Informe Assignment 1

ICAI. Machine Learning.

Álvaro Rodríguez González, Pablo Sanz Caperote

Curso 2021-22. Última actualización: 2021-10-24

Índice

Análisis exploratorio de los datos. **3**

Regresión Logística **5**

Caso general 5

Optimizada 5

KNN **5**

Árboles de decisión **6**

Caso general 7

Optimizada 7

SVM **8**

SVM Lineal 8

SVM Radial 10

Redes Neuronales **11**

Caso general 11

Optimizada 11

Random Forest **11**

Comparación de modelos **11**

Conclusiones **11**

Extra **11**

Análisis exploratorio de los datos.

Comenzamos cargando tanto los datos como las librerias que necesitaremos para el desarrollo de nuestros modelos. Tras ello hacemos una primera exploración rápida de estos a través de la tabla de R.

A continuación lo primero que hacemos es ver cómo está estructurado nuestro conjunto de datos y nos damos cuenta de que todas las variables son de tipo numérico (int o num). Esto nos supone un problema para trabajar con los modelos ya que necesitamos que nuestra variable de salida (en este caso sera DIABETES) sea un factor. Por ello transformamos la variable DIABETES en factor, donde también cambiamos los 0's por "No" y los 1's por "Si" ya que si dejasemos los 0's y 1's no estariamos trabajando correctamente con factores.

Tras este cambio, lo que haremos será hacer un summary de la tabla de datos con la finalidad de ver si esta tiene algun valor nulo (NA's). La función nos devuelve que no existe ningún NA, por ello podemos pasar a buscar valores atípicos dentro de nuestras variables.

Para encontrar los outliers lo primero que haremos será una representación gráfica de todas las variables juntas (un ggpairs) con la finalidad de ver como se comportan. En este gráfico observamos que las variables GLUCOSE, BLOODPRESS, SKINTHICKNESS, BODYMASSINDEX e INSULIN tienen valores los cuales podriamos considerar atípicos. Para una mejor valoración haremos boxplots de las diferentes variables.

Tanto en la variable GLUCOSE como en la variable BODYMASSINDEX existen datos que toman el valor 0, lo cual es absolutamente imposible. Por ello, tendremos que decidir que hacer con ellos. A nuestro parecer existen tres posibles opciones, eliminar dichas observaciones (con la consecuente pérdida de información del resto de variables) o sustituirlas por alguna medida de tendencia central como puede ser la media o la mediana. Como no sabemos que opción nos dará un mejor resultado en nuestros modelos lo que haremos será hacer un modelo sencillo que enfrente a la variable salida con la variable a la que queremos evaluar que hacer con los valores atípicos (en nuestro caso dicho modelo será una regresión logística). Finalmente mirando los resultados de la tabla 1 decidimos que los valores iguales a 0 de la variable GLUCOSE los sustituiremos por la mediana mientras que en BODYMASSINDEX lo haremos por la media. Cabe destacar que puede que los valores de accuracy sean más altos que los de la media o mediana pero estamos primando tener más información del resto de variables que tener un poco más de accuracy sobre una variable.

En el resto de variables hemos hecho el mismo proceso y toda la información de los modelos esta en la tabla 1. Si es cierto que hay que hacer especial hincapié en una de las variables, INSULIN. Esta tiene una gran cantidad de 0's que no aparecen como outliers (se debe a que la gran cantidad de 0's afecta a los valores de la media y cuantiles) en los boxplot pero que si que hacen la función de outliers. Debido a que el número de observaciones que tienen 0 en la variable INSULIN representa casi la mitad de la muestra es obvio que se descartará la opción de eliminar dichas observaciones, y por ende solo quedará la opción de sustituir las observaciones por la media o la mediana.

Una vez resuelto el problema de los outliers analizaremos las diferentes variables continuas del conjunto de datos. Usando otra vez el ggpairs nos damos cuenta de que las escalas de nuestras variables son muy diferentes, por ello tendremos que hacer una estandarización. Pero esta se implementará cuando nos pongamos a trabajar con los modelos, por tanto ahora mismo no nos tendremos que preocupar.

