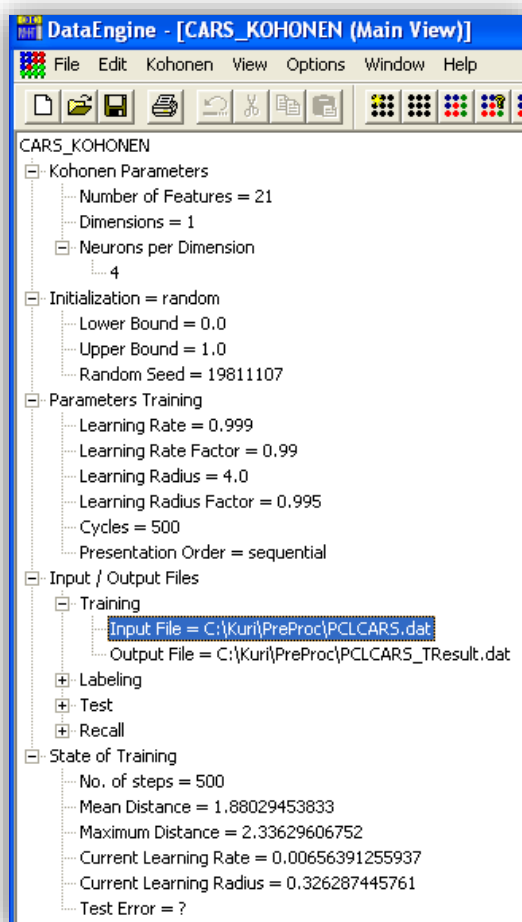
3.3.4.3. Crear la red neuronal de Konohen

Abrimos el programa Data Engine, seleccionamos del menú la opción New y seleccionamos Konohen Network. De este procedimiento se generó el archivo **CARS_KONOHEN.koh**



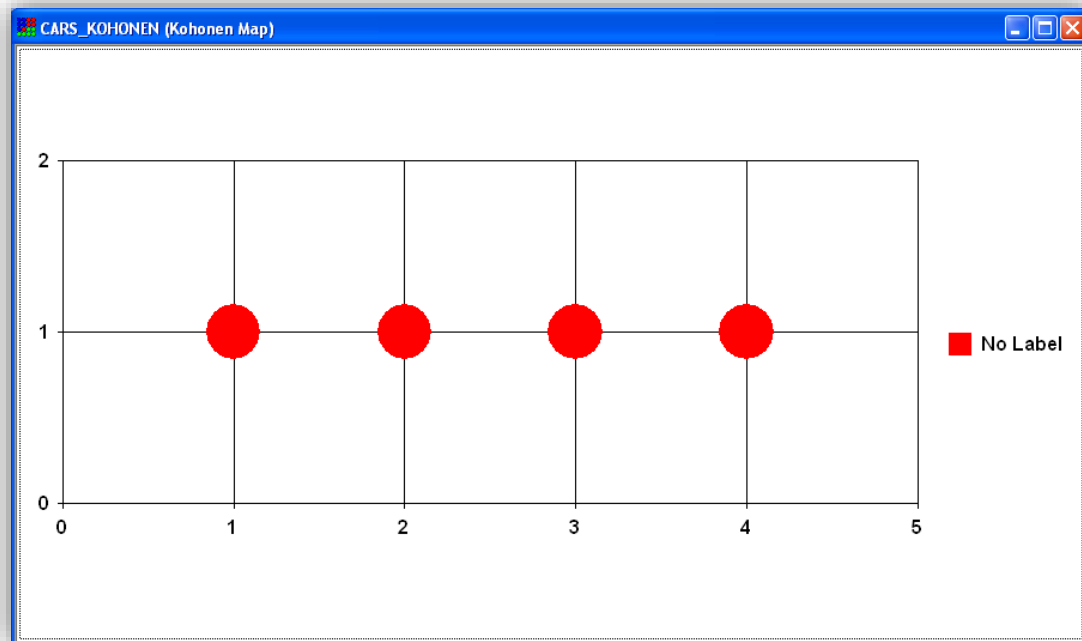
3.3.4.4. Configurar los parámetros

A continuación se muestra la configuración de la red.

