

# Predicción de la supervivencia de pacientes traumatizados graves utilizando EUSBoost

José Sanz<sup>1</sup>, Mikel Galar<sup>1</sup>, Humberto Bustince<sup>1</sup>, Cedric Marco-Detchart<sup>1</sup>,  
Carlos Gradin<sup>2</sup>, and Tomás Belzunegui<sup>2,3</sup>

<sup>1</sup> Dpto. de Automática y Computación, Universidad Publica de Navarra, Campus  
Arrosadia s/n, 31006 Pamplona, Spain,

{joseantonio.sanz,mikel.galar,bustince}@unavarra.es

<sup>2</sup> Department of Health, Universidad Publica de Navarra, Barañáin Avenue s/n,  
Pamplona, P.O. Box 31008, Spain,

cgradin@larraonaclaret.org

<sup>3</sup> Accident and Emergency Department, Hospital of Navarre,Address,Pamplona,  
Spain,

tomasbelzunegui@gmail.com

**Resumen** Los pacientes poli-traumatizados son aquellos que sufren numerosas heridas cuya gravedad implica la intervención de los servicios de urgencias para tratar de salvar sus vidas. Los sistemas de predicción de la supervivencia de dichos pacientes se utilizan para medir la calidad de los servicios de urgencias comparando las predicciones con el estado final de los pacientes. Afortunadamente, el número de pacientes que sobreviven a sus heridas es mucho mayor que el de pacientes que fallecen. Sin embargo, para los sistemas de predicción este factor implica una mayor dificultad a la hora de aprender los modelos utilizados para realizar las predicciones. El objetivo de este trabajo es aplicar EUSBoost para abordar este problema ya que ha demostrado ser una buena herramienta para abordar problemas de clasificación no balanceados como éste. El estudio experimental se lleva a cabo utilizando 462 pacientes tratados en el Hospital de Navarra.

**Keywords:** Pacientes traumatizados graves, Predicción de supervivencia, Problemas de clasificación no balanceados, Ensembles

## 1. Introducción

Los pacientes poli-traumatizados son aquellos que sufren numerosas heridas causadas por intercambios de energía [1], como accidentes o caídas. La predicción de la supervivencia de éstos pacientes es un buen indicador de la calidad de los servicios de urgencias, puesto que un buen indicador será que el número de pacientes salvados sea mayor que el de pacientes que han sido predichos como supervivientes. El objetivo de los servicios de urgencias es salvar tantas vidas como sea posible así como tratar de que los supervivientes tengan la mejor calidad de vida tras su recuperación. Este último hecho implica una reducción de los gastos derivados de los tratamientos posteriores asignados a éstos pacientes.

Por tanto, es interesante desarrollar modelos para predecir la supervivencia de los pacientes que llegan a los servicios de urgencias. Para ello, los médicos aplican habitualmente técnicas que convierten la gravedad de las heridas en un número que representa la probabilidad de supervivencia de los pacientes. Por este motivo, este problema puede ser visto como un problema de clasificación [2] puesto que existen solamente dos valores posibles como salida del sistema: *sobrevive* y *muere*. Hoy en día el uso de técnicas de minería de datos se ha extendido en gran medida para abordar problemas de clasificación [3,4].

Afortunadamente, el número de pacientes poli-traumatizados que sobreviven a sus heridas es mucho mayor que el de pacientes que fallecen. En minería de datos, este problema es conocido como problema no balanceado [6,18], puesto que existen más ejemplos de una clase (clase mayoritaria, sobrevive) que de la otra (clase minoritaria, muere). Este tipo de problemas es un reto para las técnicas de minería de datos puesto que normalmente los clasificadores tienden a predecir la clase mayoritaria para todos los ejemplos y como consecuencia, en numerosas ocasiones fallan la predicción de los ejemplos de la clase minoritaria. La predicción de la clase muere es importante puesto que se pueden analizar las variables involucradas y, de esta forma, desarrollar protocolos para reducir el número de muertes.