Variable	Eliminar Datos	Sust. Media	Sust. Mediana
GLUCOSE	0.7479623	0.7400434	0.7415142
BLOODPRESS	0.6533898	0.6536022	0.6466102
SKINTHICKNESS	0.6673865	0.632392	0.6347352
BODYMASSINDEX	0.6676503	0.6634551	0.6564784
INSULIN	-	0.6536224	0.6504231

Cuadro 1: Comparación accuracy opciones outliers

Antes de iniciar el desarrollo de los modelos debemos estudiar las posibles correlaciones que existen entre nuestras variables con el fin de poder conocer si existen variables que nos predigan lo mismo. Como se puede observar en la imagen inferior la correlación entre nuestras variables es muy baja como para contemplarse cualquier acción, por lo que podemos pasar al siguiente punto.

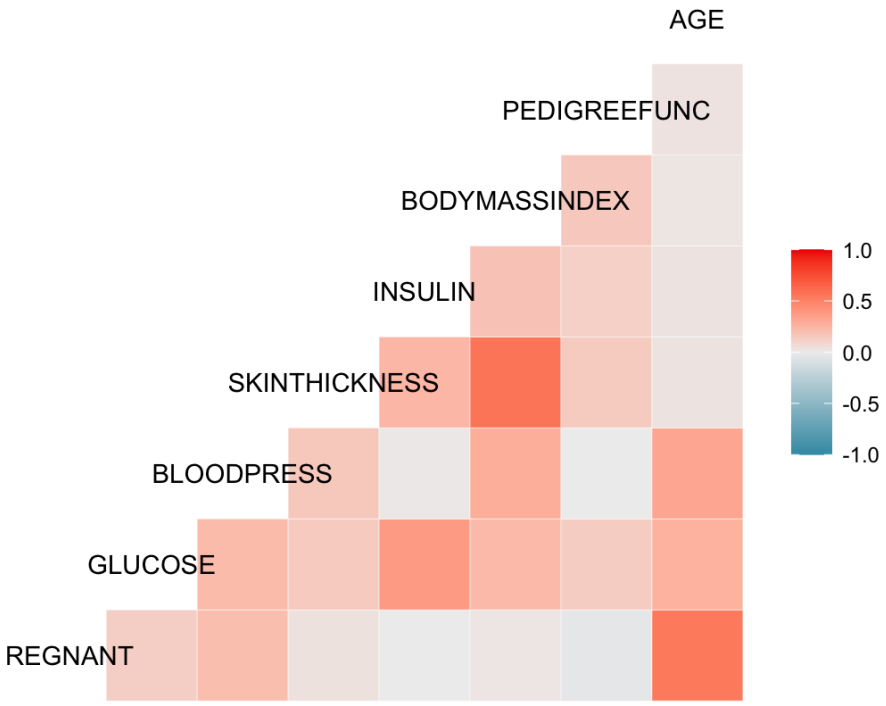

Figura 1: Matriz correlación

Por último antes de comenzar con los modelos miraremos si nuestra clase de salida esta balanceada o no. En caso de no estar deberemos contemplar si la balanceamos o no. Haciendo un table sobre la variable DIABETES observamos que nuestra clase no esta para nada balanceada 70-30. Por ello lo que haremos será mirar si en los modelos, estos son capaces de medirnos bien la salida de "Si".

Haciendo una regresión logística obtenemos que sensitividad tanto en entrenamiento como en test es muy baja, 0.6 y 0.35 respectivamente, por lo que nos vemos obligados a balancear los datos. Para ello usaremos la libreria ROSE a traves de la función ovun.sample. A su vez también definiremos los métodos de control y la partición de datos (80-20) que usaremos para nuestros modelos.