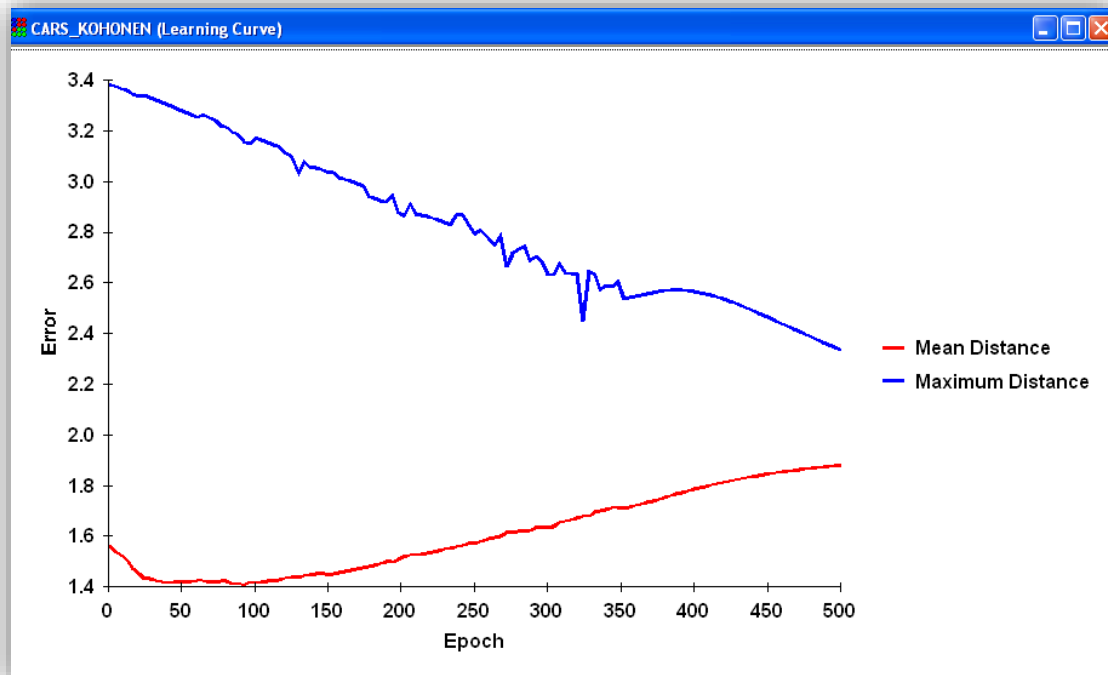


3.3.4.5. Primeros Resultados

El mapa de Kohonen lo muestra sin sus etiquetas, por lo que posteriormente se hará el proceso de labeling.