El objetivo de este trabajo es aplicar el algoritmo EUSBoost [19] (*Evolutionary Under Sampling Boosting*) para abordar la predicción de la supervivencia de los pacientes poli-traumatizados. EUSBoost es un ensemble de clasificadores (árboles de decisión C4.5 [7]). Para generarlo, se aplican un proceso de bajo muestreo evolutivo [20] introducido en el bucle del algoritmo AdaBoost.M2 [21] (Boosting). Este algoritmo ofrece unos grandes resultados al afrontar problemas de clasificación no balanceados por lo que es un buen candidato para resolver con éxito este problema. Además, no es necesario aplicar técnicas de muestreo antes de aplicar el algoritmo de aprendizaje por lo que se realiza con los pacientes reales del problema.

El estudio experimental se ha llevado a cabo utilizando los pacientes almacenados en el *Major Trauma Registry of Navarre* (MTRN) [9]. En concreto, el MTRN está compuesto por 462 pacientes tratados en los servicios de urgencias del Hospital de Navarra durante 2011 y 2012. La calidad de los resultados se mide utilizando tres métricas bien conocidas como el porcentaje de ejemplos clasificados correctamente (*accuracy*), el área bajo la curva ROC (*Area Under the Curve*, AUC) y la media geométrica (GM), que cuantifica el balance entre la precisión de clasificación en ambas clases.

El resto del trabajo está organizado de la siguiente manera: en la Sección 2 se describe el problema abordado en este trabajo. En la Sección 3 se introducen los conceptos necesarios acerca de los problemas de clasificación no balanceados y a continuación se describe el algoritmo EUSBoost en la Sección 4. Los resultados obtenidos y el correspondiente análisis se muestran en la Sección 5 y finalmente, en la Sección 6 presentamos las conclusiones del trabajo.

## 2. Descripción del problema de la predicción de la supervivencia de los pacientes poli-traumatizados

Los pacientes poli-traumatizados son personas que sufren numerosas heridas que implican un riesgo de muerte para ellas. Es una de las causas de muerte más frecuentes entre la población menor de 40 años y también implica unos costes económicos para los centros de salud [10]. El ratio de supervivencia de estos pacientes es un buen indicador de la calidad de los servicios de urgencias de un centro de salud. Específicamente, existe un tratamiento médico aprobado para dichos pacientes y además existe una relación entre las medidas terapéuticas y la supervivencia de los pacientes, que solamente puede tomar dos valores *sobrevive* o *muere*.

El objetivo de cualquier sistema de control de los hospitales es realizar una mejora continua y cuantificable de los tratamientos utilizados para tratar a los pacientes. Con este objetivo, la información obtenida de todos los pacientes poli-traumatizados tratados en los centros de salud se almacenan en un *Major Trauma Registry (MTR)* [9]. Un MTR es una fuente precisa y completa de información que permite monitorizar continuamente el proceso de asistencia en las unidades de cuidados de trauma. Un MTR bien diseñado ayuda a los directores de los hospitales a analizar la información e intentar descubrir aspectos que pueden ser cambiados con los objetivos de mejorar la calidad de vida de los supervivientes y coordinar los diferentes servicios involucrados en las unidades de cuidados. Esta monitorización y control de calidad ha permitido reducir tanto el ratio de mortalidad como el de minusvalías de éstos pacientes en los países desarrollados en los últimos años [11].

El Departamento de Urgencias del Hospital de Navarra realizó un estudio que permitió desarrollar y validar el MTR de Navarra (MTRN) [9]. Este registro esta basado en el modelo Utstein [12], que establece las variables que deben ser recogidas. Algunas de ellas se obtienen fácilmente como la edad o el genero de los pacientes mientras que otras están basadas en la gravedad de las heridas como el *Injury Severity Score (ISS)* [13], el *New Injury Severity Score (NISS)* [14], el *Revised Trauma Score (RTS)* [15].

Debemos destacar que no todos los pacientes poli-traumatizados se almacenan en el MTRN. Los criterios de exclusión son los siguientes:

- 1) El valor de la variable NISS es menor que 15.
- 2) El periodo de tiempo transcurrido entre la herida y la admisión en el hospital es mayor que 24 horas.
- 3) El paciente se ha ahogado.
- 4) El paciente se ha colgado.
- 5) El paciente se ha quemado.

