Departamento de Computación
Facultad de Informática
UNIVERSIDADE DA CORUÑA

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

# Algoritmo de explicación de anomalías en espacios mixtos categórico-continuos

**Estudiante:** Iñigo Luis López-Riobóo Botana
**Dirección:** Carlos Eiras Franco
**Dirección:** María Amparo Alonso Betanzos

A Coruña, septiembre de 2019.

*A mi padre.*

**Agradecimientos**

En primer lugar, dar las gracias a mis tutores, Carlos y Amparo, por permitirme continuar sobre su trabajo de investigación y mostrar interés en mi desarrollo de TFG, además de todas las horas invertidas en investigación, apoyo y tutorización.

A mis padres y abuelos, con especial recuerdo de mi padre, por mostrar siempre apoyo incondicional a mi esfuerzo académico y a mi progreso como ingeniero informático. A todos ellos desde siempre, mi más profundo y sincero cariño.

También mencionar al grupo LIDIA, por facilitarme espacio de trabajo en el laboratorio de la facultad y atender las necesidades que me pudieran haber surgido durante el tiempo pasado en el lugar, así como dar las gracias por obtener mi primer contrato en apoyo a la investigación durante el desarrollo del proyecto. A todos ellos mi más sincero agradecimiento.

**Resumen**

En este proyecto se presenta un método de explicación para el algoritmo ADMNC (Anomaly Detector for Mixed Numerical and Categorical inputs) desarrollado en el grupo LIDIA (Laboratorio de Investigación y Desarrollo en Inteligencia Artificial) de la Facultad de Informática de A Coruña. Para el desarrollo del método de explicación se emplean los modelos de mezcla de gaussianas y de regresión logística inicialmente planteados sobre entornos mixtos categórico-continuos para entrenar un árbol de decisión que ayude a proporcionar una explicación pre-hoc sobre el modelo de datos normales, además de una explicación post-hoc en base a múltiples estimadores sobre aquellos patrones que ya han sido indicados como anomalía por el algoritmo.

El objetivo principal por tanto es proporcionar una nueva capa de explicación que sea de utilidad al supervisor y que subsane uno de los problemas más conocidos sobre los algoritmos en Inteligencia Artificial, que es la falta de justificación y la opacidad existente en muchos de ellos sobre el proceso interno seguido.

**Abstract**

This project presents an explanation method for the algorithm ADMNC (Anomaly Detector for Mixed Numerical and Categorical inputs) developed by the LIDIA group (Laboratorio de Investigación y Desarrollo en Inteligencia Artificial) of the Computer Science Department, Faculty of A Coruña. Gaussian mixture models and logistic regression are used for this development under mixed categorical-continuous spaces for training decision trees to achieve a pre-hoc explanation of the normal data model, as well as a post-hoc explanation based on multiple estimators over patterns that had been already indicated as anomalies by the algorithm.

The main objective is to provide a new explanation layer over this method that can be useful for a supervisor and can offset one of the most well-known problems of Artificial Intelligence algorithms, that is, the lack of justification and the opacity existing on the internal process followed.

**Palabras clave:**

- Detección anomalías
- Valores atípicos
- Explicación
- Aprendizaje automático
- Regresión logística
- Mezcla gaussiana
- Patrones categórico-continuos
- Escalabilidad

**Keywords:**

- Anomaly detection
- Outliers
- Explanation
- Machine Learning
- Logistic regression
- Gaussian mixture
- Categorical-continuous patterns
- Scalability

**Hardware y software utilizado:**

- Ordenador personal MSI GE62 6QD Apache Pro.
- Entorno IDE de desarrollo Eclipse Oxigen.
- Framework distribuido Apache Spark.
- Lenguaje de programación Scala.

# Índice general

# Índice de figuras

# Índice de tablas

Capítulo 1

# Introducción

Una anomalía se entiende como un valor atípico sobre un conjunto de datos que se diferencia del resto por alguna de sus propiedades o características, y que tiene la suficiente desviación como para sospechar que fuera producido a raíz de otros mecanismos [1]. La detección de anomalías es una antigua disciplina, consolidada en el campo de la Estadística como la detección de valores atípicos. Es una rama de estudio cada vez más explotada sobre entornos muy variados, con ejemplos en detección de intrusiones en redes [2], vigilancia [3], o monitorización de maquinaria, entre otras. Resulta de gran interés para los campos de desarrollo en Inteligencia Artificial y Computación en general, dada la posibilidad de utilizarlas con algoritmos de aprendizaje, así como de dotarlas de propiedades tales como la escalabilidad en el procesamiento de grandes cantidades de datos. No obstante, la mayor parte de los algoritmos expuestos sobre estos campos carecen de una explicación sobre los resultados obtenidos. Esto es, dado un valor de salida identificado como dato normal o anomalía no se tiene la información necesaria que permitiría a un supervisor corroborar los pasos seguidos por el algoritmo. Esta es una carencia común a muchas de las técnicas de Inteligencia Artificial hoy en día, donde la mayor parte del proceso seguido funciona a modo de caja negra, sin la posibilidad de obtener orientaciones acerca del porqué de los resultados. La opacidad de los métodos de aprendizaje es un claro impedimento a la comprensión de las salidas.

Algunas de las primeras aproximaciones a la explicación de las soluciones vienen de la mano de los sistemas expertos, modulados en bases de conocimientos y motores de inferencia que pueden dar explicaciones basadas en el conocimiento previo (estáticas) o razonadas sobre inferencias realizadas sobre el problema concreto (dinámicas) [4]. Variantes de los mismos contemplan modelos adicionales de lógica difusa orientada a usuario [5] con el objetivo de mejorar la aceptabilidad y la personalización de los resultados. Si bien pueden aplicarse a dominios reales y las explicaciones pueden ser completas, el problema de estos sistemas radica en la dificultad de modelado y adquisición de conocimiento, sobre todo por depender de un experto humano en el dominio, así como el mantenimiento y refinamiento de los sistemas

de reglas, los cuales pueden acabar siendo insostenibles. Otras aproximaciones contemplan algoritmos más específicos para incluir una capa de explicación, como puede ser el algoritmo Quirk [6]. Este se centra en la detección de anomalías utilizando un método que consiste en la explicación secuencial de características o variables del dominio (SFE, por las siglas en inglés de Sequential Feature Explanation). En ese caso, se presenta cada patrón etiquetado como anómalo mediante una serie incremental de características de su vector de datos. Esta información se proporciona al supervisor para que estudie el nivel de detalle de la justificación. Esta solución facilita la explicación post-hoc sobre datos ya clasificados como anomalía por un proceso de detección previo (k-means, k-NN…) y requiere de una supervisión humana continua para comprobar el correcto etiquetado y desempeño del algoritmo, por lo que la interfaz de usuario es otro punto importante en el proceso seguido. No obstante, es incapaz de dar explicaciones (y de representar los datos) en entornos con más de tres variables continuas. Además, carece del tratamiento de las variables de tipo discreto, lo que constituye un factor perjudicial a la hora de realizar procesado sobre conjuntos de datos reales. Tanto la dimensión como la naturaleza de los datos son factores críticos. Solamente contempla el estudio del dominio continuo de las variables, empleando métodos conocidos y genéricos de aprendizaje automático para el tratamiento de las anomalías. Por último, dicho algoritmo no utiliza las variables internas al modelo para llevar a cabo una explicación previa a la detección (pre-hoc), que pudiera proporcionar información de alto nivel sobre el entorno modelado. Existen también otras aplicaciones de explicación centradas en contextos más prácticos, como puede ser el diagnóstico de sistemas continuos físico-mecánicos. En este caso, es común utilizar algoritmos de ordenamiento causal para generar grafos sobre los que poder determinar una explicación [7]. El objetivo principal es detectar posibles incidentes que pudieran haber causado un fallo durante el proceso de diagnóstico. En estos casos, la explicación se centra en métodos de exploración del grafo para buscar correlaciones entre variables del modelo, funcionando como un módulo que da soporte a la monitorización.

Este proyecto fin de grado se centra en el desarrollo de un algoritmo de explicación sobre el algoritmo de detección de anomalías [8] ya existente ideado por el grupo LIDIA de la Facultad de Informática. El objetivo principal es desarrollar una nueva capa sobre el algoritmo implementado que ayude a subsanar la falta de explicación sobre las predicciones, una de las principales carencias de las técnicas de Inteligencia Artificial en la actualidad. Para lograrlo, se necesita un procedimiento unificado de explicación de anomalías en un entorno de variables mixtas discretas y continuas que proporcionen una salida de datos útil al supervisor mediante representaciones gráficas e informes sobre las detecciones. Para el desarrollo del algoritmo se propone un método en dos etapas, comprendidas en una explicación global previa a detección (pre-hoc) y una explicación de cada uno de los patrones etiquetados como anomalía por el modelo (post-hoc). Para la primera de ellas se desarrolla un árbol de decisión para la deter-

minación de caminos relevantes y la separación del conjunto de datos de entrada. El modelado previo a detección se centra exclusivamente en las variables continuas de los patrones, y es capaz de determinar conjuntos de valores relevantes para la explicación mediante la selección de divisiones del árbol, siguiendo un proceso de poda y filtrado de los mejores nodos. Para el caso posterior a detección, se determinan posibles explicaciones de forma individual a cada uno de los patrones etiquetados mediante un sistema de reglas adaptado tanto al modelo de variables continuas como al modelo de variables categóricas. Al tratarse de un algoritmo de detección de anomalías en espacio mixto categórico-continuo, es de especial interés facilitar una explicación que englobe ambos tipos de variables, pudiendo discernir en cada caso si el etiquetado como anomalía se debe al modelo de entrenamiento para la parte continua o para la parte categórica. En otros casos dicha información puede dar indicios de si realmente se trata de un falso positivo. Además, también se busca profundizar en el estudio de la separación de los datos continuos con el aumento de la dimensión del conjunto [9] y en el uso de diferentes criterios para explicación de anomalías atendiendo a las implicaciones de los modelos de aprendizaje continuos y categóricos propuestos [8].

En el segundo capítulo se exponen los principios básicos del algoritmo ADMNC, centrándose especialmente en los modelos utilizados para cada uno de los tipos de variables y prestando especial atención al proceso de ranking mediante estimadores. El tercer capítulo expone la metodología seguida en el transcurso de este proyecto. En el cuarto capítulo se trata el núcleo principal del trabajo, centrándose en los métodos seguidos para la explicación de patrones. El quinto capítulo muestra los resultados obtenidos sobre algunos conjuntos de datos conocidos y extendidos en técnicas de aprendizaje máquina, adaptados al caso de la detección de anomalías. Se ha documentado el proceso utilizando ejemplos visuales y desglosando en detalle cada una de las pruebas realizadas. El sexto capítulo resume la planificación y los costes asociados al proyecto. El penúltimo capítulo se centra en las principales conclusiones que se han obtenido. Por último, se esboza el posible trabajo futuro sobre el algoritmo de explicación propuesto, así como de las principales alternativas que se podrían abarcar para mejorar los resultados obtenidos.

Capítulo 2

# Detector de anomalías ADMNC

## 2.1 Descripción y formulación básica

El algoritmo ADMNC [8] es un método ideado para el aprendizaje *offline* de patrones con el objetivo de detectar anomalías en un espacio mixto categórico-continuo de variables, empleando para ello dos modelos paramétricos configurables con una visión probabilística del dominio. Es un procedimiento escalable a grandes cantidades de datos que emplea dos estimadores: uno relacionado con un modelo de mezcla de gaussianas para tratar la parte continua de las variables y otro relacionado con un modelo de regresión logística para tratar la parte categórica de los datos. Es un algoritmo con una formulación que tiene como base una medida de probabilidad conjunta. Para la determinación de los mejores parámetros del modelo se lleva a cabo un ajuste de valores mediante una función de máxima verosimilitud con optimización de gradiente descendente. Al algoritmo se le presentan conjuntos de datos de entrada $D$ con variables tanto categóricas como continuas, con el objetivo de ajustar los parámetros del modelo a los patrones en condiciones normales durante entrenamiento:

$$D = \{(x_0, y_0), ..., (x_{|D|}, y_{|D|})\}. \tag{2.1}$$

Donde $x_i$ es la parte continua e $y_i$ la parte categórica de cada vector de datos del conjunto $D$. Posteriormente se asigna a cada patrón un valor numérico a modo de estimador que, si no alcanza un determinado umbral, implicará su etiquetado como anomalía por el algoritmo. Para simplificar el modelo y su aprendizaje teniendo en cuenta la naturaleza de ambos tipos de variables, se lleva a cabo una factorización de la función de densidad de probabilidad (pdf) de la forma:

$$P(y, x) = P(y|x) * P(x). \tag{2.2}$$

Donde $P(y|x)$ es la probabilidad de las variables categóricas condicionadas a las continuas y $P(x)$ la probabilidad marginal de las variables continuas. Esto nos permite trabajar en

ambas partes del modelo de forma paralela e independiente, teniendo siempre en cuenta la interacción de ambas componentes.

## 2.2 Modelos utilizados

Para el desarrollo del modelo paramétrico, se lleva a cabo un proceso de aprendizaje paralelo entre los dos métodos principales, aplicados cada uno de ellos a uno de los tipos de variables.

### 2.2.1 Modelo de mezcla de gaussianas (GMM)

Para el caso de la parte numérica del conjunto de patrones, se utiliza un modelo de mezcla de gaussianas (en adelante GMM, por las siglas en inglés de Gaussian Mixture Model), junto con un proceso de ajuste de parámetros mediante máxima verosimilitud, aplicando un proceso de *Expectation–Maximization* [10]. El objetivo principal es entrenar los parámetros de cada gaussiana para ajustarlas al grueso de los datos normales. De esta forma conseguimos distribuciones gaussianas adaptadas a los patrones (con medias $\mu$ y desviaciones $\sigma$ de dimensión igual al número de características de entrada y de acuerdo a una matriz de covarianzas). Para tratar la inicialización del modelo, se efectúa un entrenamiento previo mediante el algoritmo de agrupamiento *k-means* sobre un subconjunto de los datos. Se asignan unos primeros valores a las medias de las gaussianas según los valores de los centroides calculados por dicho algoritmo. De la misma forma se utilizan las desviaciones de los clusters para inicializar la matriz de covarianzas del modelo.

### 2.2.2 Modelo de regresión logística (LR)

Para tratar las variables categóricas de cada patrón, se utiliza un modelo de regresión logística que utiliza la codificación *One-Hot Encoding* [11] para cada uno de los vectores categóricos, de forma que quedaría representado de la siguiente manera:

$$\boldsymbol{y} = (y^0, \dots, y^k), y^j \in \{0, 1\}. \tag{2.3}$$

Con esta nueva representación, la probabilidad del vector categórico condicionada al vector continuo, $P(y|x)$ quedaría expresado en los siguientes términos:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \prod_{j=0}^{k} P(Y = y^j|(\boldsymbol{x}, \boldsymbol{m}_j), \boldsymbol{w}) \tag{2.4}$$

Donde el valor $\boldsymbol{m}_j$ es la representación *One-Hot Encoding* de $j$ y $\boldsymbol{w}$ el vector de parámetros a aprender sobre el conjunto de datos. Una vez obtenido el vector $\boldsymbol{w}$ del modelo de LR, se

computa cada término 2.4 (página 6) de la siguiente forma:

$$P(Y = y^j|(\boldsymbol{x}, \boldsymbol{m}_j), \boldsymbol{w}) = \frac{1}{1 + e^{-(2y^j-1)\langle \boldsymbol{w},(\boldsymbol{x},\boldsymbol{m_j})\rangle}} \tag{2.5}$$

El objetivo final será por lo tanto la búsqueda de parámetros óptimos para ambos modelos. Por un lado, se tendrá en cuenta la probabilidad categórica condicionada a continua $P(y|x)$ y por otro la probabilidad marginal continua $P(x)$. El vector de pesos $\boldsymbol{w}$ del modelo logístico se ajustará por descenso de gradiente estocástico [12] (SGD, de sus siglas en inglés, Stochastic Gradient Descent), mientras que el ajuste de parámetros de las gaussianas se realizará mediante *Expectation-Maximization*.

### 2.2.3 Ajuste de parámetros por máxima verosimilitud

Para la obtención de los parámetros del modelo, se lleva a cabo un proceso de ajuste de máxima verosimilitud logarítmica sobre el conjunto de datos $D$, de forma que la expresión de optimización quedaría de la siguiente forma:

$$logL(D) = \sum_{i=1}^{|D|} logP(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}) + \sum_{i=1}^{|D|} logP(\boldsymbol{x}_i) \tag{2.6}$$

Ambos sumandos se pueden tratar de forma independiente y paralela, permitiendo mejorar en eficiencia y velocidad. Esto es gracias a la arquitectura de desarrollo *Apache Spark* [1] y al uso de varios núcleos computacionales, agilizando notablemente el proceso de ajuste de los modelos. Para el caso categórico, el aprendizaje de los valores del vector de pesos $\boldsymbol{w}$ se realiza con SGD con actualización y ajuste utilizando *minilotes* [13]. La idea radica en utilizar pequeños grupos o lotes de patrones para realizar paso a paso una actualización del vector de pesos hasta la convergencia o hasta alcanzar un límite de iteraciones. El proceso de cómputo necesario para cada lote puede ser paralelizado gracias al *framework* de desarrollo.

## 2.3 Entrenamiento

El objetivo del proceso de entrenamiento es ajustar los parámetros del modelo para, posteriormente, poder calcular los estimadores de la parte categórica y continua de cada uno de los patrones. Los parámetros de la mezcla de gaussianas, así como el vector de pesos $\boldsymbol{w}$ de la parte categórica se adaptan para proporcionar valores de estimación altos sobre datos reconocidos como normales, mientras que aquellos datos detectados como anomalía obtendrán valores de estimador global bajos (En cualquier caso, la anomalía puede ser debida a ambos tipos de datos o con un mayor peso de uno de ellos).

[1] Se proporciona detalle del *framework* de desarrollo, así como de sus ventajas, en el anexo correspondiente.