Curva de aprendizaje

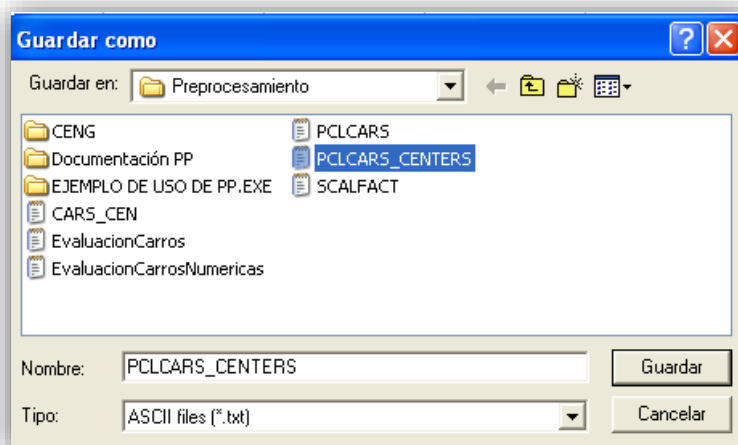


Las coordenadas de las neuronas

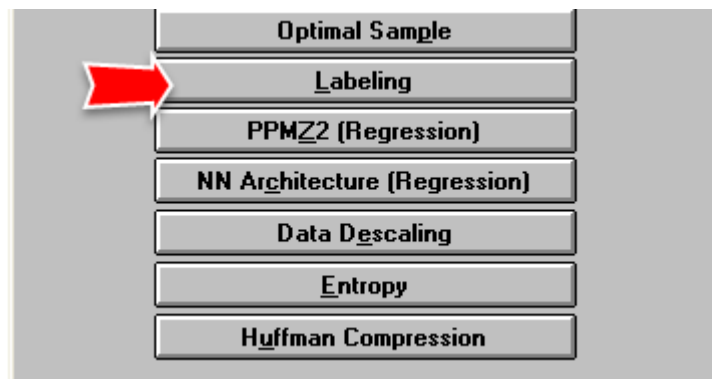
CARS_KOHONEN (Neuron Weights)																	
		? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0	? 0
1	Neuron1	0.224	0.322	0.268	0.187	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.313	0.333	0.354	0.340
2	Neuron2	0.160	0.472	0.275	0.093	0.000	1.000	0.000	0.000	0.275	0.330	0.395	0.000	0.314	0.333	0.354	0.340
3	Neuron3	0.043	0.765	0.182	0.010	0.672	0.000	0.000	0.328	0.187	0.224	0.268	0.321	0.314	0.333	0.353	0.340
4	Neuron4	0.129	0.543	0.265	0.063	0.000	0.000	1.000	0.000	0.187	0.224	0.268	0.321	0.314	0.333	0.353	0.340

3.3.4.6. Proceso de etiquetamiento (Labeling)

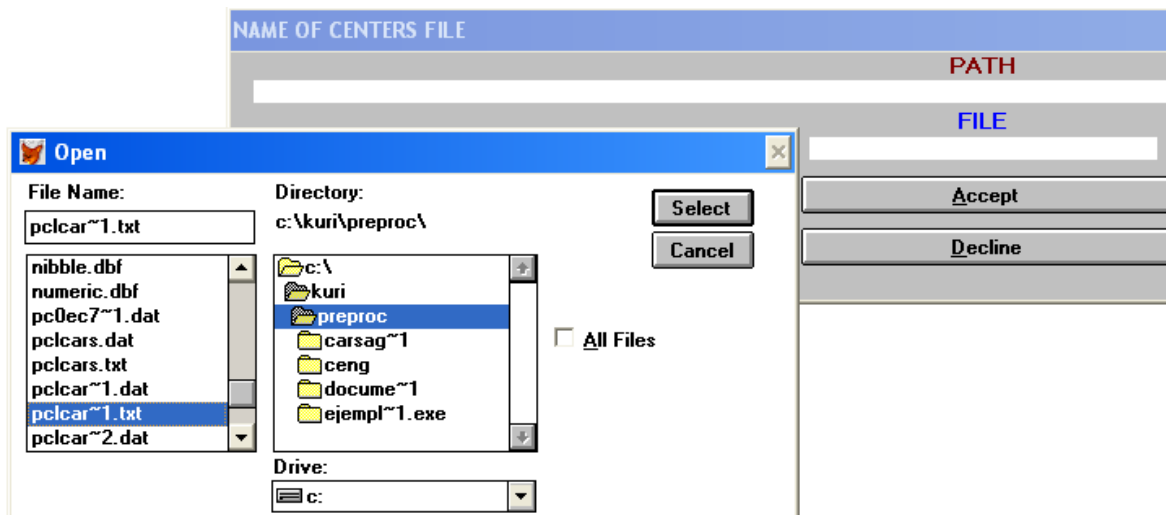
Los datos de las coordenadas anteriores, se guardan en un archivo txt llamado PCLCARS_CENTERS.txt



