Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

# Grado en Ingeniería Informática

# Informatikako Ingeniaritzako gradua

## Proyecto fin de grado

## Gradu amaierako proiektua

Estudio sobre la privacidad en el aprendizaje federado mediante el desarrollo de un sistema de recomendación online

Ibai Guilén Pacho

Director: Diego Casado Mansilla

Bilbao, mayo de 2021

# Índice general

**1. Antecedentes y justificación** **1**
1.1. Antecedentes . . . . . 1
1.1.1. Sistemas de recomendación . . . . . 1
1.1.2. Privacidad digital . . . . . 3
1.2. Justificación . . . . . 5

**2. Objetivos y Alcance** **7**
2.1. Objetivos . . . . . 7
2.1.1. Generales . . . . . 7
2.1.2. Específicos . . . . . 8
2.2. Alcance . . . . . 9
2.2.1. Dentro del alcance . . . . . 10
2.2.2. Fuera del alcance . . . . . 10

**3. Metodología** **11**
3.1. Consideraciones . . . . . 11
3.2. Metodología de Investigación . . . . . 11
3.3. Metodología de desarrollo . . . . . 13
3.4. Conclusión . . . . . 14

**4. Planificación** **17**

**5. Introducción al Federated Learning** **21**
5.1. Información general . . . . . 21
5.2. Fundamentos y Protocolo . . . . . 22
5.3. Problemas presentes . . . . . 23
5.3.1. Técnicos . . . . . 23
5.3.2. Privacidad . . . . . 24
5.3.3. Seguridad . . . . . 24

**6. Desarrollo** **27**
6.1. Requisitos del sistema . . . . . 27
6.1.1. No Funcionales . . . . . 27
6.1.2. Funcionales . . . . . 28
6.2. Gestión de riesgos . . . . . 29
6.3. Especificación del diseño . . . . . 31
6.3.1. Arquitectura . . . . . 32
6.3.2. Datos involucrados . . . . . 36
6.3.3. Protocolo de Federated Learning . . . . . 40
6.3.4. Securización de las comunicaciones . . . . . 46
6.3.5. Agregación de modelos por consenso . . . . . 50
6.4. Valoración . . . . . 63
6.5. Tecnologías utilizadas . . . . . 64
6.5.1. Sistema de recomendación . . . . . 64
6.5.2. Memoria . . . . . 69
6.5.3. Desarrollo . . . . . 69
6.5.4. Sistemas operativos . . . . . 73
6.5.5. Comunicaciones . . . . . 73
6.6. Consideraciones sobre la implementación . . . . . 74
6.6.1. Privacidad . . . . . 74
6.6.2. Seguridad . . . . . 74
6.6.3. Combinación de modelos . . . . . 75
6.6.4. Limitaciones técnicas . . . . . 75
6.7. Retos abordados en el proyecto . . . . . 77
6.7.1. Convergencia del algoritmo de aprendizaje . . . . . 77
6.7.2. Privacidad . . . . . 77
6.7.3. Seguridad en la comunicación . . . . . 78
6.7.4. Combinación de algoritmos de reducción de comunicaciones . 78
6.8. Incidencias ocurridas . . . . . 80

**A. Protección de Datos en la Unión Europea y España** **83**
A.1. Preludio . . . . . 83
A.2. Legislación . . . . . 84
A.3. Derechos fundamentales . . . . . 86

**B. Documentos adicionales** **91**
B.1. Planificación del Proyecto . . . . . 91

# Índice de figuras


1.1. Recomendaciones de Amazon (Fuente: Amazon[2]) . . . . . 2

3.1. Ciclo empírico de A.D. de Groot (Fuente: Wikipedia[4]) . . . . . 13
3.2. Modelo en el Espiral, Autor Desconocido (Fuente: Blog[5]) . . . . . 14
3.3. Metodología de trabajo del proyecto final de grado (Autor: Ibai Guillén) 15

5.1. Diagrama de una red de Federated Learning . . . . . 22

6.1. Rack para Raspberry Pi (Fuente: Amazon[11]) . . . . . 29
6.2. Raspberry Pi 3 B+ (Fuente: Wikipedia[12]) . . . . . 32
6.3. NVIDIA Jetson Nano 2GB (Fuente: NVIDIA[13]) . . . . . 32
6.4. Switch Genius . . . . . 32
6.5. Mapa de red física . . . . . 33
6.6. Mapa de la red de aprendizaje federado . . . . . 34
6.7. Montaje de la Jetson Nano . . . . . 35
6.8. Montaje del switch . . . . . 35
6.9. Montaje de las Raspberries . . . . . 35
6.10. Intercambio de claves públicas entre los participantes . . . . . 40
6.11. Intercambio de claves públicas entre los participantes . . . . . 41
6.12. Envío de la clave de cifrado del modelo al servidor de agregación . . . 42
6.13. Envío del modelo cifrado al servidor de agregación . . . . . 42
6.14. Diagrama del flujo de reentrenado de los modelos . . . . . 43
6.15. Envío de la clave de cifrado del modelo de cada participante a cada participante . . . . . 44
6.16. Envío de cada modelo cifrado a su correspondiente dueño . . . . . 44
6.17. Diagrama del flujo de la comparación de los modelos . . . . . 45
6.18. Diagrama de flujo del proceso de cifrado . . . . . 48
6.19. Diagrama de flujo del proceso de descifrado . . . . . 49
6.20. Creación de la matriz de atributos de usuario . . . . . 65
6.21. Creación de la matriz de atributos de los elementos . . . . . 66

6.22. Creación de la matriz de interacciones . . . . . 66
6.23. Ajustar datos al modelo . . . . . 67
6.24. Ajustar datos al modelo . . . . . 68
6.25. Opciones de ejecución de las Raspberries . . . . . 72
6.26. Opciones de ejecución de la Jetson Nano . . . . . 72

# Índice de tablas

4.1. Planificación del proyecto . . . . . 20

6.1. Estrategias de persuasión del sistema (Fuente: Elaboración propia). . 37
6.2. Atributos de los usuarios que participan en el sistema (Fuente: Elaboración propia). . . . . 38
6.3. Elementos de la estructura de los datos del ranking (Fuente: Elaboración propia). . . . . 39

# Capítulo 1

# Antecedentes y justificación

## 1.1. Antecedentes

Este proyecto parte del proyecto de R.Sánchez[1] donde se desarrolló un sistema de recomendación de estrategias de persuasión basado en aprendizaje activo. Para comprender los antecedentes que han incurrido en el desarrollo de este proyecto primeró hay que conocer y comprender qué son los sistemas de recomendación y después entender los problemas de seguridad y privacidad que van ligados a estos.

### 1.1.1. Sistemas de recomendación

Los sistemas de recomendación fueron mencionados por primera vez en los años 90 y han ido evolucionando hasta estar implantados en gran parte de las empresas actuales. Estos sistemas son sistemas de filtrado de información, que partiendo de una gran cantidad de datos, tanto del usuario como de los elementos a recomendar, pueden predecir cuál será el elemento más apropiado para el usuario. Estos sistemas están estrechamente relacionados con el marketing, ya que el objetivo es conseguir recomendar un elemento que sea del agrado e interés de los usuarios suele ir ligado a fines económicos.

Hoy en día se usan en multitud de ámbitos, desde las redes sociales hasta las distribuidoras de contenido como Netflix, pasando por compañías de comercio electrónico como Amazon.

A la hora de realizar la recomendación se realizan filtrados de diferente tipo, estos no son más que la forma en la que el sistema busca correlacionar los usuarios con los elementos que estos consumen, compran o ven. Entre los métodos más comunes

para relacionar esta información y obtener resultados eficientes se encuentran:

- Filtros demográficos, que recomiendan en función del sexo, edad, país, oficio,. . .
- Filtros basados en contenidos, como Youtube, que recomiendan contenidos similares a los valorados por los usuarios.
- Filtrado colaborativo, que consiste en recomendar al usuario elementos valorados positivamente por usuarios similares a él.

Sin embargo, existen sistemas híbridos que utilizan varias de las estrategias de filtrado anteriores combinadas. Un ejemplo de ello es el mencionado Amazon, que tiene uno de los algoritmos de recomendación más potentes y eficientes actualmente.

Sin embargo, existen sistemas híbridos que utilizan varias de las estrategias de filtrado anteriores combinadas. Un ejemplo de ello es el mencionado Amazon, que tiene uno de los algoritmos de recomendación más potentes y eficientes actualmente.

Esto se debe a dos cosas, en primer lugar que cuenta con una gran cantidad de información de los usuarios, tanto su edad, género, país, dirección, rutinas (si tienen Alexa), . . . como los artículos que miran, compran, añaden a la lista, etc. Con todo esto el algoritmo es capaz de ofrecer recomendaciones muy precisas a los usuarios, lo que le da una gran calidad de servicio a la empresa gracias a esto.

De hecho, en Amazon, si queremos revisar las recomendaciones tenemos un apartado propio para ellas al que se puede acceder fácilmente (Fig1.1).

Fig. 1.1: Recomendaciones de Amazon (Fuente: Amazon[2])

### 1.1.2. Privacidad digital

Queda claro que un sistema de recomendación depende plenamente de la información de la que dispone, de forma que, de cuanta más información disponga de los usuarios, más precisión tendrá en sus predicciones. Para obtener esta información muchos anunciantes se han valido de diversas herramientas y utilidades, pero si se ha de destacar alguna son las cookies de terceros. Estas, son definidas por Google[3] cómo:

> *"... son archivos que crean los sitios web que visitas para guardar información de la navegación y facilitar tu experiencia en línea. Gracias a las cookies, los sitios pueden mantener abierta tu sesión, recordar tus preferencias del sitio y proporcionarte contenido relevante en función del lugar donde te encuentres. Hay dos tipos de cookies:*
>
> - *Las cookies de origen, que las crea el sitio que visitas. El sitio se muestra en la barra de direcciones.*
> - *Las cookies de terceros, que las crean otros sitios. Parte del contenido que ves en la página web que visitas, como anuncios o imágenes, pertenece a estos sitios."*

A simple vista estas cookies parecen inofensivas, pero el problema radica en su utilización. Aunque suelen ser usadas con fines analíticos por compañías de marketing online, estas se usan para registrar el comportamiento de un usuario y crear un perfil para generar publicidad personalizada. Mediante estas se puede registrar el comportamiento de un usuario en internet, saber sus movimientos, hábitos, páginas que visita, edad, sexo, etc.

En España y en la Unión Europea existe una legislación sobre la privacidad y la protección de datos que restringe y limita los datos que estas empresas pueden registrar de nuestra navegación, el Reglamento General De Protección de Datos (RGPD). En el documento adjunto Anexo A, sección A.2 se puede consultar un breve resumen de la legislación en vigor. En lo referente a las cookies este reglamento presenta varios elementos importantes. En primer lugar, debe haber un consentimiento explicito por el usuario en el consentimiento de la política de cookies. En segundo lugar, la aceptación de las cookies de terceros no ha de ser un impedimento para el uso del servicio. En último lugar, todas las cookies y rastreadores que operen en la web del propietario deberán ser mostradas al usuario en un lenguaje sencillo.

De todos modos esta legislación no impide que se realice un perfilado del usua-

rio, que se haga un seguimiento de este por la red o que se analice el tiempo que pasa el usuario en la página. Pero con la maduración de la tecnología cada vez más gente se preocupa por el uso que le dan estas empresas a los datos y son más críticos con estas prácticas. Esto ha llegado incluso a la política, donde España esta desarrollando una carta pionera de derechos digitales, donde, entre otras cosas, recoge el derecho de los usuarios a no ser perfilados (Anexo A, sección A.3).

Hoy en día existen muchos navegadores y complementos que permiten bloquear este tipo de rastreadores, lo que ha llevado al sector y a Google a tener que explorar nuevas vías para obtener esta información. La solución propuesta por Google ha sido su nuevo sistema Federated Learning of Cohorts, aprendizaje federado de cortes, con el que se compromete a dejar las cookies de terceros.

Lo que a priori puede significar una buena noticia no es para nada la realidad. Este sistema permite a Google y a su navegador agrupar usuarios en base a sus intereses y perfiles geográficos gracias al historial de navegación y a los datos recogidos durante la navegación. Aunque permite que los usuarios no puedan ser distinguidos dentro de un grupo este perfilado supone una persecución y rastreo más grave que la acaecida por las cookies de terceros. En primer lugar, sitúa a Google como principal proveedor de marketing digital, lo que afectaría tanto a las funciones que podrían desempeñar las empresas de marketing digital como al precio que tendrían que pagar por la información.

Ante esta imposición de Google varias empresas se han pronunciado y han expresado su total desacuerdo con la política, navegadores como DuckDuckGo o Brave la han calificado de anticompetitiva, invasiva y abusiva. Esto se debe a que el navegador de Google, Google Chrome, acapara el 64.19 %[1] de la red y podría obligar a utilizar su sistema a muchos anunciantes y sitios web.

Cabe destacar que este sistema aun no ha sido aprobado por la Unión Europea, para ello se debe comprobar que cumpla el RGPD y no vulnere ningún derecho de los ciudadanos.

[1]Fuente: https://gs.statcounter.com/, 27/04/2021

## 1.2. Justificación

Este proyecto es una respuesta a todo lo anterior,al estado de la industria del marketing digital, donde se exprimen al máximo las leyes y normas sobre protección de datos y privacidad digital, y a la actitud de las grandes empresas por perfilar a los usuarios y recoger tanta información de estos como les sea posible. Se quiere demostrar que se puede utilizar el aprendizaje federado preservando la privacidad de los usuarios en un sistema de recomendación, permitiéndoles tener el control total sobre su información en su propio dispositivo.

Además, para ello se utilizarán dispositivos como las Raspberry Pis, las cuales servirán para demostrar que no se necesitan dispositivos especialmente potentes para participar en una red de aprendizaje federado. Evidenciando que es posible implementar un sistema de recomendación respetuoso con el usuario, acorde al RGPD y salvaguardando los derechos digitales.

# Capítulo 2

# Objetivos y Alcance

## 2.1. Objetivos

Los objetivos de este proyecto pueden clasificarse en varios grupos. Por un lado están los objetivos generales, objetivos que a simple vista no tienen hitos específicos de los que no se puede saber su estado o si se han cumplido o no.

Por otro lado se encuentran los objetivos específicos. Estos, al contrario que los generales, son fácilmente identificables, ya que permiten saber si la tarea se ha completado, y en caso contrario, saber en qué estado se encuentra dicha tarea. Dentro de este grupo se encuentran los objetivos específicos de desarrollo, apartado donde se agrupan los objetivos que permiten saber en qué estado de desarrollo se encuentra el proyecto y qué funcionalidades incorpora. Pero además, también se encuentran los objetivos de estudio, donde se incluyen los objetivos que tienen que ver con el estudio de los diferentes experimentos que se realizarán.

### 2.1.1. Generales

El objetivo general de este proyecto es el de desarrollar un sistema de recomendaciones online basado en federated learning. Este sistema se desarrollará sobre otro sistema de recomendaciones ya existente basado en Machine Learning, y se deberá adaptar para su correcto funcionamiento con información descentralizada.

Para respetar la privacidad y derechos de los usuarios, el proyecto deberá incluir la privacidad como patrón de diseño, cumpliendo tanto con las normativas vigentes en Europa y España como con las posibles nuevas medidas que entren en vigor en un futuro.

Por último, el sistema deberá asegurar que la información se quede en el dispositivo y no se comparta ninguna información que no sea la del propio modelo desarrollado por el participante de la red.

Una vez desarrollado implementado el aprendizaje federado se estudiará la diferencia con el sistema de recomendación centralizado, privacidad, seguridad, eficiencia, etc.

### 2.1.2. Específicos

Los objetivos específicos son fácilmente reconocibles y concretos, lo que permite saber con exactitud cuándo se han cumplido los objetivos y cuándo no.

#### 2.1.2.1. De desarrollo

Los objetivos de desarrollo tienen que ver con el correcto desarrollo del sistema de recomendación basado en federated learning.

**Parametrización y configuración:** Se tendrán que configurar las plataformas de desarrollo para poder ejecutar el sistema de recomendación centralizado con mayor rapidez y agilizar el desarrollo del proyecto. También se tendrán que configurar las Raspberries y la Jetson Nano, aprovisionarlas del software necesario para ejecutar en ellas el sistema de recomendación y ejecutarlo sin problemas.

**Desarrollo:** Se desarrollarán todas las características de los sistemas de recomendación derivadas de la fase de investigación.

#### 2.1.2.2. De investigación

Los objetivos de investigación, al contrario que los de desarrollo, tienen que ver con la búsqueda de soluciones a las incógnitas del proyecto. Es decir, suplir las limitaciones derivadas del federated learning además de la formación del propio alumno en esta tecnología.

**Formación:** El alumno deberá formarse en conocimientos sobre Inteligencia Artificial para poder comprender los conceptos. También tendrá que familiarizarse con el código del sistema de recomendación para poder realizar las modificaciones pertinentes en él. Además, el alumno deberá leer artículos científicos sobre federated

learning para comenzar a investigar las soluciones a los problemas derivados del proyecto.