Para llevar a cabo dicho proceso, se obtiene el conjunto de patrones etiquetados (como dato normal o anómalo) que se separa en conjunto de entrenamiento $E$ y test $T$, mediante una división pseudo-aleatoria de los datos en la clásica proporción 70-30. Este conjunto de entrenamiento, tras ser filtrado para que esté compuesto exclusivamente por vectores normales, se utilizará para ajustar los parámetros de ambos modelos. En la figura 2.1 (página 8) se puede observar de forma gráfica el proceso seguido. En el caso del modelo de variables categóricas, los parámetros se ajustarán utilizando SGD. En cada uno de los pasos de la optimización, el cómputo del gradiente se realizará de manera paralela sobre pequeños lotes, acelerando el proceso. Por otra parte, para aprender los parámetros de la mezcla de gaussianas, se inicializará el modelo tomando como medias los centroides de un modelo *K-Means* aprendido sobre una pequeña muestra del conjunto de entrenamiento. La matriz de covarianzas se inicializará con las desviaciones típicas de los clusters. Tras esta inicialización, el aprendizaje del modelo de la parte continua se delegará en la implementación de mezcla de gaussianas disponible en *Apache Spark*, que utiliza EM.

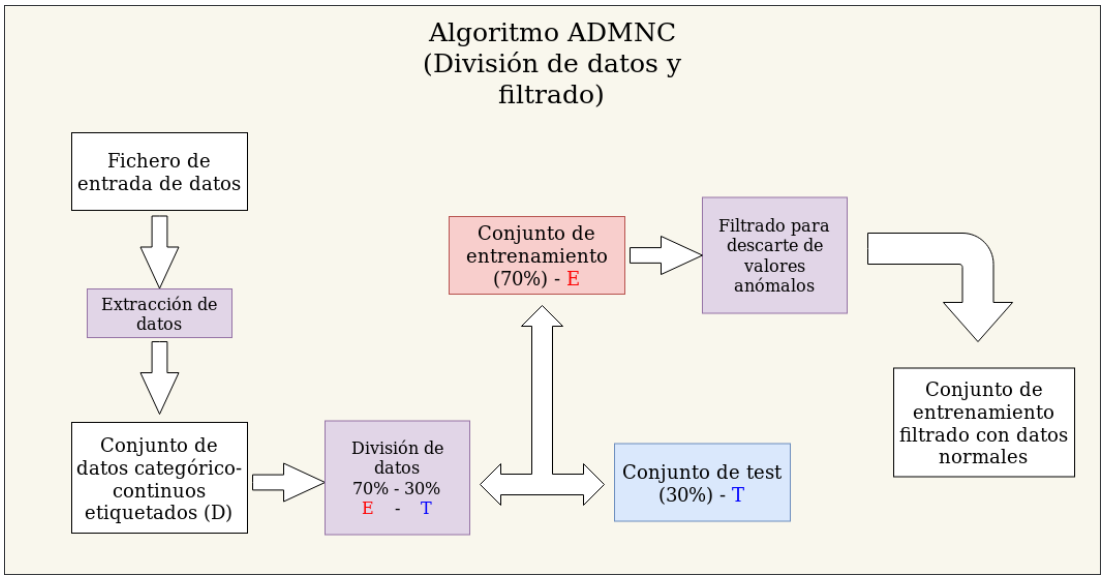



Figura 2.1: Proceso de división y filtrado de datos previo a entrenamiento

## 2.4 Detección

Una vez se tiene el modelo completamente entrenado en ambas partes, se puede obtener el estimador global a cada nuevo patrón del conjunto de test $T$ mediante el siguiente cálculo:

$$Estimator((x_i, y_i) \in T) = Log_e(LRe(y_i) * GMMe(x_i)) \tag{2.7}$$

Donde $LRe(y_i)$ corresponde al estimador logístico del cálculo 2.4 (página 6), desglosado

el término en la fórmula 2.5 (página 7). $GMMe(x_i)$ hace referencia al estimador de la parte continua del patrón que se corresponde al cálculo sobre las gaussianas del modelo:

$$GMMe(x_i) = \sum_{j=0}^{n} pdf(g_j, x_i) \quad / \quad g_0, ..., g_n \in G, x_i \in T \tag{2.8}$$

Donde, $pdf$ hace referencia a la función densidad de probabilidad de la j-ésima gaussiana $g_j$, siendo $T$ el conjunto de datos de test.

Con los cálculos de los estimadores sobre el conjunto de entrenamiento $E$ (sin anomalías), se efectúa un ordenamiento de menor a mayor por dicho estimador global y posteriormente se toma un subconjunto de los valores más bajos teniendo en cuenta la proporción especificada por el hiperparámetro de ratio de anomalías (En general, en la mayoría de las aplicaciones, el usuario puede fijar una proporción a priori de anomalías, que es la que se usa como ratio en este parámetro). De esta forma el umbral detector de anomalías se corresponde con el mayor valor de estimador sobre los elementos de dicho subconjunto. El proceso de entrenamiento, así como el cálculo del umbral detector se pueden observar en el diagrama de la figura 2.2 (página 9). Para detectar nuevas anomalías se realiza su cálculo de estimador y se comprueba si dicho valor es inferior al valor del umbral fijado anteriormente; en caso afirmativo se trataría como una detección positiva de anomalía.

El funcionamiento general de la detección de anomalías en el algoritmo, de forma esquemática simplificada, queda expuesto en la figura 2.3 (página 10).

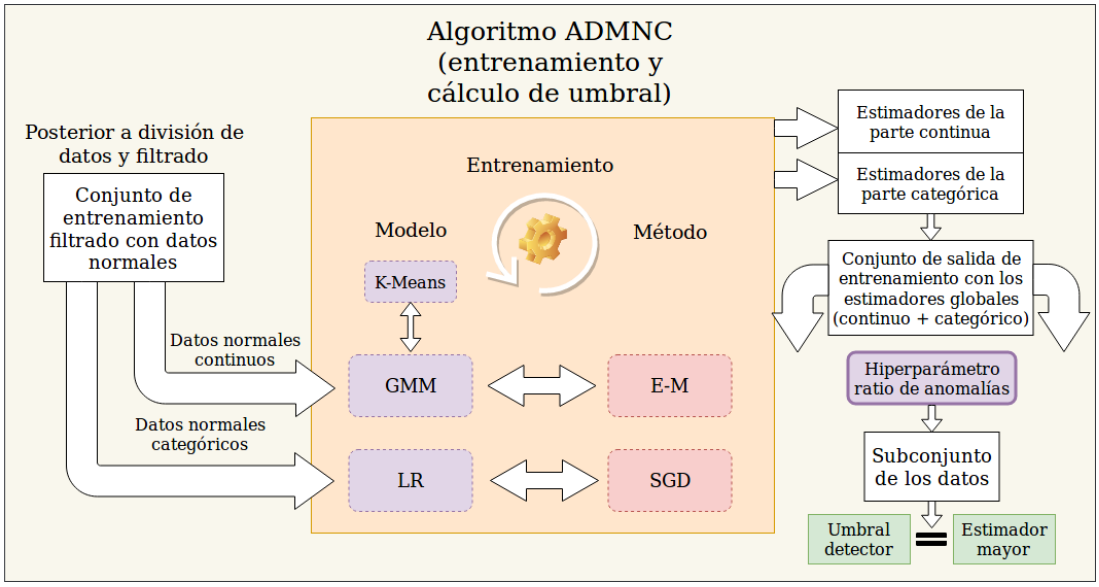



Figura 2.2: Proceso de entrenamiento y cálculo de umbral detector en ADMNC

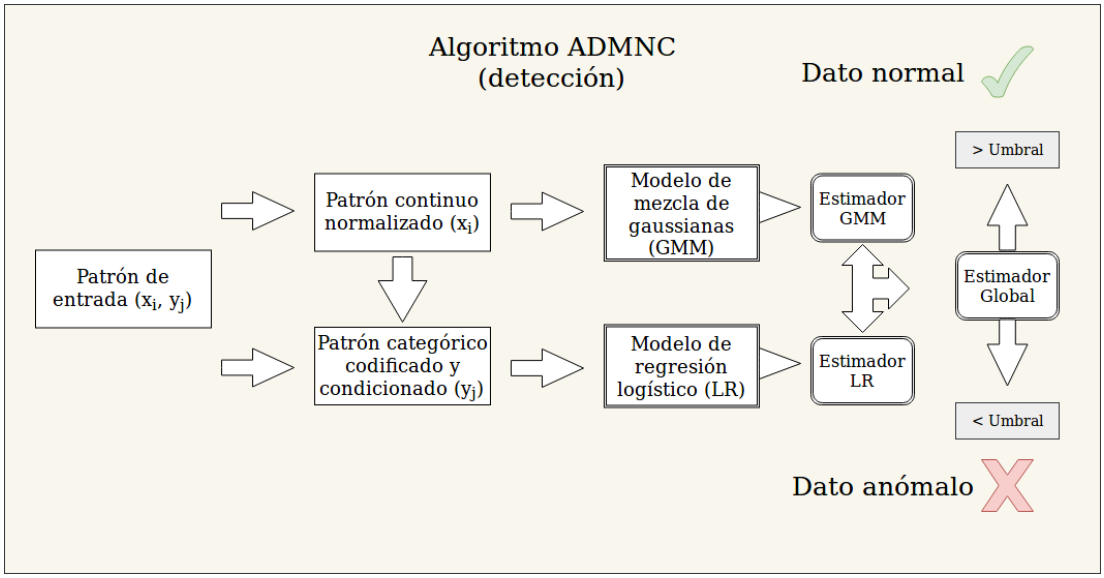



Figura 2.3: Proceso de detección de nuevas anomalías posterior a entrenamiento (offline)

Capítulo 3

# Metodología de desarrollo

PARA el desarrollo de este proyecto se optó por un ciclo de desarrollo de software iterativo clásico o incremental [14]. La razón de ello fue haber definido varios entregables con funcionalidad progresiva en las sucesivas revisiones (tanto semanales como mensuales) con los tutores del proyecto. El objetivo principal del trabajo es elaborar un algoritmo de explicación a modo de capa software adicional sobre el algoritmo ADMNC [8]. La frontera entre iteraciones corresponde a cada uno de los módulos de explicación necesarios sobre los patrones de entrada, así como a incrementos adicionales tanto para el estudio inicial del algoritmo y de análisis de requisitos, como para el refinamiento posterior mediante integración de funcionalidades sobre el producto software final.

A este ciclo de desarrollo se le suma el control del proyecto con las correcciones que consideren oportunas los tutores, y para ello se dispone de un repositorio privado para revisiones mediante control de versiones en *GitFic*, utilizando la infraestructura proporcionada por los Servicios Informáticos de la Facultad de Informática. El objetivo principal es facilitar un seguimiento eficaz sobre los cambios diarios en los códigos, así como de los posibles cambios de requisitos a la explicación a medida que se avanza en las iteraciones.

## 3.1 Iteraciones

Para avanzar paso a paso en funcionalidad sobre el algoritmo de explicación, se siguieron los siguientes entregables o incrementos que se sometieron semanalmente a las revisiones de los tutores.

### 3.1.1 Iteración 1 - Refinamiento del algoritmo ADMNC, pruebas y evaluación de métricas.

La primera de las iteraciones abarcaría el estudio del algoritmo existente, el análisis minucioso de su funcionalidad interna en cada uno de sus módulos y la revisión de código en busca

de errores que pudieran permanecer con posterioridad a su elaboración. Es una de las etapas más importantes, ya que se interiorizan los fundamentos matemáticos y estadísticos que serán necesarios para el desarrollo del algoritmo de explicación, buscando aportar transparencia al proceso interno seguido. En esta etapa se repiten los experimentos automatizados ya elaborados previamente por el grupo de desarrollo del algoritmo, así como algunas correcciones y arreglos relacionados con sus transformaciones de datos y fórmulas.

#### Análisis del algoritmo existente

En esta fase se llevó a cabo el estudio completo de los módulos de detección de anomalías, contemplando tanto los módulos de soporte (transformación de datos, entrada y salida...) como los principales (modelo logístico de procesado de categóricas y modelo de mezcla de gaussianas de procesado de continuas).

#### Corrección de errores

Sobre el estudio previo del algoritmo se efectúan correcciones a su uso, destacando la corrección del intervalo de normalización para el caso de características continuas y el ajuste de la fórmula de aprendizaje.

#### Paso de test existentes

Para comprender mejor el algoritmo se utilizaron los test proporcionados para comprobar el funcionamiento del mismo. Para cada caso de prueba se utilizó la información proporcionada por el *AUROC (Area Under Receiver Operating Characteristic Curve)*, y de otras métricas típicas en técnicas de *Machine Learning* (tasa de errores global, tasa de verdaderos positivos (TVP), tasa de verdaderos negativos (TVN), tasa de falsos positivos (TFP) y tasa de falsos negativos (TFN)).

### 3.1.2 Iteración 2 - Desarrollo de explicación del vector continuo en el conjunto de datos

En este apartado se elabora la parte de la explicación relacionada con el segmento del vector de datos continuo atendiendo al modelo de mezcla de gaussianas 2.2.1 (página 6) proporcionado en el algoritmo de partida. El objetivo principal es desglosar y emplear los componentes de aprendizaje paramétrico de cada gaussiana del modelo para poder utilizarlos en un nuevo algoritmo de explicación para segmentación, separación de datos y explicación individual de anomalías.

#### Análisis del método de mezcla de gaussianas (GMM)

En este punto se busca profundizar en el método de aprendizaje paramétrico del conjunto de gaussianas del modelo continuo de datos, con el objetivo de comprender el proceso de entrenamiento y el cálculo de su propio estimador, pudiendo utilizar dicha información para proporcionar indicaciones acerca de la parte continua de los datos. En este aspecto se tratan 2 tipos de explicaciones sobre la información de los patrones. Por un lado la *Pre-hoc* previa a detección, utilizando la información de las clases de las gaussianas ajustadas y por otro lado la *Post-hoc*, posterior a detección sobre cada dato de forma individual.

#### Diseño e implementación del algoritmo de explicación sobre la parte continua de los datos

Se elabora un método unificado para la explicación de la parte continua de los patrones 4.2 (página 18), atendiendo a la parte *Pre-hoc* para el estudio del reparto de clases sobre el modelo de gaussianas, así como de la parte *Post-hoc* para el tratamiento de anomalías mediante un sistema de reglas automatizado.

#### Pruebas sobre el algoritmo desarrollado

Especialmente se trataron los conjuntos de datos especificados en la sección 5 (página 31) para comprobar el correcto funcionamiento de la parte implementada.

### 3.1.3 Iteración 3 - Desarrollo de explicación del vector categórico en el conjunto de datos

En esta iteración se efectúa la parte de explicación relacionada con el conjunto categórico de variables de los patrones, atendiendo al modelo de regresión logística (LR) 2.2.2 (página 6) proporcionado en el algoritmo de partida. Se utiliza la estructuración de la formulación interna de aprendizaje categórico del modelo para poder añadir la funcionalidad relativa a la explicación categórica de variables, con su condicionamiento a la parte continua cuando sea posible.

#### Análisis del método de regresión logístico (LR)

Se pretende ahondar en el método de aprendizaje del modelo categórico de datos, atendiendo al ajuste de parámetros del método logístico sobre el vector de pesos $\boldsymbol{w}$, así como del método de cálculo de su estimador, con el objetivo de utilizar dicha información a modo de detección de variables causantes de un estado anómalo del patrón. En este apartado solo se

trata un tipo de explicación, la *Post-hoc* posterior a detección. Esto es debido a que el modelo categórico está orientado a facilitar una explicación sobre sus predicciones y no sobre su proceso de ajuste durante el entrenamiento, a diferencia del caso continuo.

#### Diseño e implementación del algoritmo de explicación sobre la parte categórica de los datos

Se elabora un método unificado para la explicación de la parte categórica de los patrones 4.2 (página 18), atendiendo exclusivamente al método *Post-hoc* para el conjunto de anomalías detectadas, mediante un sistema de filtrado y selección de estimadores categóricos bajos.

#### Pruebas sobre el algoritmo desarrollado

De la misma forma que en la sección 3.1.2 (página 13), se busca comprobar el correcto funcionamiento de la parte implementada.

### 3.1.4 Iteración 4 - Integración y refactorización, algoritmo de explicación final

Se trata del incremento final, que tiene por objetivo combinar los resultados de explicación de ambas partes: la explicación *Pre-hoc* de la parte continua y la *Post-hoc* tanto de la parte continua como de la categórica, todo ello en un método global unificado. El esquema de la metodología de desarrollo empleada viene indicado en la figura 3.1.