Regresión Logística

Distinguiremos entre dos modelos diferentes, el primero donde trabajaremos sobre todas las variables y veremos cuales son las más importantes y luego el optimizado que solo tomará las variables clave.

Caso general

Nuestro primer modelo de regresión logística nos da un accuracy de 0.746 y un valor de kappa de 0.4925. Pero lo realmente interesante es que el modelo nos da también las variables más importantes, si ejecutamos la función summary obtenemos que las variables PREGNANT con un p-valor de 6.32e-05, GLUCOSE con p-valor menor que 2e-16 , BODYMASSINDEX con p-valor de 8.80e-08 y PEDIGREEFUNC con p-valor de 0.00785 son las más importantes de nuestro modelo.

Una vez obtenidas cuales son las variable más importante podemos pasar a optimizar nuestro modelo.

Optimizada

En este modelo de regresión logística tendremos como inputs a las cuatro variables obtenidas en el modelo anterior y como variable de salida tendremos a DIABETES. Además estableceremos como parametro de control una validación cruzada con 10 folds.

Tras entrenar el modelo obtenemos un accuracy de 0.76 con un valor de kappa de 0.52.

KNN

El modelo KNN se basa en comparar cada uno de los datos con un número k de vecinos más cercanos. Para este modelo se necesita el hiperparámetro k, que indica el número de vecinos cercanos que se utilizan en la comparación. Para obtener el k más óptimo se realiza un barrido con diferentes k hasta obtener los resultados más óptimos.

En la siguiente figura se puede observar el barrido del parámetro k.

Se puede observar que el valor de accurancy más alto se obtiene cuando el k=2. También podemos observar que el valor de k más alto que hemos probado es 40, porque se ve claramente que el accurancy tiene una línea descendente cuanto más alto es el k. Que el valor seccionado sea $k = 2$ puede ocasionar un sobre entrenamiento y por ello vemos también el valor Kappa que en este caso es $Kappa = 0{,}5225$.

Evaluando los resultados más de cerca, tenemos la matriz de confusión:

	No	Sí
No	62	19
Sí	38	81

Cuadro 2: Matriz confusión kNN

Además de esta matriz, vamos a utilizar el indicador de la curva ROC. A continuación vemos la gráfica de la curva ROC para el conjunto de test.

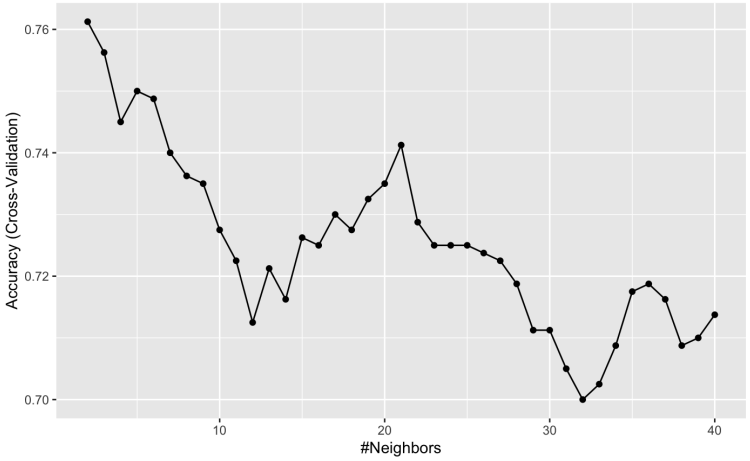

Figura 2: Valores de diferentes k

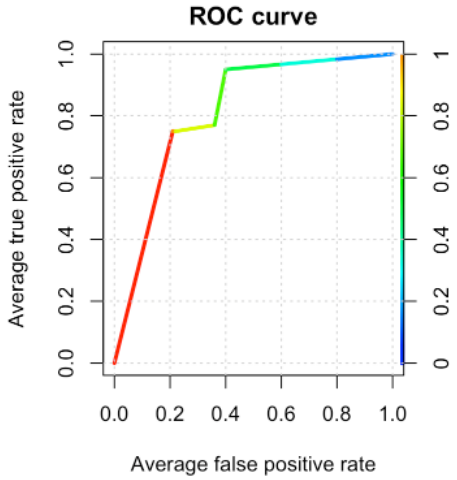

Figura 3: ROC KNN Test

El área bajo la curva tiene un valor de 0.81145 lo que es aceptable.