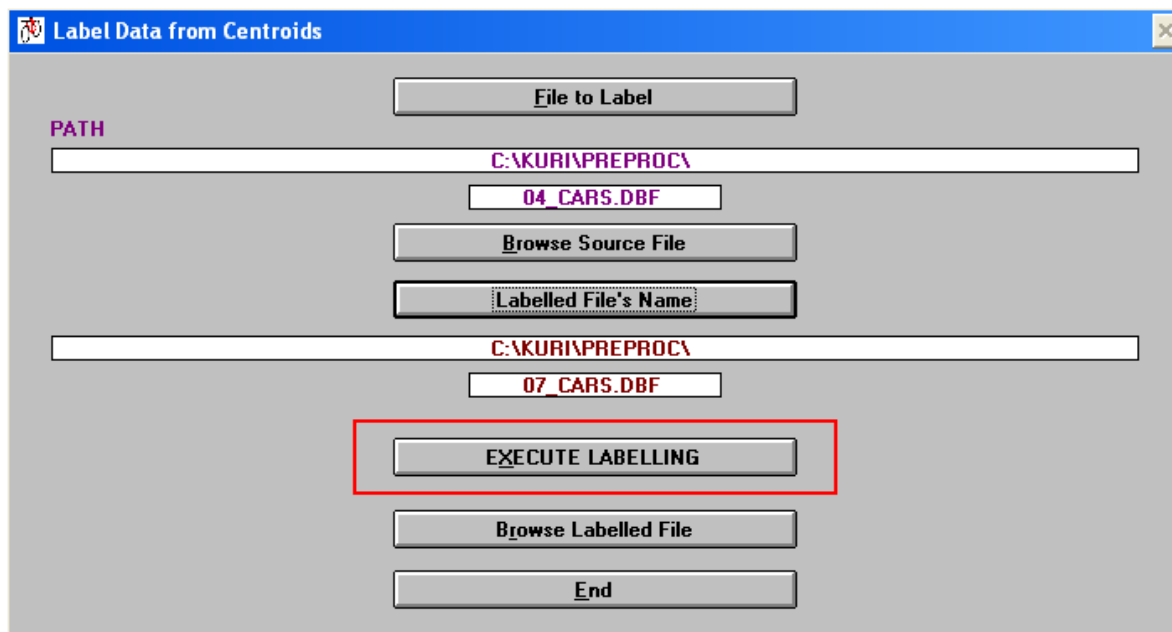
Regresamos al Programa PREPROC y seleccionamos la opción Labeling.



El sistema nos solicita el archivo con los centroides, por lo que seleccionamos el archivo anterior PCLCARS_CENTERS.txt



Ahora le pasamos la información inicial (Preprocesada) y lo guardamos en un archivo llamado 07_CARS.DBF

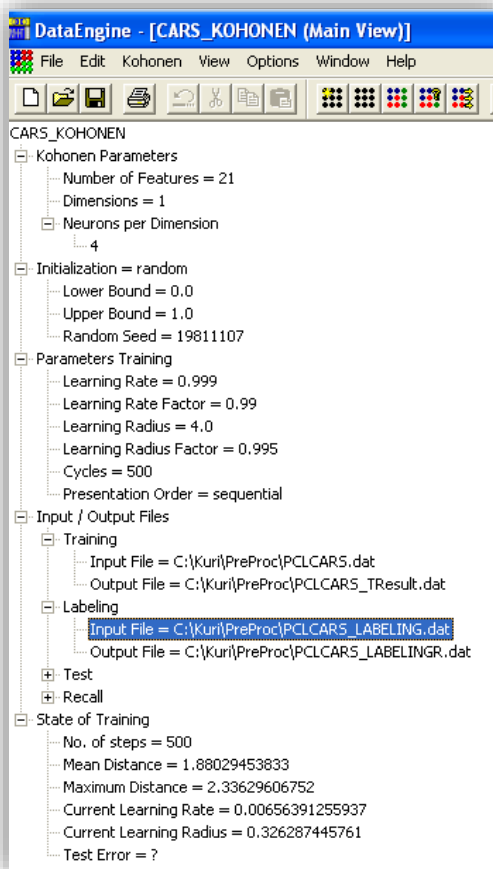


El nuevo archivo contiene los datos iniciales y cuatro variables adicionales que son las que se utilizan para el labeling.