En concreto, el MTRN almacena datos de 462 pacientes, de los cuales 368 sobrevivieron a sus heridas mientras que 94 fallecieron. Por tanto, es un problema de clasificación no balanceado como vamos a explicar en la Sección 3.

### 3. Problemas de clasificación no balanceados

Un problema de clasificación consiste en aprender una función llamada *clasificador* que permitirá predecir la clase de nuevos ejemplos. Para ello se utiliza un conjunto de entrenamiento  $\mathcal{D}_T$  compuesto por  $P$  ejemplos etiquetados  $x_p = (x_{p1}, \dots, x_{pn}), p = \{1, \dots, P\}$ , donde  $x_{pi}$  es el valor del  $i$ -ésimo atributo ( $i = \{1, 2, \dots, n\}$ ) del  $p$ -ésimo ejemplo de entrenamiento. Cada ejemplo pertenece a una clase  $y_p \in \mathbb{C} = \{C_1, C_2, \dots, C_m\}$ , donde  $m$  es el número de clases del problema. Cuando el clasificador está compuesto por varios clasificadores se le conoce como *ensemble* de clasificadores.

Un problema de clasificación no balanceada [6,18] es aquel en el que el número de ejemplos pertenecientes a las diferentes clases es considerablemente diferente. Cuando afrontamos problemas de dos clases, la clase con el mayor número de ejemplos se conoce como la clase mayoritaria (también llamada clase negativa) y la otra clase se conoce como clase minoritaria (o positiva). El ratio de no balanceo (*Imbalanced Ratio*, IR, en inglés) se calcula dividiendo el número de ejemplos de la clase mayoritaria entre el número de ejemplos de la minoritaria. Esta medida se utiliza para ofrecer información útil acerca de este tipo de problemas. Según la descripción de nuestro problema, podemos observar que es un problema de clasificación no balanceada puesto que, afortunadamente, el número de pacientes que sobreviven es mucho mayor que el de pacientes que fallecen a sus heridas. El IR del problema de este trabajo es 3.9.

Para medir la calidad de los clasificadores existen diferentes medidas apropiadas para éste contexto. En este trabajo, además del *accuracy*, vamos a utilizar dos de ellas que están construidas a partir de la matriz de confusión (Tabla 1), que almacena el número de ejemplos clasificados correcta e incorrectamente de cada clase.

**Tabla 1.** Matriz de confusión para un problema de dos clases.

	Predicción: clase positiva	Predicción: clase negativa
Clase real: positiva	Verdadero Positivo (VP)	Falso Negativo (FN)
Clase real: negativa	Falso Positivo (FP)	Verdadero Negativo (VN)

La primera medida que hemos seleccionado es la media geométrica (GM), que tiene en cuenta la precisión obtenida en cada clase del problema como se muestra en Eq. (1).

$$GM = \sqrt{VP_{rate} * VN_{rate}}, \quad (1)$$

donde  $VP_{rate} = \frac{VP}{VP+FN}$  y  $VN_{rate} = \frac{VN}{VN+FP}$  son el porcentaje de ejemplos positivos y negativos clasificados correctamente, respectivamente.

La segunda medida es el área bajo la curva ROC (*Area Under the Curve*, AUC, en inglés). Esta medida se obtiene aplicando Eq. (2).

$$AUC = \frac{1 + VP_{rate} - FP_{rate}}{2}, \quad (2)$$

donde  $FP_{rate} = \frac{FP}{V_{N+FP}}$  es el porcentaje de ejemplos negativos fallados.

## 4. EUSBoost

En esta sección vamos a describir en primer lugar el algoritmo de Bajo Muestreo Evolutivo (BME) [20] (Sección 4.1), puesto que EUSBoost lo combina con Boosting. A continuación se describe brevemente el algoritmo EUSBoost [19] (Sección 4.2).

### 4.1. Algoritmo de bajo muestreo evolutivo

El algoritmo de bajo muestreo evolutivo [20] proviene de la aplicación de la selección de prototipos evolutiva en problemas de clasificación no balanceados puesto que algunas de sus características, como la función de fitness, son adecuadas para este entorno.

