5 DE SEPTIEMBRE DE 2018

PRÁCTICA
ADVANCED DATA MINING

JORGE DEBRÁN PÉREZ

DESCRIPCIÓN Y ANÁLISIS DE LOS DATOS

Lo primero que se observa en la descripción de las variables del *dataset* es que existen dos variables que se generan después de la llamada: *duration* y *campaign*. La primera de ellas será descartada y la segunda bastará con restarle uno.

A continuación, se pasa a comprobar los *missings* y los repetidos:

Procedimiento MEANS

Variable	Etiqueta	N	N Miss
age		41188	0
duration	last contact duration, in seconds	41188	0
campaign	number of contacts performed during this campaign and for this client	41188	0
pdays	number of days that passed by after the client was last contacted from a previous campaign	41188	0
previous	number of contacts performed before this campaign and for this client	41188	0
emp_var_rate	employment variation rate - quarterly indicator	41188	0
cons_price_idx	consumer price index - monthly indicator	41188	0
cons_conf_idx	consumer confidence index - monthly indicator	41188	0
euribor3m	euribor 3 month rate - daily indicator	41188	0
nr_employed	number of employees - quarterly indicator	41188	0

Ilustración 1. Número de *missings* en el *dataset*

Se observa que hay 12 valores repetidos por lo que se obtiene una muestra final de 41176 y 20 variables.

A continuación, se pasa a estudiar cada una de las variables para su procesamiento. Para ello se va utilizar el procedimiento *ttest* para variables continuas, ya que facilita un histograma y *box plot* de una manera sencilla, y para las variables categóricas se va a utilizar una macro para generar *bar plots* por cada una de las variables.