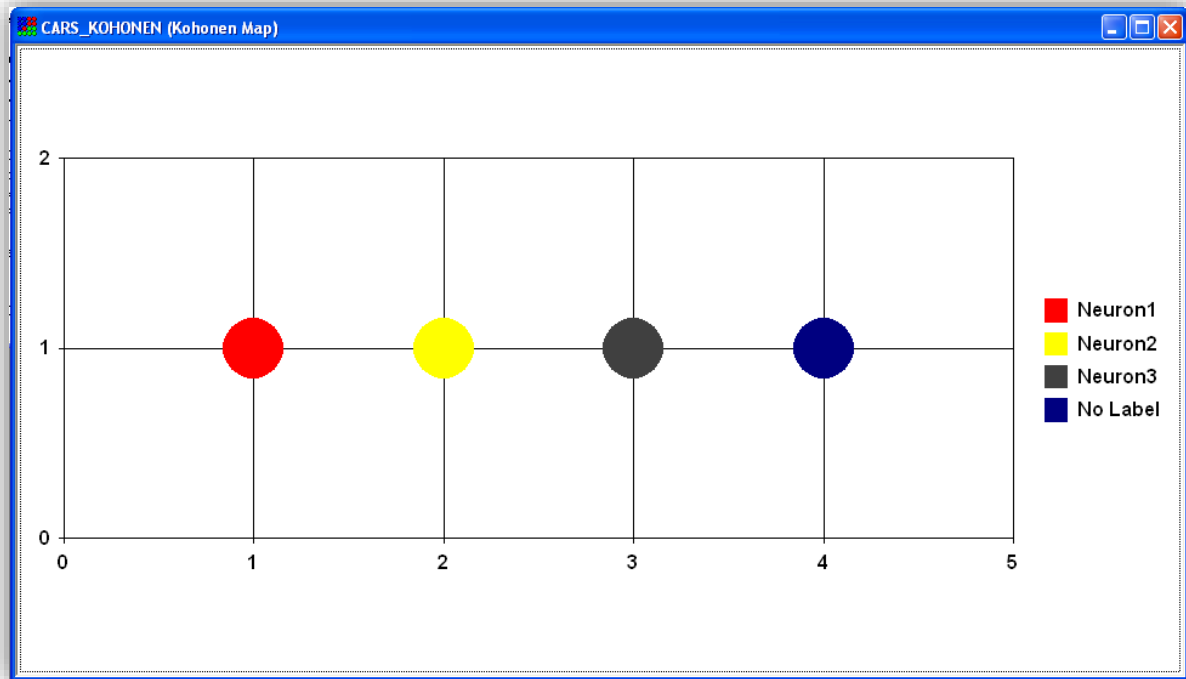
File to Scale				
C1	C2	C3	C4	
0	0	1	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	1	0	
0	0	0	0	

El archivo lo convertimos a TXT con el nombre de PCLCARS_LABELING.TXT y regresamos al programa PREPROC y lo convertimos a formato dat, quedando como PCLCARS_LABELING.DAT

Se modifica la configuración en el archivo CARS_KOHONEN.koh, se agrega el archivo de entrada y salida en la opción de LABELING.