La selección de prototipos es un proceso de muestreo que, en vez de balancear la distribución de las clases, reduce el conjunto de entrenamiento para que el algoritmo del vecino más cercano (1NN) mejore su porcentaje de acierto y alivie su carga de almacenamiento. BME trata de obtener un sub-conjunto útil del conjunto de ejemplos original. Para ello, comienza eligiendo aleatoriamente varios sub-conjuntos de ejemplos que son evolucionados (utilizando el algoritmo evolutivo CHC) hasta que el mejor sub-conjunto no pueda ser mejorado en términos de la función de fitness.

En todos los algoritmos evolutivos la representación de la solución es un factor clave. En BME, cada cromosoma es un vector binario que representa la presencia o ausencia de los ejemplos en el sub-conjunto. Para reducir el espacio de búsqueda solamente se tienen en cuenta los ejemplos de la clase mayoritaria de tal forma que los de la clase minoritaria siempre son incluidos en el sub-conjunto.

Para evaluar la calidad de los cromosomas se utiliza una función de fitness que considera el balance entre los ejemplos de la clase mayoritaria y la minoritaria junto con el rendimiento esperado al utilizar el sub-conjunto. Este rendimiento se calcula utilizando la media geométrica, que permite maximizar el rendimiento en las dos clases a la vez, obtenida por el algoritmo 1NN aplicando la técnica de *hold-one-out*. La función de fitness se muestra en la Eq. (3).

$$fitness_{EUS} = \begin{cases} GM - |\frac{n^+}{N^-} \cdot P| & \text{if } N^- > 0 \\ GM - P & \text{if } N^- = 0, \end{cases} \quad (3)$$

donde  $n^+$  es el número de ejemplos de la clase minoritaria,  $N^-$  es el número de ejemplos seleccionados de la clase mayoritaria y  $P$  es un factor de ponderación de ambos criterios cuyo valor recomendado es 0.2.

## 4.2. EUSBoost

Las técnicas aleatorias son un buen método de construir ensembles puesto que ofrecen una gran diversidad que, combinada con clasificadores base precisos se traduce en ensembles de grandes rendimientos [22]. En concreto, los ensembles para problemas no balanceados ofrecen buenos resultados como se muestra en [18].

EUSBoost es uno de los ensembles más recientes y competitivo en este ámbito [19]. Los autores de EUSBoost afirman que, a pesar de los buenos resultados de las técnicas aleatorias, la aleatoriedad sin control puede ser mejorada al abordar problemas de clasificación altamente no balanceados, lo cual puede suponer una mejora en el rendimiento del ensemble. Por este motivo utilizan la metodología de RUSBoost [16] introduciendo el proceso de bajo muestreo en el bucle del algoritmo AdaBoost.M2. Sin embargo, EUSBoost utiliza el algoritmo BME en lugar del bajo muestreo aleatorio para llevar a cabo el proceso de muestreo. De esta forma, solo los pesos de los ejemplos seleccionados por BME se utilizarán en el proceso de aprendizaje. Como consecuencia los autores de EUSBoost<sup>4</sup> modifican el algoritmo AdaBoost.M2 incluyendo las siguientes dos fases justo después de haber calculado la distribución original de los pesos en AdaBoost.M2:

- Se introduce BME, que devuelve un nuevo conjunto de datos ( $S'$ ) que incluye todos los ejemplos de la clase minoritaria y los ejemplos de la clase mayoritaria seleccionados.
- Se obtiene la nueva distribución de pesos.

Además, el entrenamiento del clasificador se modifica de tal forma que se consideran todos ejemplos del conjunto original de datos pero aquellos que no hayan sido seleccionados tendrán peso 0 y por tanto, serán ignorados.

Uno de los problemas de EUSBoost es que el objetivo de obtener clasificadores base precisos implica una pérdida en la diversidad del ensemble. Además, el hecho de tener clasificadores base más precisos no implica obtener un ensemble más preciso. Para abordar estos problemas, EUSBoost utiliza la información del problema no balanceado para impulsar la diversidad de los clasificadores base. Para ello se utiliza una función de fitness que favorece los cromosomas que combinan mejor la diversidad con el rendimiento (medido con la función de fitness mostrada en la Eq. (3)). Los autores asumen que los clasificadores base aprendidos con conjuntos de ejemplos diferentes serán diversos y en consecuencia, miden la diversidad en base al cromosoma y no a las salidas de los clasificadores del ensemble. Además, para incrementar en mayor medida la diversidad se modifican los pesos de los factores en cada iteración. Como consecuencia, la función de fitness final de EUSBoost se muestra en la Eq. (4).