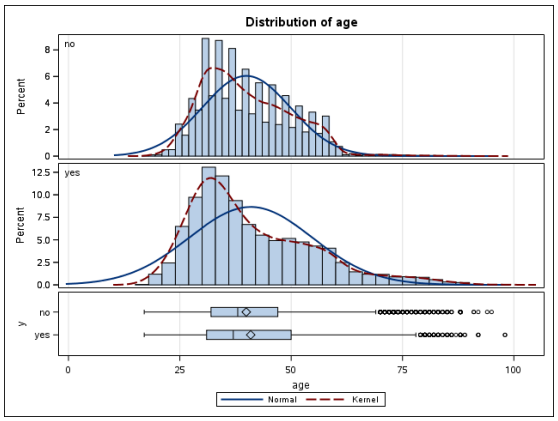

Ilustración 2. Análisis de la variable *age*

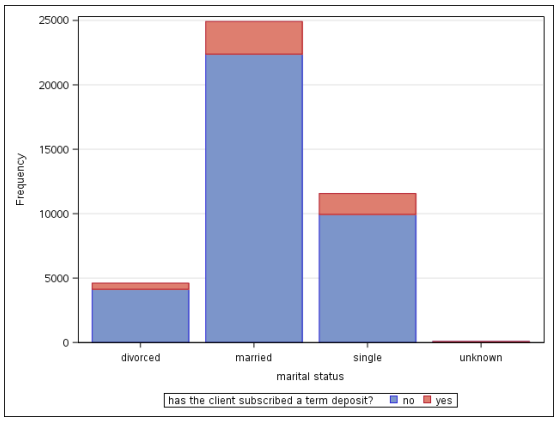

Ilustración 4. Análisis de la variable *marital*

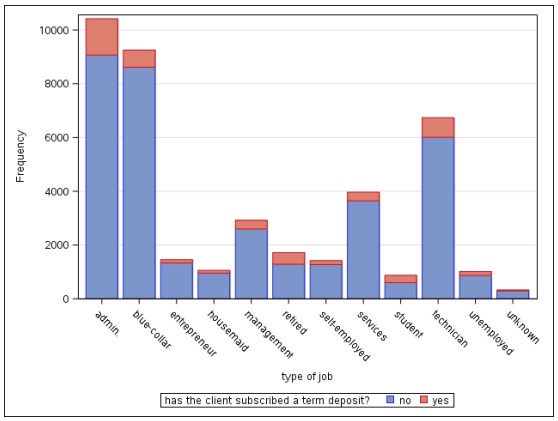

Ilustración 3. Análisis de la variable *job*

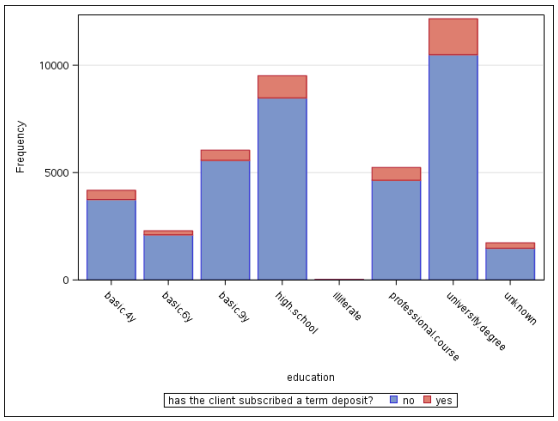

Ilustración 5. Análisis de la variable *education*

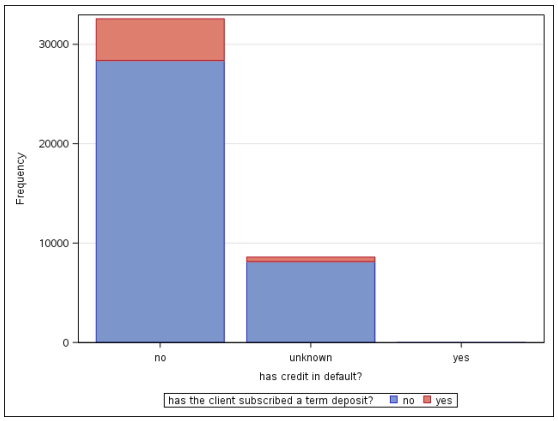

Ilustración 6. Análisis de la variable *default*

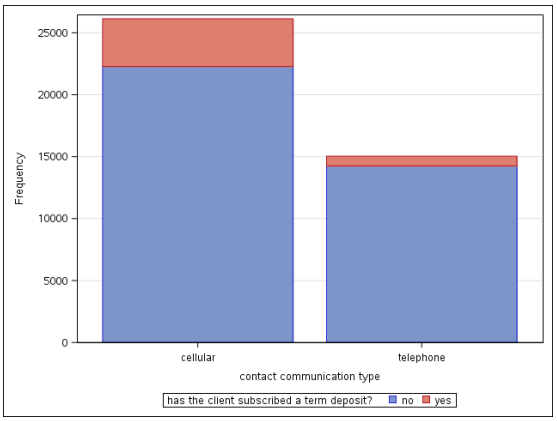

Ilustración 9. Análisis de la variable *contact*

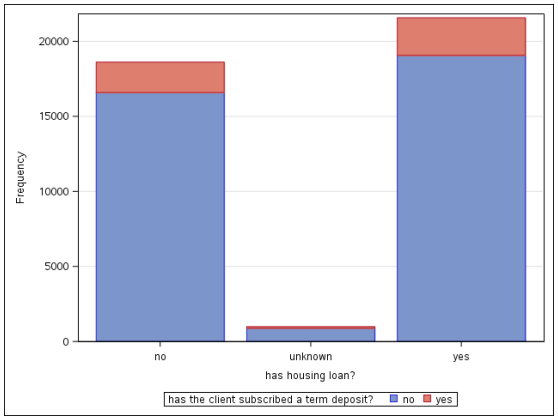

Ilustración 7. Análisis de la variable *housing*

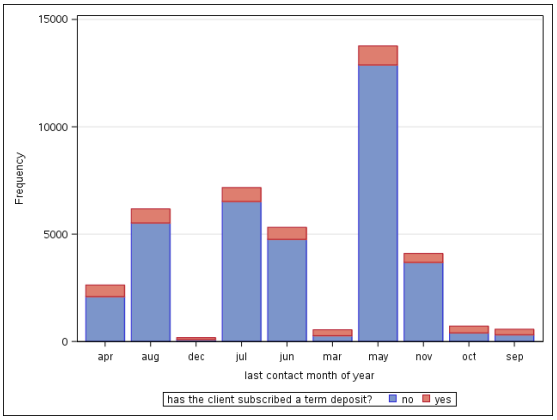

Ilustración 10. Análisis de la variable *month*

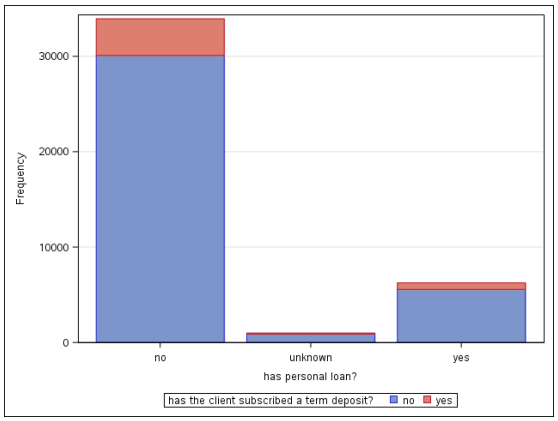

Ilustración 8. Análisis de la variable *loan*

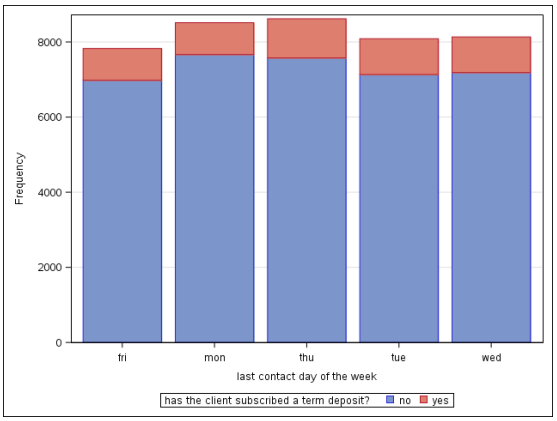

