Minería de Datos. Clasificación

Aliaksandra Skrypko

Índice

1. Introducción **3**

2. Análisis preliminar **4**

2.1. Class 4

2.2. Variables factor 5

2.3. Variables Continuas 5

2.4. Conslusiones 10

3. Modelización **11**

3.1. Train-test split 11

3.2. Árbol de clasificación 12

3.3. Random Forest 15

3.4. Perceptrón múltiple 18

3.5. Conclusiones 21

4. Entrenamientos en conjunto completo **22**

5. Conclusiones **23**

1. Introducción

El conjunto de datos utilizado es el `ILPD (Indian Liver Patient Dataset)` extraido del repositorio *UCI Machine Learning* [1].

El conjunto de datos se corresponde con pacientes hepáticos de una región de India. Cada uno de los pacientes de-identificados se clasifica por unos expertos en dos grupos según poseen la enfermedad hepática genérica o no. Los registros se describen según sus características generales (edad, sexo) y resultados de analítica de sangre (valores de bilirrubina, aminotransferasa, albúmina, proteínas, etc.). La lista completa de todas las variables se presenta a continuación.

1. Age: edad
2. Gender: sexo
3. TB: bilirubina total
4. DB: bilirrubina directa
5. Alkphos: fosfotasa alcalina
6. Sgpt: alanina aminotransferasa
7. Sgot: aspartato aminotransferasa
8. TP: proteínas totales
9. ALB: albúmina
10. A/G: ratio albúmina-globulina
11. Clase: clasificación asignada al paciente

El conjunto contiene 583 observaciones, 4 de ellas tienen valores ausentes en la variable AG (ratio albumina-globulina). A pesar de que existen diferentes técnicas para abordar el problema de valores ausentes, se opta por eliminar directamente las 4 observaciones ya que estas no son una porción significativa del conjunto. Entre las opciones existentes están también la predicción de valores ausentes (por métodos kNN o árboles) o imputación a través de la distribución de la variable en cuestión (medias, medianas, etc.).

Notar que por construcción, si la edad del paciente excede 89 años, se transcribe en la base de datos con la edad 90 independientemente del valor exacto. Esta codificación no es un problema para el análisis ya que solamente un registro tiene valor 90 asignado.

Como parte de preprocesamiento de datos se convierten las variables Gender y Class en factores y se recodifica la variable Class para que tenga niveles más fácilmente interpretables: “Si” y “No”. La decodificación de las clases “1” y “2” como pacientes

[1] https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset)

con enfermedad hepática y pacientes sin enfermedad hepática respectivamente se hace de acuerdo con la descripción del conjunto de datos proporcionada por los propietarios de los datos.

2. Análisis preliminar

2.1. Class

El estudio de la variable de clasificación revela que el conjunto de datos no es balanceado: hay menos de 30 % de casos negativos por lo que su predicción va a ser difícil (los modelos no tendrán suficientes casos para generalizar las tendencias). Además, la clasificación de los modelos no podrá ser resumida adecuadamente en una única métrica como *accuracy* y habrá que utilizar otros conceptos como *sensitivity* y *specificity*.

Tener en cuenta que la naturaleza del problema implícitamente tiene poca tolerancia a los falsos negativos (casos marcados por el algoritmo como sanos incorrectamente) y mayor tolerancia a los falsos positivos (casos identificados incorrectamente como enfermos). El razonamiento es el siguiente: el modelo claramente no puede usarse para diagnóstico por si solo (esta es la tarea de un médico), pero puede ayudar al médico a marcar casos sospechosos para investigación posterior. En este sentido, es mejor ofrecer más pruebas a un paciente sano que no ofrecerle pruebas adicionales (y, potencialmente no identificar la enfermedad) a un paciente hepático. Así, es un punto a tener en cuenta en la evaluación de los modelos futuros.

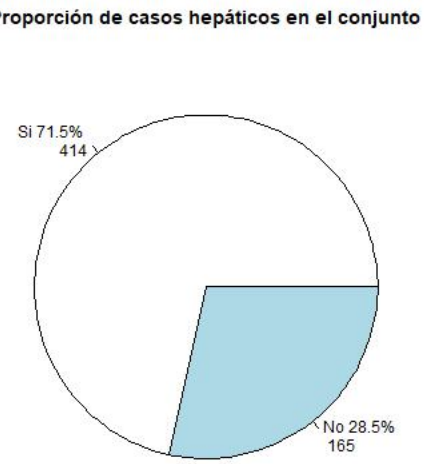

Figura 1: Composición de la variable de clasificación

2.2. Variables factor

La única variable explicativa factor en el conjunto es el género de los pacientes. Una representación gráfica de la distribución de pacientes femeninas y masculinos en el conjunto según su clase se ofrece a continuación. Hay un desequilibro notable entre la cantidad de hombres y mujeres en la muestra, una división 75-25. Sin embargo, la distribución de casos positivos y negativos según género es casi igual en ambos grupos. Un test de tipo chi-cuadrado sobre la tabla de proporciones no rechaza la hipótesis de que la clase sea independiente del género (p-valor = 0.064). Así, no se espera que la variable tenga una influencia importante en las predicciones de enfermedad.

Figura 2: Composición de la variable *gender*

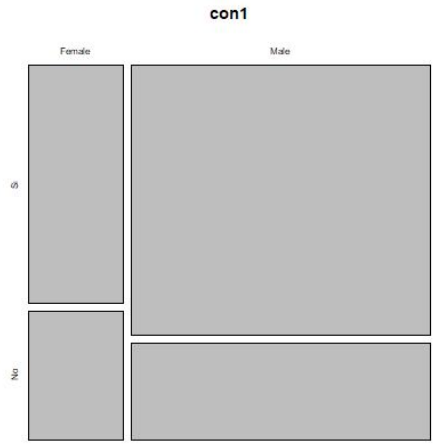

Figura 3: Composición de la variable *gender* por clases

Varios modelos y algoritmos que se usarán en el presente trabajo necesitarán que todas las variables explicativas sean de tipo numérico. Para abordar el problema se realiza recodificación de la variable *gender* con el método de one-hot-encoding para evitar posibles sesgos.

2.3. Variables Continuas

Las variables *age*, *Alkphos*, *Sgpt* y *Sgot* son de tipo entero dentro del conjunto, mientras que las variables *TB*, *DB*, *TP*, *ALB* y *AG* son reales con la precisión hasta una cifra decimal.

Se destacan muchos valores anómalos en forma de colas en varias variables. A continuación el estudio gráfico de las variables con *density plot*.