Árboles de decisión

El modelo de árboles de decisión sigue un consiste en realizar divisiones recursivas. Para ello se analiza la mejor variable para dividir el conjunto de datos y se van obteniendo diferentes ramificaciones que permiten clasificar cada uno de los datos. El número de divisiones que se ejecutan son los diferentes nodos del árbol. En estos modelos el hiperparámetro que se utiliza es uno llamado cp que sirve para penalizar a los árboles que tienen una altura mayor. Si este parámetro tuviera el valor 0, el árbol tendría un gran número de nodos y muy probablemente el modelo estaría sobre entrenado. Por ello, realizaremos un barrido de este parámetro cp yendo desde 0 hasta 0.1 en pasos de 0.001. Además, entrenaremos 3 modelos diferentes que podemos ver a continuación.

Caso general

Este modelo se entrenará con todas las variables del conjunto de datos inicial. A continuación se muestra el gráfico del accurancy variando el cp:

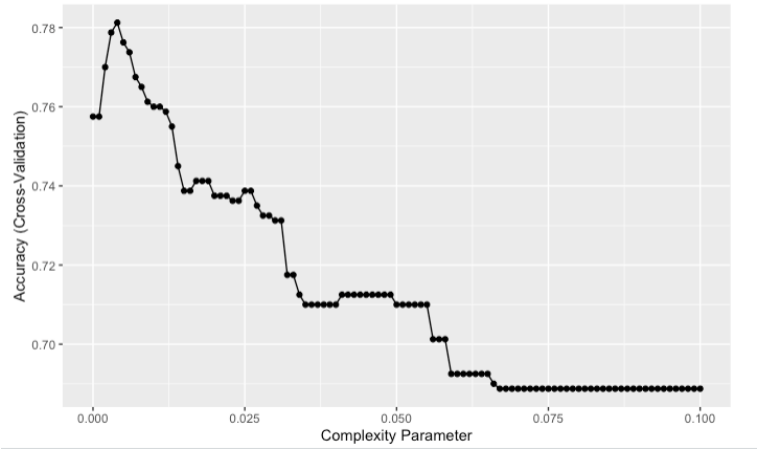

Figura 4: Accurancy y cp

Se puede ver que el valor más alto es $cp = 0{,}004$. Evaluando los resultados más de cerca, tenemos la matriz de confusión:

	No	Sí
No	74	20
Sí	26	80

Cuadro 3: Matriz confusión DecisionTree

Además de esta matriz, vamos a utilizar el indicador de la curva ROC. A continuación vemos la gráfica de la curva ROC para el conjunto de test.

El área bajo la curva tiene un valor de 0.8215 lo que es aceptable.

Optimizada

Pero además de los datos representados anteriormente, este modelo nos proporciona una tabla que indica qué variables son importantes.

Podemos ver que las variables importantes son: Glucose, Age, BodyMassIndex, Pedigreefunc, SkinThinckness y Pregnant. Aunque algunas de ellas tienen más importancia que otras, vamos a coger estas 6 para crear un nuevo modelo. Y analicemos la Accurancy en función del hoperparámetro cp:

Se puede ver que el valor más alto es $cp = 0$. Evaluando los resultados más de cerca, tenemos la matriz de confusión:

Además de esta matriz, vamos a utilizar el indicador de la curva ROC. A continuación vemos la gráfica de la curva ROC para el conjunto de test.

El área bajo la curva tiene un valor de 0.8201 lo que es aceptable.

