Máster en Ingeniería Computacional y Matemática

Trabajo Final de Máster

Análisis de datos clínicos en una Unidad de Cuidados Intensivos. Desarrollo de una metodología, procesos ETL y evaluación de modelos predictivos.

Rafael Reviriego Hernández
Máster en Ingeniería Computacional y Matemática (URV,UOC)
Inteligencia Artificial

Director TFM
David Riaño Ramos

15/Junio/2017

Este trabajo no habría sido posible sin la colaboración del grupo de investigación del departamento de Cuidados Intensivos del Hospital Universitario Joan XXIII de Tarragona:

Alejandro H Rodríguez MD, PhD.
Federico Esteban MD, PhD.
Gonzalo Sirgo Luanco MD, PhD.
Josep Gómez Álvarez MSc, PhD.
María Amparo Bodi Saera MD, PhD.

Y la ayuda del director del TFM:

David Riaño Ramos. Computer Science PhD. Associate Professor URV.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Análisis de datos clínicos en una Unidad de Cuidados Intensivos. Desarrollo de una metodología, procesos ETL y evaluación de modelos predictivos.*
Nombre del autor:	*Rafael Reviriego Hernández*
Nombre del consultor/a:	*David Riaño Ramos*
Nombre del PRA:	*Juan Alberto Rodríguez Velázquez*
Fecha de entrega (mm/aaaa):	*Jun/2017*
Titulación:	*Máster en Ingeniería Computacional y Matemática (URV, UOC)*
Área del Trabajo Final:	*Inteligencia Artificial*
Idioma del trabajo:	*Castellano*[1]
Palabras clave	*Healthcare IT, Machine Learning, MIMIC*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

La sanidad es uno de los sectores en los que el uso de técnicas avanzadas de análisis presenta más posibilidades de innovación y con mayor impacto social. En su aplicación es necesario tener en cuenta las características específicas de los datos clínicos: calidad, volumen, acceso y multimodalidad.

En este estudio se realiza un análisis de datos de pacientes ingresados en una Unidad de Cuidados Intensivos (UCI) con el objeto de estudiar la influencia de la variabilidad de la glucosa, las comorbilidades y otras variables clínicas y demográficas, en la mortalidad a corto plazo.

Se ha definido una metodología, basada en estándares, dividida en fases e iterativa, y evaluado herramientas de análisis avanzado y modelos predictivos.

Los procesos de obtención, preparación y transformación de datos se han desarrollado mediante una herramienta ETL con una arquitectura escalable, tomando como fuente la base de datos MIMIC-III, de acceso abierto y con datos reales anonimizados. Mediante test de hipótesis y usando la métrica AUC, se han comparado diferentes modelos, los mejores obtenidos (gbm, glm) tienen una eficacia superior (AUC 0.824) a los indicadores clínicos utilizados como referencia (AUC 0.725). Se ha cuantificado la ganancia de incluir las

[1] En la memoria se ha optado por mantener algunos términos en idioma inglés cuando: no existe en castellano un equivalente único, es un término usado de forma habitual por la comunidad, y/o se evitan ambigüedades. Se ha utilizado tipografía cursiva para identificar estos términos anglosajones.

comorbilidades del paciente en el análisis y estudiado la importancia de las variables de glucosa en varios grupos de pacientes.

La metodología definida, junto con los procesos ETL desarrollados, pueden servir de base para otros estudios o de orientación en la construcción de una base de datos MIMIC local.

Abstract (in English, 250 words or less):

Healthcare is one of the sectors in which the use of advanced analytical techniques presents more possibilities of innovation and with greater social impact. Anyway, it is necessary to take into account the specific characteristics of the clinical data: quality, volume, access and multimodality.

In this study, an analysis of the data from critical patients was carried out in order to study the influence of glucose variability, comorbidities and other clinical and demographic variables, in short-term mortality.

A methodology has been defined, based on standards, with iterative phases, and an evaluation of advanced analysis tools and predictive models have been performed. Data collection, preparation and transformation processes have been developed using an ETL tool with a scalable architecture, using the open access MIMIC-III database as a source of real anonymized data. Different predictive models have been compared with hypothesis tests using the AUC metric, the best models obtained (gbm, glm) have better AUC (0.824) than the clinical scores used as reference (0.725). The gain of including the patient's comorbidities in the analysis has been quantified and the importance of the glucose variables in several patient groups has been studied.

The defined methodology, together with the developed ETL processes, can serve as the basis for other studies or for guidance in the construction of a local MIMIC database.

Índice

Lista de figuras

Lista de tablas

1 INTRODUCCIÓN

El presente trabajo consiste en un análisis de datos en una unidad de Cuidados Intensivos y la evaluación de modelos predictivos aplicados al área de datos sanitarios. Se estudia la incidencia de la variabilidad de la glucosa, las comorbilidades[2] y otras variables clínicas y demográficas de pacientes, en la mortalidad antes de 28 días. Como parte de este trabajo se ha definido una metodología, desarrollado un conjunto de procesos de extracción, transformación y carga (ETL), que obtienen y transforman los datos de la base de datos (BBDD) de origen para su análisis, y evaluado dos *frameworks* sobre el lenguaje R que facilitan las tareas asociadas al análisis de datos y el desarrollo de modelos predictivos. La base de datos utilizada para el estudio, MIMIC-III, proviene de una base de datos sanitaria real, con datos anonimizados para evitar la identificación de los pacientes.

Este trabajo ha contado con la colaboración del equipo de investigación del Servicio de Medicina Intensiva del Hospital Universitario de Tarragona Joan XXIII.

1.1 Contexto y justificación del Trabajo

En los pacientes ingresados en unidades de cuidados intensivos (UCI) es común la aparición de hiperglucemia aguda (90% presentan concentraciones en sangre>110 mg/dl) [2]. Generalmente se ha considerado importante el control de la concentración de glucosa en sangre en pacientes críticos por su asociación estadística con el incremento de la mortalidad [3], [4], aunque otros estudios muestran una fuerte evidencia de correlación pero no de causalidad [5].

El control de la glucosa en sangre se logra mediante el uso de insulina IV (intravenosa) y la medición de los niveles de glucosa de forma periódica para ajustar la velocidad de perfusión y la dosis. La asociación entre la hiperglucemia y la mortalidad ha dado lugar a recomendaciones de control estricto del nivel de glucosa en sangre (140-180 mg/dl) [6], [7]. Otros estudios recomiendan relajar este tipo de control, por no encontrar evidencias significativas de su asociación con la mortalidad de los pacientes en general [8], [9], y usar un control más moderado que permita una variación más amplia del índice de glucosa (100-180 mg/dl) [10], [11]. Existen también evidencias de que el control de glucosa mediante TIR (*Time in Range*) está relacionado con la probabilidad de supervivencia del paciente [12].

Otros estudios muestran que la variabilidad de la glucosa, que se define como la desviación estándar de la media de los valores de glucosa en un intervalo temporal dado, es un predictor más potente del riesgo de

[2] Las comorbilidades se refieren a la presencia de otras enfermedades y patologías en un paciente además de la diagnosticada como principal. Se obtienen a partir de la codificación CIE9 [1].

mortalidad que la media de la concentración de la glucosa en sangre [2], [13],[14], viéndose modulada por el estado diabético o no del paciente. Pacientes con diabetes no muestran una asociación entre los niveles de glucosa y su variabilidad y la mortalidad.

En general, tanto la hiperglucemia e hipoglucemia como el incremento de la variabilidad de la glucosa, están relacionados de forma independiente con el incremento del riesgo de mortalidad en los pacientes críticos y sus efectos dependen del estado diabético del paciente [15], [16]. En recientes estudios se obtiene una asociación de la mortalidad a 30 días con la variabilidad de la glucosa en pacientes tratados con insulina intravenosa, tanto en diabéticos como no diabéticos e independiente de otras variables como hipoglucemia, edad, severidad y comorbilidades [17].

Como conclusión, existe un debate abierto sobre la causalidad de la hiperglucemia, hipoglucemia y la variabilidad de la glucosa en la probabilidad de supervivencia del paciente, así como sobre los protocolos adecuados de control, apuntándose a la automatización de dicho control [18] y la monitorización continua [19] como las principales vías de avance.

La salud es uno de los sectores en los que el uso de las nuevas tecnologías de *Big Data* y *Machine Learning* presentan más posibilidades de innovación y con mayor impacto social. Solo en Europa el gasto público en sanidad se prevé que aumente del 8% del PIB en el año 2000 al 14% en el año 2030 [20] y tendrá que enfrentarse a grandes desafíos en las próximas décadas, como el envejecimiento de la población, las crisis migratorias, riesgos de nuevas epidemias y el coste creciente de las tecnologías clínicas. En el año 2012, el *Poneman Institute* estimaba que el 30% de los datos digitales almacenados en todo el mundo correspondía a datos de salud [21]. Extraer conocimiento de esta ingente cantidad de datos se considera la forma más rápida, eficiente y menos costosa de incrementar la salud de la población general [20].

La utilización de tecnologías de información y análisis de datos es un factor fundamental para avanzar hacia ese objetivo, aunque las características del sector implican que el tratamiento automatizado de los datos de salud y la realización de estudios e investigaciones presentan dificultades específicas, entre ellas [22]:

- **Calidad de datos:** La calidad de los datos registrados y su estructura son críticos para utilizar herramientas de análisis avanzado, pero no es la prioridad cuando se trata de atender a pacientes. Muchos de los informes, anotaciones, etc, se registran en forma de texto libre o anotaciones no estructuradas. Existe un conflicto entre la rapidez y utilidad inmediata del dato clínico y la facilidad de tratamiento y análisis a posteriori.
- **Volumen de datos:** el registro automatizado de variables clínicas, resultados de laboratorio, escáneres, dispositivos de diagnóstico por imagen, etc., genera una cantidad considerable y creciente de datos.

- **Datos multimodales:** el sector sanitario se caracteriza por una diversidad de tipos de datos: imágenes, texto libre, anotaciones clínicas, datos estructurados, múltiples estándares de codificación, mediciones en tiempo real, gráficas temporales, etc, obtenidos por sistemas de características muy diferentes y con más o menos posibilidades de integración entre sí.
- **Acceso de datos:** las propias características de las organizaciones sanitarias y los requisitos de las leyes de protección de datos de salud establecen una serie de limitaciones que hacen prácticamente imposible realizar estudios o análisis de datos integrando distintas fuentes independientes o de forma pública y accesible.

Debido a estas características de los datos de salud, los dos frentes [22] más importantes que permiten avanzar en el tratamiento de datos sanitarios son: la interoperabilidad de los diferentes sistemas [23], [24] y las nuevas tecnologías de *Big Data* [25], [26], [27] , incluyendo en ellas técnicas que no son propiamente *Big Data*, pero que habitualmente entran por extensión dentro de la misma denominación: ETL (*Extraction, Transformation and Load*), análisis de datos avanzado, *Machine Learning*, modelos estadísticos, etc.

1.2 Objetivos del Trabajo

La **propuesta** de estudio parte de la problemática y el contexto indicados y marca como líneas principales:

- Evaluación de metodologías y herramientas apropiadas para: a) automatizar la selección, extracción y tratamiento de datos de una BBDD clínica real, b) realizar las tareas de búsqueda de parámetros, entrenamiento, validación y comparación entre diferentes modelos de aprendizaje supervisado adecuados a los datos de origen y c) obtener conocimiento a partir de los modelos obtenidos.
- Aplicación de los modelos predictivos y herramientas seleccionados.

Los **objetivos principales** son:

- Obtener modelos de predicción de la mortalidad a 28 días de los pacientes de UCI (Unidad de Cuidados Intensivos), teniendo como principales variables predictores la variabilidad de la glucosa y variables asociadas, junto con las empleadas habitualmente en la práctica clínica para la valoración del estado del paciente y las comorbilidades. Se utilizarán indicadores obtenidos en los tres primeros días de estancia en UCI.
- Aplicar los modelos obtenidos para analizar la importancia de la variabilidad de la glucosa en la predicción de la mortalidad.

Como **objetivos adicionales** se consideran:

- Diseñar una arquitectura escalable y aplicable a un sistema real.
- Identificar los modelos adecuados a los datos en estudio.
- Evaluar la influencia de las variables de las comorbilidades en la capacidad predictiva de los modelos.

En la concreción de la propuesta, y siguiendo criterios clínicos, se decidió incluir, además de las mediciones relacionadas con la glucosa (variabilidad, medias, desviación estándar, hipoglucemia, hiperglucemia, tiempo en rango, etc.), otras variables obtenidas en los primeros días de estancia en UCI: datos demográficos, días de estancia, edad, score SOFA [28], comorbilidades, diagnóstico de ingreso, sepsis, administración de vasoactivos, ventilación mecánica, diálisis, administración de insulina IV, administración de esteroides, tipo de nutrición y analítica básica diaria (exceso de bases, hemoglobina, lactosa, PCO2, PH, PO2, bicarbonato, leucocitos). También con criterios médicos se decidió que la población incluida en el estudio debe cumplir los siguientes criterios:

- Pacientes adultos.
- Estancia en UCI mayor que dos días.
- Más de tres mediciones de glucosa en cada uno de los tramos de 24H y 48H.
- Con diagnostico principal codificado.

1.3 Enfoque y método seguido

Para el desarrollo del estudio se ha utilizado MIMIC-III, una base de datos clínica anonimizada de uso libre, cuyas características e idoneidad para este trabajo se describirán en el capítulo 4.

El enfoque principal que ha guiado el trabajo ha sido utilizar tecnologías fácilmente replicables en un entorno investigador/sanitario y que los procesos desarrollados se puedan reutilizar con pequeñas modificaciones para otra selección de variables diferente o incluso con otra BBDD de origen.

Este criterio ha sido el empleado para decidir el uso de software *Open Source* para el diseño de la arquitectura, R como lenguaje de análisis de datos y el desarrollo de los procesos ETL mediante Pentaho PDI, tal como se detallará en los capítulos correspondientes. Los detalles más técnicos, estructuras de las BBDD, detalle de los procesos ETL, etc. se incluyen en los anexos.

En un primer paso se definió la metodología a seguir, basándose en estándares ya existentes, se diseñó y construyó la arquitectura técnica necesaria para llevar a cabo los análisis y se analizó la BBDD MIMIC-III al objeto de obtener la información necesaria para desarrollar los procesos ETL que obtienen y seleccionan los datos, realizando a continuación varias iteraciones de preparación, modelado y evaluación de modelos, hasta llegar a la fase de documentación e informe.

A grandes rasgos, el enfoque seguido y sus diferentes fases se muestran en la figura 1.

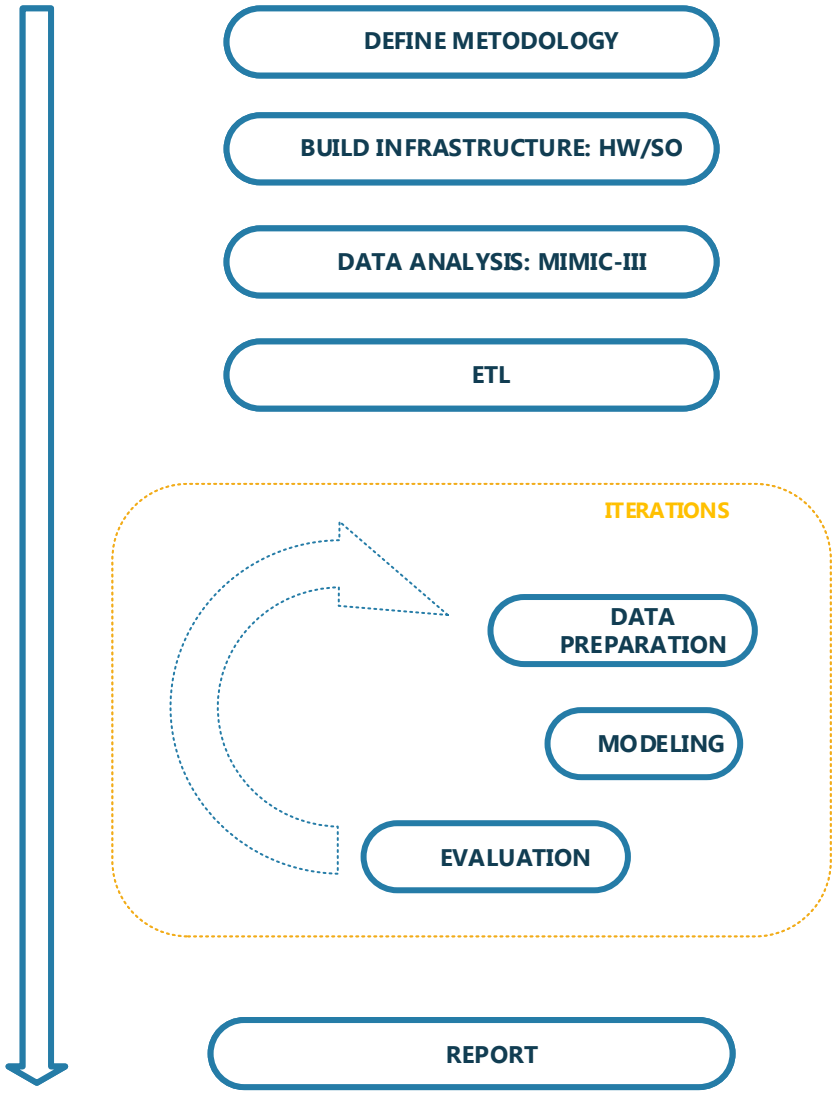

Figura 1: Fases del enfoque del trabajo

1.4 Planificación del Trabajo

Como primer paso en la elaboración del trabajo, y una vez definida la propuesta, los objetivos y el contexto de forma general, se realizó una planificación de las tareas a realizar a lo largo de 4 meses, marcando los hitos principales y el camino crítico. Dada la naturaleza del trabajo, de carácter exploratorio y evaluativo de diversas tecnologías, con riesgo de problemas técnicos, se consideró una planificación general de tipo iterativo, en la que en función del tiempo disponible se podrían realizar más o menos iteraciones de la fase de análisis. Esta planificación iterativa permite absorber posibles riesgos sin afectar a la planificación general del trabajo en las fases finales.

A lo largo de la duración del trabajo se han mantenido reuniones de seguimiento con periodicidad aproximadamente quincenal con el director del TFM y con el equipo de investigación del Hospital Universitario Joan XXIII implicado en el estudio. En estas reuniones se han presentado informes con los resultados parciales y se han recogido las valoraciones e indicaciones para los siguientes pasos. Por ejemplo, el objetivo adicional de comprobar la influencia de las variables de las comorbilidades en la capacidad predictiva de los modelos se definió a lo largo de estas reuniones de seguimiento.

La figura 2 muestra el diagrama de GANTT con la planificación inicial de las diferentes tareas del trabajo.

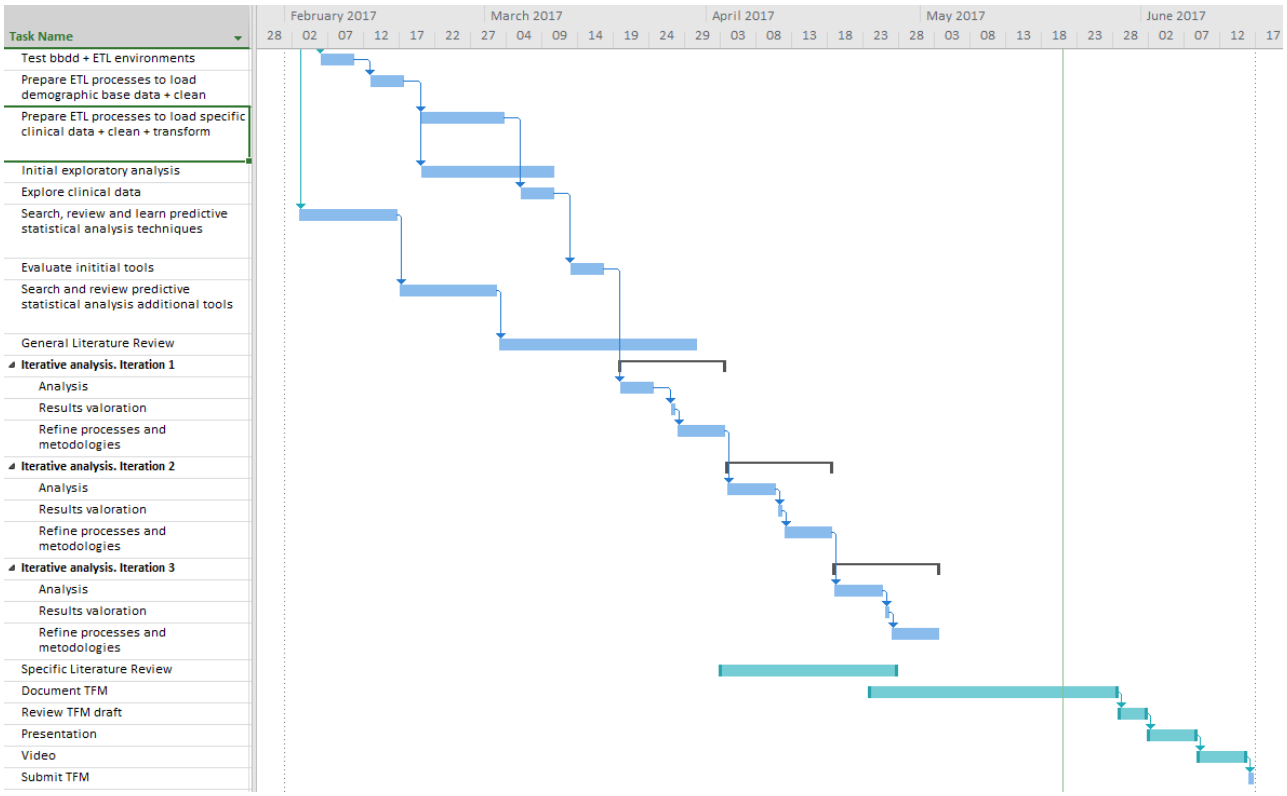

Figura 2: Diagrama de GANTT

1.5 Breve sumario de productos obtenidos

Se han obtenido los siguientes entregables, además de la presente memoria:

- Documentación técnica sobre la obtención de variables de interés de la BBDD MIMIC-III.
- Documentación técnica de las BBDD utilizadas.
- Catálogo XML de los procesos ETL para obtener y transformar los datos de la BBDD MIMIC-III a la BBDD intermedia, con una estructura adecuada para este análisis y otros similares.
- Manuales de instalación de MIMIC-III y Pentaho PDI.

- Código fuente de las transformaciones realizadas y de las tareas de *tunning*, aplicación y evaluación de modelos en R.

1.6 Breve descripción de los otros capítulos de la memoria

Esta memoria se ha estructurado de la siguiente forma, en el capítulo 2 se describirán dos metodologías existentes en el campo de la minería y análisis de datos y la combinación de ambas que se ha adaptado para este trabajo. Seguidamente en el capítulo 3, se detallará de forma breve la arquitectura *hardware*, *software* y herramientas utilizadas, pasando en el capítulo 4 a presentar la BBDD que ha servido como origen de datos, MIMIC-III. En el capítulo 5 se describirán los procesos ETL desarrollados para obtener los datos y procesarlos para su análisis mediante modelos predictivos. El fundamento teórico de los modelos usados se presenta en el capítulo 6, junto con las técnicas de evaluación, pasando en el capítulo 7 a describir los resultados más relevantes de los análisis realizados, cerrando en el último capítulo con las conclusiones y posibles líneas de trabajo futuro.

Se han incluido anexos con información técnica adicional y de detalle de la estructura de MIMIC-III, documentación de los procesos ETL, estructura de las tablas de la BBDD intermedia MIMICSEL, los campos incluidos en el análisis, un breve manual de instalación de MIMIC-III y la herramienta ETL Pentaho PDI, los criterios más importantes usados para obtener las variables del estudio y ejemplos del código fuente en R empleado en las diferentes fases del análisis.

2 METODOLOGÍA

Se describen en este capítulo dos metodologías estándar empleadas en proyectos de *datamining* y *Big Data*, sus características y la combinación de ambas utilizada finalmente en este trabajo.

2.1 CRISP-DM

La metodología CRISP-DM [29] (*Cross Industry Standard Process for Data Mining*) sigue siendo la más empleada en proyectos de *datamining*, a pesar de que ya no se mantiene [30]. Esta metodología divide los proyectos de minería de datos en varias fases realizadas de forma iterativa hasta alcanzar los resultados deseados. La figura 3 muestra el ciclo de vida de los proyectos de minería de datos según está metodología:

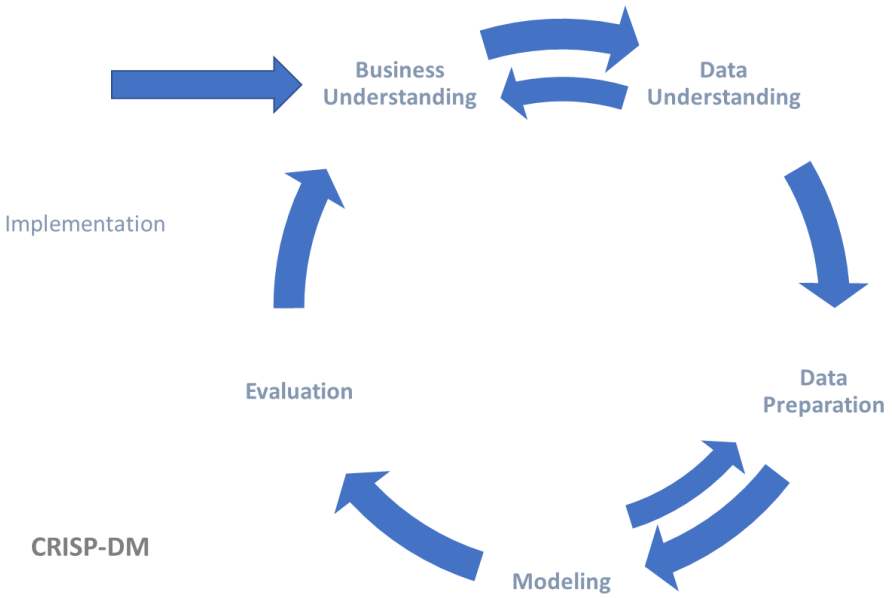

Figura 3: Metodología CRISP-DM

Los diferentes ciclos del modelo comienzan siempre en el punto inicial de conocimiento del sector *(Bussiness Understanding)* y terminan con la implementación *(Implementation)* del producto, modelo o estudio concreto. Durante el resto de fases se producen continuamente saltos adelante-atrás, dado que los resultados de una fase influyen tanto en las anteriores como en las posteriores.

La primera fase de **Conocimiento del Sector** (*Businness Understanding***)** se enfoca en entender el objetivo del proyecto, el contexto y realizar el plan preliminar. En la fase de **Conocimiento de Datos** (*Data Understanding*) se incluyen las tareas relacionadas con la recogida de los datos, el análisis exploratorio y la revisión de la calidad de los datos. Durante la fase de **Preparación de Datos** (*Data Preparation*) se seleccionan los datos que se utilizarán para las fases sucesivas, se realiza

la limpieza de datos si es preciso, se obtienen datos derivados a partir de los iniciales, se modifica la estructura, y en general todas las tareas incluidas en el tratamiento previo de los datos antes de ser utilizados por los diferentes modelos. Una vez que se dispone de los datos seleccionados y preprocesados, se procede a realizar la fase de **Modelado** (*Modeling*), en la que se utilizan los diferentes modelos predictivos. Dentro de esta fase se incluye el particionado de los datos en subconjuntos de entrenamiento y validación, la búsqueda de los parámetros adecuados para cada modelo y la comparativa de los modelos. En la fase de **Evaluación** (*Evaluation*) se revisan y evalúan los resultados obtenidos y se determina si se pasa a la fase de implantación o se refina alguno de los procesos anteriores volviendo a iniciar el ciclo en otro punto con las modificaciones requeridas. Por último, una vez desarrollados todos los pasos anteriores, se realizaría la fase de **Implantación** (*Deployment*), que dependiendo de las características concretas del proyecto puede ser: la redacción de un informe, una publicación, el despliegue en producción de un sistema, su modificación o mantenimiento, etc. La figura 4 muestra un resumen de las principales acciones en cada fase de la metodología CRISP-DM.

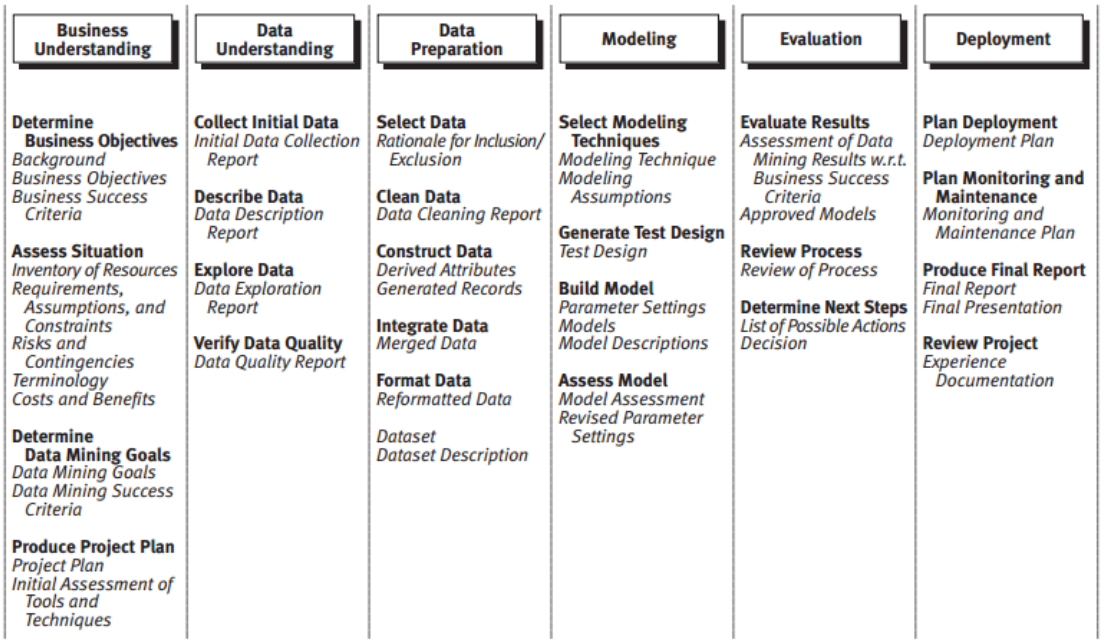

Figura 4: Acciones desarrolladas en las distintas fases de CRISP-DM. Gráfica obtenida de la documentación de referencia de la metodología CRISP-DM [31]

2.2 Healthcare Big Data Methodology

Aunque la metodología CRISP-DM es suficientemente amplia para incluir cualquier tipo de análisis y explotación de datos en campos diversos de la industria, en el sector sanitario existen estudios que describen una metodología más concreta, práctica y adecuada para la aplicación de técnicas de *Big Data*, teniendo en cuenta las características específicas del sector [25].

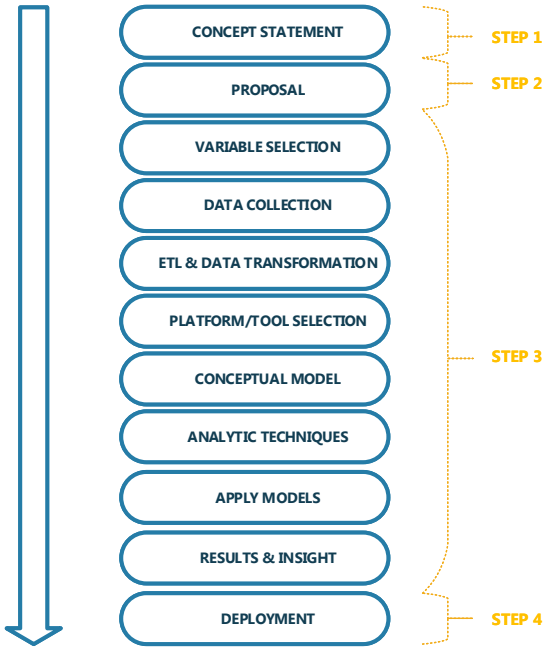

Figura 5: Metodología Healthcare Big Data.

Las fases de esta metodología se muestran en la figura 5 (Elaboración propia adaptada de [25]). En el primer paso, equipos interdisciplinares de analistas técnicos y especialistas clínicos definen el ***Concept Statement,*** donde se establece la necesidad del proyecto. En la fase de ***Proposal*** se desarrollan en detalle los requisitos y se analiza el contexto y los conocimientos previos. En los siguientes pasos se lleva a cabo la obtención y tratamiento de los datos, la selección de tecnologías y el modelado y evaluación, para acabar obteniendo en la fase de ***Deployment*** los resultados del proyecto.

2.3 Metodología adaptada

A partir de estas dos metodologías descritas, combinado sus tareas y fases se ha desarrollado una metodología adaptada a las características del presente trabajo, escalable y fácilmente aplicable en proyectos similares dentro del área sanitaria. Las fases seguidas en la metodología utilizada se describen a continuación y en la figura 6 se realiza un mapeo de dichas fases con las dos metodologías estándar descritas anteriormente.

Propuesta *(Proposal)***:** Fase inicial en la que se plantea el problema a tratar, se analiza su contexto y se definen los objetivos. Esta fase incluye la revisión de la literatura y la consulta a fuentes de conocimiento experto, tanto del apartado más técnico del trabajo: IA, modelos, técnicas y herramientas, como del ámbito específico del estudio: BBDD sanitarias, conceptos clínicos, funcionamiento de una UCI, estudios sobre el impacto de la variabilidad de la glucosa en la mortalidad, otros estudios clínicos que aplican técnicas de modelos predictivos, etc.

ETL y Preparación de datos *(ETL & Data Preparation)***:** Una vez definido el problema, su contexto y los objetivos, se identifican las fuentes de datos disponibles, el tipo y estructura de los datos y el tratamiento o agregación necesarios, la arquitectura técnica apropiada, las herramientas ETL y se desarrollan y testean los procesos utilizados para obtener los datos desde los orígenes identificados. Así, durante esta fase, primero se ha realizado una selección previa de variables con ayuda experta, se ha identificado la variable a estudiar *(outcome)*, las características de la población y una serie de variables adicionales, basándose en criterios clínicos, estadísticos y técnicos. Se han desarrollado procesos ETL que obtienen dichas variables, así como campos de agregación para cada una de ellas, por ejemplo, la media, máximo, mínimo, número y varianza en tramos de 24 horas desde el ingreso. En los criterios clínicos de selección de variables se engloban las decisiones de mantener determinadas variables, no por su capacidad predictiva, sino por su validez clínica, como pueden ser en este caso la inclusión del score SOFA[3] [28], los diagnósticos codificados con CIE9[4] [32]–[34] y las medidas relacionadas con la variabilidad de la glucosa[5]. En los criterios estadísticos se eliminan las variables fuertemente correlacionadas o con varianza casi nula, entre otros.

En esta fase, el procedimiento seguido en este estudio difiere de los otros dos modelos mencionados. Se decidió realizarla de forma que primero se desarrollaron procesos ETL para obtener un conjunto amplio de variables y se realizó posteriormente una selección de las mismas para reducirlas a un conjunto más limitado, en base a dos puntos principalmente: 1) es más eficiente incluir en el procesamiento variables que se transforman en bloque de la misma forma, que modificar los procesos ETL para ir añadiendo cada variable cuando sea necesaria. Por ejemplo, es más sencillo, y con menos riesgo de error, obtener las medias por tramos de 24H para todas las variables a la vez, que modificar la ETL cuando se necesite un nuevo valor medio de una variable determinada para un tramo horario, y 2) que desde el punto de vista estadístico tiene sentido observar el comportamiento de los datos una vez que se han transformado por los procesos ETL, no antes, dado que las propias transformaciones pueden modificar su comportamiento.

Modelo conceptual *(Conceptual Model)***:** Una vez realizada la fase anterior y con el conjunto de datos ya seleccionado y preparado, se realiza la identificación del tipo de modelo predictivo a evaluar y se obtienen los requisitos para seleccionar las herramientas necesarias. En este estudio se optó por realizar un análisis de clasificación binaria, con el o*utcome* mortalidad antes de los 28 días. Con los datos obtenidos por la ETL se podrían realizar análisis con otros tipos de modelos: clasificación

[3] Indicador que clasifica a los pacientes que ingresan en una UCI en función de su riesgo de mortalidad basado en datos clínicos y resultados de laboratorio de las primeras 24H.
[4] Estándar internacional de codificación de enfermedades. La versión actual es la 10, aunque la versión 9 todavía es la utilizada mayoritariamente en algunos países, entre ellos España.
[5] Ver apartado de contexto 1.1.

multinivel, regresión, otras variables *outcome,* etc, por eso esta fase se realiza después de la de ETL.

Iteraciones de **Selección de Plataforma/Herramientas** y **Modelado** *(Platform/Tool Selection and Modeling)***:** Una vez realizados todos los pasos anteriores, se realizan varias iteraciones de las tareas de selección de herramientas y *frameworks*, y de selección de modelos y herramientas, experimentando con diferentes configuraciones, evaluando la capacidad predictiva de los modelos y la eficacia de los diferentes *frameworks* y herramientas disponibles. De este proceso iterativo se obtiene una configuración optimizada para el problema y los datos utilizados. En función de los resultados del modelado se pueden realizar modificaciones en la selección de variables o los procesos de la ETL.

Finalmente se obtienen las conclusiones y se indican las posibles vías de desarrollo futuro en las fases de **Resultados, Conocimiento adquirido e Implantación** (*Results & Insights, Deployment*).

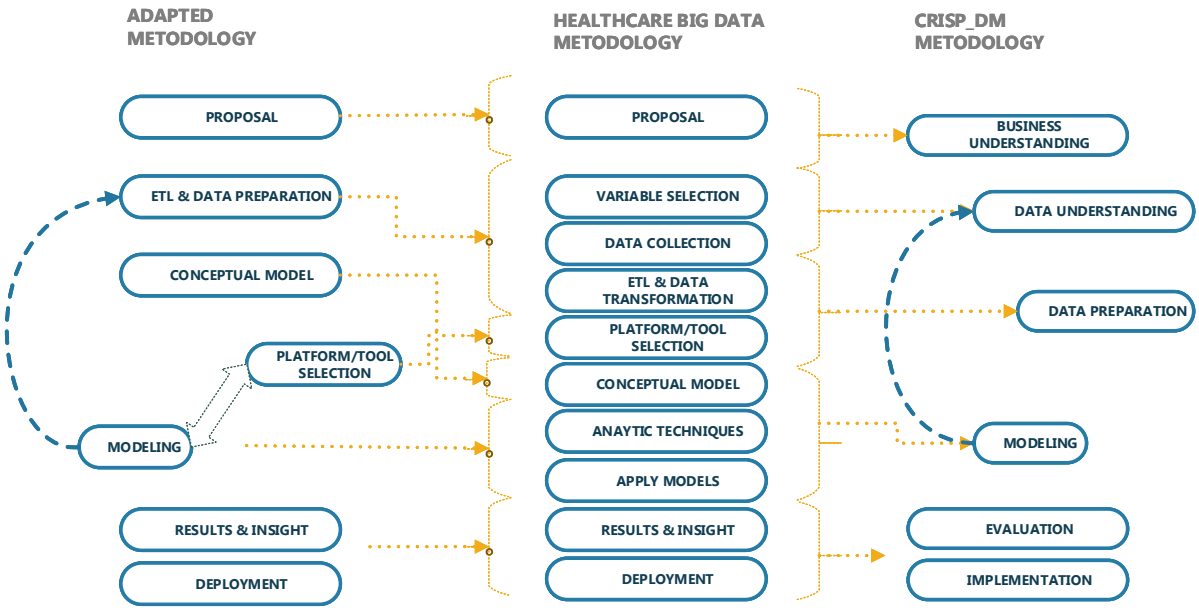

Figura 6: Mapeo entre CRISP-DM, Healthcare Big Data Metodology y la metodología adaptada.

En los siguientes capítulos se profundizará en los apartados de esta metodología adaptada. En primer lugar, se describirá la arquitectura diseñada e implementada, tanto a nivel de hardware como software, para entrar a continuación en los detalles de la BBDD MIMIC-III, que juega un papel importante en este trabajo, a partir de ahí se mostrarán los procesos ETL desarrollados y los datos obtenidos mediante su uso. Todas estas tareas se engloban dentro de la fase de **ETL y Preparación de datos**. Posteriormente, se describirán los modelos predictivos utilizados y las métricas que permiten evaluarlos, dentro de la fase de **Modelo Conceptual** y se continúa con el análisis de los datos de las fases iterativas de **Selección de Plataforma/Herramientas** y **Modelado** para terminar con la valoración de los **Resultados.**

3 ARQUITECTURA

Uno de los objetivos adicionales del presente estudio consiste en diseñar una arquitectura capaz de realizar el tratamiento de conjuntos de datos de tamaño grande, que sea modular y fácilmente replicable. Para ello es primordial implantar los diferentes procesos en un entorno similar al que podríamos encontrar en cualquier ámbito académico/hospitalario, en lugar de construir un caso de ejemplo limitado a pocos datos y difícilmente escalable.