Ilustración 11. Análisis de la variable *day_of_week*

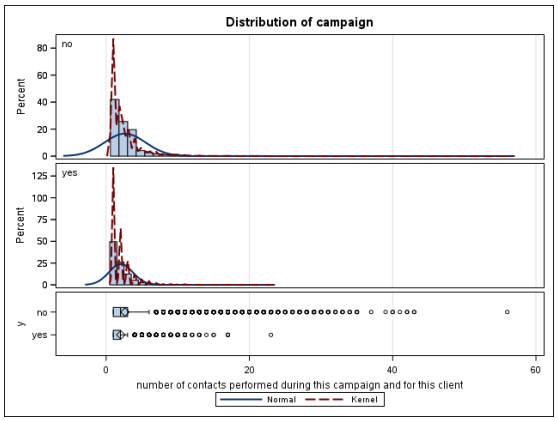

Ilustración 12. Análisis de la variable *campaign*

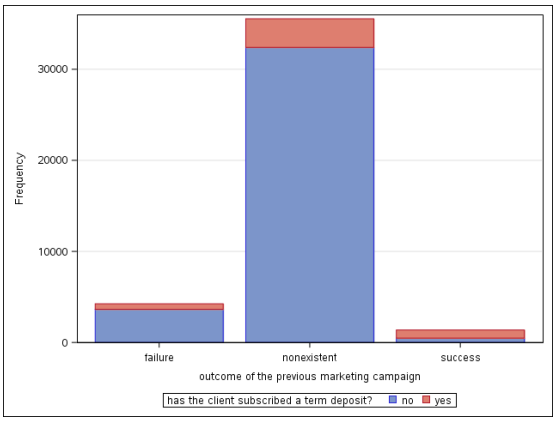

Ilustración 15. Análisis de la variable *poutcome*

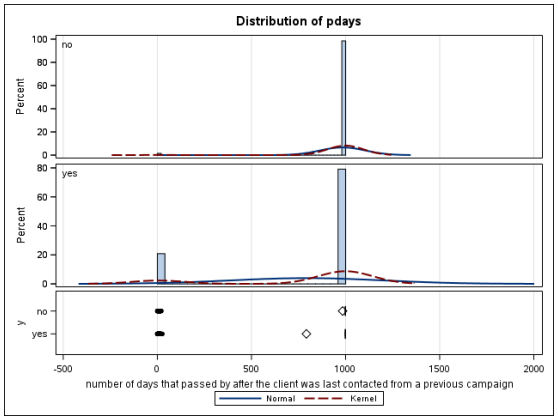

Ilustración 13. Análisis de la variable *pdays*

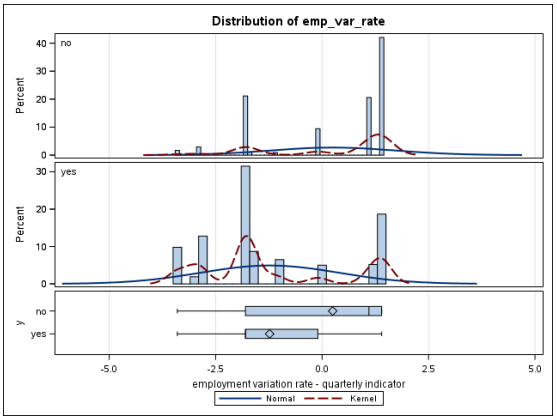

Ilustración 16. Análisis de la variable *emp.var.rate*

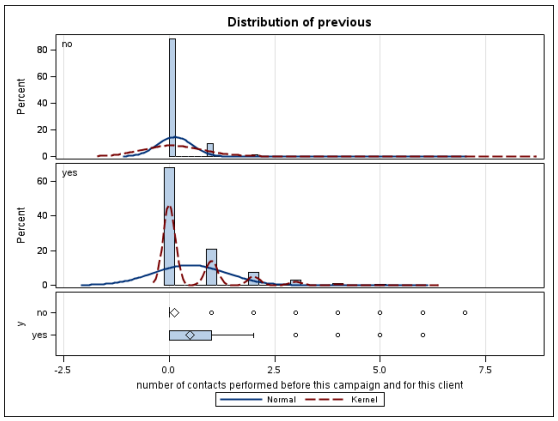

Ilustración 14. Análisis de la variable *previous*

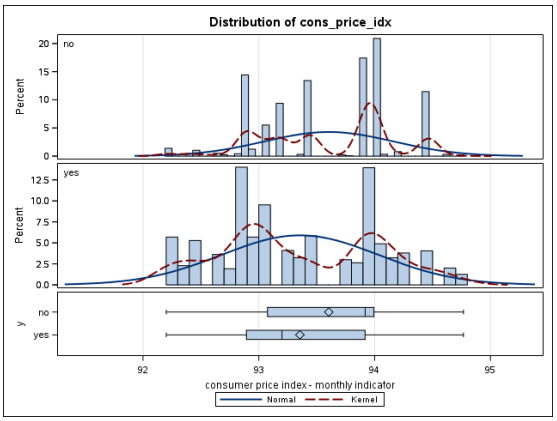

Ilustración 17. Análisis de la variable *cons.price.idx*

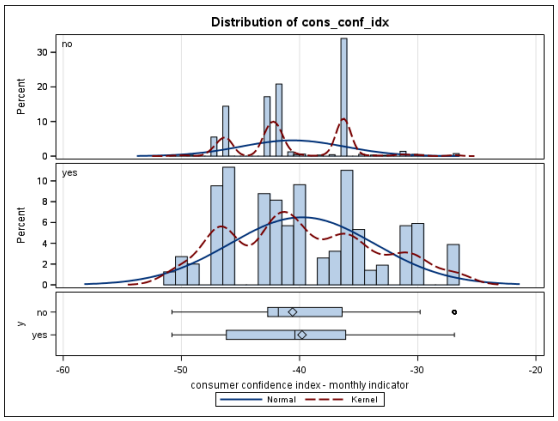

Ilustración 18. Análisis de la variable *cons.conf.idx*

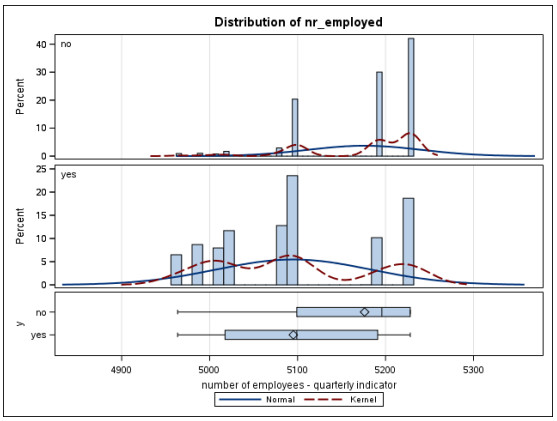

Ilustración 20. Análisis de la variable *nr.employed*

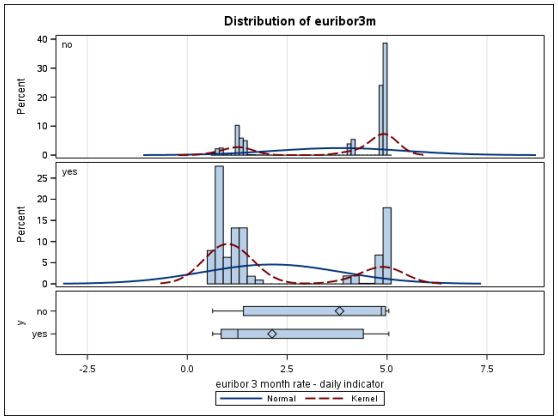

Ilustración 19. Análisis de la variable *euribor3m*