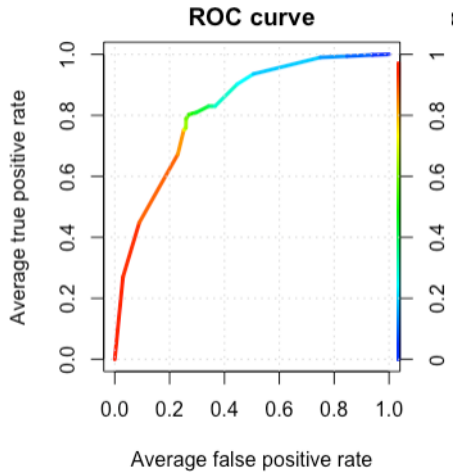

Figura 5: ROC DC Test 1

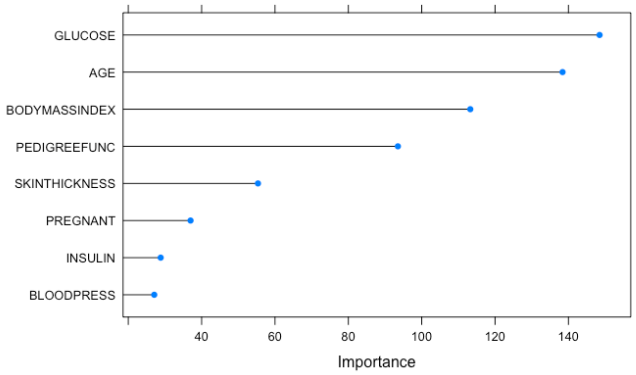

Figura 6: Importancia DT

SVM

Aquí hemos utilizado dos modelos diferentes dentro de la misma familia. El SVM lineal y el SVM Radial. No obstante, para ambos modelos se utiliza un hiperparámetro C. Para calcular cual es el valor más óptimo mostraremos un barrido de la variación del Accurancy con dicho parámetro. Analizaremos cada uno de los 2 casos:

SVM Lineal

Veamos el barrido del parámetro C con este modelo:

	No	Sí
No	69	22
Sí	31	78

Cuadro 4: Matriz confusión DecisionTree2

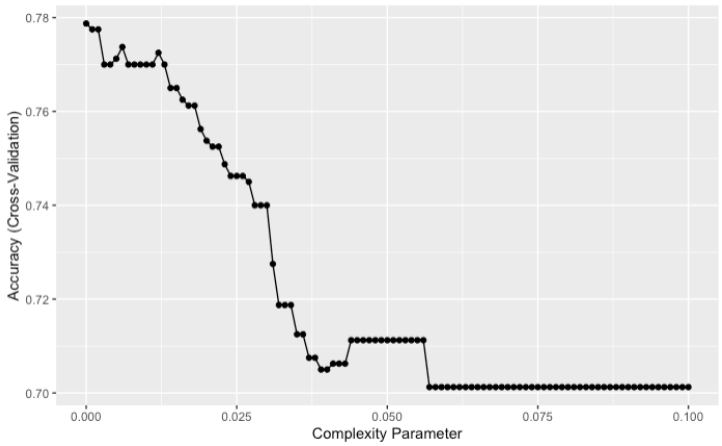

Figura 7: Accurancy y cp

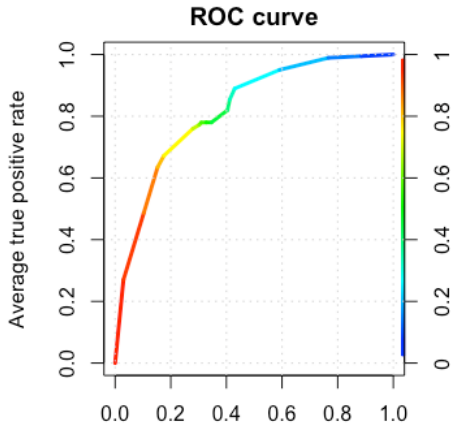

Figura 8: ROC DC Test 2

Se puede ver quelos puntos $C = 0{,}1$ y $C = 0{,}3$ generan los mismos valores de Accurancy pero nosotros nos quedaremos con el primero de ellos, $C = 0{,}1$. Evaluando los resultados más de cerca, tenemos la matriz de confusión:

	No	Sí
No	74	28
Sí	26	72

Cuadro 5: Matriz confusión SVM Lineal

Además de esta matriz, vamos a utilizar el indicador de la curva ROC. A continuación vemos la gráfica de la curva ROC para el conjunto de test.