3.1 Arquitectura Hardware/SO

A lo largo de todo el estudio se han encontrado problemas de escalabilidad y de rendimiento en muchos de los algoritmos y modelos predictivos utilizados, por tanto, se ha incorporado al propio estudio un proceso de selección que permita una mejor combinación de los factores de poder predictivo, requisitos técnicos y capacidad de trabajar con el conjunto completo de datos.

En la figura 7 se muestra la arquitectura técnica con la que se han realizado las pruebas y experimentos computacionales. Las diferentes funcionalidades del proceso de obtención de datos y análisis se han separado en diferentes equipos, de forma que la carga de trabajo se reparte y se facilita así mismo la configuración y administración.

Para el diseño hardware se optó por no virtualizar ni utilizar tecnologías *Cloud* y en su lugar emplear equipos físicos, priorizando la relación coste/beneficio. Utilizar servidores en *Cloud* habría eliminado algunas limitaciones en cuanto a la capacidad de proceso y memoria, pero también habría significado un coste total de propiedad (TCO) superior al de la adquisición de un servidor con las características necesarias para el proyecto. En cuanto a la virtualización de los servidores, puede resultar más eficiente en un entorno en que se disponga ya de la capacidad de proceso y memoria, pero no en el caso de que sea necesario sobredimensionar equipos ya existentes para adaptarlos a los requisitos técnicos necesarios para la virtualización.

La arquitectura diseñada, en cuanto al hardware, consta básicamente de tres equipos físicos conectados mediante una red gigabit:

- **DATABASE / ANALYTICS SERVER:** servidor Linux con Sistema Operativo (SO) Ubuntu Server 16.04 / 8Gb de memoria y disco SSD para la base de datos y el sistema operativo. Este equipo se utilizó como servidor del gestor de base de datos PostgreSQL [35] en el que cargar MIMIC-III y el resto de bases de datos utilizadas en el proyecto. En la fase de análisis del proyecto se utilizó también como servidor de R con el software RStudio Server [36] y como nodo principal del software de análisis H2O [37].

- **ETL SERVER:** estación de trabajo con SO Ubuntu Desktop 16.10 / 8Gb de memoria y disco SSD. La función principal de este equipo es la de desarrollar y ejecutar los procesos ETL con la aplicación Pentaho PDI 7.0 [38], realizar análisis exploratorios mediante RStudio Desktop y albergar un nodo secundario de H2O.
- **PC Client:** estación de trabajo con Windows 7 / 16Gb de memoria y 2T de disco. Este equipo se utilizó para el desarrollo de los scripts SQL y la ejecución de las consultas a las bases de datos, el desarrollo de los scripts y programas en R utilizados en el análisis de datos y como equipo de gestión. En algunas pruebas se utilizó también para albergar un nodo adicional de H2O.

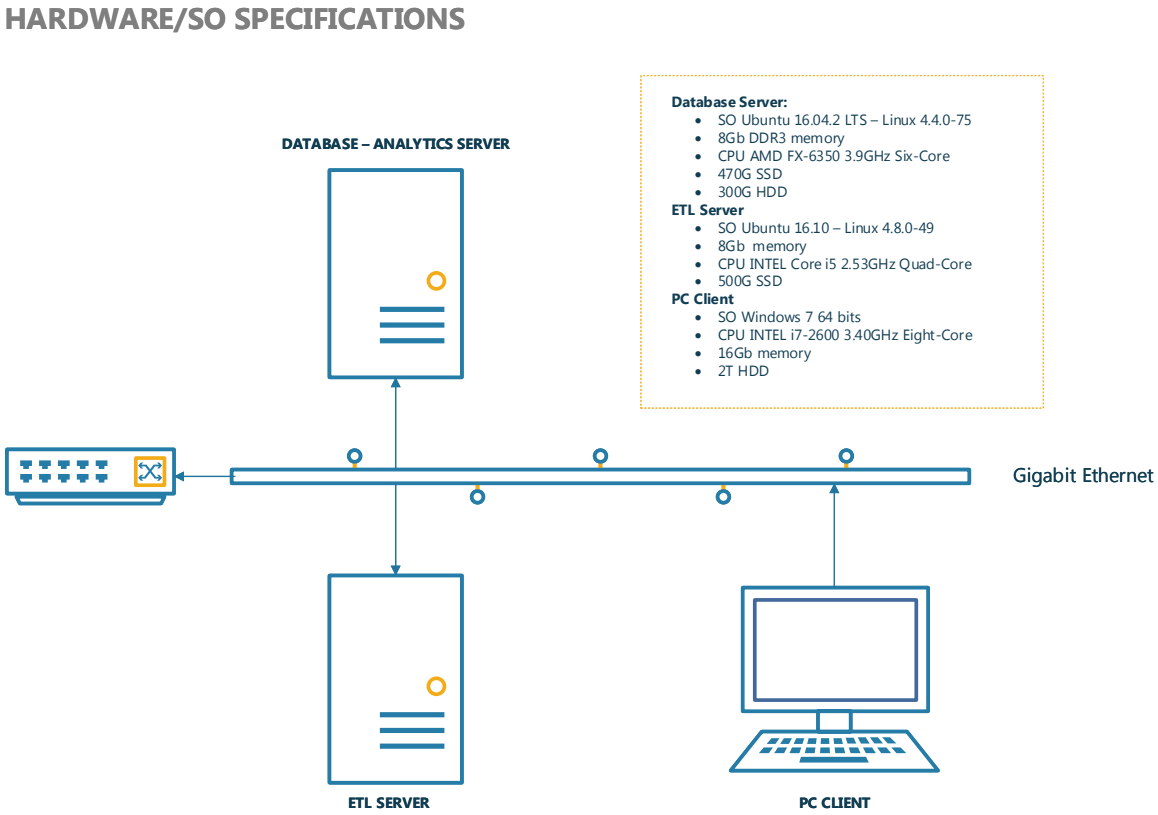

Figura 7: Arquitectura hardware

3.2 Arquitectura software

La arquitectura software sigue un esquema similar a la arquitectura hardware. Los objetivos principales para su diseño fueron: modularidad, facilidad de gestión y el reparto de la carga de proceso. Todo el software y las herramientas utilizadas tienen licencias abiertas del tipo *Open Source*, aunque también disponen de versiones de pago o subscripciones con soporte.

Las principales herramientas utilizadas para realizar el análisis se muestran en la figura 8. Todas son multiplataforma, aunque su uso sobre SO Linux facilita la gestión, simplifica la configuración y minimiza el coste.

SOFTWARE ARCHITECTURE

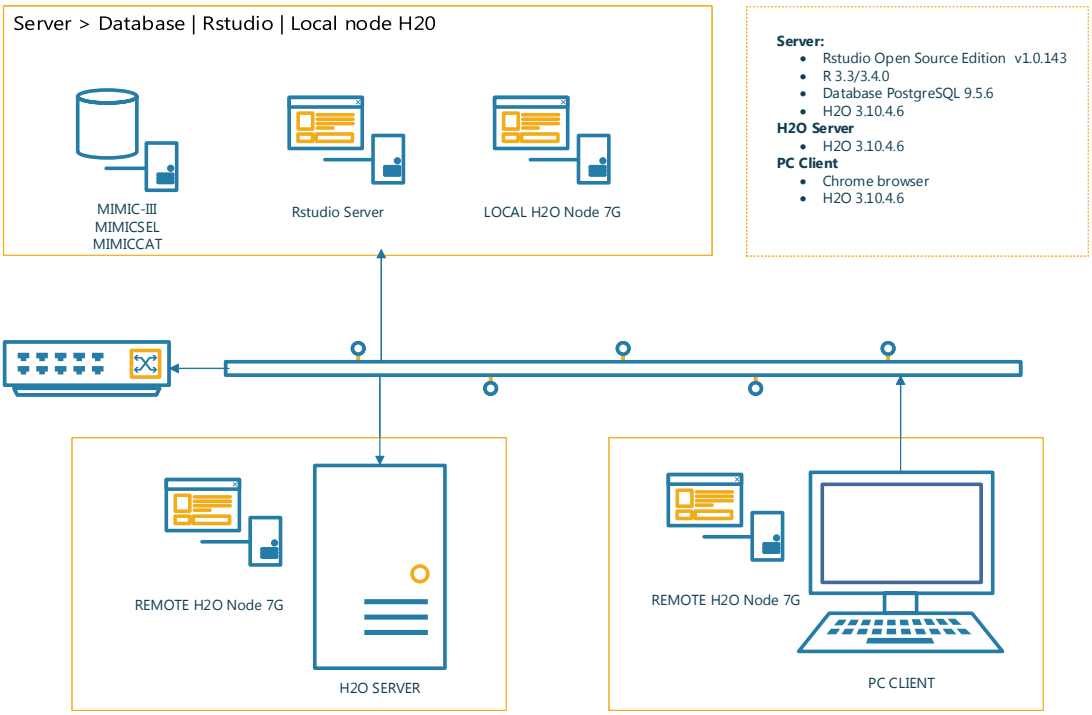

Figura 8: Arquitectura Software

Se detallan a continuación los diferentes componentes de la arquitectura software:

- **PostgreSQL:** [35] gestor de base de datos relacional SQL, cuenta con abundante documentación [39] y junto con MySQL [40] son los gestores de bases de datos relacionales *Open Source* más utilizados.

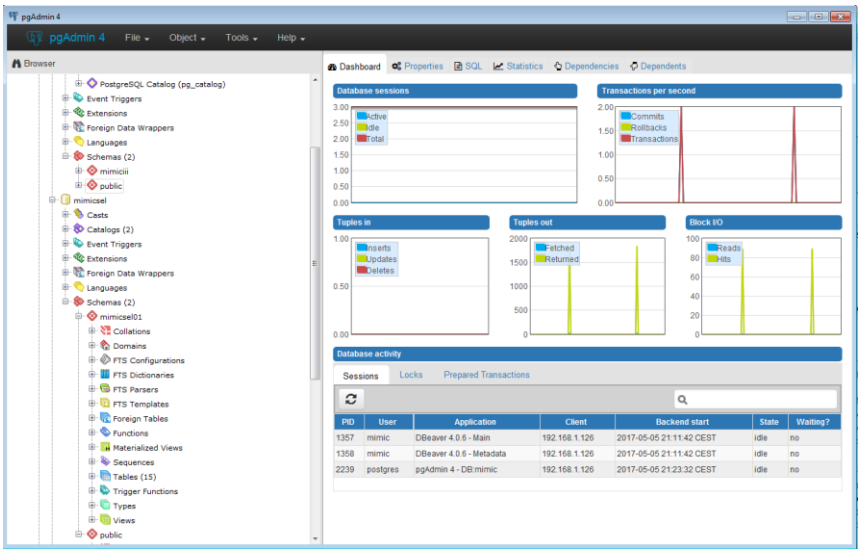

Figura 9: Interfaz de administración del gestor de bases de datos PostgreSQL

- **Pentaho PDI ETL:** [38] en el caso de las herramienta ETL, existen dentro del *Open Source* dos alternativas sólidas para realizar estos procesos: Pentaho y Talend [41]. Ambas soluciones disponen de versiones libres con suficiente potencia y flexibilidad para realizar cualquier proceso ETL, tanto por la librería de componentes que incluyen como por la posibilidad de programarlos a medida. La diferencia básica entre las dos aplicaciones es que Talend crea código Java ejecutable de forma independiente, mientras que los procesos de Pentaho se lanzan desde el entorno de desarrollo visual de la propia aplicación o desde línea de comandos.

  Tanto Pentaho PDI como Talend disponen de varios módulos enfocados a distintas áreas del tratamiento y análisis de datos, en el caso de Pentaho, el módulo que realiza las funciones de ETL y que se ha utilizado en este estudio es el PDI (*Pentaho Data Integration*), cuyo interfaz se muestra en la figura 10.

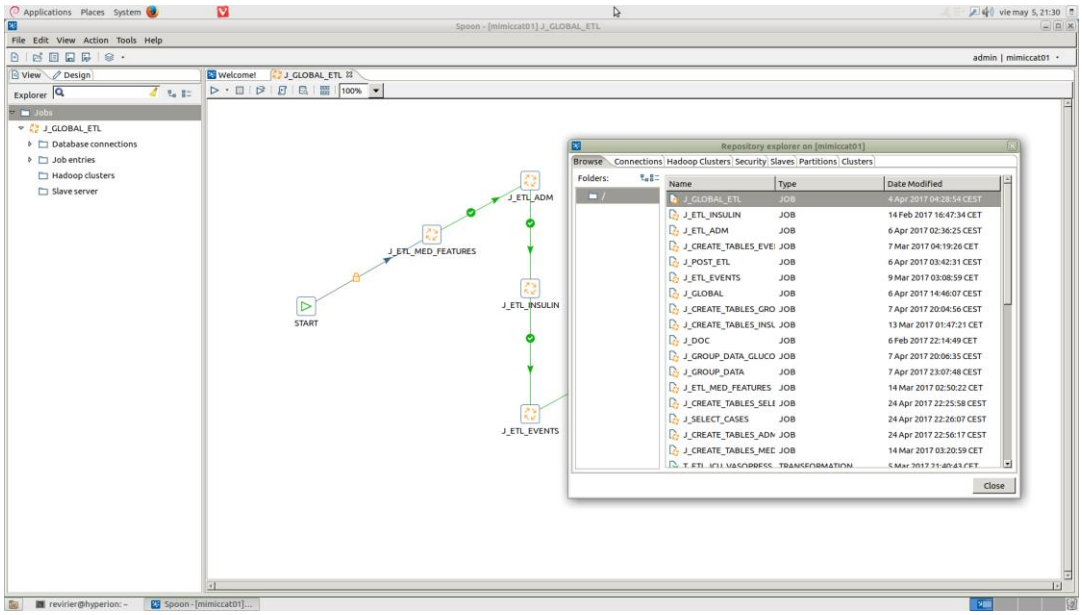

Figura 10: Interfaz de la herramienta ETL Pentaho PDI

Las principales características que han motivado la decisión de utilizar Pentaho PDI para este proyecto son las siguientes:

- **Sencillez de instalación**. No necesita instalación.
- **Facilidad de desarrollo**. Pentaho PDI dispone de un entorno de programación visual que facilita el diseño, implementación, prueba y depuración de los procesos ETL.
- **Rapidez de desarrollo**. Pentaho PDI es adecuado para desarrollos rápidos y prototipos.
- **Versatilidad**. Pentaho PDI dispone de pasos predefinidos con conexiones para múltiples sistemas de bases de datos, integración con otros sistemas (*Big Data, HL7, Web Services, etc.*) y soporte de múltiples formatos de fichero,

así como capacidad de programar en JavaScript o Java en caso de necesidades más específicas.
    - **Velocidad y paralelismo**. Pentaho PDI es capaz de procesar registros con una velocidad de 20000 a 100000 filas por segundo o más, dependiendo de la capacidad de los sistemas con los que se conecte, y permite paralelizar de forma muy sencilla los procesos.
    - **Multiplataforma.** Pentaho PDI funciona sobre java, es independiente del Sistema Operativo.
    - **Rapidez de aprendizaje.** La curva de aprendizaje de Pentaho PDI es extremadamente rápida.
    - **Comunidad activa.** Existe una gran comunidad de usuarios y documentación de soporte y ayuda disponibles.

- **R+RStudio:** [42] [36] **R** es uno de los lenguajes *Open Source* de análisis estadístico de datos más usados, dispone de miles de paquetes [43] con algoritmos y funciones de tratamientos de datos y análisis y es utilizado tanto por la comunidad investigadora como en el ámbito de la empresa.

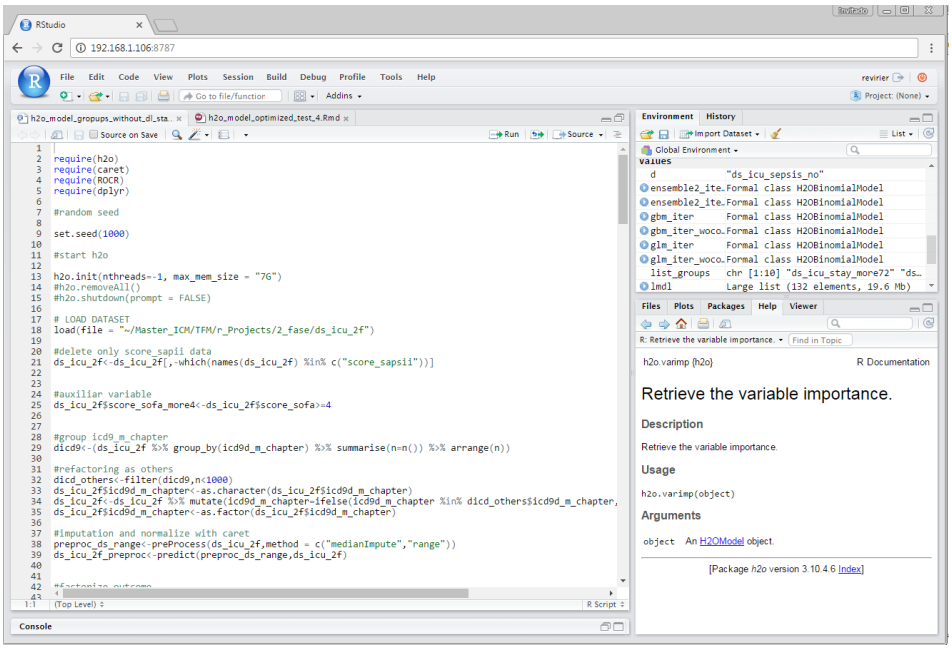

Figura 11: Interfaz RStudio Server en un navegador web

Aunque para el uso de modelos predictivos existen otras alternativas, principalmente Python con sckit-learn [44], librerías de *Machine Learning* para MATLAB [45], entornos completos como Rapid Miner [46], Weka [47] o Knime [48], se han considerado en este trabajo más importantes la flexibilidad, el potente ecosistema, la facilidad de análisis de tipo exploratorio y la existencia de distintos *frameworks* de aplicación de modelos predictivos del lenguaje R.

R se puede utilizar desde línea de comandos o desde un entorno IDE, el más utilizado es RStudio, que cuenta con versiones

profesionales y de pago y se puede utilizar tanto como entorno de escritorio *standalone* o en modo servidor (RStudio Server), accediendo desde un navegador web. Además del procesamiento remoto, la ventaja del RStudio Server es la persistencia de sesiones. En la figura 11 se muestra el interfaz de RStudio accediendo desde un navegador web.

- **H2O:** [37] es una plataforma *Open Source* de *Machine Learning* que funciona sobre Java y que dispone de varios modelos predictivos. Se puede utilizar como nodo local integrado con R o formando un cluster independiente con varios nodos. Utiliza procesamiento en memoria y paralelismo para obtener una gran rapidez de proceso. Tiene la desventaja de que incorpora solo unos pocos modelos predictivos, aunque frente a esa aparente limitación cuenta con una serie de ventajas que han decidido su utilización:

    - **Sencillez de instalación:** la instalación básicamente consiste en copiar un directorio.
    - **Sencillez de gestión:** crear un cluster o utilizar un nodo local desde R se realiza con una simple instrucción.
    - **Rapidez:** la rapidez de procesamiento de los modelos implementados en H2O es un orden de magnitud mayor que en las alternativas evaluadas.
    - **Escalabilidad:** el uso de H2O permite entrenar los modelos utilizados con el conjunto de datos completos.
    - **Monitorización:** el estado del cluster H2O se puede monitorizar de forma sencilla desde los logs de ejecución o desde su interfaz web.

En la figura 12 se muestra el interfaz web de H2O.

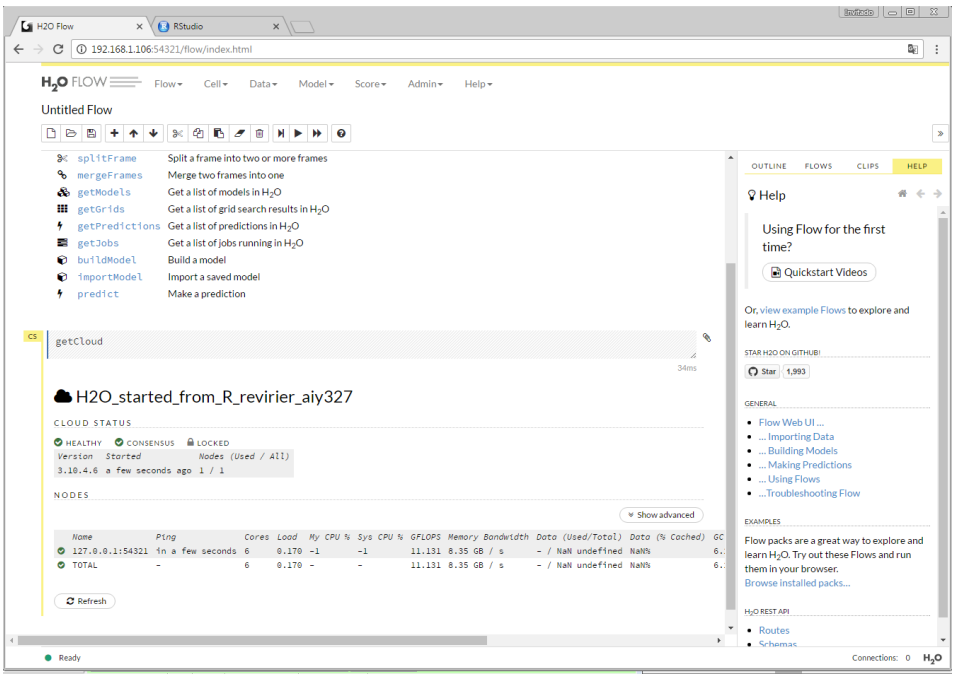

Figura 12: Interfaz web H2O

4 MIMIC-III

En este capítulo se describe la BBDD MIMIC-III utilizada para el desarrollo del estudio, introduciendo el contexto, estructura y características y las ventajas que presenta para este tipo de estudios.

4.1 MIMIC

Las dificultades para realizar estudios e investigaciones utilizando datos sanitarios reales son, por una parte, las relacionadas con la especial privacidad de los datos de salud, y por otra, la existencia de barreras organizativas y tecnológicas, consideraciones ambas que dificultan enormemente el acceso a los datos clínicos y su tratamiento y análisis.

Para evitar estas dificultades está disponible para la comunidad investigadora y educativa la base de datos MIMIC-III (*Medical Information Mart for Intensive Care*) [49, p.], accesible de forma abierta. Esta BBDD es una evolución de la BBDD MIMIC-II [50] creada por el *Laboratory of Computational Physicology* del MIT [51] con el objetivo de proporcionar herramientas para la creación de conocimiento clínico a través de la aplicación de técnicas de análisis de datos y aprendizaje automático.

La base de datos MIMIC-III [52] contiene datos de más de cuarenta mil pacientes y sesenta mil episodios de hospitalización de las unidades de cuidados intensivos del *Beth Israel Deaconess Medical Center* (Boston, Massachussets) registrados entre los años 2001 y 2012. La versión usada en este estudio es MIMIC-III v1.4 (2 septiembre 2016).

Como se resume en la figura 13, esta BBDD dispone de información de datos demográficos de pacientes, resultados de laboratorio, mediciones de signos vitales, anotaciones de evolución, informes de alta, prescripción, etc. Los datos identificativos de pacientes, personal clínico y fechas están anonimizados para cumplir con las leyes de protección de datos, aunque al contener información clínica real y por el especial cuidado con que se debe tratar este tipo de datos, para poder acceder y descargar la BBDD es necesario completar un proceso formal de registro, superar un curso online sobre las consideraciones éticas y legales de la investigación en especímenes humanos y firmar un acuerdo sobre uso adecuado de los datos, tal como se indica en la web de MIMIC-III [53].

En la misma web citada [52] se proporcionan las instrucciones para obtener la BBDD y cargarla en los gestores MySQL, Oracle y PostgreSQL [54], así como la posibilidad de obtener una máquina virtual que contenga la BBDD MIMIC-III y un servidor PostgreSQL preconfigurado, utilizando Vagrant, y que puede ser útil en algunos entornos por su facilidad de uso.

Los datos contenidos en la BBDD MIMIC-III han sido obtenidos, preprocesados y anonimizados a partir de los diferentes sistemas del hospital, por lo que es representativa de la heterogeneidad de los datos

sanitarios, aunque se ha realizado un trabajo de limpieza y eliminación de inconsistencias que facilita el análisis de la información.

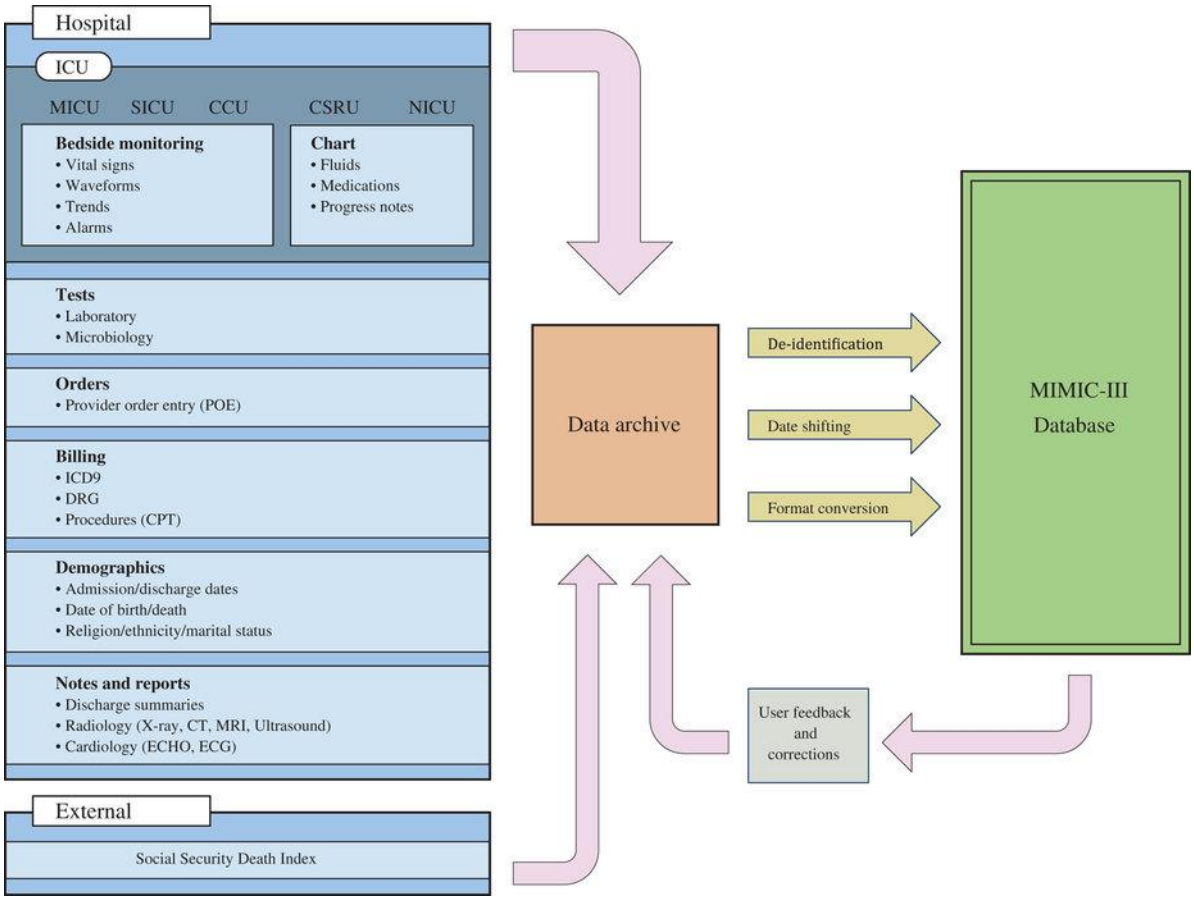

Figura 13 Modelo de construcción de MIMIC-III. Obtenido de [49]

MIMIC-III destaca por ser la única BBDD accesible de forma libre de su tipo [49], contiene datos de más de una década, lo que permite estudios a medio y largo plazo, y no existen restricciones al tipo de estudio a realizar. Gracias a esas facilidades se pueden encontrar múltiples publicaciones e investigaciones realizadas tomando como base los datos de MIMIC-II y III, como por ejemplo: el uso de modelos predictivos para obtener diagnósticos según el contexto [55], desarrollo de herramientas web para visualización de datos [56], análisis de diseño de cohortes [57], estudios de supervivencia según fluidos usados para resucitación [58], predicción de mortalidad usando el algoritmo Superlearner [59], predicción de sepsis usando modelos de aprendizaje automático [60], modelos predictivos para clasificación [61], desarrollo de modelos de scoring en pacientes de cirrosis [62], análisis del uso de herramientas como Hadoop/ Rapid Miner con datos clínicos [63], estudio de la influencia de los niveles de glucosa en la mortalidad [3], entre otros.

Entre las ventajas de MIMIC-III está la documentación existente, como por ejemplo las guías disponibles en el sitio web principal [52] y el libro *Secondary Analysis of Electronic Health Records* [64], disponible en libre descarga, que realiza un completo recorrido por las fases y técnicas de análisis de datos y aplicación de modelos tomando MIMIC-III como base.

Es importante destacar también la existencia de una comunidad activa de investigadores [65] que han implementado vistas SQL para obtener nuevos datos agregados, scores, etc.

Los tipos de datos que se encuentran disponibles en la BBDD MIMIC-III son principalmente los que se detallan en la tabla 1 (adaptada de https://www.nature.com/articles/sdata201635/tables/3)

TYPE	DESCRIPTION
Facturación	Datos codificados y destinados principalmente a facturación: GRD (grupos relacionados de diagnósticos), CIE9 [66, p. 9] [67]
Demográficos	Detalles del paciente, episodios de hospitalización en UCI, unidades de enfermería, fechas y horas de ingreso y alta, etc.
Diccionarios	Diccionarios de codificación CIE9, ítems de laboratorio, ítems de eventos, etc.
Intervenciones	Procedimientos realizados al paciente: diálisis, estudios RX, cirugías, etc.
Laboratorio	Datos de analíticas y estudios.
Medicación	Datos de prescripción y administración de medicación.
Anotaciones	Anotaciones en texto libre de evolución, cuidados de enfermería, resumen de alta, indicaciones médicas, etc.
Constantes	Datos de constantes vitales, pulsaciones, presión, temperatura, etc.
Informes	Documentos en texto libre o semiestructurado de informes de alta, evolución, electrocardiogramas, RX, imagen, etc.

Tabla 1: Tipos de datos disponibles en MIMIC-III.

La BBDD MIMIC-III se ha construido a partir de los datos de diferentes aplicaciones departamentales del hospital, de gestión de UCI y registros del *Social Security Administration Death Master File*. Una de las características importantes a tener en cuenta durante la exploración y uso de MIMIC-III es que los datos específicos de las UCIs se registraron mediante dos sistemas diferentes: Metavision y Carevue y se consolidaron ambos en MIMIC-III, aunque dadas sus diferentes capacidades y características, los diccionarios de datos usados son diferentes y algunos datos como la entrada de fluidos se mantienen en distintas tablas según el sistema de origen. Este punto es importante porque implica que casi todas las consultas a la BBDD MIMIC-III se deben duplicar y desarrollar con criterios diferentes según el sistema de registro utilizado.

Los únicos diccionarios estandarizados en MIMIC-III son los correspondientes a la codificación CIE9, los relacionados con la facturación (GRDs) y las pruebas de laboratorio, que disponen de un campo mapeado con LOINC [68], el resto de ítems usados no tienen una correspondencia estandarizada ni documentada, siendo muchos de ellos

resultado de introducción de texto libre. Este punto constituye una de las mayores dificultades del uso de MIMIC-III ya que obliga a un esfuerzo de búsqueda sobre miles de ítems no estandarizados para identificar las variables a extraer. Para facilitar esta tarea es importante analizar si las variables que se desean obtener de la BBDD MIMIC-III están ya entre las usadas en las vistas y las consultas disponibles en el repositorio de MIMIC-CODE [69]. Más adelante y en los anexos se detallarán las principales variables usadas en este informe y la forma de obtenerlas.

La técnica empleada al anonimizar los datos de la BBDD MIMIC-III también tiene importancia a la hora de desarrollar consultas, principalmente las fechas y determinadas anotaciones de texto libre [70]. En MIMIC-III se mantienen consistentes los intervalos de fechas y tiempo para cada episodio y paciente, desplazándose las fechas reales a un año entre 2100 y 2200. Para los pacientes de más de 89 años se elimina también la posibilidad de conocer la edad real.

4.2 Estructura de tablas y vistas

La estructura de MIMIC-III es relacional y contiene 26 tablas enlazadas, generalmente mediante los campos de identificador de paciente **SUBJECT_ID**, de episodio **UCI ICUSTAY_ID** y/o de episodio de hospitalización **HADM_ID**.

En la tabla 2 (adaptada de la documentación de MIMIC-III https://www.nature.com/articles/sdata201635/tables/4 y con datos propios) se describen brevemente las tablas más importantes desde el punto de vista de este estudio, con algunos datos de su dimensión.

TABLE/N ROWS	DESCRIPTION
ADMISSIONS 58976 registros 46420 pacientes únicos	Datos correspondientes a los episodios de hospitalización, diagnóstico de ingreso (no codificado) y datos administrativos (ADT). https://mimic.physionet.org/mimictables/admissions/
CHARTEVENTS 330712483 registros 46467 pacientes únicos	Tabla general de registro de eventos relacionados con el paciente y los episodios de UCI y hospitalización, recoge registros introducidos manualmente por enfermería, constantes vitales, pruebas de laboratorio y monitorización, se corresponde con la información mostrada en los monitores de cada paciente en la UCI. El diccionario de ítems asociado es **D_ITEMS**. https://mimic.physionet.org/mimictables/chartevents/
D_ICD_DIAGNOSES 14567 registros	Diccionario de diagnósticos codificados CIE9. https://mimic.physionet.org/mimictables/d_icd_diagnoses/
D_ICD_PROCEDURES 3882 registros	Diccionario de procedimientos codificados CIE9. https://mimic.physionet.org/mimictables/d_icd_procedures/

D_ITEMS 12487 registros **Agrupados por tipo:** 7212 chartevents 2938 inputevents_cv 1161 outputevents 436 microbiologyevents 422 inputevents_mv 193 datetimeevents 125 procedureevents_mv **Agrupados por origen:** 9059 carevue 2992 metavision 436 hospital	Diccionario de ítems utilizado en las tablas de eventos. Los eventos registrados por Carevue y Metavision están duplicados y con diferente código numérico. Los datos de entrada provenientes de Carevue se registran en texto libre, por lo que existen eventos duplicados también por esa causa. El campo **dbsource** indica el sistema de origen. https://mimic.physionet.org/mimictables/d_items/
D_LABITEMS 753 registros **Agrupados por categoría:** 410 Hematology 274 Chemistry 34 Blood Gas 19 CHEMISTRY 13 HEMATOLOGY 3 BLOOD GAS	Diccionario de ítems de laboratorio, contiene la correspondencia con la codificación LOINC. https://mimic.physionet.org/mimictables/d_labitems/
DIAGNOSES_ICD 651047 registros 58976 episodios de hospitalización codificados (100%)	Tabla de diagnósticos asociados a un episodio de hospitalización. Es necesario transformar los códigos para obtener los códigos CIE9 estándar. https://mimic.physionet.org/mimictables/diagnoses_icd/
DRGCODES 125557 registros 58890 registros de hospitalización codificados (99.85%)	Tabla con los GRD (Grupos Relacionados de Diagnósticos) correspondientes a los episodios de hospitalización. https://mimic.physionet.org/mimictables/drgcodes/
ICUSTAYS 61532 registros 57786 episodios de hospitalización **Agrupados por sistema:** 37776 carevue 23620 metavision 136 both	Datos de episodios de UCI. El campo **dbsource** indica el sistema de gestión de UCI utilizado. https://mimic.physionet.org/mimictables/icustays/
INPUTEVENTS_CV 17527935 registros 33799 episodios de UCI (54.93%)	Datos de eventos de entrada de fluidos (sueros, medicación intravenosa, bolos, insulina IV, etc…) para Carevue. Asociados al episodio de UCI. https://mimic.physionet.org/mimictables/inputevents_cv/

INPUTEVENTS_MV 3618991 registros 23386 episodios de UCI (38.01%)	Datos de eventos de entrada de fluidos (sueros, medicación intravenosa, bolos, insulina IV, etc…) para Metavision. Asociados al episodio de UCI. https://mimic.physionet.org/mimictables/inputevents_mv/
LABEVENTS 27854055 registros 58151 episodios de hospitalización (98.60%)	Resultados de laboratorio asociados al episodio de hospitalización, para asociarlos al de UCI es necesario hacerlo por fecha y hora. https://mimic.physionet.org/mimictables/labevents/
NOTEEVENTS 2083180 registros 58361 episodios de hospitalización (98.96%) **Agrupados por categoría** 822497 Nursing/other 522279 Radiology 223556 Nursing 209051 ECG 141624 Physician 59652 Discharge summary 45794 Echo 31739 Respiratory 9418 Nutrition 8301 General 5431 Rehab Services 2670 Social Work 967 Case Management 103 Pharmacy 98 Consult	Registro de notas en texto libre o semiestructurado: notas de evolución, informes RX, ecografías, ECG, informes de alta, etc, asociados al episodio de hospitalización. https://mimic.physionet.org/mimictables/noteevents/
PATIENTS 46520 registros 9974 éxitus en hospital (21.44%) **Agrupados por género** 20399 F 26121 M	Tabla de datos de pacientes e información de éxitus. https://mimic.physionet.org/mimictables/patients/
PRESCRIPTIONS 4156450 registros 52151 episodios UCI (84.75%) 50216 episodios de Hospitalización (85.15%) **Agrupadas por tipo** 3216882 MAIN 925089 BASE 14479 ADDITIVE	Datos de prescripción de medicamentos. No existe en MIMIC-III relación entre esta tabla y una orden médica o un evento de administración de medicamentos. Los datos están asociados al episodio de UCI o al de hospitalización. Tampoco se indica si la orden se cancela con posterioridad. https://mimic.physionet.org/mimictables/prescriptions/

PROCEDURES_ICD 240095 registros 52243 episodios de hospitalización codificados (88.58%)	Tabla de procedimientos CIE9 asociados a un episodio de hospitalización. Es necesario transformar los códigos para obtener los códigos CIE9 estándar. https://mimic.physionet.org/mimictables/procedures_icd/
VENTDURATIONS 51786 registros 27912 episodios de UCI (45.36%)	Tabla con los registros de duración de ventilación mecánica asociados al episodio de UCI.

Tabla 2: Tablas de MIMIC-III utilizadas.

Los datos de entrada/salida de fluidos son los más complejos de tratar y que presentan más problemas de consistencia y diferente tratamiento según que el sistema origen sea Metavision o Carevue, tal como se indica en la propia documentación: https://mimic.physionet.org/mimicdata/io/. En este estudio se han utilizado algunas variables contenidas en dichas tablas, procesando los datos y consolidando los datos de Carevue y Metavision mediante los procesos ETL.

Uno de los puntos fuertes de MIMIC-III es, como se ha indicado, la existencia de una comunidad investigadora activa que desarrolla vistas SQL para obtener datos adicionales de la BBDD MIMIC-III. En este estudio se han utilizado varias de ellas directamente y otras para obtener información útil en la selección de variables realizada.

Estas vistas están disponibles en el repositorio MIMIC-CODE [69] https://github.com/MIT-LCP/mimic-code/tree/master/concepts son necesarias para el proceso ETL desarrollado en Pentaho. Se trata de vistas materializadas que se deben ejecutar o instalar previamente a la ejecución de la ETL.

En la tabla 3 se indican las vistas utilizadas o revisadas en el presente estudio y los tipos de datos que contienen.