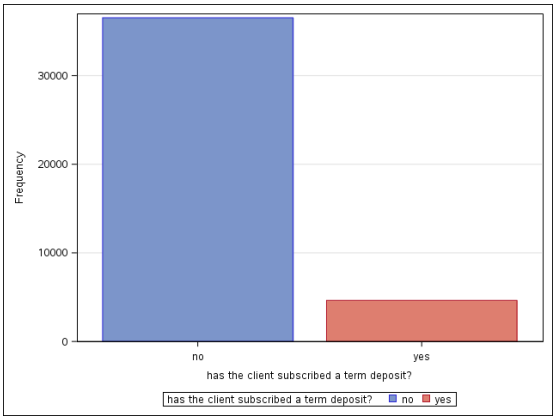

Ilustración 21. Análisis de la variable *y*

Analizando cada una de las variables se sacan las siguientes conclusiones:

- *age*: se puede generar una variable categórica con 8 niveles sin perder en exceso mucho detalle.
- *job*: tiene un error en uno de sus valores (“admin.”).
- *default*: no se tendrá en cuenta debido a la distribución de sus valores.
- *month*: se puede generar una variable indicando el trimestre del año aportando valor al *dataset*.
- *campaign*: los valores por encima de 3 se han de agrupar.
- *pdays*: se puede generar una variable categórica de 5 niveles, agrupando por semanas o no contactado.
- *previous*: se puede agrupar en contactado y no contactado anteriormente.
- *emp.var.rate*, *cons.price.idx*, *cons.conf.idx*, *euribor3m* y *nr.employed*: se pueden agrupar en diferentes niveles cada una de ellas.