El área bajo la curva tiene un valor de 0.819 lo que es aceptable.

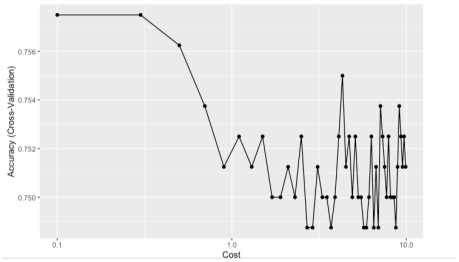

Figura 9: SVM Lineal parámetro

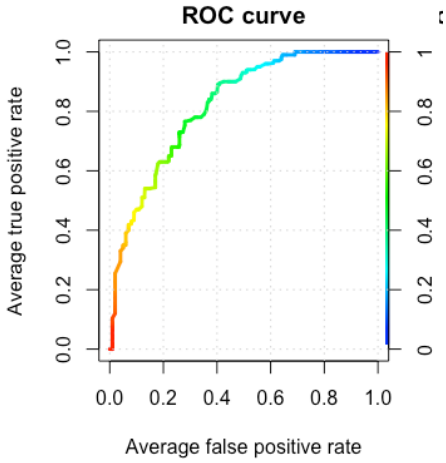

Figura 10: ROC SVM Lineal

SVM Radial

En este modelo no solo tenemos que modifical el parámetro C como ocurría en el DVM Lineal, sino que también tenemos que ver el parámetro Sigma.

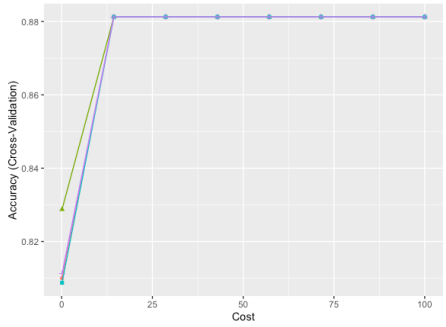

Figura 11: Params SVM Radial

Se puede ver que los diferentes valores de sigma convergen cuando C es grande. Por este motivo vamos a coger la línea que siempre está por encima y que representa el valor $sigma = 450$ y $C = 14{,}5$ Evaluando los resultados más de cerca, tenemos la matriz de confusión:

Además de esta matriz, vamos a utilizar el indicador de la curva ROC. A continuación vemos la gráfica de

	No	Sí
No	100	30
Sí	0	0

Cuadro 6: Matriz confusión SVM Radial

la curva ROC para el conjunto de test.

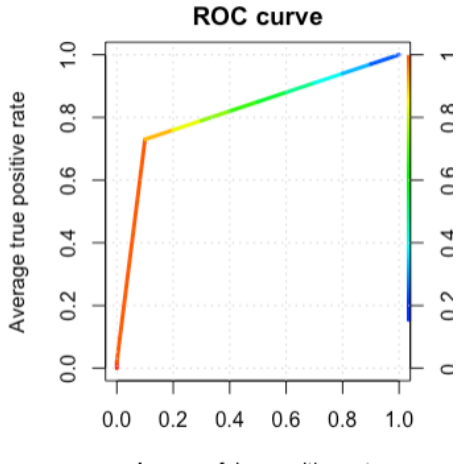

Figura 12: ROC SVM Radial

El área bajo la curva tiene un valor de 0.85 lo que bastante bueno.

Redes Neuronales

Caso general

Optimizada

Random Forest

Comparación de modelos

Conclusiones

Extra