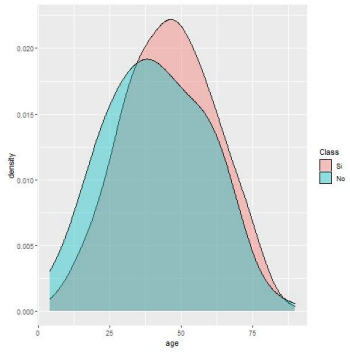

Figura 4: *age*

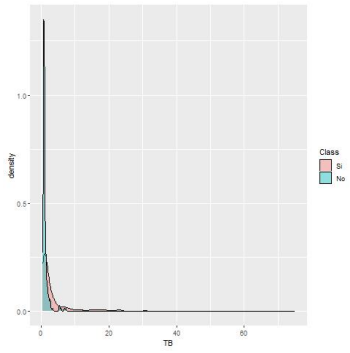

Figura 5: *TB*

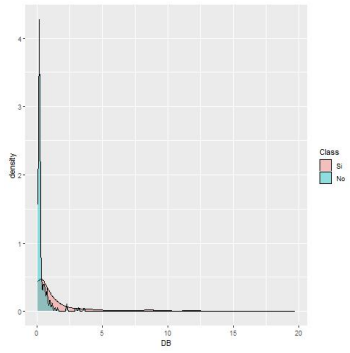

Figura 6: *DB*

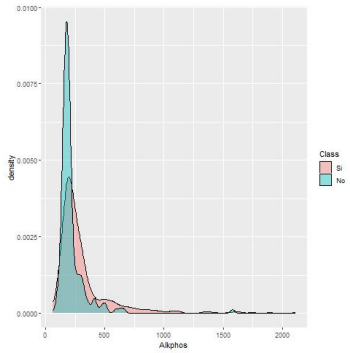

Figura 7: *Alkphos*

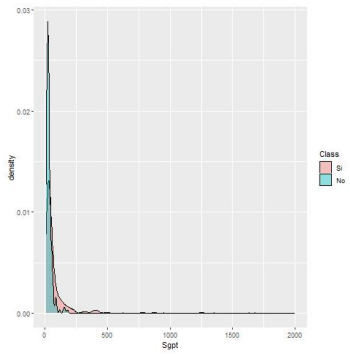

Figura 8: *Sgpt*

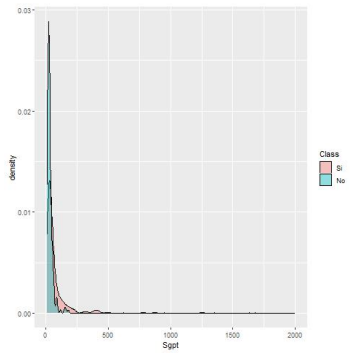

Figura 9: *Sgot*

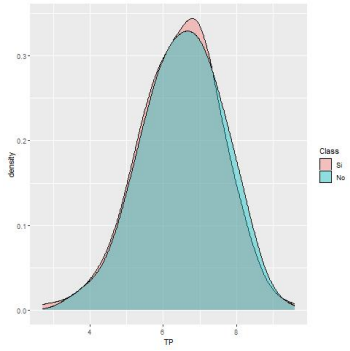

Figura 10: *TP*

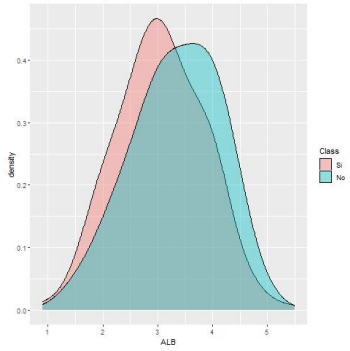

Figura 11: *ALB*

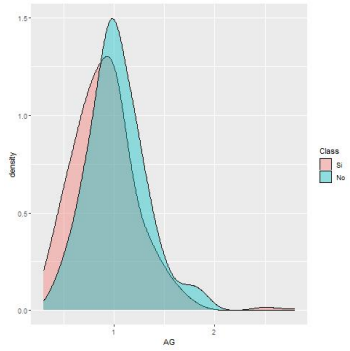

Figura 12: *AG*

En la siguiente tabla se resumen los p-valores obtenidos en un test de Wilcoxon donde se contrasta la igualdad de medianas por clases en las variables. Todas las variables salvo *TP* tienen diferencias significativas a un nivel de 0.05.

Variable	p-valor
age	0.002732
TB	2.748e-13
DB	6.45e-13
Alkphos	9.937e-11
Sgpt	3.703e-12
Sgot	1.311e-13
TP	0.4462
ALB	5.931e-05
AG	5.812e-06

Tabla 1: Contraste de Wilcoxon por clases

De una simple exploración visual de variables por separado (sin contar posibles correlaciones) se han identificado varios puntos importantes.

Los resultados identifican dos problemas con los datos:

1. Existen en el conjunto las variables (*gender* y *TP*) cuyas distribuciones son idénticas para los pacientes de diferentes clases. Eso es, estas variables por si solas no pueden usarse para predicciones de enfermedad en pacientes, pero no se descarta que su interacción con el resto de las variables las haga útiles para los algoritmos de aprendizaje automático.

2. Surge un gran problema de valores aberrantes en los conjuntos: las variables *TB*, *DB*, *Alkphos*, *Sgpt* y *Sgot* tienen colas muy grandes con valores ordenes de magnitud diferentes a las medias y cuartiles. En total unos 285 registros se encuentran fuera de los "bigotes" de diagramas de cajas de al menos una variable (eso es, fuera de 3Q+1.5*rango intercuartílico). Al tratarse de un número tan elevado, no pueden modificarse los valores ya que constiuyen una parte esencial del conjunto de datos.

Correlaciones

La correlación entre las variables se estudia gráficamente a continuación, se añaden además los valores del coeficiente de correlación entre cada par de variables posible. Notar correlaciones extremadamente altas entre varios pares de variables: *TB* y *DB* (ambas bilirrubina) a $r = 0.87$, *Sgpt* y *Sgot* (aminotransferasas) a $r = 0.79$, *TP* y *ALB* (bilirrubina y albúmina) a $r = 0.78$ y *ALB* Y *AG* (albúmina y ratio albúmina/globulina) a $r = 0.69$. Sin embargo, no se encuentran en el conjunto variables que sean combinaciones lineales de las otras.

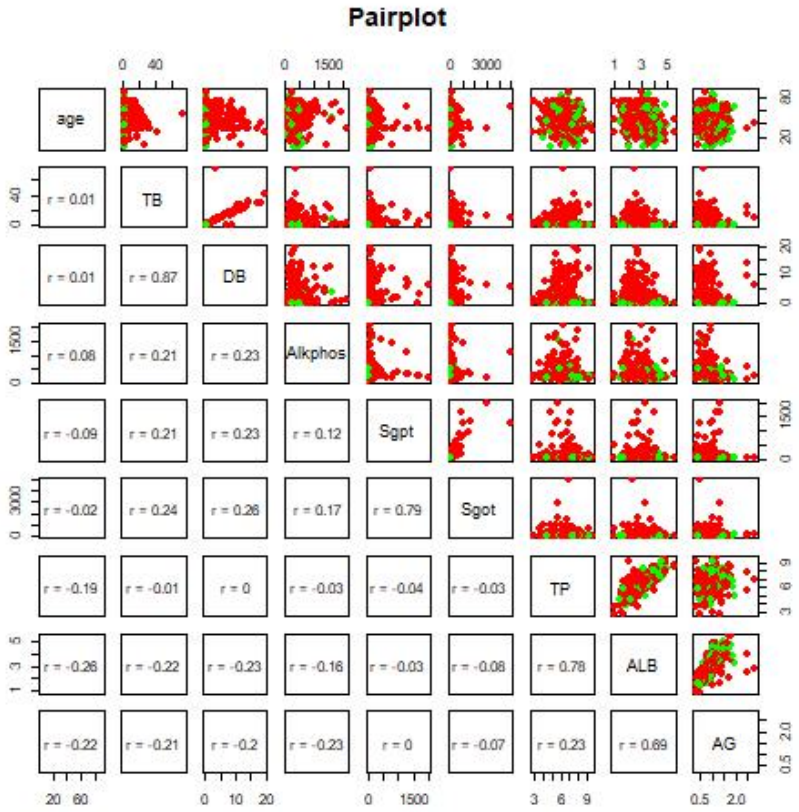

Figura 13: Correlaciones entre variables continuas

En vista de los resultados, no se elimina ninguna variable del conjunto: el número de variables explicativas actual (10) es bastante reducido y manejable para los métodos de aprendizaje automático convencionales y para el conjunto de datos utilizado puede ser más perjudicial perder la información contenida en estas variables.

Se opta además por no hacer ningún tratamiento específico a los datos para eliminar las colas de las variables que las presentan. Como ya se ha comentado, casi la mitad de registros tienen valores anómalos en una o varias variables. Esto resulta problemático: sub-representación de valores grandes en la muestra hace que su clasificación sea más difícil y la generalización de modelos a casos extra-muestrales puede fallar. Son las cuestiones a tener en cuenta en la construcción e interpretación de los resultados de modelos.

PCA

Una vez estudiadas las variables por grupos y sus correlaciones, se analizan los patrones internos y agrupaciones potenciales dentro del conjunto. Una de las técnicas usuales para ello es el análisis de las componentes principales PCA.

El cálculo indica que las dos primeras componentes principales solo explican un 44 % de variabilidad en la muestra, un resultado muy bajo. La representación gráfica 2D no

revela clustering natural de los datos.