Por último, se observa que la variable dependiente (*y*) no se obtiene una muestra cuya distribución se asemeje a una distribución normal y debido a que es una variable dicotómica no se puede aplicar ninguna transformación. Además, por esta razón se van a utilizar solo los modelos de regresión logística y red neuronal y se descarta la regresión lineal ya que no se puede aplicar y el modelo GLM ya que se utilizaría una regresión binomial binaria que sería lo mismo que una regresión logística.

MODELOS DE ENTERPRISE MINER

REGRESIÓN LOGÍSTICA

Para poder utilizar de la forma más sencilla la macro facilitada, se ha utilizado el procedimiento *glmmod* para generar todas las variables *dummy* tanto de las ya existentes como de las interacciones que se quieren comprobar.

En un primer intento se ha intentado trabajar con todas las variables y todas sus interacciones generando un *dataset* con 3782 variables, el cual ha sido imposible de utilizar. Por ello se ha reducido las interacciones dejándolo en un *dataset* con 348 variables, en este caso se ha podido trabajar aunque de forma muy costosa.

Una vez analizado los resultados devueltos por la macro se selecciona las variables para el modelo final. Las variables finales seleccionadas son: *emp_var_r*euribor3mC 1 2*, *nr_employedCat 1* y *pdaysCat 5*, gracias a las cuales se obtiene la siguiente tabla de sensibilidad y especificidad y su curva ROC:

Tabla de clasificación

	Correcto		Incorrecto		Porcentajes				
Nivel de prob	Evento	No-evento	Evento	No-evento	Correcto	Sensi-bilidad	Especi-ficidad	Falso POS	Falso NEG
0.050	4639	0	36537	0	11.3	100.0	0.0	88.7	.
0.100	2981	29557	6980	1658	79.0	64.3	80.9	70.1	5.3
0.150	2302	33607	2930	2337	87.2	49.6	92.0	56.0	6.5
0.200	2269	33713	2824	2370	87.4	48.9	92.3	55.4	6.6
0.250	2269	33713	2824	2370	87.4	48.9	92.3	55.4	6.6
0.300	2269	33713	2824	2370	87.4	48.9	92.3	55.4	6.6
0.350	2269	33713	2824	2370	87.4	48.9	92.3	55.4	6.6
0.400	934	36095	442	3705	89.9	20.1	98.8	32.1	9.3
0.450	873	36165	372	3766	90.0	18.8	99.0	29.9	9.4
0.500	873	36165	372	3766	90.0	18.8	99.0	29.9	9.4
0.550	873	36165	372	3766	90.0	18.8	99.0	29.9	9.4
0.600	873	36165	372	3766	90.0	18.8	99.0	29.9	9.4
0.650	873	36165	372	3766	90.0	18.8	99.0	29.9	9.4
0.700	873	36165	372	3766	90.0	18.8	99.0	29.9	9.4
0.750	0	36537	0	4639	88.7	0.0	100.0	.	11.3
0.800	0	36537	0	4639	88.7	0.0	100.0	.	11.3
0.850	0	36537	0	4639	88.7	0.0	100.0	.	11.3
0.900	0	36537	0	4639	88.7	0.0	100.0	.	11.3
0.950	0	36537	0	4639	88.7	0.0	100.0	.	11.3
1.000	0	36537	0	4639	88.7	0.0	100.0	.	11.3

Ilustración 22. Tabla de clasificación

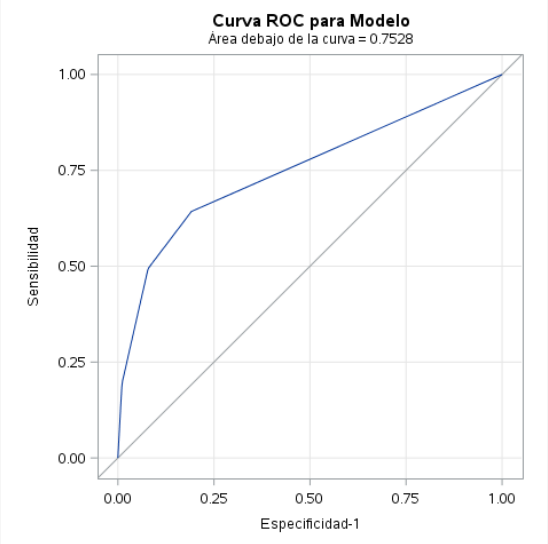

Ilustración 23. Curva ROC

REDES NEURONALES

Para el estudio del modelo de redes neuronales se va a utilizar los nodos de *SAS Miner*. Para ello se han utilizado los nodos de "Red neuronal" y de "HP Red neuronal" de forma conjunta a las técnicas de *k-fold* y de *train-test-validation*. Además, se ha utilizado tanto el *dataset* original como el *dataset* generado en el modelo de regresión logística con las variables *dummy*.