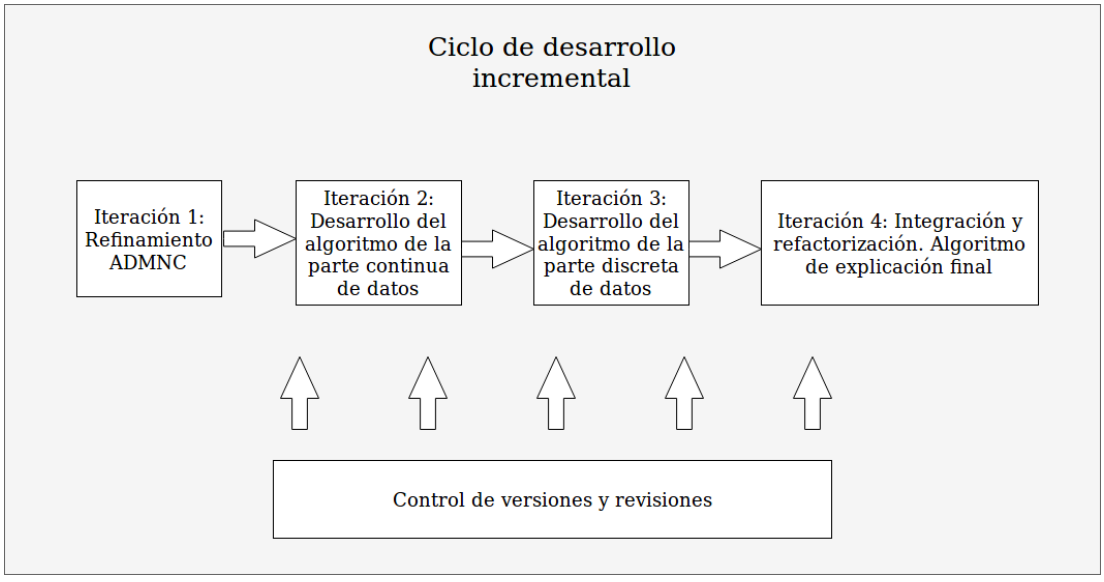



Figura 3.1: Metodología: ciclo de desarrollo incremental seguido en este Trabajo Fin de Grado

Capítulo 4

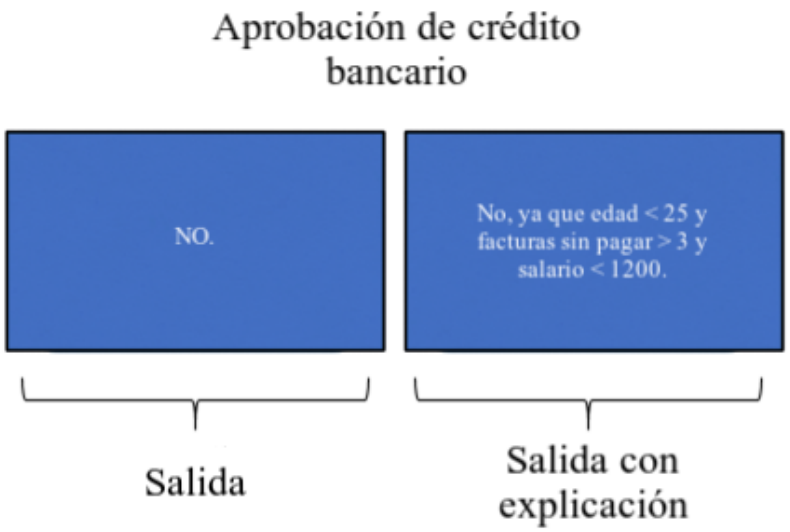
# Algoritmo de explicación de anomalías

## 4.1 Fundamentos y principios básicos

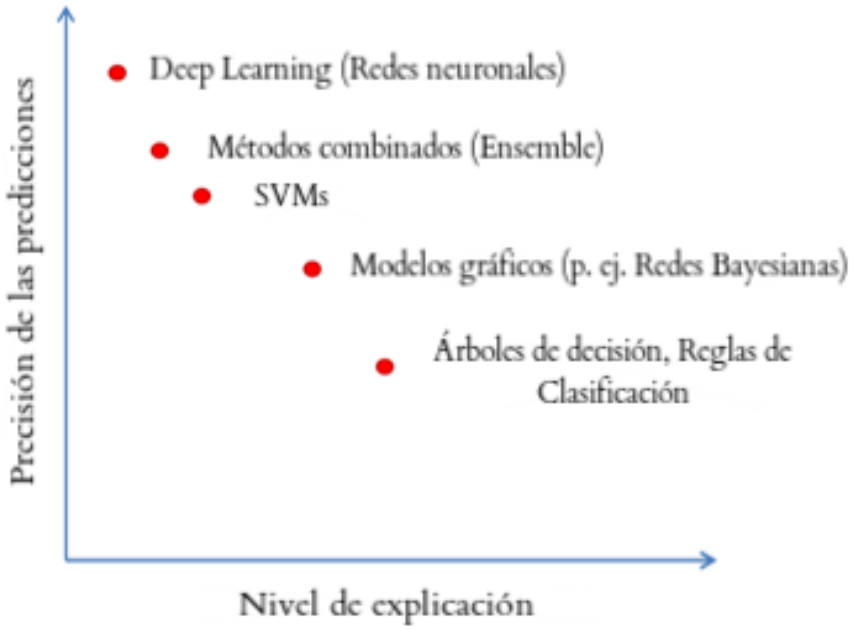
La escasa o inexistente explicación que proporcionan los algoritmos sobre los resultados obtenidos es una de las principales carencias que hoy en día sufren las técnicas de aprendizaje automático [15]. Lo que en la actualidad se conoce como XAI (*Explainable Artificial Intelligence*) o EML (*Explainable Machine Learning*) aún no tiene una definición clara, pudiéndola entender no obstante como «Ciencia de la comprensión sobre lo que un modelo hizo o pudiera haber hecho». El deber de explicación surge de la necesidad de transparencia hacia el usuario debido ya no sólo a un aspecto social, sino también y entre otras cosas, a la entrada en vigor del nuevo Reglamento General Europeo de Protección de Datos (RGPD) [16], el cual regula de forma más estricta el uso de técnicas de *Machine Learning*, relacionado sobre todo con el control de datos personales y el deber de información. Específicamente, en lo dispuesto en el punto número 71 se indica que « *El interesado debe tener derecho a no ser objeto de una decisión (...) que evalúe aspectos personales relativos a él, y que se base únicamente en el tratamiento automatizado y produzca efectos jurídicos en él o le afecte significativamente de modo similar, como la denegación automática de una solicitud de crédito en línea o los servicios de contratación en red en los que no medie intervención humana alguna (...) Dicho tratamiento debe estar sujeto a las garantías apropiadas, entre las que se deben incluir la información específica al interesado y el derecho a obtener intervención humana, a expresar su punto de vista, a recibir una explicación de la decisión tomada después de tal evaluación y a impugnar la decisión* » [17].

Este nuevo derecho a la explicación va en busca de una Inteligencia Artificial más transparente y ética, donde cada toma de decisiones del algoritmo pueda ser justificada, obviando el uso a modo de «caja negra» habitual hasta el momento. En la figura 4.1a [18] (página 16) se ve un ejemplo ilustrativo de lo que se busca con este nuevo reglamento. Donde antes una

salida válida pudiera ser una respuesta afirmativa o negativa (o en este caso, una detección positiva o negativa sobre anomalía), ahora se busca una explicación sobre el proceso interno seguido, mediante un desglose de información de las variables involucradas en esta toma de decisión.




(a) Explicación posible a una toma de decisión de un algoritmo de IA




(b) Balance explicación-predicción de varios métodos de IA

Figura 4.1: Nociones a la explicación frente a la predicción en IA

Este objetivo es alcanzable mediante técnicas de aprendizaje automático como árboles de decisión o reglas de clasificación, que es lo que se propone en la sección 4.2 (página 18). Sin embargo, en determinadas situaciones y dependiendo de la complejidad interna del algoritmo, este objetivo es difícil de conseguir, sobre todo por la representación interna del conocimiento o por el uso de los métodos de aprendizaje profundos (*Deep Learning*), que resulta en los conocidos algoritmos de «caja negra». También hay que tener en cuenta el compromiso existente entre la precisión de la predicción y las capacidades explicativas de cada método 4.1b [18] (página 16) y en cada caso se tiene que elegir aquel que se adapte mejor al problema propuesto. No obstante queda expuesta la carencia de transparencia de los métodos de aprendizaje profundos [19], que a pesar de su gran precisión o fiabilidad, no se adaptarían en su estado actual al nuevo reglamento impuesto por la Unión Europea. Además, aquellos algoritmos que proporcionan explicación tienen también otras ventajas, como son la capacidad de identificar sesgos en los datos, evaluar la generalidad de los mismos, extraer información y permitir análisis de hipótesis.

El tratamiento *Post-Hoc* es el más extendido en los entornos de explicación de algoritmos. Dado un modelo entrenado y un conjunto de datos de test, se presenta una predicción para cada uno de ellos y sobre estas salidas se acopla el módulo de explicación, tal y como se indica en la figura 4.2 [18]. Hay que tener en cuenta que se trata de una rama de investigación bastante

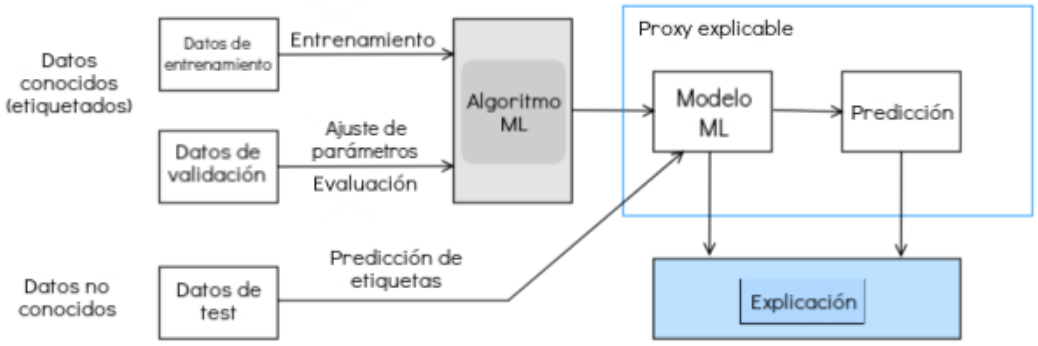



Figura 4.2: Explicación Post-Hoc genérica sobre las predicciones de un modelo entrenado

reciente, en la que todavía no se contemplan formas estándar de representación de las salidas, ni se dispone de las métricas necesarias para establecer comparativas entre distintos métodos. Existe gran interés en el desarrollo de un *framework* EML para explicación algorítmica, teniendo como características principales la justicia, la privacidad, la fiabilidad, la robustez y la causalidad. La misma agencia del Departamento de Defensa de Estados Unidos para el desarrollo de proyectos de investigación avanzados (DARPA) tiene una línea de investigación

reciente en XAI [20] donde se abarca la problemática comentada en este apartado. El objetivo a medio plazo propuesto en dicho programa es la creación o la modificación de técnicas de aprendizaje máquina para su adaptación a nuevos modelos de explicación. Se busca integrarlos con los avances en interfaces persona-máquina con el objetivo final de lograr un diálogo o interacción más clara y natural con el usuario final. Estas capacidades se consideran como una de las piezas fundamentales de lo que se conoce como la «tercera ola de la Inteligencia Artificial»

## 4.2 Método propuesto

Para el caso de la detección de anomalías y centrándonos en el algoritmo ADMNC [8], se proponen dos tipos de explicaciones para abarcar la problemática de la transparencia de los algoritmos. Por un lado se presenta una explicación *Pre-Hoc* previa a detección, cuyo objetivo es reutilizar las variables internas del modelo de aprendizaje para poder obtener información acerca de la segmentación o partición de los datos. Se centra en la parte continua de los patrones, ya que la naturaleza del modelo de mezcla de gaussianas (GMM) nos permite generar un conjunto de datos etiquetados gracias al método de aprendizaje paramétrico de cada una de ellas, de forma que a cada nuevo patrón le podemos asignar su clase según la gaussiana con mayor probabilidad de pertenencia de acuerdo a la $pdf$, o función de densidad de probabilidad. Ese conjunto de datos etiquetados gracias a la información proporcionada por el modelo nos permitirá utilizar una técnica de aprendizaje que tiene características favorables a la explicación, siendo este el caso de los árboles de decisión para clasificación.

Por otro lado se presenta una explicación *Post-hoc* para ambos tipos de datos, es decir, tanto de la parte continua como categórica del patrón. Esta explicación sigue la idea expresada en la figura 4.2 (página 17), con el valor añadido de que se trabaja de forma dual y complementaria con datos de dos naturalezas distintas y se busca una solución que integre ambas partes. Para el caso del conjunto continuo de datos se utiliza un sistema de reglas sobre el modelo de gaussianas, con el soporte adicional del modelo de aprendizaje del árbol de decisión. Para el caso de la parte categórica, el objetivo principal es deshacer la formulación interna del algoritmo para buscar la explicación en términos de uno o varios componentes que hicieran disminuir el valor del estimador categórico. Cuando sea posible, también se puede obtener la información del condicionamiento o relación existente con la parte continua del vector, otorgando aún más información sobre el proceso seguido. El algoritmo final de explicación unificado se puede observar en la figura 4.3 (página 19).

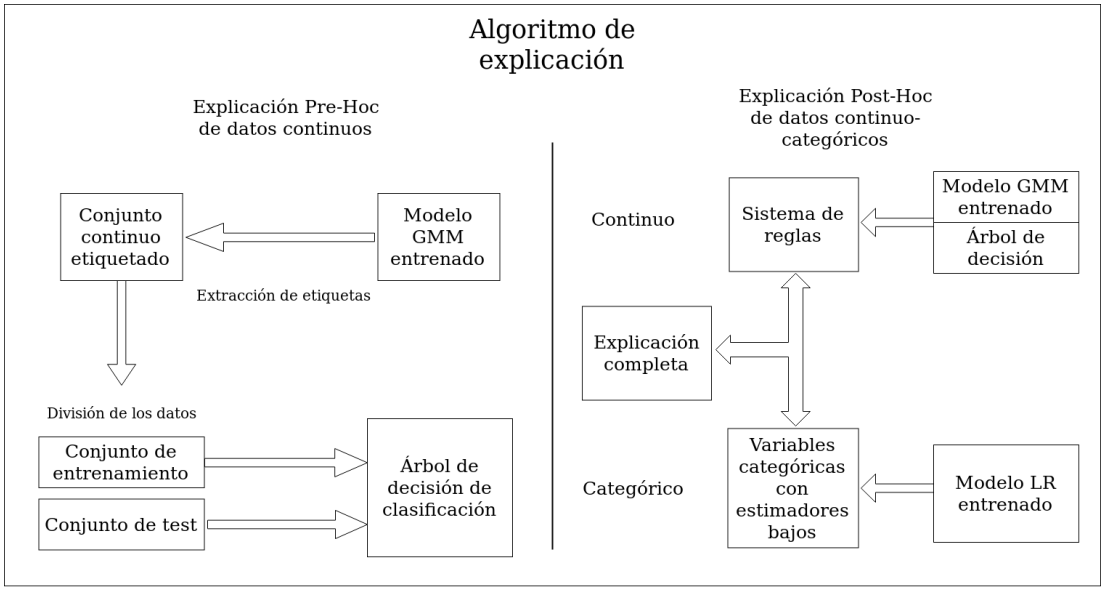



Figura 4.3: Algoritmo de explicación de anomalías propuesto

### 4.2.1 Explicación pre-hoc

Partimos del conjunto de datos continuo con las etiquetas extraídas del modelo de gaussianas. Con estos patrones se lleva a cabo una división aleatoria en la proporción 70-30 para generar los conjuntos de entrenamiento y de test. El árbol de decisión aprende dicho conjunto de entrenamiento indicando como máximo número de clases el número de gaussianas del modelo de detección de anomalías y utilizando exclusivamente la parte continua de los patrones. Como criterio de impureza para la división de nodos se emplea la medida estándar de entropía. Esta métrica se puede entender como una medida de la falta de certeza o del desorden sobre un conjunto de ejemplos, esto es, da una idea de cómo de mezclados se encuentran los datos de las diferentes clases. El objetivo principal de estos árboles es encontrar las divisiones de variables que permitan separar de la mejor forma posible los datos de las diferentes clases, para que en sucesivas divisiones, cada nodo sea más representativo de una de ellas. En cada paso se busca el umbral de característica que minimice lo mejor posible el valor de la entropía. Además, se restringe a cuarenta el número de posibles divisiones a elegir sobre una característica seleccionada en un paso de división. La máxima profundidad del árbol está especificada a un nivel cinco para no complicar la explicación de variables con caminos demasiado complejos, considerando en todo momento el compromiso entre calidad de separación y calidad de explicación.

Con el modelo del árbol entrenado, ahora se tiene la información suficiente para realizar predicciones sobre nuevos datos, que consistirían en realizar recorridos por dicho árbol hasta llegar a un nodo hoja que proporcione la información de certeza sobre la pertenencia a una u

otra clase. No obstante, el interés aquí recae en los recorridos hacia los nodos con conjuntos de datos mejor segmentados y más representativos. El proceso a seguir sobre este modelo es el de búsqueda de mejores nodos y de extracción de caminos en las divisiones sucesivas que nos aporten una noción a la explicación *Pre-hoc* sobre el conjunto de datos. Se buscan grupos de variables reducidos (caminos cortos) que expliquen de la forma más clara posible el mayor número de clases sobre el conjunto de patrones. Para ello se necesita obtener información acerca del reparto de datos en cada nodo, por lo que se efectúa el recorrido de cada uno de ellos por el árbol teniendo en cuenta los valores de las características continuas. Se necesita la información sobre el identificador del nodo, la densidad de patrones totales (sin diferenciar clase), la clase de mayor predicción en el nodo (con su valor de confianza), la medida de impureza del mismo y la proporción de patrones de cada una de las clases de forma separada. Una idea de la información a mostrar se puede visualizar en la figura 4.4 (página 20). Para efectuar la selección de los mejores nodos, realizamos una búsqueda en anchura por el árbol gestionando listas de exploración y de selección/filtrado de nodos, siendo necesario cumplir en su totalidad los siguientes criterios heurísticos (como puede verse en la Figura 4.5a de la página 22), que fueron seleccionados dados los buenos resultados obtenidos:


Figura 4.4: Información de interés acerca de un nodo del árbol de decisión. De arriba a abajo y de izquierda a derecha: Identificación del nodo (en rojo), densidad de patrones sobre el total del conjunto (en azul), clase de mayor predicción sobre el nodo (en azul), probabilidad de la clase (en azul), valor de impureza del nodo (en verde) y valores de densidad de patrones por clase (en naranja).

- Para que un nodo hijo sea seleccionado respecto a su padre, debe reducir el valor de impureza como mínimo al 50% con respecto a su progenitor, es decir, debe hallar una división lo suficientemente buena como para mejorar notablemente la separación de los datos de las diferentes clases, siendo así más representativo de una de ellas.

- Además, dicho nodo con esa reducción de impureza debe contemplar un buen valor de certeza acerca de la predicción sobre la clase mayoritaria, con una probabilidad igual o superior al 80%. Esto interesa ya que para que un camino sea representativo de una clase, el nodo final seleccionado de esa ruta debe tener un valor de predicción alto para esa clase, con certeza suficiente como para considerarlo representativo de ese grupo.

- Por otro lado, el nodo debe tener una densidad total lo suficientemente significativa respecto al conjunto de datos, o en su defecto, debe ocurrir que la densidad de patrones

de la clase mayoritaria sea lo suficientemente grande como para que el nodo pueda caracterizar a esa clase. Este último criterio busca que el nodo seleccionado cumpla con un mínimo de datos para que el camino de variables continuas sea lo suficientemente representativo como para considerar relevante esa explicación.