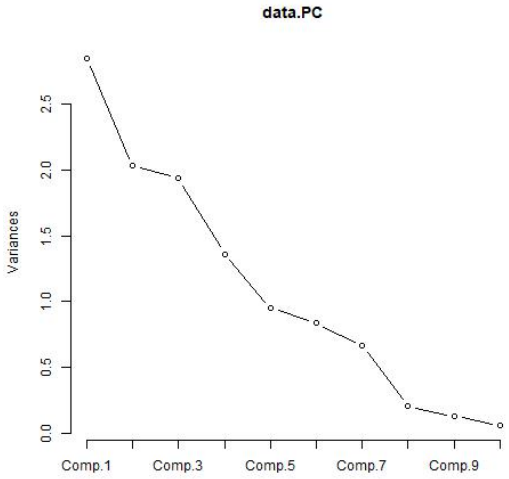

Figura 14: Varianza de componentes principales

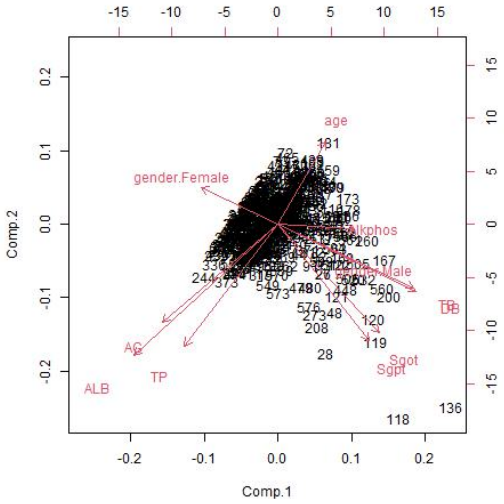

Figura 15: *BiPlot* PCA

Una técnica tan potente como es el PCA en 3 componentes principales (casi un 62 % de variabilidad de la muestra) ofrece una separación de datos en dos clusters (correspondientes a los pacientes masculinos y femeninos) que tienen aproximadamente la misma forma triangular. Los casos no hepáticos están concentrados en las bases de los triángulos muy ¡mezclados con los pacientes no hepáticos. De ahí surgen dudas de si puede existir un clasificador potente sobre el conjunto de datos: los casos de ambas clases se parecen mucho entre sí.

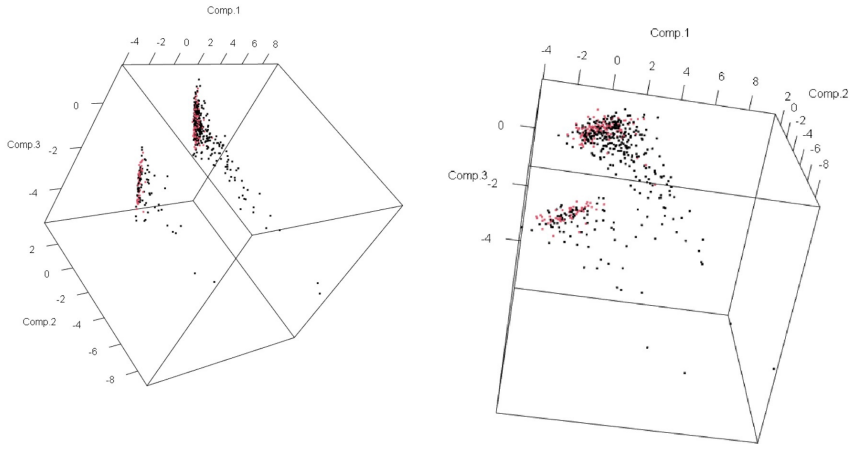

Figura 16: PCA 3D

Clustering

En linea con lo comentado sobre la técnica de PCA analizamos también posible clustering de los datos con métodos tipo *k-means*.

Un estudio de distancias entre los casos individuales no encuentra un clustering fuerte en un número reducido de grupos.

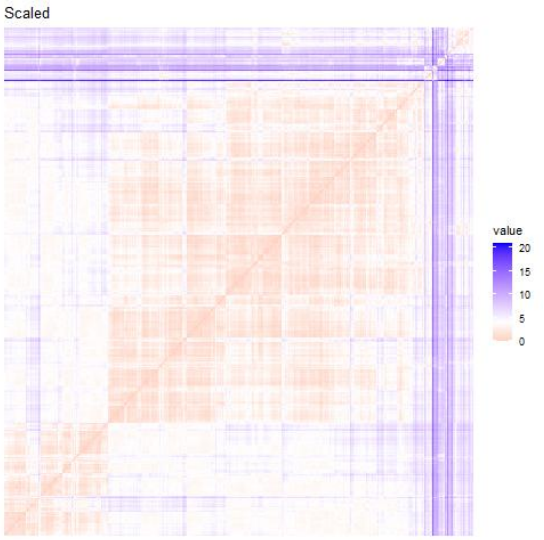

Figura 17: Distancias entre casos

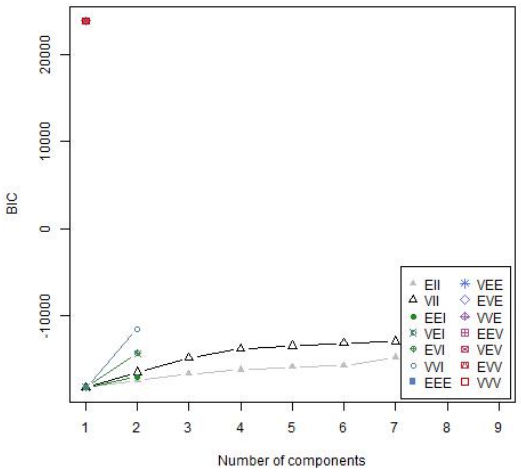

Figura 18

La selección del número de clusters mediante el valor de BIC con la función *MClust* devuelve el único valor de 1, y no se consigue un clustering que esté cercano a la clasificación deseada (binaria). Notar también que se selecciona la opción de un solo cluster para todos los datos.

2.4. Conslusiones

Se identifican varios puntos importantes a partir del análisis preliminar de los datos:

- Las variables *gender* y *TP* potencialmente no tienen relación con la clasificación de los pacientes en sanos y enfermos.
- La mitad de variables explicativas presenta colas muy fuertes desequilibradas en sus distribuciones. Casi la mitad de registros está afectada por estas colas en una o varias variables. Surgen preguntas de si los modelos de clasificación podrán generalizar bien los casos con *outliers*.
- Un estudio PCA no encuentra agrupaciones relevantes en los datos y no consigue separar casos positivos de negativos en hasta 3 dimensiones. No está claro si se podrá obtener una clasificación buena de los pacientes si los grupos están tan parecidos entre sí.
- Herramientas de clustering no encuentran agrupaciones en los datos en hasta 9 componentes individuales. Otra vez más, parece que todos los elementos del conjunto son muy parecidos entre sí.

3. Modelización

3.1. Train-test split

Para evitar contaminar la variable primero separar el conjunto de datos original no procesado en un conjunto de entrenamiento y otro de test guardando la proporción de casos positivos y negativos en cada muestra en acuerdo con su proporción en el conjunto total. El conjunto de entrenamiento es luego normalizado, los mismos valores de media y desviación se aplican también para escalar el conjunto test y no introducir sesgos innecesarios.

Es muy probable que, debido a la composición particular del conjunto de datos la partición *train-test* no sea la mejor opción de evaluación de modelos. Notar que la presencia de muchos valores anómalos en variables y también el disbalance de clases pueden introducir un *sample bias* en el entrenamiento (no hay valores "raros" suficientes y el modelo no generaliza) y en la evaluación del rendimiento final del modelo (hay muchos valores "raros" en conjunto test que no se han visto en el entrenamiento).

Sin embargo, en un principio, se arrancan los modelos con una partición de datos *train-test* clásica. La selección ofrece posibilidades de ver fácilmente las matrices de confusión de modelos y todas las métricas deseadas. Al final del trabajo, una vez vistos los algoritmos o tipos de modelos buenos para el tratamiento de datos disponibles, se puede estimar el rendimiento de los modelos con los métodos más apropiados.

A lo largo del trabajo se ajustan 3 tipos de modelos diferentes: un árbol, un *random forest* y un perceptrón múltiple. La selección de los hiperparámetros correspondientes a los modelos se realiza en el proceso de validación cruzada con repeticiones 5-5 (5 hojas y 5 repeticiones). En todos los casos la métrica con la que se selecciona el mejor modelo es el valor de ROC.

Tener en cuenta que el disbalance de clases en el conjunto de datos original (traducido también al conjunto de entrenamiento) obliga al uso de técnicas de re-muestreo. Se emplean, dentro de las posibilidades del paquete CARET las técnicas de *up-* y *down-sampling*, algoritmos SMOTE y ROSE.