$$fitness_{EUSBoost} = fitness_{EUS} \cdot \frac{1,0}{\beta} \cdot \frac{10,0}{IR} \cdot \beta \quad (4)$$

<sup>4</sup> Para conocer los detalles de EUSBoost ver [19]

donde  $fitness_{EUS}$  es la función de fitness mostrada en la Eq. (3), IR es el ratio de no balanceo, Q el el Q-estadístico global <sup>5</sup> y  $\beta$  es un factor de ponderación que cambia en cada iteración:  $\beta = \frac{N-t-1}{N}$ .

## 5. Estudio experimental

En esta sección mostramos los resultados obtenidos por las propuestas consideradas en este estudio para abordar la predicción de la supervivencia de pacientes poli-traumatizados.

El entorno experimental utilizado para desarrollar los experimentos junto con los algoritmos utilizados en la comparativa se introducen en la Sección 5.1. Los resultados y su correspondiente análisis se muestran en la Sección 5.2.

### 5.1. Entorno experimental

El conjunto de datos contiene información de 462 pacientes almacenados en el MTRN que han sido tratados en el Hospital de Navarra entre el 1 de Enero de 2011 y el 31 de Diciembre de 2012. Las variables contempladas son las especificadas por el modelo Utstein [12], en total se almacenan 53 variables. 368 de los 462 pacientes sobrevivieron a sus heridas mientras que los 94 restantes fallecieron. Estos datos suponen el 79.65 % y el 20.35 % de los pacientes, respectivamente.

Para evaluar el rendimiento de los clasificadores, uno de los modelos más utilizados es el modelo de validación cruzada de  $k$  particiones (k-SCV). En nuestro caso, hemos aplicado 10-SCV ( $k = 10$ ). Es decir, hemos dividido el conjunto de datos en 10 particiones con el mismo número de pacientes manteniendo en cada una de ellas la distribución de las clases. A continuación, unimos 9 de ellas para aprender los clasificadores y utilizamos la restante para evaluar la calidad del sistema. Este proceso se repite 10 veces considerando una partición de test diferente en cada caso. Por tanto, al acabar el proceso habremos evaluado todos los pacientes como ejemplos de test. El resultado final es la media de las 10 particiones de test.

En cada partición utilizamos tres métricas ampliamente utilizadas para medir el rendimiento de los clasificadores: el *accuracy*, la media geométrica (GM) y el área bajo la curva ROC (AUC). Tal y como hemos descrito en la Sección 3, la primera de ellas es una métrica estándar en problemas de clasificación mientras que las dos restantes son métricas muy apropiadas para evaluar el rendimiento de los clasificadores en problemas de clasificación no balanceados.

Con respecto a la configuración de EUSBoost, debemos señalar que el clasificador base es el árbol de decisión C4.5 [7], utilizando 0.25 como nivel de confianza, la corrección de Laplace y 2 ejemplos por hoja como mínimo. El ensemble está formado por 10 clasificadores de acuerdo a las recomendaciones para Boosting. Finalmente, para aplicar el algoritmo BME se han utilizado poblaciones de 50 individuos, 10.000 iteraciones, la probabilidad de inclusión de HUX

<sup>5</sup> Ver [19] para detalles sobre el cálculo de este factor.

es 0.25, la GM como función de fitness, la selección por mayoría, la distancia Euclídea, el balanceo activado y el valor de  $P$  es 0.2.

Para la comparativa hemos considerado la versión sensible al coste de C4.5 ( $C45\_CS$ ) [8] puesto que es habitualmente utilizado en este contexto y dos ensembles como AdaBoost.M2 [21] y RUSBoost [16]. Para realizar una comparativa justa hemos utilizado la misma configuración que en EUSBoost para los dos ensembles y  $C45\_CS$  es configurado como los clasificadores base de EUSBoost.

