Practice 2.6. Classification Hackathon

Authors: Julia Hernández Elena, Federico Soriano Palacios

Data preprocessing:

Primero comprobamos si había valores ausentes, pero no había. Pasamos a buscar atípicos y encontramos un valor en la variable X8 bastante más grande de la media (en torno a 13) y lo eliminamos.
Por otro lado, miramos con "ggpairs" las correlaciones entre variables. Aquí nos dimos cuenta que X7 y X3 estaban muy correlacionadas (0.991). En un primer lugar, eliminamos la variable X7 del modelo.
Después de está limpieza de datos empezamos con el análisis. Empezamos por modelos lineales y vimos que no nos daban buenos resultados. Añadimos alguna variable al cuadrado que parecía tener relación cuadrática en "ggpairs" pero los resultados no fueron buenos. Pasamos al modelo KNN y al Decision Tree y mejoraba el rendimiento del modelo.
El SMV Radial también nos daba buenos valores. Pero tras probar el MLP nos dimos cuenta de que sin duda es el mejor modelo para el análisis de estos datos.
Tras hacer varias pruebas cambiando parámetros y analizando la importancia de las variables del modelo nos dimos cuenta que el X7 era con diferencia la variable más importante. Por tanto, ya que tenía tanta correlación con X3 decidimos probar a realizar el análisis eliminando X7 en vez de X3.
Finalmente, decidimos hacer diferentes pruebas eliminando variables y ajustando los valores del parámetro de penalización y las iteraciones.

Model comparison:

NULL: En estos casos vimos que el modelo no daba buenos resultados y decidimos no estudiar los conjuntos de entrenamiento y validación

Model	Structure	Inputs	E training	E cross val	E validation
Regresión Lineal Logística		X1, X2, X3, X4, X5, X6, X8, X9, X10	Accuracy: 0.6692 Kappa: 0.3384	Accuracy Kappa 0.6391731 0.2786134	Accuracy: 0.6181 Kappa: 0.2362
K Vecinos más Cercanos	k: 21	X1, X2, X3, X4, X5, X6, X8, X9, X10	Accuracy: 0.7728 Kappa: 0.5449	Accuracy Kappa 0.7465504 0.4926081	Accuracy: 0.7688 Kappa: 0.5373
Análisis Discriminante Lineal		X1, X2, X3, X4, X5, X6, X8, X9, X10	Accuracy: 0.6692 Kappa: 0.3384	Accuracy Kappa 0.6391889 0.2786657	Accuracy: 0.6181 Kappa: 0.2362
		X3, X10	NULL	Accuracy Kappa 0.6242686 0.2483545	NULL
Análisis Discriminante Cuadrático		X1, X2, X3, X4, X5, X6, X8, X9, X10	Accuracy: 0.8027 Kappa: 0.6055	Accuracy Kappa 0.7702862 0.5408876	Accuracy: 0.799 Kappa: 0.5979
Árbol de decisión	cp: 0.0040	X1, X2, X3, X4, X5, X6, X8, X9, X10	Accuracy: 0.9301 Kappa: 0.8602	Accuracy Kappa 0.8601508 0.7202338	Accuracy: 0.8291 Kappa: 0.6582
	cp: 0.0095	X1, X2, X3, X5, X9, X10	Accuracy: 0.8964 Kappa: 0.7926	Accuracy Kappa 0.8501647 0.7002059	Accuracy: 0.8241 Kappa: 0.648
Máquinas de Vectores de Soporte Radiales	c: 55 sigma: 0.030	X1, X2, X3, X4, X5, X6, X8, X9, X10	Accuracy: 0.8939 Kappa: 0.7877	Accuracy Kappa 0.7990682 0.5979477	Accuracy: 0.8291 Kappa: 0.6582
Perceptrón Multicapa	size: 20 decay: 0.4	X1, X2, X3, X4, X5, X6, X8, X9, X10	NULL	Accuracy Kappa 0.8452735 0.6904971	NULL

	size: 20 decay: 0.1	X1, X2, X3, X9, X10	Accuracy: 0.9563 Kappa: 0.9126	Accuracy Kappa 0.8877291 0.7752976	Accuracy: 0.8794 Kappa: 0.7587
	size: 20 decay: 0.1 iteraciones: 200	x1, x2, x7, x9, x10	Accuracy: 0.99 Kappa: 0.98	Accuracy Kappa 0.9500266 0.9000174	Accuracy: 0.9548 Kappa: 0.9096
	size: 5 decay: 0.01 iteraciones: 200	X1, X2, X4, X5, X6, X7, X8, X9, X10	Accuracy: 0.9513 Kappa: 0.9026	Accuracy Kappa 0.8950258 0.7900434	Accuracy: 0.8894 Kappa: 0.7789

*Nota: en caso de que se quieran reproducir los resultados la semilla utilizada ha sido: "`set.seed(2019)`".

Conclusions:

Una vez decidimos que X7 era la variable más importante del conjunto de datos, ajustamos la red neuronal varias veces. Primero eliminamos variables cuya importancia era muy limitada en los gráficos del modelo. En un primer lugar eliminamos X8 y X6. Los valores de precisión y de Kappa aumentaron, por lo que decidimos eliminar también X3, X4 y X5, ya que también parecían poco relevantes. Con este conjunto de variables (x1, x2, x7, x9, x10) ajustamos el número de neuronas y el parámetro de penalización, mediante el uso de validación cruzada y la ayuda del "`ggplot(mlp.fit)+scale_x_log10()`".
Finalmente, ajustamos el número de iteraciones para evitar sobreentrenar el modelo. Obtuvimos muy buenos resultados al comprobar nuestros datos en la página web. Un 94% con todas las variables menos la X3 y un 95% usando X1, X2, X7, X9 y X10.