- *Down-sampling*: dejar como está la clase minoritaria y muestrear la clase mayoritaria para que tenga la misma incidencia en el conjunto. El problema clave de este método es que limita severamente el conjunto de datos para el entrenamiento.
- *Up-sampling*: dejar la clase mayoritaria como está y hacer un muestreo con reemplazamiento de la clase minoritaria para igualar la incidencia. En este caso, obviamente pueden repetirse los registros no representativos de la clase negativa e introducirse un sesgo.
- SMOTE y ROSE: tienen como objetivo sintetizar nuevos datos de la clase minoritaria mientras que hacen un down-sampling de la clase mayoritaria.

Se desconoce de antemano si alguna de las técnicas será mejor que la otra (o si la mejor opción para estos datos en concreto es no hacer ningún muestreo). Se ajustan los 5 modelos diferentes (4 técnicas de *sampling* y el conjunto de entrenamiento sin alteraciones) para cada uno de los tipos de algoritmos. Los resultados se comparan a través de los valores de ROC, sensibilidad y especificidad en el conjunto de entrenamiento en validación (gráficamente con intervalos de confianza a 95 %) y los valores particulares de algunos estadísticos sobre el conjunto test.

3.2. Árbol de clasificación

Se utiliza un árbol tipo *rpart* con particionamiento recursivo. El único hiperparámetro es la complejidad.

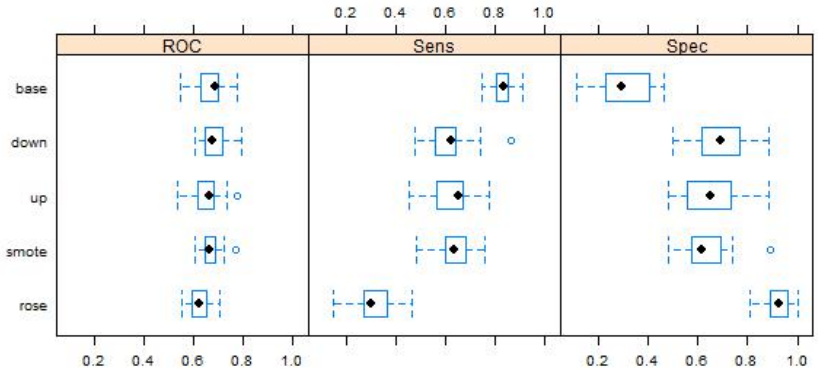

Figura 19: Modelos tipo árbol con diferentes tipos de muestreo

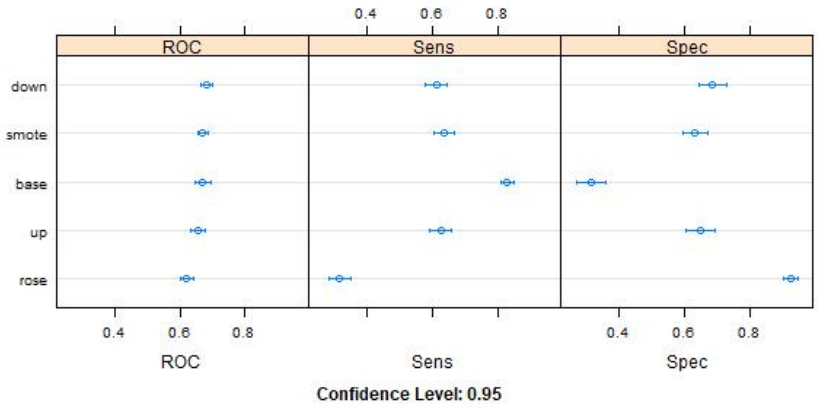

Figura 20: Modelos tipo árbol con diferentes tipos de muestreo

Métrica	árbol				
	base	**down**	**up**	**rose**	**smote**
Accuracy	0.7043	0.6522	0.6783	0.5739	0.6261
p-valor (ACC >NIR)	0.6264	0.9368256	0.823783	0.9995	0.9829937
Kappa	0.236	0.2912	0.3187	0.2602	0.2438
Mcnemar's Test P-Value	0.3912	0.0008989	0.008529	3.263e-10	0.0007937
Sensitivity	0.8293	0.6220	0.6707	0.4268	0.5976
Specificity	0.3939	0.7273	0.6970	0.9394	0.6970

Tabla 2: Medidas de rendimiento de modelos sobre el conjunto test

Los valores de ROC, sensitividad y especificidad son muy bajos en el *resampling* y el en conjunto de test en todos los modelos entrenados. Además, ningún modelo de tipo árbol de partición ofrece un balance bueno entre especificidad y sensitividad. El modelo con *sampling* ROSE muestra resultados muy altos de especificidad y bajos en sensitividad: indicativo de que casi todos los casos se clasifican como negativos. Un resultado justo al revés se obtiene con el modelo base (sin *sampling* especial): casi todas las instancias se clasifican como positivas.

Notar que el valor de accuracy ROC no es muy representativo del poder de clasificación de los modelos en el caso de conjuntos disbalanceados: se obtienen valores muy altos sin muchos aciertos en la clase minoritaria.

Los valores del índice kappa sobre el conjunto de test indican una concordancia muy leve entre los valores reales y predichos. Un contraste con NIR no rechaza la hipótesis de que una exactitud similar podría obtenerse al asignar todos los valores del conjunto a la clase mayoritaria. El test de Mcnemar no rechaza la hipótesis de que la proporción entre las clases en las predicciones es la misma que en las clases observadas para el modelo base.

Es difícil determinar cual es el mejor modelo entre los 5 entrenados para el árbol de clasificación: los resultados son insatisfactorios en el conjunto de test y en validación cruzada sobre el conjunto de prueba. Puede proponerse un cambio del límite de probabilidad para la asignación de clases: clasificación como "Si" si la probabilidad de su asignación es mayor de 0.33 (alternativamente se prueban valores de 0.66); sin embargo los resultados no mejoran. Notar también, que el uso de representación de datos en componentes principales no lleva a mejora de resultados, por lo que de aquí en adelante no se considera esta opción.

A modo de ejemplo, se muestran a continuación los resultados particulares del ajuste del árbol de partición con la técnica de *up-sampling*: selección del valor de hiperparámetro de complejidad, representación gráfica del árbol final, gráfica de la importancia de variables individuales y las curvas ROC y *lift*.

Entre los puntos relevantes destacar que el valor de ROC que corresponde a todos los valores del hiperparámetro estudiados están en la vecindad del valor óptimo (menos de un 2 %) por lo que la diferencia entre escoger un valor del parámetro u otro es

casi inexistente. La variable género no ha sido implicada en la clasificación en ningún momento, y la variable *TP* tiene una importancia mínima: resultados esperados.