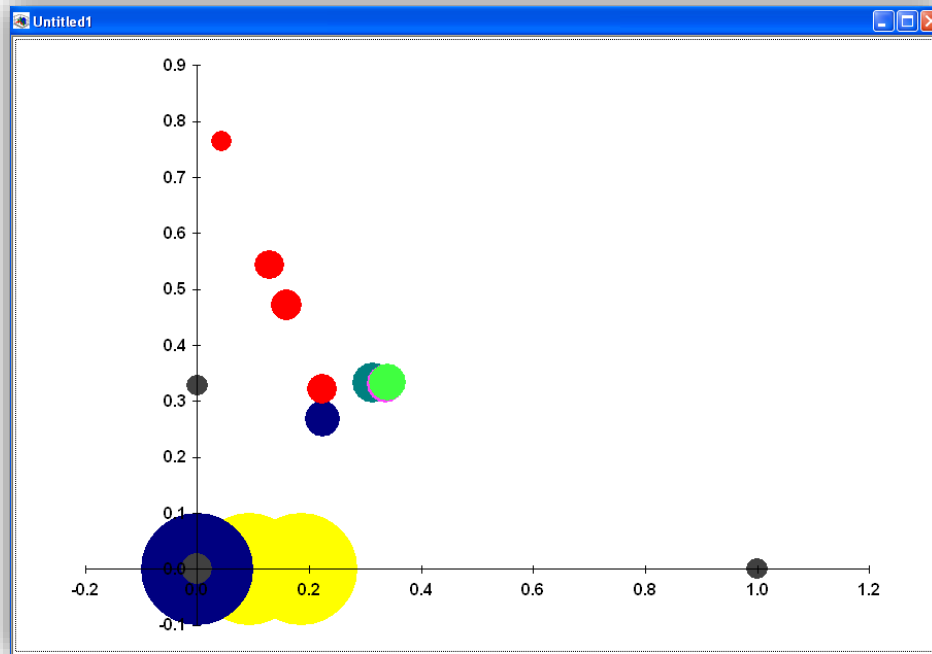


Ahora seleccionamos la opción de Labeling, y el sistema envía un warning de que al menos una neurona no se ha agrupado.



Volvemos a revisar las coordenadas de las neuronas y mostramos algunas de las estadísticas que se generan.

CARS_KOHONEN (Neuron Weights)		?	?	?	?	?	?	?	?	?	?	?	?	?
		0	1	2	3	4	5	6	7	8	9	10	11	12
1	Neuron1	0.224	0.322	0.268	0.187	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.313
2	Neuron2	0.160	0.472	0.275	0.093	0.000	1.000	0.000	0.000	0.275	0.330	0.395	0.000	0.314
3	Neuron3	0.043	0.765	0.182	0.010	0.672	0.000	0.000	0.328	0.187	0.224	0.268	0.321	0.314
4	No Label	0.129	0.543	0.265	0.063	0.000	0.000	1.000	0.000	0.187	0.224	0.268	0.321	0.314



General Statistics - C:\Kuri\PreProc\CARS_KOHONEN.koh (Neuron Weights)												
		?	?	?	?	?	?	?	?	?	?	?
1	Minimum	0.043	0.322	0.182	0.010	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	Maximum	0.224	0.765	0.275	0.187	0.672	1.000	1.000	0.328	0.275	0.330	0.395
3	Mean Value	0.139	0.525	0.247	0.088	0.168	0.500	0.250	0.082	0.162	0.194	0.233
4	Variance	0.006	0.034	0.002	0.005	0.113	0.333	0.250	0.027	0.013	0.019	0.028
5	Standard Deviation	0.075	0.184	0.044	0.074	0.336	0.577	0.500	0.164	0.116	0.139	0.166
6	Range	0.181	0.443	0.093	0.176	0.672	1.000	1.000	0.328	0.275	0.330	0.395
7	Skewness	-0.417	0.544	-1.945	0.756	2.000	0.000	2.000	2.000	-1.200	-1.199	-1.199
8	Kurtosis	0.776	0.954	3.826	1.056	4.000	-6.000	4.000	4.000	2.324	2.323	2.322
9	Sum	0.556	2.101	0.990	0.353	0.672	2.000	1.000	0.328	0.650	0.778	0.931
10	Sum of Squares	0.094	1.206	0.251	0.048	0.452	2.000	1.000	0.107	0.146	0.209	0.300
11	Number of Values	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
12	Number of missing Values	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

4. Conclusiones

Cuando los centros están suficientemente diferenciados esto nos permite crear clasificadores confiables, pero en este caso se tuvieron muchos problemas para obtener dado que trabajamos con datos categóricos que nos inducen al problema de la indistinguibilidad difusa de los grupos, la distribución de los datos es uniforme y no existe correlación entre las variables.

5. Bibliografía

1. Kuri-Morales, A: Data Base Analysis using a Compact Data Set, IEEE International Congress on big Data.