Además, hemos considerado dos métodos habitualmente utilizados para abordar el problema de este trabajo para contrastar los resultados obtenidos mediante EUSBoost:

- TRISS [17]. Es un método estándar para predecir la probabilidad de supervivencia y es utilizado para comprobar la calidad de los servicios de urgencias. Este método está basado en una regresión logística y sus variables de entrada son el ISS [13], el RTS [15] y la edad que es categorizada.
- Regresión logística desarrollada por el cuerpo médico del Hospital de Navarra [5] ( $RL_{HN}$ ). La variables de entrada contempladas son la edad, el RTS, el NISS y la comorbilidad previa.

Debemos destacar que para todos los métodos de la comparativa hemos utilizado las mismas variables de entrada que para  $RL_{HN}$ , es decir, la edad, el RTS, el NISS y la comorbilidad previa.

## 5.2. Comparación de EUSBoost y las regresiones logísticas

El objetivo de este estudio es comparar el rendimiento obtenido por EUSBoost con los resultados ofrecidos por los dos métodos basados en regresiones logísticas que se suelen utilizar en los hospitales para afrontar este problema.

En la Tabla 2 se muestran los resultados obtenidos por EUSBoost, TRISS, ( $RL_{HN}$ ),  $C45\_CS$ , AdaBoost.M2 y RUSBoost. Los resultados de cada medida de rendimiento se muestran por columnas. Estas son: AUC, media geométrica (GM), precisión en la clase mayoritaria ( $TN_{rate}$ ) y minoritaria ( $TP_{rate}$ ) y el *Accuracy* ( $Acc$ ). Para cada una de ellas se destaca en **negrita** el mejor resultado.

**Tabla 2.** Resultados obtenidos en test por los 3 clasificadores considerados en este estudio.

Clasificador	AUC	GM	$TN_{rate}$	$TP_{rate}$	Acc
EUSBoost	<b>0.84</b>	<b>0.83</b>	0.80	<b>0.88</b>	0.81
TRISS	0.70	0.65	0.95	0.45	0.84
$RL_{HN}$	0.76	0.71	<b>0.96</b>	0.56	<b>0.88</b>
$C45\_CS$	0.77	0.76	0.70	0.84	0.73
AdaBoost.M2	0.72	0.68	0.93	0.52	0.84
RUSBoost	0.81	0.80	0.83	0.80	0.82



De los resultados mostrados en la Tabla 2 podemos observar que EUSBoost ofrece los mejores resultados tanto en AUC como GM. En ambos casos la mejora de EUSBoost con respecto a los dos modelos basados en regresión logística es muy notable y está basada en el gran aumento en el  $TP_{rate}$ . *C45\_CS* consigue acercarse a AUSBoost en  $TP_{rate}$  pero a costa de empeorar notablemente el  $TN_{rate}$ , lo cual implica que obtiene peores resultados en AUC y GM. Finalmente, si analizamos los dos ensembles, podemos observar que AdaBoost.M2 presenta unos resultados similares a los de las regresiones logísticas mientras que RUSBoost ofrece unos resultados más competitivos pero sin llegar a la calidad de los ofrecidos por EUSBoost. Teniendo en cuenta todos estos hechos, podemos concluir que el algoritmo EUSBoost es muy apropiado para abordar el problema de este trabajo puesto que ofrece un buen balance entre la clasificación de ambas clases.

## 6. Conclusiones

La predicción de la supervivencia de los pacientes poli-traumatizados es un problema muy importante para los servicios de urgencias de los hospitales. Estos problemas son no balanceados puesto que el número de pacientes que sobreviven a sus heridas y mucho mayor que el que fallece. Para resolver este problema se han utilizado de forma tradicional métodos basados en regresiones logísticas. Dichos clasificadores ofrecen buenos resultados pero como todos los métodos de minería de datos pueden sufrir problemas derivados de las características especiales de los problemas no balanceados.

En este trabajo hemos aplicado el algoritmo EUSBoost para abordar dicho problema. Esta técnica es un ensemble construido mediante la hibridación de Boosting y BME. La calidad de sus resultados se ha comparado con la de los ofrecidos por dos regresiones logísticas utilizando 462 pacientes almacenados en el MTRN. Los resultados obtenidos muestran que EUSBoost es una técnica muy apropiada para abordar este problema puesto que permite mejorar los resultados obtenidos por todos los clasificadores utilizados en la comparativa en términos de AUC y GM, puesto que obtiene un buen balance entre el porcentaje de acierto en los pacientes que sobreviven y que mueren.