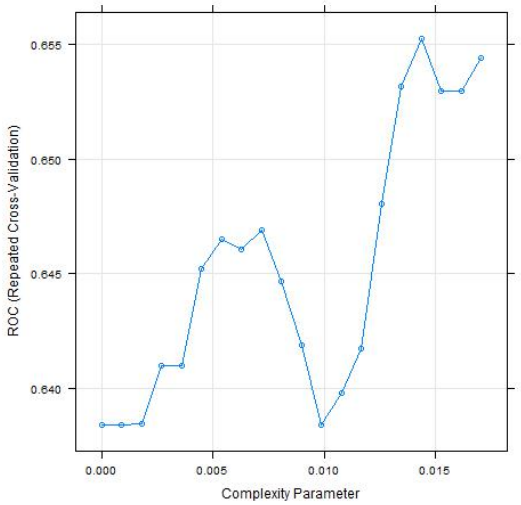

Figura 21: Selección del parámetro de complejidad

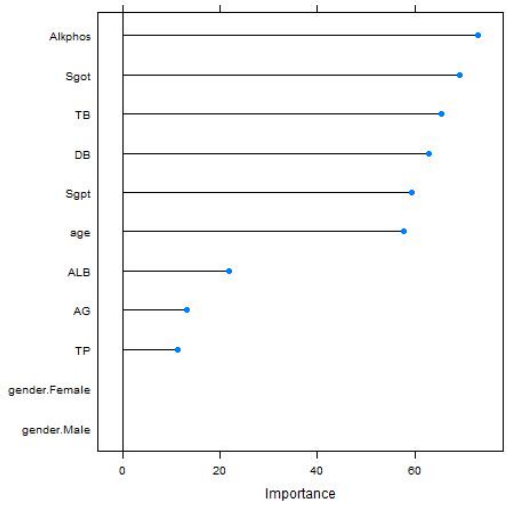

Figura 22: Importancia de variables en el modelo final

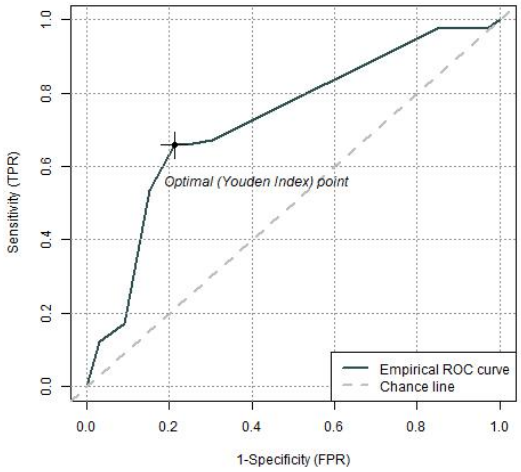

Figura 23: Curva ROC

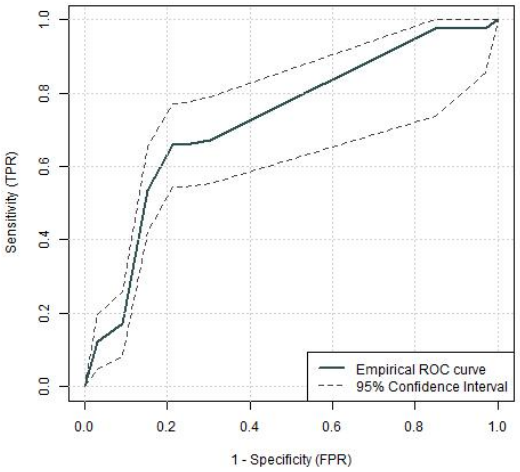

Figura 24: Curva ROC con IC a 95 %

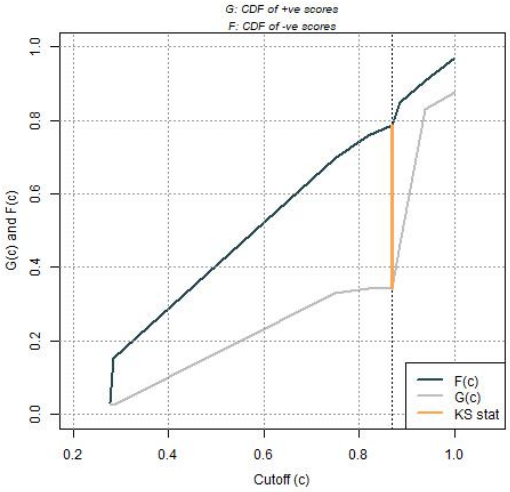

Figura 25: Estadístico KS

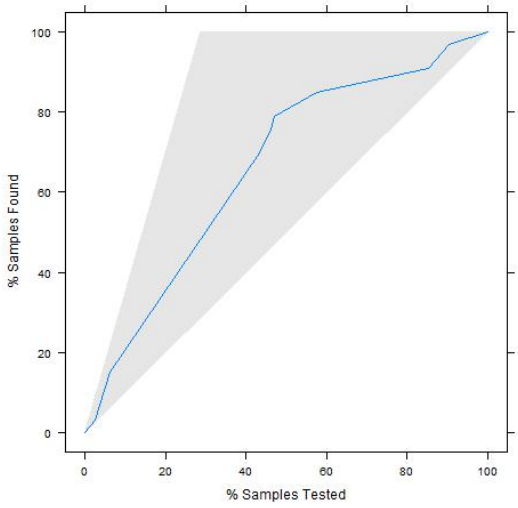

Figura 26: Curva lift

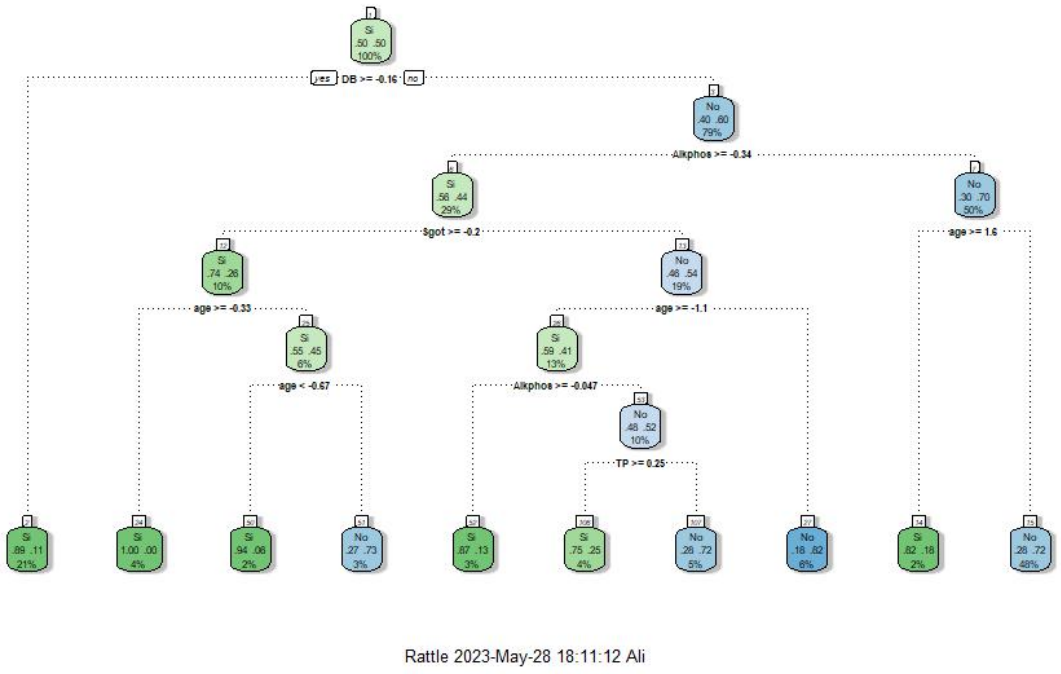

Figura 27: El árbol final

3.3. Random Forest

Se selecciona el algoritmo RF por su facilidad de interpretación y rapidez en ajuste. Su idea principal es construir muchos árboles diferentes durante el entrenamiento y considerar la clasificación final por el voto mayoritario de los clasificadores individuales.

El único hiperparámetro, el número de variables seleccionadas al azar, se ajusta en el proceso de validación cruzada.

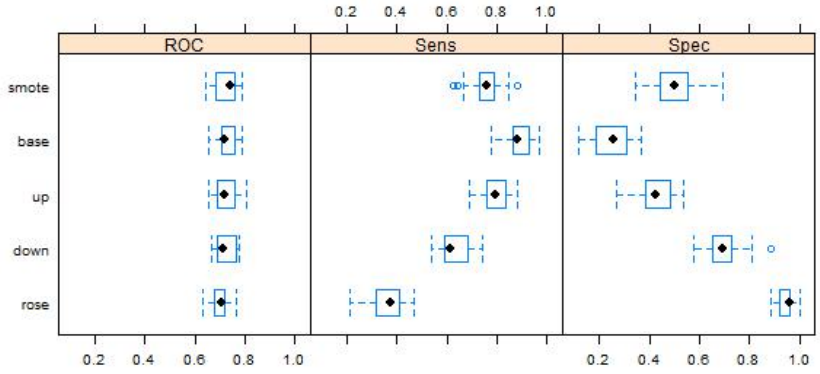

Figura 28: Modelos tipo RF con diferentes tipos de muestreo

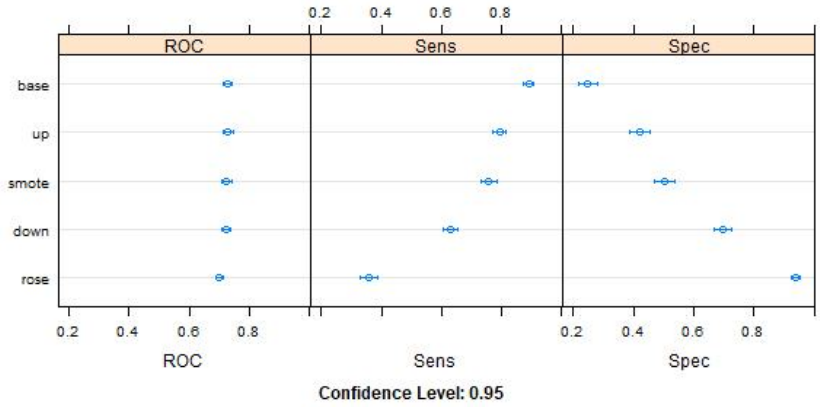

Figura 29: Modelos tipo RF con diferentes tipos de muestreo

Métrica	**random forest**				
	base	**down**	**up**	**rose**	**smote**
Accuracy	0.7391	0.6696	0.7304	0.5739	0.713
p-valor (ACC >NIR)	0.3071	0.8708431	0.3837	0.9995	0.5468
Kappa	0.3259	0.3267	0.3698	0.2696	0.3516
Mcnemar's Test P-Value	0.3613	0.0006577	0.4725	4.983e-11	0.1637
Sensitivity	0.8537	0.6341	0.7805	0.4146	0.7439
Specificity	0.4545	0.7576	0.6061	0.9697	0.6364