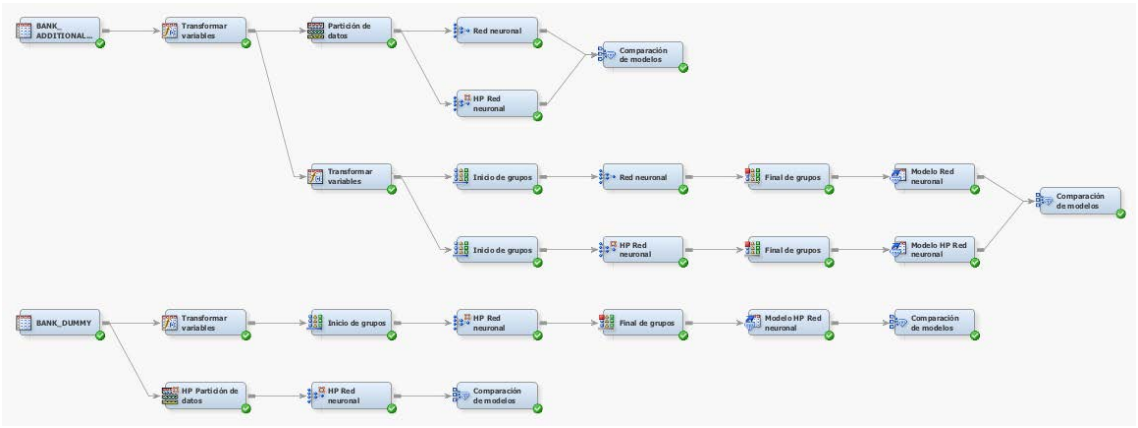

Ilustración 24. Diagrama de redes neuronales

El mejor de los modelos ha sido usando el *dataset* con las variables *dummy* utilizando *k-fold* dando un área en la curva ROC de 0.817.

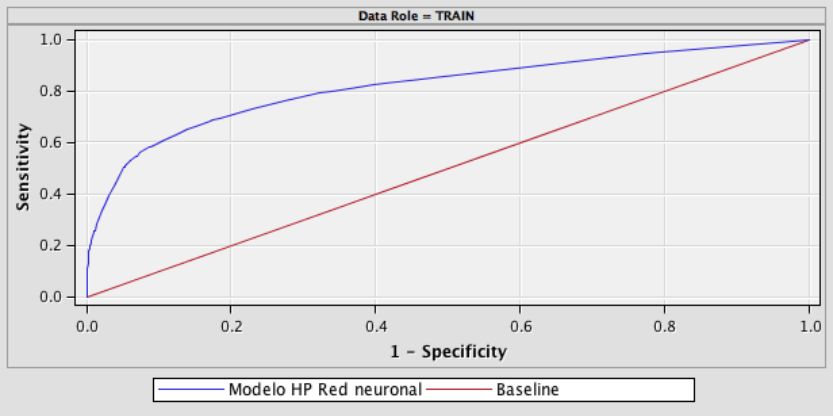

Ilustración 25. Curva ROC

COMPARACIÓN DE MODELOS

Viendo que los nodos de *SAS Miner* tienen suficientes opciones de configuración, su velocidad y su facilidad, se opta por generar varios diagramas comparando diferentes *datasets*, modelos y técnicas de validación.