**Sistema de recomendación:** Se deberá investigar la forma de combinar modelos de IA de cada Raspberry. Además, se investigarán y realizarán las modificaciones pertinentes en el sistema de recomendación para cumplir los siguientes puntos:

- Que sea capaz de obtener las recomendaciones para un único usuario en concreto.
- Que sea capaz de guardar y cargar los modelos entrenados de LightFM.
- Que el modelo de LightFM sea capaz de admitir aprendizaje incremental.

#### 2.1.2.3. De estudio

Los objetivos de estudio tienen que ver con el análisis de los resultados del sistema de recomendación basado en federated learning.

**Análisis:** El estudio del sistema tendrá que analizar la diferencia de rendimiento entre el sistema de recomendación con información centralizada y distribuida. También tendrá que analizar cómo afectan los distintos modelos y su combinación al rendimiento del sistema de recomendación.

**Privacidad:** Hay que valorar y analizar la privacidad y seguridad de los datos personales de los usuarios tanto en el modelo central, como en el distribuido. Se deberán evitar los problemas de seguridad y privacidad implícitos de la implementación de un sistema de recomendación basado en federated learning

## 2.2. Alcance

El alcance se centra en concretar qué objetivos entran en el desarrollo del proyecto y cuáles no. Para ello se ha dividido este apartado en dos grupos, los objetivos y tareas que entran en el alcance del proyecto y los que no. La segregación en estos grupos nos permite concretar con más precisión los límites del proyecto, evitando que se amplíe más allá de sus límites.

### 2.2.1. Dentro del alcance

Entran dentro del alcance los objetivos y tareas derivadas del desarrollo de un sistema de recomendación basado en federated learning, tanto las tareas de parametrización y configuración del hardware como el desarrollo de software. Esto incluye todos los objetivos específicos de desarrollo mencionados en el apartado anterior.

Del mismo modo quedan dentro del alcance los objetivos de investigación del apartado anterior, ya que son indispensables para el correcto desarrollo del sistema de recomendación

También se incluye el estudio y análisis de los resultados del sistema de recomendación, así como su precisión y sus inconvenientes, es decir, todo lo comentado en los objetivos específicos de estudio.

Para finalizar se incluirá un estudio de la estabilidad legal del sistema en el tiempo. Esto incluye la corroboración del cumplimiento de las actuales medidas de protección de datos y el estudio de las posibles futuras medidas, legislaciones y derechos que limiten el uso de información en Europa.

Se recogera el desarrollo de todo el proyecto en la memoria técnica que será entregada como proyecto fin de grado.

### 2.2.2. Fuera del alcance

Fuera del alcance queda cualquier estudio de mercado sobre si la solución sería viable económicamente además de cualquier estudio de alternativas a federated learning.

No se incluirá ni se recogerá ninguna legislación fuera del marco jurídico Español y del Ordenamiento jurídico de la Unión Europea. Lo que deja fuera del alcance cualquier derecho, legislación o restricción de cualquier estado u organización ajeno a los comentados.

Queda fuera del alcance también la encriptación de las comunicaciones entre los diferentes dispositivos.

# Capítulo 3

# Metodología

La metodología es un factor muy importante de este proyecto, es la hoja de ruta a seguir, y por ello hay que definirla con precisión y saber cuál es la más apropiada.

## 3.1. Consideraciones

Debido a los objetivos mencionados en el apartado correspondiente anterior, queda claro que existen varios tipos de objetivos en el proyecto. La metodología deberá ayudar a cumplir todos los objetivos específicos para cumplir los objetivos generales. Sin embargo, dentro de los específicos se puede observar que coexisten tres tipos diferentes: de desarrollo, de investigación y de estudio.

Es difícil utilizar una misma metodología para el correcto desarrollo de los tres objetivos, ya que pertenecen a distintas disciplinas, así pues, se ha optado por combinar distintas metodologías para ello. De esta forma se consigue conducir todos los objetivos en una misma dirección. Las metodologías que se combinarán serán:

- Metodología Cuantitativa, metodología de investigación.
- Modelo espiral, metodología de desarrollo de software.

## 3.2. Metodología de Investigación

La metodología cuantitativa es una metodología de investigación que se centra en la recopilación y el análisis de los datos, lo que nos servirá como guía para los objetivos de estudio y el análisis de los resultados de los experimentos.

Esta metodología dice que se deben llevar a cabo varias fases de las que solo se realizarán las siguientes tres:

1. Definir claramente el problema y lo que se quiere hacer.
2. Delimitar el problema, concretar el alcance de la investigación.
3. Revisión de la literatura, buscar los conocimientos necesarios para realizar el estudio.

Después de haber realizado las tres fases de la metodología de la investigación se aplicará la investigación empírica al estudio, ya que encaja perfectamente con el planteamiento de este proyecto. Esta forma de investigación es una forma de obtener conocimiento a través de la observación o experiencia. Además, mediante este método se nos incita a probar distintos experimentos y modificaciones de estos con el objetivo de encontrar resultados diferentes.

Para aplicar esta metodología se usará Ciclo empírico de A.D. de Groot (Fig.3.1). Este ciclo nos ayuda a ordenar y entender las fases del proyecto, a conocer los siguientes pasos y a no perdernos.

En el primer paso se encuentra la observación. Como el alumno ya ha adquirido la formación en el apartado tres de la metodología cuantitativa podrá empezar observando cómo funciona el sistema de recomendación. A medida que se den más vueltas al modelo, esta observación se realizará sobre los resultados de los experimentos a la hora de aplicar federated learning.

En el segundo paso, el de inducción, se plantean las ideas o hipótesis acerca del paso de observación. En la primera vuelta, estas hipótesis e ideas serán sobre lo que ha aprendido el alumno y sobre cómo se aplicará el federated learning al proyecto. En las demás vueltas serán sobre los resultados observados y cómo estos afectan al proyecto y al estudio y qué mejoras podrían realizarse.

Tanto la fase de deducción, donde se formulan los experimentos y pruebas a realizar, como la de pruebas, donde se realizan los experimentos para probar las hipótesis y recopilar datos, se sustituirán por la metodología de desarrollo de software en espiral. De esta forma, el modelo del ciclo empírico quedaría combinado con el de espiral para que la investigación y los experimentos que se están realizando mediante software sean concretos y tengan una correcta gestión (véase fig.3.3).

En el paso de evaluación, se interpretan los resultados de los experimentos de los anteriores dos pasos. En este apartado se hará un gran uso de la estadística para mostrar la información, tanto gráfica como analiticamente.

Se pueden realizar tantas vueltas al ciclo empírico como sean necesarias.

Fig. 3.1: Ciclo empírico de A.D. de Groot (Fuente: Wikipedia[4])

## 3.3. Metodología de desarrollo

El modelo en espiral (Fig.3.2) formará parte del ya mencionado Ciclo empírico y agrupará los pasos de formulación de los experimentos y su ejecución. Se ha optado por incluir esta metodología dentro de otra para detallar más profundamente cómo será el proceso de desarrollo de software derivado de la investigación realizada.

Se ha elegido este modelo entre otros por varias razones importantes:

- Define claramente lo que es un ciclo completo, los pasos a dar y las tareas a realizar. También fija los objetivos al inicio de cada ciclo de la espiral, lo que unido al paso de inducción del ciclo empírico implica que se definen objetivos claros y concisos sobre las hipótesis planteadas.

- Al ser un modelo continuo, se pueden realizar tantas iteraciones sobre la espiral como se necesiten, permitiendo desarrollar tantos experimentos como hipótesis se plantean en el apartado de inducción del ciclo empírico. Además, cada vuelta de la espiral incluye el desarrollo de prototipos (lo que en este caso sería un experimento), que al contar con una fase de integración consigue que puedan coexistir todos en el mismo entorno de desarrollo.

- Cumple con las fases de Deducción y Testeo del ciclo empírico ya que incluye los pasos de estas en la fase de desarrollo de la espiral.

- Por último y más importante, se ha elegido esta metodología porque tiene en cuenta la gestión y análisis de riesgos. Esto tiene gran importancia al tratarse de un proyecto complejo con múltiples dispositivos y tanta cantidad de pruebas a realizar.

Fig. 3.2: Modelo en el Espiral, Autor Desconocido (Fuente: Blog[5])

## 3.4. Conclusión

Partiendo del ciclo empírico, escisión de la metodología de investigación cuantitativa, se hará especial incapié en la continua reflexión y teorización sobre el modelo de fededrated learning implantado.

Para el correcto desarrollo del software, las anteriormente mencionadas fases de deducción y testeo serán incluidas en el modelo de la espiral, consiguiendo así, una gestión eficiente y precisa sobre los pasos a dar tanto en investigación como en el desarrollo de software.

Como se ha comentado anteriormente, la agrupación de estas dos metodologías permitirá combinar la investigación con el desarrollo de software, descubriendo a base de prueba y error y a base de analizar los resultados las mejores soluciones para el sistema de recomendación basado en federated learning.

Por lo cual, una vez el alumno haya definido claramente el problema, su alcance y se haya formado en la tecnología, podrá comenzar a utilizar el ciclo presente en la figura 3.3.

Fig. 3.3: Metodología de trabajo del proyecto final de grado (Autor: Ibai Guillén)

# Capítulo 4

# Planificación

La planificación de este proyecto se ha realizado con un diagrama GANTT. Este esta presente en la tabla 4.1, en caso de querer observar la tabla a una escala mayor esta se encuentra en el anexo B sección B.1.

La planificación de este proyecto es un proceso complejo que ha de contener todos los pasos necesarios para conseguir el objetivo del estudio. Para ello se ha dividido en 4 fases de 3 semanas, empezando la primera el 22 de Febrero de 2021 y terminando la ultima fase el 16 de mayo de 2021.

Las tareas de este proyecto se han agrupado en varios grupos:

- El registro del proyecto.
- La formación.
- Configuración del hardware.
- Adaptación del sistema de recomendación online a aprendizaje federado.
- Desarrollo de la memoria.
- Estudio del rendimiento

A su vez, estos grupos cuentan con subtareas más concretas que permiten programar adecuadamente la duración del proyecto.

- El registro del proyecto.
    - Definición de título y descripción.
    - Registro del proyecto a través del formulario.

- La formación.
  - Formación sobre conceptos de Machine Learning.
  - Formación sobre conceptos de los sistemas de recomendación.
  - Formación sobre conceptos de Federated Learning.
  - Formación sobre LightFM.
  - Familiarización con el sistema de recomendación de R.Sánchez.
- Configuración del hardware.
  - Configuración de las RaspBerrys.
  - Configuración de la JetsonNano.
  - Aprovisionamiento de los Sistemas Operativos.
  - Instalar y ejecutar sistema de recomendaciones.
- Adaptación del sistema de recomendación online a aprendizaje federado.
  - Guardado , cargado y envío de modelos de inteligencia artificial.
  - Agregación de los modelos de inteligencia artificial.
  - Recomendación para un único usuario.
  - Combinar recomendaciones para un único usuario.
  - Combinar recomendaciones para varios usuario.
  - Reentrenar modelos.
  - Visualización de los resultados de rendimiento por gráficas.
- Desarrollo de la memoria.
  - Resumen.
  - Introducción.
  - Antecedentes y Justificación.
  - Objetivos y Alcance.
  - Consideraciones éticas.
  - Metodología.
  - Planificación.
  - Introducción al Federated Learning.

  - Desarrollo.
  - Presupuesto.
  - Conclusiones y Trabajo futuro.
  - Bibliografía.
  - Definiciones y Acrónimos.
  - Anexos.

- Estudio del rendimiento.
  - Estudio del rendimiento de un sistema de recomendaciones centralizado.
  - Estudio del rendimiento de un sistema de recomendaciones de modelos agregados.
  - Estudio de rendimiento del sistemas de recomendaciones final.

| EDT | TÍTULO DE LA TAREA |
|---|---|
| 0 | Registrar proyecto |
| 0.1 | Definición de título y descripción |
| 0.2 | Registro del proyecto a través del formulario |
| 1 | Formación |
| 1.1 | Formación sobre conceptos de Machine Learning |
| 1.2 | Formación sobre conceptos de los sistemas de recomendación |
| 1.3 | Formación sobre conceptos de Federated Learning |
| 1.4 | Formación sobre LightFM |
| 1.5 | Familiarización con el sistema de recomendación de R.Sánchez |
| 2 | Configuración del hardware |
| 2.1 | Configuración de las RaspBerrys |
| 2.2 | Configuración de la JetsonNano |
| 2.3 | Aprovisionamiento de los Sistemas Operativos |
| 2.4 | Instalar y ejecutar sistema de recomendaciones |
| 4 | Adaptación del sistema de recomendación online a aprendizaje federado |
| 4.1 | Guardado , cargado y envío de modelos de inteligencia artificial |
| 4.2 | Agregación de los modelos de inteligencia artificial |
| 4.3 | Recomendación para un único usuario |
| 4.4 | Combinar recomendaciones para un único usuario |
| 4.5 | Combinar recomendaciones para varios usuario |
| 4.6 | Reentrenar modelos |
| 4.7 | Visualización de los resultados de rendimiento por gráficas |
| 3 | Desarrollo de la memoria |
| 3.1 | Resumen |
| 3.2 | Introducción |
| 3.3 | Antecedentes y Justificación |
| 3.4 | Objetivos y Alcance |
| 3.5 | Consideraciones éticas |
| 3.6 | Metodología |
| 3.7 | Planificación |
| 3.8 | Introducción al Federated Learning |
| 3.9 | Desarrollo |
| 3.10 | Presupuesto |
| 3.11 | Conclusiones y Trabajo futuro |
| 3.12 | Bibliografía |
| 3.13 | Definiciones y Acrónimos |
| 3.14 | Anexos |
| 5 | Estudio del rendimiento |
| 5.1 | Estudio del rendimiento de un sistema de recomendaciones centralizado |
| 5.2 | Estudio del rendimiento de un sistema de recomendaciones de modelos agregados |
| 5.3 | Estudio de rendimiento del sistemas de recomendaciones final |

Tabla 4.1: Planificación del proyecto

# Capítulo 5

# Introducción al Federated Learning

Es de gran importancia comprender y conocer lo que es el Federated Learning para entender tanto el desarrollo, como la resolución de los problemas y retos que supone su implementación, por eso, en el presente apartado se explicará a conciencia de que se trata esta tecnología.

## 5.1. Información general

El aprendizaje federado, es una técnica de aprendizaje automático que entrena un algoritmo a través de diversos dispositivos descentralizados, manteniendo siempre la información en cada uno de ellos (Fig5.1). El algoritmo que se busca entrenar en los dispositivos puede ser de cualquier tipo dentro del campo del aprendizaje automático, redes neuronales artificiales, aprendizaje profundo, árbol de decisión, etc. Por lo cual, se puede decir que el aprendizaje federado es una tecnología habilitadora para el entrenamiento de modelos de aprendizaje automático en redes de dispositivos.

Al contrario del aprendizaje automático convencional que reúne toda la información en un mismo computador, el aprendizaje federado busca entrenar un algoritmo en cada uno de los dispositivos que participen en la red, para luego combinarlos y conseguir un modelo de aprendizaje robusto. Los dispositivos que forman parte de esta red son denominados participantes y todos ellos forman la denominada como red de aprendizaje colaborativo. En esta red, todos aprenden de todos sin necesidad de compartir su información, consiguiendo mejorar la precisión y eficacia del algoritmo.

Esta red permite repartir la carga de trabajo entre todos los participantes, pues-

Fig. 5.1: Diagrama de una red de Federated Learning

to que cada uno se encarga de entrenar su propio modelo. Esto es posible gracias a que cada vez los dispositivos móviles tienen procesadores más rápidos y más potentes, lo que permite que sean capaces de realizar tareas más complejas y pesadas. De esta forma, se consigue que cada dispositivo sea propietario exclusivo de su información y no tenga que compartirla con ningún otro. Además, hoy en día los costes de comunicación son superiores a los de computación, por lo que cuanto más acciones se puedan computar en los dispositivos de los participantes menor será el consumo del entrenamiento del algoritmo.

Sin embargo, debido a estas características propias del aprendizaje federado también sufre de grandes inconvenientes. El hecho de que la red está formada por diferentes participantes la hace heterogénea, con diferentes capacidades de cómputo y diferentes velocidades de conexión entre los distintos participantes, lo que puede incurrir en desconexiones o pérdidas de paquetes durante el proceso de aprendizaje.

## 5.2. Fundamentos y Protocolo

En general existen dos tipos de entidades en los sistemas de Federated Learning, los propietarios de la información (participantes de la red) y el propietario del modelo (servidor de Federated Learning). Los propietarios de la información entrenan localmente el modelo con la información de la que disponen, mientras que el propietario del modelo agrega los diferentes modelos recibidos de los participantes para mejorar el algoritmo.