Tabla 3: Medidas de rendimiento de modelos sobre el conjunto test

Las mejoras obtenidas con el random forest (un algoritmo mucho más potente que un

árbol unitario y que tiene bastante flexibilidad) han sido escasas. El valor del índice kappa sobre el conjunto test subió, pero los modelos siguen teniendo problemas en clasificar bien los casos positivos y negativos a la vez.

Cabe señalar que los muestreos tipo base y rose otra ves más dan los peores resultados (y, quizá, no son adecuados para el problema). El modelo que ofrece los mejores resultados en términos del balance de ROC, sensibilidad y sensitividad es el modelo con muestreo *up* (el muestreo SMOTE mostrando resultados muy similares).

A continuación se presentan el proceso de selección de los hiperparámetros del modelo, importancia de las variables, curvas roc y lyft en el caso del muestreo *up*.

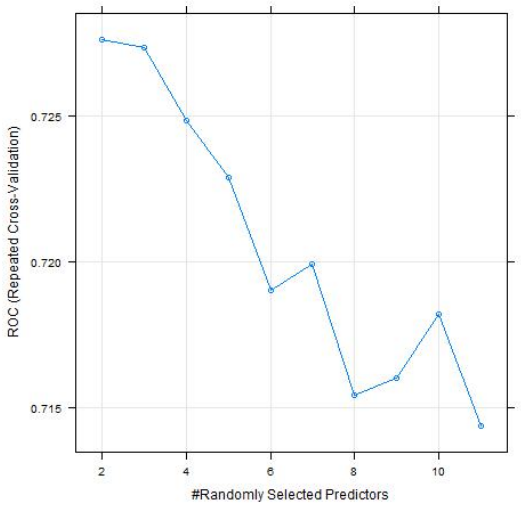

Figura 30: Selección del número de predictores

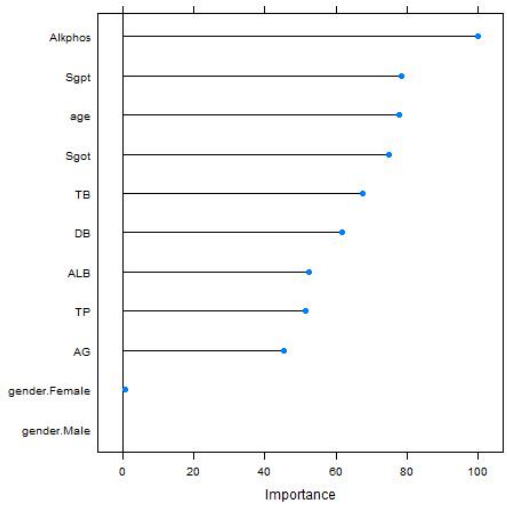

Figura 31: Importancia de variables en el modelo final

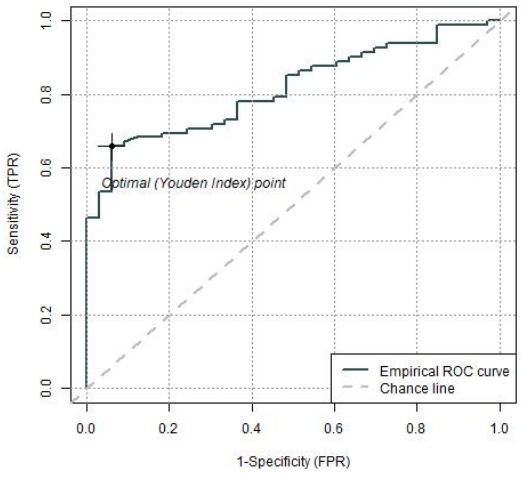

Figura 32: Curva ROC

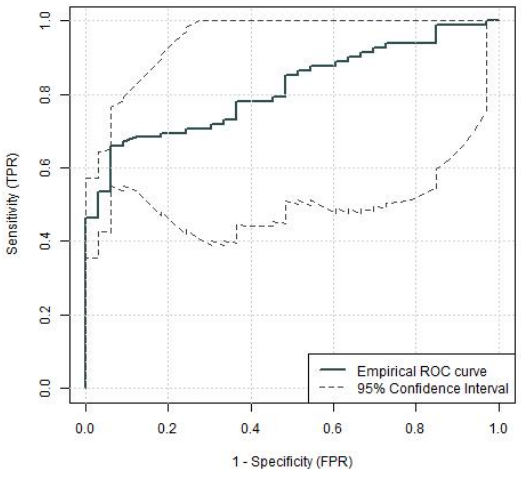

Figura 33: Curva ROC con IC a 95 %

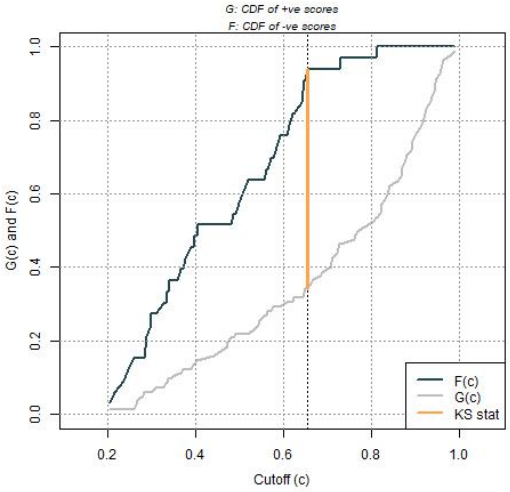

Figura 34: Estadístico KS

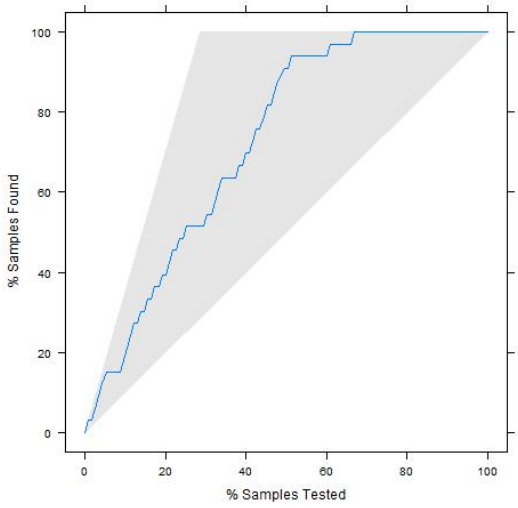

Figura 35: Curva lift

3.4. Perceptrón múltiple

Por último, se opta por ajustar un modelo de tipo perceptrón multicapa para los datos. Constituye parte de métodos conocidos como redes neuronales artificiales y el único hiperparámetro a ajustar en esta implementación particular del algoritmo es el número de capas ocultas.