TIPO	DESCRIPCION / VISTA
comorbidity	Vistas de comorbilidad, son equivalentes a las comorbilidades obtenidas mediante los procesos ETL. No se han utilizado en este estudio. https://github.com/MIT-LCP/mimic-code/tree/master/concepts/comorbidity
firstday	Vistas de datos del primer día: height-first-day.sql (altura) / weight-first-day.sql (peso) / vitals-first-day.sql (constants vitals) / rrt-firs-day.sql (ventilación mecánica) Necesarias para la carga de scores: echo-data.sql / ventilation-durations.sql / urine-output-first-day.sql / ventilated-first-day.sql / gcs-first-day.sql / labs-fist-day.sql / blood-gas-first-day.sql / blood-gas-first-day-arterial.sql https://github.com/MIT-LCP/mimic-code/tree/master/concepts/firstday
sepsis	Datos de sepsis

	angus.sql https://github.com/MIT-LCP/mimic-code/tree/master/concepts/sepsis
scores	Cálculo de scores de severidad: apsiii.sql / lods.sql / mlods.sql / oasis.sql / qsofa.sql / saps.sql / sapsii.sql / sirs.sql / sofá.sql Obtienen los scores APSIII [71], LODS [72], MLODS [72], OASIS [73], QSOFA [74], SAPS [75], SAPSII [76], SOFA [28], [77], SIRS [78] https://github.com/MIT-LCP/mimic-code/tree/master/concepts/severityscores
vassopresor-durations	Cálculo de duración de diferentes vasopresores: adenosine-durations.sql / dobutamine-durations.sql / dopamine-durations.sql / epinephrine-durations.sql / isuprel-durations.sql / milrinone-durations.sql / norepinephrine-durations.sql / phenylephrine-durations.sql / vasopressin-durations.sql / vasopressor-durations.sql https://github.com/MIT-LCP/mimic-code/tree/master/concepts/vasopressor-durations
concepts	Datos de respiración mecánica: ventilator-durations.sql Diálisis: rrt.sql https://github.com/MIT-LCP/mimic-code/tree/master/concepts

Tabla 3: Vistas MIMIC-III utilizadas

Todos los datos que se pueden obtener de las vistas se indican en el anexo correspondiente.

4.3 Criterios de obtención variables

Los datos para realizar el análisis se han obtenido mediante los procesos ETL detallados más adelante y documentados en los anexos. Una vez definidos los tipos de variables propuestos para realizar el estudio, se realizó un análisis para identificar qué datos era posible obtener de las tablas de la BBDD MIMIC-III, cuáles de las vistas, de una transformación sencilla combinando campos o mediante lógica implementada en las consultas SQL o en los procesos ETL y cuales por su naturaleza necesitaban de una labor previa más intensa de identificación y pruebas.

En este apartado indicaremos los tipos de datos incluidos en el estudio que resultaron de obtención más compleja. Se ha considerado importante mencionarlos con el objetivo de facilitar posteriores análisis sobre la BBDD MIMIC-III. Los detalles se desarrollan en los anexos.

- **Alergia insulina:** los datos de alergias se encuentran de forma semiestructurada en los informes de alta.
- **Administración insulina IV:** los datos de administración de insulina IV se encuentran en tablas diferentes según que el sistema de gestión empleado para su registro fuese Metavision o Carevue.
- **Administración nutrición:** se obtienen de la tabla **chartevents.**

- **Administración esteroides:** se obtienen por búsqueda de nombre de droga en la tabla de prescripciones, el que esté prescrito no significa que luego se administre.
- **Prescripción insulina:** se realizó una búsqueda en el *National Drug Code Directory* de la *US Food and Drug Administration* [79] para identificar los diferentes nombres comerciales y genéricos bajo los que se podía encontrar la insulina en la tabla de prescripciones.
- **Administración de glucosa:** Se obtiene de **chartevents.**
- **Comorbilidades:** se han obtenido a partir de los datos del estudio de Elixhauser, A. [1], existen vistas equivalentes en MIMIC-III.
- **Analíticas:** los datos de los identificadores de los ítems se obtuvieron a partir de las vistas existentes para MIMIC-III (*Baseexcess, Hemoglobin, Lactate, PCO2, PH, PO2, Bicarbonate, Creatinine, Whiteblood*).
- **Diabetes:** se utilizan los códigos CIE9.
- **Clasificación de Servicios:** los servicios se clasifican en MEDICAL o SURGICAL según la información de la descripción del servicio proporcionada en la web de MIMIC-III [80].
- **Sepsis/Infección:** se obtienen de una vista de MIMIC-III que utiliza la codificación CIE9 para obtener los datos de infección, disfunción de órganos, sepsis explícita y ventilación mecánica y obtener a partir de ellos el indicador angus [81].
- **Vasopresores:** se obtienen de una vista de MIMIC-III. Solo se consideran la administración de vasopresores con una duración de más de 0.5 H.
- **Diálisis:** se obtiene de las vistas disponibles para MIMIC-III (rrt).

5 ETL

En este apartado se introduce el uso de una herramienta ETL, se describe de una forma general la aplicación seleccionada y su funcionamiento, así como la estructura de los procesos desarrollados. El detalle de dichos procesos se documenta en los anexos. Dentro de la metodología empleada, el uso de una ETL es una pieza fundamental que permite obtener los datos de la BBDD de origen, realizar parte del preprocesado, agrupar y consolidar los datos y es fácilmente adaptable a otras BBDD de origen modificando las consultas de entrada.

5.1 Contexto

Como primer paso antes de realizar cualquier tarea de análisis de datos es necesario obtener los datos en bruto (*raw data*) y procesarlos para obtener datos consistentes que se puedan utilizar en un análisis estadístico [82]. Generalmente se considera que el 80% del tiempo dedicado a un análisis de datos se emplea en el proceso de limpiar y preparar los datos [83] [84]. En la figura 14 se indican los pasos de un análisis de datos (Elaboración propia a partir de [82] pag. 7).

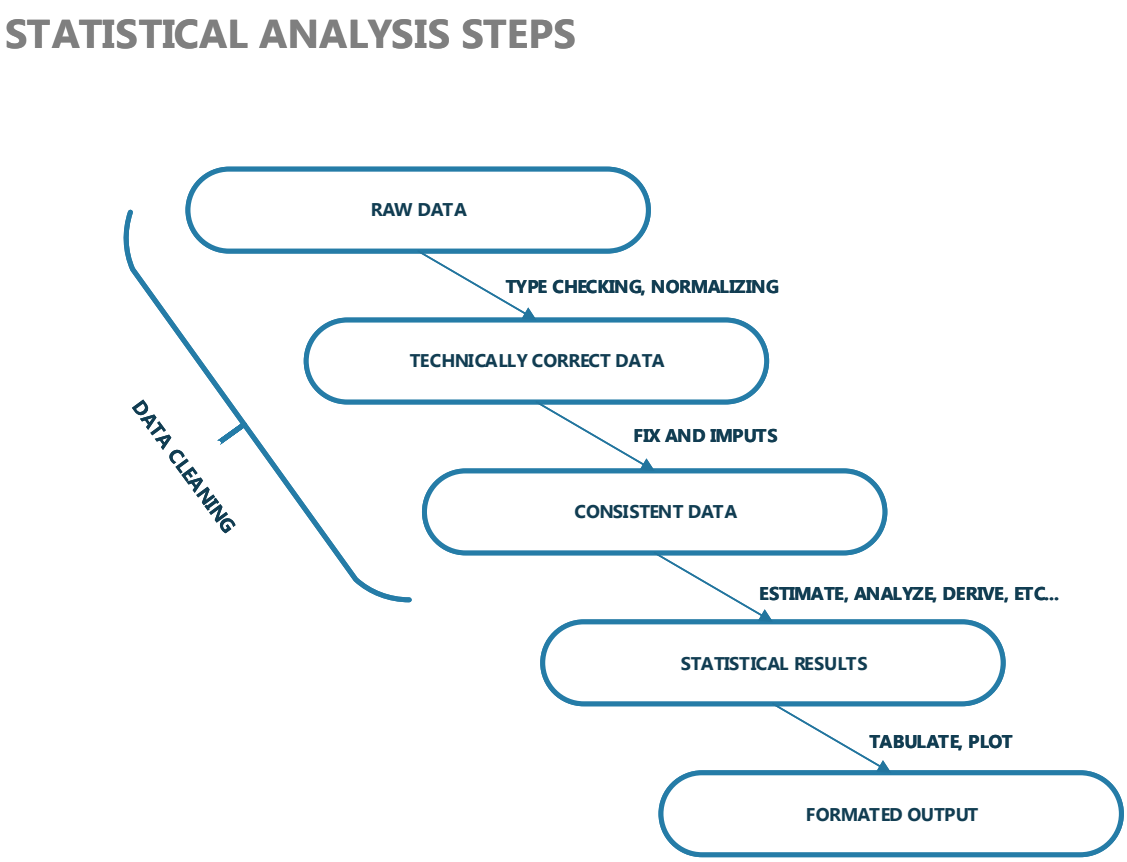

Figura 14: Pasos de un análisis estadístico.

En este estudio la fuente de datos es la BBDD MIMIC-III, que los almacena en una estructura relacional que es necesario convertir a una estructura

apropiada para el análisis estadístico o *tidy* [84], es decir, transformar la estructura de múltiples tablas con distinto número de filas y columnas enlazadas mediante claves a una estructura donde cada variable es una columna independiente, cada observación forma una fila y cada tipo de observación diferente constituye una tabla.

Existen varias alternativas técnicas para realizar este procesado, la mayor parte emplean las capacidades del lenguaje R para obtener, transformar y filtrar los conjuntos de datos y convertirlos en datos *tidy*. Teniendo en cuenta que los datos de origen en este caso se obtienen de una BBDD SQL, también era posible obtener los datos desde una sesión R directamente mediante consultas SQL.

En este estudio se ha considerado más eficiente utilizar una herramienta ETL para realizar la mayor parte de la obtención y preprocesado de los datos, por varios motivos, principalmente:

- Las variables utilizadas en el análisis se han ido perfilando durante el propio estudio. Es más sencillo obtener un conjunto de variables mediante una ETL y con posterioridad seleccionar/tratar las más interesantes que mantener una serie de consultas SQL e ir modificándolas en función de necesidades cambiantes.
- Desarrollar y mantener consultas SQL para obtener cientos de variables es una tarea compleja y con riesgo de errores.
- Al utilizar una herramienta ETL la carga de proceso se reparte entre el servidor de BBDD y el de ETL, las consultas a la BBDD son más sencillas y rápidas y el proceso computacionalmente más intensivo se realiza en memoria en el servidor ETL. El resultado final es un proceso global mucho más rápido.
- Una vez desarrollados los procesos ETL, son sencillos de mantener.
- Durante la implementación de los procesos ETL las capacidades de visualización y depuración en tiempo real de la ETL simplifican la identificación y corrección de problemas y errores.
- El uso de una herramienta ETL permite detectar cuellos de botella en los procesos y paralelizar de forma selectiva donde sea necesario para optimizar la velocidad del flujo de datos.

5.2 Arquitectura ETL

Mediante los procesos ETL, los datos se obtienen de la base de datos MIMIC-III y hojas Excel (los criterios para el cálculo de las comorbilidades), se procesan y se guardan en otra base de datos diferente, MIMICSEL, que es la utilizada para el análisis posterior.

Simplificando, la arquitectura ETL implementada en Pentaho PDI se muestra en la figura 15.

ETL

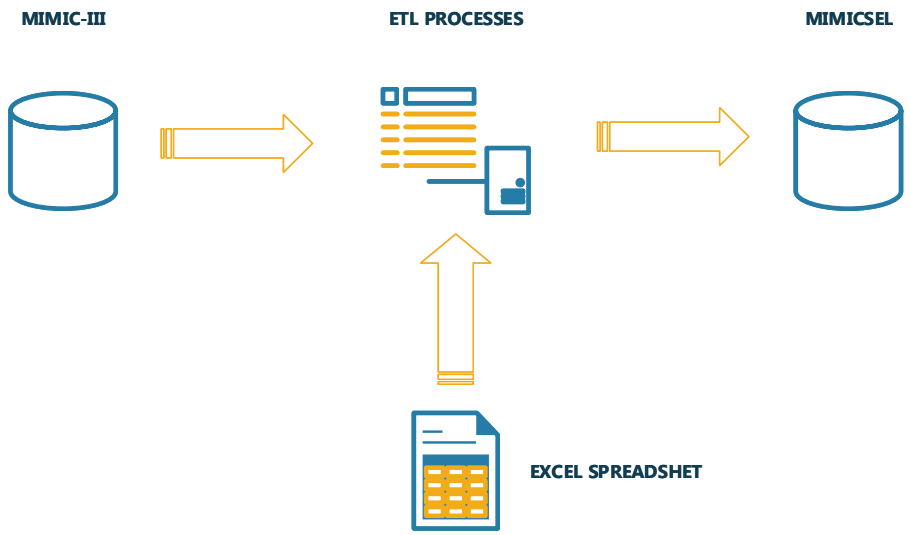

Figura 15: Arquitectura ETL

El diseño de los procesos ETL se ha realizado de forma modular, como resume la figura 16. En una primera parte de los procesos se obtienen los datos individuales, se transforman, normalizan y se guardan en la estructura de tablas intermedias. En ese punto, se llevan a cabo las tareas de agregación de datos: cálculo de medias, mínimos, máximos, porcentajes, etc.., por episodio y por tramos de 24H y se guardan en tablas de agregados. Por último, los procesos de selección de datos obtienen en una sola tabla todas las variables necesarias para realizar los análisis.

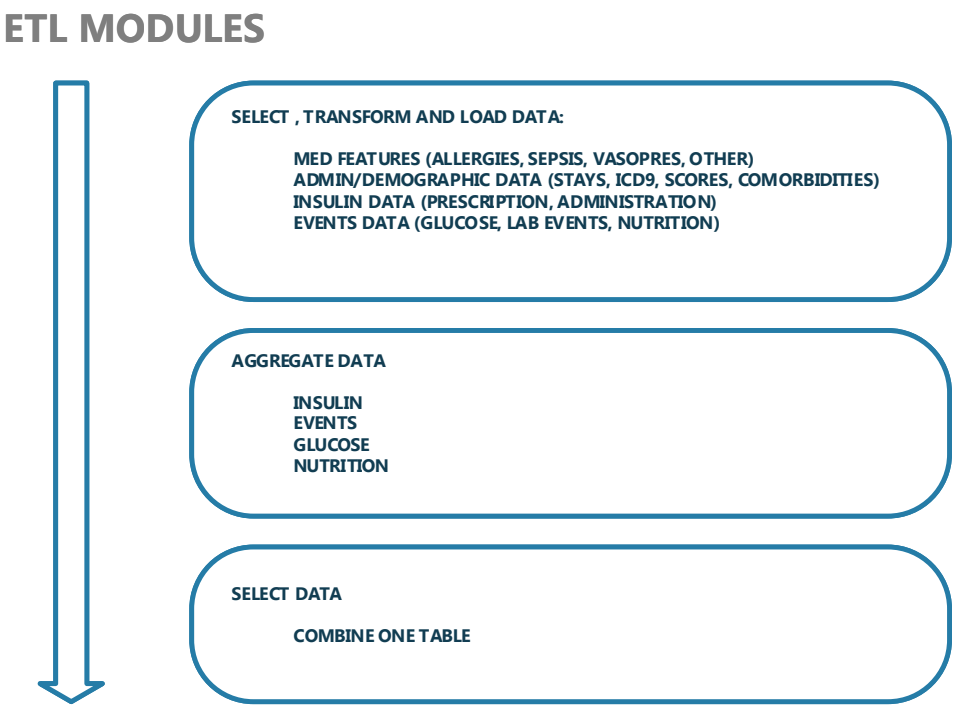

Figura 16: Estructura modular de la ETL

Los procesos ETL crean automáticamente las tablas necesarias en la base de datos intermedia MIMICSEL y utilizan tablas temporales en memoria para almacenar resultados parciales, con lo que se obtiene una gran rapidez de proceso, unos 12 minutos en procesar millones de registros.

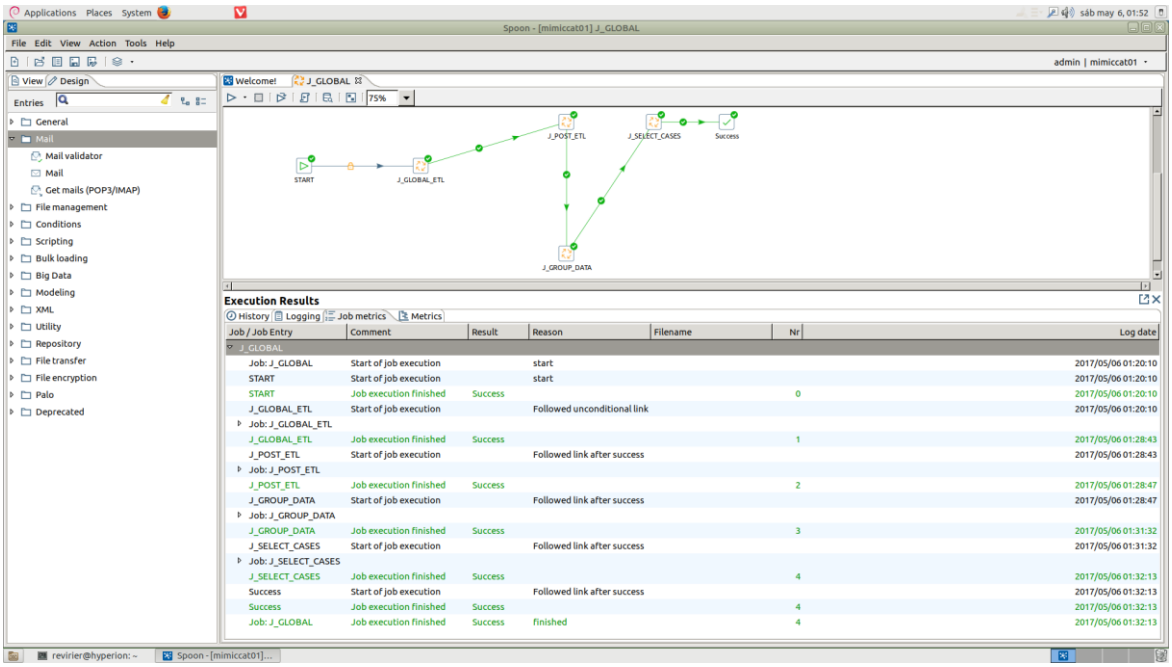

Figura 17: Pantalla d desarrollo de Pentaho PDI.

El diseño de los procesos ETL en Pentaho se realiza en el entorno visual de la herramienta (figura 17), utilizando los distintos componentes: pasos (*steps*), transformaciones (*transformations*) y trabajos (*jobs*) para organizar los procesos, como se describe a continuación:

5.2.1 Pasos:

Los pasos son los componentes básicos que realizan el procesamiento de los datos, se conectan entre sí de forma que la/s salidas de un paso constituyen la/s entrada/s de otro/s. Los pasos son hilos java independientes y asíncronos, una vez que se ejecuta un proceso de Pentaho PDI, permanecen activos, procesan los datos que les llegan en función de su configuración y tipo y los envían a otros pasos. El flujo de datos entre pasos está compuesto por campos y el procesado de los datos entrantes por cada paso puede dar lugar a campos nuevos que se añadan al flujo, eliminación de campos, operaciones de agrupamiento, etc. Pentaho PDI dispone de múltiples tipos de pasos predefinidos para realizar operaciones de lectura desde BBDD, desde ficheros, llamadas a *web services*, escritura en BBDD, a ficheros, unión de flujos de datos, operaciones de agrupamiento, cálculos estadísticos y matemáticos, consultas SQL, etc. Existen pasos programables en Java y Javascript para realizar operaciones que no estén dentro de las opciones de pasos disponibles o en las que sea más sencillo programar que utilizar un paso predefinido. En la figura 18 se muestran algunos de los pasos existentes.

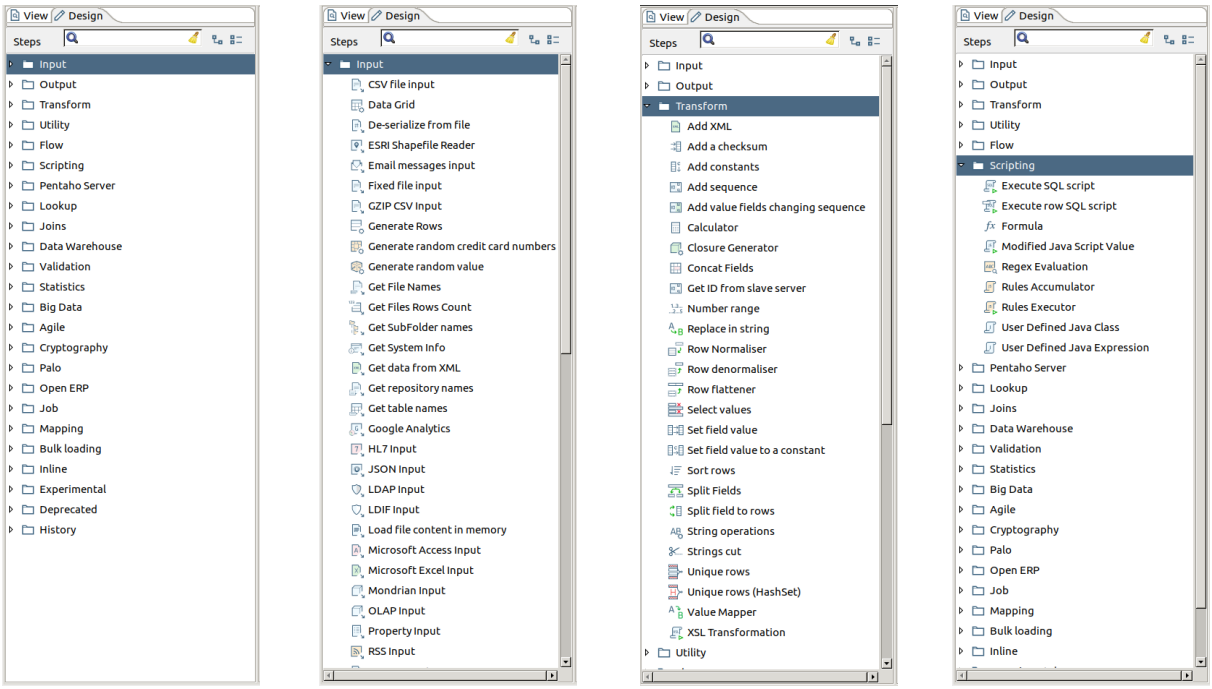

Figura 18: Ejemplos de pasos disponibles en las transformaciones de Pentaho PDI.

5.2.2 Transformaciones:

Los pasos se organizan y estructuran en transformaciones. Cada transformación tiene uno o varios pasos de entrada de datos, de proceso de datos y de salida a otras transformaciones, bases de datos o ficheros. En las transformaciones se indica el sentido en que los datos se transfieren, bifurcan o unen mediante conectores que representan el flujo de datos. En la figura 19 se muestra un ejemplo de transformación con cuatro pasos, que obtienen datos de una tabla, seleccionan las filas únicas, añaden un campo de secuencia y guardan el resultado en otra tabla.

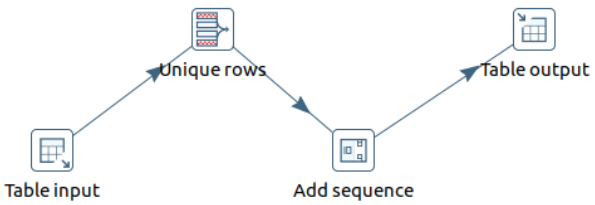

Figura 19: Ejemplo de transformación en Pentaho PDI

5.2.3 Trabajos:

Las diferentes transformaciones se organizan en trabajos, que realizan la orquestación de las diferentes tareas de procesamiento de datos. Los trabajos pueden contener secuencias de transformaciones, pero también pueden utilizar sus propios componentes para realizar control de flujo,

scripts, comprobar condiciones, ejecutar otros procesos, etc. En la figura 20 se muestran algunos de los componentes disponibles para los trabajos.

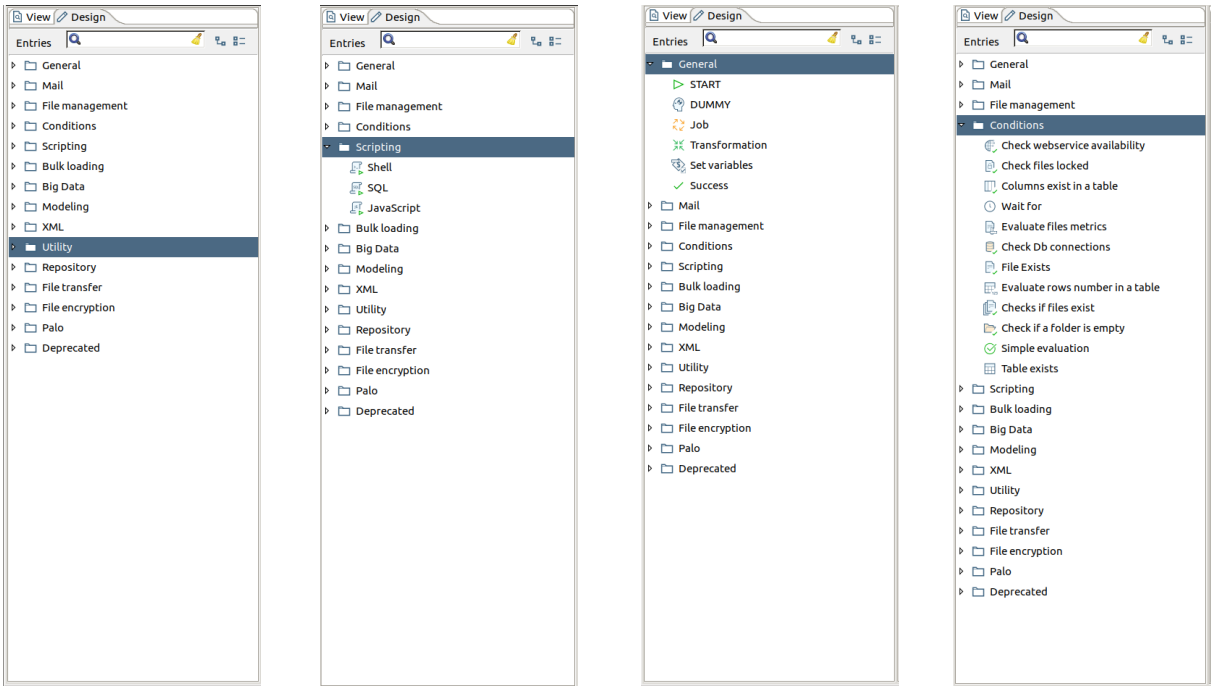

Figura 20: Ejemplo de pasos disponibles en los trabajos de Pentaho PDI.

Los trabajos pueden contener también otros trabajos, permitiendo así varios niveles en el diseño de los procesos ETL. La figura 21 es un ejemplo de trabajo sencillo que contiene otro trabajo y una transformación.

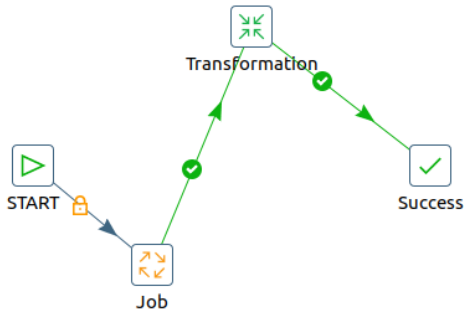

Figura 21: Ejemplo de trabajo en Pentaho PDI.

En las siguientes secciones se describe de manera general los diferentes módulos del proceso, sus entradas y sus salidas. La estructura de tablas creada y los diferentes trabajos y transformaciones se detallan en los anexos.

5.3 Proceso ETL desarrollado

El proceso ETL en Pentaho PDI se ha desarrollado de forma modular, separando las diversas tareas en trabajos y transformaciones que se

encadenan para construir el proceso total, formado por 17 trabajos y 23 transformaciones.

El proceso ETL se lanza desde un trabajo inicial **J_GLOBAL** que aparece en la figura 22 y que incluye la secuencia de los trabajos principales del proceso: **J_GLOBAL_ETL**, **J_POST_ETL**, **J_GROUP_DATA** y **J_SELECT_CASES**.

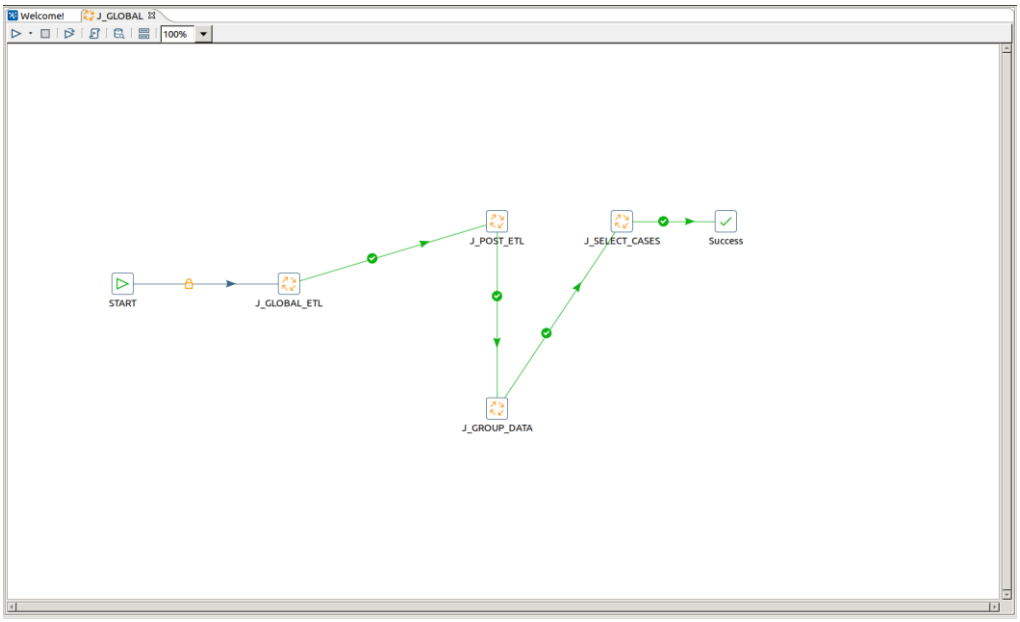

Figura 22: Trabajo principal de la ETL desarrollada.

- **J_GLOBAL_ETL** (figura 23)**:** procesos de ETL que obtienen los datos de los indicadores clínicos, los datos administrativos, los referidos a la prescripción y administración de insulina y los de mediciones de laboratorio y eventos de la BBDD MIMIC-III y los transforman y cargan en la BBDD intermedia MIMICSEL.

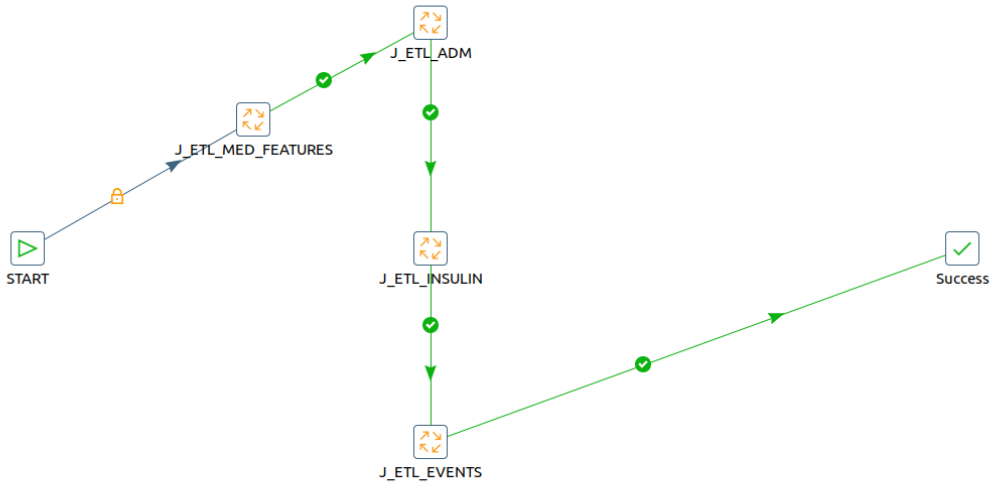

Figura 23: Trabajo J_GLOBAL_ETL.

- **J_POST_ETL:** realiza transformaciones que es más eficiente realizar una vez obtenidas las tablas intermedias de la BBDD MIMICSEL.
- **J_GROUP_DATA** (figura 24)**:** Realiza la agregación de datos necesaria para el análisis, por ejemplo, el cálculo de medias, máximos, mínimos, desviaciones estándar, para los diferentes indicadores, por tramo de 24H/48H/72H, etc. En este trabajo se realiza el procesado de los campos que se agruparan por cada episodio. El tratamiento de los datos de nutrición, insulina y glucosa se realiza de forma separada al resto de eventos.

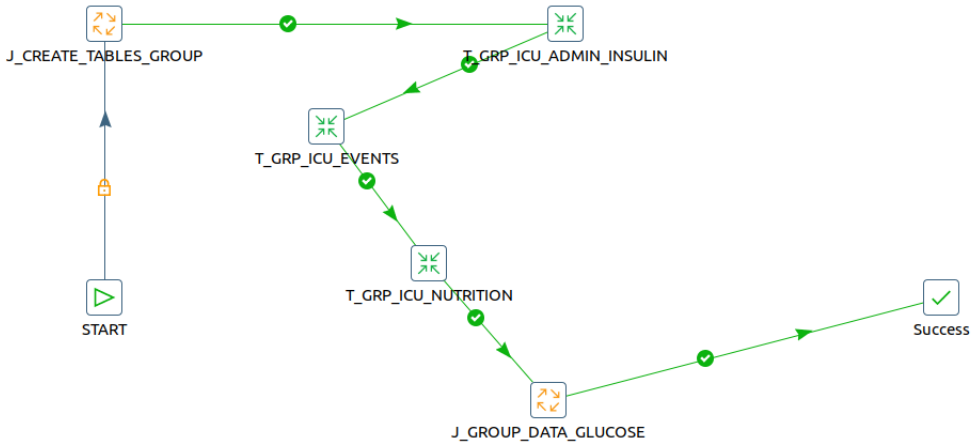

Figura 24: Trabajo J_GROUP_DATA.

- **J_SELECT_CASES** (figura 25)**:** extraen de las tablas de MIMICSEL el conjunto de datos necesario para el análisis, creando una tabla con los criterios de filtrado de población para su tratamiento desde R.

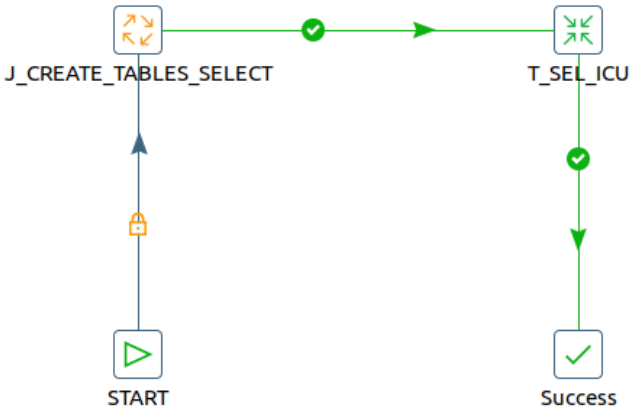

Figura 25: Trabajo J_SELECT_CASES.

El mapa completo de los trabajos y transformaciones desarrolladas se muestra en la figura 26.

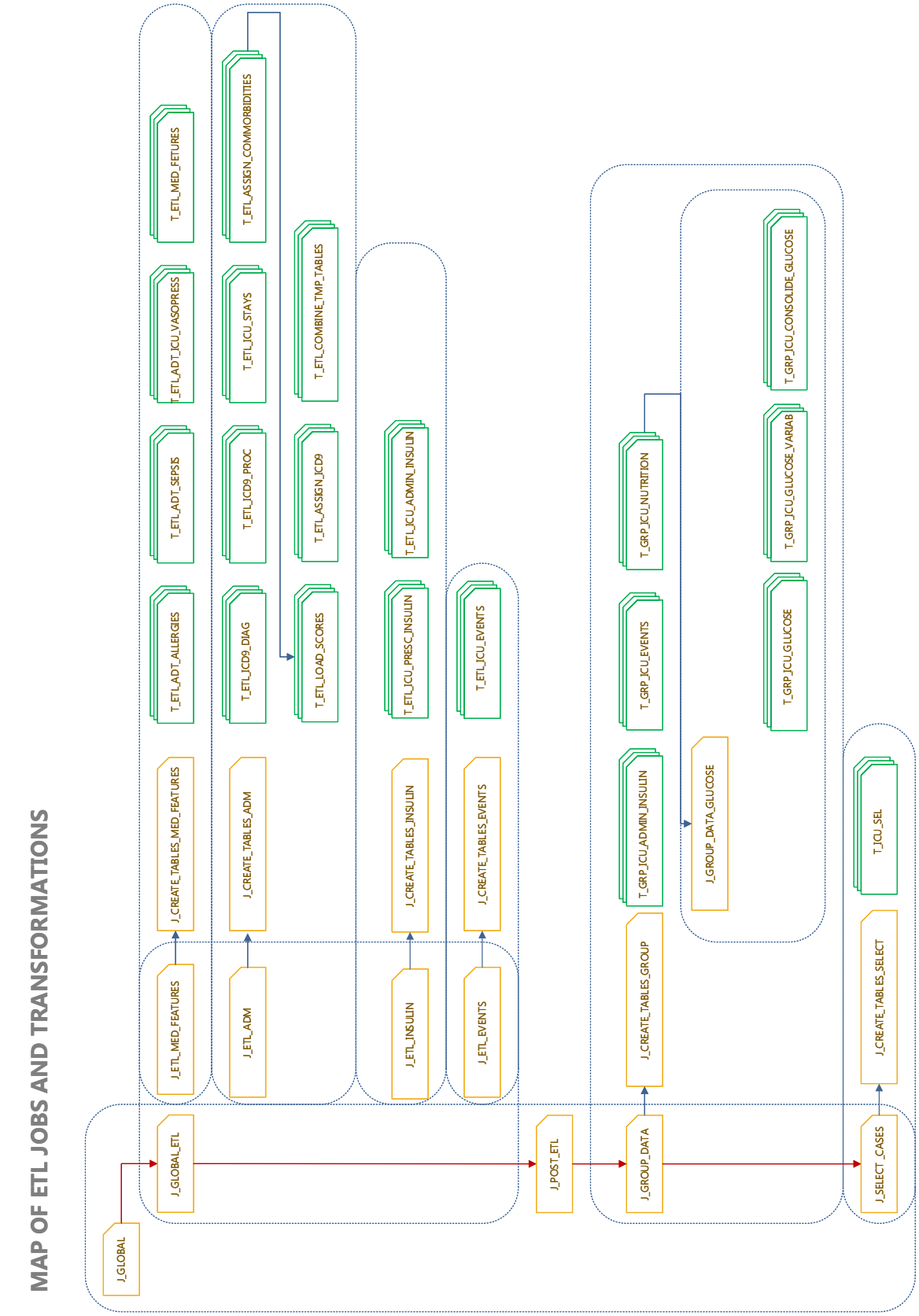

Figura 26: Mapa completo de los trabajos y transformaciones ETL desarrolladas en Pentaho PDI.

5.4 Base de datos intermedia MIMICSEL

Como resultado de la ejecución de los procesos ETL se construye una base de datos intermedia MIMICSEL con una serie de tablas de detalle y agregados que contienen los datos a partir de los cuales se realiza el análisis. Las tablas obtenidas se pueden dividir en tres clases: maestras de codificación, tablas de detalle y tablas de agregados.

- Maestras de codificación CIE9: utilizadas para disponer de la codificación CIE9 con su formato estándar, diferente del utilizado en la BBDD MIMIC-III.
- Tablas de detalle **DP_<nombre_tabla>** que contienen la información seleccionada, transformada y estructurada a partir de los datos de las tablas de la BBDD MIMIC-III
- Tablas de agregados **DG_<nombre_tabla>** que contienen información agregada por episodio UCI y tramos horarios obtenida a partir de los datos de las tablas **DP_<nombre_tabla>** anteriores.
- Tablas de selección **DS_<nombre_tabla>** que consolidan en una sola tabla las observaciones y variables para el análisis posterior.

Este proceso se muestra en la figura 27.

Los procesos ETL obtienen los datos necesarios y realizan un preproceso para obtener las tablas **DP_XXXX**. A partir de estas tablas se obtienen las tablas de datos agregados **DG_XXXX** y a partir de estas se realiza la selección de casos en **DS_XXXX**.