La comunicación puede ser llevada a cabo de diferentes maneras, de hecho, hay diseños de protocolos de aprendizaje federado como el desarrollado por los autores

del trabajo [6], que lo dividen en 3 fases:

- Selección de los participantes: El servidor elige un grupo de dispositivos conectados para participar en la red.
- Configuración: El servidor se configura en base al mecanismo de agregación definido y envía a los participantes el modelo global.
- Reporte: El servidor recibe la actualización de cada modelo de su respectivo participante. Más tarde los agrega usando el algoritmo elegido.

Además, estos autores definen un parámetro de población con el objetivo de limitar el acceso en caso de que haya muchos participantes, evitando que se conecten a la vez y saturen la red.

## 5.3. Problemas presentes

### 5.3.1. Técnicos

Como se ha mencionado antes, implementar el aprendizaje federado implica problemas y costes que hay que asumir y tratar con el objetivo de crear un sistema eficiente y estable.

Cada actualización del modelo conlleva millones de parámetros, además se llevan tantas actualizaciones a cabo como precisión se quiera en el modelo. La gran dimensión de las actualizaciones puede incurrir en un aumento de los costes de conexión y generar un cuello de botella que puede verse agravado por los participantes, ya sea por la asimetría en las velocidades de internet o por la asimetría en la capacidad de cómputo. Es por eso, que para mejorar el rendimiento y reducir el coste de comunicación se deben considerar varias cosas.

- En primer lugar, todos los dispositivos deben contar con una conexión estable, preferiblemente conexión por cable, o en su defecto conexión inalámbrica por Wi-Fi, descartando por completo todas aquellas conexiones por telefonía móvil.
- En segundo lugar, para minimizar el número de rondas de comunicación se puede realizar computación adicional en los dispositivos para que la agregación global se realice antes.

- También se puede reducir el tamaño de las comunicaciones, ya sea por compresión del modelo, compresión de los datos, o por el uso de una pequeña muestra representativa de estos en vez de enviarlos por completo.
- Por último, se pueden minimizar las actualizaciones de los modelos en base a la importancia de estas. Evitando enviar el modelo global hasta que este no suponga una clara mejora para los participantes.

### 5.3.2. Privacidad

Uno de los objetivos principales del Federated Learning es la protección de la privacidad de los usuarios. Los usuarios solo necesitan compartir los parámetros del modelo de ML, en lugar de compartir los datos reales. No obstante, varios estudios han demostrado que estos modelos aún pueden ser susceptibles a ataques informáticos que puedan extraer datos reales.

Véase un ejemplo de vulnerabilidad con FaceScrub, un dataset con más de cien mil imágenes de caras de 530 personas. Mientras se desarrollaba el modelo ML, los desarrolladores se percataron de que, tan solo con inspeccionar el modelo ML, los atacantes podían extraer el 90 % de la información real.

Por mucho que se haya avanzado en el desarrollo de modelos que protejan a los usuarios contra posibles ataques informáticos, esta susceptibilidad a exploits aun sigue patente. Con tan solo acceder al modelo ML de un sistema de reconocimiento por voz, es posible lograr extraer datos como la etnia o género de los usuarios [7].

En un artículo publicado por expertos en ciberseguridad, se especifica otro caso en el que, mediante ataques relativamente sencillos contra plataformas de la talla de Amazon Machine Learning, es posible extraer modelos basados en árboles de decisión o en regresión logística [8].

### 5.3.3. Seguridad

Los modelos de datos de Federated Learning son también susceptibles a ataques de envenenamiento de datos. Un participante puede ir entrenando datos y enviándolos al servidor para su posterior procesamiento y análisis. Sin embargo, otro participante puede crear una serie de datos corruptos y enviarlos al servidor con el fin de generar falsos parámetros. Este tipo de ataques, con la desclasificación de los procesos Deep Learning como principal objetivo, tienen una posibilidad de éxito del 90 % si

se ejecutan bien [9].

A diferencia de los ataques de envenenamiento de datos, cuyo objetivo es la generación de datos corruptos para alterar el resultado del modelo generado, los ataques de envenenamiento por modelo buscan alterar el modelo generado en su totalidad. Cuando el modelo de datos generado a través de Federated Learning es muy grande y cuenta con una gran cantidad de participantes, este tipo de ataque tiende a ser más exitoso [10].

El motivo del éxito de este método es que la alteración y generación de datos corruptos para el ataque mencionado anteriormente depende directamente de la cantidad de datos que se encuentre en el servidor. Al contrario, en el caso de este tipo de ataques, el modelo generado es envenenado en su totalidad, independientemente de la cantidad de datos que este albergue.

Existen maneras de prevenir este ataque, como por ejemplo, analizando si el modelo generado por un participante puede ayudar a mejorar el modelo global de datos. Si se da el caso de que no, este modelo será marcado como potencialmente peligroso, y se analizarán los cambios realizados en el modelo de manera exhaustiva para comprobar si se trata de un atacante.

Otra solución parecida consiste en recorrer el servidor y ver si un modelo es muy diferente del resto de modelos. Si es muy diferente, se realiza un análisis de este para comprobar de primera mano si se trata de un atacante.

# Capítulo 6

# Desarrollo

## 6.1. Requisitos del sistema

Como bien se ha mencionado previamente, este proyecto trata de adaptar el sistema de recomendación realizado por R.Sánchez [1] a un sistema de recomendación basado en Federated Learning que permita el aprendizaje colaborativo de los participantes. Es por ello que para conocer los requisitos iniciales de diseño del sistema de recomendación se remite al trabajo del autor.

En este apartado se presentarán únicamente los requisitos de la adaptación del sistema y se partirá del trabajo realizado por R.Sánchez. Se presentarán dos tipos de requisitos. En primer lugar los requisitos no funcionales de diseño, las condiciones que debe cumplir la especificación del diseño para cumplir los objetivos del proyecto. En segundo lugar, los requisitos funcionales, las condiciones que ha de cumplir el sistema para ser capaz de funcionar sobre el hardware del que se dispone.

### 6.1.1. No Funcionales

En cuanto a los requisitos no funcionales se encuentran todos aquellos que se han tenido en cuenta a la hora de plantear el diseño del sistema, es decir, todos los inherentes de la privacidad como patrón de diseño. Por ello el sistema :

**RNF1** Mantendrá toda información de los usuarios en sus dispositivos.

**RNF2** No expondrá la información de los modelos de inteligencia de los usuarios a otros usuarios.

**RNF3** No discriminará entre los modelos de inteligencia artificial a la hora de tomar decisiones.

**RNF4** No perjudicará deliberadamente al modelo de un participante.

**RNF5** No compartirá información que no sea sintética.

**RNF6** No extraerá información de los modelos de los participantes.

**RNF7** Combinará los diferentes modelos de los participantes sin acceder a su información.

**RNF8** Cumplirá tanto con la LOPD-GDD como con el RGPD (véase ApéndiceA, sección A.2).

**RNF9** Respetar los derechos digitales de los usuarios y cumplir con la carta de derechos digitales de España (véase ApéndiceA, sección A.3).

### 6.1.2. Funcionales

Los requisitos funcionales son básicos en este proyecto para que el sistema sea compatible y ejecutable en el hardware. Por ello, para que el sistema sea compatible con el harware se han definido los siguientes requisitos:

**RF1** El sistema ha de ser ejecutable bajo la arquitectura ARMv8.

**RF2** El sistema ha de ser ejecutable bajo la arquitectura x86 y x64.

**RF3** El sistema ha de ser ejecutable bajo la arquitectura AArch64.

**RF4** El sistema ha de ser ejecutable en Linux.

**RF5** El sistema ha de ser ejecutable en Windows.

Además, al margen de los requisitos de compatibilidad también se encuentran los requisitos de recursos, y es que teniendo un hardware poco potente es imprescindible que el software sea lo más óptimo posible para aprovecharlo al máximo. Es por eso que también se definen los siguientes requisitos:

**RF6** El sistema ha de ser capaz de entrenar los modelos en un espacio de tiempo razonable, siendo este siempre inferior a los 100 segundos por cada 10 epochs.

**RF7** El sistema de recomendación no puede colapsar el dispositivo que lo ejecute,

**RF8** El sistema de recomendación debe permitir que el dispositivo pueda funcionar con normalidad durante su ejecución.

## 6.2. Gestión de riesgos

El motivo de esta sección viene de la definición del modelo en espiral como metodología clave para el desarrollo de este proyecto. Este modelo recoge la identificación y gestión de riesgos como pieza fundamental del ciclo de vida, y en un proyecto con tal complejidad técnica y con tantos dispositivos involucrados, es indispensable tener un plan de acción ante cualquier incidente.

Con tantos dispositivos que gestionar, el proceso de configuración de estos puede llegar a ser algo pesado, y más si alguno de estos falla o se rompe. Si se rompiese o se corrompiese una tarjeta SD de una Raspberry Pi habría que reinstalar y configurar todos los paquetes a mano, proceso que puede ser bastante largo puesto que requiere de interacción humana. Para evitar esto se ha tomado la dedcisión de crear un script de aprovisionamiento tanto para la Nvidia Jetson Nano como para las Raspberry Pis, de esta manera, en caso de que una tarjeta SD se averiase, bastaría con cambiarla, configurar el protocolo SSH para acceder remotamente a ella y ejecutar el script. Cuando el proceso terminase el dispositivo estaría listo para poder ejecutar el proyecto sin ningún problema.

Las tareas de inteligencia Artificial no son especialmente livianas para un dispositivo, lo que durante un entrenamiento prolongado en el tiempo podría incurrir en un aumento de temperatura de este. Para evitar que los dispositivos alcancen temperaturas elevadas o que sufran cualquier tipo de cortocircuito por su manipulación inadecuada se ha optado por instalarlos en un rack (Fig 6.1). Esta carcasa aparte de proteger los dispositivos viene equipada con ventiladores que ayudaran a que la temperatura de estos sea siempre la adecuada.

Fig. 6.1: Rack para Raspberry Pi (Fuente: Amazon[11])

Durante el desarrollo se tiene que acceder a estos dispositivos en multitud de ocasiones, para ello se utiliza el protocolo SSH. Sin embargo, al realizarse este proyecto en un ámbito doméstico, existe la probabilidad de que el router sea reiniciado o apagado por cualquier causa externa al proyecto. En tal caso puede que las direcciones IP sean sustituidas y haya que realizar un análisis de la red o acceder fisicamente a estos dispositivos para averiguar la IP. Para evitar este proceso se ha instalado el paquete *AVAHI*[1] que permite acceder mediante SSH a las raspberrys sustituyendo la dirección IP por el nombre de servidor de los dispositivos.

Para gestionar los cambios que se irán haciendo sobre el software sin problemas es indispensable contar con herramientas de control de versiones. Estas herramientas permiten gestionar cuándo se han realizado qué cambios y en caso de necesitarse se podrían revertir con facilidad. En este caso, tanto para el desarrollo del proyecto como para el desarrollo de la memoria se utilizará Git como controlador de versiones. Además, se utilizará Github como repositorio de código en la nube, para así poder clonar el repositorio a los dispositivos facilmente.

[1] AVAHI es un servicio que facilita el descubrimiento de dispositivos en la red local a través de mDNS/DNS-SD.

## 6.3. Especificación del diseño

Teniendo en cuenta los anteriores requisitos y los objetivos del proyecto la especificación del diseño concretará como se han diseñado las diferentes funcionalidades y como se ha implementado el Federated Learning en el proyecto. Para ello se dividirá el capitulo en las siguientes secciones: Arquitectura, Datos, Modelo de aprendizaje federado.

En primer lugar se describirá la arquitectura del proyecto, tanto la configuración de los dispositivos como el como se interconectan y comunican entre sí. En segundo lugar se explicará como se han generado los datos sintéticos y cómo y para que se han utilizado. Por último lugar se especificará el modelo de aprendizaje federado, donde se describirán tanto el protocolo de entrenamiento como el protocolo de comunicación.

### 6.3.1. Arquitectura

En este apartado se explicarán los componentes que toman parte en la arquitectura, como se deben conectar para emular una red de aprendizaje federado y el propio montaje físico de los dispositivos. De esta forma, se podrá comprender porque se han seleccionado los dispositivos que van a participar en el aprendizaje federado, cual es el mapa de red de una red de aprendizaje federado, como se va a representar localmente y como queda el montaje de todos los dispositivos acorde al mapa de red.

#### 6.3.1.1. Componentes

Para la recreación de un ejemplo real de una red de aprendizaje distribuido se han utilizado dispositivos de capacidad de cómputo acotada como son las Raspberry Pi. De esta forma, se pretende demostrar la viabilidad de la aplicación del aprendizaje federado a campos como el del internet de las cosas, donde los dispositivos participantes disponen de escasa capacidad de cómputo.

De la misma forma, con el objetivo de demostrar que no se necesita un supercomputador para realizar una agregación de modelos, se ha utilizado la NVIDIA Jetson Nano como núcleo central para ello.

Los dispositivos que se utilizarán en concreto son los siguientes: (x4) Raspberry Pi 3 B+ (Fig. 6.2), (x1) NVIDIA Jetson Nano 2GB (Fig. 6.3) y (x1) Genius switch (Fig. 6.4).

Fig. 6.2: Raspberry Pi 3 B+ (Fuente: Wikipedia[12])

Fig. 6.3: NVIDIA Jetson Nano 2GB (Fuente: NVIDIA[13])

Fig. 6.4: Switch Genius

#### 6.3.1.2. Mapa de red

Las conexiones entre los dispositivos que interactúan en la red (Raspberries y Jetson Nano) será por cable, conectando todos al mismo switch para permitir la comunicación entre ellos. A su vez, este switch se conectará al router que provee de acceso a internet, permitiendo así que los ordenadores de la red puedan acceder a estos dispositivos y estos dispositivos puedan conectarse a internet para lo que necesiten. Todo ello queda representado en el mapa de de la figura 6.5, que sería el mapa de red físico del que parte el proyecto.

Fig. 6.5: Mapa de red física

Esta arquitectura de red lo que permite es que desde el ordenador 1 se pueda acceder a cualquier dispositivo mediante ssh para su configuración y simulación. A su vez, estos dispositivos permanecen físicamente separados los unos de los otros mientras mantienen acceso a internet para realizar tareas de descarga de paquetes, actualizaciones, envío de datos, etc. En este caso, para la transferencia de archivos y ficheros se utilizará el protocolo scp, ya que no entra dentro del alcance el desarrollo de ningún sistema de comunicación y el protocolo scp es fácilmente combinable con el protocolo ssh.

Sin embargo, desde el punto de vista de la red de aprendizaje federado que quiere simular, el mapa de red incluiría los siguientes cambios:

- Cada participante tendría su propia red de área local.
- La conexión al servidor de agregación se haría a través de internet.
- Los participantes no tendrían ninguna conexión ni relación entre sí.

De este modo, el mapa de la red de aprendizaje federado representado en el proyecto quedaría acorde a la figura 6.6, aunque el mapa de red sobre el que se opere físicamente sea el de la figura 6.5.

Fig. 6.6: Mapa de la red de aprendizaje federado

#### 6.3.1.3. Montaje físico

Una vez realizados todos los mapas y diagramas para la configuración de la red se procedió a conectar los dispositivos acorde al mapa de red físico (fig. 6.5). Para este montaje se acoplaron las Raspberries que representan a los participantes en el rack mencionado en la sección de gestión de riesgos 6.2 mencionada en este mismo capítulo. Después de conectar todos los cables el montaje quedó como se puede ver en las siguientes imágenes.

Fig. 6.7: Montaje de la Jetson Nano

Fig. 6.8: Montaje del switch

Fig. 6.9: Montaje de las Raspberries

La Jetson Nano y cada Raspberry tienen un cable ethernet conectado al switch, el cual a su vez cuenta con otro cable ethernet por el cual se conecta al router (no presente en las fotos).

### 6.3.2. Datos involucrados

Este proyecto al tratarse de una modificación del sistema de recomendación de R.Sánchez se heredarán todos los datos. Cabe mencionar que R.Sánchez utilizó información proporcionada por el proyecto Green Soul[14] para crear su sistema de recomendación. El proyecto Green Soul trataba de persuadir a los usuarios para que aumentase su conciencia energética y cambiaran sus hábitos de consumo, para ello, se realizaron una serie de encuestas donde se les preguntaba a los usuarios por sus datos personales y se les pedía que ordenaran en función de su criterio qué estrategias de persuasión les parecían más efectivas.

Por tanto, las estrategias de persuasión que se recomiendan pertenecen al ámbito de la concienciación del consumo energético, aunque podrían ser extrapolables a otros casos. De todas formas, se eludirán los detalles sobre estas estrategias ya que no son el caso de estudio y no entran en el alcance de este proyecto.