**Acknowledgments.** Este trabajo ha sido financiado parcialmente por el Ministerio de Ciencia y Tecnología de España con el proyecto TIN2013-40765-P, por la red TIN2014-56381-REDT y por el Departamento de Salud del Gobierno de Navarra con el proyecto PI-019/11.

## Referencias

1. W. Haddon Jr., Advances in the epidemiology of injuries as a basis for public policy, Public Health Reports 95 (5) (1980) 411–421.
2. R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd Edition, John Wiley, 2001.

3. J. Sanz, D. Bernardo, F. Herrera, H. Bustince Sola, H. Hagaras, A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data, *Fuzzy Systems, IEEE Transactions on* In press. doi:10.1109/TFUZZ.2014.2336263.
4. J. Sanz, M. Galar, A. Jurio, A. Brugos, M. Pagola, H. Bustince, Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system, *Applied Soft Computing Journal* 20 (2013) 103–111.
5. T. Belzunegui, C. Gradín, M. Fortún, A. Cabodevilla, A. Barbachano, J. Sanz, Major trauma registry of navarre (spain): The accuracy of different survival prediction models, *American Journal of Emergency Medicine* 31 (9) (2013) 1382–1388.
6. N. V. Chawla, N. Japkowicz, A. Kolcz, Editorial: special issue on learning from imbalanced data sets, *SIGKDD Explorations* 6 (1) (2004) 1–6.
7. J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
8. K. M. Ting, An instance-weighting method to induce cost-sensitive trees, *IEEE Transactions on Knowledge and Data Engineering*, 14 (3) (2002) 659–665.
9. B. O. de Navarra, no. 79, 30 June 2010 - navarra.es., accessed 10/15/2010 (2010). [http://www.navarra.es/home/\\_es/Actualidad/BON/Boletines/2010/79/Anuncio-16/](http://www.navarra.es/home/_es/Actualidad/BON/Boletines/2010/79/Anuncio-16/)
10. S. Polinder, W. Meerdling, S. Mulder, E. Petridou, E. van Beeck, EURO COST reference group. Assessing the burden of injury in six European countries, *Bull World Health Organization* 85 (1) (2007) 27–34.
11. D. Pollock, P. McClain, Trauma registries: Current status and future prospects, *Journal of the American Medical Association* 262 (16) (1989) 2280–2283.
12. K. G. Ringdal, T. J. Coats, R. Lefering, S. Di Bartolomeo, P. A. Steen, O. Roise, L. Handolin, H. M. Lossius, The utstein template for uniform reporting of data following major trauma: A joint revision by scantem, tarn, dgu-tr and ritg, *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 16 (1) (2008) –.
13. S. Baker, B. O’Neill, W. Haddon Jr, W. Long, The injury severity score: a method for describing patients with multiple injuries and evaluating emergency care, *The Journal of trauma* 14 (3) (1974) 187–196.
14. L. W. Osler T, Baker SP, A modification of the injury severity score that both improves accuracy and simplifies scoring., *J Trauma* 43 (6) (1997) 922 – 925.
15. H. R. Champion, W. J. Sacco, W. S. Copes, D. Gann, T. A. Gennarelli, M. E. Flanagan, A revision of the trauma score, *Journal of Trauma* 29 (5) (1989) 623–629.
16. C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: A hybrid approach to alleviating class imbalance, *IEEE Transactions on Systems and Man and Cybernetics and Part A*, 40 (1) (2010) 185–197.
17. C. Boyd, M. Tolson, W. Copes, Evaluating trauma care: The triss method, *Journal of Trauma* 27 (4) (1987) 370–378.
18. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42 (4) (2012) 463 –484.
19. M. Galar, A. Fernández, E. Barrenechea, F. Herrera, EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, *Pattern Recognition*, 46 (2013), 3460–3471
20. S.García, F.Herrera, Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy, *Evolutionary Computation*, 17 (2009), 275–306.

21. R. E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, *Machine Learning*, 37 (1999) 297–336.
22. L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.