VENTAJA: Desacoplamiento / Fácil mantenimiento | **INCONVENIENTE:** Más laborioso inicialmente

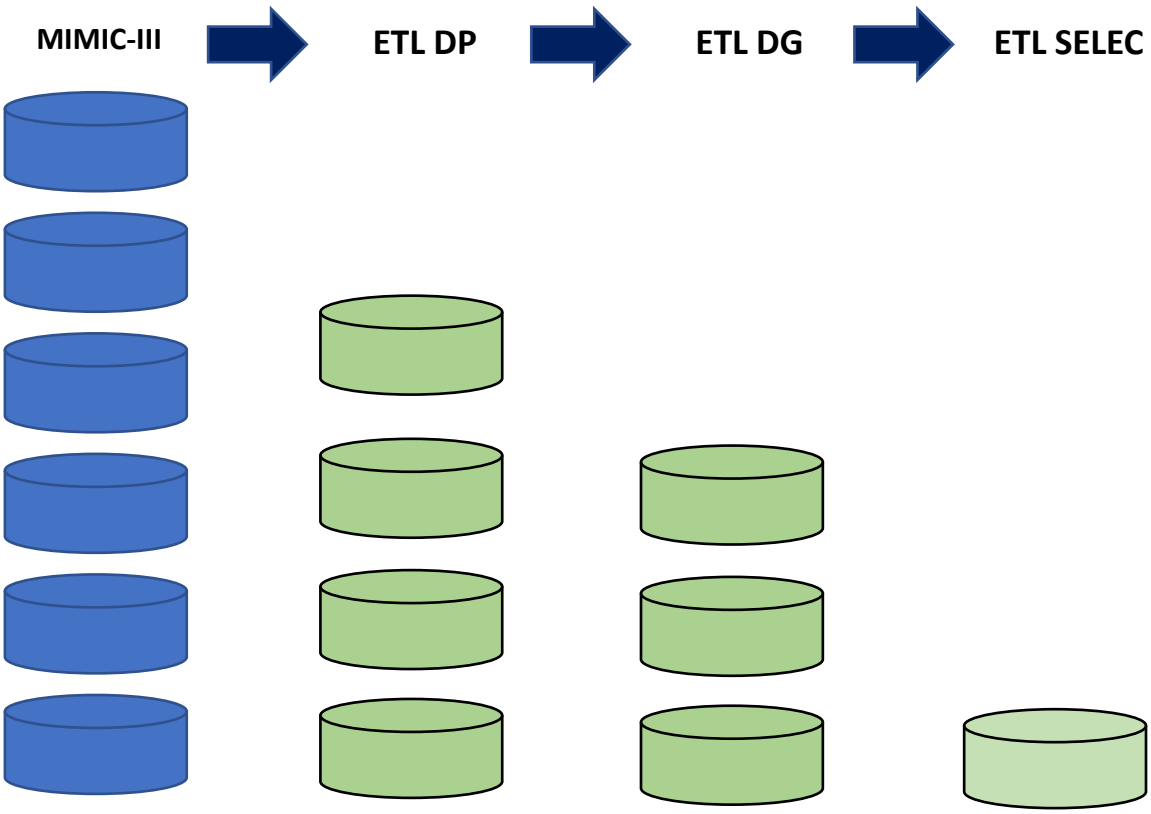

Figura 27: funcionalidad de las tablas obtenidas mediante los procesos ETL.

Las tablas de detalle, agregados y selección obtenidas en la BBDD MIMICSEL se detallan en las tablas 4, 5 y 6.

TABLE	DESCRIPTION
DP_ICU_STAYS	Tabla principal con los datos de episodios de ICU. Se obtiene a partir de las tablas de MIMIC-III **icustays**, **admissions**, **patients**, **diagnoses_icd**, **drgcodes**, **procedures_icd**
DP_ICD9_DIAG	Tabla auxiliar diagnósticos CIE9. Se obtiene a partir de las tablas de MIMIC-III **d_icd_diagnoses**, **diagnoses_icd**
DP_ICD9_PROC	Tabla auxiliar procedimientos CIE9. Se obtiene a partir de las tablas de MIMIC-III **d_icd_procedures, procedures_icd**
DP_PRESC_INSULIN	Tabla con información de las prescripciones de insulina para episodios ICU. Se obtiene a partir de las tablas de MIMIC-III **prescriptions, icustays**
DP_ADMIN_INSULIN	Tabla con información de las administraciones de insulina para episodios ICU. Se obtiene a partir de las tablas de MIMIC-III **inputevents_cv, inputevents_mv, icustays, admissions, d_items**
DP_ICU_EVENTS	Tabla con información de eventos de insulina, glucosa y nutrición para episodios ICU. Se obtiene a partir de las tablas de MIMIC-III **chartevents, d_items, icustays, admissions**

Tabla 4: Tablas de detalle obtenidas mediante la ETL.

TABLE	DESCRIPTION
DG_ADMIN_INSULIN	Datos de número de administraciones de insulina IV y cantidad, agrupados por tramos de 24H desde ingreso ICU
DG_ADMIN_INSULIN_AGG	Datos de número de administraciones de insulina IV y cantidad para los tramos de 24/48/72H desde ingreso ICU, agrupados por ingreso ICU
DG_ICU_EVENTS	Datos de número de medidas de laboratorio (Glucose / PH / PO2 / Whiteblood / Lactate / Hemoglobin / Baseexcess / PCO2 / Bicarbonate / Creatinine) agrupados por tramos de 24H desde ingreso ICU.
DG_ICU_EVENTS_AGG	Datos de número de medidas de laboratorio (Glucose / PH / PO2 / Whiteblood / Lactate / Hemoglobin / Baseexcess / PCO2 / Bicarbonate / Creatinine) para los tramos de 24/48/72H desde ingreso ICU, agrupados por ingreso ICU.
DG_ICU_GLUCOSE_DETAIL	Datos de control de glucosa agrupados por tramos de 24H desde ingreso ICU.
DG_ICU_GLUCOSE_AGG	Datos de control de glucosa para los tramos de 24/48/72H desde ingreso ICU, agrupados por ingreso ICU.
DG_ICU_NUTRITION	Datos de tipo de nutrición (TPN / NPO / PPN / TFE / OTH) por episodio UCI y por tramo horario de 24H desde ingreso.
DG_ICU_NUTRITION_AGG	Datos de tipo de nutrición (TPN / NPO / PPN / TFE / OTH) para los tramos de 24/48/72H desde ingreso ICU, agrupados por ingreso ICU.

Tabla 5: Tablas de agregados obtenidas mediante la ETL.

TABLE	DESCRIPTION
DS_ICU_STAYS_1	Tabla con los datos del resto de tablas de la BBDD MIMICSEL: datos administrativos de pacientes, estancia de hospitalización, estancia en UCI, mediciones primer día, codificación, comorbilidades, Scores, datos clínicos, prescripción insulina, administración insulina, nutrición, laboratorio, glucosa. Se filtra por episodios UCI con un diagnóstico codificado al menos, paciente adulto, con 3 medidas de glucosa en cada uno de los dos primeros tramos de 24H y con estancia de al menos dos días en la UCI.

Tabla 6: Tabla de selección obtenida mediante la ETL.

En la tabla 7 se proporciona información sobre el número de registros de cada una de las tablas obtenidas en la BBDD MIMICSEL:

TABLE	N ROWS	TABLE	N ROWS
DP_ICU_STAYS	121 columnas 61346 episodios UCI 46419 pacientes distintos	**DP_ICD9_DIAG**	5 columnas 14710 registros
DP_ICD9_PROC	5 columnas 3898 registros	**DP_PRESC_INSULIN**	6 columnas 26807 registros
DP_ICU_EVENTS	12 columnas 5448255 registros 59.913 episodios UCI Por tipo de evento: 1277860 GLUCOSE 927320 DIET 500584 PH 465769 PO2 465755 PCO2 409203 HEMOGLOBIN 366054 BICARBONATE 358181 CREATININE 331257 WHITEBLOOD 184774 BASEEXCESS 161498 LACTATE	**DP_ADMIN_INSULIN**	12 columnas 398569 registros 10491 episodios UCI
DG_ADMIN_INSULIN	5 columnas 29953 registros	**DG_ADMIN_INSULIN_AGG**	10 columnas 9513 registros
DG_ICU_EVENTS	92 columnas 258052 registros	**DG_ICU_EVENTS_AGG**	270 columnas 54598 registros
DG_ICU_GLUCOSE_DETAIL	22 columnas 227207 registros	**DG_ICU_GLUCOSE_AGG**	77 columnas 54598 registros
DG_ICU_NUTRITION	7 columnas 163576 registros	**DG_ICU_NUTRITION_AGG**	31 columnas 42992 registros
DS_ICU_STAYS_1	508 columnas 20445 registros		

Tabla 7: Dimensión y registros de las tablas de MIMICSEL.

6 MODELOS PREDICTIVOS

Una parte fundamental del estudio realizado ha consistido en la evaluación y prueba de diferentes modelos predictivos, realizando una primera selección entre los disponibles para clasificación binaria y pruebas con conjuntos parciales o completos de los datos objeto de análisis. Los criterios principales que han guiado la decisión de la lista inicial de modelos a evaluar han sido los siguientes:

- Modelos de **tipo clasificación en dos clases**. El objetivo a predecir en este estudio es de tipo binario, la mortalidad en 28 días.
- Disponibilidad del modelo para uso en **R.** En la práctica este criterio no elimina modelos dada la ingente cantidad existente en R, uno de los lenguajes de programación más utilizados para este tipo de análisis.[85]
- Disponibilidad del modelo en **diferentes *frameworks*** de R para facilitar las tareas de preprocesado, ejecución y validación de los modelos.
- **Robustez** del modelo. Se ha optado por modelos con unos requisitos mínimos de preprocesado de datos. Los datos se han tratado con los métodos habituales para evitar nulos, uniformizar las escalas y factorizar algunas variables, pero no se ha modificado el posible sesgo de las distribuciones de cada variable ni, salvo el predictor, se han estratificado al realizar las validaciones cruzadas. El objetivo era obtener un conjunto de modelos que fuese capaz de obtener predicciones significativas con un mínimo tratamiento de los datos para facilitar su uso en la práctica clínica.

6.1 Modelos

Con los criterios indicados, y después de una serie de pruebas con los modelos soportados por los *frameworks* de R: CARET[6] [86] (http://topepo.github.io/CARET/train-models-by-tag.html#Two_Class_Only) y MLR[7] [87] (https://mlr-org.github.io/mlr-tutorial/release/html/integrated_learners/index.html), se seleccionaron variantes de modelos del tipo *Generalized Lineal Model, adaBoost, Random Forest, Gradient Boosting, NeuralNet, Naive Bayes y SuperLearner*, que se describen de forma breve a continuación [88], [89], [90]

6.1.1 Generalized Lineal Model

Los modelos **glm** son una generalización de los modelos de regresión lineal, los parámetros se obtienen mediante el método de la máxima verosimilitud y la variable respuesta se construye como una combinación lineal de los predictores, de forma similar a la regresión lineal. La

[6] *Classification And Regression Training*
[7] *Machine Learning in R*

generalización se obtiene suponiendo una distribución de probabilidad de tipo exponencial para la variable de respuesta y utilizando una función *link* para relacionar la media de esta distribución con la combinación lineal de los predictores.

En un modelo para clasificación binaria, o regresión logística, la distribución de probabilidad utilizada es Binomial: [91] o Bernoulli [92]:

$$Binomial \rightarrow P(X = k) = \binom{n}{k} p^k (1-p)^{n-k},$$
$$Bernoulli \rightarrow P(X = k) = p^k (1-p)^{1-k}.$$

Y como función *link* se emplea la función *logit*, de forma que la regresión logística se puede expresar como:

$$\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \beta_k x_k = \ln\left(\frac{p}{1-p}\right).$$

Donde $\left(\frac{p}{1-p}\right)$ es la ratio de probabilidades (*odds ratio*) de éxito y $\beta_k x_k$ son los términos de la combinación lineal de los predictores.

Con la regresión logística se modela la probabilidad de que la respuesta pertenezca a una determinada clase, con la ventaja de que es posible obtener la importancia de los predictores o covariables y que los coeficientes obtenidos tienen una interpretación muy similar a la regresión lineal, en este caso un incremento de una unidad en una variable determinada lo que indica es una multiplicación de la ratio de probabilidades de la clase fijada como éxito por e^{β}

La regresión logística relaja varios de los presupuestos de la regresión lineal, en este caso la relación lineal se da entre las variables y la ratio de probabilidades y no se requiere la normalidad de los residuos, la homoestacidad, ni un especial cuidado en la normalización y escalado de las variables [88].

Algunas implementaciones [93] permiten utilizar regularización del tipo *Lasso* (L_1) o *Ridge* (L_2) para limitar el crecimiento de los coeficientes en modelos con gran número de variables. La regularización *Lasso* penaliza la suma del valor absoluto de los coeficientes:

$$\|\beta\|_1 = \sum_{k=1}^{n} |\beta_k|,$$

mientras que la regularización *Ridge* lo hace con la suma cuadrática:

$$\|\beta\|_2 = \sum_{k=1}^{n} \beta_k^2.$$

6.1.2 adaBoost

Forman parte de una familia de modelos de clasificación que utilizan las técnicas de *boosting* adaptativo, en las que se combinan clasificadores débiles para formar un clasificador fuerte. Un clasificador débil puede consistir simplemente en una regla del tipo *atributo=valor.*

El modelo **adaBoost** es iterativo, parte de un conjunto de ***M*** modelos $G_m(x)$ a los que les asocia un peso uniforme w_m en la primera iteración, estos modelos se entrenan y se calcula el error ponderado, como la suma de las observaciones mal clasificadas por su peso y divididas por la suma de pesos en cada iteración *i*:

$$err_i = \frac{\sum_{j=1}^{m} w_j I\left(y_j \neq G_i(x_j)\right)}{\sum_{j=1}^{m} w_j}.$$

Y se ajusta el peso de cada modelo mediante el logaritmo de la ratio entre la exactitud y el error:

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1 - err_i}{err_i}\right).$$

El peso de cada modelo se multiplica por e^{α_i} si ha clasificado incorrectamente y por $e^{-\alpha_i}$ si lo ha hecho correctamente. Se renormalizan los pesos para que su suma sea 1 y se itera de nuevo.

Una vez realizado el número de iteraciones definido se obtiene el clasificador como el signo de la suma ponderada de la salida de todos los modelos:

$$G(x) = sign\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right).$$

El modelo **adaBoost** generalmente necesita solo un parámetro que está relacionado con el número de iteraciones o con el número de clasificadores débiles a utilizar.

6.1.3 Random Forest

Los árboles de decisión se pueden usar tanto para modelos de regresión como de clasificación. El objetivo de los modelos de este tipo es dividir el conjunto de datos en particiones más homogéneas, de forma al realizar la partición se maximice (o minimice según el caso) la métrica utilizada, que puede ser la exactitud de la clasificación, la homogeneidad de los grupos, el error de clasificación o indicadores como la entropía de una partición:

$$S = \sum_{i=1}^{n} -p_i \log_2 p_i,$$

el índice GINI, que para un problema de dos clases con probabilidades p_1 y p_2 se calcula de la forma:

$$G = 2p_1p_2,$$

o la ganancia de información, también para dos clases:

$$info = -[p_1 log_2 p_1 + p_2 \log_2 p_2].$$

Para evitar el sobreentreno (*overfitting*) se limita la profundidad de los árboles o se utiliza alguna técnica de poda (*prune*) en la que el árbol de deja crecer hasta su máxima profundidad y una vez completado se eliminan ramas con un criterio determinado de coste-complejidad, que generalmente es parametrizable.

El modelo **Random Forest** utiliza como base la técnica del *bagging* (*bootstrap aggregation*), en la que se combinan múltiples modelos de árboles de decisión entrenados con una muestra de los datos originales obtenida mediante *bootstrap*, es decir, muestreos con reemplazo, y se obtiene la predicción como el promedio de las predicciones de los diferentes modelos así entrenados. Random Forest genera un número ***M*** de modelos mediante iteraciones en las que los árboles construidos para cada muestra se particionan utilizando sólo el mejor de un subconjunto ***k*** del total de variables de los datos. Se generan así del orden de 1000 o más árboles diferentes sin podar, que se promedian para realizar las predicciones, reduciendo la correlación entre los modelos utilizados.

Los modelos **Random Forest** generalmente admiten como parámetros el número de predictores ***k*** (m_{try}) seleccionados en cada iteración y el número de árboles ***M*** total. La recomendación es fijar inicialmente ***k*** a la raíz cuadrada del número de predictores y ***M***=1000.

Este tipo de modelos no necesitan un preprocesado específico de los datos y son relativamente insensibles a la parametrización. Desde el punto de vista computacional tienen la ventaja de su facilidad de paralelización, al poder entrenarse los árboles de forma independiente.

6.1.4 Gradient Boosting

Gradient Boosting utiliza también un conjunto de modelos débiles para construir uno robusto como **adaBoost** o **Random Forest**, aunque en este caso la creación de los modelos es diferente y se realiza de forma aditiva y minimizando una función de pérdida. Combina la técnica de optimización *gradient-descend* con las ventajas estadísticas del *boosting* [94] [95].

Para el caso de los problemas de clasificación en dos clases (-1,1), el modelo **Gradient Boosting** utiliza generalmente árboles de decisión como modelos base, y la función de pérdida log-verosimilitud binomial negativa [95]:

$$L(y, F) = \log(1 + e^{-2yF}), \qquad y\epsilon\{-1,1\},$$

donde

$$F(x) = \frac{1}{2}\log\left[\frac{\Pr(y=1|x)}{\Pr(y=-1|x)}\right].$$

El modelo final se calcula mediante iteraciones utilizando *boosting*, de forma similar a **Random Forest**. En la primera iteración se inicializan todas las predicciones con un valor constante [96]:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

A partir de ahí, en cada iteración *m* se obtiene un muestreo (*bootstrap*) de los datos y se entrena un modelo de árbol de regresión $h_m\ (x)$ usando los pseudo-residuos (gradiente de la función de pérdida) como respuesta

$$r_{im} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)},$$

Se calcula el multiplicador γ_m para cada hoja del árbol

$$\gamma_{im} = \arg\min_{\gamma} \sum L(y_i, F_{m-1}(x_i) + \gamma),$$

y se actualiza el modelo total de la forma:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

El modelo final es el resultado de la suma de los modelos obtenidos en las diferentes iteraciones.

Se utiliza también un factor de regularización (*shrinkage*) de forma que cada modelo obtenido en una iteración determinada se escale por un factor *v* cuando es añadido al modelo total, de la forma:

$$F_m(x) = F_{m-1}(x) + v * \gamma_m h_m(x),$$

empíricamente se ha observado [95] que valores pequeños del parámetro de *shrinkage* v dan como resultado una mejor eficacia del modelo. Los parámetros principales para los modelos de Gradient Boosting son el número de iteraciones, el tamaño de los árboles de decisión y el parámetro *shrinkage* utilizado.

6.1.5 NeuralNet

Las redes neuronales se construyen a partir de un modelo simplificado del funcionamiento de las neuronas. Están formadas por nodos estructurados en capas, con una capa de entrada, otra de salida y una o varias ocultas

intermedias. Las neuronas de la capa de entrada toman como entrada los datos de los predictores, en la capa de salida se obtienen las predicciones y las neuronas de las capas intermedias operan con los datos de entrada de cada capa y los transfieren a la entrada de la siguiente. Las funciones de transformación utilizadas, así como la estructura y número de las diferentes capas, son las que definen el funcionamiento de la red neuronal.

Cada neurona realiza una transformación no lineal de alguna o todas de las variables de entrada. Esta transformación, conocida como la función de activación de la neurona, puede ser la misma para toda la red o diferente en cada una de las capas. Existen numerosas funciones de activación [97] y para problemas de clasificación las más usadas son la función *softmax* [90]:

$$f_i(\boldsymbol{X}) = \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}}$$

o la rectificadora (*maxout)* [98]:

$$f_i(\boldsymbol{X}) = \max_{\mathrm{i}}(0, x_i)$$

En problemas de clasificación binaria, el número de neuronas de la primera capa se corresponde con el número de predictores, la última capa solo tiene una neurona y el número de neuronas de las capas intermedias es variable.

La salida de cada neurona de la primera capa oculta es una combinación lineal de los predictores, de la forma:

$$h_k(x) = g\left(\beta_{0k} + \sum_{i=1}^{P} x_i \beta_{ik}\right),$$

con *g(x)* la función de activación utilizada. De forma similar las neuronas de la segunda capa oculta son una combinación lineal de las salidas de la primera capa oculta y así de forma sucesiva hasta llegar a las neuronas de la capa de salida, que en el caso de clasificación binaria sería una combinación lineal de la forma:

$$f(x) = \gamma_0 + \sum_{k=1}^{H} \gamma_k h_k$$

Entrenar una red neuronal significa estimar todos los parámetros β, γ (pesos) de cada una de las neuronas en cada capa que, por ejemplo, en el caso de una red con P predictores, una capa oculta de H neuronas y una sola neurona de salida son $H(P+1) + H + 1$ parámetros, lo que da idea de la magnitud del problema de la estimación de parámetros en una red neuronal.

Para el caso del problema de clasificación, entrenar una red neuronal consiste en minimizar la entropía de la red

$$R(\theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} log f_k(x_i)\,, con\ \theta\ el\ conjunto\ de\ parámetros$$

utilizando la técnica de la retropropagación (*back-propagation*) con las derivadas de la función de activación para calcular el gradiente. En un primer paso se fijan los pesos de la red neuronal de forma aleatoria y se calculan los valores predichos para los datos de entrenamiento, en una segunda fase se calcula el error de las predicciones y se actualizan los pesos de las neuronas de cada capa hacia atrás, de la última hacia la primera, con el gradiente de la función de error multiplicado por un factor de aprendizaje γ

$$\beta_{km}^{(r+1)} = \beta_{km}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{(r)}}$$

este proceso es iterativo y se realiza un número de veces fijado por parametrización o hasta que se alcance un umbral de error máximo determinado.

El proceso de retropropagación se puede realizar para cada observación individual en lugar de para los valores promediados de todas las observaciones, mediante la técnica de *stochastic gradient descend*, que facilita el entrenamiento de las redes neuronales en tiempo real.

Para mitigar la tendencia de las redes neuronales al *overfitting* se utilizan técnicas de penalización como el *weight decay,* en la que se multiplican por una constante λ la suma de los cuadrados de los pesos y se añaden a la función de coste utilizada.

$$\tilde{R}(\theta) = R(\theta) + \lambda\theta^2$$

La parametrización de las redes neuronales es compleja, dado el número de parámetros que pueden llegar a tener y las relaciones no lineales entre ellos y el resultado obtenido, para ello es útil el uso de métodos de optimización combinatoria.

6.1.6 Naive Bayes

Estos modelos se basan en la regla de Bayes y en el supuesto de independencia de las probabilidades de los predictores. Aplicada a problemas de clasificación, la regla de Bayes se puede formular como:

$$\Pr[Y = C_l|X] = \frac{\Pr[Y]\,\Pr[X|Y = C_l]}{\Pr[X]},$$

donde $\Pr[Y = C_l|X]$ es la probabilidad a posteriori que se pretende estimar, es decir la probabilidad de que la variable respuesta Y pertenezca a la clase C_l dados los predictores X, $\Pr[Y]$ es la probabilidad a priori de la

variable respuesta, $\Pr[X]$ es la probabilidad de los predictores y $\Pr[X|Y = C_l]$ es la probabilidad condicional de los predictores dados los datos asociados a la clase C_l.

Los modelos **Naive Bayes** simplifican el cálculo de estas probabilidades suponiendo la independencia de las probabilidades de los predictores, de forma que la probabilidad total de cada uno de los términos anteriores se pueda calcular como producto de las probabilidades de cada una de las variables:

$$\Pr[X|Y = C_l] = \prod_{j=1}^{P} \Pr[x_j|Y = C_l]$$

Los modelos **Naive Bayes** calculan las probabilidades individuales a partir de las frecuencias observadas de los datos, en el caso de variables categóricas, y estimando las densidades de probabilidad, en el caso de las funciones continuas.

La ventaja de los modelos **Naive Bayes** es su rapidez de cálculo y el no necesitar parámetros adicionales en general, aunque su eficacia depende de lo ajustado que sea a los datos el supuesto de independencia de las variables.

6.1.7 SuperLearner- Ensemble

Como se indicará en posteriores secciones, una métrica adecuada para estudios en las áreas de bioestadística y sanidad es el AUC (*area under the curve)* de las curvas ROC (*response operating characteristic*), que permite evaluar el comportamiento de un modelo predictivo en situaciones de desbalanceo de clases y donde es importante conocer el ratio de falsos positivos y falsos negativos por su diferente influencia en el objeto de estudio, por ejemplo la mortalidad de pacientes [99]. La mayor parte de los modelos de clasificación se basan en la optimización de una función objetivo relacionada con la precisión o el error de clasificación y en problemas con clases desbalanceadas consiguen resultados diferentes según la clase predicha sea mayoría o minoritaria. Volviendo al ejemplo de la mortalidad en pacientes, es posible que un modelo clasifique el 100% de los casos de pacientes que sobreviven y en cambio obtenga un resultado muy pobre respecto de los pacientes que no lo hacen, cuando justamente el objeto del estudio es analizar las variables que influyen en la no supervivencia del paciente.

Una forma de introducir la evaluación de funciones objetivo enfocadas a la maximización de la métrica AUC en modelos diseñados para utilizar otras métricas de eficacia es emplear técnicas de apilamiento (*stacking*) de modelos, diseñando meta-modelos e introduciendo una función objetivo en su selección. **SuperLearner** [100] es un meta-modelo de ensamblaje de distintos modelos (o el mismo con parametrización diferente) que utiliza técnicas de validación cruzada para obtener una

combinación ponderada de los modelos según la función objetivo deseada.

SuperLearner [101] obtiene la combinación óptima de una serie de L modelos base:

$$\{\psi_1, \dots, \psi_L\},$$

que se entrenan cada uno mediante técnicas de validación cruzada con un conjunto de n observaciones para obtener una matriz de $n \times L$ predicciones, denominada datos de nivel uno, que se utilizan a su vez junto con el conjunto original de variables respuesta para entrenar el metamodelo de forma que se minimice la función objetivo mediante técnicas de optimización. Para obtener los pesos asociados a cada uno de los modelos base, se pueden utilizar modelos de regresión lineal, logística, etc. En caso de usar regresión lineal, el metamodelo $\hat{\psi}$ trata de optimizar los pesos de cada modelo base para obtener la combinación lineal:

$$\hat{\psi} = \sum_{i=1}^{L} \beta_i \psi_i,$$

que minimice el error cuadrático para el conjunto de observaciones O:

$$L(O, \hat{\psi}) = \left(Y - \hat{\psi}(X)\right)^2.$$

De forma general se puede considerar como un problema de minimización del riesgo de validación cruzada (*minimum cross-validated risk*)

$$R_{CV}(\beta) = \sum_{i=!}^{n} \left(Y_i - m(Z_i|\beta)\right)^2,$$

donde $m(Z_i|\beta)$ es el metamodelo utilizado para estimar los pesos de los diferentes modelos base y Z los valores de las predicciones de nivel 1. De esta forma obtenemos los parámetros β_n:

$$\beta_n = \arg\min_{\beta} R_{CV}(\beta),$$

Y el metamodelo

$$\hat{\psi}(z) \equiv m(z, \beta_n)$$

El uso de **SuperLearner** se ha mostrado efectivo en varios *benchmarks* con clases no balanceadas [99] [59], tiene la ventaja de que no precisa de parametrización adicional a la utilizada en los modelos componentes del ensamblaje.

Existen básicamente dos implementaciones del algoritmo **SuperLearner**, una en R [102] y otra en H2O [103].

6.2 Evaluación de modelos

Una vez preparados los datos y seleccionados los modelos candidatos se deben tener en cuenta una serie de factores: búsqueda de los parámetros adecuados (***tunning***) según las características de los datos, particionado (***splitting***) en subconjuntos de test y training, evaluación de la eficiencia de los modelos mediante técnicas de remuestreo (***resampling***) y la comparación entre los diferentes modelos usando alguna métrica adecuada (***scoring***). [89], siguiendo las fases indicadas en la figura 28.

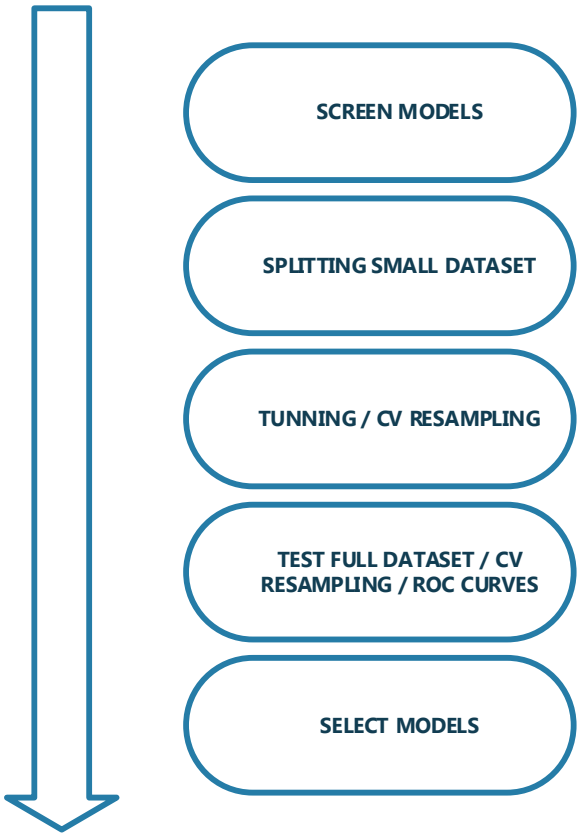

Figura 28: Evaluación de modelos predictivos.

6.2.1 Tunning

Existen diferentes técnicas para encontrar los parámetros adecuados para un modelo aplicado a unos datos determinados. El problema de la selección de parámetros se puede considerar un problema de optimización combinatoria en un espacio multidimensional. En algunos modelos sencillos con pocos parámetros y comportamientos lineales frente su modificación es posible abordar el problema de una forma manual, en otros, como por ejemplo las redes neuronales, se deben utilizar otras técnicas debido a la alta dimensionalidad del espacio de búsqueda, la no linealidad de la respuesta y la existencia de múltiples mínimos locales. Con paquetes de R como CARET [86] y H2O [104] es posible definir un conjunto de valores límite para los diferentes parámetros y realizar una búsqueda manual o aleatoria [105] (*hyperparameter*

tunning), obteniendo gráficas que muestran el comportamiento del modelo frente a la parametrización. El *framework* de R MLR [106] permite utilizar una variedad más amplia de técnicas [107] como optimización con estrategias evolutivas (CMA), *simulated annealing* e *Iterate Racing* [108]. Este último método utiliza la técnica de optimización llamada r*acing* [109], en la que se realizan test en paralelo con varias parametrizaciones de modelos, descartándose las peores y realizando iteraciones sucesivas con las mejores, empleando técnicas estadísticas para obtener muestras representativas del espacio de parámetros mediante distribuciones de probabilidad que se actualizan en las sucesivas iteraciones y técnicas para evitar los mínimos locales.

En este análisis la parametrización inicial se ha realizado de forma semimanual con CARET y refinándola posteriormente con el algoritmo iRace de MLR.

6.2.2 Splitting

Al trabajar con un conjunto de datos fijo, siempre existe el riesgo de *overfitting* [89], cuando los modelos entrenados con un conjunto de datos son capaces de predecir un porcentaje cercano al 100% de los datos utilizados para el entrenamiento pero se comportan con una exactitud muy pobre con datos nuevos. Este riesgo de *overfitting* es mayor en modelos del tipo de redes neuronales, debido al gran número de parámetros que se ajustan en el proceso de entrenamiento [110] y se intenta mitigar mediante regularizaciones (L1 y L2).

El método general para evitar *overfitting* y evaluar la calidad de los modelos frente a datos nuevos, es separar el conjunto de datos en dos subconjuntos, uno de entrenamiento (*training*) con el que se realiza todo el proceso de *tunning* y entrenamiento y otro de test, con el que se valida el comportamiento del modelo frente a datos nuevos. Cuando se trata de evaluar el comportamiento de diferentes modelos y se dispone de datos suficientes, la recomendación es separar el conjunto de datos en tres subconjuntos [90] Entrenamiento/Validación/Test. Con el subconjunto de entrenamiento se construyen los modelos, el de validación se emplea para estimar la calidad de los diferentes modelos y realizar la selección y el de test para estimar el error al aplicarlo a datos nuevos (Ver figura 29).

En caso de variables respuesta desbalanceadas, es importante asegurarse que la distribución de probabilidades de los subconjuntos de datos de entrenamiento y test respecto de la variable respuesta sea la misma, lo que se conoce como estratificación. Este punto podría tenerse en cuenta para todas las variables predictores, pero en nuestro caso hemos utilizado muestras de tamaño suficiente y eliminado las variables de varianza casi nula por ser técnicamente menos complejo. Los *frameworks* utilizados CARET, MLR y H2O disponen de funciones para realizar el estratificado de los datos para la variable respuesta, pero no para el resto de variables predictoras.

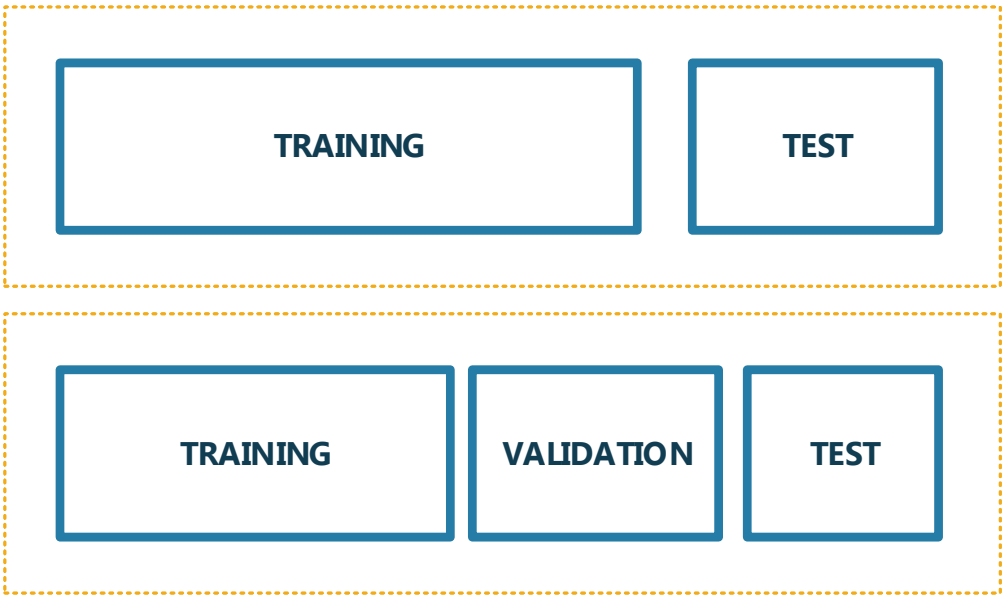

Figura 29: Splitting de datos.

En general se han utilizado particiones entrenamiento/test del orden de 60/40, 75/25 según el experimento a realizar, obteniendo unas métricas comparables a las obtenidas en la validación cruzada.

6.2.3 Resampling

Dividir el conjunto de datos en subconjuntos de entrenamiento/test o entrenamiento/validación/test ayuda a detectar y evitar el *overfitting*, comparando las métricas obtenidas cuando se introducen en el modelo los datos de entrenamiento y los de test, pero no basta para obtener valores estadísticamente significativos del comportamiento del modelo y, por otra parte, existe generalmente una limitación con el número de observaciones 'no vistas' por el modelo que se pueden emplear para caracterizar su comportamiento. Estos problemas se acentúan con conjuntos de datos limitados [89].

Un método probado para evitar este tipo de problemas y obtener indicadores estadísticamente significativos es el remuestreo (*resampling*) de los datos empleados para entrenar el modelo. Existen varias técnicas, las más utilizadas son la K-validación cruzada (*k-Fold Cross-Validation*) [89] [90] y 'dejar uno fuera' (*leave-one-out*).

La validación cruzada (CV) es el método más sencillo e implementado, consiste en dividir el conjunto de datos de entrenamiento en *K* subconjuntos y para cada uno de ellos utilizar los *K-1* subconjuntos restantes para entrenar el modelo, usando para caracterizar el modelo el K-subconjunto seleccionado para realizar la predicción. Esto se repite para los *K* subconjuntos en los que se ha dividido el conjunto de entrenamiento. La figura 30 muestra esquemáticamente el funcionamiento de la validación cruzada.

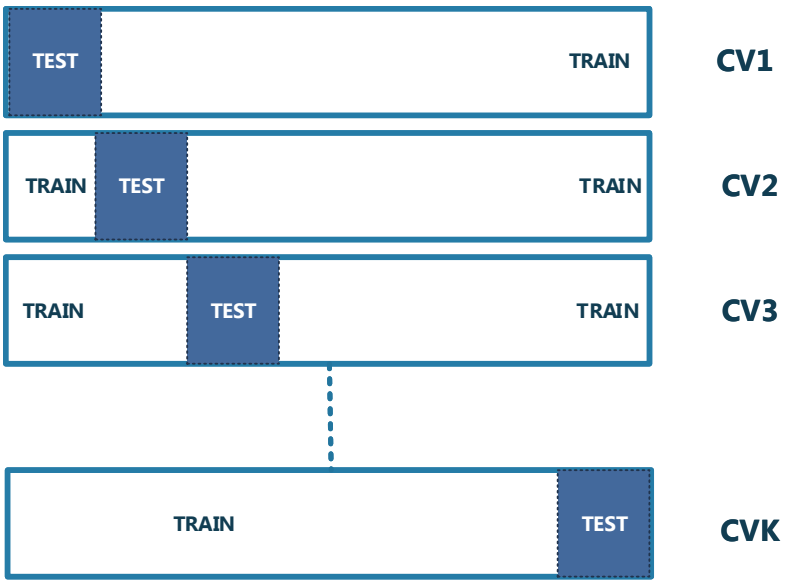

Figura 30: Validación cruzada.

La métrica usada para validar el modelo se promedia entre las obtenidas en las *K* validaciones realizadas, obteniéndose así la estimación del error de predicción según la función de pérdida utilizada *L:*

$$CV(m) = \frac{1}{K}\sum_{i=1}^{K} L\left(y_i, m^{-K_i}(x_i)\right),$$

Donde m^{-K_i} indica la predicción obtenida usando la partición K_i del modelo entrenado con las particiones $K_{j\neq i}$. Generalmente se usan valores de entre K=5 y K=10.

El método de dejar uno fuera (*leave-one-out*) es el caso extremo de la K-validación cruzada, dejando en cada iteración solo una observación fuera y entrenando el modelo con el resto de observaciones, en este caso *K=N* donde *N* es el número de observaciones y el error de predicción se obtiene entonces de la forma:

$$CV(m) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, m^{-K_i}(x_i)\right),$$

En la práctica, el compromiso de usar K=10 es generalmente utilizado, en conjuntos de datos grandes K=3 puede llegar a ser bastante preciso. Un valor de *K* grande implica mayor coste computacional, menor bias y mayor varianza de los resultados, mientras que un valor pequeño de *K* es computacionalmente menos costoso, obtiene menor varianza y mayor bias. [89], [90].

6.2.4 Curvas ROC

La evaluación de calidad de los diferentes modelos se puede realizar utilizando diferentes métricas para cuantificar el error de predicción,

diferenciando principalmente si se trata de modelos de regresión, donde se evalúan los residuos o modelos de clasificación, en los que se suelen utilizar medidas de error de clasificación.

En el caso de los modelos de clasificación binarios [111] [89] se utiliza la tabla de confusión y sus indicadores asociados. La matriz de confusión indica las observaciones correctamente e incorrectamente clasificadas:

		OBSERVED	
		EVENT	**NON EVENT**
PREDICTED	**EVENT**	tp	fp
	NON EVENT	fn	tn

Donde:

- tp → *True Positives*
- fp → *False Positives*
- tn→ *True Negatives*
- fn→ *False Negatives*

Y a partir de estos valores se definen los indicadores siguientes, donde *n=tp+fp+fn+tn* es el número de observaciones total:

- ***Accuracy*****:** $a = (tp + tn)/n$
- ***Precision*****:** $p = tp/(tp + fp)$
- ***Sensitivity / Recall / True Positive Rate*****:** $fpr = tp/(tp + fn)$
- ***Specifity / True Negative Rate*****:** $tnr = tn/(tn + fp)$

Que se pueden interpretar de la forma siguiente: *Sensitivity/Recall* indica la bondad del modelo para detectar los casos positivos, *Specifity* indica la bondad del modelo para evitar los falsos positivos y *Precision* indica cuantos de los eventos clasificados como positivos son relevantes [112].

Estos indicadores se pueden combinar en uno solo con el índice de *Youden* [89]:

$$J = Sensitivity + Specifity - 1,$$

aunque el método más general y ampliamente usado en la literatura biomédica son las curvas ROC (*Receiver Operating Characteristic*) [89] [113] [114] [115], calculadas a partir de las clases obtenidas por el modelo cuando se varía el punto de corte que decide la pertenencia a la clase del evento desde el 0% al 100% de forma continua. Para ello se obtienen las predicciones del modelo como probabilidades y los indicadores de *Sensitivity (tpr)* y *1-Specifity (fpr)* para todo el rango. En la curva resultante se observa el comportamiento del modelo para los distintos valores del punto de corte y el área bajo la curva (AUC) es un indicador de la eficacia del modelo, cuanto mayor sea, mejor es la capacidad predictiva. La línea diagonal que marca el área de 0.5 se corresponde con un modelo con una

predicción aleatoria y es lo que se obtendría para un modelo con nula efectividad predictiva.

La figura 31 es un ejemplo de curva ROC (demostración para Wolfram http://demonstrations.wolfram.com/HowReceiverOperatingCharacteristicCurvesWork/)

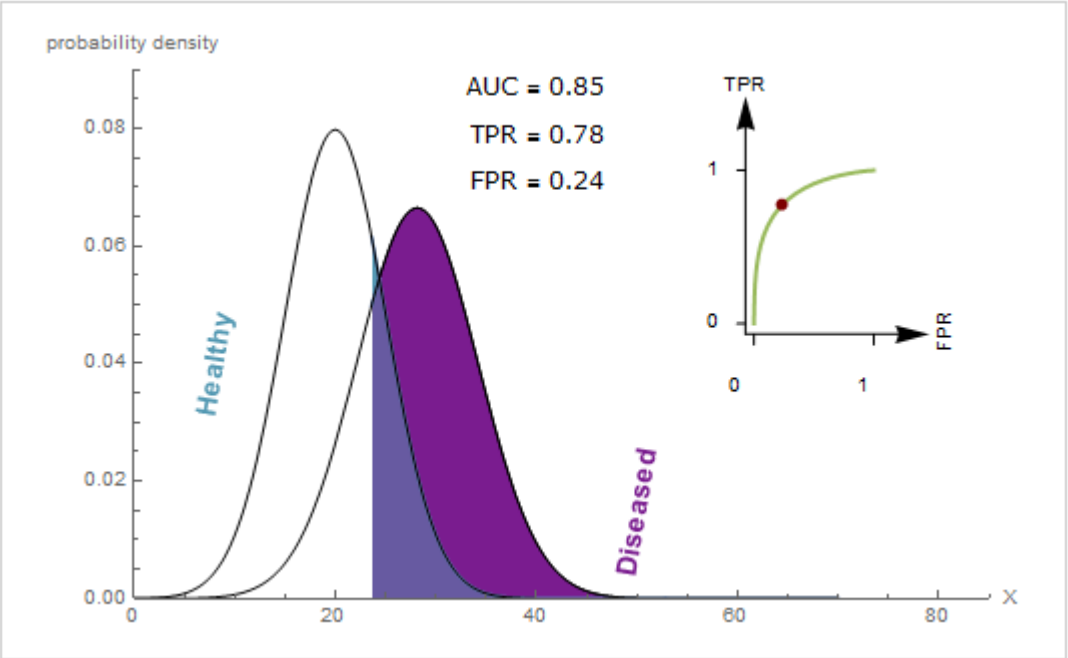

Figura 31: Curvas ROC.

La gráfica ROC permite, además de evaluar la eficacia de un modelo, elegir el punto de corte que constituya el mejor compromiso entre sensibilidad (*Sensitivity)* y especificidad (*Specificity*). Las dos variables están relacionadas inversamente, cuando una aumenta la otra disminuye y viceversa. Otra ventaja de esta métrica es que es insensitiva frente al desbalanceo de las clases, aunque en contra tiene que al agregar el comportamiento del modelo en el indicador AUC se pierde información de los detalles de la curva en caso de que interesen regiones específicas, para lo que se han desarrollado métricas que miden áreas parciales (pAUC) [116].

La curva ROC y el área bajo la curva AUC son los indicadores principales que se utilizarán en el análisis para evaluar los diferentes modelos.

7 ANÁLISIS DE DATOS

En este apartado se detalla el proceso de análisis de datos realizado, la evaluación de las diferentes herramientas y modelos y los principales resultados obtenidos. En los anexos se incluyen más información de los programas desarrollados, los conjuntos de datos y detalles técnicos.

Aunque el desarrollo del análisis se ha realizado de forma iterativa, para su claridad se expone en las siguientes fases secuenciales: análisis exploratorio de datos, preparación adicional de datos, selección de modelos/herramientas, *tunning* de modelos, *benchmarking* y aplicación de modelos. Estas fases se corresponden con las tareas iterativas de **Platform/Tool Selection** y **Modeling** de la metodología adaptada [8] tal como se indica en la figura 32.

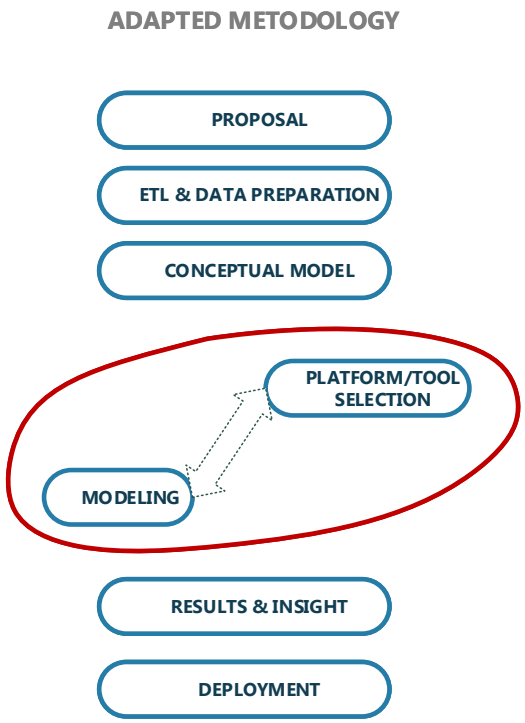

Figura 32: Metodología adaptada

7.1 Análisis exploratorio

A partir de los procesos ETL ejecutados sobre la BBDD MIMIC-III se obtiene en la BBDD intermedia MIMICSEL la tabla **DS_ICU_STAYS_1,** seleccionando los registros mediante los siguientes criterios:

- Pacientes adultos.
- Estancia en UCI mayor que dos días.
- Más de tres mediciones de glucosa en cada uno de los tramos de 0-24H y 24-48H.

[8] Ver apartado 2.3

- Diagnostico principal codificado.

El *outcome*, o variable para la que obtenemos las predicciones, es la mortalidad a 28 días del ingreso en UCI **icu_dead_before_28**, usado comúnmente como indicador de mortalidad a corto plazo en entornos de cuidados intensivos [117], aunque no existe un criterio fijo sobre el número de días a considerar [118], dependiendo este número del tipo y objetivos del estudio.

La tabla **DS_ICU_STAYS** mencionada contiene 508 variables, incluyendo datos administrativos de pacientes, estancias, codificación, comorbilidades, scores, prescripción y administración de insulina, nutrición, resultados de laboratorio y mediciones de glucosa, entre otros. Se realizó una primera selección de 168 variables con los siguientes criterios: 1) se incluyeron los datos relacionados con los niveles de glucosa y su variabilidad, 2) para las mediciones de laboratorio se seleccionaron los valores medios por tramo de 24H, 3) se descartaron variables puramente administrativas, 4) no se incluyeron tampoco las variables de agregados para todo el episodio de UCI y 5) se mantuvieron todos los *scores* para obtener comparativas en el análisis exploratorio inicial.

La estrategia de selección inicial de variables está alineada con el objetivo de obtener modelos predictivos para los datos de los tres primeros días de estancia en UCI, analizando la influencia de las variables asociadas a la glucosa y las comorbilidades. El conjunto de datos seleccionado se detalla en los anexos y consta de 20445 observaciones y 168 variables.

En la tabla 8 se muestra la población considerada para la población en general y agrupada por variables seleccionadas por su de interés clínico. Cada registro u observación se corresponde con un episodio de UCI independiente.

GROUP	Nº ROWS	MORT. 28 DAYS
Población	20445 episodios UCI	18.70%
Estancia en UCI >72H	**SI:** 14430 (70.58%)	21.60%
	NO: 6015 (29.42%)	11.73%
SOFA >= 4	**SI:** 13230 (64.71%)	22.78%
	NO: 7215 (35.29%)	11.20%
Diabetes	**SI:** 7815 (38.22%)	17.34%
	NO: 12630 (61.78%)	19.54%
Administración insulina IV	**SI:** 6912 (33.81%)	16.29%
	NO: 13533 (66.19%)	19.93%
Sepsis ingreso	**SI:** 9266 (45.32%)	27.44%
	NO: 11179 (54.68%)	11.45%

Tabla 8: Mortalidad en población seleccionada y agrupada.

En las gráficas 33 y 34 se muestra la distribución de datos de algunas variables significativas relacionadas con la glucosa y agrupadas por tramos de 14H.

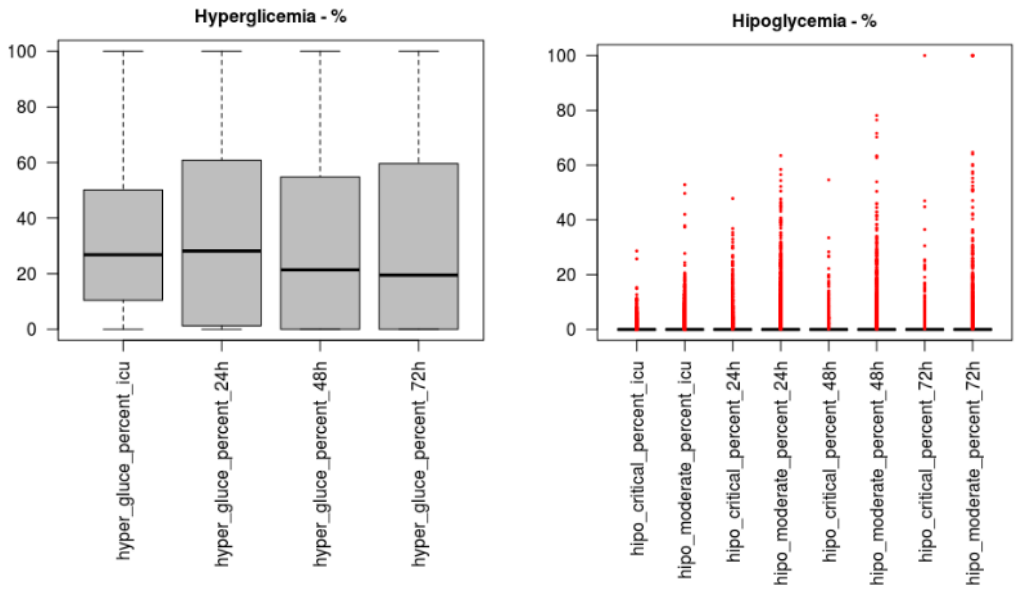

Figura 33: Porcentaje del tiempo en hiperglucemia e hipoglucemia por tramo de 24H.

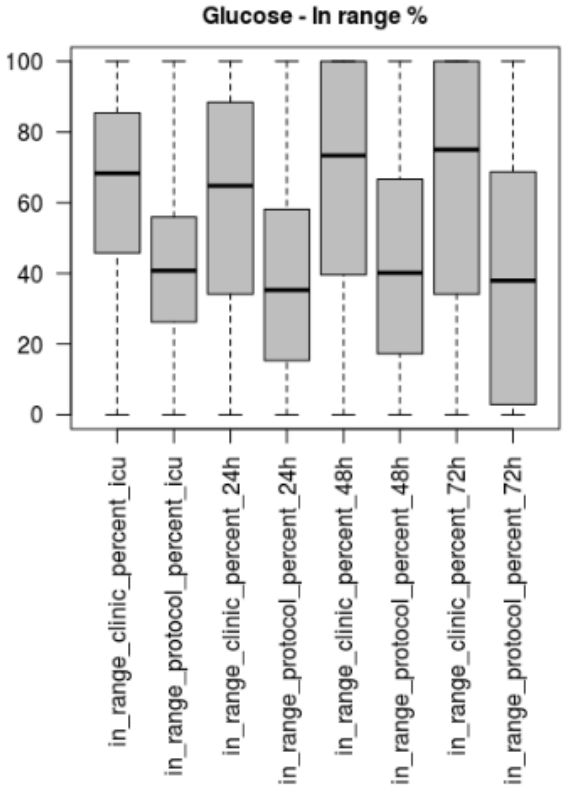

Figura 34: Glucosa. Porcentaje del tiempo en rango por tramos de 24H.

La figura 35 indica la distribución de las medias de variables clínicas agrupadas por tramos de 24H,

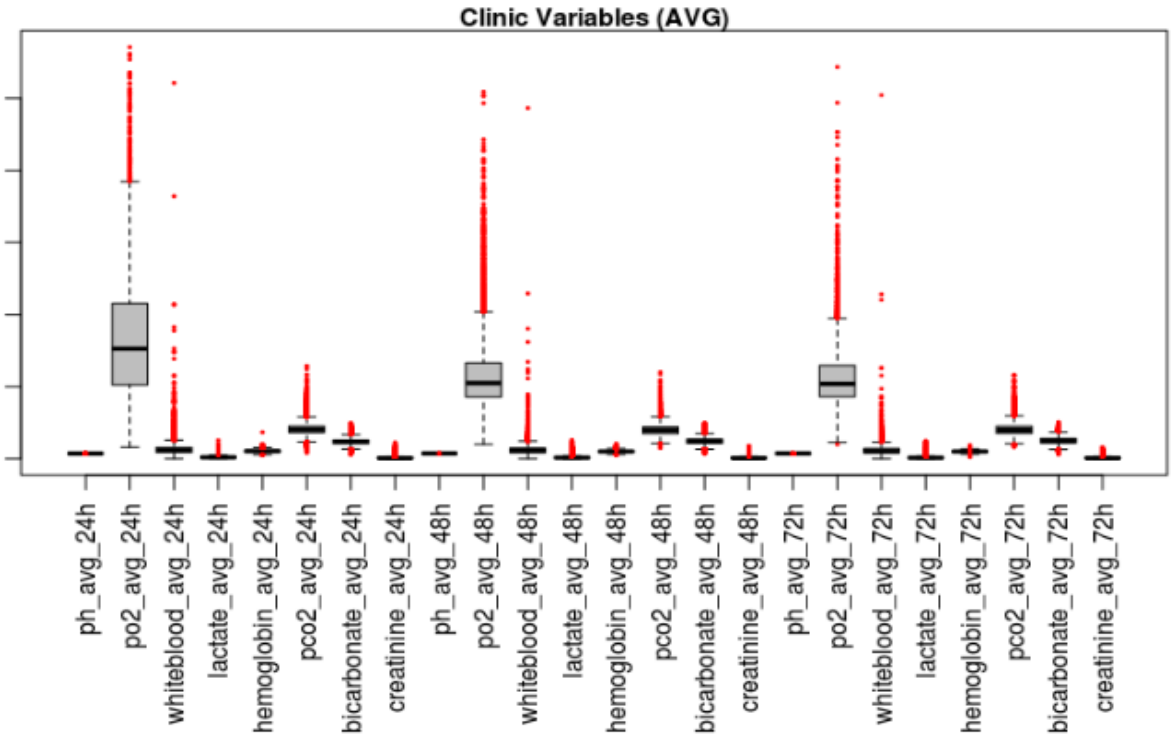

Figura 35: Variables clínicas por tramo de 24H.

y en la figura 36 se encuentra la distribución de la edad, el tiempo de estancia en UCI (*ICU LOS*) y la variabilidad de la glucosa para los tramos de 24H y 48H.

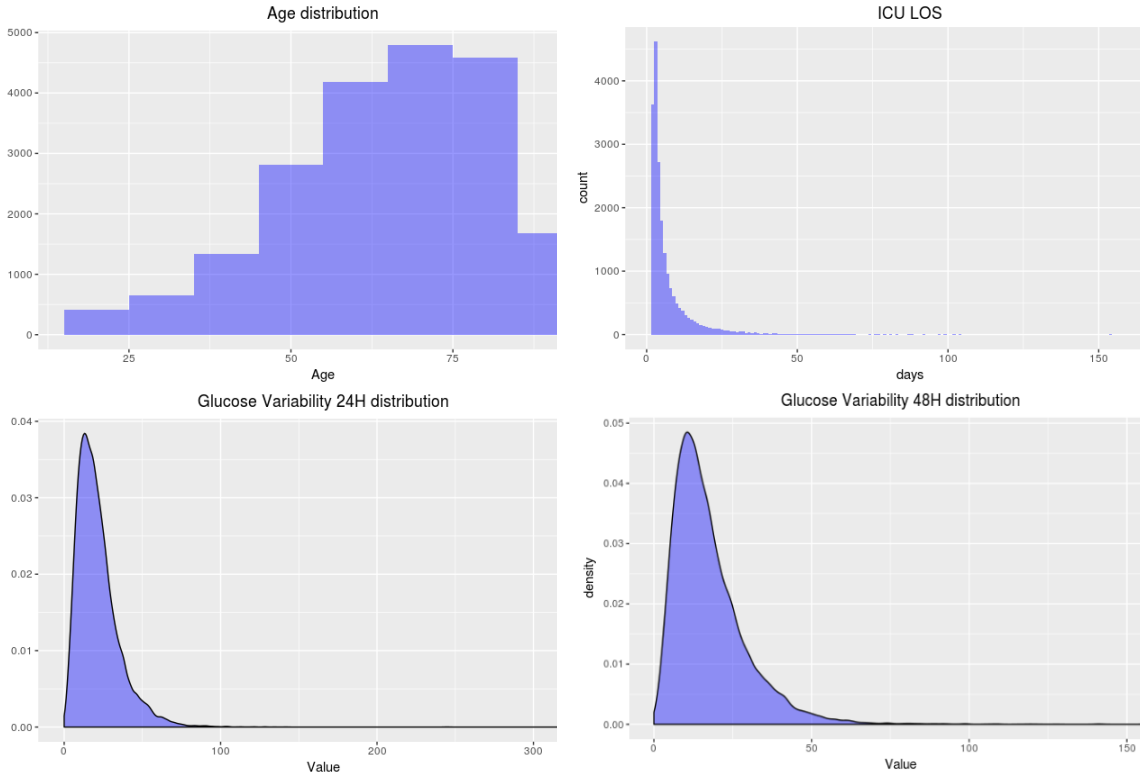

Figura 36: Distribución de edad, días de estancia en UCI, variabilidad de glucosa en 0-24H y en 24H-48H.

Con el fin de establecer una referencia inicial, se ha evaluado la capacidad predictiva de los scores disponibles en MIMIC-III, mediante una regresión logística, y obteniendo el valor de AUC de cada *score* para el *outcome* la mortalidad a 28 días. Los valores obtenidos para el área bajo la curva ROC para cada uno de ellos son APSIII (0.7052), LODS (0.6588), MLODS (0.6496), OASIS (0.6856), QSOFA (0.5672), SAPS (0.6559), SAPSII (0.725), SOFA (0.6544), SIRS (0.567) [9]. En la figura 37 se representa la distribución y las curvas ROC de los *scores* SAPSII y SOFA.

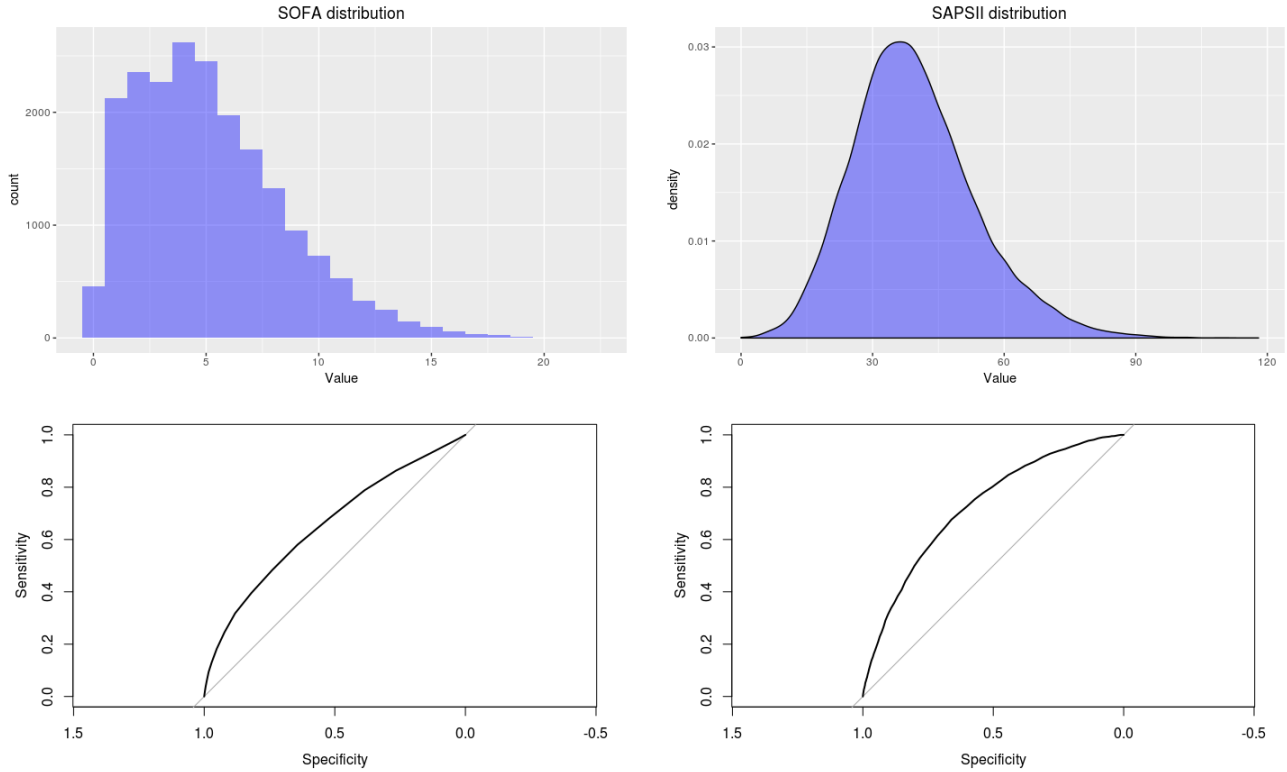

Figura 37: Scores SOFA y SAPSII, distribución y curvas ROC.

El *score* SAPSII, con un AUC de 0.725, se puede considerar la base de referencia con la que comparar si la capacidad predictiva de los modelos obtenidos en el análisis es suficiente o no.

7.2 Preparación adicional de datos

La preparación de datos obtenidos con la ETL se ha realizado utilizando R y CARET, en paralelo con el análisis detallado del conjunto de datos, y ha consistido principalmente en:

- Convertir todos los valores lógicos a numéricos (1/0).
- Recodificar la variable *outcome* **icu_dead_before_28** a un factor de dos clases (X0, X1).

[9] Los *scores* disponibles en MIMIC-III son los siguientes: APSIII [71], LODS [72], MLODS [72], OASIS [73], QSOFA [74], SAPS [75], SAPSII [76], SOFA [28], [77], SIRS [78]. Estos *scores* se usan para evaluar el estado de los pacientes críticos y ayudar a la decisión clínica [119], [120].

- Recodificar variables: convertir campos de dos factores a numéricos (1/0), factorizar edad por décadas, unificar campos de diagnóstico principal y secundario de diabetes.
- Eliminación de campos no relevantes para el estudio, campos con varianza casi nula, mediciones de laboratorio para los tramos de 48H y 72H con más de 20% de valores nulos, medidas para más de 72H, campos de desviación típica para los que existe su correspondiente campo de valor medio, campos de tiempo en rango de glucosa para los que existe su correspondiente campo de porcentaje en rango.
- Eliminación de variables de *scores,* salvo SOFA por formar parte de la selección realizada con criterio médico.
- Análisis y Eliminación de *outliers*.
- Imputación de variables nulas al valor más frecuente para datos factoriales y a la mediana para datos numéricos.
- Escalado de las variables numéricas al rango [0,1].

Como ejemplo de algunas de las operaciones de preparación de datos: el *framework* CARET facilita el preprocesado de los datos con funciones para realizar la imputación a campos nulos

```
preproc_ds_range<-preProcess(ds_icu,method=c("medianImpute","range"))
ds_icu<-predict(preproc_ds_range, ds_icu);
```

detectar campos con varianza casi nula

```
nearZeroVar(ds_icu);
```

con alta correlación

```
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.5);
```

o combinaciones lineales

```
comboInfo<-findLinearCombos(ds_icu_1f[sapply(ds_icu,is.numeric)]).
```

Una vez realizada la preparación de datos se obtuvo un conjunto de datos con 20445 observaciones y 79 variables, detallado en los anexos, con los que se llevaron a cabo las siguientes fases.

7.3 Selección de modelos/herramientas

Para la selección de herramientas y modelos se realizaron tests con los *frameworks* CARET y MLR*,* la aplicación H2O y diferentes modelos predictivos, utilizando R en todos los casos. Se decidió no realizar selección de variables, tanto por el elevado tiempo necesario, tal como se confirmó en las pruebas realizadas en CARET con modelos de regresión logística y optimización con algoritmos genéticos, como por la existencia en muchos modelos de funciones internas de selección de predictores significativos, lo que se indica en la propia documentación del *framework* CARET [121].

En los siguientes apartados se detallan las pruebas realizadas con CARET, MLR y H2O, para a continuación describir la selección realizada y su justificación.

7.3.1 CARET

CARET dispone de soporte para más de 233 modelos predictivos y utiliza una estructura unificada que facilita su uso. Después de varios tests iniciales, los modelos seleccionados fueron **adaboost** *(AdaBoost Classification Trees),* **glm** *(Generalized Linear Regression),* **ranger** (*Random Forest*), **RRF** (*Regularized Random Forest*), **gbm** (*Stochastic Gradient Boosting*), **nnet** (*Neural Network*), **pcaNNet** (*Neural Networks with Feature Extraction*), **nb** (*Naive Bayes*). La parametrización óptima de los modelos se realizó de forma semimanual e iterativa, utilizando *grids* de parámetros, analizando las gráficas de resultados (ver ejemplos en la figura 38) y evaluando el indicador AUC obtenido.

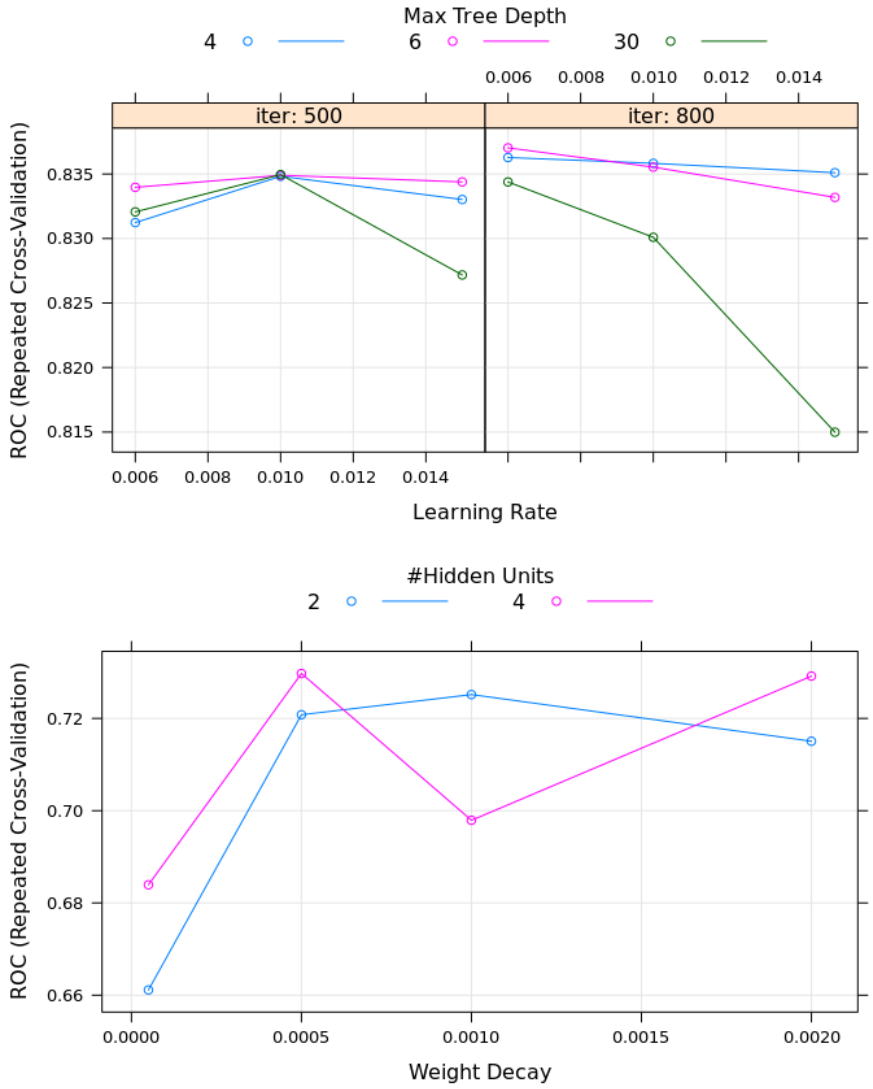

Figura 38: Ejemplos de gráficas de tunning de parámetros en CARET para los modelos ***ada*** *y* ***nnetPCA****.*

En la tabla 9 se indican los resultados obtenidos con la mejor parametrización. Se obtuvo el indicador AUC de la validación cruzada del subconjunto de entrenamiento (CV AUC) y el correspondiente a aplicar el modelo al subconjunto de test (TEST AUC).

Stratified partition 1228 obs. train, 818 obs. test, 79 vars, 3-CV, down subsampling.			
MODEL	PARAMS. (Best Model)	CV AUC	TEST AUC
glm		0.7186076	0.7445
ranger	mtry = 8	0.7669735	0.7683
RRF	mtry = 13, coefReg = 0.98, coefImp = 0.004	0.7590004	0.7529
gbm	n.trees = 300, interaction.depth = 4, shrinkage = 0.08, n.minobsinnode = 20	0.7648372	0.7487
nnet	size = 2, decay = 0.001	0.7040042	0.6593
pcaNNet	size = 2, decay = 0.009	0.7383237	0.7358
nb	fL = 0, usekernel = TRUE, adjust = 4	0.7445217	0.7608
adaBoost	iter = 3000, maxdepth = 3, nu = 0.006	0.7775294	0.7695

Tabla 9: Resultados tunning de parámetros en CARET.

7.3.2 H2O

Tomando como base la parametrización realizada en CARET se realizaron pruebas con los modelos equivalentes disponibles en H2O: **gbm**, **glm**, **RF** (*Random Forest*), **Naive Bayes, Deep Learning** (*Neural Nets*) y el modelo **Ensemble**.

La tabla 10 muestra los resultados con la parametrización realizada en H2O. En este caso se dispone de la media de los indicadores AUC obtenidos del entrenamiento de los modelos (TRAIN AUC), la validación cruzada (CV AUC), ambos obtenidos con el subconjunto de entrenamiento, y el valor correspondiente a la validación con el subconjunto de test (TEST AUC).

Stratified partition 12268 obs train, 8177 obs test, 75 vars, 10-CV				
MODEL	PARAMS. (Best Model)	TRAIN AUC	CV AUC	TEST AUC
glm		0.8121705	0.8041160	0.8168504
RF	ntrees = 1000, mtries = 8, nfolds = 10	0.9999997	0.8320329	0.8228742
gbm	ntrees = 1000, max_depth = 4, min_rows = 2, learn_rate = 0.001	0.8118567	0.7812560	0.7954309

Deep Learning	nfolds = 10, hidden = 10, variable_importances = TRUE, input_dropout_ratio = 0.1, distribution = "bernoulli",	0.8590460	0.7890243	0.8095943
nb		0.7263558	0.7196371	0.7494017
Ensemble	todos los modelos anteriores menos glm	1		0.8353542

Tabla 10: Resultados tunning de parámetros en H2O

En las pruebas realizadas con los modelos de H2O destacó su capacidad para procesar conjuntos de datos de mayor número de registros sin presentar problemas de memoria o tiempos de proceso elevados. (10x registros en el subconjunto de entrenamiento respecto al resto de pruebas)

7.3.3 MLR

Utilizando el *framework* MLR se realizaron pruebas de *benchmarking* y *tunning* con los modelos **bartMachine** (*Bayesian Additive Regression Trees*), **ada** (*AdaBoost*), **earth** (*Discriminant Analysis*), **C50** (*Tree*), **ctree** (*Clasification Tree*), **binomial** (*Logistic Regression*), **Ranger** (*Random Forest*), **gbm** (*Gradient Boost Machines*), **nnet** (*Neural Nets*) y **Naive Bayes**. Usando un conjunto de datos de entrenamiento con 1228 observaciones y otro de test de 818, se obtuvieron los valores para el indicador TEST AUC indicados en la tabla 11 y las curvas ROC de la figura 39.

MODEL	TEST AUC
bartMachine	0.8180123
ada	0.7970333
earth	0.7696733
C50	0.6788047
ctree	0.7191328
binomial	0.7974493
Ranger	0.8040362
gbm	0.7122210
nnet	0.7188201
Naive Bayes	0.7709142

Tabla 11: AUC para el subconjunto de test. Modelos MLR.

En las pruebas en MLR con los modelos indicados, **bartMachine** obtuvo los mejores valores de la métrica AUC, pero presentó problemas técnicos de compatibilidad de versiones de Java, uso excesivo de memoria y otros problemas y errores que motivaron su descarte de la selección de modelos.

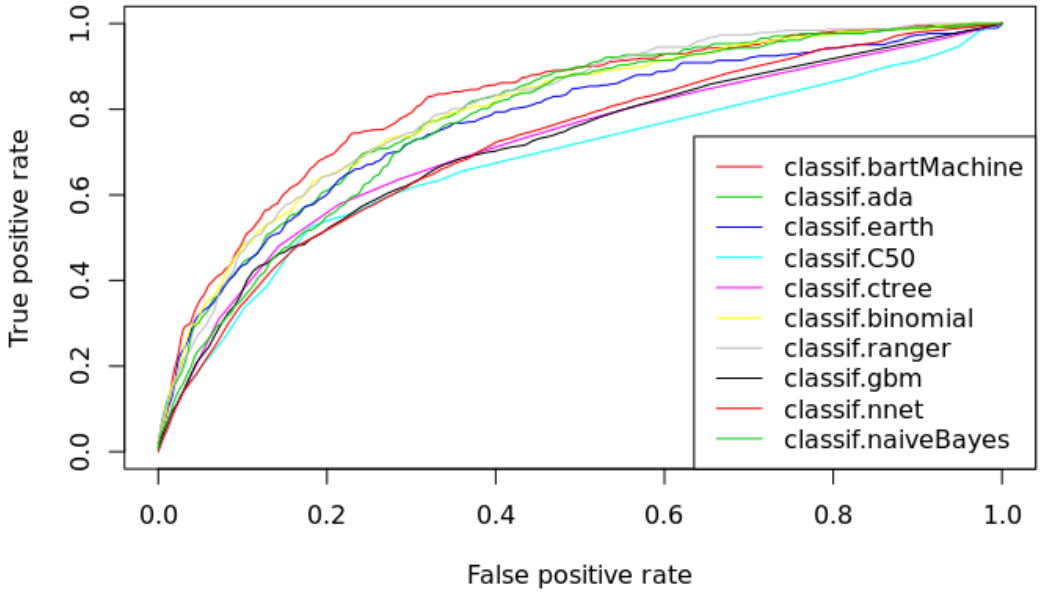

Figura 39: Curvas ROC modelos MLR.

7.3.4 Framework/Modelos seleccionados

En las diferentes pruebas de selección de modelos realizadas, MLR se ha mostrado una opción más sólida y completa que CARET, tanto por la parametrización por defecto de los modelos, las funcionalidades más avanzadas de *tunning*, *benchmarking* y visualización, como por una mayor consistencia desde el punto de vista estadístico (amplia variedad de métricas, test de hipótesis, visualización de datos) y su orientación a la realización de experimentos (combinación de resultados parciales, definición de tareas con diferentes conjuntos de datos).

En cuanto a la selección de modelos, los disponibles en H2O se han mostrado más eficaces, rápidos y escalables que el resto. Permiten realizar el entrenamiento con un conjunto mayor de datos sin problemas y, en caso de ser necesario, se pueden crear fácilmente *clusters* con nodos adicionales para ampliar la capacidad de proceso y memoria.

Teniendo en cuenta estas consideraciones, se ha optado por combinar los modelos de H2O, que obtienen de media mejores indicadores AUC y tienen capacidad para procesar todo el conjunto de datos, con las funcionalidades avanzadas para realizar *benchmarking* y *tunning* del *framework* MLR. Los modelos de H2O seleccionados finalmente fueron **c**lassif.h2o.glm, **classif.h2o.gbm**, **classif.h2o.randomForest**, **classif.h2o.deeplearning** y **classif.h2o.ensemble**.

Como principio general para la selección de modelos, no siempre es más adecuado guiarse sólo por los resultados de las métricas obtenidas, como se indica en M. Kuhn and K. Johnson [89], para elegir los modelos se considerará el modelo más simple que razonablemente explique los

datos. En este caso también se han tenido en cuenta criterios de estabilidad técnica, escalabilidad y homogeneidad en el uso de los modelos para facilitar los experimentos posteriores y su uso en entornos reales.

En la tabla 12 se muestra, como resultado de las pruebas realizadas, una comparativa de los *frameworks* CARET, MLR, la aplicación H2O y el uso de los modelos directamente desde R.

	CARET	**MLR**
PROS	+200 models Documentation, Books Simple and structured First step for learning Everybody uses it Preprocessing, partition and imputation Simple	+80 models Stable / Fast Good default model parametrization Benchmarks/ Tuning/Feature selection Statistical experiments Visualization Superset of Caret
CONS	Problems with big datasets Parallelism problems Not all models 'work'	Less documentation Steepest learning curve Complex Lack of some functions Less known than CARET
	R MODELS	**H2O**
PROS	All models are available in R Flexibility	Very fast Native parallel architecture Good documentation Scalable / Stable
CONS	Not all models work Parallelism problems Need frameworks to simplify the workflow	Only 4 models + stack Need a framework to simplify the workflow

Tabla 12: Comparativa de frameworks.

Con la selección de *frameworks* y de modelos realizada, el siguiente paso es optimizar la parametrización de dichos modelos antes de aplicarlos a los conjuntos de datos.

7.4 Tunning de modelos seleccionados

Utilizando el *framework* MLR se realizó la optimización automática de los parámetros de los modelos utilizando el algoritmo *iRace* [108], tomando como punto de partida el conjunto de parámetros empleado en las anteriores pruebas. En MLR la métrica utilizada para realizar la optimización de parámetros es MMCE (*mean misclassification error*) [122].

MLR permite realizar el *tunning* de parámetros para todos los modelos en bloque, aunque en este caso la complejidad de la parametrización y los resultados obtenidos con el modelo **classif.h2o.deeplearning** al realizar

el *tunning* conjunto llevaron a optimizar de forma separada este modelo, utilizando un conjunto mayor de datos.

La tabla 13 muestra los resultados de la mejor parametrización obtenida para los modelos **classif.h2o.glm**, **classif.h2o.gbm** y **classif.h2o.randomForest**.

Stratified partition 1229 obs tunning, 20445 obs test, 3-CV, 300 experimentos max. Irace tunning		
MODEL	PARAMS. (Best Model)	TEST MMCE
classif.h2o.glm	alpha=0.7473238	0.1640988
classif.h2o.gbm	Ntrees=1255 max_depth=6 learn_rate=0.007709469	0.1620936
classif.h2o.randomForest	Mtries=14 Ntrees=1763 max_depth=5	0.1802884

Tabla 13: Tunning modelos H2O - MLR - glm/gbm/randomForest.

Las gráficas *boxplot* del error MMCE y las curvas ROC obtenidas para estos modelos se encuentran en las figuras 40 y 41.