Teniendo en cuenta todo lo anterior, este capítulo se ha redactado para entender la gestión de los datos en este proyecto, ya que algunos son de carácter personal y es de suma importancia preservar su privacidad. Estos datos se pueden dividir en tres categorías:

- **No sensibles**, la información de los elementos a recomendar, que no son ni personales ni suministrados por ninguna persona.
- **Sensibles**, datos que hacen referencia a información personal o que son proporcionados por las personas involucradas en el cuestionario.
- **Sintéticos**, datos de usuarios generados aleatoriamente para paliar la necesidad de datos personales del modelo de consenso (sección 6.3.5).

#### 6.3.2.1. No sensibles

Como se ha mencionado antes, en la categoría de datos no sensibles se encuentran los datos que no son personales ni hacen referencia a ningún usuario. En esta categoría únicamente aparecen aquellos datos que hacen referencia a los elementos que se van a recomendar en el sistema y sus atributos.

Cada estrategia de persuasión que pueda recomendar el sistema contará con hasta dos atributos para que de esta forma el sistema pueda conocer mejor las estrategias y realizar recomendaciones más precisas en base a esta información. En la tabla 6.1 se

muestran las estrategias de persuasión involucradas en el sistema de recomendación y sus atributos.

| ID | Descripción | Atributo 1 | Atributo 2 |
|---|---|---|---|
| **V2** | Reconocimiento público/social de mi contribución al ahorro de energía. | Reconocimiento social | - |
| **V5** | Recibir información relacionada con la energía de una forma simple y estéticamente atractiva. | Atractivo físico | - |
| **V6** | Recibir beneficios como recompensa por mejorar mi rendimiento energético(horarios de trabajo flexibles, saltarse ciertas tareas, etc.). | Condicionamiento | - |
| **V7** | Recibir un reconocimiento por lograr ahorros de energía de manera colectiva yo y mi equipo. | Reciprocidad | Reconocimiento social |
| **V10** | Mis altos gerentes están comprometidos con el ahorro de energía. | Autoridad | Demostración social |
| **V11** | Poder monitorear mi propio desempeño energético en tiempo real. | Monitorización propia. | - |
| **V15** | Información sobre el efecto que mis acciones pueden tener sobre el consumo de energía. | Causa efecto | - |
| **V17** | Evacuación comparativa de mi desempeño en el ahorro de energía respecto a usuarios parecidos a mí. | - | - |
| **V19** | Consejos y sugerencias sobre el ahorro de energía al día o a la semana. | Sugerencia | - |
| **V20** | Avances y consejos sobre las lecciones aprendidas de usuarios parecidos mí en acciones específicas de ahorro de energía. | Similaridad | - |

Tabla 6.1: Estrategias de persuasión del sistema (Fuente: Elaboración propia).

#### 6.3.2.2. Sensibles

En cuanto a los datos sensibles hay que mencionar dos grupos, los datos personales de los usuarios y los rankings de los usuarios. Este primer grupo está formado por los datos personales de los usuarios participantes. Estos intervienen en dos puntos clave: en el entrenamiento de los modelos de inteligencia artificial y a la hora de agregar estos modelos. El grupo de los datos de los rankings está formado por las preferencias de los anteriores usuarios a la hora de ordenar las estrategias de persuasión, es decir, el ranking de estrategias para cada usuario.

Tanto los datos personales de los usuarios como los datos de los rankings de estos son necesarios para entrenar un modelo de inteligencia artificial. Los atributos de los usuarios, al igual que los atributos de las estrategias de persuasión, sirven para relacionar los distintos usuarios con los diferentes rankings y de esta forma poder realizar recomendaciones en base al tipo de usuario.

##### 6.3.2.2.1. Datos de usuarios

Los datos personales de los usuarios que intervienen en este proyecto son los presentes en la tabla 6.2. Estos datos han sido transformados a objetos json para su posterior utilización, formando una lista de objetos en los que cada uno representa los datos personales de un usuario.

| ID | Nombre | Descripción |
|---|---|---|
| **0** | Edad | Rango de edad |
| **1** | Género | Género |
| **2** | Educación | Nivel educativo |
| **3** | País | País de residencia |
| **4** | Cultura de trabajo | Cultura de trabajo |
| **5** | PST | Tipo de perfil de usuario en base a los propuestos por Dan Lockton[15] : *"Pinball, Shortcut, Thought-ful"* |
| **6** | Barreras | Barreras ante el cambio |
| **7** | Intenciones | Intenciones ante el cambio |
| **8** | Confianza | Confianza ante el cambio |

Tabla 6.2: Atributos de los usuarios que participan en el sistema (Fuente: Elaboración propia).

##### 6.3.2.2.2. Rankings

Los usuarios a parte de introducir sus datos personales anteriores también introducen el orden de las estrategias de persuasión en base a la relevancia que consideran que tienen. Este ranking de estrategia es recogido y transformado a formato json formando una lista de objetos json en la que cada objeto representa los datos de la tabla 6.3.

| Nombre | Descripción |
|---|---|
| **User ID** | Identificador único del usuario |
| **Item ID** | Identificador único de la estrategia de persuasión |
| **Ranking** | Valor en el ranking de la estrategia de persuasión |

Tabla 6.3: Elementos de la estructura de los datos del ranking (Fuente: Elaboración propia).

#### 6.3.2.3. Sintéticos

En cuanto a los datos sintéticos se debe tener en cuenta que los datos sensibles han sido directamente suministrados por los usuarios, por lo cual es técnicamente imposible acceder a ellos desde el servidor de agregación, ya que sería una evidente violación de la privacidad de los usuarios. Esto genera un gran problema debido a que el propio modelo de consenso, explicado en la sección 6.3.5, requiere de datos de usuarios para funcionar.

La forma de abordar el problema de la necesidad de datos es mediante la creación de usuarios sintéticos como se explica en la subsección 6.3.5.1. Estos usuarios sintéticos tendrán los mismos atributos que los usuarios normales (tabla 6.2), tendrán que clasificar las mismas estrategias de persuasión (tabla 6.1) y el ranking se gestionará de la misma forma (tabla 6.3).

### 6.3.3. Protocolo de Federated Learning

El protocolo de Federated Learning que se va a reproducir en este proyecto es una adaptación de lo que proponen los autores de [16]. En este protocolo se definen tres fases importantes, la selección de participantes, la configuración del mecanismo de agregación y el envío reporte de cada participante.

Para este caso se han introducido leves cambios en este protocolo que se irán explicando en las sucesivas partes del mismo.

#### 6.3.3.1. Selección de participantes

La selección de participantes al tratarse de un caso de estudio concreto no se limita ni se controla de ninguna forma, simplemente se ha implementado una red en la que participan cuatro dispositivos diferentes.

#### 6.3.3.2. Intercambio de claves

El intercambio de claves es una de las novedades introducidas sobre el protocolo previamente mencionado, en este paso el objetivo es que los diferentes dispositivos de la red compartan sus claves públicas para poder asegurar la privacidad de la red, imagen 6.11.

La utilización de estas claves públicas y su función están definidas en el apartado de securización de las comunicaciones 6.3.4.

Fig. 6.10: Intercambio de claves públicas entre los participantes

#### 6.3.3.3. Entrenamiento

El entrenamiento es algo que en el protocolo se omite pero que es importante mencionar. Este proceso se lleva a cabo en el dispositivo del participante y conlleva dos tareas:

- En primer lugar este participante crea su modelo de inteligencia artificial, lo entrena con los datos que él elija y lo almacena.

- En segundo lugar este modelo es cifrado y enviado al servidor. El proceso de cifrado se puede observar en el diagrama 6.18 de la sección de securización de las comunicaciones 6.3.4.

Fig. 6.11: Intercambio de claves públicas entre los participantes

#### 6.3.3.4. Comunicación con el servidor

La comunicación con el servidor es un proceso crucial en el que se debe asegurar que los modelos de los participantes lleguen sin modificaciones y sin ser interceptados por ningún atacante. Para securizar este proceso se ha trabajado en una sección donde se explica más a fondo los pasos llevados para conseguir que las comunicaciones sean seguras, sección 6.3.4.

Debido a la seguridad de la red los participantes deberán enviar tanto los modelos cifrados como las claves cifradas como se puede observar en las siguientes imágenes.

Fig. 6.12: Envío de la clave de cifrado del modelo al servidor de agregación

Fig. 6.13: Envío del modelo cifrado al servidor de agregación

#### 6.3.3.5. Agregación de los modelos

En este proceso como tal no va a ocurrir una agregación de modelos, sino más bien una compartición de conocimiento de forma indirecta mediante el modelo de consenso. Este sistema está explicado con detalle en la sección 6.3.5 de este capítulo.

De todas formas para verlo de una forma más sencilla se puede observar el siguiente diagrama 6.14 en el cual se muestra el flujo de los datos y de los modelos para ser agregados. El proceso que recorre este diagrama puede ser dividido en 4 partes:

- Se reciben los modelos de los participantes (en este caso 4), se descifran siguiendo el diagrama 6.19 explicado con detalle en en la sección de securización de las comunicaciones 6.3.4 y se guardan los modelos para su posterior utilización.
- Una vez se cuente con los modelos descifrados se generan los usuarios sintéticos, después, con el modelo de cada participante se realiza la predicción del ranking

para estos usuarios y se almacena para su posterior utilización.

- Mediante el modelo de consenso explicado en el apartado 6.3.5 estas predicciones son convertidas a una única predicción por usuario sintético generado. Después la información se almacena para su uso final.

- Por último se coge toda la información de los usuarios sintéticos y los rankings consensuados para reentrenar el modelo de cada participante.

Fig. 6.14: Diagrama del flujo de reentrenado de los modelos

#### 6.3.3.6. Comunicación con los participantes

La comunicación con los participantes se realiza de la misma forma que los participantes se comunican con el servidor, la única diferencia es que el receptor en este caso son los participantes.

Fig. 6.15: Envío de la clave de cifrado del modelo de cada participante a cada participante

Fig. 6.16: Envío de cada modelo cifrado a su correspondiente dueño

#### 6.3.3.7. Comparación de modelos

En el momento que el modelo llega al participante este tiene que valorar si realmente le supone una mejora o no, para ello como puede verse en el diagrama 6.17 existen varias fases a llevar a cabo.

En primer lugar, al igual que como con el servidor de agregación se debe descifrar el modelo recibido. Después, compararlo con el modelo del participante, esto se puede hacer tratando de analizar la precisión de las predicciones para varios subconjuntos de datos de test con los que cuente el dispositivo. En resumen, predecir para usuarios existentes su ranking con los dos modelos, y quedarse con el modelo que más acierte.

Fig. 6.17: Diagrama del flujo de la comparación de los modelos

### 6.3.4. Securización de las comunicaciones

Para preservar la seguridad y privacidad en el protocolo de aprendizaje federado propuesto es imprescindible securizar las comunicaciones. Para securizar las comunicaciones se han llevado a cabo dos acciones clave, la primera en lo que concierne al protocolo de comunicación y la segunda en cuanto al cifrado de la comunicación.

#### 6.3.4.1. Protocolo de comunicación

En cuanto al protocolo de comunicación se ha optado por la utilización del protocolo SCP (Secure Copy Protocol), conocido en español como protocolo de copia segura. Este protocolo requiere del protocolo SSH (Secure Shell), conocido en español como terminal seguro.

El protocolo SSH permite el acceso remoto a un servidor o dispositivo por un canal seguro a través de una clave. Además, el protocolo usa técnicas de cifrado que impiden que la información que viaje en la red lo haga de forma legible. El protocolo SCP permite la transferencia de archivos entre dispositivos de la red usando el protocolo SSH como base.

#### 6.3.4.2. Cifrado de la comunicación

Aunque las comunicaciones en el protocolo SCP y SSH vayan cifradas son susceptibles de ser atacadas. Para mejorar este aspecto se ha realizado un sistema de cifrado de los modelos que viajen por la red para preservar la privacidad y seguridad de los participantes.

Para esta labor se han utilizado el método criptográfico de criptografía híbrida, que usa tanto la criptografía simétrica como asimétrica.

**6.3.4.2.1. Cifrado simétrico** El cifrado simétrico es aquel sistema de cifrado en el que la misma clave sirve para cifrar como para descifrar el mensaje.

El problema de este sistema es que toda la seguridad reside en la clave empleada en el cifrado. Esto es un gran impedimento a la hora de gestionar comunicaciones debido al cómo gestionar el envío de esta clave, ya que si no se hace por un canal seguro el mensaje sera fácilmente descifrable.

Por otro lado, este sistema cuenta con dos grandes ventajas, es sencillo, es rápido y requiere menos recursos para llevarlo a cabo.

**6.3.4.2.2. Cifrado asimétrico** El cifrado asimétrico es aquel sistema de cifrado que consta de dos claves criptográficas, una pública y una privada. La clave pública es utilizada para cifrar el mensaje mientras que la privada es utilizada para descifrarlo.

Este sistema cuenta con tres grandes desventajas que impiden su uso en exclusivo o su uso con grandes volúmenes de información a cifrar.

- Se necesita más procesado de cómputo para generar la clave.
- Las claves asimétricas son mucho más grandes que las simétricas.
- El mensaje cifrado ocupa más espacio que el original.

Debido a que los dispositivos con los que se cuenta no tienen una gran capacidad de cómputo no se podría establecer un tamaño de clave lo suficientemente grande como para poder cifrar el modelo de inteligencia artificial completo. Este modelo pesa alrededor de los 7Kb y para cifrarlo con este sistema habría que generar una clave del mismo tamaño mayor.

Sin embargo, este sistema al constar de dos claves independientes es más sencillo salvaguardar la privacidad y seguridad de los integrantes.

**6.3.4.2.3. Cifrado híbrido** Teniendo en cuenta los anteriores dos tipos de cifrado se decidió utilizar el sistema híbrido para obtener los resultados más óptimos, protegiendo la privacidad y pudiendo realizar las tareas de cifrado desde dispositivos con menor capacidad computacional.

El proceso de cifrado se ha realizado de acuerdo al diagrama de flujo representado en la figura 6.18. En este diagrama se pueden observar los pasos que se han realizado para la encriptación de la información, en primer lugar el modelo de LightFM se convirtió a bytes y se comprimió para reducir el volumen de datos a enviar por la red, pudiendo agilizar así las comunicaciones. Después se cifró simétricamente, generando una clave y un mensaje cifrado. Por un lado este mensaje cifrado sería comprimido y se guardaría en un fichero para su posterior envío. Por otro lado, la clave de cifrado simétrica se encriptó asimétricamente con la clave pública del receptor del mensaje, dando como resultado una clave de cifrado simétrico cifrada

asimétricamente. Esta clave sería guardada en un fichero para ser enviada al igual que el mensaje.

Fig. 6.18: Diagrama de flujo del proceso de cifrado

#### 6.3.4.3. Descifrado del mensaje

El descifrado del mensaje es el proceso inverso al cifrado de la figura 6.18. Para ello, en un primer momento se reciben tanto el modelo cifrado simétricamente como la clave del modelo cifrada asimétricamente. En primer lugar, la clave que se ha recibido se descifra asimétricamente con la clave privada propia, ya que esta sólo es descifrable por el receptor. Una vez descifrada la clave, se descomprime el modelo y se descifra con esta. Por último, el resultado es descomprimido y convertido a objeto LightFM para su posterior utilización.

Fig. 6.19: Diagrama de flujo del proceso de descifrado

### 6.3.5. Agregación de modelos por consenso

A la hora de agregar los diferentes modelos de los participantes se estudiaron los distintos algoritmos de los sistemas de aprendizaje federado, pero todos tenían algo en común, accedían a los datos del modelo. Aunque estos datos de los modelos sean meras matrices y números, el objetivo de este proyecto era no acceder en ningún momento a la información presente en los modelos de inteligencia artificial de los participantes, por lo que se buscó otra solución.

La solución llegó con la idea de que no hacía falta combinar los distintos modelos para crear uno, sino que más bien, deberían compartir sus formas de operar ante ciertos casos. Esta idea se empezó a explotar mediante la generación de usuarios sintéticos, los cuales cada modelo de cada participante de la red de aprendizaje federado debería predecir su ranking. Cuando cada participante predice el ranking para cada usuario sintético están demostrando cómo actuarían ante ciertas entradas de datos, por lo que si se compartiera este conocimiento los participantes se verían beneficiados de las formas de operar de otros participantes.

El problema es que, desde un punto de vista objetivo, puede que algunas formas de operar no sean del todo precisas, por lo que no se debería compartir los rankings predichos por todos los modelos sin ningún filtrado. Para realizar este filtrado y con el objetivo de llegar a un punto intermedio, donde las predicciones que da cada modelo para cada usuario se combinan y todos los modelos lleguen a un acuerdo para el ranking de un usuario, se creó el modelo de consenso.

Este modelo de consenso se explicará en los siguientes apartados, donde se explicará cómo se generan los datos sintéticos, como se generan los rankings, las explicaciones matemáticas del consenso y el proceso de reentrenado de los modelos.

#### 6.3.5.1. Generar datos personales sintéticos

Uno de los elementos imprescindibles para poner en práctica el sistema de agregación por consenso es la necesidad de datos de usuarios, que con el objetivo de evitar cualquier violación de privacidad serán creados en el servidor, siendo de ahora en adelante denominados como usuarios sintéticos.