Para el caso de eliminación o descarte de nodo, se tienen en cuenta dos casos diferentes. Por un lado, un criterio de eliminación general de nodos y por otro un criterio de descarte atendiendo a la relación padre-hijo sobre la clase de mayor predicción. Para el criterio general de eliminación de nodos se necesita cumplir con al menos uno de los siguientes puntos (ver figura 4.5b en página 22):

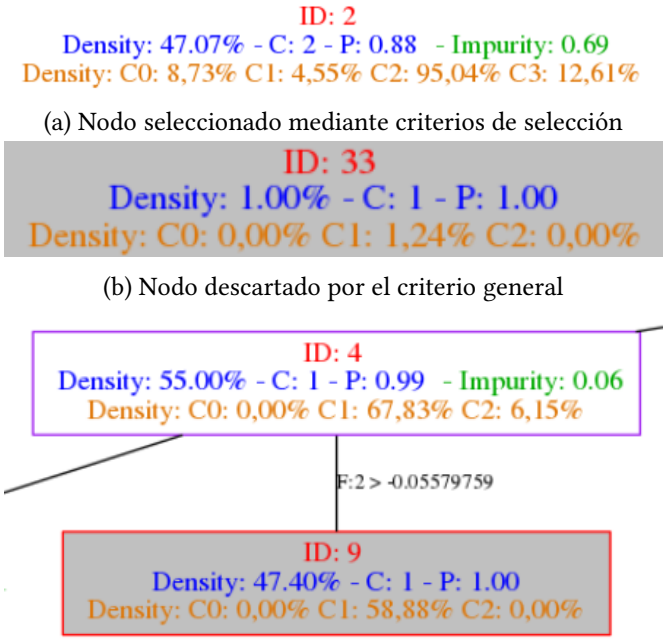
- Cuando un determinado conjunto de datos es difícilmente separable y la división mantiene un valor semejante o incluso peor de impureza, el nodo es descartado. Esto quiere decir que la división escogida en el método de aprendizaje del árbol fue deficiente, al no haber podido segmentar los datos.
- Aún habiendo reducido la impureza, si el nodo objetivo presenta un valor muy bajo de densidad total de patrones y además la densidad de su clase de mayor predicción es baja, el nodo es descartado. Esto quiere decir que aunque haya mejorado la división, el nodo no es representativo del conjunto de datos de una de las clases, por lo que tampoco aportaría explicación.

El caso de filtrado de nodos atendiendo a la relación padre-hijo se da cuando uno de ellos es seleccionado respecto a la división del nodo padre, pero se efectúan comprobaciones adicionales sobre la clase de mayor predicción de ambos (padre e hijo) para ver si el aumento en el alcance de la explicación compensa el aumento de complejidad asociado (ver figura 4.5c en la página 22). Este criterio tiene por objetivo eliminar los nodos hijos de la explicación si no se obtiene una clara ventaja sobre la definición de la clase mayoritaria. En este caso se descartaría si cumple las siguientes condiciones:

- Tanto el padre como el hijo tienen la misma clase de mayor predicción.
- La confianza de la predicción del padre ya es muy alta (mayor o igual al 90%).
- La impureza del hijo no se reduce lo suficiente respecto a la del padre, o en su defecto, la impureza del padre ya es lo suficientemente baja como para ser aceptable.
- La división en el árbol produce una disminución superior a un umbral sobre la densidad de la clase de mayor predicción. Esto supone un compromiso entre la calidad de separación de los datos y la cantidad de la clase de mayor predicción. A la hora de obtener una explicación prima un recorrido más sencillo teniendo algo menos de certeza pero garantizando una mayor cantidad de datos del conjunto representativo de dicha clase.

Finalmente, tras la selección de nodos y los dos tipos de filtrados, tenemos cuatro tipos de entes: los nodos seleccionados, los nodos filtrados genéricos, los nodos filtrados en la relación de clase padre-hijo y los nodos no seleccionados. El siguiente paso sería refinar la estructura del árbol mediante un post-filtrado, ya que los criterios de selección y eliminación de nodos mediante este recorrido en anchura pueden dejar caminos inconsistentes. De esta forma, debemos tener en cuenta los siguientes dos puntos:

- Si un nodo es seleccionado, su camino hacia la raíz no puede contemplar nodos eliminados. De ser el caso, se deben limpiar, dejándolos sin seleccionar para mantener un camino sin roturas.
- Si se cumple con el primer punto, y un nodo se ha eliminado, los sucesivos hijos del mismo deben ser filtrados con el mismo criterio. Esto asegura que un camino sea completamente eliminado cuando a la altura de un nivel $n$ ya se había descartado. El camino en sus sucesivos niveles ($n+1$, $n+2$, ... ) también es eliminado.




(a) Nodo seleccionado mediante criterios de selección

(b) Nodo descartado por el criterio general

(c) Ejemplo de filtrado de nodo en la relación padre-hijo respecto a la clase de mayor predicción

Figura 4.5: Ejemplos de selección y filtrado de nodos mediante exploración del árbol de decisión

Una vez llevado a cabo todo este proceso, tendremos la representación final del árbol de decisión para clasificación con el conjunto de caminos seleccionados y un desglose de las clases

explicadas del conjunto de datos normales. Para este método *Pre-hoc* se propone una representación gráfica con una tabla a modo de leyenda[1] que facilita la tarea de un supervisor humano a la hora de comprobar los resultados sobre las predicciones de las gaussianas. El resultado de explicación estaría comprendido por las rutas o caminos de variables continuas que definen un determinado conjunto de patrones de una clase sobre un nodo con las condiciones de selección indicadas en esta sección 4.2.1 (página 20). El resultado visual es muy intuitivo y la tabla actúa a modo de resumen sobre los resultados obtenidos. El supervisor puede obtener información sobre los caminos relevantes con su desglose de variables y a qué clase se está refiriendo. Se proporciona información sobre el porcentaje de clase que se explica y sobre la proporción total de datos con la que se trabaja. Además, también se indican recomendaciones para el ajuste de parámetros del modelo cuando el algoritmo detecta una explicación insuficiente por falta de razonamiento sobre las clases. El detalle de estas salidas viene indicado en el apartado correspondiente de resultados (sección 5.1, página 31).

### 4.2.2 Explicación post-hoc

Para el caso *Post-hoc*, se trabaja tanto con la parte continua como con la parte discreta de los datos, tal y como viene indicado de forma esquemática en la figura 4.3 (página 19). El objetivo de este método es tratar de forma paralela cada una de las partes para integrarla finalmente en una explicación que es posterior a la detección y se aplica sobre cada muestra detectada como anomalía por el algoritmo. El propósito de este método es proveer de una explicación en lenguaje natural que aporte de manera rápida y sencilla el detalle suficiente para saber qué componentes fueron los causantes de la detección como anomalía, sin necesidad de facilitar detalles más complejos de bajo nivel (o de la formulación interna de ADMNC). Sin embargo, además de recibir la explicación textual, el supervisor puede solicitar información de grano más fino sobre un ejemplo concreto, por lo que la explicación debe contemplar ambos casos de detalle. Para esta parte, el algoritmo genera 2 tipos de ficheros: uno en texto plano, con la información de más bajo nivel a efectos de depuración o paso a paso de razonamiento, y otro fichero con formato HTML estándar más visual para ser tratado directamente por el usuario. Estas dos salidas representan ambos niveles de detalle a la hora de realizar la explicación.

Una vez entrenado y ya en modo producción, el algoritmo ADMNC analiza los datos de entrada para realizar el cálculo de estimadores y determinar en consecuencia cuales de esos datos constituyen una anomalía. Sobre los resultados anómalos se efectúa una ordenación de menor a mayor por el valor del estimador global y es ese conjunto el que constituye la entrada al método de explicación aquí descrito. Tratando el caso continuo, principalmente se utiliza información facilitada por el modelo entrenado de gaussianas (GMM), aunque también se reutiliza cierta información del modelo de árbol de la explicación previa *Pre-hoc*. Esto se

[1] Véase anexo para detalle sobre *GraphViz* y el formato .dot

integra en un sistema sencillo de reglas de detección, además de proporcionar información adicional sobre el patrón que pudiera ser de utilidad a la hora de analizar la anomalía. Para el caso discreto o categórico, la explicación se centra más en la formulación planteada en la sección 2.2.2 (página 6) del modelo de regresión logística. En este caso se buscan los términos $y^j$ de la fórmula 2.4 (página 6) que hacen que el producto que define a $P(y|x, w)$ sea bajo. Dicho de otro modo, se descomponen los cálculos en busca de los términos *OneHotEncoding* $y^j$ de menor estimador categórico con el objetivo de decodificar su representación. Con esa transformación se llega al origen de la detección de la anomalía en función de una o varias de sus variables categóricas. Asimismo, también se pretende dar información sobre su condicionamiento continuo, siempre que sea posible.

Sobre el documento HTML de anomalías, se proporciona información al supervisor para conocer en cada caso el conjunto de datos que se está tratando. Se muestra una cantidad de patrones detectados como anómalos en la proporción especificada por el hiperparámetro de ratio de anomalías, que hacía referencia a la cantidad de resultados que quería visualizar el usuario. Para cada dato se indica la información acerca de su vector continuo y categórico, acompañado cada uno de ellos con su explicación en lenguaje natural. Una vez el supervisor comprende la razón de la detección, puede consultar información más detallada sobre los desgloses tanto de la parte continua como de la categórica. La estructura del informe se organiza en conjuntos de desplegables para mejorar la visualización y el ordenamiento de la información. Con el fichero en texto plano se busca proporcionar información adicional a modo de depuración del algoritmo, así como posibilitar la comprobación paso a paso del razonamiento. Normalmente no es necesaria su consulta ya que la información de interés del mismo se encuentra especificada en el documento HTML.

#### Post-hoc del vector continuo de datos

Tal y como se indicó previamente en la figura 4.3 (página 19), en este caso se utiliza la información proporcionada por el modelo entrenado de mezcla de gaussianas (GMM) junto con información de soporte del árbol de decisión *Pre-hoc*. Para ello se ideó un sistema de reglas sencillo que proporciona posibles razones para los valores bajos en el cálculo del estimador continuo (ver ecuación 2.8, en la página 9). Para cada patrón, el supervisor obtiene la información acerca de la clase con el mayor valor de predicción de acuerdo al modelo GMM, el valor del estimador continuo sobre el vector de datos, y el valor del estimador global del modelo mixto. Se contemplan los siguientes casos de reglas, además de los indicadores en base al árbol de decisión:

- Si para el patrón continuo $\boldsymbol{x}_i$ se tiene un valor inferior a cero en todos los resultados de la función densidad de probabilidad (pdf) de las gaussianas del modelo GMM, quiere decir que la probabilidad de pertenencia a cualquiera de ellas es muy baja y es muy

improbable que pertenezca al conjunto de datos normales (ya que el modelo se ajusta a datos no anómalos durante el entrenamiento).

- Si el patrón continuo $\boldsymbol{x}_i$ está asignado a una gaussiana con un peso (fracción de datos sobre el conjunto original de entrenamiento que son representativos de dicha gaussiana) en el modelo entrenado inferior al ratio de anomalías, significa que $\boldsymbol{x}_i$ se asemeja a un conjunto de puntos lo suficientemente infrecuentes como para considerarse anomalías. Esta regla se aplica también sobre una variante del conjunto de datos (considerando además el conjunto de test), teniendo en cuenta las mismas consideraciones.

Utilizando información del árbol de decisión y a modo de aviso, con menos importancia que las reglas anteriormente descritas, podemos diferenciar otros dos casos de detección:

- Si más del 80% de las divisiones continuas en el árbol de decisión se toman sobre valores de variables que se encuentran en los límites de los intervalos de entrada, entonces el patrón no se dirige de forma clara a una de las clases de gaussianas y esto puede indicar la presencia de un dato anómalo. Esta observación no ofrece certeza, por lo que se aporta como información adicional al modelo.

- Si el patrón continuo $\boldsymbol{x}_i$ pertenece a un nodo hoja del árbol de decisión con una densidad de patrones ínfima, quiere decir que su recorrido de variables para su separación es característico de sí mismo o de unos pocos datos, y se diferencia de conjuntos mayores sobre otros nodos hoja, que de nuevo, pudiera significar la separación de datos normales y anómalos. Se aporta como información adicional al modelo aunque no se tiene la certeza como para establecerla como una regla.

Una vez expuesta esta información al supervisor, se provee a éste de otros datos que pudieran terminar de matizar la explicación *Post-hoc* continua obtenida. Para cada uno de los patrones, se facilita la distribución de los datos en cada una de las clases de acuerdo al modelo GMM. Además, se muestran los valores de la función de densidad de probabilidad del patrón a cada una de las gaussianas, que se pudieran razonar como una medida de pertenencia difusa a cada una de las clases. Asimismo, el recorrido particular de cada uno de ellos por el árbol de decisión da una noción sobre las variables continuas determinantes del proceso de razonamiento seguido.

#### Post-hoc del vector categórico de datos

Para el caso de la información categórica se muestra el valor del estimador del modelo de regresión logística, así como del estimador global combinado, cuyo valor final es el causante de haber sido detectado como anomalía por el algoritmo. Para el procesamiento de la información categórica se tiene en cuenta la representación interna codificada mediante *One-Hot*

*Encoding*, de forma que para poder interpretar los datos en función de las variables de partida es necesaria una descodificación de los términos con los valores de estimador más bajos. Teniendo en cuenta esta representación y considerando la formulación de la sección 2.4 (página 6), para un caso codificado del vector categórico en la ecuación 2.3 (página 6) tendremos tantos factores de probabilidad como longitud de la representación *One-Hot Encoding* del mismo. La longitud de la codificación dependerá del número de valores posibles que puede tomar cada uno de los términos categóricos del vector. Para razonar el valor obtenido en el estimador categórico, se mantiene una lista $E$ de los términos que dieron dicho resultado, ordenándolos de menor a mayor por su estimador de la siguiente forma:

$$E = [(est_0, t_0),\ \dots, (est_{n-1}, t_{n-1})]\ \ /\ \ (est_i <= est_{i+1})\ \ \forall i \in [0, n-2] \tag{4.1}$$

Para realizar la selección de los valores más bajos se efectúa un doble filtrado mediante ratio y umbral. Para el primer caso, siendo $n$ la longitud de la codificación del vector categórico de datos y $E$ la lista de tuplas (estimador, término) ordenados de menor a mayor, se construye el vector de ratios $R$ de la forma:

$$R = [r_0,\ \dots, r_{n-2}]\quad /\quad r_i = \frac{est_{i+1}}{est_i}\ \ \forall i \in [0, n-2] \tag{4.2}$$

Sobre este vector de ratios $R$ realizamos un recorrido hasta encontrar un incremento consecutivo del orden igual o superior a 1.25, en cuyo caso se fragmenta la lista de estimadores $E$ por esa posición, desechando los índices superiores y transformándola en $E'$. Posteriormente se aplica el criterio de filtrado por valor máximo de estimador, de forma que se eliminan todos aquellos términos que habían superado dicho umbral. Finalmente se obtiene la lista de tuplas filtrada tanto por ratios como por umbral $E''$. El pseudocódigo explicativo del proceso de filtrado seguido se indica en la figura 1 (página 27).

Con esta información ya se conocen los términos de la formulación 2.5 (página 7) más bajos causantes de un estimador categórico inferior. Para dar una noción acerca de los resultados de cada uno de ellos, se proporciona al supervisor la media de los estimadores y el número de términos que cumplieron los criterios de selección. A continuación, se utiliza la lista $E''$ para extraer la información sobre las variables categóricas involucradas. Conociendo la posición del vector codificado a la que se hace referencia en el término $t_i$ de $E''$ y teniendo la información sobre el número de valores que puede tomar cada variable categórica, se descodifica dicho término para obtener la posición del vector de variables categóricas de partida. Todo el proceso seguido está resumido en la figura 4.6 (página 27). De esta forma ya se puede justificar el valor de estimador bajo en función de un dato (o varios) del vector de variables categóricas extraído inicialmente del conjunto de datos.

El supervisor visualiza la información sobre cada estimador que cumplió los criterios de

**Algoritmo 1:** Filtrado de lista de términos categóricos.

**Datos:** vector de ratios $R$, lista $E$, umbral de filtrado $umbral_{est}$.
**Resultado:** Lista filtrada $E''$.
**inicio**
  $n \longleftarrow Long(R)$
  $i \longleftarrow 0$
  **mientras** $((i < n)$ **y** $(R[i] < 1.25))$ **hacer**
    $i++$
  **fin**
  **si** *i==n* **entonces** $E' \longleftarrow E$
  **en otro caso** $E' \longleftarrow dividirVector(E, i)$ // *subvector izquierdo.*

  $l \longleftarrow Long(E')$
  $i \longleftarrow 0$
  **mientras** $(i < l)$ **hacer**
    **si** $(E'[i].estimador >= umbral_{est})$ **entonces**
      $E' \longleftarrow eliminarPosicion(E', i)$
    **fin**
    $i++$
  **fin**
  $E'' \longleftarrow E'$
  **devolver** $E''$
**fin**

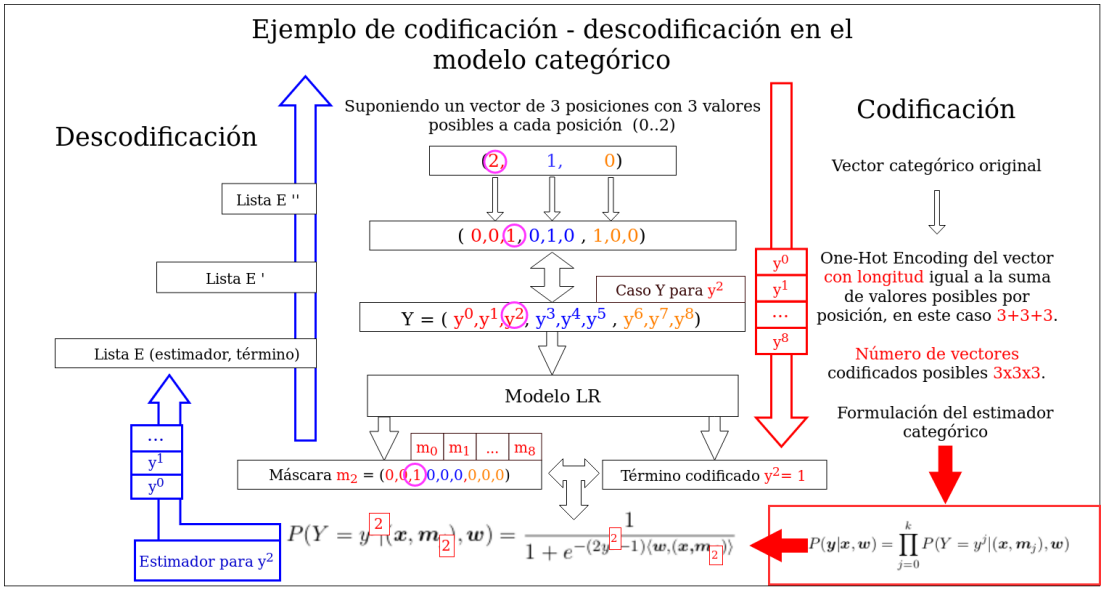



Figura 4.6: Proceso de descodificación categórica para extracción de las variables causantes de estimadores bajos.

filtrado, y para cada uno de ellos se indica además la posición junto con el valor de variable categórica involucrada. Teniendo en cuenta el criterio de condicionamiento continuo inicialmente planteado en la ecuación 2.2 (página 5), es interesante mostrar su relación con el vector continuo de datos siempre que sea posible. Para ello se considera el cálculo planteado en la ecuación 2.5 (página 7). Se realiza la descomposición de la fórmula de cada uno de los términos seleccionados centrándose en buscar el sumando mayor o menor (dependiendo del signo del exponente del denominador) del producto escalar de la expresión $\langle \boldsymbol{w}, (\boldsymbol{x}, \boldsymbol{m_j}) \rangle$ que hace que el resultado del estimador sobre dicho término $t_i$ sea más bajo. Si este sumando se sitúa en la parte del vector $\boldsymbol{x}$, quiere decir que existe un condicionamiento continuo claro sobre el desarrollo de dicho estimador categórico y que se puede aportar como información adicional a la explicación *Post-hoc* de la variable. Si se sitúa en la parte del vector $\boldsymbol{m_j}$, indica una baja frecuencia de ese valor categórico en esa posición, pero nada se puede asumir acerca de su condicionamiento continuo. Si el valor de $\boldsymbol{y}^j$ en la ecuación 2.5 (página 7) es un cero, interesa el sumando mayor del producto escalar para que el denominador sea mayor y por lo tanto el estimador sea inferior. En el caso de un valor uno de $\boldsymbol{y}^j$, interesa el sumando menor del producto escalar. El resumen de todo el proceso seguido se puede observar en la figura 4.7 (página 28). En cualquiera de los casos, si se tiene la información sobre el condicionamiento continuo, se muestra tanto la posición como el valor del vector de datos involucrado.

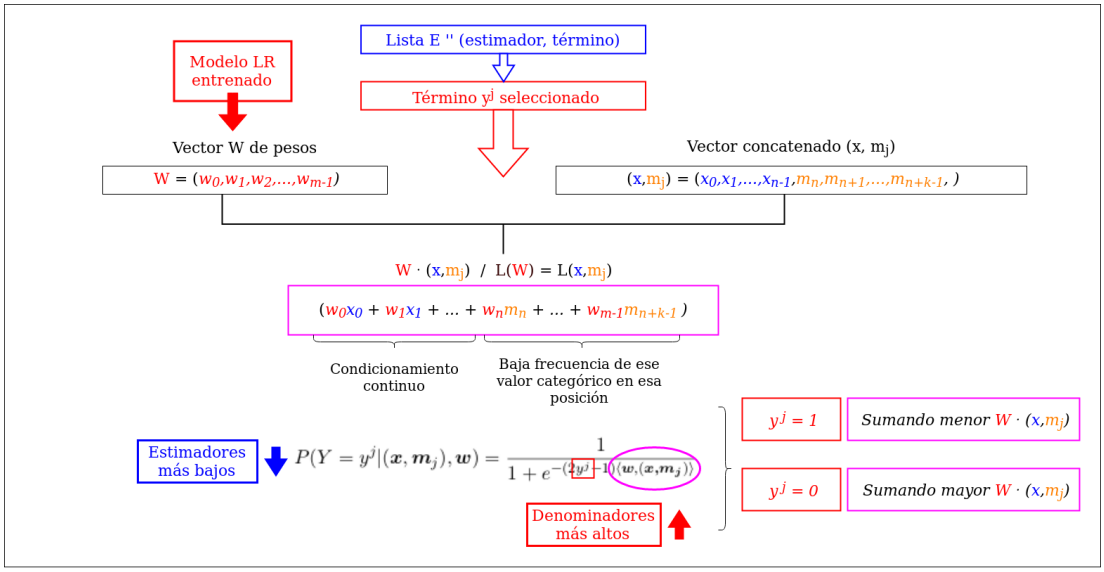



Figura 4.7: Cálculo del condicionamiento continuo sobre los términos categóricos seleccionados.

## 4.3 Consideraciones del algoritmo

El método de EML aquí expuesto se presenta como una solución *Ad-hoc* al problema de explicación planteado inicialmente. El objetivo es exponer una solución factible y realista a la falta de transparencia de los algoritmos con un ejemplo práctico de aprendizaje paramétrico en el campo de la detección de anomalías. Existen en la todavía escasa literatura del campo otros métodos a la hora de aportar una explicación, como es el caso de Lime [21] o Shap [22]. El primero de ellos se centra exclusivamente en explicaciones basadas en modelos lineales dispersos, y no trata la paralelización de los cálculos. El segundo propone varios modelos para explicación de una forma más completa, pero realiza varias suposiciones sobre los modelos de predicción de partida (precisión local, ausencia y consistencia [22]), que restringen su aplicación en entornos reales. Por estas razones se consideró interesante abordar una aproximación propia, que contemplase la formulación del algoritmo de anomalías a la vez que reutilizase sus predicciones en otros modelos de aprendizaje, como los árboles de decisión. En la mayoría de los casos en el dominio de EML, se buscan soluciones *Post-hoc* que aborden la explicación sobre las predicciones de cada patrón. Lo interesante del método expuesto recae en una solución unificada que contempla también un método *Pre-hoc*, que reutiliza la información del modelo de aprendizaje en otros sistemas de clasificación más transparentes con una explicación más natural y sencilla.

El método definido sobre el árbol de decisión puede ser generalizado a cualquier otro algoritmo de aprendizaje multiclase sobre el que poder abordar la separación y explicación de los datos. En cambio, el método *Post-hoc* tiene una dependencia clara del modelo de detección de partida, ya que para ambas naturalezas de los datos (variables continuas y categóricas), se trabaja con la formulación propuesta por ADMNC [8]. A la hora de tratar la explicación *Post-hoc* categórica se llegó a cuestionar el método de aprendizaje y se discutieron otras posibilidades a implementar, sobre todo por la dificultad que radica en una representación fidedigna del condicionamiento continuo de las variables categóricas de la ecuación 2.5 (página 7). El inconveniente que puede tener este método de aprendizaje se centra en el ajuste de los pesos del vector $\boldsymbol{w}$. Cuando el sumando seleccionado (siguiendo la figura 4.7, página 28) se sitúa en la región de la máscara $\boldsymbol{m_j}$, quiere decir que ese valor en esa posición categórica es infrecuente en el conjunto de datos, pero no aporta información de interés sobre el condicionamiento de las variables continuas. Este caso de explicación puede ser debido a la falta de convergencia, a un ajuste grueso de hiperparámetros o a una insuficiente cantidad de patrones en el conjunto de datos.

Capítulo 5

# Resultados obtenidos

## 5.1 Experimentos realizados

En este apartado se presentan todas las pruebas realizadas sobre conjuntos de datos etiquetados que son estándar en los trabajos de detección de anomalías. Hay que considerar que lo que se plantea aquí es el caso ideal en el que se tiene la información sobre un conjunto de datos normales para entrenamiento, que permiten al modelo ajustarse sobre un contexto conocido y que posteriormente pueden ser utilizados para discernir entre patrones que son anómalos y otros que no lo son. Un caso real contemplaría mayor desconocimiento sobre los conjuntos de datos, donde tanto valores normales como anómalos se encuentran mezclados. A priori no se tiene el conocimiento ni la certeza sobre la naturaleza de los datos normales y menos aún sobre lo que se puede considerar anomalía. Sin embargo, en este caso se hace necesario tratar con datos etiquetados para poder evaluar y medir los resultados obtenidos. La capacidad de discernir ambos casos y adaptar el modelo a la detección de datos anómalos dependerá de la robustez del método. Las pruebas aquí planteadas se exponen sobre conjuntos de datos muy conocidos en el estudio de aprendizaje máquina, trabajando de forma *offline* sobre ellos.

Los ficheros que agrupan los conjuntos de datos siguen un convenio de formato adaptado al algoritmo [1]. En relación a los parámetros de ajuste utilizados en la explicación, se adjuntan en la tabla 5.1 (página 32) los valores utilizados para el caso del árbol de decisión. El número de clases a separar dependerá del número de gaussianas especificadas en el modelo de partida y tanto la métrica de separación como el límite de profundidad y el número máximo de bins vienen especificados en detalle en la sección correspondiente *Pre-hoc* 4.2.1 (página 19). Para el caso *Post-hoc* los valores de referencia se muestran en la tabla 5.2 (página 32). Todos los parámetros previos se usaron tanto para el ajuste de las reglas de la parte continua como para el criterio de selección y filtrado de estimadores de la parte categórica. El reparto de datos en

[1] Se proporciona detalle de los ficheros en el anexo correspondiente.

| Parámetros de ajuste del árbol de decisión | |
|---|---|
| Número de clases | Número de gaussianas del modelo GMM |
| Métrica de impureza | Entropía |
| Máxima profundidad | 5 |
| Número máximo de bins | 40 |

Tabla 5.1: Parámetros del modelo de árbol de decisión.

| Parámetros de ajuste de la explicación Post-hoc | |
|---|---|
| Margen de separación al límite de continuas | 5% del intervalo |
| Densidad mínima de nodo hoja | 0.25% sobre el conjunto total |
| Umbral de filtrado categórico por ratios | 1.25 |
| Umbral de filtrado categórico por estimador | 0.40 |
| Umbral de caminos en límites de continuas | 80% |

Tabla 5.2: Parámetros del modelo de explicación Post-hoc.

los experimentos se realizó en todo caso de forma pseudo-aleatoria, trabajando con conjuntos de entrenamiento del 70% y con conjuntos de test del 30%. Las métricas que se muestran para cada caso comprenden desde la matriz de confusión de clases en test hasta los valores de precisión, sensibilidad y tasas de error. Para estos valores se llevaron a cabo repeticiones de cinco experimentos para el cálculo de medias $\mu$ y desviaciones típicas $\sigma$. Para realizar las pruebas, se utilizaron tanto conjuntos de datos generados artificialmente como otros reutilizados y adaptados a la detección de anomalías, que tienen como fuente repositorios de datos reconocidos [23]. Conseguir estos conjuntos de datos con anomalías etiquetadas es realmente complicado en entornos reales, por lo que para las pruebas aquí expuestas se trabajó sobre conjuntos (adaptados de sistemas de clasificación) que son estándar en los trabajos de detección de anomalías. Por motivos de legibilidad, los resultados sobre la explicación *Pre-hoc* y el informe de reporte *Post-hoc* son accesibles desde el repositorio *online* especificado en el anexo A (página 55).

### 5.1.1 GaussianArt - Conjuntos de datos artificiales

Para este caso, se plantearon dos versiones de conjuntos de datos generados de forma artificial. El objetivo principal fue validar el método de explicación y el funcionamiento del árbol de decisión (en selección y poda de nodos). Se planteó un problema de clases binario y de fácil separación con patrones de tres características, de las cuales dos de ellas son continuas y una discreta. Los resultados relativos a matrices de confusión con ajuste paramétrico a dos gaussianas se pueden visualizar en las tablas 5.3 y 5.4 (página 33). Se puede observar que la

predicción del árbol es prácticamente perfecta. Esto se debe a conocer la distribución de los datos a priori y a la facilidad de separación de los mismos. El resto de métricas calculadas aparecen en las tablas 5.5 y 5.6 (página 33). Se puede observar que al tratarse de un conjunto de datos generado de forma sencilla y de separación poco compleja, el sistema de aprendizaje es muy fiable. Lo interesante aquí es comprobar el comportamiento del algoritmo de filtrado y selección de nodos del árbol de explicación *Pre-hoc*, que podemos visualizar en la figura 5.1 (página 34).

| **Matriz de confusión (GaussianArt V1)** | | |
|---|---|---|
| | Patrones predichos a G 0 | Patrones predichos a G 1 |
| Patrones de G 0 | 105 | 0 |
| Patrones de G 1 | 1 | 96 |

Tabla 5.3: Matriz de confusión del conjunto GaussianArt V1

| **Matriz de confusión (GaussianArt V2)** | | |
|---|---|---|
| | Patrones predichos a G 0 | Patrones predichos a G 1 |
| Patrones de G 0 | 1565 | 0 |
| Patrones de G 1 | 0 | 1469 |

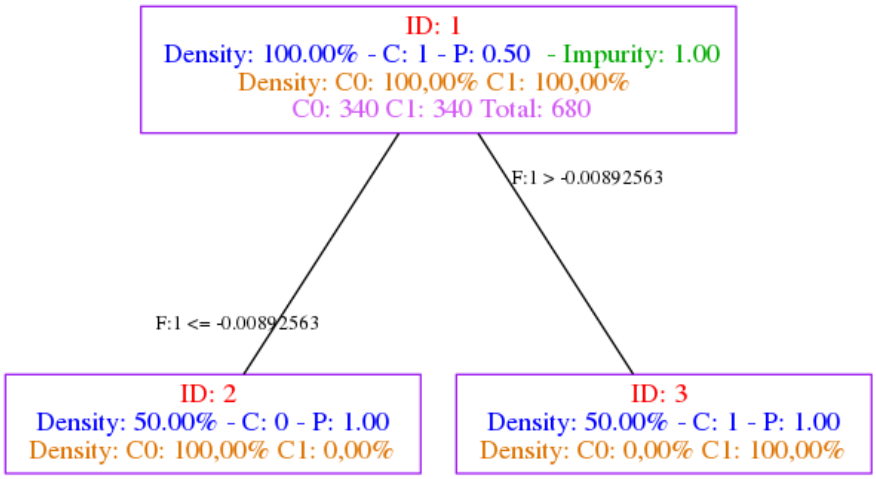
Tabla 5.4: Matriz de confusión del conjunto GaussianArt V2

| **Métricas de evaluación GaussianArt V1** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Tasa de falsos positivos (G 0) | 0 | 0 |
| Tasa de falsos positivos (G 1) | 0.0038 | 0.0061 |
| Precisión (G 0) | 1 | 0 |
| Precisión (G 1) | 0.9963 | 0.0059 |
| Tasa de verdaderos positivos (G 0) | 0.9962 | 0.0061 |
| Tasa de verdaderos positivos (G 1) | 1 | 0 |
| Tasa de error en test | 0.0019 | 0.0030 |
| Tasa de error en entrenamiento | 0 | 0 |

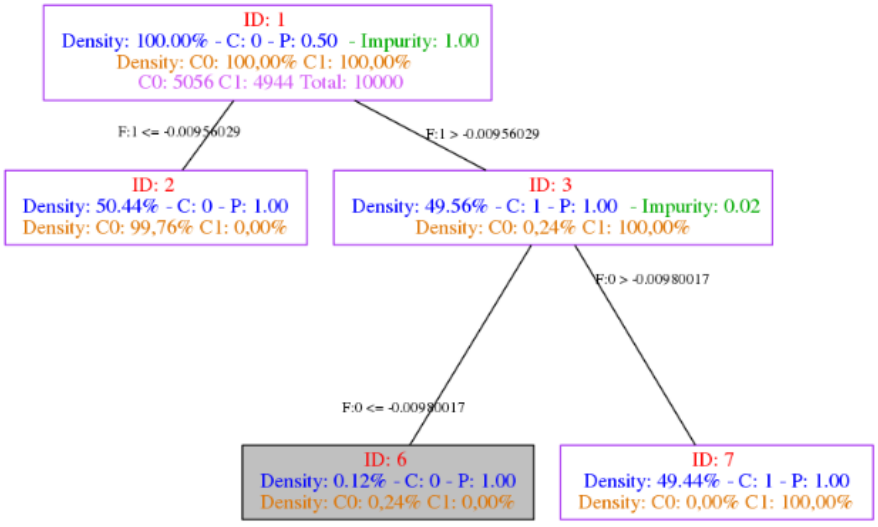
Tabla 5.5: Métricas de evaluación GaussianArt V1 con repeticiones de cinco experimentos

### 5.1.2 Abalone 1-8

Abalone es un conjunto de datos extraídos del *UCI Machine Learning Repository* [23]. Hace referencia a una especie marina de «orejas de mar», gasterópodos del género *Haliotis*. El




(a) Resultado *Pre-hoc* sobre conjunto GaussianArt V1




(b) Resultado *Pre-hoc* sobre conjunto GaussianArt V2

Figura 5.1: Resultados de GaussianArt en ambas versiones del conjunto de datos

| **Métricas de evaluación GaussianArt V2** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Tasa de falsos positivos (G 0) | 0 | 0 |
| Tasa de falsos positivos (G 1) | 0 | 0 |
| Precisión (G 0) | 1 | 0 |
| Precisión (G 1) | 1 | 0 |
| Tasa de verdaderos positivos (G 0) | 1 | 0 |
| Tasa de verdaderos positivos (G 1) | 1 | 0 |
| Tasa de error en test | 0 | 0 |
| Tasa de error en entrenamiento | 0 | 0 |

Tabla 5.6: Métricas de evaluación GaussianArt V2 con repeticiones de cinco experimentos

cometido original de este conjunto de datos era el estudio de sistemas de clasificación múltiple, y fue adaptado posteriormente a la detección de anomalías. En este caso, se utilizaron los datos etiquetados de la clase uno como los valores normales y los patrones etiquetados a la clase ocho como los datos anómalos. Los resultados sobre matrices de confusión y métricas de evaluación se pueden observar en las figuras 5.7 (página 35), 5.8 (página 36), 5.9 (página 36) y 5.10 (página 36). En este caso se trabajó con un modelo paramétrico para ajuste de cuatro gaussianas, ya que obtenía los mejores valores a la hora de realizar las pruebas de rendimiento [8]. Se puede observar como la confusión entre las clases es mayor, así como que aumentan las tasas de error, debido a la dificultad de aprendizaje que tiene el árbol a la hora de buscar umbrales continuos de separación. Este caso se aproxima más a los resultados típicos en situaciones reales no ideales, donde la dimensión de características es un factor importante a tratar (en contraposición a lo que ocurría en el caso de los conjuntos artificiales).

| **Matriz de confusión Abalone 1-8** | | | | |
|---|---|---|---|---|
| | Predichos G 0 | Predichos G 1 | Predichos G 2 | Predichos G 3 |
| Patrones de G 0 | 8 | 1 | 3 | 37 |
| Patrones de G 1 | 0 | 345 | 33 | 0 |
| Patrones de G 2 | 1 | 46 | 466 | 62 |
| Patrones de G 3 | 2 | 5 | 74 | 182 |

Tabla 5.7: Matriz de confusión sobre conjunto de datos Abalone 1-8 con cuatro gaussianas en el modelo.

### 5.1.3 Abalone 9-11

El caso de Abalone 9-11 siguió las mismas pautas especificadas para el caso previo de resultados, pero en este caso se utilizaron los datos etiquetados de la clase nueve como los

| **Métricas de evaluación Abalone 1-8 (TFP)** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Tasa de falsos positivos (G 0) | 0.0873 | 0.0952 |
| Tasa de falsos positivos (G 1) | 0.1096 | 0.0518 |
| Tasa de falsos positivos (G 2) | 0.0465 | 0.0204 |
| Tasa de falsos positivos (G 3) | 0.0758 | 0.0439 |

Tabla 5.8: Métricas de TFP en el conjunto Abalone 1-8

| **Métricas de evaluación Abalone 1-8 (Precisión)** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Precisión (G 0) | 0.7723 | 0.1039 |
| Precisión (G 1) | 0.7725 | 0.0875 |
| Precisión (G 2) | 0.6838 | 0.0888 |
| Precisión (G 3) | 0.8075 | 0.0725 |

Tabla 5.9: Métrica de precisión en el conjunto Abalone 1-8

| **Métricas de evaluación Abalone 1-8 (Sensibilidad y tasas de error)** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Sensibilidad (G 0) | 0.6404 | 0.2517 |
| Sensibilidad (G 1) | 0.7220 | 0.2296 |
| Sensibilidad (G 2) | 0.6360 | 0.1152 |
| Sensibilidad (G 3) | 0.7480 | 0.1392 |
| Tasa de error en test | 0.2124 | 0.0061 |
| Tasa de error en entrenamiento | 0.1974 | 0.0109 |

Tabla 5.10: Métricas de sensibilidad y tasas de error en Abalone 1-8

valores normales y los patrones etiquetados a la clase once como los datos anómalos. En la figura 5.11 (página 37) podemos ver el resultado de la matriz de confusión para el árbol de decisión entrenado, mientras que en la tabla 5.12 (página 37) se visualizan las tasas de error relativas al proceso de entrenamiento y test. Se trabajó con un modelo de entrenamiento paramétrico de seis gaussianas. El objetivo fue probar distintas configuraciones y comprobar la robustez del método, así como los resultados finales sobre el árbol de decisión si se varía el número de clases a considerar.

| **Matriz de confusión Abalone 9-11** | | | | | | |
|---|---|---|---|---|---|---|
| | Pred G 0 | Pred G 1 | Pred G 2 | Pred G 3 | Pred G 4 | Pred G 5 |
| Patrones de G 0 | 165 | 0 | 74 | 21 | 1 | 4 |
| Patrones de G 1 | 0 | 129 | 0 | 0 | 12 | 0 |
| Patrones de G 2 | 39 | 0 | 220 | 34 | 3 | 0 |
| Patrones de G 3 | 12 | 7 | 13 | 162 | 50 | 0 |
| Patrones de G 4 | 0 | 5 | 0 | 51 | 175 | 0 |
| Patrones de G 5 | 14 | 0 | 20 | 0 | 0 | 17 |

Tabla 5.11: Matriz de confusión Abalone 9-11

| **Tasas de error Abalone 9-11** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Tasa de error en test | 0.2479 | 0.0407 |
| Tasa de error en entrenamiento | 0.2306 | 0.0401 |

Tabla 5.12: Tasas de error del conjunto Abalone 9-11

### 5.1.4 Abalone 11-29

En el caso de Abalone 11-29 se trabajó sobre el mismo conjunto de datos, pero se utilizaron los datos etiquetados de la clase once como los valores normales y los patrones etiquetados de la clase veintinueve como los datos anómalos. La razón de realizar experimentación varias veces sobre el mismo conjunto de datos se debe a la facilidad de generación de conjuntos de datos tomando pares de clases del fichero original para adaptarlas a la detección de anomalías. En las figuras 5.13 (página 39) y 5.14 (página 39) se pueden comprobar los resultados obtenidos para este caso, que son mejores que en los otros dos estudios previos. La razón de ello puede ser el par de clases seleccionadas a diferenciar, con una separación continua de los datos más sencilla de modelar. Se vuelve a trabajar con un modelo de entrenamiento paramétrico de cuatro gaussianas, ya que es el que de forma genérica daba mejores resultados sobre la detección de anomalías [8].

### 5.1.5 German credit

El conjunto de *German Credit Data* toma como punto de partida una población de mil individuos representados en términos de variables continuas y categóricas que dan una noción acerca del riesgo que supone la concesión de un crédito bancario a su persona. La fuente de los datos es nuevamente *UCI Maching Learning Repository* [23]. Originalmente solo trataba valores discretos y fue adaptado posteriormente a valores continuos. La aplicación del algoritmo ADMNC [8] no supone un problema, ya que como se comentó anteriormente, su modelo trata ambos tipos de variables. Este sería uno de los casos de gran interés para aplicar técnicas de EML, especialmente porque el resultado del algoritmo afecta directamente a una persona física o jurídica que puede ejercer su derecho a la explicación, tal y como se comentaba en la sección 4.1 (página 15).

Se utilizó un modelo de ajuste paramétrico de tres gaussianas, ya que al tratarse de un conjunto reducido de datos fueron más que suficientes para ajustarse a los datos normales. En la figura 5.15 (página 39) se visualiza el resultado de la matriz de confusión para el árbol de decisión entrenado y se puede comprobar que el error de predicción de las clases es muy bajo. Además, en la figura 5.16 (página 39) tenemos el conjunto de métricas de evaluación con el que se puede comprobar la calidad de la separación de los datos y las bajas tasas de error. En este caso es importante mencionar que la separación de datos *Pre-hoc* dio unos resultados muy buenos, siendo capaz de caracterizar casi en su totalidad a las tres clases de gaussianas, explicadas en función de un grupo de variables continuas significativas.

### 5.1.6 Arritmia

El conjunto *Arrhythmia Data Set* se extrajo del mismo repositorio de datos comentado anteriormente [23]. Los patrones etiquetados para clasificación se organizan en una clase normal de frecuencias cardíacas frente a catorce clases de diferentes tipos de arritmias. El fichero de partida fue pre-procesado para adaptarlo al caso de la detección de anomalías, pero la transformación principal fue tomar la clase de datos normales como la que necesita el modelo de anomalías para entrenamiento y el conjunto de clases de arritmias para los datos anómalos, sin hacer distinción entre sus distintos tipos. De esta forma el método generaliza cualquier tipo de arritmia como una anomalía cardíaca, simplificando el problema de detección a un supuesto binarizado (si el dato es o no un valor anómalo). En este caso se trabajó con una gran cantidad de dimensiones, con un total de doscientas setenta y ocho características, de las cuales más de doscientas son valores continuos. La complejidad en este caso recae en el tamaño de los vectores de datos, donde el aprendizaje se puede complicar atendiendo a problemas conocidos en el ámbito de aprendizaje máquina, como es la maldición de las dimensiones [9]. Para casos con tal cantidad de características se requiere de conjuntos de entrenamiento muy

| **Matriz de confusión Abalone 11-29** | | | | |
|---|---|---|---|---|
| | Predichos G 0 | Predichos G 1 | Predichos G 2 | Predichos G 3 |
| Patrones de G 0 | 107 | 0 | 3 | 0 |
| Patrones de G 1 | 3 | 134 | 15 | 110 |
| Patrones de G 2 | 10 | 0 | 264 | 11 |
| Patrones de G 3 | 0 | 24 | 52 | 482 |

Tabla 5.13: Matriz de confusión Abalone 11-29

| **Tasas de error Abalone 11-29** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Tasa de error en test | 0.1847 | 0.0080 |
| Tasa de error en entrenamiento | 0.1623 | 0.0127 |

Tabla 5.14: Tasas de error del conjunto Abalone 11-29

| **Matriz de confusión German Credit** | | | |
|---|---|---|---|
| | Predichos G 0 | Predichos G 1 | Predichos G 2 |
| Patrones de G 0 | 202 | 0 | 1 |
| Patrones de G 1 | 1 | 46 | 1 |
| Patrones de G 2 | 4 | 0 | 43 |

Tabla 5.15: Matriz de confusión del conjunto German Credit.

| **Métricas de evaluación German Credit** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Tasa de falsos positivos (G 0) | 0.0417 | 0.0116 |
| Tasa de falsos positivos (G 1) | 0.0234 | 0.0146 |
| Tasa de falsos positivos (G 1) | 0.0091 | 0.0068 |
| Precisión (G 0) | 0.9440 | 0.0392 |
| Precisión (G 1) | 0.9140 | 0.0792 |
| Precisión (G 2) | 0.9460 | 0.0344 |
| Tasa de verdaderos positivos (G 0) | 0.9660 | 0.0248 |
| Tasa de verdaderos positivos (G 1) | 0.9280 | 0.0544 |
| Tasa de verdaderos positivos (G 2) | 0.8440 | 0.1352 |
| Tasa de error en test | 0.0396 | 0.0091 |
| Tasa de error en entrenamiento | 0.0268 | 0.0168 |

Tabla 5.16: Métricas de evaluación para German Credit Data

grandes y ese es justo el problema que plantean los datos extraídos. Únicamente se presentan cuatrocientas veinte muestras (los ficheros son accesibles desde el anexo A, página 55), que a su vez son divididos en grupos de entrenamiento y test. Esto resulta en una cantidad de patrones poco representativa en contraposición al número de dimensiones, pero sirva este escenario de ejemplo para exponer un problema común en la minería de datos.

En la figura 5.17 (página 40) se observa la matriz de confusión para el árbol de decisión entrenado sobre un modelo paramétrico de dos gaussianas. El conjunto de entrenamiento queda muy reducido en tamaño una vez se hace la partición de los datos, y los pocos resultados clasificados presentan una mayor confusión en la detección, tal y como se preveía en este mismo apartado. En la tabla 5.18 (página 40) se pueden ver las métricas de evaluación para este conjunto de datos, remarcando la variación entre el error en entrenamiento y test. Esta diferencia entre ambos valores da una noción de la falta de datos para tener un conjunto representativo del dominio, ya que, una vez se entrena con los datos disponibles (con un error razonable), la presentación de nuevos patrones en test casi duplica la tasa de error. Además, las clases presentan problemas en su detección, ya que muchos de los patrones etiquetados en la primera gaussiana (G 0) son detectados como pertenecientes a la segunda (G 1), lo que hace que tanto tasas de falsos positivos como verdaderos positivos disminuyan considerablemente.

| **Matriz de confusión Arritmia Data Set** | | |
|---|---|---|
| | Predichos G 0 | Predichos G 1 |
| Patrones de G 0 | 29 | 23 |
| Patrones de G 1 | 12 | 53 |

Tabla 5.17: Matriz de confusión del conjunto de datos arritmia

| **Métricas de evaluación Arritmia Data Set** | | |
|---|---|---|
| | Media $\mu$ | Desviación típica $\sigma$ |
| Tasa de falsos positivos (G 0) | 0.1584 | 0.0295 |
| Tasa de falsos positivos (G 1) | 0.4996 | 0.0773 |
| Precisión (G 0) | 0.7484 | 0.0390 |
| Precisión (G 1) | 0.6396 | 0.0551 |
| Tasa de verdaderos positivos (G 0) | 0.5012 | 0.0782 |
| Tasa de verdaderos positivos (G 1) | 0.8420 | 0.0292 |
| Tasa de error en test | 0.3275 | 0.0314 |
| Tasa de error en entrenamiento | 0.1846 | 0.0222 |

Tabla 5.18: Métricas de evaluación para Arritmia Data Set

## 5.2 Comentarios a los resultados y comparativas

El método de explicación aquí expuesto trata dos vertientes diferentes. Los resultados *Pre-hoc* obtenidos sobre el árbol de decisión son satisfactorios en la mayoría de los casos, donde hay suficiente cantidad de ejemplos y se trabaja en un número de dimensiones razonables. En esas situaciones, tanto las matrices de confusión como las diferentes tasas (TFP, TVP, Tasa de error...) dan una noción de la buena separación y clasificación de los datos ajustados a las gaussianas. Se probaron conjuntos de datos con grados dispares de dificultad con el objetivo de poner a prueba la fiabilidad y robustez del método de explicación. Se llegó a una buena aproximación *Post-hoc* sobre las muestras detectadas como anomalías. Desde un primer momento el propósito era facilitar una explicación simplificada y de alto nivel que pudiera ayudar a la tarea de un supervisor a la hora de realizar comprobaciones sobre las detecciones del algoritmo. Además, proveer datos adicionales para su justificación, realizando un estudio más profundo de la formulación y del ajuste paramétrico favoreció la implantación de un nuevo método *Ad-hoc* acoplado al algoritmo ADMNC [8], dotándolo de un módulo propio para justificación de resultados.

La mayoría de los métodos EML del estado del arte [15] [20] centran sus estudios de explicación exclusivamente en una visión *Post-hoc* sobre las predicciones de los modelos entrenados. Si bien es cierto que es la aproximación más interesante, un método adicional previo a detección sobre la formulación interna del algoritmo puede aportar mucha otra información de interés que puede proveer de otra solución independiente o complementaria sobre el algoritmo propuesto, lo cual garantiza una mayor fiabilidad y transparencia al tener más información disponible para la explicación.

Capítulo 6

# Planificación y evaluación de costes

La planificación de este proyecto está directamente influenciada por el ciclo de desarrollo que se utilizó y que ha sido descrito en la sección 3 (página 11), donde cada uno de los hitos más importantes se corresponde con la finalización de uno de los incrementos del método. El objetivo de este apartado es exponer las pautas generales que se siguieron a la hora de organizar el proyecto, prestando especial atención a cada una de las fases de análisis-diseño-implementación así como a las fases previas de documentación y estudio, teniendo en cuenta en todo momento el tiempo invertido en las mismas. Para llevar un control de las tareas realizadas durante el transcurso de los meses, se optó por mantener un borrador de trabajo en el que se tenía un registro de las actividades a realizar día por día que se obtenían durante las reuniones semanales con los directores del proyecto, estimando «grosso modo» las horas-hombre de esfuerzo para cada una de las tareas. Además de ello, también se mantuvo dicho control para poder contabilizar los costos aproximados del proyecto tomando un precio/hora-hombre de referencia. La utilización de ese documento a modo de *To-Do List* sirvió como guía para ajustar la planificación y los plazos.

## 6.1 Diagrama de Gantt

En la figura 6.1 (página 45) vemos el desglose de tareas e hitos que determinan el Diagrama de Gantt del proyecto de desarrollo del algoritmo de explicación. Cada una de las tareas se desglosa en subtareas de grano más fino en la planificación, utilizando para ello un código de colores que facilite su legibilidad. Sobre la línea de tiempo del proyecto se visualizan un total de cinco hitos, dos de los cuales marcan el inicio y fin de proyecto mientras que el resto marca la finalización de una iteración en el método de desarrollo incremental, con un entregable software funcional. El inicio del trabajo tuvo lugar a comienzos del segundo cuatrimestre 2018-2019 y se prolongó hasta finales de verano de ese mismo año. La duración total del proyecto abarca de forma aproximada 7 meses de desarrollo, teniendo en cuenta todos

los días naturales. Los primeros treinta días hábiles de trabajo se dedicaron a la inmersión en tecnologías, uno de los aspectos más importantes para poder tratar con las herramientas y algoritmos de los que no se tenía conocimiento hasta el momento. El estudio del lenguaje de programación *Scala* y del *framework* de desarrollo *Apache Spark* fueron las bases de conocimiento necesarias para poder realizar el proyecto. Otro aspecto importante fue el estudio minucioso del algoritmo existente de detección de anomalías, ADMNC [8], desarrollado en el grupo LIDIA de la FIC. Es fundamental conocer los fundamentos teórico-prácticos propuestos para este método, ya que el nuevo algoritmo de explicación trabaja acoplado al mismo como un módulo adicional. Posteriormente se suceden de forma ordenada las tareas que engloban cada uno de los incrementos en el desarrollo, con una especificación de subtareas que coincide con lo indicado en el capítulo 3 (página 11). La primera de las iteraciones continúa el trabajo de análisis del algoritmo ADMNC, pero centrándose de forma concreta en su implementación y en los casos de prueba para verificación. Tanto la segunda como la tercera iteración siguen las fases típicas de ciclo de desarrollo software, contemplando el diseño e implementación y posterior etapa de pruebas. El incremento que concierne a la parte continua de los datos es más extenso en tiempo, esto debido a que los datos continuos se utilizan en la explicación *Pre-hoc*, pero también de forma complementaria en la *Post-hoc*, donde intervendría el siguiente incremento de la parte categórica, exclusivamente con explicación *Post-hoc*. Finalmente la fase de integración y refactorización tiene por objetivo reorganizar y limpiar el código de aquellas anotaciones superfluas y de detalles sobre depuración.

## 6.2 Costes del proyecto

Para tomar una referencia sobre los costes del proyecto se consideran los bienes materiales que fueron necesarios para su realización, así como el cómputo total de horas de esfuerzo dedicadas al desarrollo del mismo. Se tomaron varias referencias [24] [25] [26] a la hora de realizar una estimación acerca del precio/hora de esfuerzo [27] [28], suponiendo el caso del perfil de programador junior. En el desglose de la tabla de la figura 6.1 (página 45) se puede ver el cálculo de horas así como el precio posterior a impuestos asociado a las mismas y el cálculo del salario bruto total. En otro cuadro de la tabla se muestra el coste asociado a los materiales utilizados y finalmente el cálculo total del proyecto.

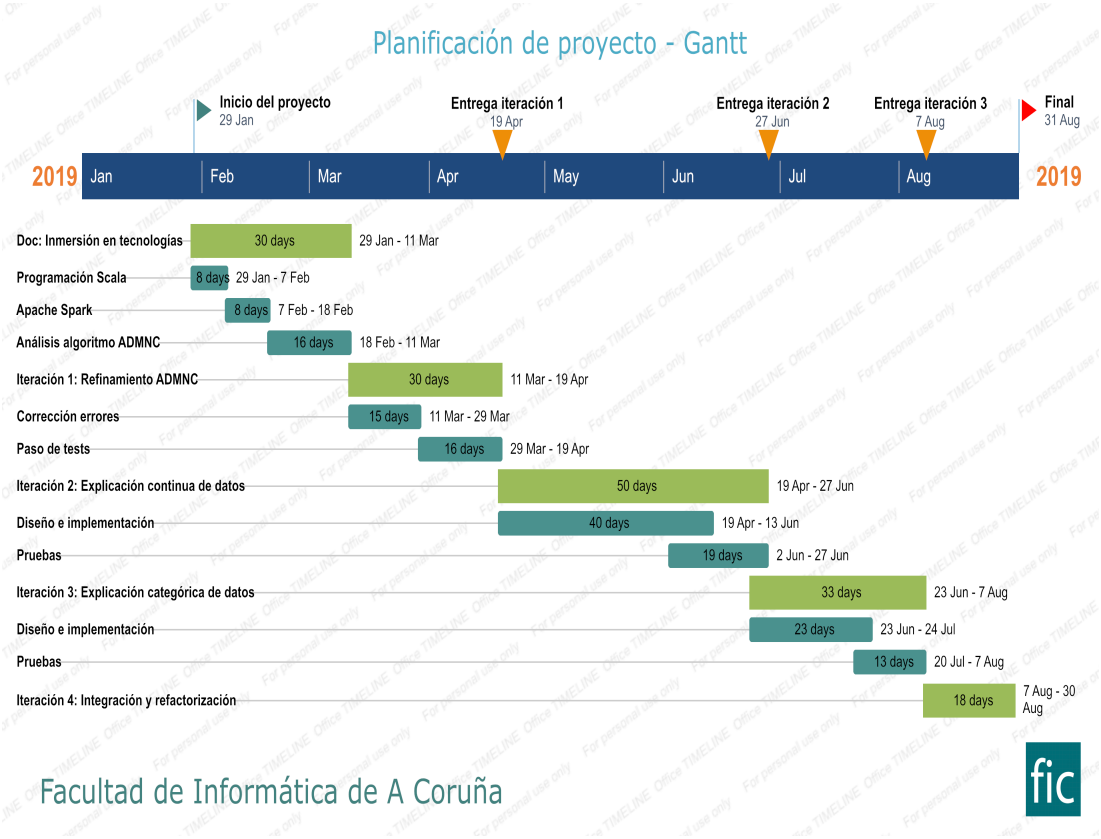



Figura 6.1: Diagrama de Gantt de planificación del proyecto

| **Detalle de horas** | **Cantidad** | **Precio/hora** | **Coste asociado** |
|---|---|---|---|
| 29 de enero - 6 de mayo (8 horas / semana) | 14 semanas = 112 horas | 9 €/hora | 1008 € |
| 11 de junio - 31 de agosto (22 horas / semana) | 12 semanas = 264 horas | 9 €/hora | 2376 € |
| Horas extra laborables | 50 horas | 10 €/hora | 500 € |
| TOTAL COSTE NETO HORAS | | | 3884 € |
| TASA IMPOSITIVA REAL 36,2% | | | 1406,01 € |
| TOTAL COSTE HORAS | | | 5290,01 € |
| **Materiales utilizados** | **Cantidad** | **Precio** | **Total** |
| Ordenador personal MSI GE62 6QD Apache Pro | 1 | 1.235,59 € | 1.235,59 € |
| **Coste total del proyecto** | | | 6525,60 € |

Tabla 6.1: Tabla de costes del proyecto

## Capítulo 7

# Conclusiones

Los algoritmos de explicación en el campo de la Inteligencia Artificial son necesarios para proporcionar transparencia y justificación a las predicciones de los modelos. Su interés va más allá de motivos sociales o legales, en un afán por comprender los procesos de razonamiento internos que hasta ahora funcionaban como cajas negras. En la actualidad, en el dominio de EML siguen sin existir estándares aceptados ni frameworks específicos de desarrollo. Es una rama de investigación muy reciente con tendencia al alza, sobre todo por la expansión en estos últimos años de la minería de datos y la Inteligencia Artificial a todos los ámbitos de la vida. Los derechos que las personas pueden ejercer sobre estos sistemas obligan en cierta manera a proveer de las herramientas que garanticen el cumplimiento de estas nuevas normativas, especialmente en aquellos casos donde se tomen decisiones críticas que afecten a la vida de las personas.

El algoritmo de explicación propuesto sobre el método de detección de anomalías AD-MNC [8] permite justificar las predicciones anómalas en términos de sus variables continuas y categóricas mediante un modelo unificado que integra la información de ambas naturalezas y que tiene por objetivo garantizar su transparencia. La mayoría de los métodos de explicación disponibles en el estado del arte únicamente contemplan una aproximación *Post-hoc* sobre sus predicciones, mientras que el método aquí descrito sigue ambas vertientes. Por un lado, la explicación *Pre-hoc* permite una interpretación del modelo de datos continuo previo a detección, de forma que se puede obtener información acerca de la generalización de los datos, identificar sesgos, formular hipótesis y caracterizar las clases de datos en términos de una o varias variables continuas. Por otro lado, la explicación *Post-hoc* sigue la aproximación típica en el campo de EML, obteniendo una justificación de los resultados posterior a la detección sobre el modelo entrenado. Se utiliza para ello tanto los valores continuos como categóricos. El estudio de un sistema de reglas, junto con el apoyo de la información del árbol de decisión y el desglose de estimadores categóricos, permitieron obtener explicaciones de alto nivel en lenguaje natural, que pueden ser de utilidad a un supervisor humano a la hora de monitorizar

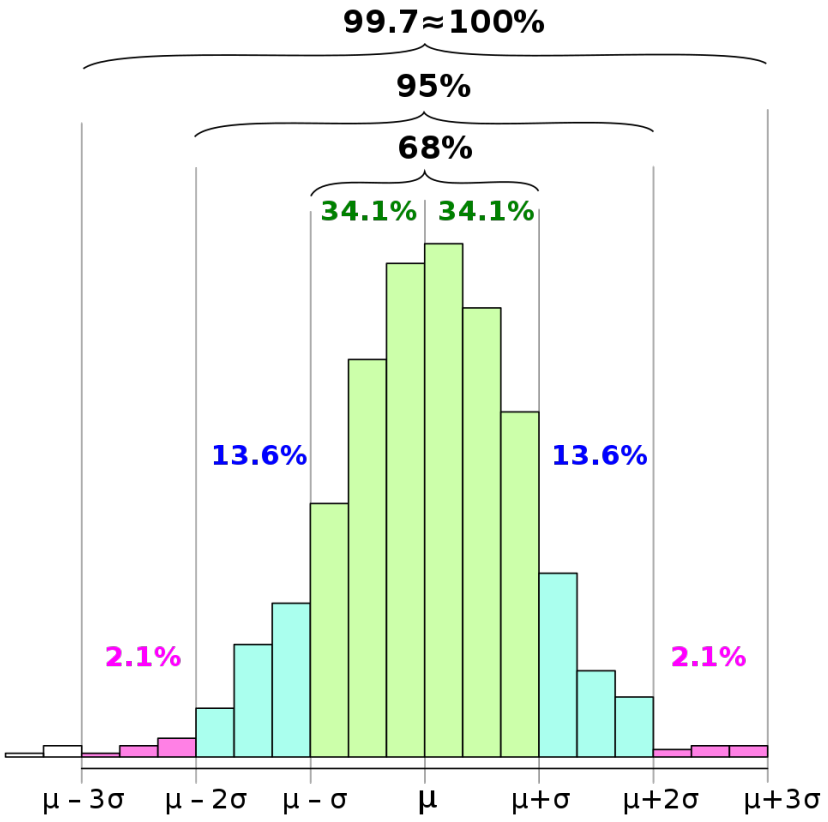
o evaluar las predicciones sobre el algoritmo de partida.

El método definido permite aportar una solución *Ad-hoc* sobre el algoritmo ADMNC, que profundiza en el estudio de modelos de EML, una rama de investigación muy reciente y en crecimiento durante estos últimos años, como comentamos en la sección 4.1 (página 15). El análisis profundo del método de detección de anomalías permitió implementar algunas mejoras, solucionar pequeños errores y ser críticos con su formulación, llegando incluso a plantear nuevos modelos de aprendizaje con el objetivo de mejorar las explicaciones indirectamente, tal y como se indica en el trabajo futuro que se presenta en el siguiente capítulo 8 (página 51). El haber implementado dicho algoritmo sobre el framework *Apache Spark* proporciona grandes ventajas a la hora de aportar escalabilidad a los problemas planteados. El procesamiento en paralelo optimizado permite trabajar de forma simultánea sobre varios lotes de datos, pudiendo agilizar notablemente los procesos de entrenamiento y de predicción. Además, la carga computacional se puede repartir entre varios núcleos de procesamiento (sobre un único PC o un cluster organizado), siguiendo un modelo de programación *MapReduce* [1]. El aspecto más favorable del algoritmo gira en torno a la capacidad de explicación y a la propiedad de transparencia y de justificación que adapta un caso particular de detección de anomalías a la tendencia actual en el campo de la Inteligencia Artificial, que es la XAI, la IA explicable.

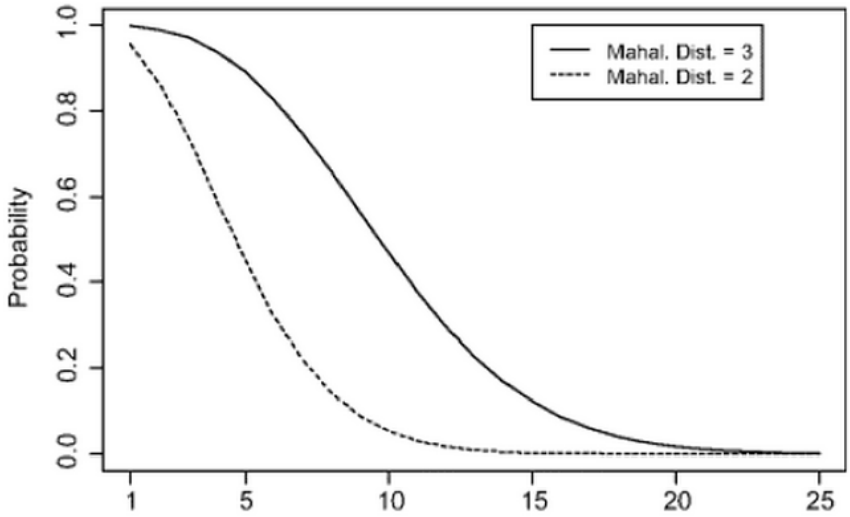
A pesar de todas las ventajas que plantea el método, también surgieron ciertos inconvenientes relacionados en mayor medida con lo que se conoce como la «Maldición de las dimensiones» [9]. Este concepto hace referencia a la problemática subyacente al excesivo número de características que pueden presentar los patrones de determinados dominios de datos, haciendo del aprendizaje una tarea tediosa y afectando negativamente a los modelos de predicción. Esto afecta de forma directa a las explicaciones, ya que el método se basa en el modelo entrenado para poder obtener las justificaciones. El objetivo de la explicación es presentar la información de la forma más clara y concisa, y el número de dimensiones sobre el conjunto de datos puede plantear serios problemas. Además de ello, a mayor número de dimensiones, menor es la certeza que se tiene sobre los criterios de distancias a las distribuciones [29], como sucede en el caso de la métrica de *Mahalanobis* a cada una de las gaussianas, capítulo 8 (página 51). La probabilidad de que un patrón se sitúe entre la media de datos y un número de desviaciones determinadas (1-$\sigma$, 2-$\sigma$...) decrece con el número de dimensiones. En la figura 7.1 (página 50) se puede ver de forma gráfica la problemática expuesta. En la subfigura 7.1a se observa la regla 3$\sigma$ que define el reparto natural de los datos que siguen una distribución normal convencional (como es el caso), teniendo en cuenta los valores de una, dos y tres desviaciones estándar respecto de la media. Por otro lado, la subfigura 7.1b muestra la relación dimensión-probabilidad teniendo en cuenta el caso de desviación 2$\sigma$ y 3$\sigma$ para la métrica *Mahalanobis* sobre la distribución. Se puede observar que la certeza de la medida

[1] Se proporciona detalle adicional al framework de desarrollo en el apéndice.

decrece con la dimensión, de forma que a partir de un determinado número de ellas se pierde toda la noción de dicha medición. Por este motivo, la reducción de dimensión es una línea clara de trabajo futuro 8 (página 51), que contempla la aplicación de técnicas tales como el análisis de componentes principales o la selección de características.




(a) Regla $3\sigma$ sobre distribución normal de datos




(b) Comparativa dimensión-probabilidad para la distancia *Mahalanobis*

Figura 7.1: La problemática de la alta dimensión de los patrones

# Capítulo 8

# Trabajo futuro

Una de las posibles ramas de investigación para mejora o ampliación del algoritmo de explicación recae en el tratamiento previo de los datos para aquellas situaciones que complican las detecciones favorables, tal y como se comentaba en la sección 5.2 (página 41). Es el caso de la alta dimensión de los conjuntos de datos, donde existen muchas características superfluas que pueden provocar sesgos en la predicción y consecuentemente generalizar de forma errónea sobre conjuntos de test. En este ámbito se podrían explorar varias aproximaciones:

- Una de las técnicas que se puede abordar es el análisis de componente principales [30] (ACP, o de sus siglas en inglés, PCA). Este método se basa en aplicar una reducción de dimensión sobre el conjunto de datos numérico de un patrón transformando sus características de partida en un menor número de variables no correlacionadas. La base matemática del método se centra en exponer en un nuevo sistema de coordenadas la estructura interna de los datos mediante una explicación de la varianza de los mismos. Es muy útil en aquellos entornos de alta dimensión, donde se proporciona al usuario una proyección de los patrones de partida en una representación gráfica simplificada desde el punto de vista más informativo de los mismos.

- Otra técnica útil a la hora de simplificar los conjuntos de datos y hacerlos más significativos es la selección de características [31] [32] (conocida comúnmente como *Feature Selection* o FS). Con este método, el objetivo principal es extraer las variables más relevantes de cada uno de los patrones del conjunto para poder simplificar los modelos de predicción y hacer más fáciles las interpretaciones de los resultados. Con la selección de características aplicada al algoritmo propuesto, los tiempos de entrenamiento se reducirían drásticamente en aquellos conjuntos de gran dimensión y el ajuste de gaussianas se simplificaría de forma notable. Asimismo, se filtrarían aquellas características redundantes y se favorecería la capacidad de generalización del modelo de aprendizaje,

evitando el sobre-ajuste sobre los conjuntos de datos. El principio a considerar si se utiliza esta técnica es que los datos *a priori* presentan variables que son irrelevantes y que estas pueden ser eliminadas sin incurrir en una apreciable pérdida de información. Se aplica en las situaciones en las que el número de muestras es demasiado bajo respecto a la cantidad de variables del vector de datos, como en el caso mostrado en la sección de resultados. 5.1.6 (página 38).

Además de estas posibles líneas futuras, también resulta interesante profundizar en el desarrollo de los métodos de explicación, especialmente en el *Post-Hoc*, con el objetivo de mejorar en precisión y robustez sobre los resultados obtenidos, pudiendo destacar:

- Una opción interesante sería un refinamiento del sistema de reglas inicialmente planteado para la explicación de la parte continua. Con la adición de nuevos casos de justificación se podrían abarcar más situaciones sobre el ajuste de las gaussianas, pero se requeriría de un mayor estudio para asegurar la robustez y capacidad de generalización de las explicaciones. Una nueva medida interesante podría ser un criterio de distancias de un patrón concreto a cada una de las gaussianas, utilizando para ello el vector de medias y la matriz de co-varianzas [33]. En espacios de baja dimensión, medidas como la distancia *Mahalanobis* son un buen punto de partida a la hora de establecer criterios de distancia y puede proporcionar información adicional acerca de la pertenencia del patrón a cada una de las gaussianas. Además, también puede ser útil en el proceso de aprendizaje, durante el cálculo de los parámetros de ajuste.

- Otro punto importante para mejorar el desarrollo de explicación pudiera estar en un replanteamiento del condicionamiento continuo en el modelo de aprendizaje logístico de la sección 4.3 (página 29). En algunas ocasiones los resultados obtenidos para la explicación de bajos estimadores categóricos eran insuficientes para justificar el condicionamiento al vector continuo de datos, dando lugar a una redundancia sobre el valor categórico, con lo que únicamente se podría argumentar la baja probabilidad de tener ese valor en esa posición. Se podría implementar un nuevo modelo que interrelacionara en mayor grado cada una de las variables continuas sobre las categóricas, de forma que el ajuste del vector de pesos $\boldsymbol{w}$ fuera más significativo y aportara información de mayor interés a la hora de obtener un método de explicación de alto nivel. Esto supondría un replanteamiento del modelo de regresión logística sobre el algoritmo de partida [8], pero no afectaría al método de explicación propuesto en este desarrollo 4.2 (página 18).

# Apéndices

## Apéndice A

# Resultados a la explicación

En este capítulo de anexo se incluye un acceso a repositorio *online* para la visualización de todos los resultados del algoritmo aquí expuesto. La razón de ello es la falta de legibilidad y de calidad de imagen si esta se incluye en la memoria, por lo que para aquellos casos que se requiera de una vista al detalle de alguno de los resultados, se recomienda encarecidamente consultar el siguiente enlace:

- Acceso público a usuarios UDC.

Dicho directorio público se organiza en tres carpetas, una de ellas contiene todos los diagramas de árboles de decisión acompañados de sus tablas explicativas (*TFG_PREHOC*) mientras que la otra incluye todos los informes html de la explicación *Post-hoc* sobre el modelo de anomalías (*TFG_POSTHOC*). Por último, la tercera carpeta (*TFG_CONJUNTOS_DE_DATOS*), incluye cada uno de los ficheros utilizados con la información de todos los patrones. En relación a su formato, cada patrón viene indicado en una única línea. El primer bit representa la clase anómala o no anómala (al tratarse de un conjunto etiquetado, puede ser filtrada en entrenamiento). A continuación viene el vector de datos, indicando cada posición mediante un índice numérico seguido de dos puntos. Primeramente aparecen los valores del vector categórico y después el conjunto consecutivo de valores continuos. Como precondición a su procesamiento, los valores categóricos deben estar identificados por enteros positivos comenzando por el cero y de forma incremental. Los valores continuos pueden tomar cualquier valor sobre el conjunto de $\mathbb{R}$, aunque se les puede aplicar de forma opcional normalización en el intervalo $[-1, 1]$.

## Apéndice B

# Apache Spark

Apache Spark es un *framework* de código abierto de propósito general para desarrollo distribuido en *clusters*, que tiene como propiedades fundamentales la escalabilidad y el paralelismo. Las operaciones que proporciona contemplan de forma implícita esa distribución de datos, que se aplica a cada una de las máquinas de forma proporcionada. Además, también es compatible en un entorno pseudo-distribuido local, en varios núcleos computacionales (físicos o lógicos) sobre un único ordenador. *Apache Spark* presenta una API sencilla y librerías muy completas.

La abstracción principal que utiliza como estructura de datos es el RDD (del inglés, *Resilient Distributed Dataset*), que representa a conjuntos de datos de solo lectura distribuidos en múltiples máquinas. Son el núcleo de *Spark*, sobre el que se realizan todas las operaciones. De cara al programador es transparente, ya que únicamente se trabaja sobre ellos mediante operaciones de mapeo y reducción, pero a bajo nivel cada nodo se encarga de parte del trabajo, donde hay redundancia, tolerancia a fallos y distribución de tareas. El término *Resilient* hace referencia a que es *Spark* el que se encarga de reconstruir los datos si alguno se tuvo que borrar o hubo algún fallo en las máquinas.

Para trabajar con los RDDs, se tienen que tener en cuenta las siguientes cuatro etapas [1], en este orden:

1. **Creación**: A partir de una variable, fichero, u otra fuente de datos.

2. **Transformaciones**: Conocidas también como operaciones de mapeo.

3. **Acciones**: Conocidas también como operaciones de reducción.

4. **Almacenamiento**: Si se requiere salvar el conjunto de datos transformado.

[1] Información del anexo extraída del Taller de Introducción a Apache Spark, por Carlos Eiras Franco, MUBICS - Octubre 2017

La máquinas obedecen a dos tipos diferentes de roles, que dependiendo de cada uno de ellos, cumplirá una función diferente dentro del cluster:

- **Driver**: Máquina que se encarga de distribuir y coordinar las tareas, ejecuta el programa general.

- **Worker**: Se encargan de ejecutar el trabajo. Ejecutan las funciones que se pasan a las transformaciones y acciones, que son operaciones *MapReduce*.

Además, entre ellos puede existir un gestor de clúster, o no tenerlo y ejecutarse en modo *standalone*. En la figura B.1 (página 58) vemos de forma intuitiva como sería la organización y coordinación de trabajo entre las diferentes máquinas, teniendo en cuenta el rol de cada una de ellas.

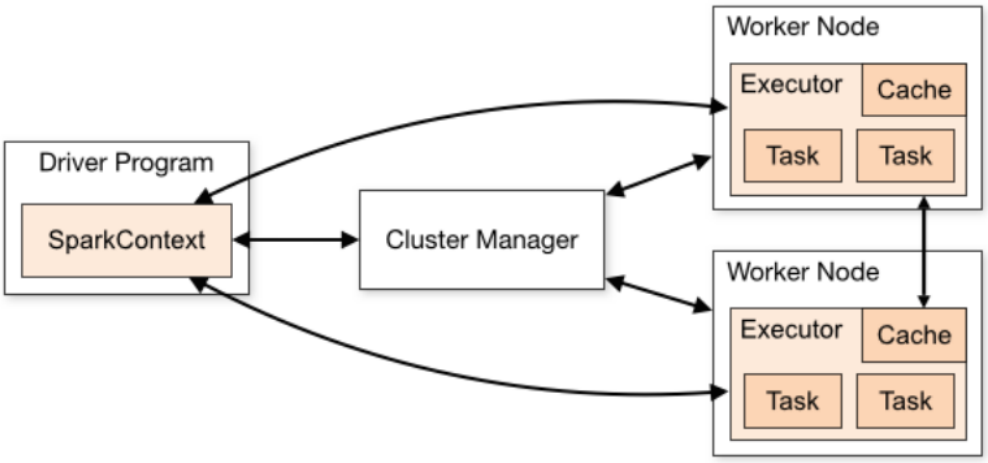



Figura B.1: Organización de máquinas en el cluster en Apache Spark

Apache Spark basa las transformaciones de datos en operaciones de mapeo y las acciones a realizar en operaciones de reducción. El modelo de programación *MapReduce*, visible en la figura B.2 (página 59), y su flujo de datos se pueden entender siguiendo los siguientes puntos:

- **Lector de entrada**: Divide el conjunto de datos en varias particiones equitativas a distribuir entre las diferentes máquinas del cluster, quedando cada una de ellas asignada a una operación de mapeo.

- **Operación Map**: Para cada partición de datos, en forma de pares clave-valor, se procesa cada una de ellas a través de una función pasada como parámetro que genera su transformación correspondiente.

- **Función de particionado**: Se encarga de distribuir los resultados de mapeo entre cada una de las máquinas que realizarán las reducciones, utilizando para ello operaciones *hash* sobre la clave de los datos, con el objetivo de balancear la carga en el cluster.
- **Función de comparación**: Para ordenamiento de datos transformados en el cluster.
- **Operación de reducción**: Siguiendo el orden de datos, se aplica la función de reducción una vez por clave de datos, de forma que para cada conjunto con la misma clave se produce una salida.
- **Escritor de salida**: Salva los resultados de la acción de reducción en almacenamiento estable.

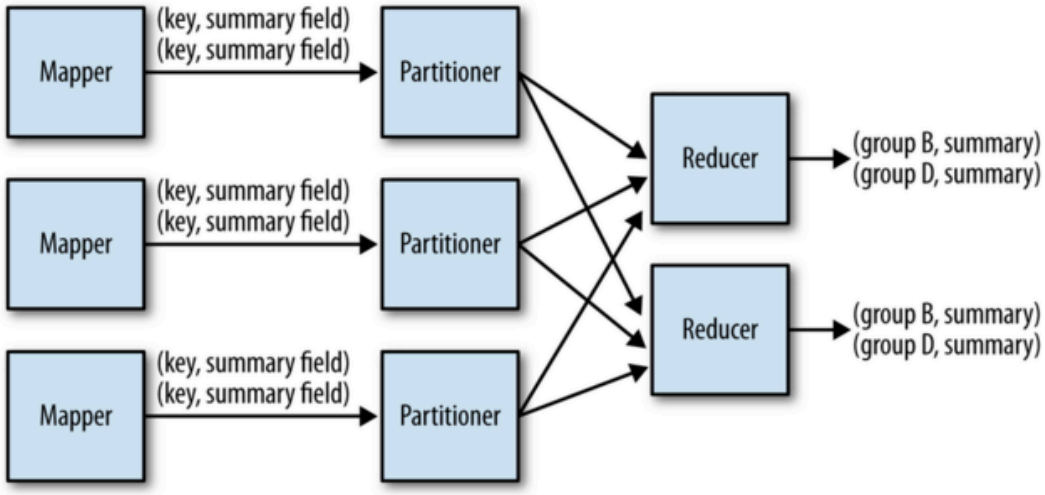



Figura B.2: Modelo de programación *MapReduce* simplificado.

## Apéndice C

# GraphViz

GraphViz [1] es un paquete de herramientas gráficas de código abierto que permite el dibujado de grafos mediante scripts en formato DOT. Las visualizaciones en forma de grafo permiten representar información estructurada mediante redes de nodos y arcos direccionales. Tiene importantes aplicaciones en campos como bioinformática, desarrollo software, bases de datos, *machine learning*... La razón de utilizarlo en la explicación *Pre-hoc* de este proyecto es debido a la simplicidad y facilidad del formato DOT para la estructuración de los datos así como a la gestión de estilos gráficos. *GraphViz* permite representar árboles de decisión de forma clara, además de poder trabajar con tablas a modo de leyenda.

La herramienta que se utilizó para la generación de imágenes por linea de comandos (a partir del fichero fuente) fue *dot* [34], que permite el dibujado jerárquico o por capas de grafos dirigidos. Es la herramienta por defecto cuando se trabaja en redes de nodos dirigidas, aunque existen muchas otras (*neato*, *fdp*, *twopi*...). Para generar los archivos finales en formato PNG se utilizó el siguiente comando, tomando de partida el fichero en formato dot con todos los datos que genera de forma automática el método de explicación *Pre-hoc*.

```
$ dot -Tpng fichero_generado_metodo_prehoc.dot > fichero_salida.png
```

Todos los resultados obtenidos son accesibles desde el anexo A (página 55). En relación al formato *dot*, podemos destacar tres componentes principales:

- **Grafos**: Pueden ser dirigidos y engloban el conjunto de elementos y atributos de toda la representación.

```
    (di)graph G {
        <contenido>
    }

```

[1] Documentación en: https://graphviz.gitlab.io/documentation/

- **Nodos**: Estructuras principales contenidas en el grafo y piezas para la construcción de los mismos, se identifican con una etiqueta única y pueden contener atributos [35] entre corchetes, siguiendo una nomenclatura de pares nombre-valor.

```
<etiqueta_nodo> [nombre_atributo_1 = valor_atributo_1, ... ,
nombre_atributo_n-1 = valor_atributo_n-1; nombre_atributo_n =
valor_atributo_n]

```

- **Arcos**: Representan uniones entre nodos. Nuevamente pueden presentar atributos [35] y dependiendo de si se trata de un grafo convencional o dirigido tendrán una forma u otra.

```
<etiqueta_nodo_1> -- <etiqueta_nodo_2> [nombre_atributo =
valor_atributo, ...] //convencional
<etiqueta_nodo_1> -> <etiqueta_nodo_2> [nombre_atributo =
valor_atributo, ...] //dirigido

```

Para incluir finalmente las tablas a modo de leyenda, se utilizó uno de los nodos de forma especial con el que se siguió el formato estándar HTML para tablas, que es compatible con la herramienta *GraphViz*.

# Relación de acrónimos

**LIDIA** *Laboratorio de Investigación y Desarrollo en Inteligencia Artificial.*

**ADMNC** *Anomaly Detector for Mixed Numerical and Categorical inputs*

**SFE** *Sequential Feature Explanation*

**GMM** *Gaussian Mixture Model*

**LR** *Logistic Regression*

**EM** *Expectation Maximization*

**SGD** *Stochastic Gradient Descent*

**AUROC** *Area Under Receiver Operating Characteristic*

**XAI** *Explainable Artificial Intelligence*

**EML** *Explainable Machine Learning*

**DARPA** *Defense Advanced Research Projects Agency*

**PCA** *Principal Component Analysis*

**FS** *Feature Selection*

**RDD** *Resilient Distributed Dataset*

# Glosario

**Anomalía** Valor atípico sobre un conjunto de datos que se diferencia del resto por alguna de sus características o propiedades.

**AUROC** El área bajo la curva ROC, es una medida de evaluación de clasificadores que emplea la tasa de verdaderos positivos frente a la tasa de falsos positivos para establecer una medida de la idoneidad de los sistemas de detección en técnicas de aprendizaje automático

**Tasa de verdaderos positivos (TVP)** Proporción de datos que han sido clasificados correctamente como positivos sobre el conjunto total de datos verdaderamente positivos.

**Tasa de verdaderos negativos (TVN)** Proporción de datos que han sido clasificados correctamente como negativos sobre el conjunto total de datos verdaderamente negativos.

**Tasa de falsos positivos (TFP)** Proporción de datos que han sido clasificados erróneamente como positivos sobre el conjunto total de datos verdaderamente negativos.

**Tasa de falsos negativos (TFN)** Proporción de datos que han sido clasificados erróneamente como negativos sobre el conjunto total de datos verdaderamente positivos.

**Explicación Pre-hoc** Explicación de los resultados iniciales del algoritmo previo a detección, esto es, sobre la información del modelo entrenado para obtener una noción sobre la separabilidad o agrupamiento de los datos.

**Explicación Post-hoc** Explicación de los resultados del algoritmo posterior a detección y sobre cada muestra de forma individual, indicando en cada caso las razones de una posible detección positiva.

**One-Hot Encoding** Codificación binaria de longitud igual al número posible de valores que puede tomar una variable de partida donde únicamente se permite un bit a uno (especificando uno de los valores posibles) sobre el resto de valores a cero de la codificación.

**Hiperparámetro** Valores que se configuran antes del entrenamiento de los modelos y que se usan para parametrizarlo e instanciarlo. Hacen referencia a configuraciones que no modelan los datos directamente, pero que influyen en la capacidad y en las características de aprendizaje del modelo.

**Aprendizaje *offline*** Aprendizaje automatizado sobre un conjunto de datos completo y estático sobre el que no aplican restricciones de tiempo. El algoritmo no toma nuevos datos ni opera en tiempo real.

**Método *Ad hoc*** Método específicamente elaborado para un problema o fin preciso que puede no ser generalizable ni utilizable para otros propósitos.

# Bibliografía


[1] D. M. Hawkings, *Identification of outliers*. Springer, 1980, vol. 11.

[2] V. Kumar, "Parallel and distributed computing for cybersecurity," *IEEE Distributed Systems Online*, vol. 6, no. 10, 2005.

[3] A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A review of anomaly detection in automated surveillance," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1257–1272, Nov 2012.

[4] T. Koussev, M. P. Weiss, and K. Reiss, "A graphic explanation environment for expert systems," in *Second International Conference on Software Engineering for Real Time Systems, 1989.*, Sep. 1989, pp. 11–15.

[5] Zhang Bofeng, Wang Na, Wu Gengfeng, and Li Sheng, "Research on a personalized expert system explanation method based on fuzzy user model," in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788)*, vol. 5, June 2004, pp. 3996–4000 Vol.5.

[6] T. Mokoena, O. Lebogo, A. Dlaba, and V. Marivate, "Bringing sequential feature explanations to life," in *2017 IEEE AFRICON*, Sep. 2017, pp. 59–64.

[7] C. B. Low, D. Wang, S. Arogeti, and J. B. Zhang, "Causality assignment and model approximation for hybrid bond graph: Fault diagnosis perspectives," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 570–580, July 2010.

[8] C. Eiras-Franco, D. Martínez-Rego, B. Guijarro-Berdiñas, A. Alonso-Betanzos, and A. Bahamonde, "Large scale anomaly detection in mixed numerical and categorical input spaces," *Information Sciences*, vol. 487, pp. 115 – 127, 2019. [En línea]. Disponible en: http://www.sciencedirect.com/science/article/pii/S0020025519302014

[9] G. V. Trunk, "A problem of dimensionality: A simple example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 3, pp. 306–307, July 1979.

[10] I. F. Iatan, "The expectation-maximization algorithm: Gaussian case," in *2010 International Conference on Networking and Information Technology*, June 2010, pp. 590–593.

[11] M. Cassel and F. Lima, "Evaluating one-hot encoding finite state machines for seu reliability in sram-based fpgas," in *12th IEEE International On-Line Testing Symposium (IOLTS'06)*, July 2006, pp. 6 pp.–.

[12] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 161–168. [En línea]. Disponible en: http://papers.nips.cc/paper/3323-the-tradeoffs-of-large-scale-learning.pdf

[13] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan, "Better mini-batch algorithms via accelerated gradient methods," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 1647–1655. [En línea]. Disponible en: http://papers.nips.cc/paper/4432-better-mini-batch-algorithms-via-accelerated-gradient-methods.pdf

[14] C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," *Computer*, vol. 36, no. 6, pp. 47–56, June 2003.

[15] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[16] A. Calder, *Reglamento General de Protección de Datos (RGPD) de la UE: Una guía de bolsillo*. IT Governance Publishing, 2016. [En línea]. Disponible en: http://www.jstor.org/stable/j.ctt1m3p208

[17] D. O. de la Unión Europea. (2016, apr) Reglamento (ue) 2016/679 del parlamento europeo y del consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la directiva 95/46/ce (reglamento general de protección de datos). [En línea]. Disponible en: https://www.boe.es/doue/2016/119/L00001-00088.pdf

[18] A. Calvo. (2019, apr) We are ready for machine learning explainability? [En línea]. Disponible en: https://towardsdatascience.com/we-are-ready-to-ml-explainability-2e7960cb950d

[19] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th*

*International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2018, pp. 80–89.

[20] D. Gunning. (2018, mar) Explainable artificial intelligence (xai). [En línea]. Disponible en: https://www.darpa.mil/program/explainable-artificial-intelligence

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1135–1144.

[22] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [En línea]. Disponible en: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[23] D. Dua and C. Graff, "UCI machine learning repository," 2017. [En línea]. Disponible en: http://archive.ics.uci.edu/ml

[24] W. B. Services. (2018, jan) La realidad del perfil de informático junior en españa según los informes. [En línea]. Disponible en: https://www.xataka.com/tecnologiazen/la-realidad-del-perfil-de-informatico-junior-en-espana-segun-los-informes

[25] Indeed. (2019, jul) Salarios para empleos de programador/a junior en españa. [En línea]. Disponible en: https://www.indeed.es/salaries/Programador/a-junior-Salaries

[26] GlassDoor. (2019, jun) Junior programmer salaries in spain. [En línea]. Disponible en: https://www.glassdoor.com/Salaries/spain-junior-programmer-salary-SRCH_IL.0,5_IN219_KO6,23.htm

[27] Iberley. (2016, apr) Caso práctico: Cálculo del valor de una hora ordinaria de trabajo. [En línea]. Disponible en: https://www.iberley.es/practicos/caso-practico-calculo-valor-hora-ordinaria-trabajo-7191

[28] Neuvoo. (2019, jun) ¿cuanto ganaría por año con un salario por hora de 9 €? [En línea]. Disponible en: https://neuvoo.es/convert/?salary=9&from=hour&to=year&lang=es&hw=37.5

[29] P. Bajorski, *Statistics for Imaging, Optics, and Photonics*. WILEY, 2012.

[30] B. Schölkopf, J. Platt, and T. Hofmann, *In-Network PCA and Anomaly Detection*. MITP, 2007. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/6287317

[31] S. G. Devi and M. Sabrigiriraj, "Feature selection, online feature selection techniques for big data classification: - a review," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, March 2018, pp. 1–9.

[32] S. Visalakshi and V. Radha, "A literature review of feature selection techniques and applications: Review of feature selection in data mining," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, Dec 2014, pp. 1–6.

[33] A. Chokniwal and M. Singh, "Faster mahalanobis k-means clustering for gaussian distributions," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2016, pp. 947–952.

[34] A. Labs. (2019, jan) The dot language. [En línea]. Disponible en: https://graphviz.gitlab.io/_pages/doc/info/lang.html

[35] ——. (2019, jan) Graphviz attributes. [En línea]. Disponible en: https://graphviz.gitlab.io/_pages/doc/info/attrs.html