PRÁCTICA

ADVANCED DATA MINING

- SEPTIEMBRE DE 2018 -

GUILLERMO PEDERNAL SOTO

PRIMERA PARTE: DESCRIPCIÓN DE LOS DATOS

➢ OBJETIVO

El objetivo de esta práctica es solucionar un pequeño caso práctico de banca utilizando las herramientas del entorno SAS. Para ello disponemos de una tabla que servirá como fuente de datos, que analizaremos, trataremos y sobre la que aplicaremos unos modelos para llegar a unas conclusiones tal y como se pedía en el enunciado.

El caso práctico consiste en descubrir el perfil de cliente más propenso a contratar un determinado producto bancario.

➢ ANÁLISIS DATOS

Vamos a comenzar con un preprocesamiento de los datos y el correspondiente análisis.

Los datos se nos presentan en la tabla “bank_additional_full”. Dicha tabla cuenta con 41.188 registros (filas), correspondiendo cada uno al perfil de un cliente determinado y proporcionándonos información sobre sus características mediante 21 variables (columnas).

Estas variables incluyen cosas habituales como edad, trabajo, estado civil, nivel de educación, etc. Y otras más específicas como si el cliente tiene préstamos, como respondió el cliente a otras campañas o el tiempo que ha pasado desde la última comunicación con él.

Por último, la variable dependiente, que nos define cuáles son los clientes que han contratado el producto y cuáles no. Esta variable dependiente es la clave de la cuestión e intentaremos llegar a un modelo lo más acertado posible del cliente tipo que contrata el producto en base a estos datos. Podemos ir más lejos, por ejemplo con una regresión logística y predecir si un cliente va o no a contratar el producto en vista de sus datos.

Una vez terminada esta exploración preliminar vamos a cargar los datos en SAS y hacer una serie de comprobaciones.

```
1
2  ods listing close;
3
4  ods listing gpath="/home/guillermopedernal0/my_courses/GuillermoPedernal/data_output";
5
6
7  libname lib '/home/guillermopedernal0/my_courses/GuillermoPedernal/my_project';
8
9  /* bancos */
10 data bancos (drop = y);
11  set lib.bank_additional_full;
12  if y = "no" then yn = 1; else yn = 0;
13 run;
```

En primer lugar, podemos ver las frecuencias de aparición de cada una de las variables con un “proc freq”.

```
15 /*Data cooking*/
16 proc freq data=bancos; run;
17
```

De este análisis observamos que no hay ausencia de datos (“missings”) y de todos los clientes sabemos si han contratado o no el producto. Más concretamente, un 11,27% de los clientes de la base de datos lo contrataron frente al 88,73% que no lo hizo.

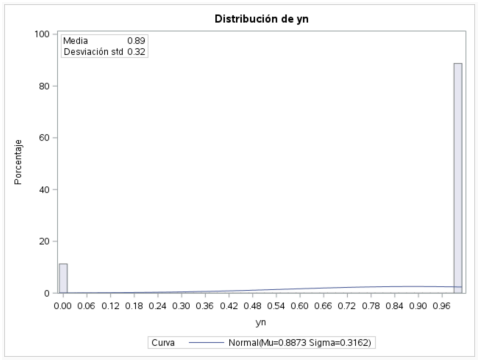
yn	Frecuencia	Porcentaje	Frecuencia acumulada	Porcentaje acumulado
0	4640	11.27	4640	11.27
1	36548	88.73	41188	100.00

A continuación, efectuamos el análisis de normalidad como hemos visto en los ejemplos de clase, para descubrir si la distribución de los clientes entre los que han contratado o no el producto sigue una distribución normal. Tratándose de una variable dicotómica parece evidente que no será así.

```
18 /* normalidad */
19 proc univariate data=bancos normal plot;
20  var yn;
21  qqplot yn / NORMAL (MU=EST SIGMA=EST COLOR=RED L=1);
22  HISTOGRAM /NORMAL(COLOR=MAROON W=4) CFILL = BLUE CFRAME = LIGR;
23  INSET MEAN STD /CFILL=BLANK FORMAT=5.2;
24 run;
```