Estos usuarios sintéticos tendrán los mismos atributos que cualquier usuario del sistema, es decir, contarán con todos los atributos presentes en la tabla 6.2. Como ya se ha dicho antes, el proceso de creación de estos datos es crucial para el correcto

funcionamiento del sistema, ya que si los usuarios generados fueran de un perfil concreto el aprendizaje no sería favorable para todos los participantes de la red, puesto que lo más seguro es que únicamente el participante con más usuarios de este perfil sea beneficiado mientras que para los demás participantes supondrá una pérdida de precisión por el ruido introducido en su modelo. Es por eso que se ha tomado la decisión de generarlos aleatoriamente, evitando así estancarse en perfiles concretos y tratando de favorecer a todos los participantes.

#### 6.3.5.2. Generar ranking de los usuarios

Los datos de los usuarios sintéticos por sí solos no son suficientes ni para entrenar ni para mejorar ningún modelo de inteligencia artificial, hace falta saber el ranking que estos usuarios sintéticos darían a las diferentes estrategias de persuasión, tabla 6.1.

Esta cuestión no puede ser abordada aleatoriamente como la anterior, ya que cada usuario ordena en base a su opinión y sus preferencias las estrategias de persuasión, por lo cual, de hacerlo aleatoriamente se distorsionaría la realidad y no sería representativo. Además, si este ranking se generase aleatoriamente y se compartiera con los participantes de la red, distorsionaría por completo sus modelos, lo que llevaría a una pérdida inmensa de precisión.

Es debido a esta problemática que se ha creado el método de consenso. Este método trata de representar la *"opinión"* de cada participante de la red mediante la estimación del ranking para cada usuario sintético, todas estas *"opiniones"* serán puestas en común mediante el algoritmo de consenso, lo que dará como resultado un ranking por usuario sintético que represente a todos los usuarios de la red.

#### 6.3.5.3. Consenso

El proceso de consenso comienza con la ya mencionada creación de usuarios sintéticos aleatoriamente, pero la generación del ranking de estos se lleva a cabo de la siguiente manera.

**6.3.5.3.1. Estructuras básicas** En primer lugar hay que definir la lista de las estrategias $E$ a ser ordenadas por los modelos $m$ de los participantes, que en este caso serán las mismas que las de la tabla 6.1. Estas estrategias serán representadas como $e_j$, siendo $j$ la cantidad de estrategias de la lista $E$ y $e_j$ la estrategia final de esta lista.

$$E = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & \cdots & e_j \end{bmatrix}$$

A la hora de crear una predicción de ranking, denotada $r$, para las estrategias $e$ de $E$ se representará cada estrategia con un valor $a$, siendo $a$ el puesto de la estrategia de persuasión en el ranking. Por lo cual, $a$ ha de ser siempre menor o igual que el número total de estrategias y mayor que cero, para así mantenerse dentro del rango de $E$ ($0< a \leq j$), puesto que cada predicción $r$ no es más que una mera valoración de un usuario de las $j$ estrategias de la lista $E$.

$$r = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & \cdots & a_j \end{bmatrix}$$

De esta forma se puede afirmar que el primer elemento del ranking tendrá el valor 1 y el último el valor $j$, quedándose las demás estrategias con el resto de los puestos en ese rango. Lo que en un conjunto de 10 estrategias se traduce en que la mejor estrategia obtendría el valor 1 y la peor el valor 10.

Sin embargo, Cuando se trata de realizar varias predicciones se simboliza como una matriz $R$ donde cada predicción es representada por $r_i$, en el que $r$ denota la predicción del ranking e $i$ denota la cantidad de predicciones $r$ realizadas.

$$R = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_i \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{pmatrix}$$

**6.3.5.3.2. Predicción** Una vez entendidas las estructuras básicas que se utilizarán en el método de consenso este apartado presentará su aplicación, ya que en este apartado se crearán las predicciones que serán consensuadas a posteriori para generar los datos de reentrenamiento.

Se debe tener en cuenta que las operaciones anteriores cambian por completo cuando se realizan predicciones para un conjunto $U$ de usuarios, y si además estas predicciones se hacen con los diferentes modelos $m$ de los participantes las operaciones se complican bastante.

En primer lugar, se ha de especificar que el conjunto de usuarios $U$ estará formado por los $k$ identificadores de los usuarios, representados como $u_k$. Este conjunto de identificadores se crea cuando se generan los datos personales sintéticos y recoge todos los identificadores de todos los usuarios generados aleatoriamente.

$$U = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & \cdots & u_k \end{bmatrix}$$

Para realizar las distintas predicciones con el modelo $m$ de cada participante de la red de aprendizaje federado hay que tener en cuenta que las predicciones $R$ pertenecerán al modelo $m$, por lo que se denotará a partir de ahora como $R_m$. Además, esta matriz de predicciones $R_m$ estará formada por las distintas predicciones para los distintos usuarios $u$, lo que será representado como $r_u$.

$$R_m = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_u \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{u1} & a_{u2} & \cdots & a_{uj} \end{pmatrix}$$

**6.3.5.3.3. Redistribución de las predicciones** Las predicciones realizadas con cada modelo $m$ sobre el conjunto de usuarios $U$ puede ser interpretada de diferente forma, en un principio se entiende que cada modelo $m$ genera una predicción $R$ para cada uno de los $u$ usuarios generados, pero, del mismo modo, se puede entender que cada usuario $u$ obtiene una predicción $r$ por cada modelo $m$, y que con la unión de todas las predicciones $r$ de todos los modelos $m$ para este mismo usuario $u$ se puede construir una matriz $R$.

Para entender lo anterior con más facilidad se ha de comprender que ya no solo se cuenta con una matriz de predicciones $R$, sino que se cuenta con una matriz $P$ con $m$ matrices de predicciones $R$, denotada como $P_m$. Esta matriz representa todas las predicciones $R$ realizadas por cada modelo $m$ participante en la red de aprendizaje federado para un mismo conjunto de usuarios $U$.

$$P_m = \left( \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{u1} & a_{u2} & \cdots & a_{uj} \end{pmatrix}_{1} \quad \cdots \quad \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{u1} & a_{u2} & \cdots & a_{uj} \end{pmatrix}_{m} \right)$$

Teniendo en cuenta esta estructura de datos, la separación de los usuarios ha de hacerse tomando cada predicción $r_u$ de cada participante de forma separada, uniendo las que corresponden al mismo usuario $u$ en una nueva matriz de predicciones $R$, que al pertenecer al usuario $u$ será denotada como $R_u$. Cada predicción $r$ de $R_u$ es extraída de la predicción $r_u$ de cada matriz de predicciones de cada modelo $R_m$. Además, todas las matrices de predicción $R$ de todos los usuarios $u$ serán agrupadas bajo el termino $P_u$. De esta forma puede afirmarse que $P_u$ está formada por las distintas predicciones $R_u$, las cuales a su vez están formadas por las distintas predicciones $r_u$ extraídas de $R_m$.

$$R_m = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{u1} & a_{u2} & \cdots & a_{uj} \end{pmatrix} \equiv \begin{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_1 \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_u \end{pmatrix}$$

Predicciones del modelo $m$ para $u$ usuarios.

$$P_m = \begin{pmatrix} R_1 & R_2 & \cdots & R_m \end{pmatrix} \equiv \begin{pmatrix} \begin{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_1 \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_u \end{pmatrix}_1 & \cdots & \begin{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_1 \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_u \end{pmatrix}_m \end{pmatrix}$$

Lista de las predicciones de cada modelo $m$ para $u$ usuarios.

$$R_u = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mj} \end{pmatrix} \equiv \begin{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_1 \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_m \end{pmatrix}$$

Predicciones para el usuario $u$ realizadas por los $m$ modelos.

$$P_u = \begin{pmatrix} R_1 & R_2 & \cdots & R_u \end{pmatrix} \equiv \begin{pmatrix} \begin{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_1 \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_m \end{pmatrix}_1 & \cdots & \begin{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_1 \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_m \end{pmatrix}_u \end{pmatrix}$$

Lista de las predicciones para cada usuario $u$ realizadas por los $m$ modelos.

Para concluir con lo anteriormente explicado, este proceso puede definirse como una transformación de las predicciones que da cada modelo para un grupo de usuarios en las predicciones de cada usuario según cada modelo, $P_m \Longrightarrow P_u$. Además, hay que tener en cuenta que el tamaño de las predicciones de cada usuario depende de la cantidad de modelos que participen en la red de aprendizaje federado, de forma que cuantos más modelos participen más predicciones tendrá cada usuario.

El proceso de transformación de $P_m$ en $P_u$ explicado previamente y representado mediante las matrices es un proceso que se lleva a cabo en tres partes, primero el de definición de las variables necesarias, segundo el de predecir de los rankings para los $U$ usuarios con los $m$ modelos y el tercero en el que estas predicciones de los modelos son reordenadas para agruparlas por usuario y no por modelo. El proceso está representado mediante el algoritmo 1.

En este algoritmo la declaración de variables se produce entre las líneas 3 y 9, donde se respeta la nomenclatura utilizada hasta el momento, de forma que, si no se entiende alguna variable se puede acudir a las explicaciones previas para comprender su función.

Después de esto, se realizan las predicciones de los rankings para todos los usuarios generados sintéticamente, donde $P_m$ está formada por *len(m)* matrices, donde cada matriz $R_m$ representa los valores de ranking $r_m$ que da el modelo $m$ para cada usuario $u$.

Una vez habiendo completado las predicciones con todos los modelos y haber obtenido la lista de matrices $P_m$, es el momento de agrupar los datos por usuarios $u$ en vez de por modelos $m$, $P_m \Longrightarrow P_u$. Para este proceso se recorre la lista de usuarios sintéticos $U$ y las predicciones $r_u$ que cada modelo da para cada uno de ellos. De esta forma se consigue que $P_u$ este formada por *len(U)* matrices, donde cada matriz $R_u$ representa los valores de ranking $r_u$ que ha dado cada modelo $m$ para el usuario $u$.

**Algorithm 1** Redistribución de las predicciones

```
1:  procedure OBTENER RANKINGS PARA LOS USUARIOS
2:  Declaración de las variables necesarias:
3:      E ← [...]            ▷ Lista de las estrategias a ordenar
4:
5:      U ← [...]            ▷ Lista de los identificadores de los usuarios sintéticos
6:
7:      P_u ← [len(U)]       ▷ Lista de las predicciones de cada usuario
8:
9:      M ← [...]            ▷ Lista de los modelos de los participantes
10:
11:     P_m ← [len(M)]       ▷ Lista de las predicciones de cada modelo
12:
13: Predicciones de los modelos:
14:     for int(participante) ← 0 to range(len(M)) do
15:         R_m ← M[participante].predict(U)  ▷ Predecir con el modelo del partici-
                                                pante el ranking para los usuarios
16:
17:         P_m[participante] ← R_m           ▷ Añadir las predicciones del modelo
                                                a la lista de todos los participantes
18:     end for;
19:
20: Redistribución de las predicciones en base a usuarios:
21:     for int(usuario) ← 0 to range(len(U)) do
22:         R_u ← [...]      ▷ Lista de las predicciones para el usuario
23:
24:         for int(participante) ← 0 to range(len(M)) do
25:             r_u ← P_m[participante][usuario]   ▷ Predicción del usuario
26:
27:             R_u[participante] ← r_u            ▷ Añadir la predicción a la lista de
                                                     predicciones del usuario
28:         end for;
29:
30:         P_u[usuario] ← R_u   ▷ Añadir la lista de predicciones del usuario a la
                                   lista de predicciones de todos los usuarios
31:     end for;
```

**6.3.5.3.4. Consensuar predicciones** El proceso de consenso es el proceso clave a la hora de agregar los distintos modelos, ya que es gracias a este proceso que se consiguen los datos para reentrenar los modelos. El proceso parte de la lista $P_u$ formada por *len(U)* predicciones $R_u$, a las cuales se les aplica una función estadística *func()* sobre los $a_{j,m}$ elementos de los múltiples rankings para converger en un único valor la diferente lista de valores. Después de aplicar dicha función sobre todas las columnas $j$ de la predicción $R_u$ se consigue unificar cada columna $j$ en un único valor representado como $b_j$, con esto se obtiene una única predicción para el usuario $u$, a la cual denotaremos a partir de ahora como $q_u$. La lista de todas las predicciones consensuadas para el conjunto de usuarios $U$ será denotada como $R_q$.

$$R_u = \begin{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_1 \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} a_1 & a_2 & \cdots & a_j \end{pmatrix}_m \end{pmatrix} \Longrightarrow \begin{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}_1 & \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}_2 & \cdots & \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}_j \end{pmatrix}$$

$$q_u = \begin{pmatrix} func \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}_1 & func \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}_2 & \cdots & func \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}_j \end{pmatrix}$$

$$q_u = \begin{pmatrix} func \begin{pmatrix} a_1 & a_2 & \cdots & a_m \end{pmatrix}_1 & func \begin{pmatrix} a_1 & a_2 & \cdots & a_m \end{pmatrix}_2 & \cdots & func \begin{pmatrix} a_1 & a_2 & \cdots & a_m \end{pmatrix}_j \end{pmatrix}$$

$$q_u = \begin{pmatrix} b_1 & b_2 & \cdots & b_j \end{pmatrix}$$

$$R_q = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_u \end{pmatrix} \equiv \begin{pmatrix} \begin{pmatrix} b_1 & b_2 & \cdots & b_j \end{pmatrix}_1 \\ \begin{pmatrix} b_1 & b_2 & \cdots & b_j \end{pmatrix}_2 \\ \vdots \\ \begin{pmatrix} b_1 & b_2 & \cdots & b_j \end{pmatrix}_u \end{pmatrix}$$

Las funciones para calcular el valor de consenso $b_j$ que se proponen son las métricas estadísticas de la media($\bar{x}$), mediana($\tilde{x}$) y moda ($\hat{x}$). En este proyecto se ha descartado la utilización de la media ya que en caso de que los valores $a_{j,m}$ fueran muy dispares podría distorsionarse bastante el resultado $b$ y perder eficiencia. Por ello solo se utilizarán los algoritmos de mediana($\tilde{x}$) y moda ($\hat{x}$).

$$q_u = \left( \left( a_1 \quad a_2 \overset{\frown}{\cdots} a_m \right)_1 \quad \left( a_1 \quad a_2 \overset{\frown}{\cdots} a_m \right)_2 \quad \cdots \quad \left( a_1 \quad a_2 \overset{\frown}{\cdots} a_m \right)_j \right)$$

Representación del consenso por moda

$$q_u = \left( \left( a_1 \quad a_2 \overset{\sim}{\cdots} a_m \right)_1 \quad \left( a_1 \quad a_2 \overset{\sim}{\cdots} a_m \right)_2 \quad \cdots \quad \left( a_1 \quad a_2 \overset{\sim}{\cdots} a_m \right)_j \right)$$

Representación del consenso por mediana

Con el objetivo de crear una función más precisa de consenso se elaboró un método híbrido el cual combina tanto la métrica de la moda como la de la mediana, algoritmo 2. Este método funciona de la siguiente forma: en primer lugar lee la lista de elementos $a_{j,m}$ y si existe algún valor duplicado más de *len(m)/2* veces automáticamente asigna ese valor como $b_j$, de lo contrario calculará la mediana de la lista y asignaría ese valor como $b_j$. Se podrían tomar otras decisiones en función de la desviación típica de la lista e incluso desarrollar un algoritmo más complejo para optimizar este proceso, pero dicha tarea se sale del alcance del proyecto.

Como elemento adicional se ha considerado la posibilidad de otorgar un peso al modelo $m$ de cualquier participante en concreto, de forma que este tenga más repercusión en el resultado del consenso. Este peso solo será otorgado al modelo que sea más afín al usuario sintético, lo que puede ser calculado por una función o seleccionado por algún atributo en concreto (país, puesto de trabajo, género, etc.). En este caso este peso se da en función del atributo país, de forma que el modelo del país correspondiente al usuario sintético creado será el que tenga más trascendencia a la hora de elaborar el ranking consensuado. Este peso será representado como $W_c$ y su valor será un número entero de rango $0 \leq W_c \leq$ *len(m)/2 +1*. Este rango permite no añadir ninguna importancia extra a ningún modelo $m$ o permitir que sea este modelo en exclusiva el que genere el ranking para este usuario $u$, ya que debido a la función de consenso, cuando existen *len(m)/2 +1* veces un valor en una lista de elementos $a_{j,m}$ este valor es asignado automáticamente por ser moda.

Este peso se otorga directamente sobre la predicción $r_u$ del modelo $m$ cuando este genera $R_m$, al extraerse cada predicción $r_u$ de $R_m$ para formar $R_u$ esta predicción se introducirá $W_c$ veces más en $R_u$ para que a la hora de aplicar *func()* esta predicción $r_u$ de $R_m$ tenga mayor importancia respecto a la del resto. De esta forma, como se ha dicho antes, si $W_c$ fuese cero todos los modelos tendrían la misma importancia a la hora de calcular el consenso y si fuera *len(m)/2 +1* este valor sería la moda y por ello sería su valor directamente asignado a $b_j$.