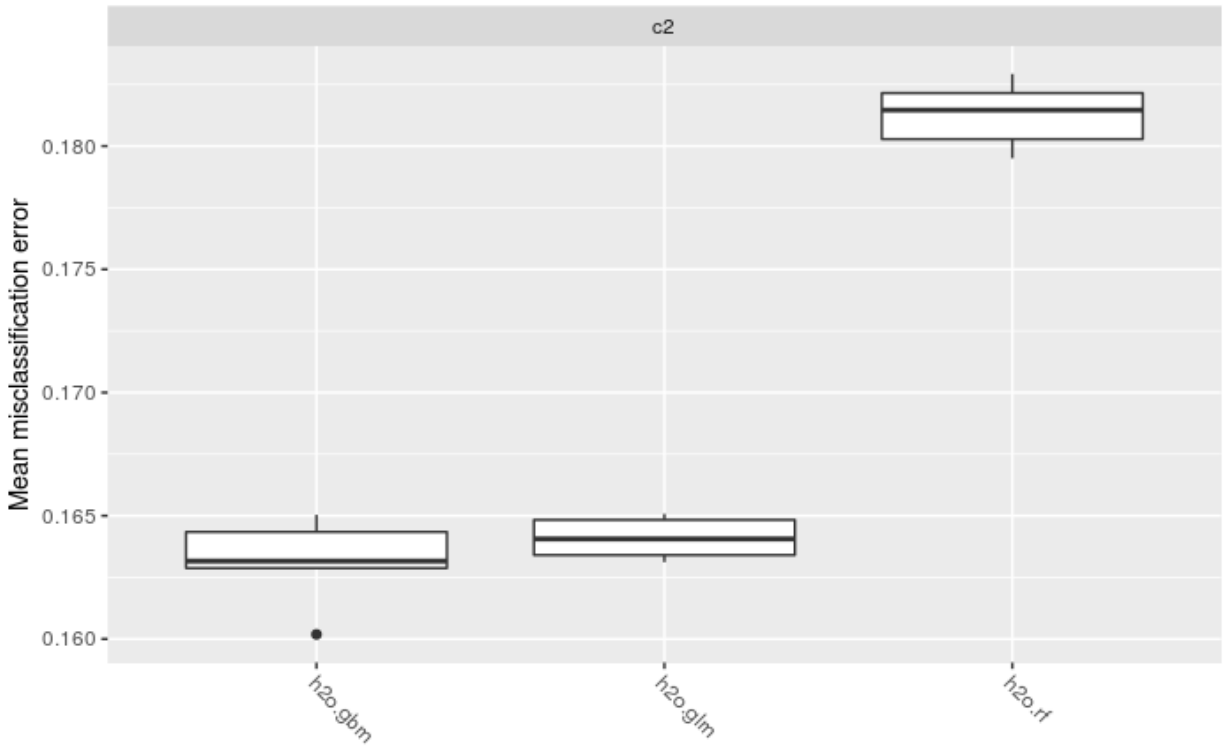

Figura 40: Mean misclassification error glm/gbm/randomForest.

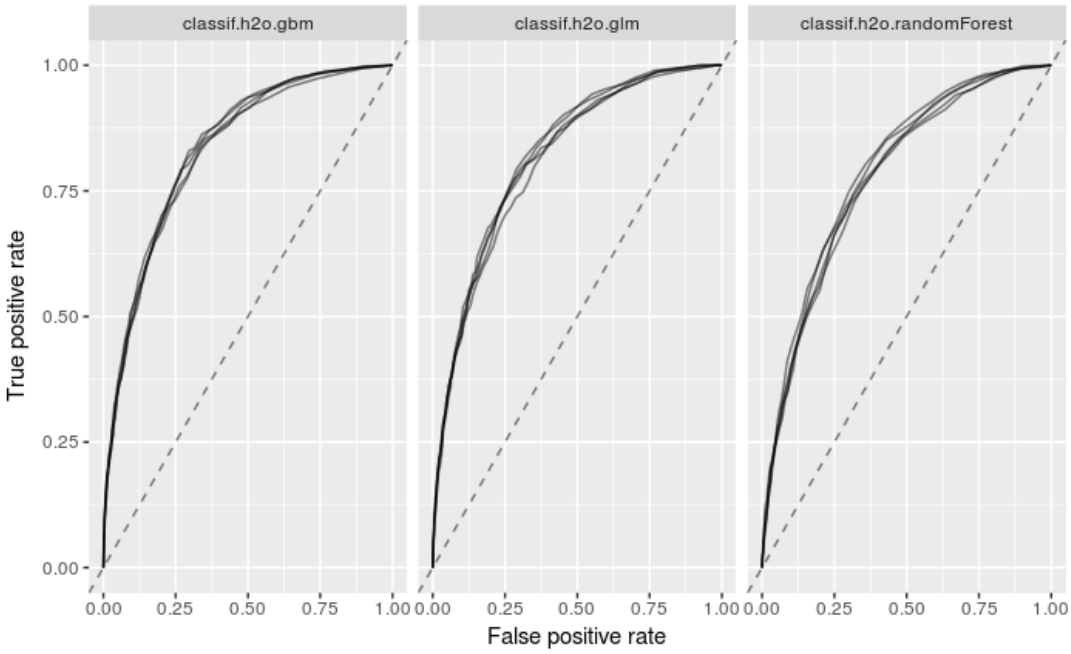

Figura 41: Curvas ROC - glm/gbm/randomForest.

Los resultados de la mejor parametrización obtenida para el modelo **classif.h2o.deepLearning** se encuentran en la tabla 14, y en la figura 42 se muestran las gráficas de *boxplot* y la curva ROC para este modelo.

Stratified partition 4090 obs tunning, 20445 obs test, 2-CV, 200 experimentos max. Irace tunning.		
MODEL	PARAMS. (Best Model)	TEST MMCE
classif.h2o.deeplearning	hidden=c(1000,1000) epochs=72 / rho=0.9615025 epsilon=1.883177e-07 / balance_classes=TRUE shuffle_training_data=TRUE fast_mode=TRUE / l1=1e-6	0.2017093

Tabla 14: Tunning modelo H2O – MLR – deeplearning.

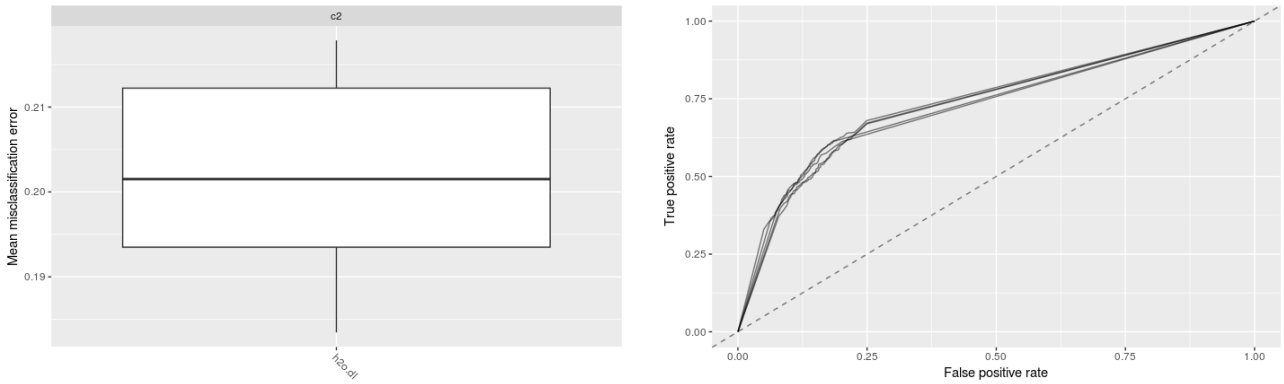

Figura 42: Mean misclassification error y curva ROC para modelo deeplearning.

7.5 Benchmarking

Con los parámetros obtenidos y los modelos seleccionados se realizó la comparativa de modelos, siguiendo las recomendaciones [89] (pag. 78) de utilizar para conjuntos de datos grandes 10-CV y no mezclar dicha fase de *benchmarking* con el *tunning* de parámetros. Para realizar el benchmarking se utilizaron las funcionalidades específicas de MLR [123] para este tipo de tareas y los modelos **classif.h2o.glm, classif.h2o.gbm, classif.h2o.randomForest** y **classif.h2o.deeplearning**.

Se realizaron dos comparativas de modelos, una con el conjunto completo de datos y otra dividiéndolo en grupos según una serie de variables, con el objeto de comprobar el comportamiento de los modelos en ambos casos y hasta qué punto la parametrización realizada seguía siendo válida en los diferentes grupos.

7.5.1 Conjunto completo

Para el *benchmark* del conjunto completo de datos se obtuvieron las métricas AUC y MMCE de la tabla 15 y las gráficas de violín con la distribución de los indicadores para las iteraciones de la validación cruzada en las figuras 43 y 44.

20445 obs test, 79 vars, 10-CV			
MODEL	MEAN AUC	MEAN MMCE	TIME
gbm	0.8353075	0.1636585	106.2498
glm	0.8112075	0.1659573	3.5877
randomForest	0.7847361	0.1821473	61.9151
deeplearning	0.7850761	0.1845442	602.1327

Tabla 15: Benchmarking modelos MLR- H20.

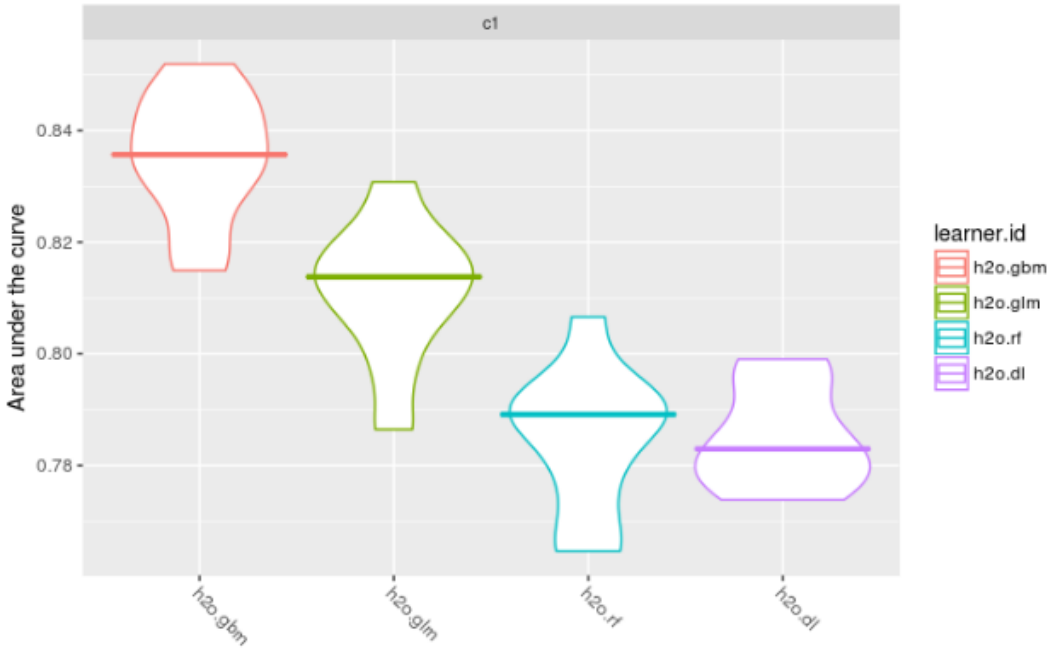

Figura 43: Benchmarking de modelos H2O. AUC.

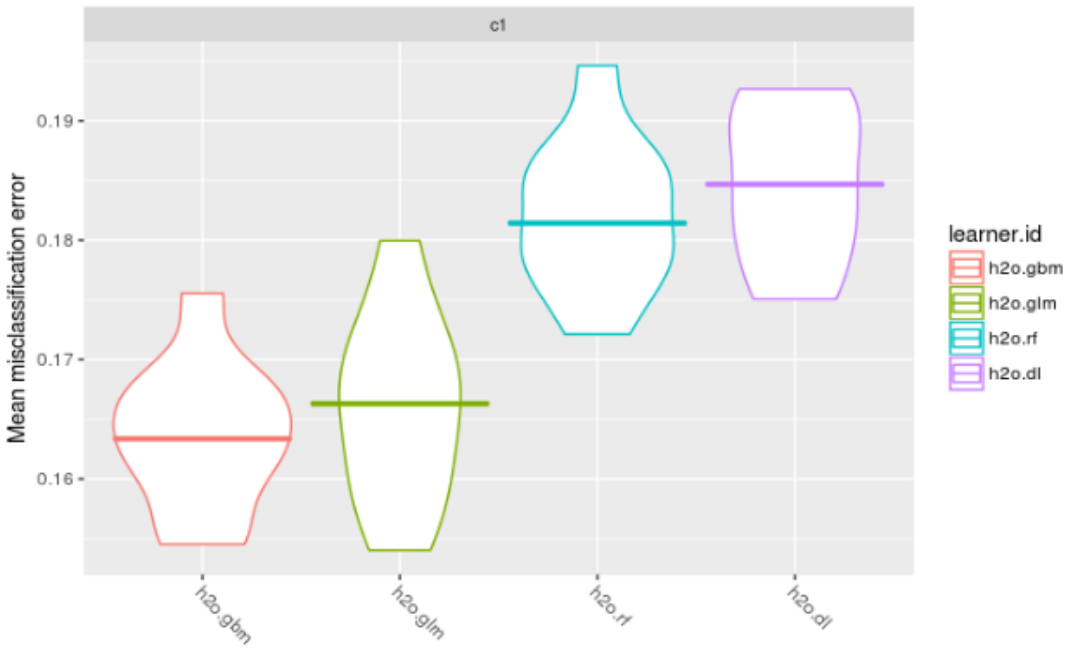

Figura 44: Benchmarking de modelos H2O. MMCE.

La figura 45 muestra las curvas ROC con los intervalos de confianza de 95% para los resultados de las validaciones cruzadas de cada modelo.

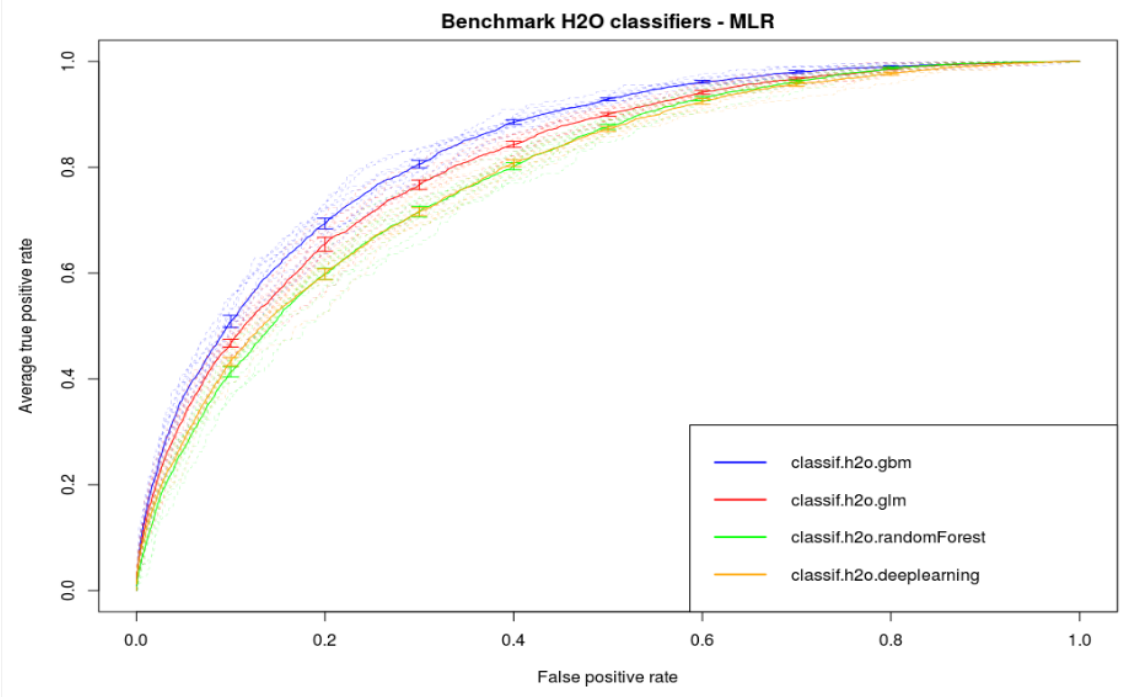

Figura 45: Benchmarking de modelos H2O. Curvas ROC con CI de 0.95 para las mediciones de cada CV

Con el conjunto completo de datos el mejor modelo es el **gbm**, seguido del **glm,** superando al resultado del *score* tomado como referencia (SAPSII AUC 0.725). Los resultados de **deepLearning** y **randomForest**[10]

[10] En lo sucesivo se entiende que los modelos utilizados son siempre los de H2O y se obvia el prefijo classif.h2o usado cuando se les llama desde MLR.

son comparables, siendo el tiempo de proceso del modelo de redes neuronales diez veces mayor.

7.5.2 Grupos

Para realizar el *benchmarking* por grupos se dividió el conjunto de datos por las variables de interés mostradas en la tabla 16 junto con el número de observaciones de cada grupo:

GROUP	VARIABLE	N ROWS
Estancia en UCI > 3 días	icu_stay_more72	14430
Estancia en UCI <= 3 días	icu_stay_less72	6015
Score SOFA al ingreso >=4	score_sofa_more4	13230
Score SOFA al ingreso <4	score_sofa_less4	7215
Diabético	diag_has_diabetes_yes	7815
No Diabético	diag_has_diabetes_no	12630
Tratamiento insulina IV	icu_has_admiv_insulin_yes	6912
Sin tratamiento insulina IV	icu_has_admiv_insulin_no	13533
Sepsis al ingreso	med_has_sepsis_yes	9266
Sin sepsis al ingreso	med_has_sepsis_no	11179

Tabla 16: Agrupación utilizada para benchmarking de modelos.

Los resultados obtenidos para el benchmarking con validación cruzada 10-CV se encuentran en la tabla 17.

MODEL	GROUP	MEAN AUC	MEAN MMCE	TIME (ms)
gbm	icu_diabetes_no	0.8287795	0.17205067	71.4253
glm	icu_diabetes_no	0.8091781	0.17260491	2.9320
randomForest	icu_diabetes_no	0.7841264	0.18978622	45.5295
deeplearning	icu_diabetes_no	0.7827990	0.20071259	346.9321
gbm	icu_diabetes_yes	0.8162047	0.15853667	55.4785
glm	icu_diabetes_yes	0.8126025	0.15482659	2.5772
randomForest	icu_diabetes_yes	0.7914515	0.17120650	34.4226
deeplearning	icu_diabetes_yes	0.7803536	0.17350862	201.0140
gbm	icu_insulineiv_no	0.8042925	0.17704936	77.2010
glm	icu_insulineiv_no	0.7920866	0.17860114	3.1316
randomForest	icu_insulineiv_no	0.7664846	0.19411936	47.9091
deeplearning	icu_insulineiv_no	0.7458940	0.20342971	358.0062
gbm	icu_insulineiv_yes	0.8681353	0.14829329	51.5943
glm	icu_insulineiv_yes	0.8587362	0.14221954	2.4964
randomForest	icu_insulineiv_yes	0.8508606	0.15943573	32.0305
deeplearning	icu_insulineiv_yes	0.8349064	0.15958086	166.1068
gbm	icu_sepsis_no	0.8436368	0.10823899	71.4600
glm	icu_sepsis_no	0.8147637	0.10859669	2.8170
randomForest	icu_sepsis_no	0.8139742	0.11450065	43.7694
deeplearning	icu_sepsis_no	0.7957100	0.12147924	280.9742
gbm	icu_sepsis_yes	0.7791926	0.23257017	57.4769
glm	icu_sepsis_yes	0.7736799	0.22987248	2.6845
randomForest	icu_sepsis_yes	0.7419123	0.25944290	35.4301
deeplearning	icu_sepsis_yes	0.7160929	0.27574015	267.5957

gbm	icu_sofa_score_less4	0.8143393	0.11074108	52.1830
glm	icu_sofa_score_less4	0.8064948	0.10963151	2.5035
randomForest	icu_sofa_score_less4	0.8029356	0.11198877	32.6785
deeplearning	icu_sofa_score_less4	0.7712846	0.12765146	137.9767
gbm	icu_sofa_score_more4	0.8154025	0.19546485	72.2360
glm	icu_sofa_score_more4	0.7962623	0.19599395	2.9860
randomForest	icu_sofa_score_more4	0.7702484	0.21814059	45.7557
deeplearning	icu_sofa_score_more4	0.7628698	0.22547241	357.0396
gbm	icu_stay_less72	0.8530945	0.09626426	46.2874
glm	icu_stay_less72	0.8468536	0.09227395	2.4033
randomForest	icu_stay_less72	0.8366129	0.10906407	28.8654
deeplearning	icu_stay_less72	0.8207605	0.10823683	125.1738
gbm	icu_stay_more72	0.8065761	0.19306999	77.7253
glm	icu_stay_more72	0.7902540	0.19189189	3.1919
randomForest	icu_stay_more72	0.7667606	0.20990991	48.4478
deeplearning	icu_stay_more72	0.7436908	0.22571033	446.4488

Tabla 17: Benchmarking modelos MLR- H20. Por grupos.

En la figura 46 se muestra la media del indicador AUC agregando los valores de las todas las validaciones cruzadas.

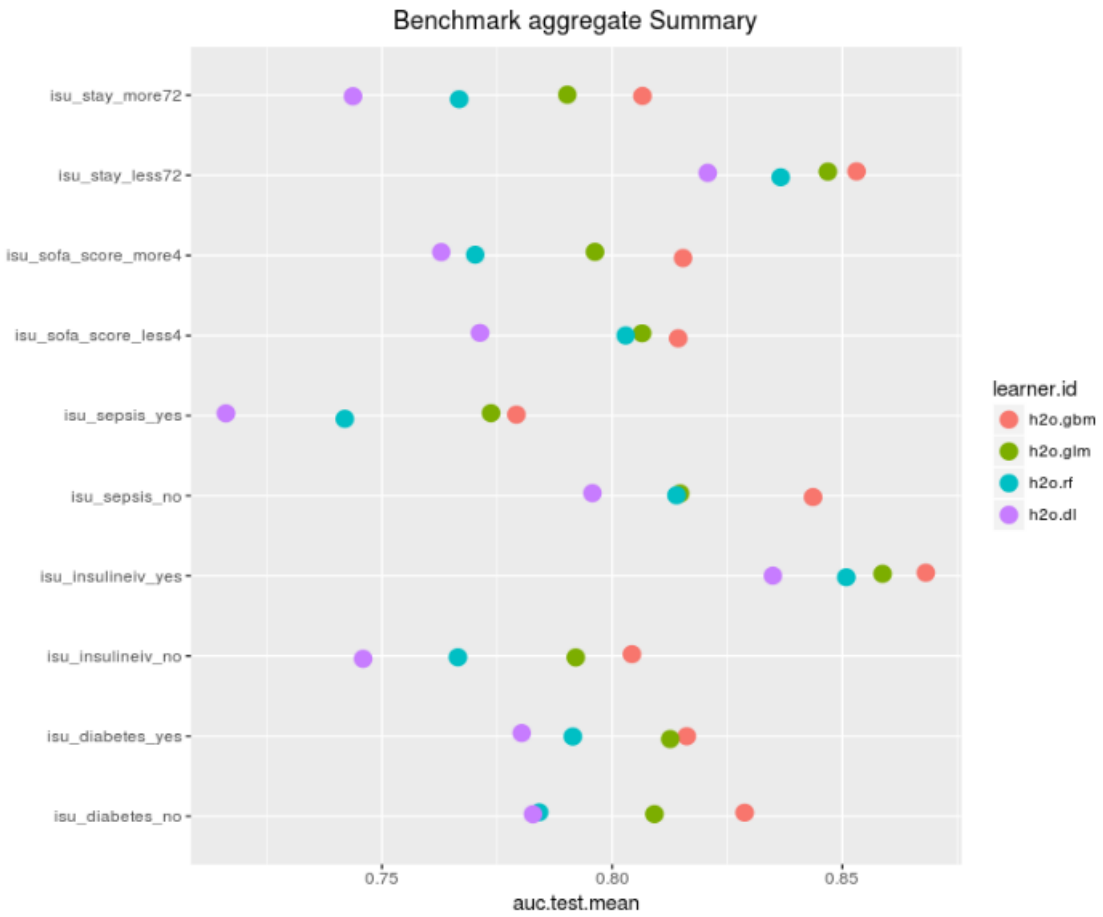

Figura 46: Media de AUC para las medidas agregadas.

Y en la figura 47 los resultados para cada uno de los grupos y modelos de los indicadores de MMCE y AUC.

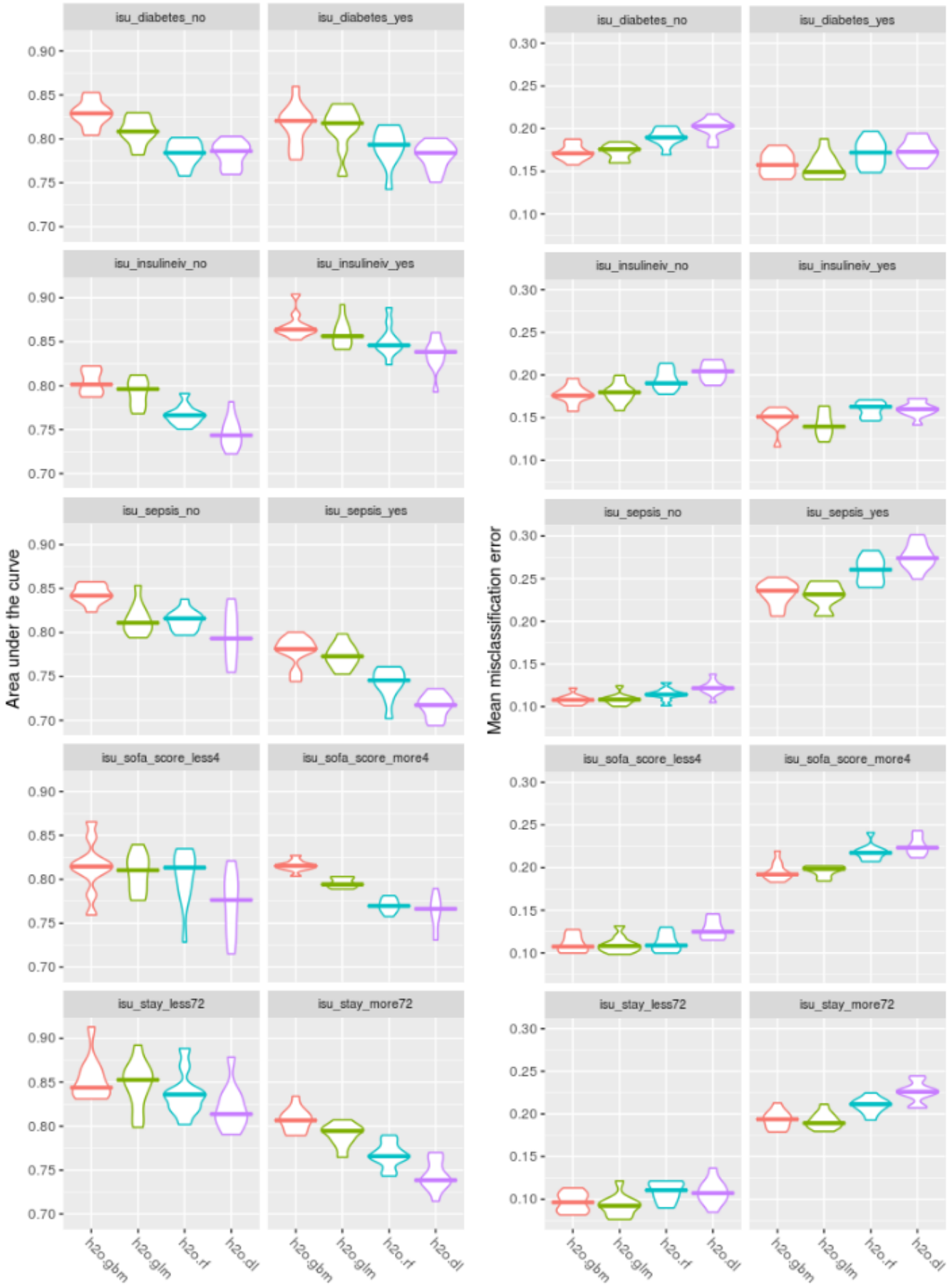

Figura 47: AUC y MMCE por grupos y modelo.

Observando las gráficas es claro que el modelo que mejor se comporta en la práctica totalidad de grupos, tanto para el indicador AUC, como para el MMCE, es **gbm**, seguido de **glm**.

En la gráfica 48 se puede observar el rango de cada modelo, obtenido a partir de las medidas agregadas de las validaciones cruzadas. Un menor rango está asociado a un mejor rendimiento. Para la métrica AUC el rango de los diferentes modelos es el mismo para todos los grupos, mientras que para el MMCE se van alternando entre **gbm** y **glm**.

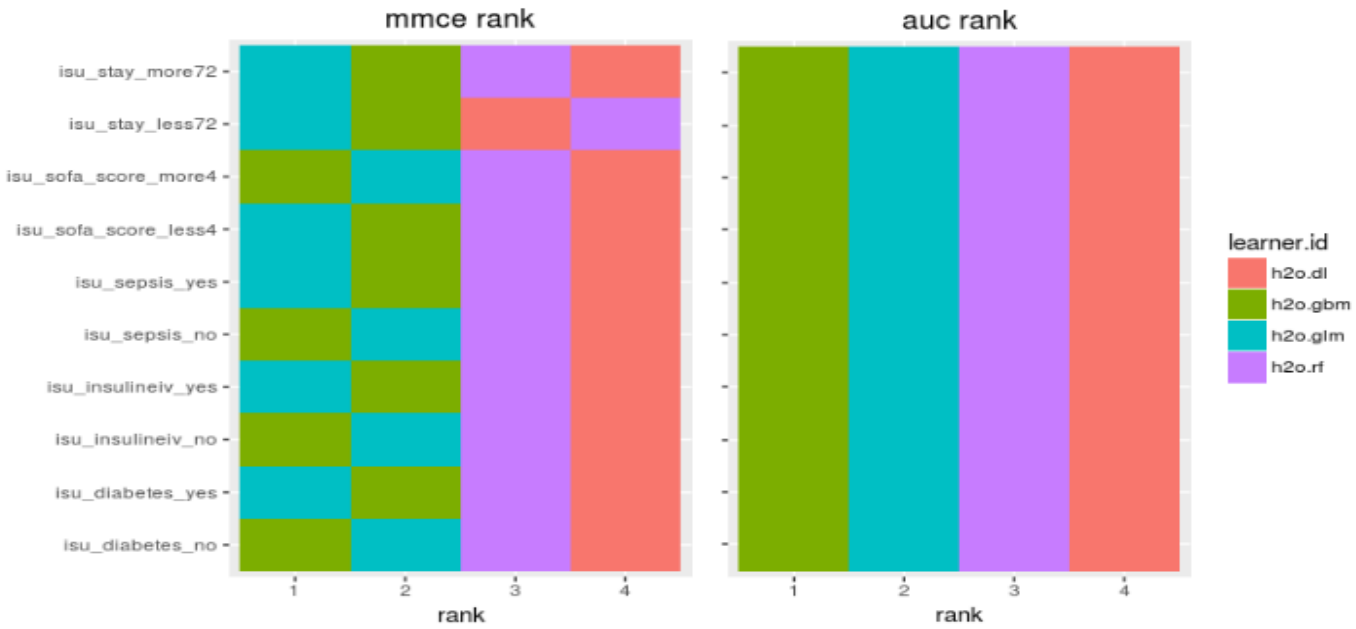

Figura 48: Rank de los modelos para AUC y MMCE.

MLR dispone de varios test de hipótesis estadísticos que permiten comparar la equivalencia o no de los diferentes modelos para los grupos utilizados. Aplicando el test de hipótesis paramétrico de Friedman, equivalente al ANOVA para medidas repetidas [124], se rechaza la hipótesis nula de equivalencia de rango entre los distintos modelos (*p-value* de 1.38e-06 para AUC y 9.355e-06 para MMCE). Para las medidas agregadas de MMCE el test post hoc de Friedman-Nemenyi muestra claramente un comportamiento similar de los pares de modelos **gbm-glm** y **deeplearning-randomForest** (p-value<0.05):

```
        Pairwise comparisons using Nemenyi multiple comparison test
            with q approximation for unreplicated blocked data

data:  mmce.test.mean and learner.id and task.id

                          classif.h2o.gbm classif.h2o.glm classif.h2o.randomForest
classif.h2o.glm           0.98572         -               -
classif.h2o.randomForest  0.04627         0.01706         -
classif.h2o.deeplearning  0.00039         8.8e-05         0.50835
```

El mismo test aplicado a la métrica AUC, no muestra esa separación en grupos tan clara.

```
        Pairwise comparisons using Nemenyi multiple comparison test
            with q approximation for unreplicated blocked data

data:  auc.test.mean and learner.id and task.id

                          classif.h2o.gbm classif.h2o.glm classif.h2o.randomForest
classif.h2o.glm           0.307           -               -
classif.h2o.randomForest  0.003           0.307           -
classif.h2o.deeplearning  1.2e-06         0.003           0.307
```

En el *framework* MLR también se pueden obtener los diagramas de diferencias críticas usando el test de Bonferroni-Dunn, que realiza una comparación por pares de modelos contra uno tomado como base [124]. En este caso indicamos como base el modelo **gbm** y obtenemos los gráficos de diferencias críticas para las métricas de AUC y MMCE. El valor de la diferencia crítica se calcula mediante la fórmula:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}},$$

donde N es el número de grupos de datos en este caso, *k* el número de modelos y q_α proviene del rango estadístico de student dividido por $\sqrt{2}$ [124].

El resultado se encuentra en las gráficas 49 y 50 para los indicadores AUC y MMCE.

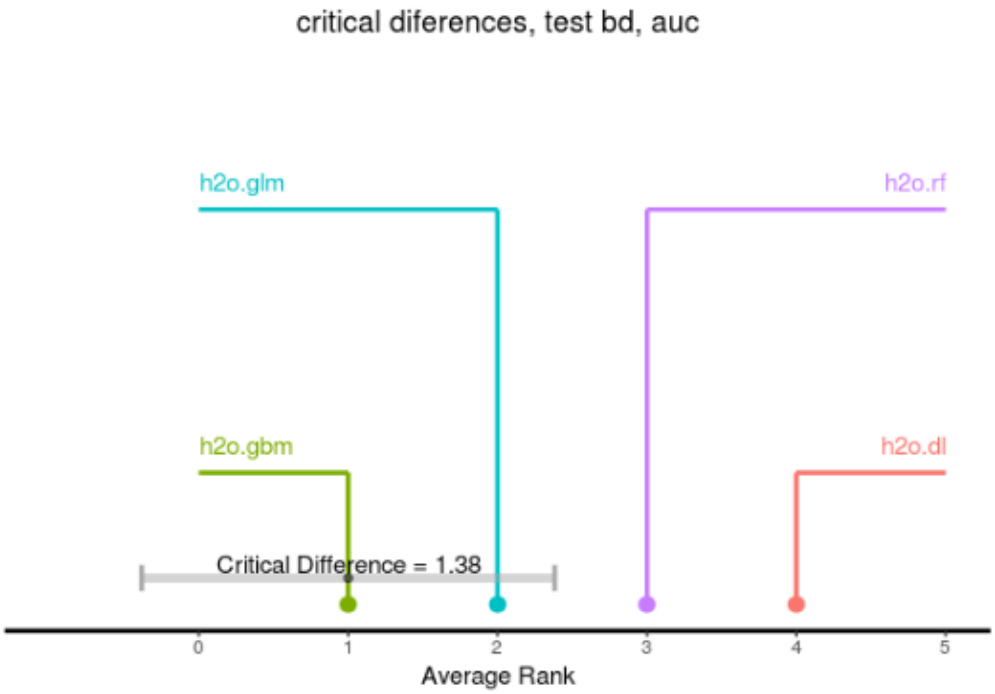

Figura 49: Gráfica de diferencias críticas. AUC.

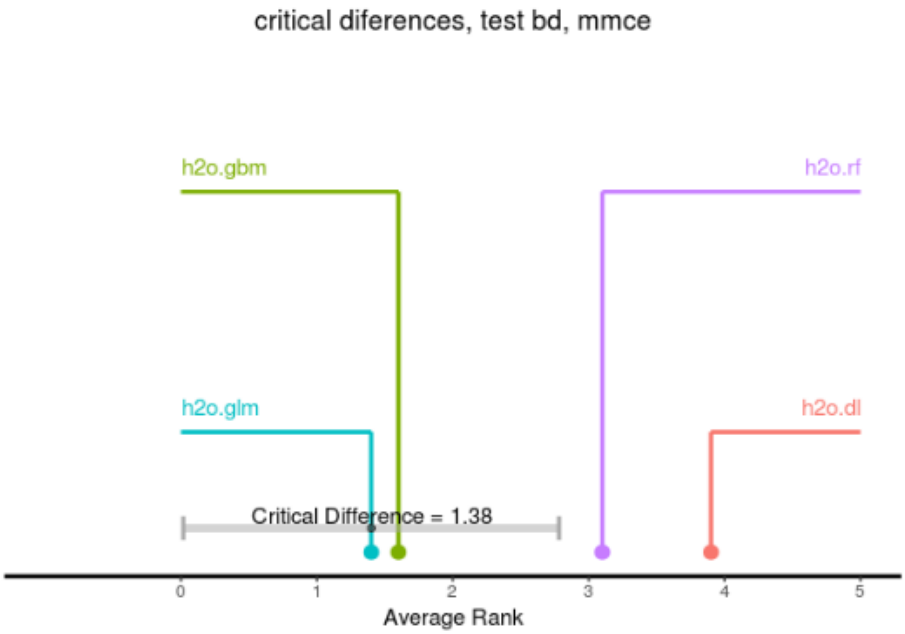

Figura 50: Gráfica de diferencias críticas. MMCE.

En estas gráficas de diferencias críticas, se observa que los modelos **gbm** y **glm** son estadísticamente equivalentes (ambos están dentro de la barra de diferencia critica), siendo para la métrica AUC el **gbm** el que tiene un rango más bajo y, por tanto, el mejor modelo, y el **glm** para la métrica MMCE.

Como conclusión, los modelos seleccionados como más adecuados para el conjunto de datos obtenidos son **gbm** y **glm**, en ese orden, si consideramos como objetivo la métrica AUC, tanto para el conjunto completo de datos como para los diferentes grupos obtenidos.

7.6 Aplicación de modelos y evaluación

Siguiendo la metodología definida[11] y una vez realizadas las tareas incluidas en la fase de selección de plataforma y herramientas, la selección de los modelos más adecuados, realizado el *tunning* de parámetros y evaluado el rendimiento de los modelos, la siguiente fase en la metodología es la aplicación de los modelos para obtener conocimiento del conjunto de datos.

Uno de los objetivos adicionales del estudio es el análisis de la influencia de las variables de comorbilidades en la capacidad predictiva de los modelos. Para estudiar esta influencia se han aplicado los modelos **gbm** y **glm**, tanto al conjunto completo de datos como a los grupos del apartado anterior, obteniendo las métricas AUC resultado de incluir o excluir las variables de comorbilidades en bloque.

Las gráficas de importancia de las variables para cada uno de los modelos permiten estudiar su influencia en el resultado final (Figuras 51-54).

[11] Ver apartado 2.3

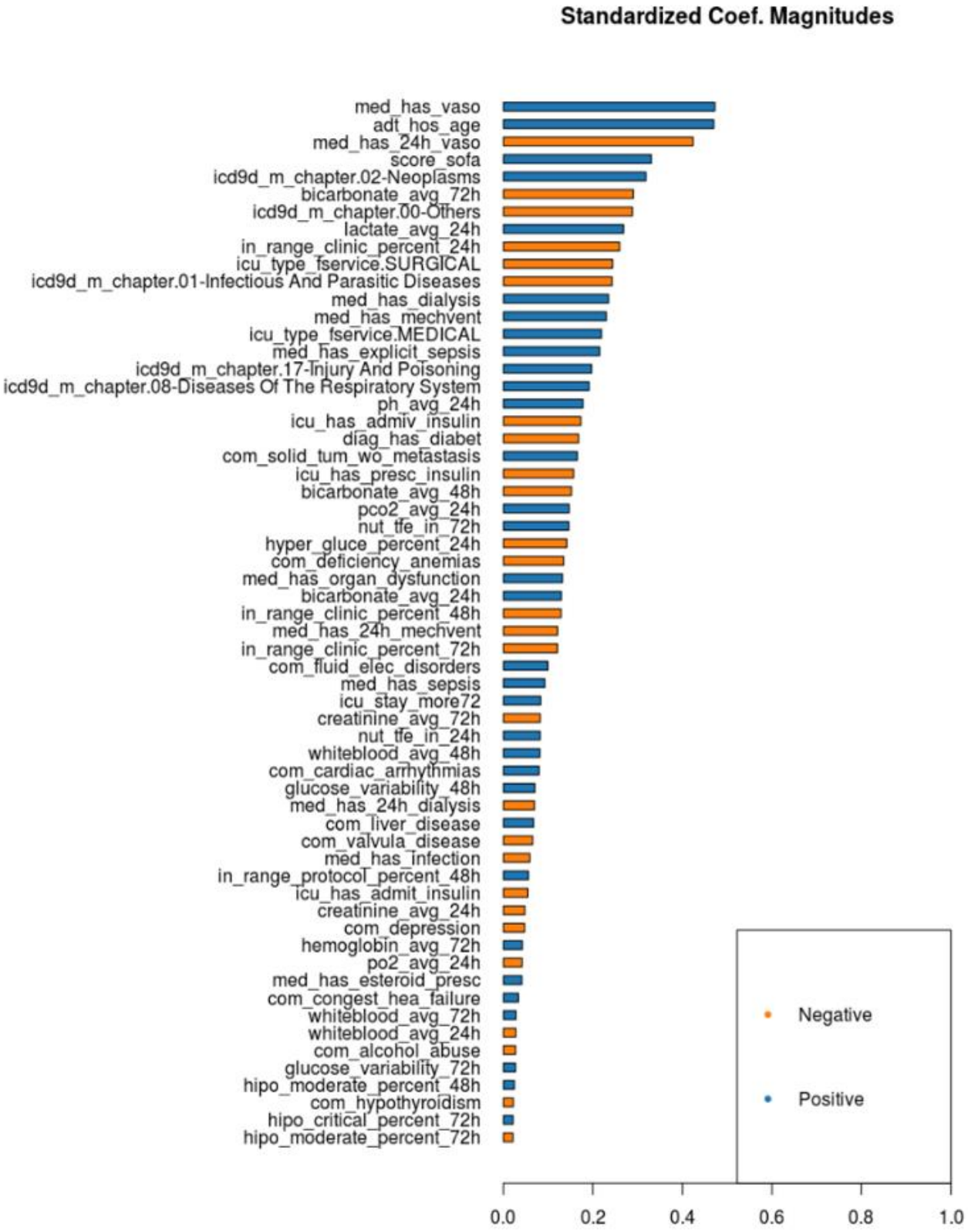

Figura 51: Importancia de variables para el modelo glm con comorbilidades (top 60).

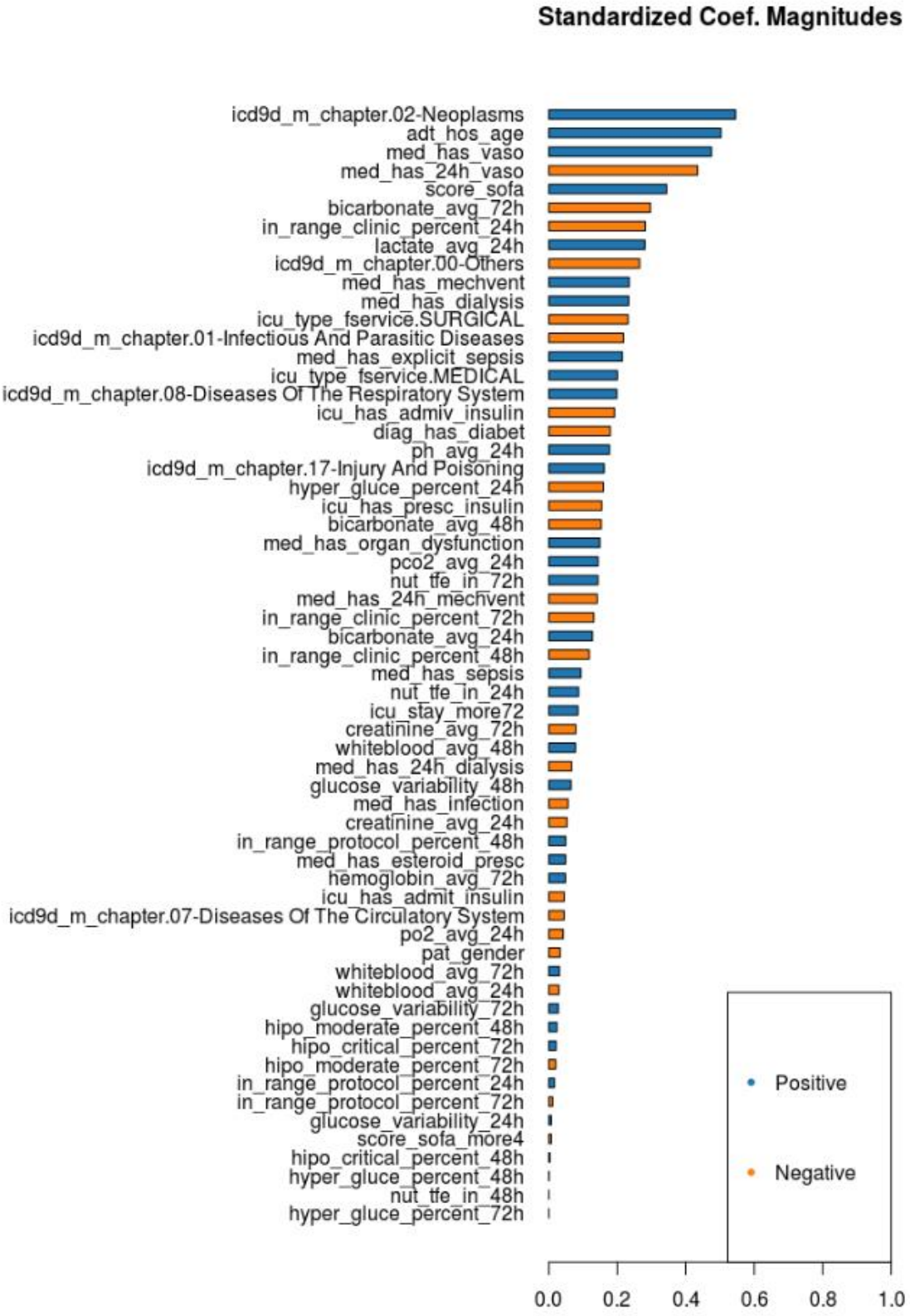

Figura 52: Importancia de variables para el modelo glm sin comorbilidades (top 60).

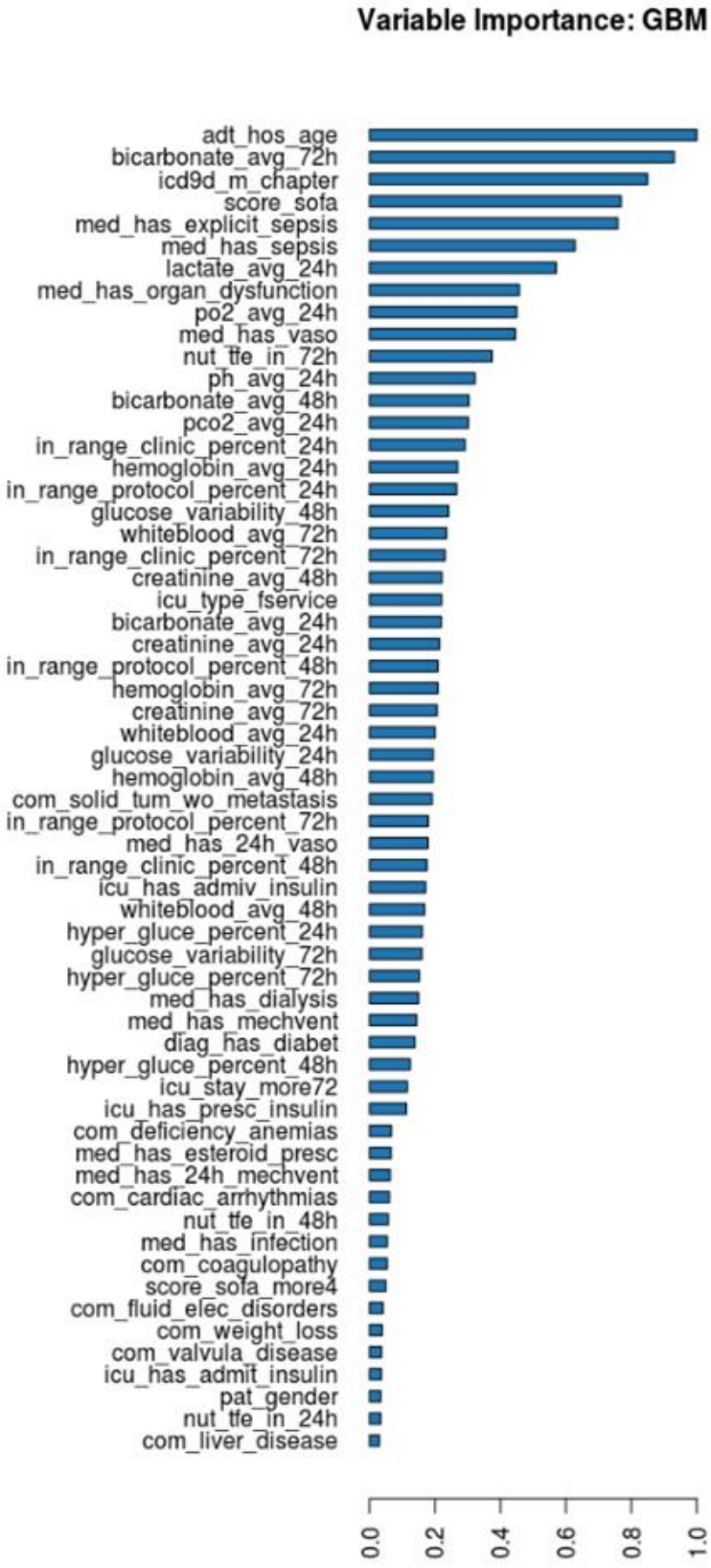

Figura 53: Importancia de variables para el modelo gbm con comorbilidades (top 60).

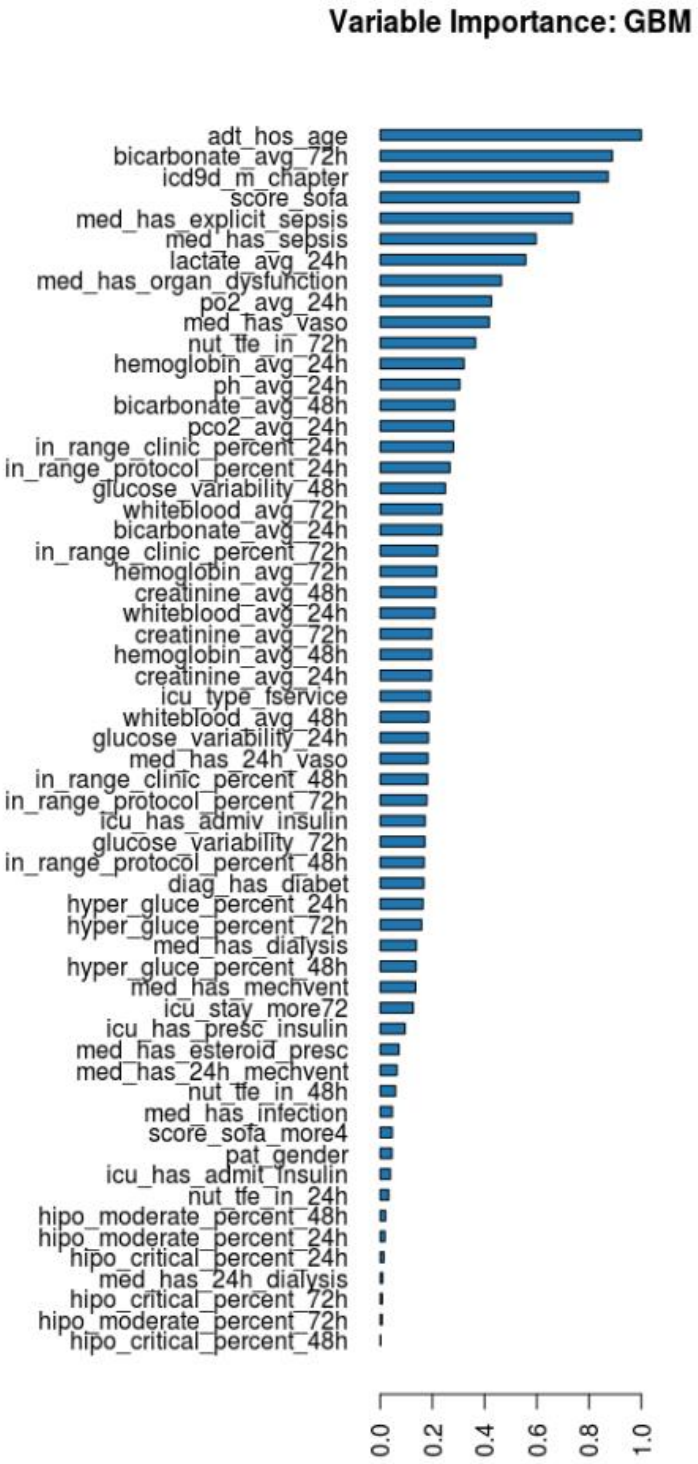

Figura 54: Importancia de variables para el modelo gbm sin comorbilidades.

Del análisis de la importancia de las variables en ambos modelos, se observa que en ambos modelos la variable relacionada con la glucosa con más importancia es **in_range_clinic_percent_24h.** En el modelo **glm**, tanto incluyendo como no incluyendo comorbilidades, el resto de las variables relacionadas con la glucosa tienen una importancia media o

baja, mientras que en los modelos **gbm** están casi todas en la zona media, salvo las de hipoglucemia, que son las que menos importancia tienen (o no aparecen) en todos los casos. Los detalles de las variables relacionadas con la glucosa se encuentran en los anexos[12].

Los resultados de las métricas AUC para el conjunto completo de datos se muestran en la tabla 18, donde la ganancia en el indicador CV AUC usando las variables de comorbilidades es de 0.364583% para el modelo **gbm** y de 0.224546% para el modelo **glm** (0,640864% y 0,653299% respectivamente, para TEST AUC)

Stratified partition 12268 obs train, 8177 obs test (60/40 train/test). 10-CV 79 vars/60 vars (with/without comorbities).

MDL	COMORBIDITIES	TRAIN AUC	CV AUC	TEST AUC
glm	TRUE	0.8173366	0.8083730	0.8105587
glm	FALSE	0.8122919	0.8065619	0.8052977
gbm	TRUE	0.9805331	0.8236553	0.8286023
gbm	FALSE	0.9776612	0.8206633	0.8233259

Tabla 18: Análisis de influencia de comorbilidades. Conjunto completo.

Aplicando los modelos **gbm** y **glm** al conjunto de datos dividido en los mismos grupos definidos anteriormente y evaluando las métricas AUC resultado de incluir o no las variables de comorbilidades en bloque se obtiene la tabla 19. La ganancia media del indicador CV AUC para el modelo **gbm** obtenida al incluir en el conjunto de datos las variables de comorbilidad es de 0.771106% ±0.320137 95% CI y para el modelo **glm** de 0.259880% ±0.242234 95% CI (0.620988% ±0.349801 y 0.420361% ±0.283152 respectivamente, para TEST AUC).

Stratified partition, 60/40 train/test. 10-CV 79 vars/60 vars (with / without comorbities).

MDL	GROUP	COMORB	TRAIN AUC	CV AUC	TEST AUC
glm	icu_stay_more72	TRUE	0.805989608	0.794142885	0.780706437
glm	icu_stay_more72	FALSE	0.800388147	0.790687428	0.777082199
gbm	icu_stay_more72	TRUE	0.987188177	0.801418040	0.790410241
gbm	icu_stay_more72	FALSE	0.985728000	0.796784781	0.784204127
glm	icu_stay_less72	TRUE	0.877347757	0.845441140	0.839426678
glm	icu_stay_less72	FALSE	0.869744845	0.845991528	0.828260557
gbm	icu_stay_less72	TRUE	1.000000000	0.855254489	0.828998841
gbm	icu_stay_less72	FALSE	1.000000000	0.847073058	0.822768530
glm	icu_sofa_score_more4	TRUE	0.802185840	0.789444846	0.798479559
glm	icu_sofa_score_more4	FALSE	0.796597766	0.786805219	0.797858278
gbm	icu_sofa_score_more4	TRUE	0.990322648	0.804806707	0.807873161
gbm	icu_sofa_score_more4	FALSE	0.986311411	0.798469988	0.810393192
glm	icu_sofa_score_less4	TRUE	0.826331693	0.787077876	0.808529279

[12] glucose_variability_24h glucose_variability_48h hipo_critical_percent_24h hipo_moderate_percent_24h hyper_gluce_percent_24h in_ range_clinic_percent_24h in_range_protocol_percent_24h hipo_critical_percent_48h hipo_moderate_percent_48h hyper_gluce_percent_48h in_range_clinic_percent_48h in_range_protocol_percent_48h hipo_critical_percent_72h hipo_moderate_percent_72h hyper_gluce_percent_72h in_range_clinic_percent_72h in_range_protocol_percent_72h glucose_variability_72h

glm	icu_sofa_score_less4	FALSE	0.821085625	0.790199885	0.804932413
gbm	icu_sofa_score_less4	TRUE	0.999995669	0.766770340	0.814220339
gbm	icu_sofa_score_less4	FALSE	0.999995533	0.752827209	0.807001834
glm	icu_sepsis_yes	TRUE	0.788996593	0.770768910	0.768339493
glm	icu_sepsis_yes	FALSE	0.780229939	0.766369867	0.763512478
gbm	icu_sepsis_yes	TRUE	0.995482484	0.767045724	0.773928928
gbm	icu_sepsis_yes	FALSE	0.993173159	0.763929925	0.763922210
glm	icu_sepsis_no	TRUE	0.825251868	0.800461977	0.823651905
glm	icu_sepsis_no	FALSE	0.819618165	0.798804012	0.822783630
gbm	icu_sepsis_no	TRUE	0.998294909	0.819425154	0.834485113
gbm	icu_sepsis_no	FALSE	0.997852345	0.814500802	0.834393106
glm	icu_insulineiv_yes	TRUE	0.875139749	0.852089753	0.848100932
glm	icu_insulineiv_yes	FALSE	0.871069784	0.852892029	0.848762604
gbm	icu_insulineiv_yes	TRUE	0.999926874	0.857737002	0.849438682
gbm	icu_insulineiv_yes	FALSE	0.999893571	0.854079466	0.847251032
glm	icu_insulineiv_no	TRUE	0.799652228	0.785987834	0.796147282
glm	icu_insulineiv_no	FALSE	0.792910919	0.782430702	0.791751997
gbm	icu_insulineiv_no	TRUE	0.989394751	0.790283989	0.801735940
gbm	icu_insulineiv_no	FALSE	0.986370096	0.788017285	0.793799384
glm	icu_diabetes_yes	TRUE	0.828726499	0.803731799	0.810367947
glm	icu_diabetes_yes	FALSE	0.819404618	0.799302041	0.807856037
gbm	icu_diabetes_yes	TRUE	0.999776513	0.791761393	0.809854926
gbm	icu_diabetes_yes	FALSE	0.999803343	0.784002890	0.803384152
glm	icu_diabetes_no	TRUE	0.822191567	0.808460241	0.800714031
glm	icu_diabetes_no	FALSE	0.814948570	0.803717372	0.798012699
gbm	icu_diabetes_no	TRUE	0.993929645	0.822154916	0.808752338
gbm	icu_diabetes_no	FALSE	0.991931172	0.815629769	0.803183636

Tabla 19: Análisis de influencia de comorbilidades. Por grupos.

En la figura 55 se puede observar la gráfica de los datos de CV AUC con y sin comorbilidades para los distintos grupos y los modelos **gbm** y **glm**.

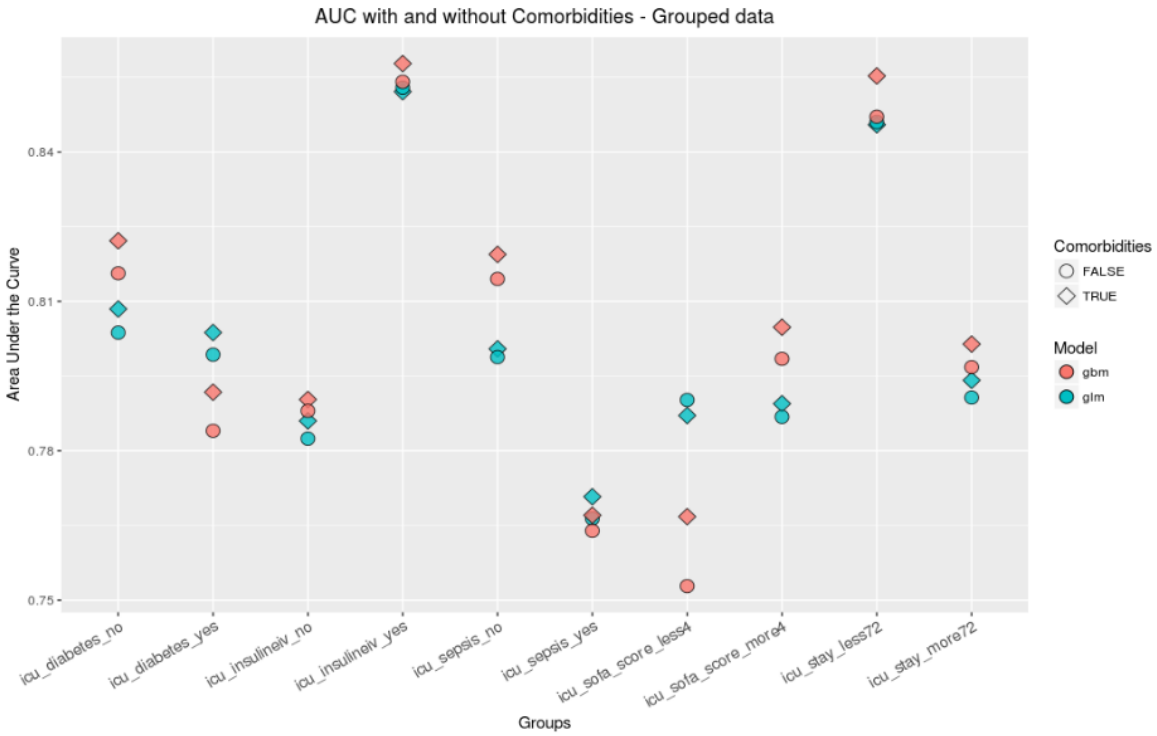

Figura 55: Comparativa de AUC (CV) con/sin campos de comorbilidades

En esta gráfica se observa que los grupos donde se obtiene mayor poder predictivo son los de pacientes tratados con insulina IV y los que tienen una estancia de menos de tres días en UCI (media de 0,851319808). El de menor poder predictivo es el **gbm** para el grupo de pacientes con un score SOFA menor de 4, con un valor AUC medio de 0,759798775, que sigue siendo un valor de AUC superior al mejor valor obtenido con los *scores* (SAPSII 0.725) [13]

En la figura 56 se muestran las curvas ROC para estos dos grupos, incluyendo las variables de comorbilidades,

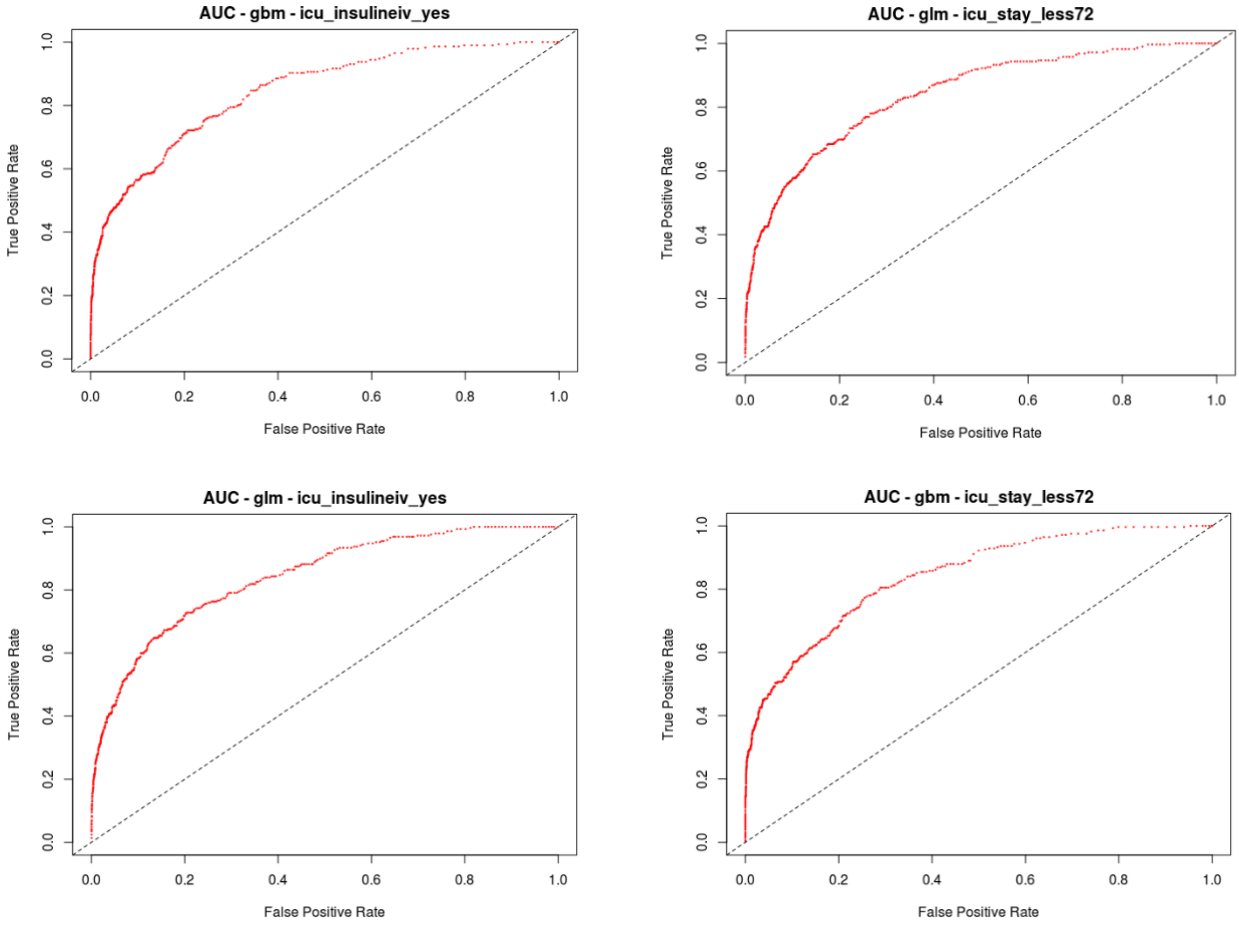

Figura 56: Curvas ROC glm/gbm

y el top 30 de las variables más importantes para ambos modelos y grupos en las figuras 57 a 60. Donde se observa que la importancia de las variables es diferente en cada caso, siendo en los modelos **gbm** en los que mayor importancia tienen las variables relacionadas con la glucosa.

[13] Ver apartado 7.1

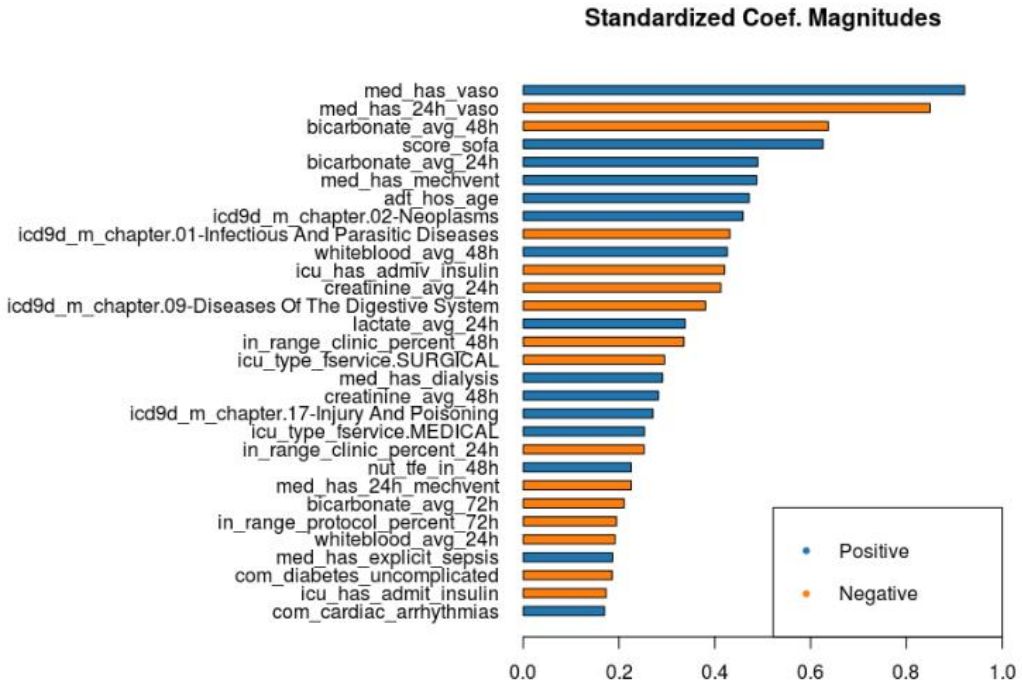

Figura 57: Importancia de variables, modelo glm. Top 30. Pacientes con estancia menor de 3 días en UCI.

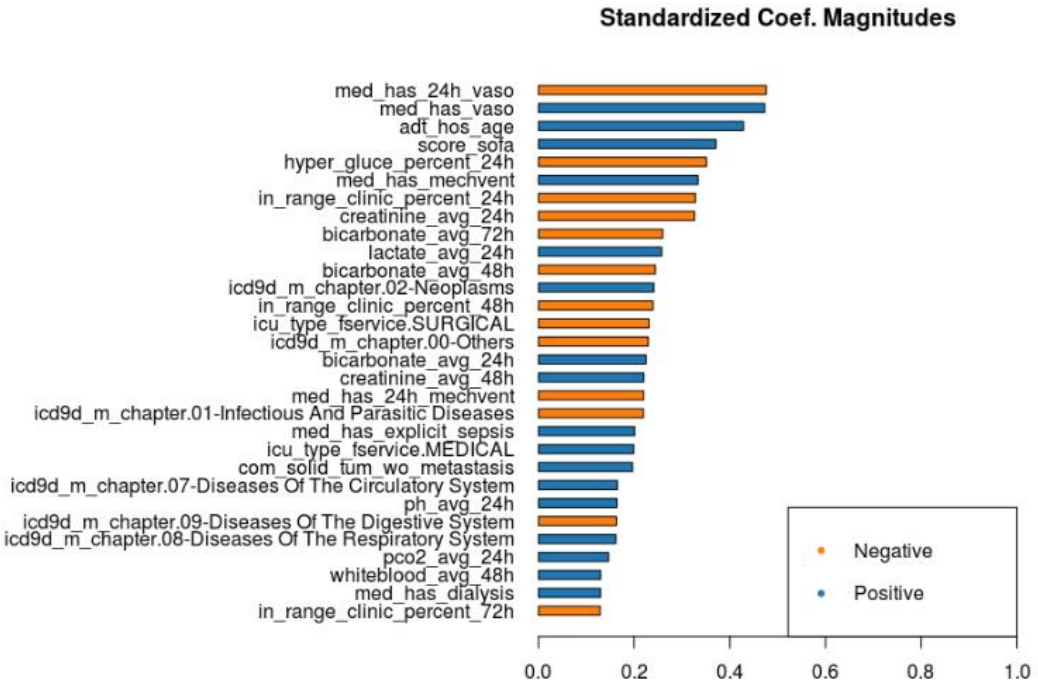

Figura 58:Importancia de variables, modelo glm. Top 30. Pacientes no tratados con insulina IV.

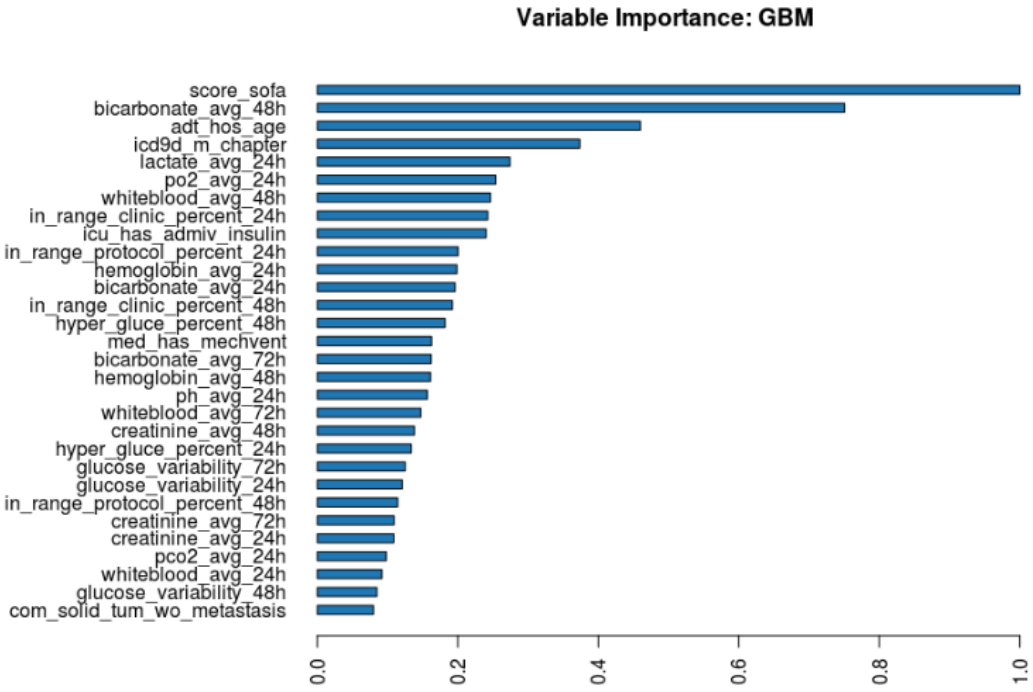

Figura 59: Importancia de variables, modelo gbm. Top 30. Pacientes con estancia menor de 3 días en UCI.

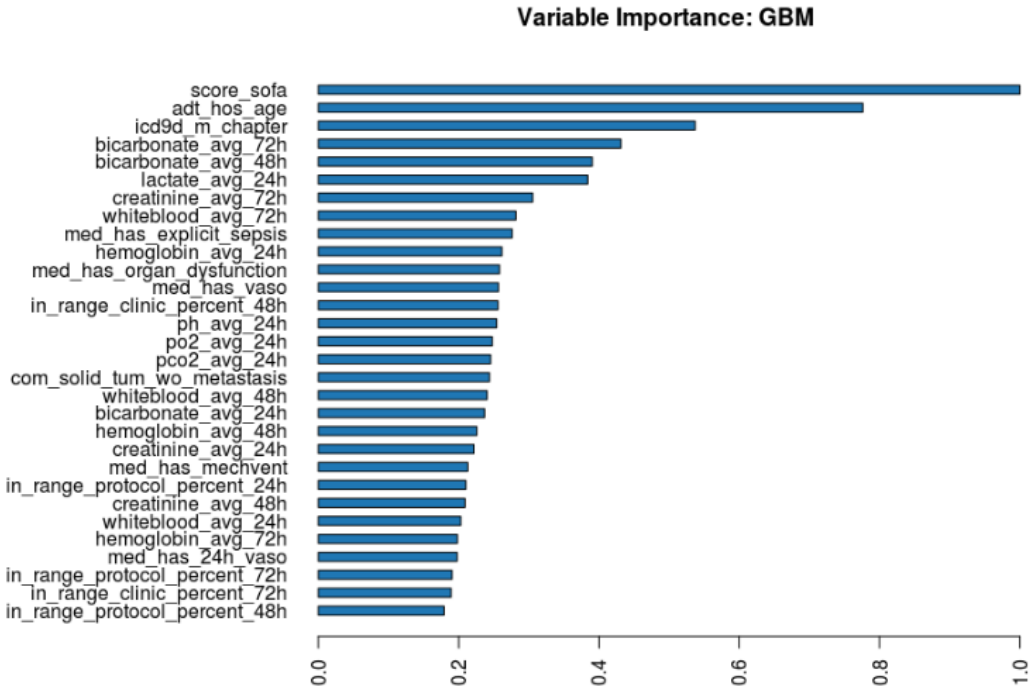

Figura 60: Importancia de variables, modelo gbm. Top 30. Pacientes no tratados con insulina IV.

En las figuras 61 a 63 se muestra la importancia de las variables (top 30) para los grupos de diabéticos y no diabéticos, donde se observa que para el modelo **gbm** y el grupo de pacientes sin diabetes, la importancia de la variabilidad de la glucosa **glucose_variability_48h** y de las variables relacionadas **in_range_clinic_percent_48h/24h** es mayor que en el grupo de diabéticos.

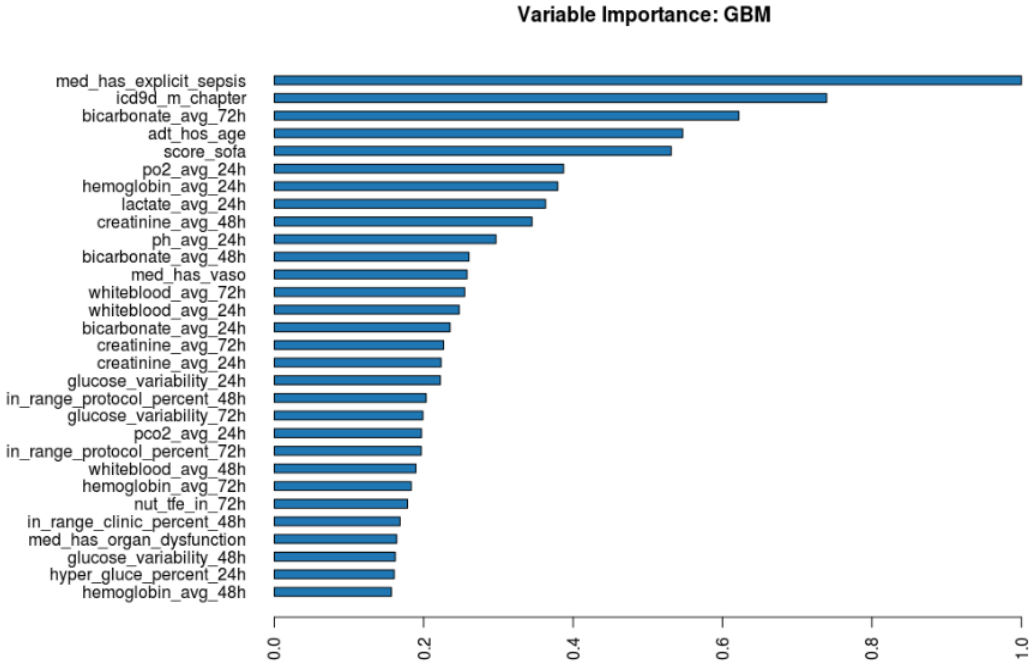

Figura 61: Importancia de variables, modelo gbm. Top 30. Pacientes con diabetes.

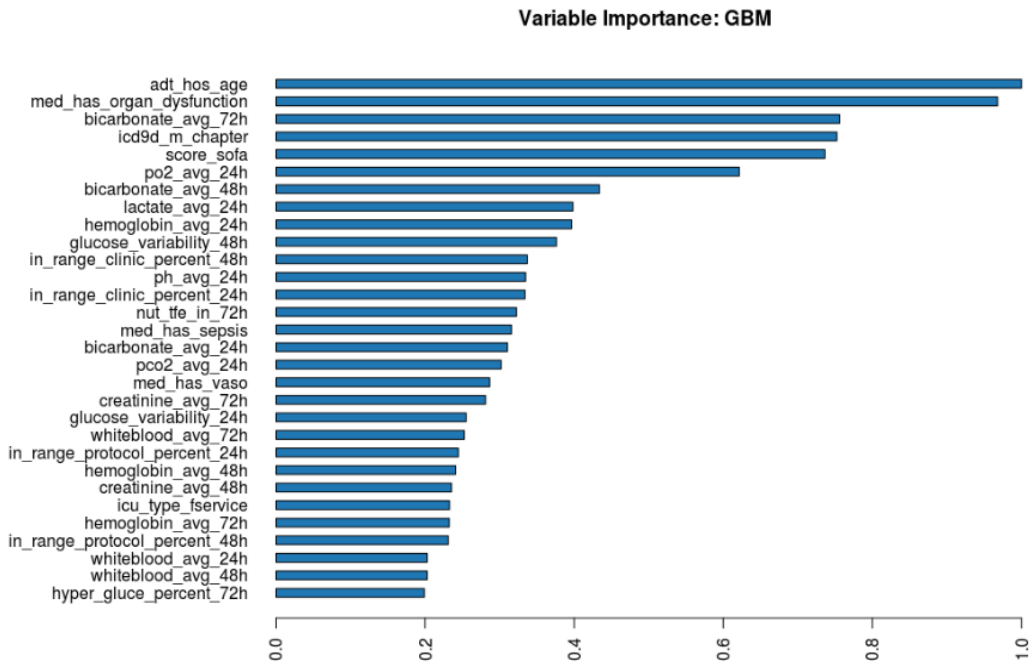

Figura 62:Importancia de variables, modelo gbm. Top 30. Pacientes sin diabetes.

En el caso del modelo **glm** también se aprecia diferencia, aunque no tan clara teniendo las variables relacionadas con la glucosa, **in_range_clinic_percent_48**, e **in_range_clinic_percent_24h** más peso en el caso de pacientes no diabéticos que diabéticos.

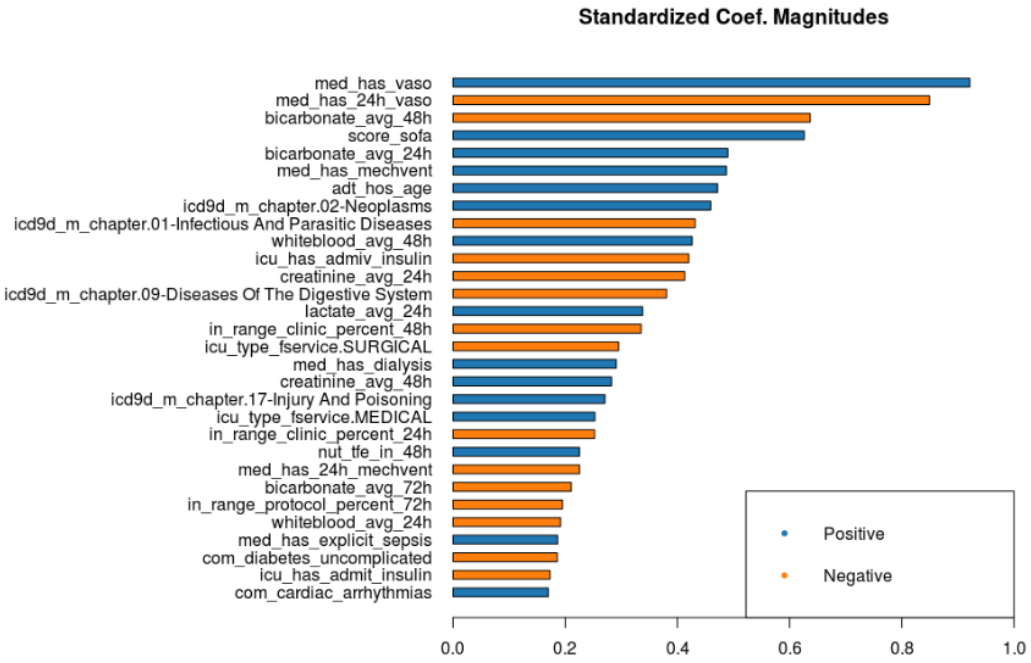

Figura 63: Importancia de variables, modelo glm. Top 30. Pacientes con diabetes.

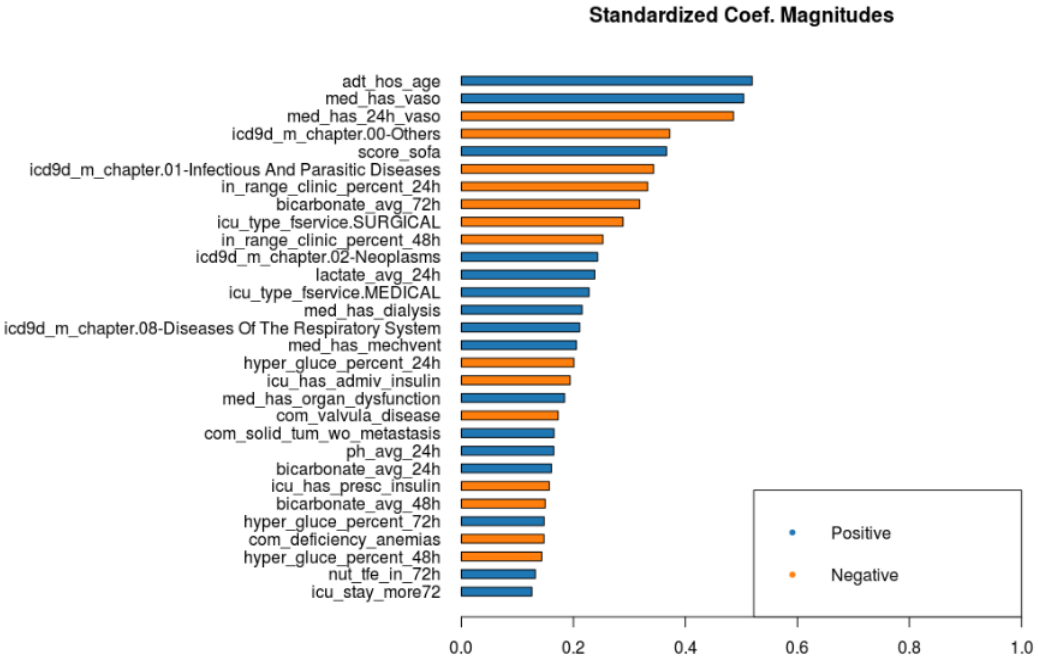

Figura 64: Importancia de variables, modelo glm. Top 30. Pacientes sin diabetes.

8 Conclusiones

Se incluyen en este capítulo las conclusiones del estudio, una valoración crítica y posibles líneas de trabajo futuro.

8.1 Conclusiones del trabajo

Las conclusiones del trabajo se pueden desglosar en tres apartados, por una parte, las relativas al uso de la BBDD MIMIC-III, por otra las asociadas a la metodología, la arquitectura técnica y herramientas utilizadas y finalmente las relacionadas con los resultados del análisis de datos.

En el primer punto, el uso de la BBDD MIMIC-III para analizar datos sanitarios, se considera que es probablemente la mejor alternativa existente para desarrollar estudios que puedan ser replicados y compartidos con la comunidad investigadora y educativa, evitando problemáticas relacionadas con la privacidad y especial sensibilidad de los datos sanitarios, las dificultades de integración entre diversas aplicaciones, etc. En todo caso, es necesario tener en cuenta ciertas características específicas de MIMIC-III debido a las diferencias entre el sistema sanitario de EEUU y los sistemas sanitarios públicos existentes en España, principalmente, el carácter privado de las organizaciones sanitarias de EEUU y su orientación del uso del sistema de codificación CIE9 no tanto hacia objetivos sanitarios como hacia la gestión de la facturación de los procedimientos clínicos [125]. Otro punto a considerar en MIMIC-III es la existencia de muchos datos semiestructurados, introducidos manualmente en texto libre y en dos sistemas diferentes, lo que obliga a realizar un esfuerzo de búsqueda para obtener ciertas variables, sin acceso a los sistemas reales para su validación. Aún con estas consideraciones, La BBDD MIMIC-III, fruto de un considerable esfuerzo, cubre con creces sus objetivos y este trabajo habría sido imposible sin su existencia.

En este trabajo se ha desarrollado una metodología, adaptada a partir de otros dos estándares en minería de datos y *Big Data*, que se ha mostrado eficaz como marco de las diferentes fases del análisis de datos. Se ha implementado una arquitectura técnica basada en el uso de la ETL Pentaho PDI, el *framework* MLR de *Machine Learning* sobre R y los modelos predictivos de H2O, que se considera satisfactoria desde el punto de vista de la eficiencia, la velocidad de tratamiento de datos, la escalabilidad y la posibilidad de adaptarse a otros proyectos en entornos hospitalarios o de investigación. Una parte considerable del tiempo empleado en un análisis de datos se dedica a las tareas de obtención y tratamiento de datos y en ese punto, los procesos ETL desarrollados, han cubierto sin problemas los objetivos. El uso de R para realizar este tipo de análisis se justifica por la disponibilidad en dicho lenguaje de *frameworks* y gran cantidad de modelos. En este sentido, el desarrollo del trabajo ha mostrado, a nuestro juicio, que el *framework* MLR es más adecuado para las tareas de *benchmarking* y *tunning* de modelos que el paquete CARET,

a pesar de la fuerte implantación de este último, que se explica sobre todo por la amplia y clara documentación disponible. En todo caso, nada impide usar los dos *frameworks de forma* simultánea cuidando no solapar funciones, realizar el preprocesado de datos y particionado con CARET y el *benchmarking* y *tunning* con MLR, por ejemplo.

En cuanto a la evaluación de modelos predictivos, se han realizado múltiples pruebas hasta llegar a una combinación satisfactoria. En R existen una cantidad inmensa de modelos y no todos presentan unas características de estabilidad y escalabilidad adecuadas para trabajar con conjuntos grandes de datos en tiempos razonables. Desde ese punto de vista, los modelos de H2O han sido capaces de tratar el conjunto completo de datos, obteniendo buenos resultados con tiempos más reducidos por su procesamiento en memoria (del orden de horas en lugar de decenas de horas, en este estudio).

En resumen, mediante la evaluación de diferentes herramientas, tecnologías y modelos se ha llegado a una combinación fiable, rápida y escalable para el tipo de datos clínicos estudiados.[14]

Un apunte interesante respecto de la selección de modelos es que se han obtenido buenos resultados con un modelo conocido y 'clásico' como **glm** y no con otro más 'innovador' como **deeplearning.** Por una parte, el coste de optimización de parámetros y de entrenamiento de una red neuronal es muy alto y no se han obtenido resultados destacables frente al resto de modelos, y por otra parte la selección de variables realizada previamente, con prevalencia de variables que por la experiencia médica tienen relación con la mortalidad, puede haber favorecido los modelos que utilizan relaciones más directas para obtener las predicciones. La ventaja asociada es que se obtienen modelos más 'interpretables' aunque se debe ser siempre cauto a la hora de cuantificar la importancia de una variable en el *outcome* fuera de modelos lineales o logísticos.

Finalmente, en la aplicación de los modelos predictivos se ha obtenido que los modelos con mejores resultados han sido **gbm** y **glm**, (AUC **gbm** 0.8353 / AUC **glm** 0.8112) por encima de los obtenidos para el *score* tomado como referencia (AUC APSIII 0.725). Los modelos **gbm** y **glm** obtienen las mejores predicciones en el conjunto de datos completo y en los distintos grupos en los que se ha dividido este. Mediante test paramétricos de hipótesis se concluye que para la métrica AUC el mejor modelo es el **gbm**. En los diferentes grupos de población se observa que el poder predictivo de los modelos seleccionados es mayor en los pacientes con estancia menor de tres días y en los tratados con insulina IV.

Se ha obtenido también la ganancia del indicador AUC al incluir las variables de comorbilidades en el conjunto de predictores (**gbm** 0.3646% / **glm** 0.2245%, para el conjunto completo y **gbm** 0.771106% / **glm**

[14] Ver tabla 12, apartado 7.3.4

0.2598803% de media en los diferentes grupos). Desde el punto de vista clínico se considera una ganancia de potencia predictora interesante, dada la facilidad para obtener las comorbilidades de un paciente con el diagnóstico inicial.

En cuanto al análisis de la importancia de las variables relacionadas con la glucosa, se observa que tienen casi todas una importancia media, salvo la variable **in_range_clinic_percent_24h,** con mayor importancia en todos los modelos. En general, las variables relacionadas con la glucosa tienen más importancia en los modelos **gbm**, tanto para el conjunto de datos como en los diferentes grupos. Para los grupos de diabéticos y no diabéticos, se observa que para el modelo **gbm** y el grupo de pacientes sin diabetes, la importancia de la variabilidad de la glucosa **glucose_variability_48h** y de las variables relacionadas **in_range_clinic_percent_48h/24h** es mayor que en el grupo de diabéticos.

Concluyendo, se han conseguido unos modelos con mayor capacidad de predicción que los *scores* tomados como base, seleccionando variables relacionadas con la glucosa, comorbilidades y datos de analítica básica. Sin una exploración más exhaustiva no es posible concluir que la variabilidad de la glucosa tiene más importancia que el resto de variables elegidas, aunque se obtienen resultados que indican que en algunos grupos sí que parecen tener más influencia, en línea con varios estudios mencionados [15], [16]. Como se ha indicado, es necesario ser cauto en la interpretación de la importancia de las variables en modelos diferentes de los lineales, en los logísticos (**glm**) tiene una relación exponencial con la ratio de probabilidades, mientras que en los **gbm** está relacionada con el peso relativo de las variables en las iteraciones del modelo [95].

8.2 Valoración crítica

El desarrollo del presente trabajo ha cumplido los objetivos y las expectativas iniciales, sobre todo en el apartado de evaluación técnica y en el desarrollo de una metodología de análisis y aplicación de modelos predictivos a datos sanitarios, teniendo en cuenta la escala temporal limitada del mismo. La metodología seguida se ha mostrado suficientemente robusta y eficaz para este tipo de proyectos. Las distintas fases del trabajo no han estado exentas de problemas tanto técnicos como asociados a la propia estructura de datos de la BBDD MIMIC-III, que han obligado a desarrollar en mayor grado de lo estimado inicialmente los procesos de ETL y dedicar tiempo a la construcción de una arquitectura hardware y software escalable. En este trabajo se ha obtenido experiencia directa con la aplicación de modelos a datos clínicos reales y en el tratamiento de dichos datos con herramientas ETL, experiencia que es trasladable a entornos similares. La planificación inicial se ha ajustado a lo largo de los cuatro meses del trabajo, aunque en líneas generales los plazos de las distintas fases se han mantenido respecto lo previsto inicialmente. Entre los problemas encontrados y para los que ha sido preciso obtener solución, destacar por ejemplo el elevado tiempo de

búsqueda de parámetros para los modelos de redes neuronales, las necesidades de memoria y CPU para buena parte de los modelos disponibles en R, errores encontrados en modelos a priori prometedores, como los modelos Ensemble, que han impedido su inclusión en el trabajo, problemas de compatibilidad de versiones de java y algunos paquetes de R, la dificultad para identificar algunas variables en la BBDD MIMIC-III, la estructura de las tablas de entradas de líquidos en MIMIC-III, etc. Algunos problemas se solucionaron modificando la arquitectura hardware y/o software, investigando en webs de soporte, (*stackoverflow* y canales de los fabricantes, principalmente), empleando conjuntos de datos más reducidos o buscando alternativas viables para obtener la combinación presentada aquí.

8.3 Líneas de trabajo futuro

Una vez realizado el desarrollo de una ETL para obtener datos de la BBDD MIMIC-IIII, evaluar y seleccionar los modelos con mayor capacidad predictiva y realizar un análisis de datos agrupando por varias variables, queda un margen amplio para futuras investigaciones sobre esta base, entre ellas:

- Cuantificación del efecto de la variabilidad de la glucosa en las métricas AUC.
- Análisis comparativo detallado de la importancia de las variables en los distintos grupos.
- Análisis de la influencia en la mortalidad de otras variables disponibles en la BBDD MIMIC-III.
- Aplicación de algoritmos de optimización combinatoria de selección de variables en los modelos **gbm** y **glm** para obtener modelos equivalentes con el menor conjunto de variables.
- Inclusión de la dimensión temporal en los análisis en lugar utilizar agregados con valores medios por tramo.

Otra futura línea de trabajo interesante va en la dirección, no tanto de refinar y profundizar el análisis aquí desarrollado a partir de la BBDD MIMIC-III, sino en aprovechar la experiencia general obtenida, la metodología y los procesos ETL para construir una MIMIC-III local a partir de los diferentes sistemas de información de un hospital, desarrollando una ETL que realice la agregación y consolidación de datos y la anonimización y procesado necesario, de forma análoga a como se ha construido la BBDD MIMIC-III, con la ventaja que podría servir de plataforma piloto para desarrollar y evaluar modelos y realizar investigaciones aplicadas a datos semejantes a la práctica clínica local y las características de la población.

9 Glosario

AUC: *Area Under the Curve*
ADT: *Admission, discharge and transfer system*
BBDD: Base de datos
CARET: *Classification And Regression Training*
CIE9: Clasificación Internacional de Enfermedades 9ª Revisión
CV: *Cross Validation*
DRG: *Diagnostic Related Group*
ETL: *Extraction, Transformation and Load*
GRD: Grupo Relacionado de Diagnóstico
IDE: *Integrated Development Environment*
ICD9: *International Classification of Diseasses 9º Version*
ICU: *Intensive Care Unit*
IV: Intravenosa
MLR: *Machine Learning in R*
MMCE: M*ean Misclassification Error*)
MIMIC: *Medical Information Mart for Intensive Care*
PIB: Producto Interior Bruto
PDI: *Pentaho Data Integration*
ROC: *Response Operating Characteristic*
SQL: *Structured Language Query*
SO: Sistema Operativo
SOFA: *Sequential Organ Failure Assessment*
TIR: *Time In Range*
TOC: *Total Cost of Ownership*
UCI: Unidad de Cuidados Intensivos

10 Anexos

1. **TABLAS MIMIC-III y VISTAS MIMIC-III**
2. **PROCESO ETL**
3. **TABLAS MIMICSEL**
4. **CONJUNTOS DE DATOS**
5. **MANUAL DE INSTALACIÓN MIMIC / PENTAHO PDI**
6. **CRITERIOS OBTENCIÓN VARIABLES**
7. **CÓDIGO FUENTE**

11 Referencias

[1] A. Elixhauser, C. Steiner, D. R. Harris, and R. M. Coffey, “Comorbidity Measures for Use with Administrative Data,” *Medical Care*, vol. 36, no. 1, pp. 8–27, 1998.

[2] M. Egi and R. Bellomo, “Reducing Glycemic Variability in Intensive Care Unit Patients: A New Therapeutic Target?,” *Journal of Diabetes Science and Technology*, vol. 3, no. 6, pp. 1302–1308, Nov. 2009.

[3] Y. Zhang and M. S. C. Hemond, “Uncovering the Predictive Value of Minimum Blood Glucose Through Statistical Analysis of a Large Clinical Dataset,” *AMIA Annu Symp Proc*, vol. 2009, pp. 725–729, 2009.

[4] W. L. Wahl, M. Taddonio, P. M. Maggio, S. Arbabi, and M. R. Hemmila, “Mean glucose values predict trauma patient mortality,” *J Trauma*, vol. 65, no. 1, pp. 42-47; discussion 47-48, Jul. 2008.

[5] T. N.-S. S. Investigators, “Hypoglycemia and Risk of Death in Critically Ill Patients,” *New England Journal of Medicine*, vol. 367, no. 12, pp. 1108–1118, Sep. 2012.

[6] T. N.-S. S. Investigators, “Intensive versus Conventional Glucose Control in Critically Ill Patients,” *New England Journal of Medicine*, vol. 360, no. 13, pp. 1283–1297, Mar. 2009.

[7] S. Derde, I. Vanhorebeek, and G. Van den Berghe, “Insulin Treatment in Intensive Care Patients,” *Horm Res Paediatr*, vol. 71, no. 1, pp. 2–11, Nov. 2008.

[8] M. M. Treggiari, V. Karir, N. D. Yanez, N. S. Weiss, S. Daniel, and S. A. Deem, “Intensive insulin therapy and mortality in critically ill patients,” *Critical Care*, vol. 12, p. R29, 2008.

[9] D. E. G. Griesdale *et al.*, “Intensive insulin therapy and mortality among critically ill patients: a meta-analysis including NICE-SUGAR study data,” *CMAJ*, vol. 180, no. 8, pp. 821–827, Apr. 2009.

[10] M. A. Kovalaske and G. Y. Gandhi, “Glycemic Control in the Medical Intensive Care Unit,” *Journal of Diabetes Science and Technology*, vol. 3, no. 6, pp. 1330–1341, Nov. 2009.

[11] K. Amrein *et al.*, “Glucose control in intensive care: usability, efficacy and safety of Space GlucoseControl in two medical European intensive care units,” *BMC Endocrine Disorders*, vol. 14, p. 62, 2014.

[12] J. S. Krinsley, “Glycemic control in the critically ill: What have we learned since NICE-SUGAR?,” *Hospital Practice*, vol. 43, no. 3, pp. 191–197, Jul. 2015.

[13] R. Brunner, G. Adelsmayr, H. Herkner, C. Madl, and U. Holzinger, “Glycemic variability and glucose complexity in critically ill patients: a retrospective analysis of continuous glucose monitoring data,” *Critical Care*, vol. 16, p. R175, 2012.

[14] I. A. Meynaar, S. Eslami, A. Abu-Hanna, P. van der Voort, D. W. de Lange, and N. de Keizer, “Blood glucose amplitude variability as predictor for mortality in surgical and medical intensive care unit patients: a multicenter cohort study,” *Journal of Critical Care*, vol. 27, no. 2, pp. 119–124, Apr. 2012.

[15] J. S. Krinsley, “Glycemic control in the critically ill - 3 domains and diabetic status means one size does not fit all!,” *Critical Care*, vol. 17, p. 131, 2013.

[16] J. Clain, K. Ramar, and S. R. Surani, “Glucose control in critical care,” *World J Diabetes*, vol. 6, no. 9, pp. 1082–1091, Aug. 2015.

[17] M. J. Lanspa, J. Dickerson, A. H. Morris, J. F. Orme, J. Holmen, and E. L. Hirshberg, “Coefficient of glucose variation is independently associated with mortality in critically ill patients receiving intravenous insulin,” *Critical Care*, vol. 18, p. R86, 2014.

[18] J. Blaha *et al.*, “Comparison of Three Protocols for Tight Glycemic Control in Cardiac Surgery Patients,” *Diabetes Care*, vol. 32, no. 5, pp. 757–761, May 2009.

[19] C. De Block, B. Manuel-y-Keenoy, P. Rogiers, P. Jorens, and L. Van Gaal, “Glucose control and use of continuous glucose monitoring in the intensive care unit: a critical review,” *Curr Diabetes Rev*, vol. 4, no. 3, pp. 234–244, Aug. 2008.
[20] “European Health Parliament - 7 recommendations on Healthcare in Europe,” *issuu*. [Online]. Available: https://issuu.com/europeanhealthparliament/docs/ehp_papers_boek_schermversie. [Accessed: 11-May-2017].
[21] “Comment: Health networks - delivering the future of healthcare.” [Online]. Available: https://www.buildingbetterhealthcare.co.uk/technical/article_page/Comment_Health_networks__delivering_the_future_of_healthcare/94931. [Accessed: 11-May-2017].
[22] T. S. P. T. M. M. T. T. P. 6 O. 4 A. E. L. 32 08302 Mataró, “Informe Big Data Technologies in Healthcare: Necesidades, oportunidades y retos en el sector salud,” *TicSalut*. [Online]. Available: http://www.ticsalut.cat/actualitat/es_flashticsalut/article/406/informe-big-data-technologies-in-healthcare-necesidades-oportunidades-y-retos-en-el-sector-salud. [Accessed: 11-May-2017].
[23] “Interoperabilidad: La torre de babel de los sistemas de salud,” *Hackathon Nacional de Salud*, 21-Mar-2016. .
[24] “What is Interoperability?,” *HIMSS*, 09-Mar-2016. [Online]. Available: http://www.himss.org/library/interoperability-standards/what-is-interoperability. [Accessed: 11-May-2017].
[25] W. Raghupathi and V. Raghupathi, “Big data analytics in healthcare: promise and potential,” *Health Inf Sci Syst*, vol. 2, Feb. 2014.
[26] “Data-driven healthcare organizations use big data analytics for big,” *IBM Big Data & Analytics Hub*. [Online]. Available: http://www.ibmbigdatahub.com/whitepaper/data-driven-healthcare-organizations-use-big-data-analytics-big-gains. [Accessed: 02-Jun-2017].
[27] B. Kayyali, D. Knott, and S. V. Kuiken, “The big-data revolution in US health care: Accelerating value and innovation | McKinsey & Company.” [Online]. Available: http://www.mckinsey.com/industries/healthcare-systems-and-services/our-insights/the-big-data-revolution-in-us-health-care. [Accessed: 02-Jun-2017].
[28] J. L. Vincent *et al.*, “The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure. On behalf of the Working Group on Sepsis-Related Problems of the European Society of Intensive Care Medicine,” *Intensive Care Med*, vol. 22, no. 7, pp. 707–710, Jul. 1996.
[29] “CRISP-DM by Smart Vision Europe.” [Online]. Available: http://crisp-dm.eu/. [Accessed: 18-May-2017].
[30] “CRISP-DM, still the top methodology for analytics, data mining, or data science projects.” [Online]. Available: http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html. [Accessed: 18-May-2017].
[31] R. Wirth, “CRISP-DM: Towards a standard process model for data mining,” in *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 2000, pp. 29–39.
[32] “List of ICD-9 codes,” *Wikipedia*. 30-Dec-2016.
[33] “The Web’s Free ICD-9-CM & ICD-10-CM Medical Coding Reference.” [Online]. Available: http://www.icd9data.com/. [Accessed: 06-Feb-2017].
[34] “WHO | International Classification of Diseases,” *WHO*. [Online]. Available: http://www.who.int/classifications/icd/en/. [Accessed: 02-Jun-2017].
[35] “PostgreSQL: The world’s most advanced open source database.” [Online]. Available: https://www.postgresql.org/. [Accessed: 03-Feb-2017].
[36] “RStudio,” *RStudio*. [Online]. Available: https://www.rstudio.com/products/rstudio/. [Accessed: 05-May-2017].

[37] “H2O.ai.” [Online]. Available: http://www.h2o.ai/. [Accessed: 03-Feb-2017].
[38] “http://www.pentaho.com/product/version-7-0-update,” *Pentaho*. [Online]. Available: http://www.pentaho.com/product/version-7-0-update. [Accessed: 05-May-2017].
[39] “PostgreSQL wiki.” [Online]. Available: https://wiki.postgresql.org/wiki/Main_Page. [Accessed: 05-May-2017].
[40] “MySQL.” [Online]. Available: https://www.mysql.com/. [Accessed: 05-May-2017].
[41] “Talend Real-Time Open Source Big Data Integration Software,” *Talend Real-Time Open Source Data Integration Software*. [Online]. Available: https://www.talend.com/. [Accessed: 05-May-2017].
[42] “R: The R Project for Statistical Computing.” [Online]. Available: https://www.r-project.org/. [Accessed: 05-May-2017].
[43] “CRAN - Mirrors.” [Online]. Available: https://cran.r-project.org/mirrors.html. [Accessed: 05-May-2017].
[44] “scikit-learn: machine learning in Python — scikit-learn 0.18.1 documentation.” [Online]. Available: http://scikit-learn.org/stable/. [Accessed: 13-May-2017].
[45] “Machine Learning with MATLAB - MATLAB & Simulink.” [Online]. Available: https://www.mathworks.com/solutions/machine-learning.html. [Accessed: 02-Jun-2017].
[46] “Data Science Platform,” *RapidMiner*. [Online]. Available: https://rapidminer.com/. [Accessed: 02-Jun-2017].
[47] “Weka 3 - Data Mining with Open Source Machine Learning Software in Java.” [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: 02-Jun-2017].
[48] “KNIME | Open for Innovation.” [Online]. Available: https://www.knime.org/. [Accessed: 02-Jun-2017].
[49] A. E. W. Johnson *et al.*, “MIMIC-III, a freely accessible critical care database,” *Scientific Data*, vol. 3, p. 160035, May 2016.
[50] R. Mark, “The Story of MIMIC,” in *Secondary Analysis of Electronic Health Records*, Springer International Publishing, 2016, pp. 43–49.
[51] “LCP welcome.” [Online]. Available: http://lcp.mit.edu/index.shtml. [Accessed: 11-May-2017].
[52] “MIMIC.” [Online]. Available: https://mimic.physionet.org/about/mimic/. [Accessed: 11-May-2017].
[53] “Requesting access.” [Online]. Available: https://mimic.physionet.org/gettingstarted/access/. [Accessed: 11-May-2017].
[54] “Downloading the database.” [Online]. Available: https://mimic.physionet.org/gettingstarted/dbsetup/. [Accessed: 11-May-2017].
[55] W. Farhan, Z. Wang, Y. Huang, S. Wang, F. Wang, and X. Jiang, “A Predictive Model for Medical Events Based on Contextual Embedding of Temporal Sequences,” *JMIR Med Inform*, vol. 4, no. 4, p. e39, Nov. 2016.
[56] J. Lee, E. Ribey, and J. R. Wallace, “A web-based data visualization tool for the MIMIC-II database,” *BMC Medical Informatics and Decision Making*, vol. 16, p. 15, 2016.
[57] C. Chronaki, A. Shahin, and R. Mark, “Designing Reliable Cohorts of Cardiac Patients across MIMIC and eICU,” *Computing in cardiology*, vol. 42, p. 189, 2015.
[58] M. M. Aboelsoud, O. Siddique, A. Morales, Y. Seol, and M. O. Al-Qadi, “Fluid Choice Matters in Critically-ill Patients with Acute Pancreatitis: Lactated Ringer’s vs. Isotonic Saline,” *R I Med J (2013)*, vol. 99, no. 10, pp. 39–42, Oct. 2016.
[59] R. Pirracchio, “Mortality Prediction in the ICU Based on MIMIC-II Results from the Super ICU Learner Algorithm (SICULA) Project,” in *Secondary Analysis of Electronic Health Records*, Springer International Publishing, 2016, pp. 295–313.
[60] T. Desautels *et al.*, “Prediction of Sepsis in the Intensive Care Unit With Minimal Electronic Health Record Data: A Machine Learning Approach,” *JMIR Med Inform*, vol. 4, no. 3, p. e28, Sep. 2016.

[61] Cynthia Alexander Rascon, “Predictive Modeling ICU Health Outcomes with MIMIC III Data,” Aug-2016.
[62] X.-D. Zhou *et al.*, “Quick chronic liver failure-sequential organ failure assessment: an easy-to-use scoring model for predicting mortality risk in critically ill cirrhosis patients,” *Eur J Gastroenterol Hepatol*, Feb. 2017.
[63] S. V. Poucke *et al.*, “Scalable Predictive Analysis in Critically Ill Patients Using a Visual Open Data Analysis Platform,” *PLOS ONE*, vol. 11, no. 1, p. e0145791, ene 2016.
[64] *Secondary Analysis of Electronic Health Records | MIT Critical Data | Springer*. .
[65] “GitHub - MIT-LCP/mimic-code: MIMIC Code Repository: Code shared by the research community for the MIMIC-III database.” [Online]. Available: https://github.com/MIT-LCP/mimic-code. [Accessed: 23-Feb-2017].
[66] “eCIE-Maps - CIE-9-MC.” [Online]. Available: http://eciemaps.mspsi.es/ecieMaps/browser/index_9_mc.html. [Accessed: 09-Feb-2017].
[67] “Ministerio de Sanidad, Servicios Sociales e Igualdad - Portal Estadístico del SNS - Estadisticas y Estudios - Normalización y Codificaciones.” [Online]. Available: https://www.msssi.gob.es/estadEstudios/estadisticas/normalizacion/clasifEnferm/home.htm. [Accessed: 11-May-2017].
[68] “LOINC — The freely available standard for identifying health measurements, observations, and documents.” .
[69] “MIT-LCP/mimic-code,” *GitHub*. [Online]. Available: https://github.com/MIT-LCP/mimic-code. [Accessed: 02-Feb-2017].
[70] I. Neamatullah *et al.*, “Automated de-identification of free-text medical records,” *BMC Medical Informatics and Decision Making*, vol. 8, p. 32, 2008.
[71] W. A. Knaus *et al.*, “The APACHE III Prognostic System: Risk Prediction of Hospital Mortality for Critically III Hospitalized Adults,” *Chest*, vol. 100, no. 6, pp. 1619–1636, Dec. 1991.
[72] J.-R. L. Gall *et al.*, “The Logistic Organ Dysfunction System: A New Way to Assess Organ Dysfunction in the Intensive Care Unit,” *JAMA*, vol. 276, no. 10, pp. 802–810, Sep. 1996.
[73] A. E. W. Johnson, A. A. Kramer, and G. D. Clifford, “A new severity of illness scale using a subset of Acute Physiology And Chronic Health Evaluation data elements shows comparable predictive accuracy,” *Crit. Care Med.*, vol. 41, no. 7, pp. 1711–1718, Jul. 2013.
[74] M. Singer *et al.*, “The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3),” *JAMA*, vol. 315, no. 8, pp. 801–810, Feb. 2016.
[75] J. R. Le Gall *et al.*, “A simplified acute physiology score for ICU patients,” *Crit. Care Med.*, vol. 12, no. 11, pp. 975–977, Nov. 1984.
[76] J.-R. L. Gall, S. Lemeshow, and F. Saulnier, “A New Simplified Acute Physiology Score (SAPS II) Based on a European/North American Multicenter Study,” *JAMA*, vol. 270, no. 24, pp. 2957–2963, Dec. 1993.
[77] F. L. Ferreira, D. P. Bota, A. Bross, C. Mélot, and J. L. Vincent, “Serial evaluation of the SOFA score to predict outcome in critically ill patients,” *JAMA*, vol. 286, no. 14, pp. 1754–1758, Oct. 2001.
[78] R. C. Bone *et al.*, “Definitions for Sepsis and Organ Failure and Guidelines for the Use of Innovative Therapies in Sepsis,” *Chest*, vol. 101, no. 6, pp. 1644–1655, Jun. 1992.
[79] “National Drug Code Directory.” [Online]. Available: http://www.accessdata.fda.gov/scripts/cder/ndc/default.cfm. [Accessed: 07-Feb-2017].
[80] “SERVICES.” [Online]. Available: https://mimic.physionet.org/mimictables/services/. [Accessed: 13-May-2017].
[81] D. C. Angus, W. T. Linde-Zwirble, J. Lidicker, G. Clermont, J. Carcillo, and M. R. Pinsky, “Epidemiology of severe sepsis in the United States: analysis of incidence,

outcome, and associated costs of care," *Crit. Care Med.*, vol. 29, no. 7, pp. 1303–1310, Jul. 2001.
[82] E. de Jonge and M. van der Loo, "Tutorial: An introduction with data cleaning with R," in *useR!2013*, 2013.
[83] T. Dasu and T. Johnson, "Data Quality," in *Exploratory Data Mining and Data Cleaning*, John Wiley & Sons, Inc., 2003, pp. 99–137.
[84] H. Wickham, "Tidy data," *The Journal of Statistical Software*, vol. 59, no. 10, 2014.
[85] "R vs Python for Data Science: The Winner is …." [Online]. Available: http://www.kdnuggets.com/2015/05/r-vs-python-data-science.html. [Accessed: 13-May-2017].
[86] M. Kuhn, "The caret Package." [Online]. Available: http://topepo.github.io/caret/train-models-by-tag.html#Two_Class_Only. [Accessed: 13-May-2017].
[87] "Integrated Learners - mlr tutorial." [Online]. Available: https://mlr-org.github.io/mlr-tutorial/release/html/integrated_learners/index.html. [Accessed: 23-May-2017].
[88] R. M. Forte, *Mastering Predictive Analytics with R*. Birmingham, England; Mumbai India: Packt Publishing - ebooks Account, 2015.
[89] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, 2013 edition. New York: Springer, 2013.
[90] *The Elements of Statistical Learning - Data Mining, Inference, | Trevor Hastie | Springer*. .
[91] "Binomial distribution," *Wikipedia*. 02-May-2017.
[92] "Bernoulli distribution," *Wikipedia*. 11-May-2017.
[93] "GLM — H2O 3.10.4.7 documentation." [Online]. Available: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glm.html. [Accessed: 14-May-2017].
[94] "GBM — H2O 3.10.4.7 documentation." [Online]. Available: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/gbm.html. [Accessed: 14-May-2017].
[95] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 200110.
[96] "A Kaggle Master Explains Gradient Boosting," *No Free Hunch*, 23-Jan-2017. [Online]. Available: http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/. [Accessed: 15-May-2017].
[97] "Activation function," *Wikipedia*. 13-May-2017.
[98] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout Networks," *arXiv:1302.4389 [cs, stat]*, Feb. 2013.
[99] E. LeDell, M. J. van der Laan, and M. Peterson, "AUC-Maximizing Ensembles through Metalearning," *Int J Biostat*, vol. 12, no. 1, pp. 203–218, May 2016.
[100] "ck37/superlearner-guide," *GitHub*. [Online]. Available: https://github.com/ck37/superlearner-guide. [Accessed: 19-Apr-2017].
[101] der L. M. J. van, E. C. Polley, and A. E. Hubbard, "Super Learner," *Statistical Applications in Genetics and Molecular Biology*, vol. 6, no. 1, 2007.
[102] E. Polley, E. LeDell, C. Kennedy, S. Lendle, and M. van der Laan, "SuperLearner: Super Learner Prediction," 01-Dec-2016. [Online]. Available: https://cran.r-project.org/web/packages/SuperLearner/index.html. [Accessed: 15-May-2017].
[103] "Stacked Ensembles — H2O 3.10.4.7 documentation." [Online]. Available: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/stacked-ensembles.html. [Accessed: 15-May-2017].
[104] "Grid (Hyperparameter) Search — H2O 3.10.4.7 documentation." [Online]. Available: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/grid-search.html. [Accessed: 15-May-2017].
[105] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[106] “Advanced Tuning - mlr tutorial.” [Online]. Available: https://mlr-org.github.io/mlr-tutorial/release/html/advanced_tune/index.html. [Accessed: 15-May-2017].
[107] “TuneControl function | R Documentation.” [Online]. Available: https://www.rdocumentation.org/packages/mlr/versions/2.10/topics/TuneControl. [Accessed: 15-May-2017].
[108] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle, “The irace package: Iterated racing for automatic algorithm configuration,” *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.
[109] O. Maron and A. W. Moore, “The Racing Algorithm: Model Selection for Lazy Learners,” *Artificial Intelligence Review*, vol. 11, no. 1–5, pp. 193–225, Feb. 1997.
[110] “Data Science 101: Preventing Overfitting in Neural Networks.” [Online]. Available: http://www.kdnuggets.com/2015/04/preventing-overfitting-neural-networks.html. [Accessed: 15-May-2017].
[111] C. Elkan, *Evaluating Classifiers*. 2012.
[112] “Precision and recall,” *Wikipedia*. 02-May-2017.
[113] T. Fawcett, “An Introduction to ROC Analysis,” *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[114] “The Receiver-Operating Characteristic (ROC) analysis: Fundamentals and applications in clinical psychology (PDF Download Available),” *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/256454600_The_Receiver-Operating_Characteristic_ROC_analysis_Fundamentals_and_applications_in_clinical_psychology. [Accessed: 16-May-2017].
[115] D. G. Altman and J. M. Bland, “Statistics Notes: Diagnostic tests 3: receiver operating characteristic plots,” *BMJ*, vol. 309, no. 6948, p. 188, Jul. 1994.
[116] H. Ma, A. I. Bandos, H. E. Rockette, and D. Gur, “On use of partial area under the ROC curve for evaluation of diagnostic performance,” *Stat Med*, vol. 32, no. 20, pp. 3449–3458, Sep. 2013.
[117] S. D. Hanekom, M. Faure, and A. Coetzee, “Outcomes research in the ICU: An aid in defining the role of physiotherapy,” *Physiotherapy Theory and Practice*, vol. 23, no. 3, pp. 125–135, Jan. 2007.
[118] G. D. Rubenfeld, D. C. Angus, M. R. Pinsky, J. R. Curtis, A. F. Connors, and G. R. Bernard, “Outcomes Research in Critical Care,” *Am J Respir Crit Care Med*, vol. 160, no. 1, pp. 358–367, Jul. 1999.
[119] A. G. Rapsang and D. C. Shyam, “Scoring systems in the intensive care unit: A compendium,” *Indian J Crit Care Med*, vol. 18, no. 4, pp. 220–228, Apr. 2014.
[120] A. Saleh, M. Ahmed, I. Sultan, and A. Abdel-lateif, “Comparison of the mortality prediction of different ICU scoring systems (APACHE II and III, SAPS II, and SOFA) in a single-center ICU subpopulation with acute respiratory distress syndrome,” *Egyptian Journal of Chest Diseases and Tuberculosis*, vol. 64, no. 4, pp. 843–848, Oct. 2015.
[121] M. Kuhn, “The caret Package - Feature Selection Overview.” [Online]. Available: https://topepo.github.io/caret/feature-selection-overview.html. [Accessed: 19-May-2017].
[122] “Implemented Performance Measures - mlr tutorial.” [Online]. Available: http://mlr-org.github.io/mlr-tutorial/release/html/measures/index.html. [Accessed: 20-May-2017].
[123] “Benchmark Experiments - mlr tutorial.” [Online]. Available: https://mlr-org.github.io/mlr-tutorial/release/html/benchmark_experiments/index.html. [Accessed: 24-May-2017].
[124] J. Demšar, “Statistical Comparisons of Classifiers over Multiple Data Sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1--30, Jan. 2006.
[125] B. Misset *et al.*, “Reliability of diagnostic coding in intensive care patients,” *Crit Care*, vol. 12, no. 4, p. R95, 2008.