Con estos resultados:

Test para normalidad				
Test	Estadístico		P valor	
Kolmogorov-Smirnov	D	0.52654	Pr > D	<0.0100
Cramer-von Mises	W-Sq	2667.426	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	13224.93	Pr > A-Sq	<0.0050



La distribución no tiene nada que ver con una normal, con la posibilidad de adoptar sólo dos valores y estando claramente desequilibrada hacia el lado del “no”, tal y como ya vimos en el análisis de frecuencias.

Una transformación logarítmica tal y como vimos en el ejemplo de clase para ver si se asemeja más a la normalidad no tiene sentido al estar jugando con valores absolutos de la variable dependiente (“1” o “0”, “sí” o “no”).

➢ VARIABLES QUE INTERVENDRÁN EN EL MODELO

Para decidir cuáles son las variables que más peso tienen en la decisión de un cliente a la hora de contratar o no el producto y por lo tanto cuáles son las variables que deben aparecer en el modelo, vamos a utilizar un procedimiento de SAS de tipo glmselect, adaptando una macro como la vista en los ejemplos de clase.

Con esta macro vamos a utilizar varias semillas para probar aleatoriamente diferentes variables e interacciones de variable para comparar los errores resultantes de cada uno de estos modelos de prueba hasta llegar al modelo definitivo, por su frecuencia de aparición en el procedimiento aleatorio, su bajo error y sin olvidar que a igualdad de prestaciones el modelo más sencillo suele ser el más acertado.

He optado por guardar la información de los modelos de prueba en un fichero .txt tal y como vimos en uno de los ejemplos de clase.

Las macros a utilizar tendrían un aspecto de este estilo:

```
%macro primera;
data;  file &rutal. ; run;
%do semilla=12355 %to 12365;
ods output   SelectionSummary=modelos;
ods output    SelectedEffects=efectos;
ods output    Glmselect.SelectedModel.FitStatistics=ajuste;
proc glmselect data=bancos plots=all seed=&semilla;
  partition fraction(validate=0.4);
  class job marital education default housing loan contact month
                   day_of_week poutcome;
  model yn = age job marital education default housing loan contact month
                   day_of_week duration campaign pdays previous poutcome emp.var.rate
                   cons.conf.idx euribor3m nr.employed
  / selection=stepwise(select=aic choose=validate) details=all stats=all;
run;
ods graphics off;
ods html close;
data union;i=12;set efectos;set ajuste point=i;run;
data;semilla=&semilla;file &ruta2. mod;set union;put effects @80 nvalue1 @95 semilla;run;
%end;
%mend;

%primera;
```

Y las siguientes, tomando en cuenta las interacciones entre variables:

```
model yn = age job marital education default housing loan contact month
                day_of_week duration campaign pdays previous poutcome emp.var.rate
                cons.conf.idx euribor3m nr.employed
                job*marital job*education job*default job*housing job*loan job*campaign
```

Lamentablemente, me ha resultado imposible ejecutar estas macros en mi equipo, variando cada una de sus partes y probando en todos los programas que tenemos disponibles en el ecosistema SAS. Como entiendo que el objetivo de la práctica no era la programación y no dispongo de más tiempo para dedicarle continuaré con unos resultados inventados.

Todo el listado de modelos obtenido se agrupa en una tabla y se ordena por frecuencia de aparición, de modo que nos quedan unos modelos candidatos. Entre ellos, elegiríamos el modelo "champion" como el que tenga mayor frecuencia, menor error y menor complejidad. Sería interesante en este punto ver las variables que más se repiten entre los modelos candidatos, pues posiblemente serán las variables más determinantes para clasificar a los clientes.