**Algorithm 2** Consenso de los datos

```
procedure REALIZACIÓN DEL CONSENSO
Declaración de las variables necesarias:
    P_u ← [···]                ▷ Lista de las predicciones de cada usuario

    R_q ← [···]                ▷ Lista de rankings consensuados

Consensuar:
    for int(usuario) ← 0 to len(P_u) do
        R_u ← P_u[usuario]                ▷ Lista de predicciones para un mismo usuario

        columnas, filas ← shape(R_u)      ▷ Obtener la forma de la matriz de predicciones

        q_u ← [columnas]                  ▷ Inicializar lista de valores para el ranking consensuado

        for int(columna) ← 0 to columnas do
            r_i ← R_u[:, columna]         ▷ Lista de todos los valores de la misma columna

            unicos ← unique(r_i)          ▷ Cantidad de valores diferentes en la columna

            if unicos > len(m)/2 then
                q_u[columna] ← mod(r_i)   ▷ Asignar valor por moda
            else
                q_u[columna] ← med(r_i)   ▷ Asignar valor por mediana
            end for;
        end for;
        R_q[usuario] ← q_u                ▷ Guardar valor consensuado en
    end for;
    return R_q;
```

**6.3.5.3.5. Normalización del consenso** El problema es que en función de la fórmula que se aplique el rango de $q_u$ puede que no permanezca dentro del rango de $E$, lo que nos impide afirmar que el valor $b$ de cada estrategia entre dentro del intervalo $0 < a_q \leq j$. Para solucionar este problema se procesan los valores $a_{qj}$ de $q_u$ para que estos permanezcan dentro del rango. Este proceso se realiza intercambiando el valor $a_{qj}$ menor por el valor del ranking menor disponible, como se puede ver en el algoritmo 3.

**Algorithm 3** Normalización de los datos del consenso

```
procedure NORMALIZACIÓN DEL CONSENSO
Declaración de las variables necesarias:
    Indice ← 1        ▷ Se inicializa el elemento más importante del ranking como 1

    q_u ← [···]       ▷ Valores del ranking consensuado
    aux ← q_u         ▷ Variable auxiliar

Normalización del consenso:
    for int(i) ← 0 to len(aux) do
        posicion ← whereIsMin(aux)      ▷ Posición de la lista donde está el
                                          menor elemento

        q_u[posicion] ← Indice          ▷ Se normaliza el valor del ranking

        drop(aux[posicion])             ▷ Se elimina el elemento asignado de la
                                          lista de los siguientes a normalizar.

        Indice ← Indice + 1             ▷ Se incrementa el índice para asignar la
                                          siguiente posición del ranking
    end for;
    return q_u;
```

#### 6.3.5.4. Reentrenado

Como parte final solo queda reentrenar los modelos de los participantes con los usuarios sintéticos y sus rankings consensuados. Para ello se utilizarán los usuarios y sus atributos generados aleatoriamente antes y los rankings $R_u$ donde cada predicción $r_u$ será el verdadero ranking del usuario.

## 6.4. Valoración

La función de valoración es un elemento fundamental en cualquier sistema de recomendación, ya que es la única forma de medir la eficacia, precisión o margen de error del sistema. Esta métrica mide la diferencia entre la predicción realizada y la realidad. Para analizar este rendimiento se utilizará la métrica de rendimiento basado en la distancia normalizada (NDPM, Normalized Distance based Performance Measure). Esta métrica es usada porque al contrario que otras, no solo responde con si la predicción es acertada o no, sino que valora cuanto de acertada es.

## 6.5. Tecnologías utilizadas

### 6.5.1. Sistema de recomendación

El sistema de recomendación de R.Sánchez utiliza LightFM[2] para crear los modelos de inteligencia artificial. Para implementar el aprendizaje federado se ha tenido que modificar gran parte del sistema de recomendación previo y se ha tenido que hacer uso de las utilidades que proporciona LightFM que se van a explicar en los siguientes apartados.

#### 6.5.1.1. Creación del modelo

El paso de creación del modelo es un paso complejo puesto que conlleva la correcta gestión de muchos datos e información. Asimismo, estos datos han de estar correctamente estructurados y construidos, puesto que LightFM requiere que la información este convertida a matrices dispersas y no tolera elementos vacíos en ellas. Para ello LightFM proveé de herramientas de creación de datasets que facilitan la creación de las matrices, de forma que sea más fácil incluirlas en el modelo.

Entre estas herramientas se encuentran:

- *build_user_features(...)* Permite crear la matriz CSR de usuarios y atributos de usuarios.
- *build_item_features(...)* Permite crear la matriz CSR de items y atributos de items.
- *build_interactions(...)* Permite crear las matriz COO de interacciones y las matriz COO de sus correspondientes pesos.

Hay que tener en cuenta que las matrices CSR (Compressed Sparse Row) hacen referencia a las matrices que admiten operaciones matriciales y acceso eficientemente; y que las matrices COO (Coordinate list) hacen referencia a las matrices que soportan modificaciones eficientemente, son generalmente utilizadas para construir matrices.

**6.5.1.1.1. Atributos de los usuario** Para crear el modelo es necesario, entre otras cosas, disponer de las características de los usuarios. En este caso contamos con multitud de atributos sobre cada usuario: edad, género, nivel educativo, país, cultura de trabajo, perfil PST (Pinball, Shortcut, Thought-ful), barreras, intenciones y confianza.

[2]LightFM es una implementación en python de varios algoritmos populares de recomendación.

Para convertir toda esa información del usuario a la matriz CSR que exige LightFM habrá que llamar a *build_user_features(...)* con los IDs de usuario y y sus atributos, formando una lista que tenga listas de los IDs de los usuarios con sus listas de atributos, es decir:

**build_user_features**(*data, normalize=True*) [source]

Build a user features matrix out of an iterable of the form (user id, [list of feature names]) or (user id, {feature name: feature weight}).

Parameters:
- **data** (*iterable of the form*) – (user id, [list of feature names]) or (user id, {feature name: feature weight}). User and feature ids will be translated to internal indices constructed during the fit call.
- **normalize** (*bool, optional*) – If true, will ensure that feature weights sum to 1 in every row.

Returns: **feature matrix** – Matrix of user features.

Return type: CSR matrix (num users, num features)

Fig. 6.20: Creación de la matriz de atributos de usuario

**6.5.1.1.2. Atributos de los elementos** Además de los usuarios y de sus atributos, también se ha de disponer de los elementos a ordenar en el ranking, llamados items, y de sus características. En este caso estos items representan las estrategias de persuasión por las que se preguntó a los usuarios en el cuestionario. Sin embargo, al contrario que en el punto anterior, no contamos con tantos atributos sobre estos items, sino que cada estrategia de persuasión cuenta con dos atributos llamados dimensiones. Estos atributos no se encuentran presente en todas las estrategias, lo que da como resultado que haya algunas que tengan dos, una o ninguna dimension.

Para convertir toda esa información de los items a la matriz CSR que exige LightFM habrá que llamar a *build_item_features(...)* con los IDs de las estrategias y y sus atributos, formando una lista que tenga listas de los IDs de las estrategias con sus listas de atributos, es decir:

**build_item_features**(*data, normalize=True*) [source]

Build a item features matrix out of an iterable of the form (item id, [list of feature names]) or (item id, {feature name: feature weight}).

Parameters:
- **data** (*iterable of the form*) – (item id, [list of feature names]) or (item id, {feature name: feature weight}). Item and feature ids will be translated to internal indices constructed during the fit call.
- **normalize** (*bool, optional*) – If true, will ensure that feature weights sum to 1 in every row.

Returns: **feature matrix** – Matrix of item features.

Return type: CSR matrix (num items, num features)

Fig. 6.21: Creación de la matriz de atributos de los elementos

**6.5.1.1.3. Interacciones** Para crear las interacciones de entrenamiento y de testeo LightFM nos facilita un método para poder crearlo fácilmente. Esta herramienta se utilizará dos veces como mínimo, una con los identificadores y atributos de los usuarios que serán utilizados para entrenar el modelo y la otra para generar las interacciones de los usuarios con los que se quiere realizar las predicciones.

**build_interactions**(*data*) [source]

Build an interaction matrix.

Two matrices will be returned: a (num_users, num_items) COO matrix with interactions, and a (num_users, num_items) matrix with the corresponding interaction weights.

Parameters: **data** (*iterable of (user_id, item_id) or (user_id, item_id, weight)*) – An iterable of interactions. The user and item ids will be translated to internal model indices using the mappings constructed during the fit call. If weights are not provided they will be assumed to be 1.0.

Returns: **(interactions, weights)** – Two COO matrices: the interactions matrix and the corresponding weights matrix.

Return type: COO matrix, COO matrix

Fig. 6.22: Creación de la matriz de interacciones

### 6.5.1.2. Ajuste

Para permitir que el modelo pueda ser reajustado con nuevos datos y permitir el aprendizaje incremental LightFM implementa una variante del la herramienta *fit*$(\cdots)$. En un principio *fit*$(\cdots)$ ajusta el modelo pero si se vuelve a utilizar esta herramienta se reajusta desde cero, sin mantener ningún dato insertado en el modelo antes.

Para solucionar esto LightFM tiene una variante llamada *fit_partial(· · · )*, esta herramienta permite ajustar datos al modelo tantas veces como se quiera sin perder los datos insertados previamente.

**fit_partial**(*interactions, user_features=None, item_features=None, sample_weight=None, epochs=1, num_threads=1, verbose=False*) [source]

Fit the model.

Fit the model. Unlike fit, repeated calls to this method will cause training to resume from the current model state.

For details on how to use feature matrices, see the documentation on the `lightfm.LightFM` class.

**Parameters:**

- **interactions** (*np.float32 coo_matrix of shape [n_users, n_items]*) – the matrix containing user-item interactions. Will be converted to numpy.float32 dtype if it is not of that type.
- **user_features** (*np.float32 csr_matrix of shape [n_users, n_user_features], optional*) – Each row contains that user's weights over features.
- **item_features** (*np.float32 csr_matrix of shape [n_items, n_item_features], optional*) – Each row contains that item's weights over features.
- **sample_weight** (*np.float32 coo_matrix of shape [n_users, n_items], optional*) – matrix with entries expressing weights of individual interactions from the interactions matrix. Its row and col arrays must be the same as those of the interactions matrix. For memory efficiency its possible to use the same arrays for both weights and interaction matrices. Defaults to weight 1.0 for all interactions. Not implemented for the k-OS loss.
- **epochs** (*int, optional*) – number of epochs to run
- **num_threads** (*int, optional*) – Number of parallel computation threads to use. Should not be higher than the number of physical cores.
- **verbose** (*bool, optional*) – whether to print progress messages.

**Returns:** the fitted model

**Return type:** LightFM instance

Fig. 6.23: Ajustar datos al modelo

#### 6.5.1.3. Predicción

Para generar las predicciones LightFM implementa un método que sintetiza el proceso y permite que insertando los interacciones de test se prediga el ranking para estos usuarios.

**predict_rank**(*test_interactions, train_interactions=None, item_features=None, user_features=None, num_threads=1, check_intersections=True*) [source]

Predict the rank of selected interactions. Computes recommendation rankings across all items for every user in interactions and calculates the rank of all non-zero entries in the recommendation ranking, with 0 meaning the top of the list (most recommended) and n_items - 1 being the end of the list (least recommended).

Performs best when only a handful of interactions need to be evaluated per user. If you need to compute predictions for many items for every user, use the predict method instead.

For details on how to use feature matrices, see the documentation on the `lightfm.LightFM` class.

Parameters:
- **test_interactions** (*np.float32 csr_matrix of shape [n_users, n_items]*) – Non-zero entries denote the user-item pairs whose rank will be computed.
- **train_interactions** (*np.float32 csr_matrix of shape [n_users, n_items], optional*) – Non-zero entries denote the user-item pairs which will be excluded from rank computation. Use to exclude training set interactions from being scored and ranked for evaluation.
- **user_features** (*np.float32 csr_matrix of shape [n_users, n_user_features], optional*) – Each row contains that user's weights over features.
- **item_features** (*np.float32 csr_matrix of shape [n_items, n_item_features], optional*) – Each row contains that item's weights over features.
- **num_threads** (*int, optional*) – Number of parallel computation threads to use. Should not be higher than the number of physical cores.
- **check_intersections** (*bool, optional, True by default,*) – Only relevant when train_interactions are supplied. A flag that signals whether the test and train matrices should be checked for intersections to prevent optimistic ranks / wrong evaluation / bad data split.

Returns: the [i, j]-th entry of the matrix will contain the rank of the j-th item in the sorted recommendations list for the i-th user. The degree of sparsity of this matrix will be equal to that of the input interactions matrix.

Return type: np.float32 csr_matrix of shape [n_users, n_items]

Fig. 6.24: Ajustar datos al modelo

#### 6.5.1.4. Evaluación

La evaluación de los resultados obtenidos de las predicciones de los modelos es un apartado fundamental que permite analizar la eficacia de estas predicciones. Además, la utilización de métricas y técnicas permitirá poder comparar los resultados entre sí para poder demostrar si la predicción es correcta, aceptable o errónea.

Para este caso no se incluirá ninguna métrica ni método de LightFM, sino que se utilizará la métrica NDPM implementada y explicada en el apartado de valoración 6.4.

### 6.5.2. Memoria

La memoria técnica de este proyecto ha sido elaborada en LaTeX. Esta ha sido escrita en el editor de código *Visual Studio Code* gracias al complemento *LaTeX Workshop*, que proporciona las características básicas para la composición tipográfica de LaTeXen Visual Studio Code, y gracias a *MiKTeX*, distribución Tex para Windows, la cual permite la compilación de documentos LaTeXen Windows.

Además, para controlar el avance y las versiones de la memoria se ha utilizado el controlador de versiones GIT. Para que las versiones no permaneciesen en el ordenador local se decidió abrir un repositorio en GitHub[3] donde se ha ido desarrollando este proyecto.

Para la gestión bibliográfica se ha utilizado *Zotero*, herramienta que permite gestionar las citas bibliográficas a través de una app y un complemento en el navegador. Para exportar estas referencias a LaTeXse ha utilizado el paquete *Bibtex*.

### 6.5.3. Desarrollo

El desarrollo de este proyecto se ha realizado en el editor *Visual Studio Code* y con el leguaje de programación *Python*, al igual que la memoria se ha utilizado el controlador de versiones GIT con un repositorio en Github. A lo largo del desarrollo del proyecto han surgido diferentes necesidades que se han ido solucionando con la incorporación de ciertas herramientas, estas herramientas han sido agrupadas en función de las necesidades que han solventado en el proyecto, se detallan a continuación.

#### 6.5.3.1. Matrices

A la hora de gestionar matrices de grandes dimensiones se ha utilizado la biblioteca *NumPy*, la cual da soporte para crear grandes matrices multidimensionales y realizar operaciones matemáticas de alto nivel con ellas.

Se han usado matrices NumPy tanto para la gestión de la información y estructurar los datos en LightFM como para realizar las tareas del modelo de consenso.

#### 6.5.3.2. Cálculos estadísticos

A la hora de realizar los cálculos sobre las matrices de predicción en el modelo de consenso la mediana y moda han tenido que ser aplicadas con distintas librerías.

[3]Repositorio: $https://github.com/Ibaii99/PFG_MEMORIA$

Aunque se ha intentado utilizar *NumPy* para todo esta biblioteca no contaba con ningún método que permitiera calcular la moda de una lista, por lo cual ha habido que importar una biblioteca extra, *SciPy*.

*SciPy* permite aplicar bastantes algorítmos y herramientas matemáticas, pero solo se ha importado su paquete de funciones estádisticas *stats*.

#### 6.5.3.3. Guardado de los modelos

Uno de los principales problemas del proyecto era el envío de los modelos y para ello se necesitaba exportar el modelo a un fichero para poder enviarlo por la red. Como LightFM permite la serialización de sus modelos se elijió la biblioteca *pickle* para este proceso. Esta biblioteca implementa módulos binarios para la serialización y deserialización, por lo cual permitia tanto guardar a fichero el modelo como importarlo desde un fichero.

#### 6.5.3.4. Securización

Para la implementación de la seguridad en las comunicaciones y realizar las operaciones criptográficas se ha utilizado el paquete *cryptography*, este paquete provee de métodos de cifrado y descifrado simétricos y asimétricos de alto nivel, lo que permite un nivel de abstracción mayor y por ende una más fácil gestión.