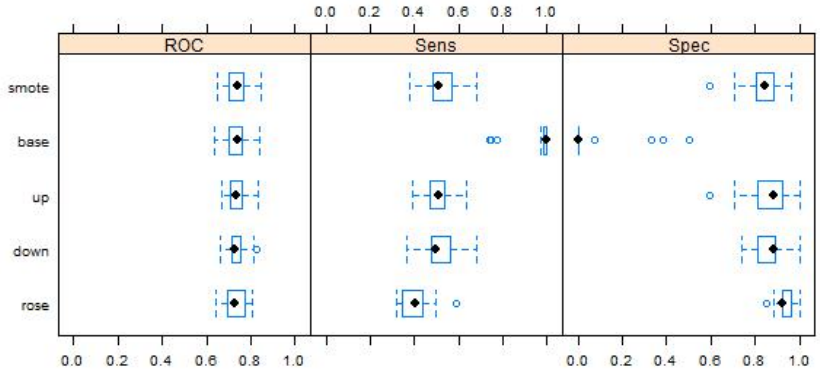

Figura 36: Modelos tipo perceptrón con diferentes tipos de muestreo

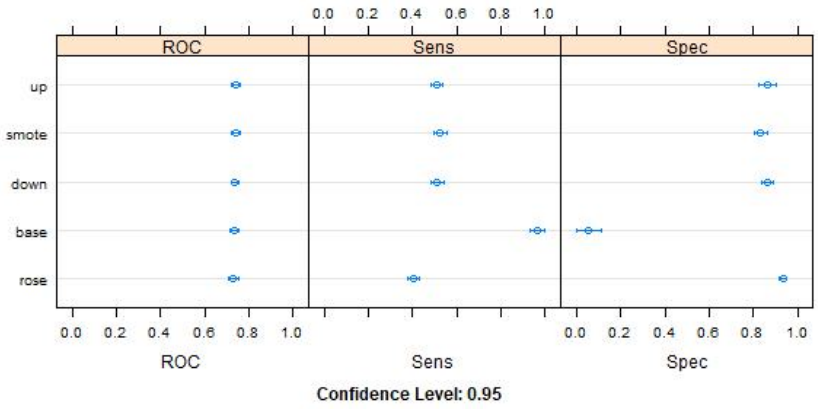

Figura 37: Modelos tipo perceptrón con diferentes tipos de muestreo

Métrica	**perceptrón multicapa**				
	base	**down**	**up**	**rose**	**smote**
Accuracy	0.713	0.6957	0.6783	0.6348	0.6696
p-valor (ACC >NIR)	0.5468	0.7005	0.8238	0.9728	0.8708
Kappa	0	0.4109	0.3773	0.3176	0.365
Mcnemar's Test P-Value	2.54e-08	2.214e-06	4.161e-06	3.543e-07	2.546e-06
Sensitivity	1.000	0.6098	0.5976	0.5366	0.5854
Specificity	0.000	0.9091	0.8788	0.8788	0.8788

Tabla 4: Medidas de rendimiento de modelos sobre el conjunto test

Los resultados del perceptrón multicapa no mejoran mucho los obtenidos con RF en términos de valores de exactitud, sensitividad o especificidad particulares. Mejora el índice kappa, eso es, las predicciones concuerdan más con los valores reales en el conjunto test. Además, las técnicas de muestreo tipo *up-sampling* y SMOTE siguen ofreciendo los mejores resultados para las tres métricas.

A continuación un ejemplo de los resultados de ajuste del modelo de *up-sampling*: selección del valor de hiperparámetro, gráfica de la importancia de variables individuales y las curvas ROC y Lyft.

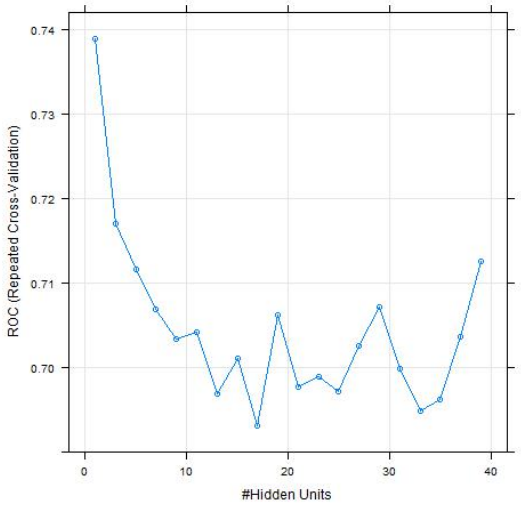

Figura 38: Selección del número de capas ocultas

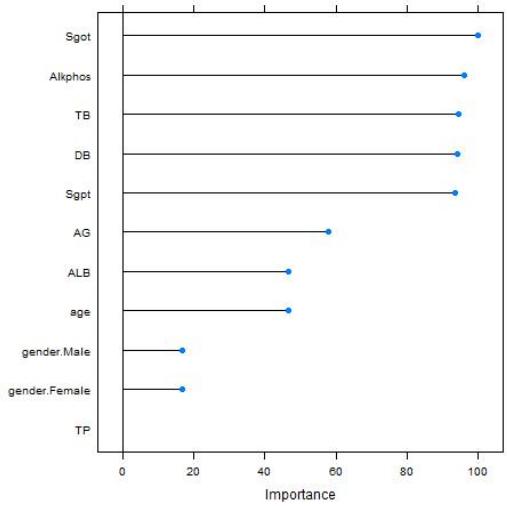

Figura 39: Importancia de variables en el modelo fina

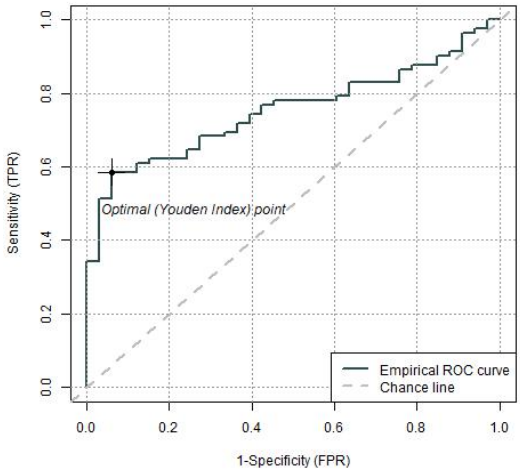

Figura 40: Curva ROC

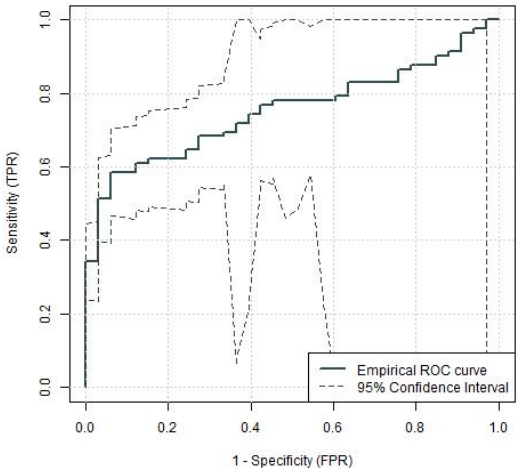

Figura 41: Curva ROC con IC a 95 %

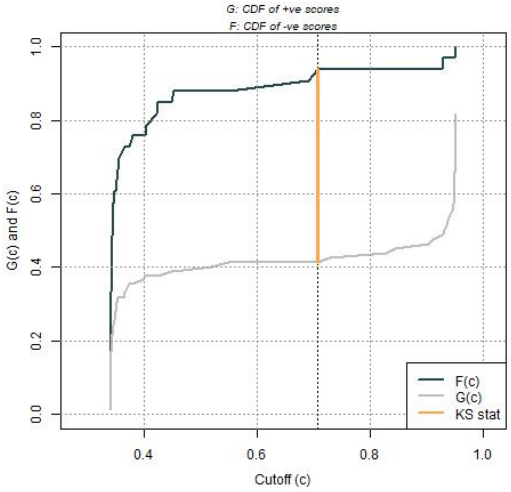

Figura 42: Estadístico KS

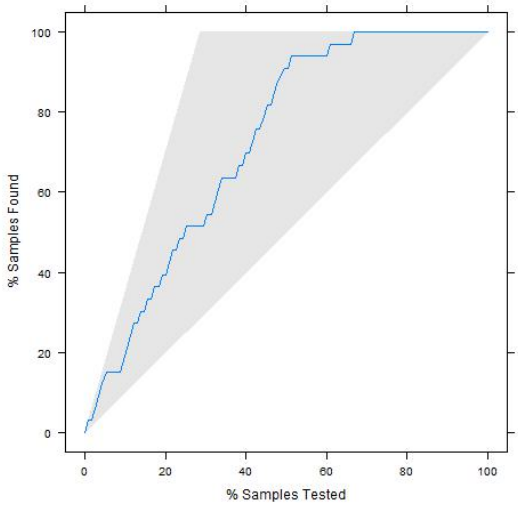

Figura 43: Curva lift

3.5. Conclusiones

- Los resultados de ajustes de 3 tipos de modelos diferentes no han sido satisfactorios: no se consigue un grado de sensitividad y especificidad altos en ninguno de los modelos ni en validación cruzada ni en el conjunto test. El valor del índice de concordancia kappa es muy bajo y la exactitud de los modelos no se diferencia significativamente de la que se podría obtener en el caso de NoInformation.
- Las técnicas de muestreo *up-sampling* y SMOTE han mostrado resultados mejores que otros métodos y se propone a continuar utilizarlas en los estudios posteriores.

Una ausencia de resultados buenos puede ser debida a un *sample bias*. Teniendo en cuenta escasez de datos en el grupo de pacientes sanos, así como gran variación en los valores de variables (colas de distribución prolongadas) quizá no son suficientes los puntos del conjunto de entrenamiento para hacer generalizaciones. Con ello, el siguiente paso de trabajo será el ajuste de los 3 tipos de modelos sobre todo el conjunto de datos. El rendimiento de los modelos se medirá en una validación cruzada 5-5 (*folds-iterations*).

4. Entrenamientos en conjunto completo

Métrica	árbol partición		random forest		perceptrón	
	up	smote	up	smote	up	smote
Accuracy	0.6418	0.6297	0.7029	0.7081	0.5475	0.485
p-valor (ACC >NIR)	1	1	0.9276	0.80091	1	1
Kappa	0.2738	0.258	0.2641	0.2984	0.0056	0.0118
Mcnemar's Test P-Value	<2e-16	<2e-16	0.3939	0.05405	<2e-16	<2e-16
Sensitivity	0.6101	0.5923	0.7986	0.7821	0.6063	0.4551
Specificity	0.7212	0.7236	0.4630	0.5224	0.4000	0.5600

Tabla 5: Medidas de rendimiento de modelos en validación cruzada

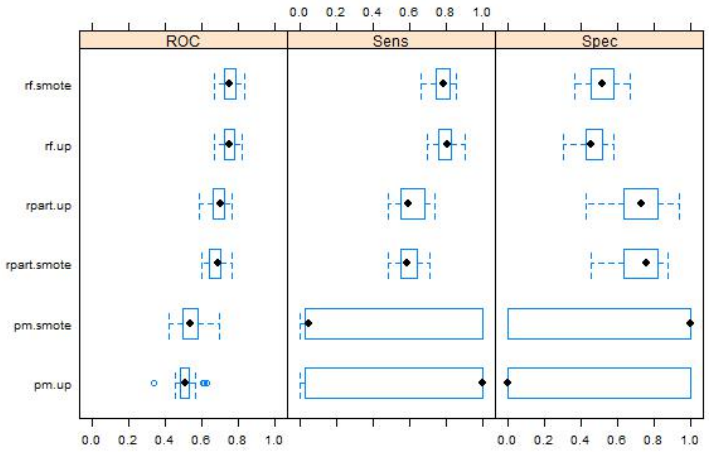

Figura 44: Modelos finales con diferentes tipos de muestreo

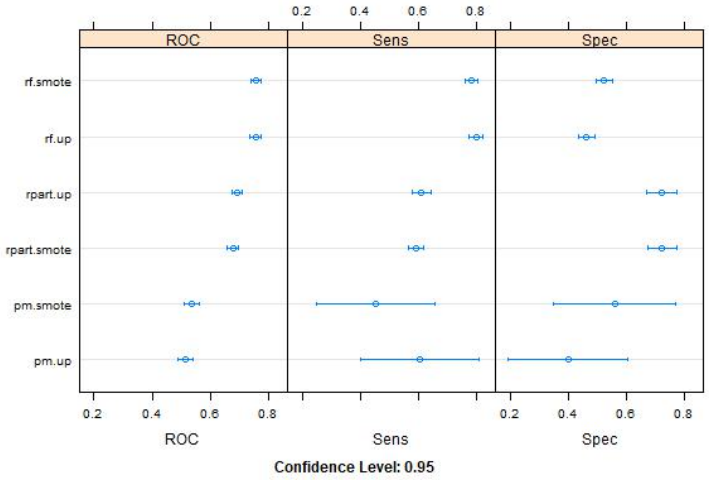

Figura 45: Modelos finales con diferentes tipos de muestreo

En los modelos tipo perceptrón multicapa varia mucho en sensitividad y especificidad, hasta el punto cuando su valor de kappa total en la validación cruzada es de 0.01 aproximadamente (un valor que indica casi ausencia de concordancia entre valores predichos y observados). A pesar de que los modelos tipo perceptrón devolvían resultados relativamente buenos en el conjunto de test, concluir que se trataba de un posible *sample bias*.

Los modelos tipo RF tienen muy buena sensitividad (clasifican muchísimos casos positivos bien) y una especificidad de alrededor de 50 %. Eso es, en aproximadamente mitad de los casos, un paciente clasificado como sano es en realidad enfermo. Sin embargo, esos son los modelos de mejor valor de ROC.

Los modelos tipo árbol ofrecen una sensitividad y especificidad moderada por encima de 50 % ambas, pero no demasiado altos.

En general, todos los modelos tienen el índice de concordancia kappa muy bajos, aunque una comparación directa entre los valores de kappa en validación cruzada 5-5 en todo el conjunto de datos y en una validación en el conjunto de test limitado no es de todo correcta. Además, en ningún caso se rechaza la hipótesis de que la clasificación ofrecida es diferente a una alcanzada por NIR (no information rate, asignación a todos los casos a la clase mayoritaria).

Por último, en este caso una comparación de curvas ROC no tiene sentido al ser el conjunto de entrenamiento el total. Pueden realizarse estudios de importancia de variables individuales, pero con los resultados obtenidos, no está claro si tienen mucho sentido.

5. Conclusiones

El trabajo realizado se concluye en los siguientes puntos:

- Un estudio preliminar de los datos señaló que existen ciertas relaciones de casi todas las variables explicativas con la variable de clasificación. Sin embargo, el análisis de agrupaciones internas (PCA y clustering) no ha indicado una diferencia apreciable entre los casos de diferentes clases. Los valores están extraordinariamente mezclados en el espacio de las 3 primeras componentes principales (61 % de variabilidad total de la muestra). Los resultados indican, que obtener unos clasificadores buenos sobre el conjunto es difícil.
- El conjunto de datos analizado es muy mal balanceado. Menos de un 30 % de los registros correspondían a los pacientes sanos, mientras que el resto eran enfermos. Este hecho, junto con el tamaño del conjunto total, hizo que sea muy difícil generalizar la información lo que impidió su correcta identificación.

  Para mitigar el problema, se han empleado varias técnicas de re-muestreo. Las técnicas de *up-sampling* y SMOTE han conducido a mejores resultados.
- Se han ajustado varios modelos de tipo árbol, *random forest* y perceptrón multicapa. Sus correspondientes hiperparámetros se han ajustado en una validación

cruzada repetida. En el conjunto test aleatorio (el mismo para todos los modelos) ninguno de los modelos ha conseguido buenos resultados de sensitividad y especificidad al mismo tiempo. El índice de concordancia kappa no ha superado el 0.41 indicando concordancia baja entre los valores reales y predichos.

- Se ha supuesto, que los resultados insatisfactorios se deben a la partición *train-test* del conjunto inicial de datos. Aunque la partición se ha realizado conservando la proporción de casos de cada clase, quizá la escasez de datos ha sido demasiado grande para que los modelos puedan generalizar bien fuera de los casos conocidos.

  Para solucionar el problema, se ha propuesto una evaluación del rendimiento de los modelos en validación cruzada, utilizando todos los datos disponibles. Sin embargo, los resultados en una validación cruzada 5-5 (hojas-repeticiones) han sido peores que en el caso de un conjunto test dedicado.

  En ningún caso y para ningún modelo se ha rechazado la hipótesis de que la exactitud de la predicción ofrecida es significativamente mejor que la que podría obtenerse en un caso de información nula (NoInformationRate, asignación de todos los casos a la clase mayoritaria).