Supondremos que el modelo final es de la siguiente forma:

$Yn = \eta + campaign + job*loan + job*default + \varepsilon$

Apliquemos el procedimiento "glmselect" a este modelo para comprobar qué tal funciona.

```
61 /* modelo */
62 proc glmselect data= bancos plots=all;
63   class loan default job;
64   model yn = campaign job*loan job*default  / selection=none details=all  stats=all;
65 run;
```

Estos son los resultados:

Procedimiento GLMSELECT
Modelo de cuadrados mínimosI (Sin selección)

Análisis de varianza					
Origen	DF	Suma de cuadrados	Cuadrado de la media	Valor F	Pr > F
Modelo	50	158.39229	3.16785	32.92	<.0001
Error	41137	3958.89236	0.09624		
Total corregido	41187	4117.28465			

Raíz MSE	0.31022
Media dependiente	0.88735
R-cuadrado	0.0385
R-Sq Ajust	0.0373
AIC	-55178
AICC	-55178
BIC	-96366
C(p)	51.00000
PRESS	3968.12237
SBC	-95928
ASE	0.09612

Se trata de un modelo con unos parámetros que pueden estar bien para clasificar los clientes entre los que contratarán o no el producto bancario, con un error “ASE” inferior a 0,1.

Con este modelo hemos eliminado la mayor parte de las variables (reducción de dimensionalidad del problema) y nos quedamos con las más determinantes, que permiten por sí solas dar una respuesta al problema con un error lo suficientemente bajo.

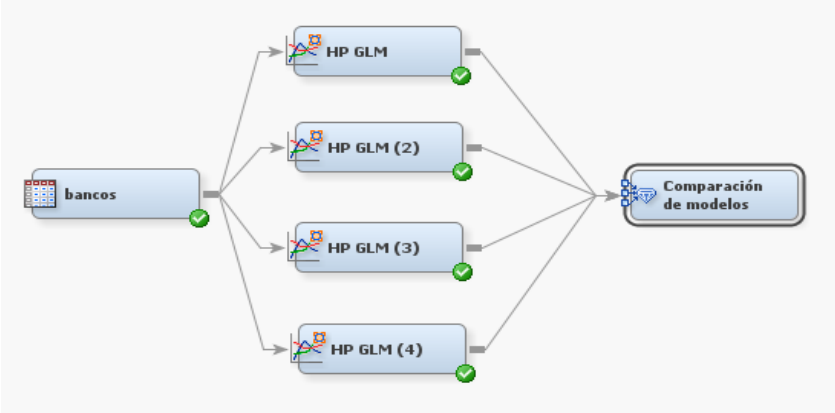
SEGUNDA PARTE: MODELOS DE ENTERPRISE MINER

SAS Enterprise Miner introduce una interfaz gráfica para modelizar sin necesidad de programación, disponiendo nodos en los que se van sucediendo las distintas acciones unidos de forma secuencial. En esta parte de la práctica plantearemos la disposición gráfica de los siguientes modelos.

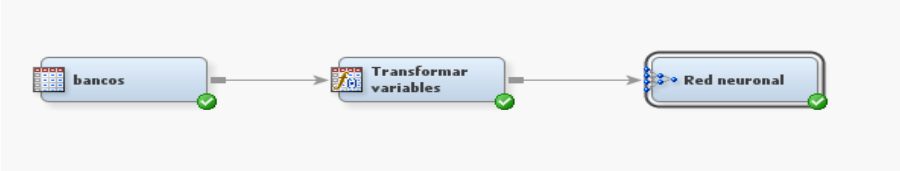
Lamentablemente, trabajar con el SAS Miner ha sido una tarea prácticamente imposible y no he podido cargar la tabla del enunciado. Los siguientes ejemplos se han realizado con una tabla cualquiera de entre las que se incluyen en los ejemplos del programa, con el fin de ilustrar cómo se modelizaría con la herramienta de flujo.