#### 6.5.3.5. Compresión

Un elemento fundamental del proyecto era minimizar el volumen de las comunicaciones puesto que es uno de los grandes inconvenientes del aprendizaje federado. Con el objetivo de minimizar el tamaño de las comunicaciones se identificaron donde se daban los mayores volúmenes de conexión y se averiguó que era en el intercambio de modelos entre los participantes y el servidor y viceversa. Para ello la compresión de las comunicaciones se realizo mediante la biblioteca *zlib*.

#### 6.5.3.6. Gestión de la ejecución

Como en este proyecto se gestionan muchos dispositivos y se tiene que reproducir en protocolo de aprendizaje federado en orden es crucial poder controlar lo que se ejecuta en cada dispositivo. Para poder gestionarlo con más facilidad se ha implementado el paquete *OptionParser* para poder definir ciertos *flags* para especificar que proceso se quiere ejecutar en el dispositivo.

En cuanto a los flags generales se encuentrán:

- -*k*, para generar las claves públicas y privadas de encriptación.
- -*e*, para encriptar los modelos con la clave pública del receptor.
- -*d*, para desencriptar el modelo recibido con la clave privada propia.

Sin embargo, debido al funcionamiento dispar del nodo respecto a los participantes hay diferentes *flags* dependiendo del dispositivo, de esta forma las Raspberries cuentan con:

- -*t*, para generar un modelo LightFM y entrenarlo con los datos del país.
- -*c*, para comparar el modelo reentrenado recibido con el que tiene el dispositivo.
- -*v*, para realizar el entrenado del modelo y la encriptación.
- -*w*, para desencriptar el modelo reentrenado y compararlo con el local.
- -*a*, flag extra solo usado en desarrollo que permite simular virtualmente a los participantes, su única utilidad ha sido durante el desarrollo y no durante el sistema de recomendación.

A su vez, el nodo de agregación ha tenido también sus diferentes tareas:

- -*s*, para generar los datos sintéticos.
- -*c*, consensuar los datos sintéticos con los modelos de los participantes.
- -*v*, desencriptar el modelo del participante, generar los datos sintéticos, realizar el consenso de los datos y encriptar los modelos reentrenados.

```
pi@pi4:~/fl_st_recommender $ python3 src/models/raspberry_procedure.py -h

Usage: raspberry_procedure.py [options]

Options:
  -h, --help            show this help message and exit
  -k, --generatekeys    generate device public and private keys
  -t, --trainmodel      train local model and export to file
  -e, --encryptmodel    encrypt trained local model and export to file
  -d, --decryptmodel    decrypt external model and export to file
  -c, --comparemodel    compare local and external model
  -v, --trainencrypt    train and encrypt
  -w, --decryptcompare  decrypt and copare models
  -a, --all             process to all users
```

Fig. 6.25: Opciones de ejecución de las Raspberries

```
nvidia@nvidia:~/Documents/fl_st_recommender$ python3 src/models/jetsonnano_procedure.py -h
Usage: jetsonnano_procedure.py [options]

Options:
  -h, --help            show this help message and exit
  -k, --generatekeys    generate device public and private keys
  -d, --decryptmodels   decrypt participant models
  -c, --consense        rebase models with synthetic retrained models
  -e, --encryptmodels   encrypt retrained models
  -s, --synthdata       generate synth data
  -v, --decryptconsenseencrypt
                        decrypt received models, consense and encrypt for each
                        participant
```

Fig. 6.26: Opciones de ejecución de la Jetson Nano

#### 6.5.3.7. Aprovisionamiento

Uno de los riesgos del proyecto que se ha incluido en la sección de gestión de riesgos ha sido la creación de un *script* que permita automatizar la instalación de todos los paquetes necesarios para la ejecución del proyecto en caso de que una tarjeta SD se averiase.

Los *scripts* que se han desarrollado para cumplir con este objetivo han sido dos, uno de aprovisionamiento del sistema linux y otro de requisitos de paquetes python. En un proyecto normal hubiera valido con un simple script de requisitos de python *requirements.txt* el cual alberga el nombre de los paquetes y la versión a instalar, proceso fácilmente realizable mediante el comando *pip install -r requirements.txt*. Sin embargo, en este proyecto los paquetes que se utilizan no son fácilmente importables de un sistema operativo a otro, por lo que ha habido que desarrollar un pequeño script de aprovisionamiento en *bash* que mediante consola instalase todos los paquetes necesario para el correcto funcionamiento de los paquetes python en las Raspberries y la Jetson Nano.

Cada tipo de dispositivo ha tenido su propio conjunto de ficheros de aprovisionamiento, de forma que ha habido un *requirements.txt* y un script en *bash* para las Raspberries y otro para la Jetson nano, ya que por sistema operativo, arquitectura y

compatibilidades los dispositivos deben funcionar con distintas versiones de ciertos paquetes y en caso de las Raspberries con algún paquete más.

#### 6.5.3.8. Reporte de resultados

Con el objetivo de representar los resultados de la precisión de los rankings de una forma vistosa y fácilmente entendible se incorporó la biblioteca *Matplotlib* para la generación de gráficos. Esta biblioteca permite crear diversidad de gráficos para representar la información, elemento imprescindible a la hora de comparar el rendimiento del sistema.

### 6.5.4. Sistemas operativos

En cuanto a sistemas operativos se ha trabajado tanto con *Windows* a la hora de realizar el desarrollo como con *Linux* a la hora de realizar la implementación. Los sistemas *Linux* que se han utilizado han sido *RaspbianOS*, sistema operativo derivado de *Debian* para las Raspberries con soporte oficial, y una adaptación de *Ubuntu* para la Jetson Nano.

### 6.5.5. Comunicaciones

Como bien se ha explicado en el apartado de securización de las comunicaciones 6.3.4 para comunicarse con los dispositivos y poder parametrizarlos y configurar la instalación se utilizará el protocolo *SSH*, el cual permite abrir una línea de comando desde un dispositivo a otro.

Además, para el intercambio de ficheros se ha utilizado el protocolo *SCP*, el cual se apoya en gran parte del protocolo *SSH* y al igual que este está explicado en el apartado de securización de las comunicaciones 6.3.4.

## 6.6. Consideraciones sobre la implementación

En este capitulo se describirán tanto las causas que han llevado a adoptar la previa especificación del diseño del sistema como las razones detrás de ellas. Con el objetivo de ayudar a entender el porqué de las decisiones tomadas se hablará sobre la privacidad, seguridad, la combinación de los modelos y las limitaciones técnicas presentes en este proyecto.

### 6.6.1. Privacidad

Con el objetivo de no vulnerar la privacidad de los datos, la agregación de los modelos será realiza por consenso sobre usuarios sintéticos. De esta forma, se evitan problemas derivados del robo de datos y las consecuencias tanto legales como personales de este.

Partiendo del principio de que si no circula ningún dato de ningún usuario por la red no se corren tantos riesgos se plantea este sistema de agregación como una alternativa viable a los sistemas tradicionales de ensamblado.

### 6.6.2. Seguridad

Este sistema está expuesto a los ataques de seguridad derivados de la implementación del Federated Learning. Sin embargo, debido a la naturaleza del algoritmo de consenso, algunos ataques podrían ser minimizados o incluso erradicados.

Un ejemplo de ello es el ataque por envenenamiento. Existen dos tipos de envenenamiento, por datos y por modelo, pero ambos consisten en que un participante de la red, con el objetivo de dañar tanto al sistema como a los participantes, envía modelos de inteligencia artificial (o los datos si es por envenenamiento de datos) debidamente manipulados. En los sistemas de agregación de modelos de inteligencia artificial por ensamblado este tipo de ataque suele ser bastante dañino, ya que es capaz de distorsionar por completo los resultados del modelo final con lo enviado por el atacante.

Sin embargo, debido a la condición del sistema de agregación por consenso, este tipo de ataques no son tan efectivos, ya que el modelo con datos envenenados podría ser ignorado si el método por el que se realiza el consenso es la moda. Con la moda se elegiría el ranking en función de los valores más frecuentes, es decir, el modelo del atacante sería ignorado por presentar predicciones que no se ajustan a ninguno de

los demás participantes.

Cabe destacar que con el objetivo de detectar ataques y encontrar posibles modelos maliciosos se podrían descartar del proceso todos aquellos modelos que presenten unas predicciones totalmente diferentes. Se podría analizar la varianza de las predicciones de cada modelo para descartar del proceso a presuntos modelos envenenados.

### 6.6.3. Combinación de modelos

Debido a que el principal objetivo de este proyecto es el desarrollo de un sistema de recomendación con aprendizaje federado con la privacidad como patron de diseño, la combinación de los diferentes modelos de los participantes ha sido un factor muy importante a tener en cuenta. En el Machine Learning tradicional esta agregación de modelos se producía mediante el ensamblado de modelos. Sin embargo, este proceso no es compatible con este proyecto puesto que para poder ensamblar los modelos habría que compartir información de los participantes de la red.

Para evitar compartir ningún tipo de información que comprometiese la seguridad y privacidad de los participantes de la red de aprendizaje federado pero conseguir que los participantes aprendiesen entre ellos, se optó por compartir los modelos de inteligencia artificial exclusivamente.

### 6.6.4. Limitaciones técnicas

Debido a que los dispositivos que iban a formar la red iban a ser 4 Rasberry Pis modelo 3b+, el procesado de los datos era un proceso de especial importancio por la *escasa* capacidad de computo de estos dispositivos. Estos dispositivos, como se puede ver en el capítulo de especificación del diseño, son dispositivos que tienen cuatro núcleos y 1.4GHz de frecuencia, lo que comparado con un ordenador de sobremesa actual supone una gran diferencia de potencia de cálculo.

Además, esa no es la unica diferencia que tienen respecto a los ordenadores de escritorio, las Raspberry Pis funcionan sobre arquitectura ARMv8 y no sobre las tan usadas x64 y x86 . Esto implica un gran problema a la hora de instalar los paquetes necesarios para que el proyecto sea ejecutable en estos dispositivos, ya que la mayoría de los paquetes necesarios están pensados para las arquitecturas más comunes (x64 y x86). Sin embargo, se han podido encontrar las versiones funcionales de todos los paquetes para estos dispositivos.

El dispositivo que iba a agregar los datos, la Nvidia Jetson Nano, ha sido más sencilla de configurar puesto que esta funciona sobre arquitectura AArch64, extensión de 64bits de la arquitectura ARM.

## 6.7. Retos abordados en el proyecto

El aprendizaje federado es una tecnología nueva que con muchas vías de investigación, el objetivo de esta sección es explicar las vías abiertas que se han estudiado y abordado.

### 6.7.1. Convergencia del algoritmo de aprendizaje

Una de las consideraciones principales del Federated Learning es la convergencia de los distintos modelos. Se trata de optimizar los modelos, minimizando los datos que se suben a estos. No obstante, la convergencia no está siempre garantizada, pero ya se cuenta con proyectos dedicados a lograr la plena convergencia del modelo global.

Este proyecto trata de abordar este reto desde otro punto de vista, en vez de intentar combinar los algoritmos tratar de que estos compartan el conocimiento a través de usuarios sintéticos como se ha explicado anteriormente. Esta solución, además, permite que este enfoque se pueda aplicar a cualquier caso de uso debido a que no tiene dependencias respecto al framework de inteligencia artificial que se esté usando. Los algoritmos clásicos de FedAvg y FedSGD están implementados en Tensorflow, pero si se quisieran usar en otro sistema habría que desarrollarlos o buscar alguna solución ajena al framework, mientras que el algoritmo de consenso es un sistema que permite converger los resultados de una matriz de predicciones en una simple lista. Que a su vez esta lista será usada para reentrenar a los participantes y mejorar su precisión. Concepto que puede ser aplicado a multitud de sistemas de recomendación.

### 6.7.2. Privacidad

El reto de la privacidad es cada vez más relevante en un mundo donde todo es cada vez más digitalizado. El Federated Learning cuenta con la privacidad como una de sus principales ventajas, pudiendo utilizar el modelo de aprendizaje de manera local, intercambiando únicamente los parámetros necesarios con el modelo global alojado en el servidor.

No obstante, un atacante puede obtener acceso a cierta información confidencial si logra acceder al modelo global. Para cubrir esta carencia, se proponen diferentes soluciones en las que se sacrifica algo de rendimiento con el fin de lograr la plena seguridad, y se está logrando un equilibrio mayor con el paso del tiempo.

Este reto ha sido solucionado mediante el enfoque de reentrenado en vez de el clásico enfoque de converger los modelos. Esto quiere decir que, en este caso, los modelos no se *"fusiona"*, sino que cada uno se reentrena con datos sintéticos para los cuales se ha estimado un resultado a través del método de consenso. De esta forma, se evita que un participante albergue en su modelo información de otro participante y podrá evitar que su información este en los dispositivos de otros.

### 6.7.3. Seguridad en la comunicación

Un reto al que se ha tenido que enfrentar el Federated Learning ha sido la exposición a ciberataques, principalmente mediante interferencias o DoS. El ataque por interferencias consiste en enviar señales de alta frecuencia para generar interferencias en las comunicaciones entre los dispositivos móviles y el servidor, algo que provoca errores en los datos transferidos entre los clientes y el servidor, y que reduce de forma drástica el rendimiento del modelo.

Sin embargo, este reto se puede abordar de manera exitosa enviando copias del modelo a través de diferentes frecuencias para reducir el riesgo de que el modelo se corrompa de algún modo.

En este proyecto la seguridad de la comunicación ha sido un punto fundamental, ya que todas las comunicaciones son cifradas con cifrado híbrido, y en caso de que un participante no recibiese su modelo por desconexión podría pedirlo de nuevo al nodo principal. Sin embargo, no se ha tenido en cuenta la conexión inalámbrica ni los ataques de denegación de servicio, esto sería un nuevo campo a estudiar y no quedaría resuelto del todo.

### 6.7.4. Combinación de algoritmos de reducción de comunicaciones

Actualmente, hay tres técnicas de reducción de comunicaciones en Federated Learning, cuya combinación puede resultar realmente efectiva a la hora de reducir el tamaño de los datos intercambiados con el modelo global. Por ahora, se están desarrollando técnicas que permitan una buena combinación de estos algoritmos con la finalidad de maximizar la eficiencia del sistema global de Federated Learning.

Con el objetivo de maximizar la eficiencia de las comunicaciones se han implementado mecanismo de compresión de los bytes del modelo y de la información a enviar

por la red, reduciendo con zlib considerablemente el tamaño de los mensajes.

## 6.8. Incidencias ocurridas

Durante el desarrollo del proyecto ocurrieron varios problemas con los dispositivos implicados que se han decidido mencionar.

En el principio del desarrollo del proyecto se optó por la utilización de otro modelo de Raspberries, menos potentes y más antiguas, que eran incompatibles con las versiones de los paquetes que había que utilizar. Tras muchas pruebas y después de intentar compilar uno a uno cada paquete para el dispositivo se decidió desistir y cambiarlos por las actuales Raspberry Pi 3B +, mucho más actuales.

Al realizar el cambio de los cuatro dispositivos y tener que reconfigurar todas las tarjetas sd se perdió gran parte del tiempo, puesto que las primeras fases de la configuración han de realizarse con la Raspberry conectada a una pantalla con teclado y ratón para habilitar los diferentes protocolos de comunicación. Una vez con esta configuración, se ejecutaron los *script*s de aprovisionamiento preparados para reinstalar todos los paquetes, lo que permitió que el proceso no se alargara mucho más.

Cuando parecía que todo estaba solucionado una Raspberry falló y se rompió, lo que provocó que hubiera que desmontar todo el rack 6.1 y poner una Raspberry nueva. Este proceso al tratarse únicamente de hardware y al intercambiarse por el mismo modelo no trajo mayores problemas puesto que se podía utilizar la misma tarjeta SD con la misma configuración.

# Bibliografía


[1] R. Sánchez-Corcuera, D. Casado-Mansilla, C. E. Borges y D. López-de-Ipiña, "Persuasion-based recommender system ensambling matrix factorisation and active learning models," en, *Pers Ubiquit Comput*, mar. de 2020, ISSN: 1617-4909, 1617-4917. dirección: http://link.springer.com/10.1007/s00779-020-01382-7 (visitado 24-04-2021).

[2] *Amazon.es: compra online de electrónica, libros, deporte, hogar, moda y mucho más.* es-es. dirección: https://www.amazon.es/ (visitado 25-04-2021).

[3] *Cómo borrar, habilitar y administrar cookies en Chrome - Computadora - Ayuda de Google Chrome.* dirección: https://support.google.com/chrome/answer/95647?co=GENIE.Platform%3DDesktop&hl=es-419 (visitado 26-04-2021).

[4] *Investigación empírica*, es, Page Version ID: 127477170, jul. de 2020. dirección: https://es.wikipedia.org/w/index.php?title=Investigaci%C3%B3n_emp%C3%ADrica&oldid=127477170 (visitado 20-04-2021).

[5] *El Ciclo de Vida del Software — Proceso Básico en Metodologías*, es, oct. de 2016. dirección: https://okhosting.com/blog/el-ciclo-de-vida-del-software/ (visitado 20-04-2021).