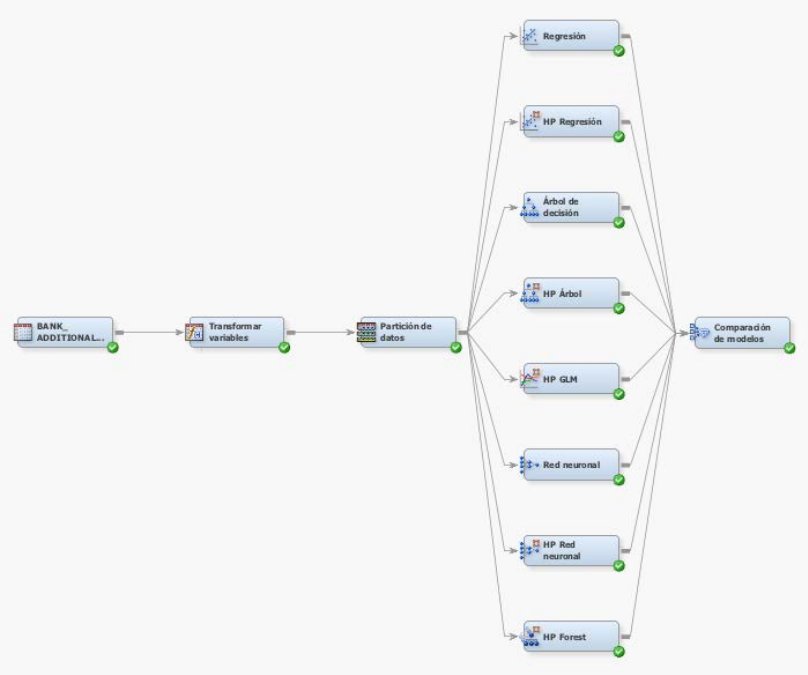

Ilustración 26. Diagrama con *dataset* original y particionado de datos

Ilustración 27. Diagrama con *dataset* original y *k-fold*

Ilustración 28. Diagrama con *dataset* con variables *dummy* y particionado de datos

Ilustración 29. Diagrama con *dataset* con variables *dummy* y *k-fold*

Se ha podido probar los modelos de “HP Red neuronal”, “HP Regresión”, “HP GLM”, “HP Forest”, “HP Árbol”, “Regresión”, “Red neuronal” y “Árbol de decisión” obteniendo los siguientes resultados:

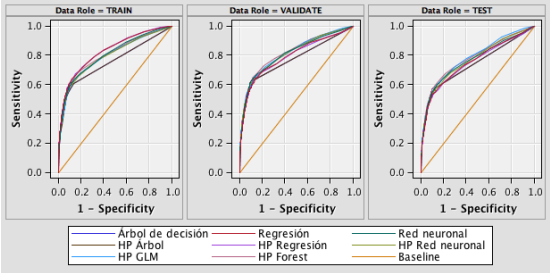

Ilustración 30. Curva ROC de *dataset* original y particionado de datos

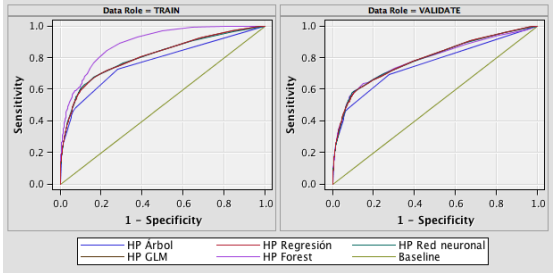

Ilustración 32. Curva ROC de *dataset* con variables *dummy* y particionado de datos

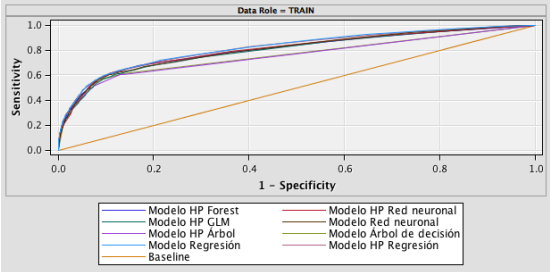

Ilustración 31. Curva ROC de *dataset* original y *k-fold*

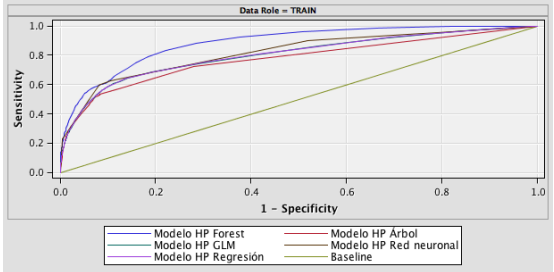

Ilustración 33. Curva ROC de *dataset* con variables *dummy* y *k-fold*

Se observa que los resultados son muy similares en las diferentes opciones probadas. El mejor resultado obtenido ha sido para el *dataset* con variables *dummy*, modelo “HP Forest” y *k-fold* dando un área en curva ROC de 0.886.

Por último, se genera un diagrama para poder comparar los modelos de regresión logística y de red neuronal generados en los pasos anteriores.

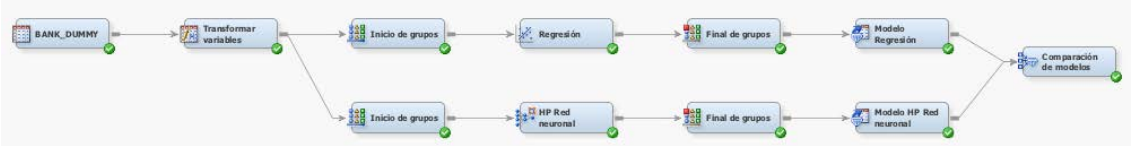

Ilustración 34. Diagrama comparativo entre regresión logística y red neuronal

A continuación, se observa la comparativa de ambos modelos donde se puede ver que el modelo de red neuronal obtiene mejor resultado que el modelo de regresión logística.

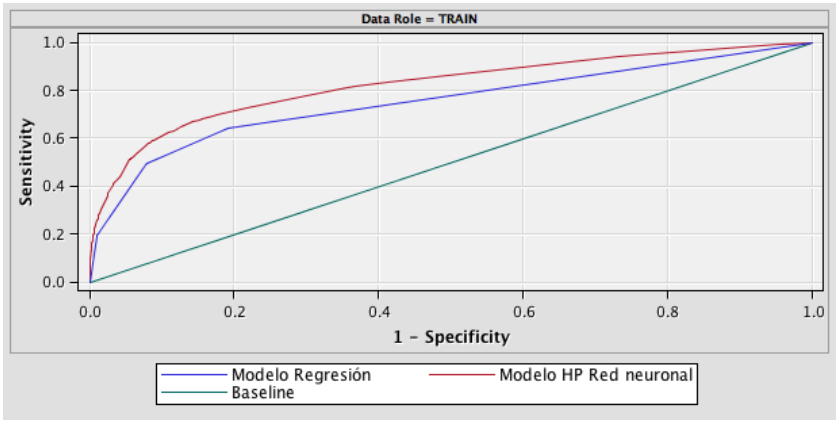

Ilustración 35. Curva ROC

SELECCIÓN DE CLIENTES

Para generar la selección de clientes se va a utilizar el modelo de regresión logística, que pese a que tiene peor puntuación funciona más rápido. Por ello, en primer lugar, se obtendrá el *dataset* original donde se crearán 3 variables con las variables finales seleccionadas en el paso del modelo de regresión logística.

Una vez generado se aplica el modelo de regresión logística añadiendo la opción *output* y su opción *predicted*, gracias a la cual se genera un nuevo *dataset* con todas las variables del *dataset* de entrada y la probabilidad del evento.

```
data bankFinal;
   id = _n_;
   set bankadditionalfull;

   if (emp_var_rate<=-1.8) and (1.3<=euribor3m<4.19) then var1 = 1;
   else var1 = 0;

   if (nr_employed<5099.1) then var2 = 1;
   else var2 = 0;

   if (pdays>=28) then var3 = 1;
   else var3 = 0;
run;

proc logistic data=bankFinal desc PLOTS(MAXPOINTS=NONE) noprint;
   model y(event='yes') = var1 var2 var3 / ctable pprob = (.05 to 1 by .05);
   output out=bankPredicted predicted=y_predicted;
run;
```

Para poder generar el 10% de los mejores clientes, en primer lugar, se ordenará de forma descendente la variable *y_predicted*. Posteriormente, se generará un nuevo *dataset* con el 10% de las primeras filas del *dataset* ordenado.

También se aprovecha para guardar en una variable el número de filas totales del *dataset* para su uso posterior.

```
proc sql noprint;
   create table bankTop10 as
      select * from bankPredicted order by y_predicted desc;
quit;

%let n_cols = &sqlobs;

data bankTop10;
   set bankTop10;
   if _n_ / &n_cols. ge .1 then stop;
run;
```

En primer lugar, para poder generar la selección aleatoria se deberá descartar los clientes generados en la selección anterior y posteriormente se usará el procedimiento *surveyselect* con la opción *method* igual a *srs* (*Simple Random Sampling*) para hacer una muestra aleatoria simple sin reemplazo.

También se usa la opción *sampsize* en vez de *samprate* que sería la opción indicada para la selección por porcentaje, pero debido a que en el *dataset* de entrada se descarta los clientes elegidos anteriormente se debe usar *sampsize* y calcular el porcentaje del 5% sobre las filas del *dataset* original.

```
proc sql noprint;
   select distinct id into :y_list separated by ' ' from bankTop10;
quit;

proc surveyselect data=bankPredicted (where = (id not in (&y_list))) noprint
   out=bankRandom5 method=srs sampsize=%sysevalf(&n_cols. * 0.05, integer);
run;
```

Por último, se unirán ambas selecciones, se ordenará como estaban en su origen y se borrarán las variables auxiliares creadas en pasos anteriores y la variable dependiente.

```
data bankFinal;
   set bankTop10 bankRandom5;
run;

proc sort data=bankFinal
   out=bankFinal(drop = id y var1 var2 var3 _LEVEL_ y_predicted); by id; run;
```