- REGRESIÓN LINEAL

- GLM



➢ REDES NEURONALES



He ejecutado los modelos con la fuente de datos de puebla asignando una variable objetivo y cambiando algunos de los parámetros para llegar a algo funcional. De cada uno de estos flujos se extraerían resultados y en el caso de haber podido cargar los datos obtendríamos una forma de clasificar los clientes entre los que contratarían o no el producto de la práctica según las variables más relevantes para cada modelo, que son precisamente las que buscamos para encontrar el conjunto target de clientes con más posibilidades de contratar.

TERCERA PARTE: MODELO DE REGRESIÓN LOGÍSTICA EN SAS

El objetivo de esta parte es crear un programa en SAS (esta vez con código y no mediante la interfaz gráfica) que modele el problema mediante una regresión logística.

Para ello recuperaremos el análisis previo que realizamos en la primera parte, asumiremos que las variables que escogimos con la simulación de "glmselect" son las adecuadas para el modelo y aplicaremos la macro que vimos en clase y que integra un "proc logistic" variando las semillas. El código es el siguiente:

```
%macro logistic (t_input, vardepen, varindep, interaccion, semi_ini, semi_fin );
ods trace on /listing;
%do semilla=&semi_ini. %to &semi_fin.;

 ods output EffectInModel= efectoslog;/*Test de Wald de efectos en el modelo*/
 ods output FitStatistics= ajustelog; /*"Estadisticos de ajuste", AIC */
 ods output ParameterEstimates= estimalog;/*"Estimadores de parametro"*/
 ods output ModelBuildingSummary=modelolog; /*Resumen modelo, efectos*/
 ods output RSquare=ajusteRlog; /*R-cuadrado y Max-rescalado R-cuadrado*/

 proc logistic data=&t_input. EXACTOPTIONS (seed=&semilla.) ;
  class &varindep.;
  model &vardepen. = &varindep. &interaccion.
     / selection=stepwise details rsquare NOCHECK;
 run;

 data un1; i=12; set efectoslog; set ajustelog; point=i; run;
 data un2; i=12; set un1; set estimalog; point=i; run;
 data un3; i=12; set un2; set modelolog; point=i; run;
 data union&semilla.; i=12; set un3; set ajusteRlog; point=i; run;

 proc append  base=t_models  data=union&semilla.  force; run;
 proc sql; drop table union&semilla.; quit;

%end;
ods html close;
proc sql; drop table efectoslog,ajustelog,ajusteRlog,estimalog,modelolog; quit;

%mend;
```

Esta es la sentencia para ejecutar la macro:

```
104 %logistic (bancos, yn, campaign job loan default, job*loan job*default, 12345, 12350);
```

Y los resultados del modelo:

Resumen de selección paso a paso

Paso	Efecto Introducido	Efecto Eliminado	DF	Número en	Chi-cuadrado de puntuación	Chi-cuadrado de Wald	Pr > ChiSq
1	job		11	1	961.2424		<.0001
2	default		2	2	329.1965		<.0001
3	campaign		41	3	190.5472		<.0001
4	job*default		12	4	37.5915		0.0002

Tabla de Effect por ProbChiSq

Effect(Efecto)	ProbChiSq(Pr > Chi-cuadrado) <.0001	0.0002	Total
campaign	0	12	12
default	12	0	12
job	24	0	24
Total	36	12	48

Efecto	Error estándar
job	0.0156
job	0.0242
default	0.0360
campaign	0.0446

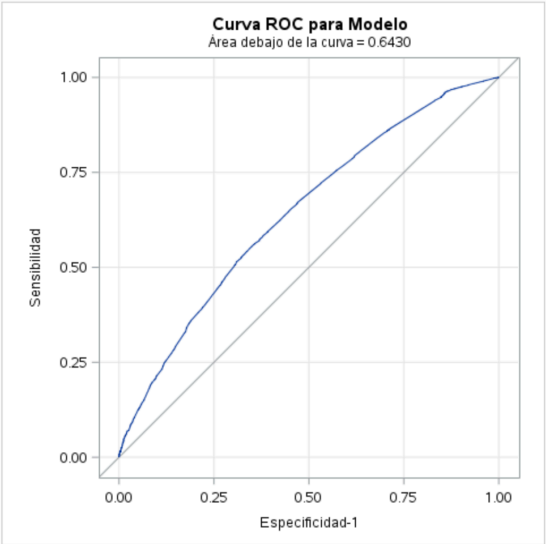
La variable “job” es la más frecuente con un P-valor inferior a 0,0001 además de presentar el error estándar más pequeño.

Con esta sentencia podemos visualizar la tabla de contingencia y la curva ROC del modelo:

```
120 /*Tabla de sensibilidad y especificidad para distintos puntos de corte y Curva ROC*/
121 proc logistic data=bancos desc  PLOTS(MAXPOINTS=NONE);
122  class loan default job;
123  model yn = campaign job*loan job*default /ctable pprob = (.05 to 1 by .05)  outroc=roc;
124 run;
```

Tabla de clasificación

Nivel de prob	Correcto Evento	Correcto No-evento	Incorrecto Evento	Incorrecto No-evento	Porcentajes Correcto	Sensi-bilidad	Especi-ficidad	Falso POS	Falso NEG
0.050	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.100	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.150	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.200	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.250	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.300	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.350	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.400	36548	0	4640	0	88.7	100.0	0.0	11.3	.
0.450	36543	0	4640	5	88.7	100.0	0.0	11.3	100.0
0.500	36539	5	4635	9	88.7	100.0	0.1	11.3	64.3
0.550	36538	11	4629	10	88.7	100.0	0.2	11.2	47.6
0.600	36498	40	4600	50	88.7	99.9	0.9	11.2	55.6
0.650	36467	62	4578	81	88.7	99.8	1.3	11.2	56.6
0.700	35780	373	4267	768	87.8	97.9	8.0	10.7	67.3
0.750	35256	608	4032	1292	87.1	96.5	13.1	10.3	68.0
0.800	35038	659	3981	1510	86.7	95.9	14.2	10.2	69.6
0.850	31402	1360	3280	5146	79.5	85.9	29.3	9.5	79.1
0.900	18934	3146	1494	17614	53.6	51.8	67.8	7.3	84.8
0.950	2252	4525	115	34296	16.5	6.2	97.5	4.9	88.3
1.000	0	4640	0	36548	11.3	0.0	100.0	.	88.7



Los resultados no son maravillosos, aunque en la tabla de contingencia se obtiene un 88,7% de posibilidades de valor correcto. La curva ROC queda muy cerca de la bisectriz del cuadrante, lo que indica que el modelo no es ideal pero tratándose de un ejemplo con variables que hemos cogido como suposición puede ser admisible.

CUARTA PARTE: SELECCIÓN CLIENTES

El objetivo de esta parte es poner en uso el modelo para una aplicación práctica. Se nos pide hacer una selección del 10% de los clientes que se encuentren en el grupo más propenso ("target") a contratar el producto y un 5% adicional tomado aleatoriamente de entre todos.

Para ello tenemos que definir el grupo "target". Del modelo de regresión logística hemos concluido que la variable "job" es la más determinante en nuestra simulación para determinar si un cliente contrata o no nuestro producto. Por simplificar, examinaremos la tabla que cruza los empleos con la variable dependiente, estableciendo el "target" como el grupo de empelo que englobe la mayoría de estos resultados afirmativos.

Tabla de job por yn

job	yn		
	0	1	Total
admin.	1352	9070	10422
blue-collar	638	8616	9254
entrepreneur	124	1332	1456
housemaid	106	954	1060
management	328	2596	2924
retired	434	1286	1720
self-employed	149	1272	1421
services	323	3646	3969
student	275	600	875
technician	730	6013	6743
unemployed	144	870	1014
unknown	37	293	330
Total	4640	36548	41188

Como podemos ver, hay tres empleos que engloban casi dos tercios de los "síes". Tomaremos por lo tanto como grupo "target" aquellos clientes que tengan empleos en la categoría "admin." + "blue-collar" + "technician".

El siguiente paso es seleccionar aleatoriamente el 10% de los clientes que pertenecen a dicho "target". Para ello, usaremos una sentencia SQL mediante el "proc SQL".

```
PROC SQL;
CREATE TABLE WORK.query AS
SELECT age , job , marital , education , 'default'n , housing , loan , contact , 'mon
FROM _TEMP0.bank_additional_full
WHERE job = 'admin.' or job = 'blue-collar' or job='technician';
RUN;
QUIT;
```

El grupo “target” que hemos definido cuenta con 26.419 clientes. De entre ellos seleccionaremos aleatoriamente un 10% (2.642 clientes), además de un 5% escogido de entre todos los clientes (2.059 clientes).

Para ello utilizaríamos un código de este estilo, continuando con el “proc SQL”:

```
143 proc SQL outobs=2059;
144     create table clientes as
145     select *
146     from bancos
147     order by ranuni(0);
148 run;
149 quit;
```

Que nos daría estos resultados:

N° total de filas: 2059 N° total de columnas: 21 Filas 1-100

	age	job	marital	education	default	housing
1	45	services	married	professional.course	no	yes
2	38	blue-collar	married	basic.4y	no	no
3	34	services	married	high.school	no	yes
4	38	blue-collar	married	basic.9y	no	no
5	38	services	married	basic.9y	no	yes
6	29	management	married	university.degree	no	no
7	28	services	single	high.school	no	yes
8	36	admin.	single	university.degree	no	yes
9	31	admin.	married	university.degree	no	yes
10	56	technician	married	professional.course	no	yes

+

N° total de filas: 2642 N° total de columnas: 21 Filas 1-100

	age	job	marital	education	default	housing
1	36	admin.	married	high.school	no	no
2	41	blue-collar	married	basic.6y	no	no
3	36	blue-collar	married	basic.6y	no	yes
4	35	technician	married	unknown	no	yes
5	45	technician	divorced	professional.course	no	yes
6	33	technician	single	university.degree	no	no
7	33	technician	single	professional.course	no	yes
8	44	technician	single	professional.course	no	yes
9	42	blue-collar	married	basic.4y	no	yes
10	29	technician	single	university.degree	no	yes

Para que el ejercicio práctico hubiera tenido más sentido deberíaos haber añadido un identificativo a cada una de las observaciones (por ejemplo, un ID númerico único para cada cliente), lo que nos permitiría con la suma de las 2 tablas obtenidas formalizar el grupo de clientes al que iría dirigida la campaña para intentar contratar el producto.

➢ CONCLUSIONES FINALES

Soy consciente de que he simplificado al máximo los ejercicios que se pedían en la práctica, especialmente con el SAS Miner (una odisea hacerlo funcionar) y las macros de glm (de nuevo, una lucha de la que no sacaba nada).

He tratado de explicar los procedimientos y llegar hasta donde he podido en cada uno de los apartados que se pedían, con las dificultades que han supuesto trabajar con las versiones libres de los programas además de no conocer el lenguaje más que de los ratitos que hemos visto en 4 clases.

Por último, el otro gran problema ha sido el plazo, con las clases metiéndose en agosto que era un mes que a priori teníamos libre y que por lo tanto ya había cargado de trabajo (y algo de vacaciones, ¡claro!).

Con todo, he elaborado este documento que es lo mejor que he podido hacer en el tiempo del que he dispuesto antes del plazo, sin poder ampliar conocimientos y con un mes entre medias desde que vimos las clases.

Espero que sea suficiente, desde luego está hecho con la mejor intención. ¡Gracias!

En Madrid, a 2 de septiembre de 2018.

Guillermo Pedernal Soto