[6] K. Bonawitz y col., "Towards Federated Learning at Scale: System Design," *arXiv:1902.01046 [cs, stat]*, mar. de 2019, arXiv: 1902.01046. dirección: http://arxiv.org/abs/1902.01046 (visitado 26-04-2021).

[7] L. Melis, C. Song, E. D. Cristofaro y V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," 2018, arXiv: 1805.04049 [cs.CR].

[8] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter y T. Ristenpart, "Stealing machine learning models via prediction APIs," 2016, arXiv: 1609.02943 [cs.CR].

[9] F. Armknecht y col., "A Guide to Fully Homomorphic Encryption," inf. téc. 1192, 2015. dirección: http://eprint.iacr.org/2015/1192 (visitado 09-05-2021).

[10] A. N. Bhagoji, S. Chakraborty, P. Mittal y S. Calo, "Analyzing federated learning through an adversarial lens," 2019, arXiv: 1811.12470 [cs.LG].

[11] *para Raspberry Pi 4 Model B,Raspberry Pi 3 B + Caja con Ventilador de refrigeración y disipador de Calor, Caja de acrílico de 4 Capas Caja apilable Cluster Caja para Raspberry Pi 3/2 Modelo B: Amazon.es: Grandes electrodomésticos*. dirección: https://www.amazon.es/Raspberry-Ventilador-refrigeraci%C3%B3n-disipador-acr%C3%ADlico/dp/B07J9VMNBL/ref=sr_1_2?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=raspberry+rack&qid=1619378660&sr=8-2 (visitado 25-04-2021).

[12] *Archivo:Raspberry Pi 3 B+ (39906369025).png*, es. dirección: https://es.wikipedia.org/wiki/Archivo:Raspberry_Pi_3_B%2B_(39906369025).png (visitado 04-05-2021).

[13] *Jetson Nano*. dirección: https://www.nvidia.com/en-il/autonomous-machines/jetson-store/ (visitado 09-04-2021).

[14] *Eco-aware Persuasive Networked Data Devices for User Engagement in Energy Efficiency — GREENSOUL Project — H2020 — CORDIS — European Commission*. dirección: https://cordis.europa.eu/project/id/696129/es (visitado 05-05-2021).

[15] D. Lockton, D. Harrison y N. A. Stanton, "Models of the user: designers' perspectives on influencing sustainable behaviour," en, *JDR*, vol. 10, n.º 1/2, pág. 7, 2012, ISSN: 1748-3050, 1569-1551. dirección: http://www.inderscience.com/link.php?id=46137 (visitado 05-05-2021).

[16] K. Bonawitz y col., "Towards Federated Learning at Scale: System Design," *arXiv:1902.01046 [cs, stat]*, mar. de 2019, arXiv: 1902.01046. dirección: http://arxiv.org/abs/1902.01046 (visitado 08-05-2021).

# Apéndice A

# Protección de Datos en la Unión Europea y España

## A.1. Preludio

Ante la llegada de las Tecnologías de la Información y la Comunicación (TIC), la Organización para la Cooperación y el Desarrollo Económicos (OCDE) decidió definir unas directrices el 23 de septiembre de 1980 que regulen *".. la protección de la privacidad y el flujo transfronterizo de datos personales ... "* [1, págs.1]. Este fue uno de los primeros intentos de regular la gestión y utilización de los datos personales.

No llegaron medidas concretas a Europa hasta el 24 de octubre de 1995, cuando el Parlamento Europeo y el Consejo Europeo publicaron la *"Directiva 95/46/CE"* [2] con el objetivo de crear un marco jurídico que garantizase la protección de los datos y la libre circulación de estos[1]. Por eso, en España se creó la *"Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal"* [3] más conocida por sus siglas LOPD, que era la encargada de garantizar el cumplimiento de la directiva europea en España.

Ante la ausencia de leyes que garantizaran el cumplimiento de los derechos humanos en internet, algunos autores empezaron a definir los derechos digitales. Una de las primeras declaraciones de derechos fundamentales tecnológicos fue la de Robert B. Gelman, que en 1997 difundió una propuesta de *"Declaración de los Derechos Humanos en el Ciberespacio"*, basada en la Declaración Universal de los Derechos Humanos de 1948.

[1] Las directivas de la UE no son legalmente vinculantes para los ciudadanos, se dirigen a todos los estados miembros y cada uno deberá transponer la directiva a las leyes locales

Sin embargo, aunque el trabajo de Robert B. Gelman fue ampliamente apoyado por una gran comunidad de expertos, no ha sido hasta la entrada en el siglo XXI (cuando las TIC ya han conseguido una gran relevancia en la sociedad) que se ha empezado a atisbar la necesidad de derechos digitales inherentes de vivir en una sociedad tecnológica. Tanto es así, que varios autores empezaron a escribir sobre la necesidad de una nueva generación de derechos humanos que incluyese el campo de la tecnología.

La corriente pro derechos digitales llegó inclusive a la Unión Europea, que en el año 2000 redactó la *Çarta de los Derechos Fundamentales de la Unión Europea"* [4] en Niza con el objetivo de *"... reforzar la protección de los derechos fundamentales a tenor de la evolución de la sociedad, del progreso social y de los avances científicos y tecnológicos."* [4, págs.5]. El problema es que no pasó a ser jurídicamente vinculante hasta el año 2009 junto con el Tratado de Lisboa. Esta carta es especialmente innovadora en cuanto a la protección de datos de carácter personal, puesto que establecía como derecho fundamental la protección de estos en el artículo 8. En este artículo se establecía que además, estos se tratarían de modo leal y con el consentimiento de la persona afectada. Asimismo, la persona afectada tendría derecho a acceder a los datos recogidos, así como a la rectificación de los mismos.

## A.2. Legislación

La legislación vigente en Europa cambió radicalmente en el 2016, el parlamento europeo adoptó nuevas medidas para la era digital.

El 5 de mayo de 2016 entró en vigor la Directiva (UE) 2016/680 [5], reconocida como la Directiva sobre protección de datos en el ámbito penal, *"... relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales por parte de las autoridades competentes para fines de prevención, investigación, detección o enjuiciamiento de infracciones penales o de ejecución de sanciones penales, y a la libre circulación de dichos datos ..."* [5, págs.1]. La Comisión Europea define esta medida en su página web [6] como:

> *"... protege el derecho fundamental de los ciudadanos a la protección de sus datos cuando los utilicen las autoridades policiales y judiciales a efectos de aplicación de la ley. Más concretamente, la Directiva garantizará que se protejan adecuadamente los datos personales de víctimas, testigos y sospechosos de delitos, además de facilitar la cooperación transfronteriza en la lucha contra la delincuencia y el terrorismo."*

El 24 de mayo de 2016 el Reglamento (UE) 2016/679 [7], reconocido como el Reglamento General de Protección de Datos (RGPD). *"... 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 ..."* [7, págs.2]. La Comisión Europea define esta medida en su página web como:

> *"... es una medida esencial para fortalecer los derechos fundamentales de las personas en la era digital y facilitar la actividad económica, ya que aclara las normas aplicables a las empresas y los organismos públicos en el mercado único digital. Además, la existencia de una norma única pone fin a la fragmentación en distintos sistemas nacionales y a las cargas administrativas innecesarias."*

Esta legislación no sería aplicable hasta dos años más tarde, el 6 y 25 de mayo de 2018 respectivamente. Dos años durante los cuales cualquier empresa de la unión o cualquier empresa que tenga negocios en la Unión Europea se debía adaptar a la nueva normativa.

El objetivo principal de este reglamento era proteger a las personas físicas en cuanto al tratamiento de sus datos personales y a la libre circulación de estos. Estos datos son de naturaleza sensible, puesto que pueden revelar la identidad de la persona, ya sea mediante el nombre completo, la dirección de su domicilio, el número de tarjeta bancaria, etc.

Entre las cuestiones más relevantes del RGPD destacan:

- La transparencia: obliga a las organizaciones a comunicar a los usuarios el tratamiento que se realiza a sus datos.

- El derecho a la rectificación y borrado: cada usuario tendrá derecho a pedir que se rectifiquen sus datos en caso de ser inexactos. Además, los usuarios tendrán derecho a que las organizaciones borren sus datos y rescindir el consentimiento

de tratamiento de ellos.

- Derechos sobre el procesamiento: cada usuario podrá solicitar la limitación de procesamiento de sus datos, así como negarse a que se usen en tomas de decisiones o en perfiles automatizados, aparte, también podrá solicitar que le entreguen sus datos en formato estructurado cuando sea posible.

La directiva sobre protección de datos en el ámbito penal y el RGPD dejaron obsoleta la LOPD española, por lo cual, para adaptarse al nuevo marco jurídico europeo el 7 de diciembre de 2018 entró en vigor la *"Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y Garantía de los Derechos Digitales"* [8] (LOPD-GDD), acorde con el RGPD.

Esta ley transpone el RGPD europeo a la legislación española, incluyendo medidas como:

- La incorporación del término privacidad desde el diseño, lo que quiere decir que deben elaborarse procesos empresariales teniendo en cuenta la LOPD-GDD desde un primer momento.
- El consentimiento, donde se obliga a las organizaciones a obtener el consentimiento explícito para el tratamiento de datos de las personas, estas personas, además, podrán solicitar la portabilidad de sus datos o la eliminación de los mismos.

Sin embargo, la LOPD-GDD no se ciñe únicamente a adaptar el RGPD a la legislación Española, fue la primera ley europea de protección de datos en incluir explícitamente los derechos digitales de las personas en los entornos digitales.

## A.3. Derechos fundamentales

Como se ha comentado en el apartado anterior, con la entrada en vigor de la LOPD-GDD España se convirtió en el primer país europeo en garantizar los derechos digitales. Estos derechos digitales llegaron en 2018 en el Título X de la LOPD-GDD, titulado *"Garantía de los derechos digitales"*, en respuesta a la Carta de los Derechos Fundamentales de la Unión Europea y a las nuevas medidas del Reglamento (UE) 2016/679 y de la Directiva (UE) 2016/680.

Varios de los derechos recogidos en la LOPD-GDD fueron:

- El derecho a la intimidad y uso de dispositivos digitales en el ámbito laboral.

- El derecho a la desconexión digital en el ámbito laboral.
- Derecho al olvido en búsquedas de Internet.
- El derecho al testamento digital.

Este paso adelante por parte del estado español incentivó a que un año después de la entrada en vigor de la LOPD-GDD, el Ilustre Colegio de la Abogacía de Barcelona (ICAB) presentase la *Çarta de Barcelona por los Derechos de la Ciudadanía en la Era Digital"* [9], apoyada por universidades y entidades de la sociedad civil.

A la vista de la relevancia e impacto del tema, el gobierno español (representado por el Ministerio de Asuntos Económicos y Transformación Digital), se inspirará en la carta del ICAB para redactar la pionera *Çarta de Derechos Digitales"* [10] de España, que contribuirá con los objetivos ya avanzados en el Título X de la LOPD-GDD. Se prevé que sea aprobada antes de 2022 para la estrategia España Digital 2025.

Esta carta cuenta con 25 puntos de alta relevancia entre los que están los recogidos en el anteriormente mencionado Título X de la LOPD-GDD, y entre otros, algunos de especial impacto en el desarrollo de este proyecto: los derechos ante la Inteligencia Artificial (Derecho XXIII) y el derecho a no ser localizado y perfilado (Derecho V).

El derecho a no ser localizado y perfilado deja claro que cada persona tiene derecho a la *"... libre autodeterminación individual (...) a no ser objeto de localización, ni a ser sometido a análisis de la personalidad o conducta que impliquen el perfilado de la persona"*, además, añade que sólo se podrán realizar *"... tratamientos de información personal con el consentimiento de la persona afectada ..."*.

En cuanto al derecho ante la Inteligencia Artificial expone como primer punto que, en lo que concierne al desarrollo y ciclo de vida de esta, *"Se deberá garantizar el derecho a la no discriminación algorítmica . . . "*. Además. en procesos de decisión automatizada las personas tienen derecho a *"Solicitar una supervisión e intervención humana"* o *Ïmpugnar las decisiones automatizadas o algorítmicas"*.

La carta, al no estar aprobada aún, no es legalmente vinculante, por lo que no es de obligado cumplimiento. Sin embargo, deja clara la orientación que van a tomar los derechos digitales en los próximos años.

# Referencias


[1] OECD, *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*, es. OECD, feb. de 2002, ISBN: 978-92-64-19719-0 978-92-64-19639-1. dirección: https://www.oecd-ilibrary.org/science-and-technology/oecd-guidelines-on-the-protection-of-privacy-and-transborder-flows-of-personal-data_9789264196391-en (visitado 28-03-2021).

[2] *DIRECTIVA 95/46/CE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 24 de octubre de 1995*. dirección: https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:31995L0046&from=ES (visitado 28-03-2021).

[3] *LEY ORGÁNICA 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal*. dirección: https://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf (visitado 28-03-2021).

[4] "Carta de los Derechos Fundamentales de la Unión Europea," pág. 17,

[5] "DIRECTIVA (UE) 2016/ 680 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016," es, pág. 43,

[6] *La protección de datos en la UE*, es, Text. dirección: https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en (visitado 27-03-2021).

[7] *REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016*. dirección: https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:02016R0679-20160504&from=ES (visitado 27-03-2021).

[8] M. B. Samper, *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*, es, 1.ª ed. Dykinson, sep. de 2020, ISBN: 978-84-13-77085-7 978-84-13-77029-1. dirección: http://www.jstor.org/stable/10.2307/j.ctv17hm980 (visitado 28-03-2021).

[9] *Carta de Barcelona por los Derechos de la Ciudadanía en la Era Digital*. dirección: https://www.icab.cat/export/sites/icab/.galleries/documents-noticies/2019/documento-carta-de-barcelona-por-los-derechos-de-la-ciudadania-en-la-era-digital-21-02-2019.pdf (visitado 01-04-2021).

[10] *Carta de Derechos Digitales*. dirección: https://portal.mineco.gob.es/RecursosArticulo/mineco/ministerio/participacion_publica/audiencia/ficheros/SEDIACartaDerechosDigitales.pdf (visitado 30-03-2021).

# Apéndice B

# Documentos adicionales

En este apartado se adjuntan los documentos que debido a su tamaño puede que no se puedan observar con claridad en la memoria del proyecto.

## B.1. Planificación del Proyecto

En la siguiente hoja se adjunta el diagrama GANTT de la planificación del proyecto a tamaño completo.

| EDT | TÍTULO DE LA TAREA | FASE UNO | | | | | | | | | | | | | | | FASE DOS | | | | | | | | | | | | | | | | | FASE TRES | | | | | | | | | | | | | | | FASE 4 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SEMANA 1 | | | | | SEMANA 2 | | | | | SEMANA 3 | | | | | SEMANA 4 | | | | | SEMANA 5 | | | | | SEMANA 6 | | | | | SEMANA 7 | | | | | SEMANA 8 | | | | | SEMANA 9 | | | | | SEMANA 10 | | | | | SEMANA 11 | | | | | SEMANA 12 | | | | |
| | | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V | L | M | X | J | V |
| 0 | Registrar proyecto | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0.1 | Definición de título y descripción | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0.2 | Registro del proyecto a través del formulario | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Formación | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1.1 | Formación sobre conceptos de Machine Learning | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1.2 | Fornación sobre conceptos de los sistemas de recomendación | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1.3 | Formación sobre conceptos de Federated Learning | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1.4 | Formación sobre LightFM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1.5 | Familiarización con el sistema de recomendación de R.Sánchez | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Configuración del hardware | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.1 | Configuración de las RaspBerrys | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.2 | Configuración de la JetsonNano | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.3 | Aprovisionamiento de los Sistemas Operativos | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.4 | Instalar y ejecutar sistema de recomendaciones | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Adaptación del sistema de recomendación online a aprendizaje federado | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.1 | Guardado , cargado y envío de modelos de inteligencia artificial | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.2 | Agregación de los modelos de inteligencia artificial | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.3 | Recomendación para un único usuario | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.4 | Combinar recomendaciones para un único usuario | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.5 | Combinar recomendaciones para varios usuario | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.6 | Reentrenar modelos | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4.7 | Visualización de los resultados de rendimiento por gráficas | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Desarrollo de la memoria | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.1 | Resumen | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.2 | Introducción | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.3 | Antecedentes y Justificación | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.4 | Objetivos y Alcance | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.5 | Consideraciones éticas | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.6 | Metodología | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.7 | Planificación | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.8 | Introducción al Federated Learning | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.9 | Desarrollo | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.10 | Presupuesto | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.11 | Conclusiones y Trabajo futuro | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.12 | Bibliografía | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.13 | Definiciones y Acrónimos | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3.14 | Anexos | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Estudio del rendimiento | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5.1 | Estudio del rendimiento de un sistema de recomendaciones centralizado | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5.2 | Estudio del rendimiento de un sistema de recomendaciones de modelos agregados | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5.3 | Estudio de rendimiento del sistemas de recomendaciones final | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |