



## GRADO EN BUSINESS ANALYTICS

TRABAJO FIN DE GRADO

### **Viabilidad de negocio de un servicio de “Machine Learning as a Service”.**

Autor: Ignacio Divassón González

Director: Carlos Bellón Núñez-Mera

Madrid



# **VIABILIDAD DE NEGOCIO DE UN SERVICIO DE “MACHINE LEARNING AS A SERVICE”.**

**Autor: Divassón González, Ignacio.**

Director: Bellón Núñez-Mera, Carlos.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

## **RESUMEN EJECUTIVO**

Este trabajo se enfoca en el desarrollo de una plataforma que facilita a las Pequeñas y Medianas Empresas (PYMES) el acceso a las capacidades predictivas de los modelos de Machine Learning (ML). Se ha modificado una aplicación web, previamente creada por el autor, para ofrecer a las PYMES una solución que mejora su rendimiento y aumenta su facturación, al permitirles utilizar técnicas predictivas de ML de forma intuitiva y guiada.

El proyecto asegura el cumplimiento del Reglamento General de Protección de Datos (GDPR) paquetizando la aplicación para que se ejecute en las computadoras de los clientes, evitando la cesión de datos a otras entidades. Adicionalmente, se desarrolló una aplicación web para facturar el uso de la plataforma sin recoger ningún dato personal, junto con modificaciones en la aplicación paquetizada para garantizar el total cumplimiento del GDPR.

Se llevó a cabo una valoración por usuario para confirmar la viabilidad económica del proyecto, proyectando un valor de empresa de 18 a 34.5 millones de euros, en función del grado de penetración en el mercado PYME en una década. Para apoyar esta meta, se propusieron acciones de marketing, incluyendo posicionamiento SEO, participación en ferias de PYMES, obtención de subvenciones para la digitalización y el potenciar el boca-oreja mediante descuentos.

En conclusión, la plataforma ofrece una solución rentable y legalmente sólida para ayudar a las PYMES a utilizar ML para extraer conocimiento de sus datos, mejorando su rendimiento de forma significativa.

**Palabras clave:** Machine Learning as a Service, WebApp, Emprendimiento, PYMES, Digitalización.

# **BUSINESS FEASIBILITY OF A MACHINE LEARNING AS A SERVICE.**

**Author: Divassón González, Ignacio.**

Director: Bellón Núñez-Mera, Carlos.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

## **ABSTRACT**

This work focuses on the development of a platform that facilitates Small and Medium Enterprises (SMEs) access to the predictive capabilities of Machine Learning (ML) models. A web application, previously created by the author, has been modified to offer SMEs a solution that improves their performance and increases their turnover by allowing them to use predictive ML techniques in an intuitive and guided way.

The project ensures compliance with the General Data Protection Regulation (GDPR) by packaging the application to run on clients' computers, avoiding the transfer of data to other entities. Additionally, a web application was developed to bill the use of the platform without collecting any personal data, along with modifications to the packaged application to ensure full GDPR compliance.

A per-user valuation was conducted to confirm the economic viability of the project, projecting an enterprise value of €18 to €34.5 million, depending on the degree of penetration in the SME market in a decade. To support this goal, marketing actions were proposed, including SEO positioning, participation in SME fairs, obtaining subsidies for digitalization and leveraging word-of-mouth through discounts.

In conclusion, the platform offers a profitable and legally sound solution to help SMEs use ML to extract knowledge from their data, significantly improving their performance.

**Keywords:** Machine Learning as a Service, WebApp, Entrepreneurship, SMEs, Digitization

## Índice de la memoria

Capítulo 1. Introducción .....	6
1.1 Contextualización.....	6
1.2 Justificación.....	8
1.3 Objetivo del Trabajo.....	9
1.4 Metodología.....	10
Capítulo 2. Digitalización de las PYMES.....	11
2.1 Las PYMES en España.....	11
2.2 Estado actual de la digitalización de las PYMES.....	11
2.3 Necesidad del Machine Learning en las PYMES.....	15
2.4 Incentivos a las PYMES en favor de la digitalización .....	16
Capítulo 3. Estado actual de la Cuestión.....	19
3.1 Cambio de paradigma de productos a servicios .....	19
3.2 El Machine Learning as a Service .....	20
3.2.1 Situación del mercado .....	20
3.2.2 Principales competidores .....	21
3.2.3 Conclusión.....	28
Capítulo 4. Idea de Negocio.....	31
4.1 Idea de Negocio de una Plataforma de Machine Learning as a Service.....	31
4.1.1 Proyecto de Machine Learning as a Service existente.....	32
4.1.2 Idea de Negocio.....	36
4.2 Barreras .....	37
4.2.1 Regulación.....	37
4.2.2 Entrada al mercado.....	37
Capítulo 5. Regulación.....	38
5.1 Ley Europea de Protección de Datos (GDPR) .....	39
5.2 Inconvenientes de la GDPR con la plataforma bajo estudio .....	42
5.3 Propuesta tecnológica para cumplir con GDPR.....	43
5.4 Implicaciones de este cambio.....	43
5.5 Cambios tecnológicos en el proyecto.....	44

5.5.1 Creación de la nueva aplicación de control.....	45
5.5.2 Garantizar la conectividad con la nueva aplicación de control .....	50
5.5.3 Informar a nuestros clientes de la necesidad de tener consentimiento para aplicar algoritmos predictivos.....	62
5.5.4 Garantizar el borrado de clientes y la descarga de su información para cumplir con la ley .....	63
5.5.5 Paquetización de la aplicación.....	67
Capítulo 6. Modelo de Negocio .....	70
6.1 Cambio de modelo de negocio .....	70
6.2 Factores diferenciadores.....	70
6.3 Acciones de marketing .....	72
6.4 Valoración .....	73
6.4.1 Precio de la plataforma y Tamaño de mercado .....	73
6.4.2 Valoración de la empresa por usuario .....	75
Capítulo 7. Conclusión.....	79
Capítulo 8. Bibliografía.....	81

## *Índice de figuras*

Figura 1. Usuarios en internet que han comprado bienes o servicios para uso privado en los últimos 12 meses .....	7
Figura 2. Interés a lo largo del tiempo de la Inteligencia Artificial en España (2009-2023) Fuente: Google Trends ( <a href="https://trends.google.es/trends/explore?date=all&amp;geo=ES&amp;q=%2Fm%2F0mkz&amp;hl=es">https://trends.google.es/trends/explore?date=all&amp;geo=ES&amp;q=%2Fm%2F0mkz&amp;hl=es</a> ). 14	
Figura 3. Tamaño de mercado “-aaS” en miles de millones de dólares 2021-2029F Fuente: Fortune Business Insights, (2022) .....	19
Figura 4. Tamaño de mercado MLaaS en España en miles de millones de dólares 2021-29F Fuente: Data Bridge Market Research,(2022).....	21
Figura 5. Fichero de trabajo para interactuar con la plataforma SageMaker. ....	23
Figura 6. Forma de interactuar con la plataforma VertexAI. ....	25
Figura 7. Forma de interactuar con la plataforma IBM Watson Studio. ....	26
Figura 8. Forma de interactuar con la plataforma Microsoft Azure.....	27
Figura 10. Estructura de carpetas del proyecto.....	47
Figura 11. Funcionamiento propuesto para la aplicación.....	51
Figura 12. Ejemplo de mal funcionamiento del flujo propuesto para la aplicación.....	51
Figura 13. Funcionamiento del inicio de sesión de los usuarios .....	53
Figura 14. Funcionamiento del proceso de uso de la plataforma y su cobro por ello .....	54

Figura 15. Imagen de la nueva forma de crear proyectos, incluyendo el checkbox para comprobar el compliance con GDPR y el link a dicha ley..... 63

## *Índice de tablas*

Tabla 1. Comparación de los principales competidores identificados .....	29
Tabla 2. Valor por usuario de la empresa Mach IA.....	76
Tabla 3. Valoración de la empresa Mach IA, llegando a un 6.5% del mercado .....	77
Tabla 4. Valoración de la empresa Mach IA, llegando a un 13.3% del mercado .....	78

## Capítulo 1. INTRODUCCIÓN

### *1.1 CONTEXTUALIZACIÓN*

Estamos actualmente en un tiempo en el que abundan los datos. Los creamos constantemente. Se estima que cada día segundo generamos 1.7 MB (Domo, 2018), o lo que es lo mismo, cada día generamos más de 146 GB de datos con nuestras búsquedas en internet, compras, likes, fotos, videos y llamadas. Además, una parte importante de los datos que generamos pueden ser fácilmente recogidos por empresas, como por ejemplo pueden ser los leads digitales que pueda tener una empresa, o las ventas que tenga, ya sean digitales o presenciales.

Según Eurostat (2023), un 75% de todos los usuarios europeos con más de 16 años, ha efectuado compras en internet en el último año. Además, esta tendencia no parece que vaya a dejar de subir, pues, como apunta el mismo estudio, el número de usuarios totales que ha efectuado compras online en el último año ha crecido a un 3.0% CAGR 2010-22, y esta tendencia no se está frenando, pues se mantiene en el 3.0% en los últimos 5 años, CAGR 2017-22.

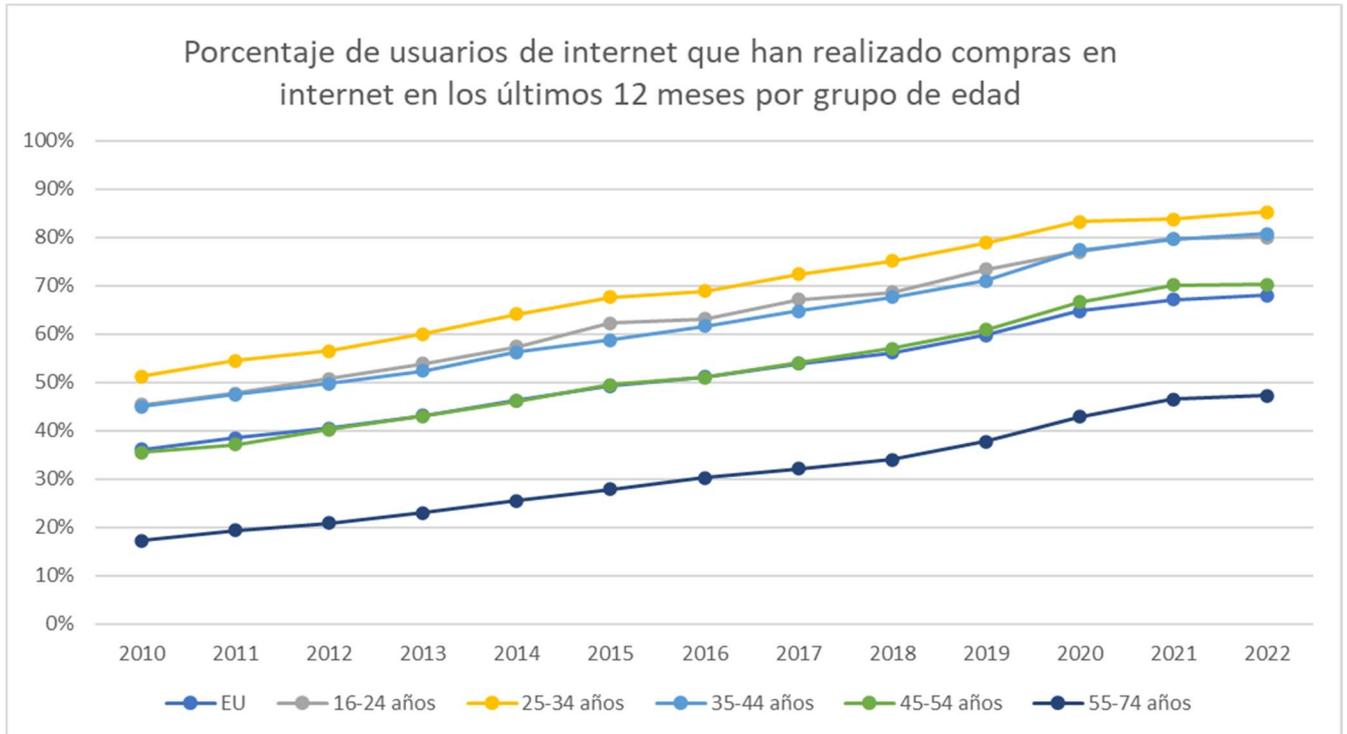


Figura 1. Usuarios en internet que han comprado bienes o servicios para uso privado en los últimos 12 meses

Fuente: Eurostat (códigos de datos: isoc\_ec\_ibuy and isoc\_ec\_ib20)

Pero de nada sirve esta gran cantidad de datos generados si no se extrae valor o conocimiento de ellos. De lo contrario su almacenamiento o recopilación resultaría completamente inútil.

Obtener valor de un conjunto de datos significa que su posesión agregue un valor añadido diferencial o una información extra a quien lo posee. Este valor diferencial puede provenir de comprender las causas de un incidente, obtener información sobre el comportamiento de un cliente, o predecir si un lead se puede convertir en cliente.

Hay infinitud de formas de obtener conocimiento a partir de un conjunto de datos, aunque sin duda una de las más potentes es el Machine Learning. El Machine Learning es una rama de la Inteligencia Artificial, en la que los algoritmos o modelos permiten a los ordenadores aprender sobre datos pasados con el fin de extraer conocimiento y poder describir o hacer predicciones a futuro (Alpaydin, 2020). Los algoritmos de Machine Learning pueden

dividirse en aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi-supervisado y entrenamiento con refuerzo (Chinnamgari, 2019).

Hasta ahora se ha avanzado mucho en la creación de muchos algoritmos que consigan predecir y tratar grandes cantidades de datos, en lo que se ha llamado la fase Big Data 1.0. Pero después de crear esos algoritmos y esa facilidad para tratar datos, viene la era del Big Data 2.0, en la que las empresas puedan utilizar algoritmos de machine learning para obtener rentabilidad y sacar valor a sus datos (Provost & Fawcett, 2013).

## ***1.2 JUSTIFICACIÓN***

Para implementar estos algoritmos las empresas requieren de *data scientists*, o científicos de datos, para que analicen y visualicen datos con el objetivo construir e implementar estos algoritmos de machine learning con la intención de extraer conocimiento que aporte valor (Pantoja, 2023). Afrontar estos costes es difícil para algunas empresas que, aunque tienen datos, no tienen los recursos financieros para contratar a un científico de datos, o la frecuencia de extracción de datos de sus datos es lo suficientemente baja como para pagar estos costos recurrentes. Además, según el informe "Uso de tecnologías digitales por empresas en España" del Observatorio Nacional de Tecnología y Sociedad (2022), las empresas que no emplean técnicas de inteligencia artificial lo hacen principalmente debido a la falta de conocimientos en el procesamiento de datos. Es importante destacar que esta razón es mucho más frecuente (más de cuatro veces) que la percepción de que dichas técnicas no resulten útiles para la empresa.

Aunque en los últimos años se estén dando pasos hacia soluciones más sencillas que no necesiten que el usuario tenga conocimientos de programación, como soluciones No-Code que cubriremos más adelante, éstas aún requieren que el usuario que quiera sacar partido a sus datos tenga conocimientos sobre el preprocesamiento de datos, la visualización de estos, o sepa qué modelos o qué hiperparámetros de los mismos utilizar con la intención de entrenar y generar los modelos.

### ***1.3 OBJETIVO DEL TRABAJO***

El objetivo del trabajo es comprobar la viabilidad de comercializar un servicio que ayude a las empresas a entrenar modelos predictivos de una manera fácil e intuitiva, ayudándoles en todos los pasos a la hora de generar modelos de machine learning. Este servicio de Machine Learning as a Service (MLaaS) será llamado Mach IA.

Este servicio de MLaaS permitiría a las empresas sin grandes recursos internos a acceder a capacidades de análisis predictivo de una forma fácil e intuitiva. Dentro de los servicios que se ofrecerían en este MLaaS están herramientas para el preprocesamiento de datos, el entrenamiento de modelos de predicción (clasificación y regresión), evaluación de los modelos entrenados, y su uso para predecir nuevos datos.

Desde el punto de vista de las PYMES, su adopción tiene especial sentido por varias razones. La primera de ellas es el ahorro de costes en comparación con el desarrollo de tecnologías similares en la propia empresa, que requiere grandes recursos tanto financieros como de personal; en segundo lugar, el MLaaS ofrece la posibilidad de centrarse en los conocimientos derivados de los datos, ya que elimina las complejidades de gestión de la infraestructura típicamente asociadas con la retención de sofisticados sistemas de ML; y, por último, la escalabilidad es una característica distintiva de esta tecnología que permite a las empresas crecer fácilmente, sin dificultar su proceso de procesamiento de datos, manteniendo de una forma simple y sencilla la toma de decisiones basada en datos.

Además de comprobar la viabilidad de comercializar este servicio propuesto, se estudiará como introducirlo legalmente en el mercado, cumpliendo todas las regulaciones para su correcta comercialización.

El último objetivo de este trabajo será estudiar la forma de comercializar la idea propuesta, además de una descripción a alto nivel sobre la empresa, que incluirán precio, costes previstos y una valoración de la empresa.

## ***1.4 METODOLOGÍA***

Con el objetivo de cumplir los objetivos del trabajo descritos anteriormente, dividiré el trabajo en las siguientes secciones.

En primer lugar, se realizará un estudio del mercado objetivo, las pequeñas y medianas empresas en España; en segundo lugar, se llevará a cabo un detallado estudio de los competidores actuales, centrándose tanto en su potencia, su precio y su sencillez; en tercer lugar se describirá la idea de negocio de Machine Learning as a Service, y su importancia para el mercado objetivo; en cuarto lugar se estudiará como adaptar la idea de negocio propuesta en un negocio comercializable para cumplir con las normativas legales; en quinto lugar se elaborarán acciones de marketing para la empresa propuesta; y en sexto lugar se describirá la empresa a alto nivel, detallando los números relacionados con la misma, como precios y costes, junto con una valoración de la empresa por usuario.

## **Capítulo 2. DIGITALIZACIÓN DE LAS PYMES**

Como se va a explicar en los siguientes epígrafes se va a analizar la importancia de las PYMES en la economía, su digitalización y posibilidad de aumentarla, teniendo también en cuenta los incentivos gubernamentales que podrían favorecer a esta idea de empresa.

### **2.1 LAS PYMES EN ESPAÑA**

Como se ha mencionado anteriormente, el público objetivo de este negocio son las PYMES. Las pequeñas y medianas empresas representan una gran parte del tejido productivo de nuestro país. Según la Unión Europea (REGLAMENTO (UE) no 651/2014), la categoría de PYMES está formada por microempresas, pequeñas y medianas empresas. En España, según datos del INE (2022), las PYMES representan el 43% del tejido empresarial, mientras que las grandes empresas, aquellas que tengan más de 250 empleados o facturación anual superior a 50 millones de euros o cuyo balance general anual supere los 43 millones de euros, según el reglamento anterior, solo representan el 0.14%. El 57% restante de las empresas constituidas en España lo forman empresas sin trabajadores.

Además, la unión de las PYMES junto con las empresas sin trabajadores, que juntas representan el 99.9% de todas las empresas, supone el 65% del PIB de nuestro país, y generan el 75% de los puestos de trabajo (INE, 2022). Por lo tanto, el público objetivo de este negocio es bastante amplio, lo que aumenta la viabilidad de un negocio destinado a este tipo de empresas.

### **2.2 ESTADO ACTUAL DE LA DIGITALIZACIÓN DE LAS PYMES**

Teniendo en cuenta la importancia de las PYMES en nuestro país, se va a estudiar su digitalización, y posteriormente su uso de estas técnicas de aprendizaje supervisado.

Para implementar estas técnicas de análisis de datos y de machine learning puede hacerse mediante la creación de los entornos de trabajo desarrollados en la propia empresa y programados en los ordenadores de esta, o utilizando servicios ubicados en servidores en la nube de proveedores de estos servicios. No obstante, la segunda opción es entendible que sea la más atractiva para las pequeñas y medianas empresas en España pues no tienen que desarrollar la infraestructura subyacente, necesaria para generar modelos de machine learning, ellos mismos con equipos dedicados a ellos, sino que pueden contratar a empresas que los ofrezcan en internet.

No obstante, en España las PYMES tienen las herramientas básicas para acceder a estos servicios que implementan machine learning y de la inteligencia artificial. Más del 85% de todas las empresas con menos de 250 empleados disponen de ordenadores en el trabajo para poder trabajar con ellos (Observatorio Nacional de Tecnología y Sociedad, 2022). Pero, aunque los ordenadores permitan realizar e implementar las técnicas del análisis de datos y machine learning debido a su potencia de cómputo, muchas veces para su correcta implementación es necesario apoyarse en técnicas, librerías y herramientas que se encuentran en internet. Además, existe el caso en el que las empresas puedan delegar esta potencia de cómputo en servidores en la nube utilizando herramientas online que intentan simplificar estas tareas.

Pero eso no es tampoco ningún impedimento para las PYMES. En España el 82% de estas empresas con menos de 250 empleados disponen de conexión a internet (Observatorio Nacional de Tecnología y Sociedad, 2022). Según el “Informe de cobertura de banda ancha” (Secretaría de Estado de Telecomunicaciones e Infraestructuras Digitales, 2022), nuestro país actualmente es el tercer país de Europa con mejor acceso y conectividad a internet, con una cobertura de 30 Mbps del 95%, lastrada eso sí por el medio rural, donde ese porcentaje cae al 65%; y con una cobertura del 87% a la asombrosa velocidad de 100Mbps. No obstante, el plan de Universalización de Infraestructuras para la Cohesión (UNICO), pretende mejorar estas asombrosas cifras llegando a alcanzar un 100% de cobertura para toda la población con una banda ancha de 100 Mbps en el año 2025 (Ministerio de Asuntos Económicos y Transformación Social, 2022).

Queda claro que existen bases suficientes para afirmar que las empresas disponen de los requisitos básicos para ejercitar el análisis de datos y el machine learning.

Pero ¿por qué las empresas querrían utilizar el análisis de estos datos y estos algoritmos en primer lugar? La respuesta obvia es que los datos de los que disponen las empresas son cruciales para entender el todo lo que rodea a las compañías, desde su relación con sus proveedores, hasta la relación con sus clientes, pasando por la forma en la que opera y crea valor la empresa.

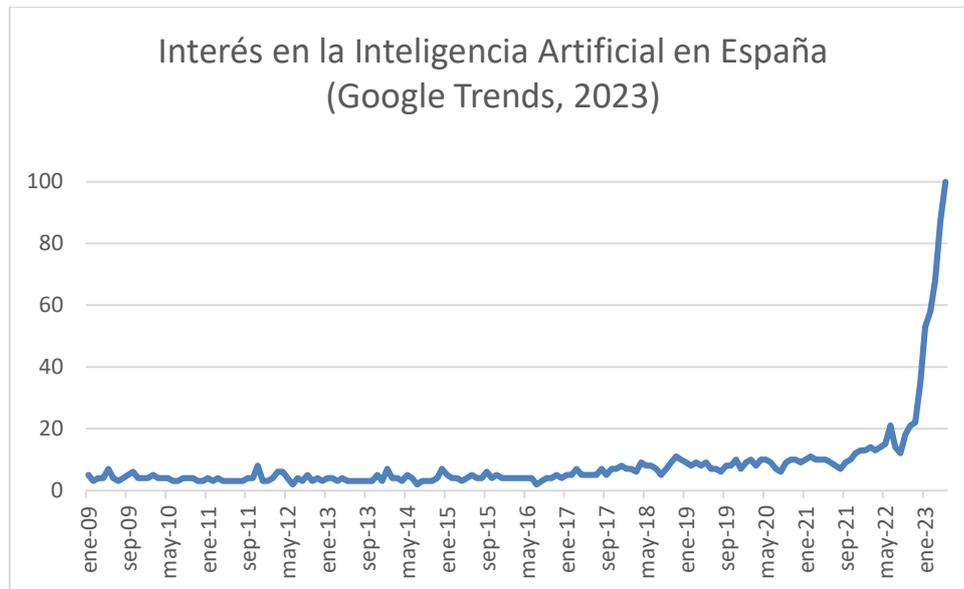
Pese a que el 94% de las empresas españolas considera que los datos son una parte crucial de su estrategia como empresa (Penteo, 2022), solo el 7% y 13% de las empresas entre 10-49 y entre 50-249 trabajadores respectivamente utilizan tecnologías de IA (Observatorio Nacional de Tecnología y Sociedad, 2022). Este dato contrasta con el 33% de las grandes empresas, pasados los 250 trabajadores.

Aunque una clara razón de esta disparidad sea la diferente cantidad de los datos que manejan las grandes empresas comparado con las pequeñas y medianas empresas al tener las grandes empresas más base de clientes finales que las PYMES, esta diferencia también es debida la diferencia de capacidades técnicas en las pequeñas y medianas empresas comparado con las grandes empresas.

Según el informe *Uso de tecnologías digitales por empresas en España* elaborado por el Observatorio Nacional de Tecnología y Sociedad (2022), los mayores impedimentos para no utilizar estas técnicas mencionadas anteriormente son sobre todo la falta de conocimientos técnicos sobre la materia, y los altos costes de llevarlo a cabo. En cambio, la última razón para no implementar soluciones de inteligencia artificial es que estas técnicas no son útiles para la propia empresa.

Como se cita en el estudio de ONTSI mencionado anteriormente, los conocimientos sobre el machine learning (área dentro de la Inteligencia Artificial), son el mayor impedimento para su expansión en el conjunto del tejido productivo español. Esto es debido puesto que su desarrollo es relativamente reciente, y, sobre todo, el interés público en ello es más reciente

aún. El interés en España por estas técnicas, según Google Trends, ha crecido exponencialmente desde mediados de 2022.



*Figura 2. Interés a lo largo del tiempo de la Inteligencia Artificial en España (2009-2023)*

*Fuente: Google Trends*

*(<https://trends.google.es/trends/explore?date=all&geo=ES&q=%2Fm%2F0mkz&hl=es>)*

Para solventar esta falta de conocimiento sobre las técnicas de inteligencia artificial del público en general, la solución más obvia sería la contratación de personal cualificado en estas técnicas. Pero debido al factor diferenciador que aporta a las empresas el sacar partido a los datos, hay actualmente una escasez de científicos de datos (Janssen, 2022). Esta escasez está provocada por la creciente demanda de estos puestos de trabajo, que se han multiplicado por siete en España entre los años 2015-2018 (Margalló et al., 2018).

Como consecuencia de este tensionamiento del mercado, y el crecimiento de la demanda de estos puestos de trabajo, han hecho que el perfil de científico de datos esté entre los perfiles con mejores salarios, concretamente en el decimoséptimo puesto, con un salario promedio por encima de los 41.000€, según el informe “Estado del mercado laboral en España” elaborado por InfoJobs y ESADE (Pérez Callejo et al., 2023). Estos altos salarios dificultan

que las pequeñas y medianas empresas, así como muchos departamentos de las grandes empresas españolas, puedan contratar a estos perfiles para sacar partido a sus datos.

Pese a que la digitalización en España está muy por encima de la media de los países de la Unión Europea, quedando en séptimo lugar en el club de los veintisiete, según el Digital Economy and Society Index (DESI) (European Commission, 2022), en lo que respecta a las técnicas que pretende proveer el servicio propuesto de Machine Learning as a Service, están bastante atrasadas viendo su avance en esas otras materias como el uso de internet o el comercio electrónico.

### ***2.3 NECESIDAD DEL MACHINE LEARNING EN LAS PYMES***

No obstante, el análisis masivo de los datos con tecnologías como la Inteligencia Artificial o el Big Data pueden ayudar enormemente a las pequeñas y medianas empresas en muchos aspectos como, por ejemplo: identificar el *buyer persona* de sus clientes, obtener una visión estratégica del negocio, segmentar clientes, diseñar campañas personalizadas para tipos de clientes, o simplemente analizar comportamientos de clientes (Blanco, 2020).

Además, existen estudios que demuestran que implementar estas tecnologías beneficia a las empresas, tanto en la información que se extrae para tomar decisiones, como para mejorar el tratamiento a los clientes. Un ejemplo es el estudio de Hansen & Bøgh (2021), que da varios ejemplos en los que se evidencia que el machine learning y la inteligencia artificial benefician a las pequeñas y medianas empresas, sobre todo si se utilizan soluciones basadas en la nube, como Microsoft Azure, puesto que este mismo estudio identifica la falta de conocimiento IT como el mayor problema que tienen estas empresas para aprovechar esta tecnología.

Otro ejemplo es el estudio de Li et al. (2023), que demuestra que el uso de algoritmos de Inteligencia Artificial (AI) a la hora de tratar la relación con los clientes digitales de las empresas mejora el rendimiento de herramientas de CRM, propias de las empresas que venden a través del comercio electrónico.

Este aspecto es muy importante para el público objetivo del que estamos hablando, las PYMES, puesto que una importante porción de estas vende sus productos a través de internet (comercio electrónico). En España el 30.9% de las empresas con menos de 250 empleados utiliza la venta electrónica (INE, 2021). Explotando los datos obtenidos de sus plataformas de venta en internet, las empresas podrían mejorar su trato con sus clientes. Así, la inteligencia artificial o machine learning pueden ayudar a las empresas a mejorar las relaciones con sus clientes, proporcionando beneficios a las empresas que las incorporan, según el libro *The art of CRM* de Fatouretchi (2019).

También el estudio *Data-driven intelligence for SME e-business: A marketing and sales perspective* (Reinhold & Reinhold, 2014) apunta en la dirección de Eduardo Esparza, ejemplificando cómo las PYMES pueden implementar sistemas de recomendación basados en machine learning para aumentar sus ventas.

Además, en España hay más PYMES que practican la venta online que la media europea, con una diferencia porcentual de 7 puntos porcentuales, mientras que tanto España tiene el mismo porcentaje de empresas que utilizan técnicas de Inteligencia Artificial según el índice DESI (European Commission, 2022). Como se ha mencionado antes, los datos que aportan a la empresa la venta de comercio electrónico son potencialmente explotables por la inteligencia artificial para mejorar la relación con el cliente y por lo tanto las ventas. Esto evidencia una vez más la poca implementación que se hace del machine learning y de la inteligencia artificial en España.

## ***2.4 INCENTIVOS A LAS PYMES EN FAVOR DE LA DIGITALIZACIÓN***

Ante esta situación en las que las PYMES españolas están infrutilizando estas técnicas de análisis de datos y de machine learning, existen muchos incentivos y planes gubernamentales en favor de la digitalización de las empresas y su uso de estas herramientas de inteligencia artificial.

En 2020 España lanzó la Estrategia Nacional de Inteligencia Artificial (ENIA) (Secretaría de Estado de Digitalización e Inteligencia Artificial, 2022), cuyo propósito es mejorar la preparación de la industria española para aumentar su competitividad a nivel europeo e internacional. Para conseguirlo, la ENIA tiene los siguientes puntos clave entre los que se van a invertir 600 millones de euros. En primer lugar, se va a destinar más inversión en investigación y desarrollo de las tecnologías dentro de la inteligencia artificial. En segundo lugar, se va a fomentar la formación y captación de perfiles con habilidades relacionadas con la inteligencia artificial. Y, en tercer lugar, se va a garantizar que este uso de la IA sea de forma adecuada, ética y responsable.

En 2021, el gobierno de España creó el Plan de Digitalización de PYMES 2021-2025 (Gobierno de España, 2021), cuyo objetivo es acelerar la digitalización de las pequeñas y medianas empresas. Para conseguirlo, este plan prevé invertir más de 4600 millones de euros en el tiempo especificado para financiar la formación, captación en tecnologías digitales para conseguir la transformación digital de este conjunto de empresas tan importantes para nuestro país.

También en 2021, desde la presidencia del gobierno, dentro del Plan de Recuperación Transformación y Resiliencia, se inaugura el Plan Nacional de Capacidades Digitales (Gobierno de España, 2021), con el que se intenta impulsar la digitalización de la economía y de la sociedad españolas. Este plan prevé invertir 11000 millones de euros en cuatro grandes áreas de actuación, entre las que están la digitalización de las empresas, favorecer la innovación, y la formación en competencias digitales.

Además, en 2022 el gobierno lanza, dentro del plan de recuperación PERTE (Proyectos Estratégicos para la Recuperación y Transformación Económica), invitaciones a las Comunidades Autónomas para que presenten proyectos de digitalización en el llamado plan RETECH (Redes Territoriales de Especialización Tecnológica) (Ministerio de asuntos económicos y transformación digital, 2022). Entre las propuestas recibidas, está la digitalización de las empresas y el impulso a la inteligencia artificial.

Todos estos planes gubernamentales se engloban todos dentro de la agenda España Digital 2026 (Gobierno de España, 2022), que es el plan que establece los pasos y estrategias a seguir para llevar a cabo la transformación digital de España.

Además de estos planes gubernamentales, tenemos que sumarle los planes supra gubernamentales que provienen de la Unión Europea. En 2020, el Consejo Europeo acordó la creación del plan de recuperación temporal Next Generation EU. Este plan, dotado de 750.000 millones de euros para los estados miembros tiene como objetivo hacer frente a las consecuencias de la pandemia del COVID-19. Entre estas ayudas destacan: el Bono Digital PYMES, que financia la implantación de soluciones digitales a las pequeñas y medianas empresas; el programa NEOTEC, que subvenciona la puesta en marcha de nuevos proyectos tecnológicos; y el programa de Modernización del Comercio, compuesto de ayudas destinadas a impulsar la competitividad y modernización de las PYMES.

Como consecuencia de estos planes se espera que el 54% de las compañías españolas aumenten su inversión en las tecnologías alrededor de los datos y la inteligencia artificial (Penteo, 2022).

## Capítulo 3. ESTADO ACTUAL DE LA CUESTIÓN

### 3.1 CAMBIO DE PARADIGMA DE PRODUCTOS A SERVICIOS

En años recientes, se ha experimentado una transformación significativa en la que un número creciente de empresas está pasando de un modelo de negocio centrado en la venta de productos a uno enfocado en la prestación de servicios y así aumentar su competitividad (Kindström, 2010). Así, las empresas están abandonando el modelo tradicional de venta de productos y herramientas a sus clientes, en el que una vez el cliente adquiría el producto o servicio en un único pago, éste era suyo, y podía utilizarlo siempre que quería. En cambio, muchas empresas están utilizando el modelo de negocio “anything-as-a-service” (XaaS), en el que la empresa proporciona un servicio continuo al cliente donde éste paga una tarifa recurrente para seguir utilizando el servicio, en lugar de ser un producto que se compra una vez. Además, se espera que este cambio de paradigma siga creciendo a un alto ritmo mundialmente, concretamente a más de un 23% de crecimiento en tamaño de mercado anualizado según Fortune Business Insights (2022).

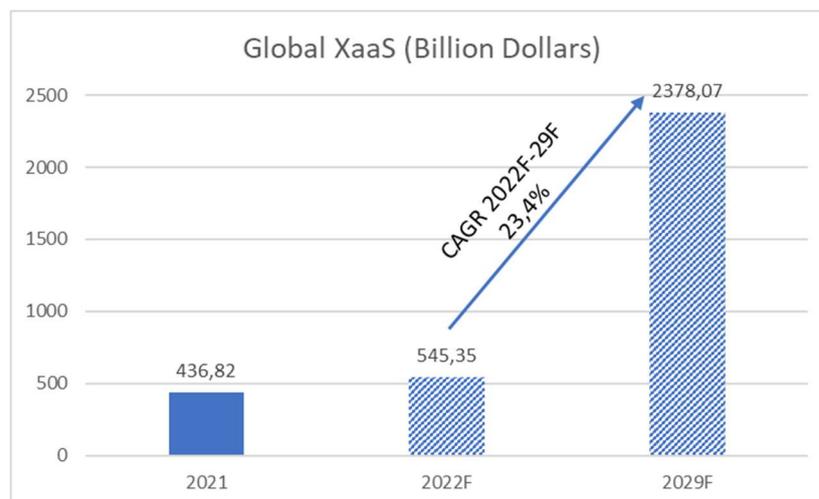


Figura 3. Tamaño de mercado “-aaS” en miles de millones de dólares 2021-2029F

Fuente: Fortune Business Insights, (2022)

Este modelo está prosperando debido a los beneficios que proporciona tanto a los proveedores como a los usuarios de estos servicios.

Los proveedores se benefician de los servicios de suscripción ya que generan ingresos recurrentes mes tras mes. Estos ingresos recurrentes son ventajosos para las empresas que proveen estos servicios (Zaki, 2019) porque: A) son relativamente estables, ya que la opción predeterminada es mantener la suscripción; B) permiten proyectar los ingresos a largo plazo; y C) ayudan a sufragar una gran parte de los costes fijos, pasando estos a ser costes variables (Lay et al., 2009), que constituyen una gran proporción de los costes en el sector de la informática, como el tiempo de computación necesario para mantener el servicio en funcionamiento. En la medida que las suscripciones a largo plazo se abonen al comienzo del periodo, esto puede además ayudar a financiar el capital circulante del proveedor de servicios.

Por otro lado, los usuarios de estos modelos de negocio orientados a servicios se benefician al delegar funciones que para ellos resultarían muy costosas, como el personal técnico o el desarrollo interno de los servicios que pueden contratar (Chimakurthi, 2021). Además, desde la perspectiva del contratante, el contrato se mantendrá mientras resulte rentable, lo que incentiva la contratación de más servicios (Ganapathy, 2020).

## ***3.2 EL MACHINE LEARNING AS A SERVICE***

### **3.2.1 SITUACIÓN DEL MERCADO**

Según un estudio de la investigadora de mercados Data Bridge Market Research (2022), se espera que el Machine Learning as a Service (MLaaS) en España crezca, de forma anualizada, más de un 40% en facturación en los años 2021-2029, como muestra la figura a continuación.



Figura 4. Tamaño de mercado MLaaS en España en miles de millones de dólares 2021-29F

Fuente: Data Bridge Market Research, (2022)

Este crecimiento indica una expectativa de intensa competencia y una demanda muy alta de estos servicios en el futuro. Por ello, es el momento oportuno para desarrollar una plataforma eficiente y potente que facilite a los usuarios, sin importar su nivel de habilidad, el acceso a estos servicios, que se prevé serán altamente solicitados en el futuro.

### 3.2.2 PRINCIPALES COMPETIDORES

Actualmente, los principales competidores del “Machine Learning as a Service” (MLaaS) son los líderes tecnológicos. Estas empresas, con grandes infraestructuras de servidores ubicadas en todo el mundo, con sus equipos de científicos e ingenieros de datos y su posición en el mercado, les permiten brindar servicios muy buenos a bajo coste.

Los principales competidores son:

- I. **Amazon Web Services (AWS):** Amazon Web Services, ofrece una gran cantidad de productos relacionados con el procesamiento de datos en la nube, y con técnicas de Machine Learning. No obstante, Amazon SageMaker

(<https://www.amazonaws.cn/en/sagemaker/>) es el que más similitudes tiene con la plataforma cuya viabilidad de negocio se va a valorar.

Esta plataforma intenta simplificar la forma en la que se programa para entrenar modelos, buscando unos buenos hiperparámetros para entrenar estos modelos predictivos, que se pueden descargar una vez se entrenan.

Una de las ventajas de esta plataforma de Amazon son los distintos modelos de machine learning que soporta. Soporta modelos desde el reconocimiento de imágenes, hasta el procesamiento de lenguaje (Natural Language Processing o NLP en inglés), pasando por el tratamiento de series temporales o modelos predictivos con datos tabulares.

Para utilizar SageMaker, es necesario programar y tener conocimientos de machine learning y análisis de datos pues la forma en la que se interactúa con esta plataforma es a través de ficheros de trabajo de programación (notebooks en inglés).

En estos ficheros de trabajo es el usuario quien tiene que elegir los modelos de aprendizaje automático que quiere entrenar con sus datos. Para realizar todos estos procesos es necesario utilizar código, aunque Amazon SageMaker tiene colecciones de ficheros de trabajo parcialmente escritos para según qué casuísticas que el usuario puede descargar de la plataforma.

Además de elegir los modelos de machine learning, el usuario tiene que elegir el error que quiere minimizar a la hora de entrenar estos modelos. Esto puede suponer un problema para los usuarios que no son avanzados en el mundo del machine learning, aunque, para los que sí que son expertos, esta personalización que ofrece la plataforma supone una ayuda muy importante.

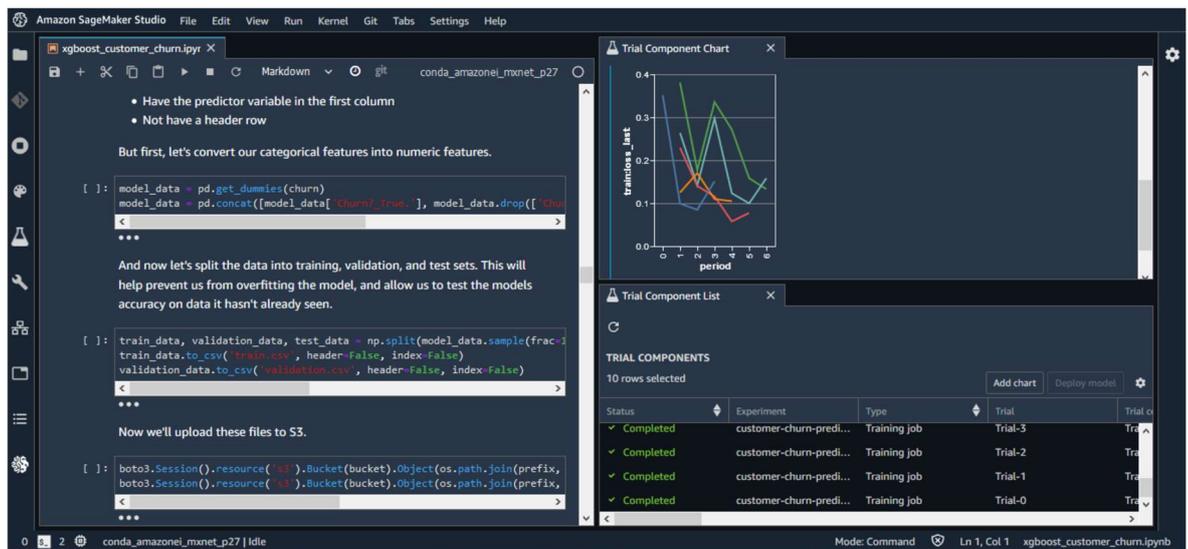
Esta plataforma espera que el usuario realice tanto el preprocesamiento de datos como la posterior visualización de los datos para entender mejor cómo se comportan sus datos (Exploratory Data Analysis, o EDA), procesos importantes y necesarios en la construcción de modelos de aprendizaje automático. Así que el usuario tiene que realizar estos pasos programando en el fichero de trabajo del proyecto.

Otra de las técnicas dentro del machine learning es la posibilidad de juntar varios modelos de aprendizaje automático entrenados para crear un modelo más complejo

y robusto, que mejora a los modelos que han sido utilizados para su creación (Valentini & Masulli, 2002). No obstante, SageMaker no permite la posibilidad de crear automáticamente estos modelos creados juntando modelos, también llamados Ensembles en inglés. En el caso de que el usuario de esta plataforma quiera mejorar los modelos ya entrenados, tendrá que juntarlos como lo haría cualquier programador fuera de la plataforma con los modelos ya entrenados.

Finalmente, SageMaker, como los otros competidores identificados, ofrece muchos niveles de coste a las empresas para adaptarse a sus necesidades.

No obstante, para valorar los costes de los competidores identificados se ha utilizado la siguiente configuración de 4 CPUs y 8 GB de RAM. Para esta configuración de SageMaker, los costes de utilizar estos ficheros de trabajo, son de 0.242 USD por hora.



*Figura 5. Fichero de trabajo para interactuar con la plataforma SageMaker.*

Fuente: <https://www.amazonaws.cn/en/sagemaker/>

- II. **Google:** De todos los servicios de computación en la nube que ofrece Google, el producto de Machine Learning más parecido a este proyecto es VertexAI (<https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform>).

Lanzado en 2022, VertexAI permite a los desarrolladores entrenar modelos de aprendizaje automático. Para ello, VertexAI utiliza una interfaz de usuario *No-Code* en la que el usuario no tiene que programar ni una línea de código. Así, los usuarios pueden arrastrar “bloques” y juntarlos como quieran para utilizar esta plataforma. Estos bloques ocultan el código al usuario, eliminando la necesidad de conocimiento de programación para usuarios inexpertos.

Pese a que eliminar la necesidad de saber programar ayuda a su uso masivo, aun así, es necesario que el usuario elija qué bloques utilizar y cómo juntarlos para poder utilizar la plataforma. Por lo que el usuario tiene que conocer qué etapas tiene la creación de modelos de machine learning, como limpieza de datos, visualización de estos datos, entrenamiento de los modelos y la búsqueda de los mejores hiperparámetros para estos modelos, y su posterior evaluación. Esto dificulta su uso por usuarios sin conocimientos de machine learning o de procesamiento de datos.

VertexAI no proporciona ninguna herramienta para visualizar los datos automáticamente (EDA), ni las formas en las que limpiar los datos. Además, se espera que sea el usuario final el que decida qué algoritmos de aprendizaje va a entrenar.

No obstante, VertexAI, a diferencia de SageMaker, facilita juntar modelos ya entrenados, Ensembles, ofreciendo la potencia de esta técnica a los usuarios inexpertos.

VertexAI tiene unos precios simplificados que disminuyen conforme se van añadiendo datos. Así, VertexAI tiene un coste de 21.25 USD por hora por cada de cada CPU para entrenar modelos, y para predecir utilizando los modelos ya entrenados, cuesta 0.2 USD cada 1.000 datos de predicción hasta 1 millón de datos. A partir de ahí, se abarata el precio, y baja a 0.1 USD por cada 1.000 datos hasta los 50 millones de datos. Finalmente, pasados los 50 millones de datos a predecir, el precio baja a 0.02 USD cada 1.000 datos.

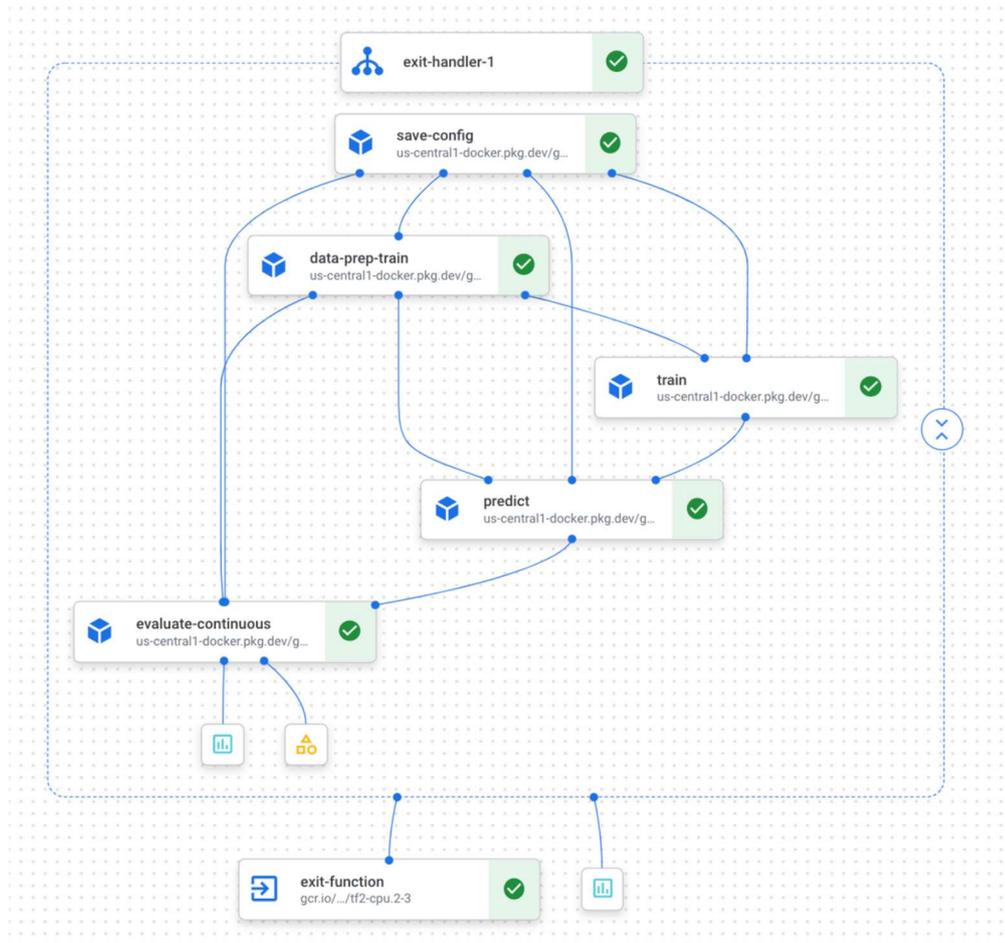


Figura 6. Forma de interactuar con la plataforma VertexAI.

Fuente: <https://www.contino.io/insights/google-vertex-ai>

III. **IBM:** IBM, a través de Watson, ofrece muchos servicios de computación en la nube, y de algoritmos de machine learning complejos como pueden ser el reconocimiento visual o el NLP (clasificación del lenguaje natural). No obstante, IBM Watson Studio es el servicio que más se parece al propuesto en este proyecto para las PYMES (<https://www.ibm.com/es-es/cloud/watson-studio>).

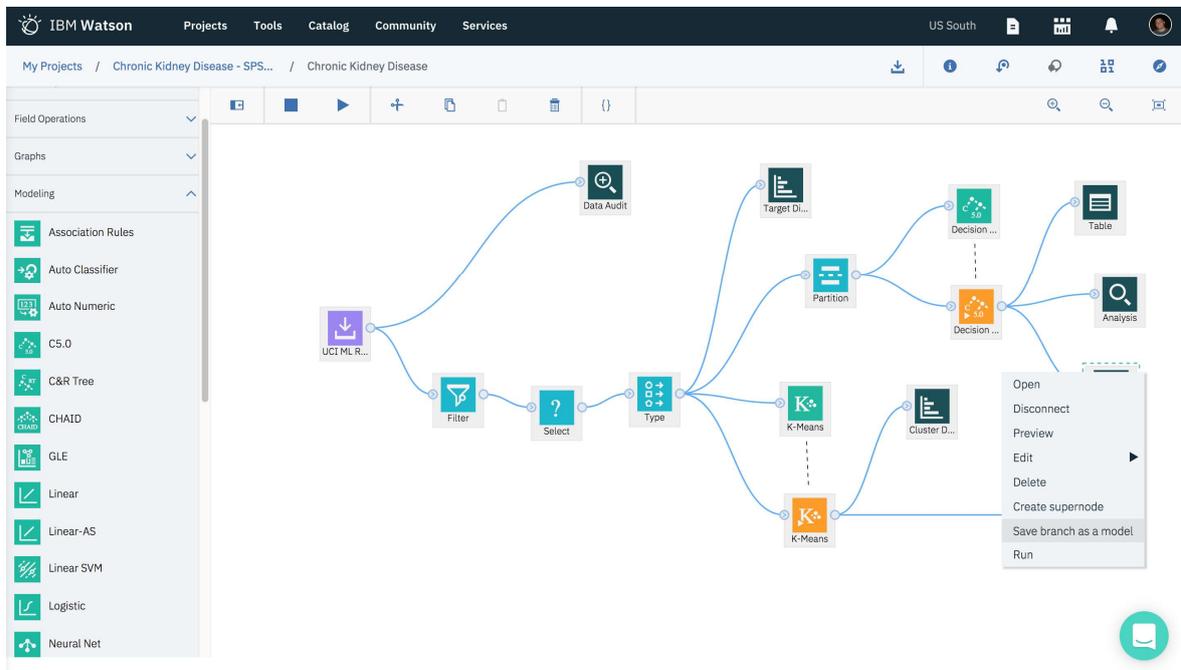
Como su competidor VertexAI, IBM Watson Studio es una plataforma No-Code basada en bloques, facilitando su uso por usuarios relacionados con el mundo del machine learning, pues, al igual que su competencia, tienen los problemas de que se

necesitan conocimientos previos para realizar todas las funciones necesarias para entrenar modelos predictivos.

No obstante, IBM Watson Studio sí que ofrece una herramienta automática e interactiva para poder visualizar y entender la relación entre los datos subidos por el usuario a la plataforma (EDA).

Finalmente, IBM Watson, no tiene ni un apartado de limpieza de datos que recomiende proactivamente al usuario qué acciones realizar.

IBM Watson Studio tiene unos costes para la arquitectura de 4CPUS y 16 GB de RAM de 1.9 USD por hora.



*Figura 7. Forma de interactuar con la plataforma IBM Watson Studio.*

Fuente: <https://www.budgetingsolutions.co.uk/technologies/ibm-watson-studio/>

**IV. Microsoft:** Microsoft Azure es otro de los competidores identificados. (<https://azure.microsoft.com/en-us/products/machine-learning>)

Microsoft Azure, como también VertexAI e IBM Watson Studio está basado en una solución No-Code utilizando bloques que sustituyen trozos de código por distintas

interfaces, acercando así esta plataforma a conocedores del mundo de machine learning que no tengan conocimientos de programación.

También como VertexAI, en Microsoft Azure el usuario tiene que elegir qué modelos de Machine Learning entrenar, junto con todas las fases que el usuario considere utilizar antes de entrenar estos modelos como qué hacer con las observaciones a las que les faltan datos, o dividir los datos en datos de entrenamiento o datos de prueba. Tampoco Microsoft Azure contine ninguna herramienta de EDA automática para visualizar los datos, pues para eso Microsoft tiene diseñada la herramienta Power BI. Asimismo, Azure no permite la creación de Ensembles de una forma fácil y sencilla. Finalmente, en cuanto a su precio, con la arquitectura de 4 CPUs y 16 GB de RAM, Azure tiene unos costes de 0.24 USD por hora.

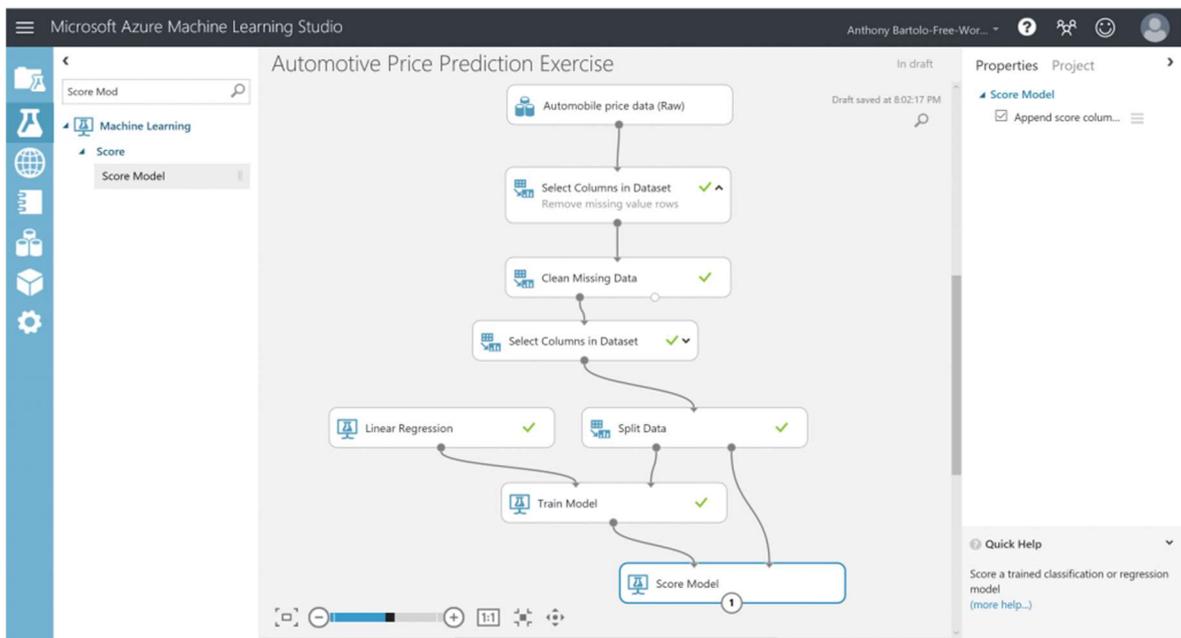


Figura 8. Forma de interactuar con la plataforma Microsoft Azure.

Fuente: <https://techcommunity.microsoft.com/t5/ITOps-Talk-Blog/Step-By-Step-Getting-Started-with-Azure-Machine-Learning/ba-p/331327>

### **3.2.3 CONCLUSIÓN**

Después de analizar los principales competidores que ofrecen soluciones de Machine Learning as a Service, podemos detallar la siguiente tabla que compara estas soluciones en diferentes áreas. Estas áreas representan segmentos importantes para la comercialización de las pequeñas y medianas empresas.

	Amazon SageMaker	Google VertexAI	IBM Watson Studio	Microsoft Azure
No es necesario programar	✗	✓	✓	✓
No es necesario tener conocimientos previos de Machine Learning	✗	✗	✗	✗
Fácil de usar	✗	✓	✗	✓
La plataforma incluye un preprocesado de datos	✗	✗	✗	✗
La plataforma incluye una herramienta que facilite visualizar los datos (EDA)	✗	✗	✓	✗
La plataforma incluye una herramienta para comparar en ella diferentes modelos	✗	✗	✗	✗
Precio	✓	~	~	✓
La plataforma ofrece la posibilidad de crear Ensembles de forma fácil	✗	✓	✗	✗

Tabla 1. Comparación de los principales competidores identificados

Con esta tabla podemos ver que las soluciones existentes en el mercado están orientadas a profesionales con habilidades y conocimientos tanto de programación como de técnicas de machine learning. Así, estas soluciones son más bien herramientas al servicio de los *data scientists* que pueden contratar las grandes empresas.

No existe una solución fácil de implementar, con una amigable interfaz de usuario que guíe al mismo por todo el proceso de preprocesado de datos, visualización de los datos, entrenamiento de los modelos, la posterior comparación de los modelos, y finalmente la predicción sobre un nuevo conjunto de datos para los usuarios inexpertos en estas técnicas.

## **Capítulo 4. IDEA DE NEGOCIO**

Puede existir un nicho prometedor para una plataforma que permita a usuarios sin experiencia técnica aprovechar los beneficios del machine learning de manera accesible y beneficiosa para las pequeñas y medianas empresas.

Considerando la creciente demanda de servicios de machine learning por parte de las PYMES en nuestro país, así como su capacidad para implementarlos (gracias a su amplio uso de ordenadores y conexión a internet), sumado al respaldo de subvenciones para su implementación, surge la viabilidad de una plataforma de Machine Learning as a Service orientada a un público no técnico. Actualmente, la mayoría de las soluciones existentes de Machine Learning as a Service están dirigidas principalmente a profesionales con conocimientos en programación, machine learning y análisis de datos.

### ***4.1 IDEA DE NEGOCIO DE UNA PLATAFORMA DE MACHINE LEARNING AS A SERVICE***

La idea de negocio consistiría en una plataforma de Machine Learning as a Service que permita a los clientes, especialmente a las PYMES y departamentos de grandes empresas que no utilicen estas técnicas de inteligencia artificial, acceder a servicios de aprendizaje automático a través del pago de una tarifa por su uso. Esta plataforma ofrecería una amplia gama de servicios que ayudan a los usuarios inexpertos entre las que se incluirían el preprocesado de datos, la visualización interactiva de los mismos (EDA), el entrenamiento de modelos de machine learning, la evaluación de estos modelos mediante métricas comunes e intuitivas, y una forma sencilla de realizar predicciones basadas en los datos proporcionados por el cliente.

Además, esta plataforma MLaaS guiará a los usuarios inexpertos a lo largo de todo el proceso de machine learning, proponiendo soluciones proactivamente a los problemas que pueda tener estos usuarios, como puede ser la forma en la que tratar los datos que faltan.

El autor de este TFG de Business Analytics ha publicado en GitHub el TFG de Ingeniería de Telecomunicaciones, que es un proyecto cuya funcionalidad es precisamente esa.

El objetivo de este TFG de Business Analytics, es valorar la viabilidad económica de este proyecto, añadiendo todos los cambios pertinentes para poder cumplir con la legalidad vigente, sobre todo la ley de protección de datos, y poder facturar por ello para tener viabilidad económica.

A continuación, se describe el proyecto creado por el autor de este mismo TFG.

#### **4.1.1 PROYECTO DE MACHINE LEARNING AS A SERVICE EXISTENTE**

Esta plataforma está desarrollada para facilitar su uso a cualquier persona que quiera entrenar modelos de machine learning utilizando sus datos independientemente de su conocimiento tanto de programación como de técnicas de machine learning y de preprocesado de datos.

La plataforma desarrollada cuenta con un sistema de autenticación de usuarios en el que los usuarios que quieran utilizar la plataforma de machine learning tienen que registrarse previamente con su dirección de correo electrónico y una contraseña. Esta funcionalidad permite garantizar la privacidad de los usuarios, pues se garantiza que un usuario no pueda acceder a los proyectos de otros usuarios.

Una vez estén registrados y hayan iniciado sesión, los usuarios dentro de la plataforma pueden crear distintos proyectos de machine learning subiendo el conjunto de datos que desean analizar y utilizar para entrenar los modelos de predicción implementados en la plataforma. Un ejemplo de este conjunto de datos podría ser para nuestro público objetivo los datos provenientes de una herramienta de CRM (Customer Relationship Management) para predecir si un cliente va a seguir comprando o no. Cada uno de los proyectos que tenga un usuario contendrá unos modelos de machine learning entrenados para el conjunto de datos

relacionados con ese proyecto. Además, la plataforma garantiza la privacidad de los usuarios al solo dejar acceder a los proyectos del usuario que tiene la sesión iniciada con la plataforma, y también permite el borrado de todos los ficheros relacionados con la plataforma de un proyecto en concreto si el usuario que lo creó así lo desea.

Una vez el usuario haya creado un proyecto (simplemente subiendo un fichero con los datos tabulados con los que quiere entrenar modelos de predicción), la plataforma le guía por el proceso de análisis y preprocesado de los datos. Inicialmente, la plataforma pide al usuario que elija la variable que quiere predecir en un futuro con los modelos de machine learning. Si seguimos el ejemplo anterior, la variable de Churn sería la variable a predecir en los modelos de machine learning en base al resto de variables que tenga nuestra base de datos de CRM.

Una vez se ha seleccionado la variable a predecir, la plataforma pide al usuario que especifique el tipo de problema que quiere solucionar. Si quiere construir un modelo de regresión, que quiere significar el predecir unos valores continuos, como puede ser el precio de un coche de segunda mano en base a datos descargados de distintas páginas web de venta de segunda mano; o un modelo de clasificación, en el que se quiere clasificar los datos en distintas categorías definidas previamente, como por ejemplo si un cliente va a seguir comprándonos o no. Como es posible que muchos usuarios no conozcan la diferencia entre regresión y clasificación, el propio programa establece por defecto la mejor opción analizando la variable a predecir en los datos que ha subido el usuario al proyecto. Así el usuario simplemente tiene que pinchar el botón de continuar dejando la opción por defecto. De todas formas, el usuario puede cambiar el tipo de predicción si los datos así lo permiten (pues no se puede hacer regresión sobre cadenas de caracteres, como por ejemplo una variable cuyos valores sean “Si”, “No”, y “Quizá”).

Tras seleccionar si el proyecto contiene un problema de clasificación o de regresión, se pide al usuario cambiar los tipos de datos de las variables con las que se van a entrenar los modelos. Esta etapa está introducida si el usuario quisiese que algunas columnas las tratase de forma distinta a como lo haría por defecto. Un ejemplo podría ser una variable que solo

tenga 5 valores, del 1 al 5, identificados inicialmente como una variable numérica y que por lo tanto implica un orden, pero en realidad, cada uno de los valores representan distintos aspectos completamente independientes unos de otros, y el usuario quisiese que fuese tratados como una variable categórica. Como en la situación anterior, este proceso necesario en cualquier preprocesamiento de datos está automatizado y el usuario no tiene por qué tomar ninguna decisión, pues se le ofrecen unas soluciones por defecto. Así, el usuario simplemente con darle a continuar confirma el tratamiento de datos que se va a aplicar a los datos de ahora en adelante en el proyecto.

Siguiendo dentro de la parte de análisis de datos y preprocesado de los mismos, se le ofrece al usuario formas en las que hacer el preprocesado de sus datos. La plataforma, analizando los datos relacionados con el proyecto, propone al usuario formas con las que tratar con diferentes datos que puedan afectar a la forma en la que se entrenan los modelos. En el caso de que los datos subidos a la plataforma contengan observaciones con datos que faltan, el programa, analizando todo el conjunto de datos propondrá al usuario qué hacer con esos datos que faltan. Entre las opciones que propone la plataforma está el borrar la columna (si los datos que faltan son la inmensa mayoría, o si los datos no aportan ningún conocimiento extra al modelo pues son únicos para cada observación); borrar las filas que contengan valores que falta; o asignarles a esos valores algún valor en concreto, como puede ser el 0 o el valor mediano o la moda. Una vez más, para cada variable, la plataforma recomienda proactivamente al usuario las acciones a tomar, lo que garantiza que un usuario sin ninguna experiencia pueda utilizar esta plataforma.

Finalmente, dentro de la parte de análisis de datos y preprocesado de los mismos, la plataforma proporciona al usuario una herramienta para visualizar de una forma interactiva los datos con los que va a entrenar a los modelos de machine learning. Esta etapa de EDA (Explanatory Data Analysis) es muy importante en el análisis de datos previo a construir modelos de machine learning pues permite al usuario entender cómo se distribuyen sus datos antes de crear los modelos. Estas visualizaciones interactivas cambian en función de los datos introducidos para poder representar de una manera óptima los datos, y entender mejor sus comportamientos.

Una vez el usuario ha completado la etapa de análisis de datos y de preprocesado, se le redirige a la parte de entrenamiento de modelos de machine learning. En esta etapa, a los usuarios se les ofrece dos formas en las que pueden entrenar los modelos de aprendizaje automático. La primera forma es la forma “Normal”, en la que no se espera que el usuario sepa qué modelos van a funcionar mejor para el conjunto de datos asociados a ese proyecto, y por tanto se entrenan todos los modelos disponibles en la plataforma, buscando antes de una manera óptima los hiperparámetros para cada uno de los modelos para ese conjunto de datos. Así, con este modo “Normal”, que está por defecto, el usuario no necesita por qué saber ni qué hiperparámetros necesita el modelo, ni mucho menos qué modelos pueden funcionar mejor. Este modo “Normal” simplemente entrena todos los modelos disponibles y posteriormente en la etapa de valoración de modelos entrenados, mirará cuál es el modelo que mejor funciona con los mejores hiperparámetros para ese conjunto de datos.

No obstante, existe un modo “Experto” en el que el usuario puede elegir los modelos de machine learning que quiere entrenar, junto con qué parte de los datos quiere destinar a valorar el modelo y cuánto a entrenar (en el modo Normal es un 30%). Posteriormente, para cada uno de los modelos seleccionados que el usuario haya seleccionado para entrenar, el usuario puede elegir los hiperparámetros para entrenar el modelo, o que la plataforma automáticamente busque de forma óptima los mejores hiperparámetros para ese modelo en concreto.

Posteriormente, una vez el usuario haya entrenado al menos un modelo, puede ir a la etapa de valoración de los modelos creados. En esta etapa se ofrece una comparación entre todos los modelos creados utilizando los datos de prueba puestos aparte durante la etapa de entrenamiento. Para comparar los modelos y valorarlos se ofrecen métricas comunes como el error cuadrático medio para regresión como la precisión en clasificación. Además de establecer una métrica en común para todos los modelos dentro de proyecto y ordenarlos en función de esa métrica, se le expone al usuario varias gráficas que explican el funcionamiento de este modelo con los datos de prueba. Estas gráficas (como puede ser una matriz de confusión o una distribución de los errores), están explicadas de una forma fácil para que el usuario pueda entender la información que transmiten. Aparte de estas gráficas

fáciles, existen gráficas que explican para los usuarios más expertos alguna información extra del modelo, o la forma en la que se buscan óptimamente los hiperparámetros. Estas gráficas están bajo una pestaña que notifica que están diseñadas para el usuario experto, y no se espera que el usuario medio de la plataforma pueda entenderlas.

Además, otra funcionalidad que permite esta plataforma es crear Ensembles, o juntar varios modelos ya entrenados para que la predicción de la suma de todos ellos mejore el rendimiento de un modelo solo. Para hacerlo, una vez se hayan entrenado más de 2 modelos en el proyecto, el usuario puede elegir qué modelos quiere juntar, y la plataforma, sin que el usuario haga nada más, junta esos modelos y lo trata como un nuevo modelo, añadiéndolo a la valoración y comparación junto a los demás modelos.

Finalmente, una vez esté entrenado algún modelo, el usuario puede predecir sobre un nuevo conjunto de datos que suba a la plataforma. Este nuevo conjunto de datos va a tratarse de igual manera que los datos originales en la fase de preprocesado, y se devolverá al usuario el conjunto de datos con la predicción de la variable a predecir especificada al comienzo del proyecto. En el caso de que haya datos que tengan que ser borrados (como en los datos originales si elegía eliminar esas observaciones), en el descargable con la predicción se seguirán mostrando esos datos solo que sin predicción. Se informará al usuario que no hay predicción al haber campos vacíos, y haberse especificado en la etapa de preprocesado que esos datos iban a ser borrados.

#### **4.1.2 IDEA DE NEGOCIO**

Este proyecto está orientado al usuario inexperto guiándole por todas las etapas del proceso de explotación de datos, y encaja perfectamente con el perfil de trabajadores de una PYME, sin conocimientos ni de programación ni del mundo del machine learning y el análisis de datos.

Así, que la idea de negocio sería desplegar esta plataforma en un servidor para que pueda ser comercializada al público general, incluyendo todos los cambios pertinentes para cumplir tanto la legalidad como poder facturar por ello.

## **4.2 BARRERAS**

Pese a ha quedado constatada la demanda de este servicio por parte de nuestro público objetivo en los dos capítulos anteriores, existen diferentes barreras a la hora de empezar a comercializar esta plataforma. Las dos barreras más grandes son la regulación y la entrada en el mercado.

### **4.2.1 REGULACIÓN**

Es muy importante poder cumplir con la legislación vigente para poder comercializar este proyecto. En concreto, deberemos introducir los cambios pertinentes para cumplir tanto con las leyes de cualquier página web en la eurozona, como en concreto la ley europea de protección de datos al tener como modelo de negocio que empresas suban datos de sus clientes para poder analizarlos y entrenar modelos de predicción con ellos.

Este problema y su solución se desarrollarán más adelante en el capítulo 5.

### **4.2.2 ENTRADA AL MERCADO**

Además, la forma en la que se comercialice esta plataforma va a ser crucial para su correcta puesta en el mercado. Al estar compitiendo con grandes multinacionales estadounidenses es necesario elaborar una forma para promocionar esta plataforma, a la vez que diseñar los precios que se cobrarán a los clientes por su uso de la aplicación.

Para ello, en el capítulo 6 se comparará el producto, ya con las modificaciones pertinentes para que cumpla la ley de protección de datos, con las empresas cubiertas anteriormente que ofrecen productos similares, y se desarrollará un plan de marketing para poder vencer las posibles barreras que nos encontremos.

## Capítulo 5. REGULACIÓN

Como se dice en el AI Index 2023 Annual Report (Clark & Perrault, 2023), el interés de los legisladores mundiales en la inteligencia artificial está creciendo, multiplicándose las menciones a la inteligencia artificial un 650% desde 2016 hasta 2022. Este interés de los legisladores puede estar siendo potenciado por el incremento del uso y de la eficacia de estas técnicas de inteligencia artificial por empresas frente a los investigadores y la academia, que se están quedando atrás por los costes que conlleva crear sistemas de inteligencia artificial de última generación, como se enuncia en este mismo informe.

Aunque el uso de la inteligencia artificial tiene el potencial de impulsar el crecimiento económico (Furman & Seamans, 2019), los reguladores están también preocupados con el uso que pueden darle al machine learning las empresas por poder violar los derechos fundamentales de sus ciudadanos cuando estas técnicas tomen decisiones que hasta ahora las tomaban humanos, como conceder hipotecas o diagnosticar cánceres (Candelon et al., 2021).

Según Bessen et al. (2020), estos algoritmos de machine learning e inteligencia artificial se basan en infinitud de datos, sobre todo datos personales, y dos de las leyes que más restringen el uso de estos datos son la ley Europea de Protección de Datos, o GDPR por sus siglas en inglés (Reglamento 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, 2016) y la ley Californiana de protección de datos también llamada CCPA por sus siglas en inglés (California consumer privacy act, 2018).

Además, existen actualmente proyectos de leyes en Europa para legislar entre otras cosas, la inteligencia artificial, en el programa de La Década Digital de Europa (European Commission, 2021), cuyos principios son: el respeto a los derechos fundamentales de los ciudadanos en el mundo digital; el fomento de la inclusión a todas las personas en el mundo digital; la libertad de elección qué servicios digitales utiliza, como la inteligencia artificial;

la libre elección de las personas a la hora de decidir cómo se tratan sus datos; la seguridad de las personas en el mundo digital; y el impulso de estas tecnologías de inteligencia artificial para mejorar el medio ambiente.

Dentro de este marco de década digital, se propone una inteligencia artificial más ética, como se recoge en el Libro Blanco de la Comisión Europea (2020) *sobre la inteligencia artificial - un enfoque europeo orientado a la excelencia y la confianza*, mientras se impulsa su uso en las empresas, estableciéndose un objetivo de un 75% de las compañías europeas utilicen servicios de computación en la nube (o cloud computing en inglés), datos masivos (o big data en inglés), o inteligencia artificial, como se detalla en el citado programa de La Década Digital de Europa.

Estas leyes frenan la competitividad de las empresas pues dificultan su acceso a los datos de usuarios (Bessen et al., 2020), pero tienen como objetivo proteger los derechos de sus ciudadanos en el mundo digital, como indican tanto la ley GDPR como la CCPA.

No obstante, la ley Europea de Protección de Datos, GDPR, es en la que se centrará este estudio, tanto por ser europea y por lo tanto la que aplica a las PYMES españolas, por ser la que está sentando cánones para el resto de las leyes de protección de datos en otros países (Albrecht, 2016).

## ***5.1 LEY EUROPEA DE PROTECCIÓN DE DATOS (GDPR)***

La ley Europea de Protección de Datos, o GDPR, se aprobó en 2016, aunque entró en vigor en mayo de 2018, dando a las empresas un plazo de 2 años para adaptarse a este cambio de legislación.

La GDPR tiene siete principios de donde se derivan algunos derechos de los usuarios. Estos principios y sus derechos relacionados son los siguientes:

- I. **Principio de legalidad, equidad y transparencia:** las entidades que recopilen datos de usuarios deben hacerlo de acuerdo con la legalidad vigente, informando en todo

momento al usuario del uso al que van a ser sometidos, y limitándose a ese uso hasta que el usuario no cambie su consentimiento.

**-De este principio se deriva el derecho del usuario a la transparencia:** que establece que toda información que se dé al usuario sobre el procesamiento de sus datos debe ser entendible por el usuario al que se está dirigiendo la entidad recopiladora. Además, la entidad recopiladora debe informar al usuario si va a enviar esta información a otra entidad, indicando la entidad en cuestión, y la seguridad de los datos a la hora de enviárselos.

- II. **Principio de limitación del propósito:** el aviso que dé la entidad que recopila datos a los usuarios tiene que informar del propósito de la recopilación de los datos. Así, los datos solo pueden ser recopilados y usados para los fines a los que el usuario ha dado su consentimiento en el mensaje de la entidad recopiladora de datos.

**-De este principio se deriva el derecho a la restricción:** este derecho establece que un usuario puede restringir a una entidad el procesamiento de sus datos personales, que aún puede almacenarlos, pero no puede realizar ningún procesamiento de los mismos.

**-De este principio también se deriva el derecho a objetar:** con este derecho, los usuarios pueden negar que sus datos sean tratados de algunas maneras, sobre todo estando relacionadas con funciones de marketing.

- III. **Principio de minimización de los datos:** las entidades deben recopilar el mínimo conjunto de datos de forma justificada para llevar a cabo los fines especificados al usuario previo a su recopilación.

**-De este principio se deriva el derecho de acceso:** este derecho establece que un usuario puede pedir a la entidad recopiladora de datos una copia de los datos que esta está procesando, para así ver los datos que ésta tiene suyos.

- IV. **Principio de exactitud:** los datos deben precisos, y sobre todo mantenerse actualizados. Además, se debe garantizar el borrado completo de los datos personales si éstos son inexactos.

**-De este principio se deriva el derecho a la rectificación:** en la que los usuarios tienen el derecho de ofrecer a la plataforma sus datos actualizados, y que éstos sean efectivamente actualizados en la plataforma, borrando los anteriores.

- V. **Principio de limitaciones de almacenamiento:** este principio se basa en la minimización de los datos y enuncia que éstos deben “mantenerse en una forma que permita la identificación de los interesados por un período no superior al necesario”. Este principio establece que los datos deben ser guardados el tiempo necesario y que se ha explicitado en el aviso al usuario antes de recoger sus datos, pues éste es el dueño de los mismos.

**-De este principio se deriva el derecho a ser olvidado:** los usuarios pueden pedir el borrado total de sus datos personales de la plataforma, y la entidad tiene que cumplir con ese derecho.

**-De este principio también se deriva el derecho a la portabilidad de los datos:** este derecho estipula que, al ser el usuario dueño de sus datos, un usuario tiene el derecho de enviar sus datos personales a otra entidad para que los procese.

- VI. **Principio de integridad y confidencialidad:** este principio requiere que los datos se protejan contra el procesamiento ilegal en caso de pérdida o robo de estos. De esta forma, las empresas al guardar los datos de sus clientes deben establecer sistemas para proteger la identidad de los clientes, recomendándose la pseudonimización o la anonimización.

- VII. **Principio de responsabilidad:** este principio establece que la entidad recopiladora de los datos es la responsable del cumplimiento de la Ley Europea de Protección de Datos (GDPR), a la vez que tiene que ser capaz de demostrar dicho cumplimiento.

**-De este principio se deriva el derecho a la compensación:** con este derecho, los usuarios tienen la potestad a una compensación por la vulneración de cualquier principio o derecho especificado en la GDPR.

## ***5.2 INCONVENIENTES DE LA GDPR CON LA PLATAFORMA BAJO ESTUDIO***

La plataforma a desarrollar va a tener como clientes a las PYMES o departamentos de las grandes empresas. Estos clientes utilizarán la plataforma subiendo los datos que éstos poseen para extraer conocimiento de éstos que pueda ayudar a la empresa. Por lo tanto, esta plataforma es solo una ayuda que automatiza la forma de entrenar modelos de machine learning, y los que deciden procesar los datos son nuestros clientes.

Por lo tanto, esta plataforma asumiría que nuestros clientes cumplen con la ley europea de protección de datos, al ser ellos los que almacenan los datos de sus clientes, y ellos los que deciden procesar estos datos.

Los mayores inconvenientes que tiene la aplicación actualmente con la ley europea de protección de datos son los siguientes:

- Nuestros supuestos clientes, PYMES o departamentos de grandes empresas, al recopilar los datos, no especificaron que estos datos se iban a enviar a otra entidad, la nuestra, para ser procesados en nuestra plataforma, y por lo tanto nuestra plataforma no puede leerlos.
- Debemos trasladar la responsabilidad a nuestros clientes (como PYMES) del tratamiento de los datos, pues no tenemos conocimiento de la información dada al usuario al recopilar sus datos.
- Además, actualmente la aplicación no permite el borrado de los usuarios, que en nuestro caso serían los usuarios que se den de alta desde las PYMES o los departamentos de grandes empresas.

Aparte de estos inconvenientes con la GDPR, la actual plataforma no contiene un método de facturación, que tendría que añadirse para que se pueda comercializar.

### **5.3 PROPUESTA TECNOLÓGICA PARA CUMPLIR CON GDPR**

Para cumplir con la ley europea de protección de datos, se propone que esta plataforma no se ejecute en la nube, que es como está diseñada actualmente, sino que sea ejecutada en los ordenadores de las empresas. Así, los datos a ser procesados por las empresas clientes no saldrán de sus dispositivos donde están almacenados.

Por lo tanto, lo que se propone es paquetizar la plataforma desarrollada para que pueda ser instalada en los ordenadores de nuestros clientes, y que éstos, abriendo la aplicación en su escritorio accedan a la misma interfaz que tiene el diseño de la plataforma actualmente, con todas sus funcionalidades.

Además, tendremos que crear una nueva aplicación que sí que instalaremos en un servidor en la nube que controlará a nuestros clientes, indicando cómo utilizan nuestra aplicación para poder facturar por ello.

### **5.4 IMPLICACIONES DE ESTE CAMBIO**

Este cambio propuesto cambia por completo el modelo de negocio. Nuestra empresa ya no es un servicio online, que ejecuta todos sus procesos (incluidos el de entrenamiento de los modelos y la predicción de ellos) en la nube, sino que se ejecutan en el ordenador de nuestros clientes.

Esto conlleva los siguientes cambios:

- No podemos cobrar como nuestros competidores por tiempo de ejecución puesto que un mayor tiempo de ejecución no se repercute en un mayor coste para esta empresa, al ejecutarse en los ordenadores de nuestros clientes.
- Deberemos cobrar por el uso de la plataforma, es decir, tanto por los modelos de machine learning que se entrenen, como por el número de datos que se predigan con esta herramienta.

- Además, como se ejecuta en los ordenadores de nuestros clientes, debemos garantizarnos que estos están siempre conectados a internet para poder facturar debidamente por el uso que se da a la plataforma. Para esto se modificará la aplicación para bloquear su uso en el caso de que no pueda acceder a nuestro servidor para informar de todo el uso que se dé a la aplicación.

## **5.5 CAMBIOS TECNOLÓGICOS EN EL PROYECTO**

Para adaptar este proyecto a la legalidad vigente y paquetizarlo, hay que crear una nueva aplicación de control cuya función sea la de monitorizar el uso que hacen los usuarios (las PYMES) de la aplicación que tienen instalada en sus ordenadores y poder facturar por ello.

Para ello, esta nueva aplicación, desplegada en servidores en la nube y accesible a todos los usuarios, deberá controlar el número de modelos que entrena un usuario y el número de datos que predice el usuario, bloqueando el uso del programa ejecutado en local por los usuarios si no se tiene la aprobación de esta nueva aplicación.

Para controlarlo, la solución óptima sería que la aplicación de escritorio paquetizada se pusiese en contacto con la aplicación de control, ubicada en internet, mediante un POST request, identificando al usuario e informando a esta aplicación de control del uso que ese usuario quiere darle a la plataforma. Esta aplicación calcularía si el usuario tiene saldo suficiente para realizar tal acción (como entrenar un modelo o realizar una predicción), y si la respuesta es afirmativa, la aplicación paquetizada realizaría tal acción.

Para realizarlo, en primer lugar, se creará la aplicación de control. Posteriormente garantiremos la conectividad entre la aplicación de control y la aplicación a paquetizar. Una vez desarrollada la conectividad entre aplicaciones, se modificará la creación de proyectos para informar a nuestros clientes (las empresas), que los datos que se suban a la plataforma deben haber sido recopilados informando al usuario de que podrían ser tratados. Después se deberá implementar que nuestros clientes puedan descargar la información que se almacena de ellos en la plataforma de control, junto con el borrado completo de su

información. Finalmente, se paquetizará la aplicación modificada que contiene toda la lógica para la creación de modelos de machine learning.

### **5.5.1 CREACIÓN DE LA NUEVA APLICACIÓN DE CONTROL**

El primer paso que haremos será la creación de una nueva aplicación que controlará el uso que hagan nuestros clientes (las PYMES) de esta plataforma en sus ordenadores locales.

Para crear una aplicación web que controle a la aplicación que se paquetizará, es necesario elegir un framework de desarrollo de aplicaciones web que gestione tanto las peticiones HTML como el acceso a las bases de datos. Hay muchos frameworks de desarrollo, pero Django ha sido elegido porque está basado en Python, por su rapidez, por su seguridad (Forcier et al., 2008), y sobre todo por similitud con la aplicación que va a ser paquetizada, que también ha sido desarrollada en Django.

Además, Django posibilita la división de un proyecto en distintas aplicaciones, lo cual favorece la organización modular del proyecto y facilita su escalabilidad (Ghimire, 2020).

Para gestionar todas estas distintas aplicaciones, un proyecto de Django tiene una carpeta general del proyecto que enruta al usuario, en función de la ruta URL a la que se accede, a las distintas aplicaciones, para que las vistas de éstas devuelvan una página web a través de los Templates de Django, que son archivos HTML que permiten que se les pase información desde las vistas de cada aplicación.

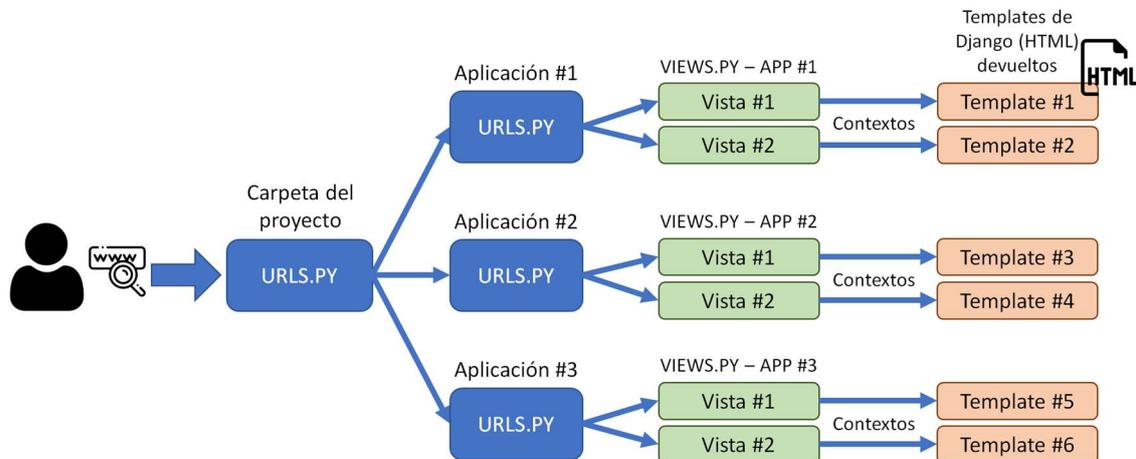


Figura 9. Funcionamiento modular de Django

Para iniciar un nuevo proyecto en Django es necesario tener instalada la librería de Django en Python, instalada gracias al instalador de paquetes de Python PIP mediante el siguiente código en el terminal:

```
pip install django
```

Una vez esté el paquete de Django instalado, se creará un nuevo proyecto en Django que será la aplicación de control, llamada "MLaaSController". Para crear un nuevo proyecto introducimos el siguiente comando en el terminal dentro de la carpeta donde querremos crear el proyecto:

```
django-admin startproject MLaaSController
```

A continuación, para cargar la base de datos por defecto, se indicará al proyecto que debe hacer las migraciones a las bases de datos necesarias (que hasta ahora son solo las bases de datos por defecto de Django).

```
python manage.py migrate
```

No obstante, aún no se ha creado ninguna aplicación que recoja la lógica para guardar la información de los clientes, así como su uso de la aplicación. Esta aplicación se llamará

“clientes”, y para crearla hay que introducir el siguiente código en el terminal en la carpeta del proyecto:

```
python manage.py startapp clientes
```

Con esta aplicación creada, la carpeta del proyecto de Django para una aplicación de control tiene la siguiente forma. Existen dos carpetas, una la general del proyecto, llamada como el mismo proyecto, “MLaaSController”, y una carpeta perteneciente a la aplicación de control de los usuarios llamada “clientes”. Además, podemos ver que está instalada la base de datos por defecto en Django, SQLite3, junto con el archivo que ayuda a ejecutar el servidor, crear aplicaciones, y gestionar las bases de datos: *manage.py*.

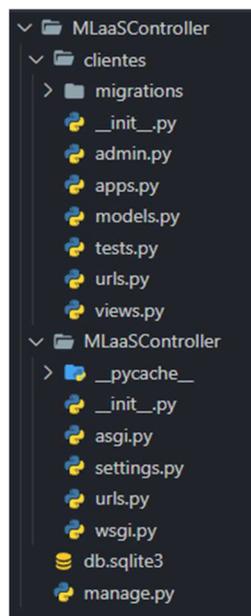


Figura 10. Estructura de carpetas del proyecto

Ahora tendremos que integrar esta nueva aplicación dentro del proyecto, a la vez que especificar qué ruta URL se le asignará a la aplicación “clientes”. Para integrar la aplicación dentro del proyecto, deberemos notificárselo a la carpeta general del proyecto, introduciendo esta nueva aplicación dentro de la lista de las aplicaciones instaladas, en el archivo *settings.py*. Además, en el archivo *urls.py* de la carpeta del proyecto, añadimos que a través de la ruta URL “/clients/”, se accederá a la aplicación de “clientes”.

Adicionalmente, se deberá crear una base de datos con los clientes en esta nueva aplicación, que contenga, tanto su usuario y contraseña, como sus datos de uso de la plataforma y saldo disponible. Para crear esta base de datos de clientes, deberemos crear una nueva clase u objeto que represente a estos usuarios, pues Django permite la programación orientada a objetos (POO por sus siglas en inglés).

Además, es necesario que estos clientes puedan registrarse o iniciar sesión en esta nueva aplicación de control para poder abonar dinero a cambio del saldo necesario para utilizar la plataforma. Para que los clientes puedan acceder a la plataforma, la nueva clase de Cliente deberá heredar de la clase por defecto de Django “AbstractBaseUser”, para que esta clase sea la que necesite autenticarse en la plataforma.

A continuación se proporciona el código necesario para crear la clase de Cliente, junto con su controlador, Cliente Manager, definidas ambas en el archivo models.py en la aplicación “clientes”:

```
1. class ClienteManager(BaseUserManager):
2.     def create_user(self, email, username, password=None):
3.         if not email:
4.             raise ValueError('Usuarios deben tener email')
5.         if not username:
6.             raise ValueError('Usuarios deben tener nombre de usuario')
7.         user = self.model(
8.             email = self.normalize_email(email),
9.             username=username,
10.            password=make_password(password)
11.        )
12.        #user.set_password(password)
13.        user.save(using=self._db)
14.        return user
15.    def create_superuser(self, email, username, password):
16.        user = self.create_user(
17.            email = self.normalize_email(email),
18.            username=username,
19.            password = password,
20.        )
21.        user.is_admin= True
22.        user.is_staff = True
23.        user.is_superuser = True
24.
25.        user.save(using=self._db)
26.        return user
27.
28. class Cliente(AbstractBaseUser):
29.     #required
30.     email = models.EmailField(verbose_name="email",max_length=100,unique=True)
31.     username = models.CharField(max_length=50,unique=True)
```

```

32.     date_joined = models.DateField(auto_now_add=True, verbose_name="date_joined")
33.     last_login = models.DateField(auto_now=True, verbose_name="last_login")
34.     is_admin = models.BooleanField(default=False)
35.     is_active = models.BooleanField(default=True)
36.     is_staff = models.BooleanField(default=False)
37.     is_superuser = models.BooleanField(default=False)
38.
39.     #lista de atributos para controlar
40.     saldo = models.FloatField(default=0)
41.
42.     num_proyectos_machine_learning = models.IntegerField(default=0)
43.
44.     num_modelos_machine_learning_entrenados_ultimo_mes = models.IntegerField(default=0)
45.
46.     num_predicciones_realizadas_ultimo_mes = models.IntegerField(default=0)
47.
48.
49.     USERNAME_FIELD = 'email'
50.     REQUIRED_FIELDS = ['username']
51.
52.     objects = ClienteManager()
53.
54.     def __str__(self):
55.         return self.username
56.
57.     def has_perm(self, perm, obj=None):
58.         return self.is_admin
59.
60.     def has_module_perms(self, app_label):
61.         return True
62.

```

Además, para poder controlar el uso de nuestros clientes de la plataforma y su saldo, tendremos que crear las siguientes funciones dentro de la clase de Cliente. Estas funciones se encargarán de modificar, para cada cliente que tengamos, su saldo, y el resto de información que se guarde de él, como puede ser el número de proyectos que tiene, el número de modelos que ha creado en el último mes, y el número de predicciones realizadas en el último mes.

```

1.     # Funciones para controlar el saldo y el uso de la plataforma
2.
3.     def aumentar_saldo(self, cantidad):
4.         self.saldo += cantidad
5.         self.save()
6.
7.     def comprobar_saldo(self, cantidad):
8.         return (self.saldo >= cantidad)
9.
10.    def disminuir_saldo(self, cantidad):
11.        if self.comprobar_saldo(cantidad):
12.            self.saldo -= cantidad
13.            self.save()
14.            return True

```

```
15.         else:
16.             return False
17.
18.     def aumentar_num_proyectos_machine_learning(self):
19.         self.num_proyectos_machine_learning += 1
20.         self.save()
21.
22.     def aumentar_num_modelos_machine_learning_entrenados_ultimo_mes(self, num_modelos=1):
23.         self.num_modelos_machine_learning_entrenados_ultimo_mes += num_modelos
24.         self.save()
25.
26.     def aumentar_num_predicciones_realizadas_ultimo_mes(self, num_predicciones):
27.         self.num_predicciones_realizadas_ultimo_mes += num_predicciones
28.         self.save()
29.
30.     def resetear_datos_ultimo_mes(self):
31.         self.num_modelos_machine_learning_entrenados_ultimo_mes = 0
32.         self.num_predicciones_realizadas_ultimo_mes = 0
33.         self.save()
34.
```

## 5.5.2 GARANTIZAR LA CONECTIVIDAD CON LA NUEVA APLICACIÓN DE CONTROL

Una vez creada esta nueva plataforma de control, la segunda tarea será obligar a que la aplicación, que se descargará en los ordenadores de nuestros clientes (PYMES), esté conectada a nuestra aplicación de control, para ser informados del uso de la plataforma por parte de los usuarios y poder cobrar por ello. Con esto impediremos que los usuarios puedan utilizar la aplicación en local sin conexión a internet y no ser cobrados por su uso.

Como se ha comentado anteriormente, la plataforma paquetizada deberá comunicarse con la plataforma de control para informar de las acciones que quiere realizar. Posteriormente, después de que la aplicación de control compruebe si el cliente tiene saldo suficiente para realizar esa acción, la aplicación paquetizada realizaría esa acción en el ordenador del cliente. Una vez terminada la acción a realizar, la aplicación paquetizada informaría a la aplicación de control, y ésta reduciría el saldo asociado a esta tarea. Este flujo se detalla en la siguiente figura.

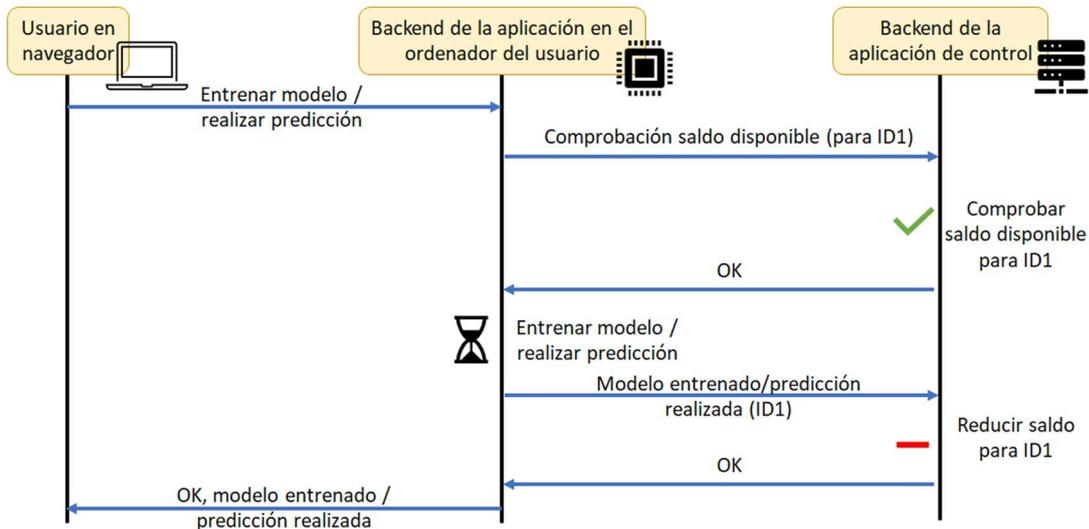


Figura 11. Funcionamiento propuesto para la aplicación

No obstante, este funcionamiento propuesto no funciona por motivos de seguridad. La aplicación de control tiene que verificar que el usuario que le notifica sobre el uso de la aplicación paquetizada está realmente dado de alta en esa aplicación. En caso contrario, cualquier persona con un navegador podría enviar información a la aplicación de control del uso que está haciendo cualquier usuario en la plataforma, aunque sea mentira, como se detalla en la figura 12.

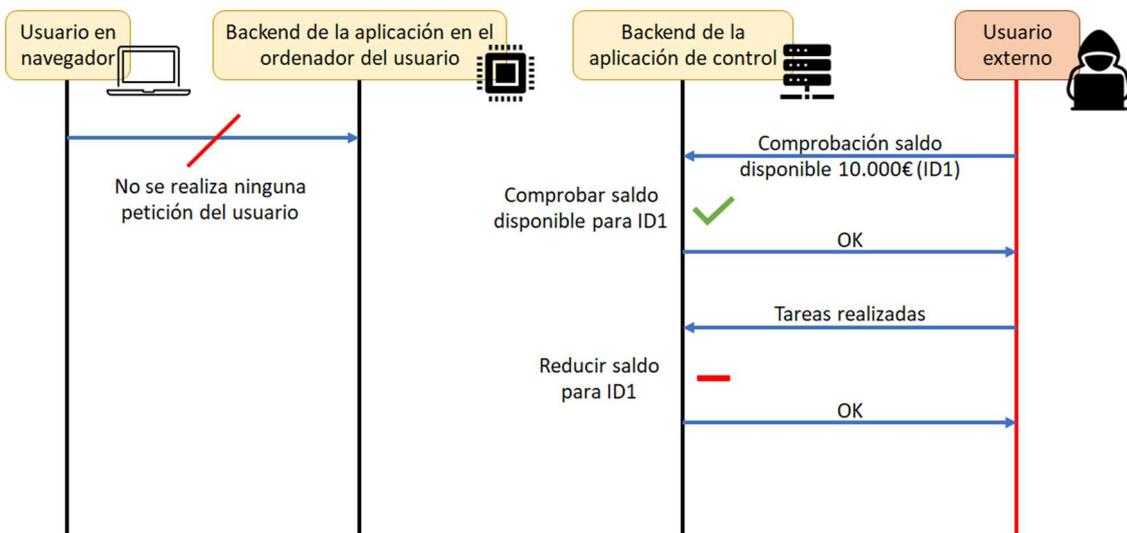


Figura 12. Ejemplo de mal funcionamiento del flujo propuesto para la aplicación

Debido a esta situación, este flujo propuesto debería cambiarse para enviar el usuario y la contraseña a la aplicación de control para poder autenticar al usuario.

Pero esto no puede hacerse, ya que la aplicación a paquetizar está desarrollada con el entorno de trabajo (o framework en inglés) de Django. Este framework, por seguridad, codifica las contraseñas mediante un proceso mediante el cual para el propio programa es imposible de obtener la contraseña utilizada por el usuario para registrarse o iniciar sesión, pues lo que se guarda es una representación de la contraseña (su hash calculado con el algoritmo PBKDF2). Además, dos contraseñas iguales nunca tendrán igual representación, pues a la contraseña que va a ser transformada, se le añade una cadena de caracteres aleatoria que hace que dos contraseñas iguales tengan representaciones completamente distintas, y, sobre todo, que no se pueda obtener la contraseña original. Este proceso es de un único sentido, y al iniciar la sesión de un usuario la plataforma calcula la representación de la contraseña introducida por el usuario y comprueba que ambas representaciones -la calculada y la guardada- sean idénticas.

Debido a esto, si se registra un usuario en ambas aplicaciones, la representación de la contraseña guardada en ambas aplicaciones (la local y la de control en la nube) serán distintas, y por lo tanto la primera aplicación no podría autenticarse en la segunda con la contraseña a la que puede acceder.

Para solucionar este problema se propone el uso de JSON Web Token, o JWT por sus siglas en inglés. Este es un mecanismo que permite garantizar que la información recibida ha sido enviada por quien dice enviarla durante un breve periodo de tiempo. Más concretamente, un JWT, es una cadena de caracteres que contiene la siguiente información:

- I. El algoritmo que se utilizará para validar el JWT.
- II. La información del Token. Esto incluye quién lo expidió (la plataforma de control), durante cuánto tiempo, y para quién está destinado (el cliente).
- III. La firma criptográfica del Token, que verifica que solo será accesible por las personas autorizadas.

Además, se deberá garantizar que la conexión se realiza mediante HTTP5 para garantizar el cifrado de estos datos, y así no solo garantizar la autenticación, sino también la integridad de la comunicación.

Para implementar esta solución, se deberá cambiar la forma en la que se registran los usuarios al siguiente flujo.

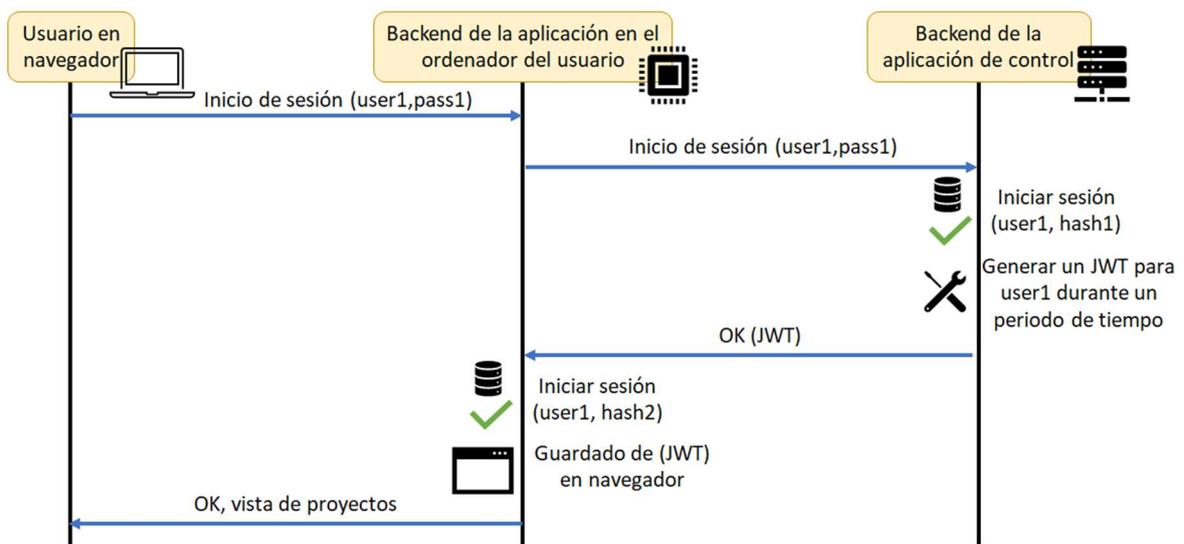


Figura 13. Funcionamiento del inicio de sesión de los usuarios

En este flujo, el usuario, cuando inicia sesión en la aplicación paquetizada, que se ejecuta en local, ésta inicia sesión en la aplicación de control utilizando el usuario y contraseña proporcionados por el usuario. La aplicación de control entonces calcula la representación de la contraseña de este usuario (hash1), y comprueba que corresponda a la del usuario. Si ambas representaciones son idénticas, la aplicación de control crea un JWT relacionado al usuario que acaba de iniciar sesión y se lo devuelve a la aplicación paquetizada además de almacenarlo temporalmente. Posteriormente, la aplicación de escritorio inicia sesión al usuario calculando primero la representación de la contraseña proporcionada (hash2), y posteriormente comprobando si ésta concuerda con la guardada en la base de datos. Además, esta aplicación paquetizada guarda este JWT en la memoria del navegador para poder autenticar al usuario en siguientes ocasiones frente a la plataforma de control. Finalmente,

una vez esté autenticado el usuario también en la aplicación paquetizada, se redirige al usuario a la vista de sus proyectos, como en la aplicación ya desarrollada.

Posteriormente, cuando este usuario quiera utilizar algún servicio de la aplicación, como entrenar un modelo o realizar una predicción, el funcionamiento de la aplicación debe ser el descrito en la figura 13. Una vez el usuario decide entrenar un modelo de machine learning, o realizar una predicción, la aplicación paquetizada envía esa información a la aplicación de control en la nube junto con el JWT guardado en la memoria de la sesión. La aplicación de control procederá primero a verificar el JWT para garantizar la autenticación de la comunicación, y posteriormente comprobará el saldo disponible para el usuario y sustraerá el saldo que requiere esa acción. Una vez se ha confirmado el JWT y se ha sustraído el saldo, se informará a la aplicación paquetizada de seguir adelante y ejecutar la acción que le había solicitado el usuario. Una vez terminada esta tarea, se informará a la aplicación de control que esta tarea se ha terminado, y posteriormente la aplicación paquetizada informará al usuario que esta tarea se ha realizado con éxito.

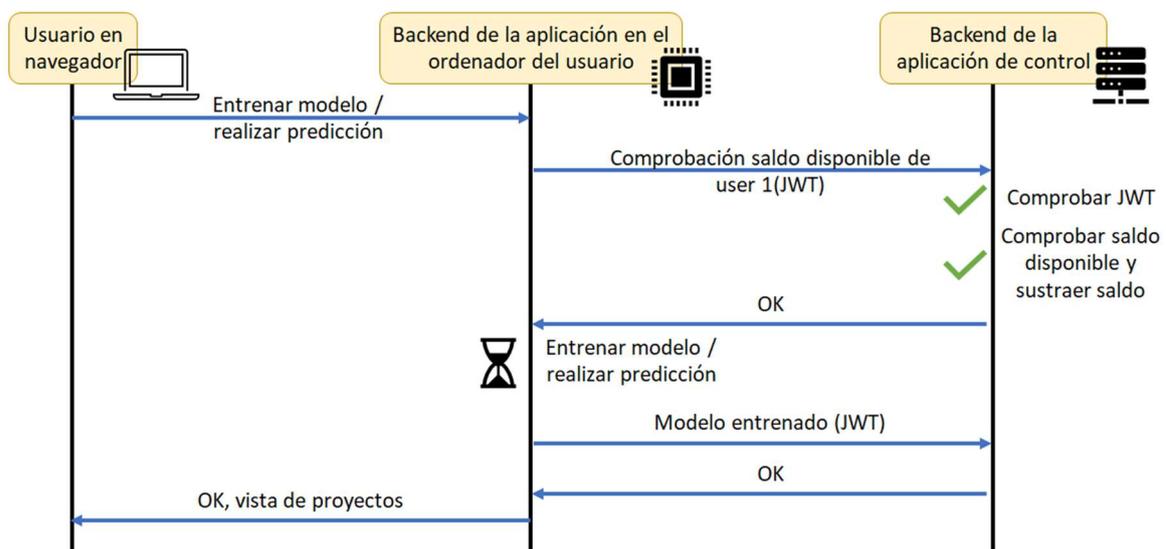


Figura 14. Funcionamiento del proceso de uso de la plataforma y su cobro por ello

Con este flujo, se consigue autenticar que es el usuario el que está haciendo realmente el uso de la aplicación (y no un tercero que intente que se le cobre más), y se cobra por el uso antes

del propio uso, previendo así que se pueda perder la conexión con la aplicación de control y no se facture por el uso de la plataforma.

Para implementar este flujo primero tendremos que habilitar JWT en la nueva aplicación de control. Para hacerlo, se utilizará el entorno de trabajo, o framework, de *Django Rest Framework*, junto con el paquete que ayudará a gestionar los JSON Web Tokens, que en este caso será *Simple JWT*. Para instalar ambos se deberá de ejecutar las siguientes líneas en el terminal:

```
pip install djangorestframework
pip install djangorestframework-simplejwt
```

Una vez descargado este entorno de trabajo, y esta librería, éstas deberán añadirse a las aplicaciones que tendrá esta nueva aplicación en el archivo *settings.py* de la carpeta general del proyecto.

Una vez estas aplicaciones están añadidas al proyecto, se crearán las clases de registro de usuarios y de inicio de sesión, cuyo funcionamiento se ha descrito en la figura 12 para devolver a la aplicación paquetizada un JWT de acceso (para acceder a la plataforma sin necesidad de volver a autenticarse), y un JWT para actualizar el JWT de acceso. El código para ambas funciones se guardará en el archivo *views.py* de la aplicación de “clientes”, y es el siguiente:

```
1. class RegisterUserView(views.APIView):
2.     permission_classes = [permissions.AllowAny, ]
3.     def post(self, request):
4.
5.         email = request.data.get('email')
6.         username = request.data.get('username')
7.         password = request.data.get('password')
8.
9.         print(email, username, password)
10.
11.        #Crear el usuario
12.        try:
13.            user =
14. Cliente.objects.create_user(email=email, username=username, password=password)
15.            user2 = authenticate(email=email, password=password)
16.            #Generar el token
17.            refresh = RefreshToken.for_user(user2)
18.            return Response({
```

```
19.         'refresh':str(refresh),
20.         'access':str(refresh.access_token),
21.     },status=status.HTTP_201_CREATED)
22.     except:
23.         return Response(status=status.HTTP_400_BAD_REQUEST)
24.
25.
26.
27. class LoginView(views.APIView):
28.     permission_classes = [permissions.AllowAny, ]
29.
30.     def post(self,request):
31.
32.         email = request.data.get('email')
33.         password = request.data.get('password')
34.         user = authenticate(email=email, password=password)
35.
36.         if user is not None:
37.             #Generar el token
38.             refresh = RefreshToken.for_user(user)
39.             return Response({
40.                 'refresh': str(refresh),
41.                 'access': str(refresh.access_token),
42.             }, status=status.HTTP_200_OK)
43.         else:
44.             #No se ha encontrado el usuario
45.             return Response({"detail": "Invalid credentials"},
46. status=status.HTTP_401_UNAUTHORIZED)
```

Con este código, tanto al registrar a un usuario, como al iniciar sesión, se proporcionan JWTs actualizados a la aplicación paquetizada cuando registre nuevos usuarios en la plataforma de control o les inicie sesión.

Además, tenemos que gestionar si la aplicación paquetizada pide actualizar el JWT sin necesidad de volver a iniciar sesión, utilizando el JWT de actualización. Para ello, se creará el siguiente punto de acceso para gestionar el actualizado del JWT y devolver unos nuevos tokens de acceso y de actualización.

```
1. class CustomTokenRefreshView(TokenRefreshView):
2.     def post(self, request, *args, **kwargs):
3.         serializer = self.get_serializer(data=request.data)
4.         try:
5.             serializer.is_valid(raise_exception=True)
6.             return Response(serializer.validated_data, status=status.HTTP_200_OK)
7.         except:
8.             return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
9.
```

Una vez se reciban ambos tokens al iniciar sesión o al registrarse, la aplicación paquetizada deberá guardarlos en la base de datos para poder autenticar a ese usuario de nuevo en la aplicación de control sin enviar de nuevo el email y la contraseña. Para esto vamos a añadir los siguientes campos y funciones a la clase Usuario en el archivo *models.py* dentro de la aplicación o módulo de Autenticación del proyecto ya desarrollado:

```
1. #Campos nuevos JWT
2.     jwt_secret_access = models.CharField(max_length=250,blank=True)
3.     jwt_secret_refresh = models.CharField(max_length=250,blank=True)

4. #Funciones nuevas JWT
5.     def set_jwt_secret(self, jwt_access, jwt_refresh):
6.         self.jwt_secret_access = jwt_access
7.         self.jwt_secret_refresh = jwt_refresh
8.         self.save()
9.
10.
11.     def get_jwt_secret_access(self):
12.         return self.jwt_secret_access
13.
14.     def get_jwt_secret_refresh(self):
15.         return self.jwt_secret_refresh
16.
```

Una vez hayamos modificado la clase Usuario, modificaremos, en el archivo *views.py* de la aplicación de Autenticación, las funciones de inicio de sesión (*login\_view* en el proyecto) y de registro (*registro* en el proyecto).

Con esta modificación, estas vistas registrarán o iniciarán sesión al usuario en primer lugar en la plataforma de control, y, una vez estén autenticados, se guardarán en la base de datos local, guardando con ellos los dos JWT (acceso y actualización) en la base de datos. A continuación se expone el código de estas dos funciones (mostrando en comentarios cuando empieza y cuándo acaba el nuevo código generado):

```
1. url_aplicacion_control = "http://localhost:8088/clients/"
2.
3. def registro(request):
4.     context = {}
5.     if request.POST: # POST request
6.         form = FormularioRegistro(request.POST)
7.
8.         if form.is_valid():
9.             email = form.cleaned_data['email']
```

```
10.         rawPassword = form.cleaned_data['password1']
11.         username = form.cleaned_data['username']
12.
13.         # Nuevo código para crear el usuario en el servidor de MLaaS
14.
15.         data = {
16.             "email": email,
17.             "username": username,
18.             "password": rawPassword
19.         }
20.         url = url_aplicacion_control+"registerUsuario"
21.         response = requests.post(url, json=data)
22.
23.         if response.status_code != 201:
24.             context['error'] = "Error al crear el usuario en el servidor de MLaaS"
25.             return render(request,"Register.html",context)
26.
27.         # Guardamos el usuario en la base de datos de Django local
28.         form.save()
29.
30.         usuario = authenticate(email=email,password= rawPassword)
31.
32.         # Guardamos el JWT en la base de datos de Django local
33.
34.         usuario.set_jwt_secret(response.json()["jwt_secret_access"],response.json()["jwt_secret_refresh"])
35.         usuario.save()
36.
37.         # Fin del nuevo código
38.
39.         login(request,usuario)
40.         return redirect('/projects/')
41.     else:
42.         context['form']=form
43.     else: # GET request
44.         form = FormularioRegistro()
45.         context['form']=form
46.         return render(request,"Register.html",context)
47.
48. def login_view(request):
49.     context = {}
50.     user = request.user
51.     print(user)
52.     if user.is_authenticated:
53.         return redirect('/projects/')
54.
55.     if request.POST:
56.         form = FormularioAuthUsuario(request.POST)
57.         if form.is_valid():
58.             email = request.POST['email']
59.             password = request.POST['password']
60.
61.             #nuevo código para autenticar al usuario en el servidor de MLaaS
62.             data = {
63.                 "email": email,
64.                 "password": password
65.             }
66.
67.             url = url_aplicacion_control+"loginUsuario"
68.             response = requests.post(url, json=data)
69.
```

```
70.         if response.status_code != 201 and response.status_code != 200:
71.             context['error'] = "Error al crear el usuario en el servidor de MLaaS"
72.             return render(request,"Register.html",context)
73.
74.
75.         user = authenticate(email = email, password =password)
76.
77.         # Guardamos el JWT en la base de datos de Django local
78.
79.         user.set_jwt_secret(response.json()["jwt_secret_access"],response.json()["jwt_secret_refresh"]
80.         )
81.         user.save()
82.
83.         # Fin del nuevo código
84.
85.         if user:
86.             login(request,user)
87.             return redirect('/projects/')
88.         else:
89.             form = FormularioAuthUsuario()
90.
91.         context['form'] = form
92.         return render(request,"Login.html",context)
```

A continuación, deberemos modificar cada una de las acciones en las que se va a informar a la plataforma de control añadiendo el siguiente trozo de código (en este caso se ha añadido en la acción de realizar predicciones.

```
1.         # Informar al servidor de control MLaaS de la predicción
2.
3.         # obtener la cantidad de datos predichos
4.         cantidad_de_datos_predichos = len(prediccion)
5.
6.         # obtener el JWT del usuario
7.         jwt_secret_access = request.user.get_jwt_secret_access()
8.
9.         # enviar la petición al servidor de control MLaaS
10.        headers = {
11.            'Authorization': 'Bearer ' + jwt_secret_access,
12.        }
13.        data = {
14.            'Prediction_length': cantidad_de_datos_predichos,
15.        }
16.        url = url_aplicacion_control+"predictData"
17.        response = requests.post(url, headers=headers, json=data)
18.
19.        if response.status_code == 402:
20.            context['error'] = "No hay suficientes créditos para realizar la
predicción"
21.            return render(request,"error.html",context)
22.
23.        if response.status_code != 200:
24.
25.            jwt_secret_refresh = request.user.get_jwt_secret_refresh()
26.            data = {
```

```
27.         'refresh': jwt_secret_refresh,
28.     }
29.     url = url_aplicacion_control+"refreshToken"
30.     response = requests.post(url, json=data)
31.
32.     if response.status_code != 200:
33.         context['error'] = "Error al obtener el JWT del servidor de MLaaS"
34.         return render(request,"error.html",context)
35.
36.         # Guardamos el JWT en la base de datos de Django local
37.
request.user.set_jwt_secret(response.json()["jwt_secret_access"],response.json()["jwt_secret_r
efresh"])
38.         request.user.save()
39.
40.
41.
42.         # Reintentar informar al servidor de control MLaaS de la predicción
43.
44.         # obtener la cantidad de datos predichos
45.         cantidad_de_datos_predichos = len(prediccion)
46.
47.         # obtener el JWT del usuario
48.         jwt_secret_access = request.user.get_jwt_secret_access()
49.
50.         # enviar la petición al servidor de control MLaaS
51.         headers = {
52.             'Authorization': 'Bearer '+ jwt_secret_access,
53.         }
54.         data = {
55.             'Prediction_length': cantidad_de_datos_predichos,
56.         }
57.         url = url_aplicacion_control+"predictData"
58.         response = requests.post(url, headers=headers, json=data)
59.
60.         if response.status_code == 402:
61.             context['error'] = "No hay suficientes créditos para realizar la
predicción"
62.             return render(request,"error.html",context)
63.
64.         if response.status_code != 200:
65.             context['error'] = "Error al informar al servidor de MLaaS de la
predicción"
66.             return render(request,"error.html",context)
67.
68.
69.         # Continuar con el proceso (en este caso de predicción devolviéndoselo al
usuario)
70.
```

Con este código en primer lugar calculamos la cantidad de datos en los que se va a realizar la predicción. A continuación obtenemos de la base de datos de clientes en local el token de acceso que autentica al usuario frente a la plataforma de control. Una vez se obenga el token de acceso informamos a la aplicación de control indicando tanto el token, como el número de observaciones a predecir. Si la plataforma de control confirma que hay fondos suficientes

en la cuenta del usuario (con un código 200), la aplicación realiza la tarea devolviéndosela al usuario.

No obstante, si la plataforma de control encuentra que el usuario no tiene fondos suficientes (devolviendo el código 402), la aplicación paquetizada informa al usuario que debe ingresar más dinero en la plataforma de control pues los fondos disponibles no son suficientes para realizar la tarea (en este caso la predicción de esos datos). Finalmente, la plataforma puede devolver que el usuario no ha podido ser autenticado, pues es posible que el token de acceso haya caducado. En este caso, la aplicación paquetizada solicita que se actualicen los tokens enviando el token de actualización (*refresh token*), guarda los tokens devueltos por la plataforma de control, y reintenta la petición para realizar la tarea.

Por parte de la plataforma de control, solo tenderemos que crearnos funciones que comprueben el saldo, y lo actualicen en la base de datos de control. El siguiente código corresponde a la tarea de realizar predicciones.

```
1. def realizarPrediccionUsuario(request):
2.     if request.method == 'POST':
3.         try:
4.             user = request.user
5.             if user is None:
6.                 return Response(status=status.HTTP_401_UNAUTHORIZED)
7.             if
user.comprobar_saldo(num_predicciones=request.data.get('Prediction_length')) == False:
8.                 return Response(status=status.HTTP_402_PAYMENT_REQUIRED)
9.
10.
user.aumentar_num_predicciones_realizadas_ultimo_mes(request.data.get('Prediction_length'))
11.         user.disminuir_saldo(request.data.get('Prediction_length'))
12.         user.save()
13.         return Response(status=status.HTTP_200_OK)
14.     except:
15.         return Response(status=status.HTTP_400_BAD_REQUEST)
16.     return Response(status=status.HTTP_400_BAD_REQUEST)
17.
```

Django automáticamente con cada petición intenta iniciar sesión a los usuarios leyendo el token de acceso en la cabecera del mensaje. Esta función en primer lugar comprueba que el token de acceso corresponda a un usuario y tenga la sesión iniciada, y en caso contrario devuelve que no está autorizado el usuario en la plataforma. Posteriormente, comprueba que

tenga saldo disponible para realizar tal acción (en este caso comprobamos que tenga saldo disponible para realizar ese número de predicciones), y en caso la plataforma devuelve que falta saldo mediante el código 402. Finalmente, si el usuario está autenticado y tiene saldo, se sustrae dicho saldo, guardándose esta información en la base de datos, y se devuelve un OK.

Con estas líneas de código, insertadas en todas las tareas que necesitan del control de la plataforma web, como el entrenamiento de modelos predictivos o la predicción sobre nuevos conjuntos de datos, se garantiza la conectividad entre ambas plataformas.

### **5.5.3 INFORMAR A NUESTROS CLIENTES DE LA NECESIDAD DE TENER CONSENTIMIENTO PARA APLICAR ALGORITMOS PREDICTIVOS**

Para que nuestros clientes (PYMES y departamentos de grandes empresas) cumplan con la GDPR, deberán informar a los usuarios que se va a tratar su información personal e identificable para extraer conocimiento, o simplemente para mejorar el servicio.

Para garantizar que nuestros clientes cumplan con la GDPR, deberemos modificar la forma en la que se crean proyectos de machine learning (momento en el cual los usuarios suben el conjunto de datos con el que se entrenarán los modelos predictivos) para solicitar que el cliente confirme que se ha informado a todos los usuarios de que sus datos van a ser tratados.

Para ello, modificaremos el formulario con el que creamos los proyectos, añadiendo una nueva entrada con el formato “checkbox” para que el usuario confirme que cumple con la Ley Europea de Protección de Datos, ofreciéndole un link a la misma por si quiere inspeccionarla más a fondo.

El formulario para crear proyectos se ha desarrollado como un Formulario Django, que automáticamente crea un Proyecto tras validar automáticamente todos los valores introducidos en el formulario. Para modificarlo, en primer lugar tendremos que añadir un atributo a la clase de Proyecto con el nombre “compliance\_GDPR”, y posteriormente

añadiremos este campo al Formulario de Django y validaremos que tenga que tener un valor de Verdadero.

## CREA UN NUEVO PROYECTO

Para crear un nuevo proyecto deberás subir un archivo con datos (csv,xlsx,...) y darle un nombre al proyecto

Subir el documento

ProjectName:

ArchivoDatos:  No se ha seleccionado ningún archivo

Compliance GDPR:

[Link a la Ley Europea de Protección de Datos](#)

*Figura 15. Imagen de la nueva forma de crear proyectos, incluyendo el checkbox para comprobar el compliance con GDPR y el link a dicha ley*

### 5.5.4 GARANTIZAR EL BORRADO DE CLIENTES Y LA DESCARGA DE SU INFORMACIÓN PARA CUMPLIR CON LA LEY

La última tarea que se deba realizar antes de paquetizar la aplicación es la de ofrecer a los usuarios tanto eliminar su cuenta y todos los datos relacionados con la misma, como poder descargarse todos sus datos de los que dispone la aplicación.

#### 5.5.4.1 Borrado de clientes

En primer lugar, en la aplicación paquetizada, dentro de la aplicación de Autenticación (la que gestiona a los clientes) crearemos una nueva vista para borrar los usuarios y todos sus documentos relacionados. Además, esta vista integra el código explicado anteriormente para notificar a la plataforma de control que también borre todos los datos del usuario, y una vez ésta confirme que estos datos han sido borrados, se borran también en la base de datos local.

```
1. def eliminar_usuario(request):
2.     user = request.user
3.     if not user.is_authenticated:
4.         return redirect('/index.html')
5.
6.     context = {}
7.
8.
9.
10.    # Informar al servidor de control MLaaS del borrado
11.
12.    # obtener el JWT del usuario
13.    jwt_secret_access = request.user.get_jwt_secret_access()
14.
15.    # enviar la petición al servidor de control MLaaS
16.    headers = {
17.        'Authorization': 'Bearer ' + jwt_secret_access,
18.    }
19.
20.    url = url_aplicacion_control+"deleteUser"
21.    response = requests.post(url, headers=headers)
22.
23.
24.    if response.status_code != 200:
25.
26.        jwt_secret_refresh = request.user.get_jwt_secret_refresh()
27.        data = {
28.            'refresh': jwt_secret_refresh,
29.        }
30.        url = url_aplicacion_control+"refreshToken"
31.        response = requests.post(url, json=data)
32.
33.        if response.status_code != 200:
34.            context['error'] = "Error al obtener el JWT del servidor de MLaaS"
35.            return render(request,"error.html",context)
36.
37.        # Guardamos el JWT en la base de datos de Django local
38.
request.user.set_jwt_secret(response.json()["jwt_secret_access"],response.json()["jwt_secret_r
efresh"])
39.        request.user.save()
40.
41.
42.        # Reintentar informar al servidor de control MLaaS del borrado
43.
44.        # obtener el JWT del usuario
45.        jwt_secret_access = request.user.get_jwt_secret_access()
46.
47.        # enviar la petición al servidor de control MLaaS
48.        headers = {
49.            'Authorization': 'Bearer ' + jwt_secret_access,
50.        }
51.
52.        url = url_aplicacion_control+"deleteUser"
53.        response = requests.post(url, headers=headers)
54.
55.        if response.status_code != 200:
56.            context['error'] = "Error al informar al servidor de MLaaS de la predicción"
57.            return render(request,"error.html",context)
58.
59.        # Ya se han borrado todos los datos del usuario en el servidor de control
60.        # Procedemos a borrar todos los proyectos en la base de datos local
```

```
61.
62.     lista_proyectos_a_eliminar = user.projects.all()
63.
64.     for proyecto in lista_proyectos_a_eliminar:
65.         proyecto.project.borrar_todos_archivos_vinculados()
66.         proyecto.delete()
67.
68.     # Finalmente borramos el usuario de la base de datos local
69.     user.delete()
70.     return redirect('/index.html')
71.
```

En la aplicación de control también tendremos que desarrollar una función que borre los usuarios cuando se lo notifica la aplicación que se ejecuta en el ordenador del cliente. De nuevo, como se está utilizando un entorno de trabajo de JWT, éste automáticamente registra a los usuarios si en la cabecera de los mensajes enviados se envía el token de acceso. La función comprueba que el usuario esté autenticado, y elimina al usuario de la base de datos.

```
1. def eliminar_usuario(request):
2.     if request.method == 'POST':
3.         try:
4.             user = request.user
5.             user.delete()
6.             return Response(status=status.HTTP_200_OK)
7.         except:
8.             return Response(status=status.HTTP_400_BAD_REQUEST)
9.     return Response(status=status.HTTP_401_UNAUTHORIZED)
10.
```

#### 5.5.4.2 Obtención de los datos guardados de los clientes

Los únicos datos que esta empresa dispondría de los clientes están ubicados en la aplicación de control, pues la aplicación paquetizada guarda los datos en el propio ordenador del cliente.

En primer lugar, crearemos una función en la clase Cliente que devuelva todos los datos relacionados con un usuario.

```
1. def obtener_todos_datos(self):
2.     return {
3.         "email":self.email,
4.         "username":self.username,
5.         "date_joined":self.date_joined,
6.         "last_login":self.last_login,
7.         "is_admin":self.is_admin,
8.         "is_active":self.is_active,
9.         "is_staff":self.is_staff,
```

```

10.         "is_superuser":self.is_superuser,
11.         "saldo":self.saldo,
12.         "num_proyectos_machine_learning":self.num_proyectos_machine_learning,
13.
14.         "num_modelos_machine_learning_entrenados_ultimo_mes":self.num_modelos_machine_learning_entrena
dos_ultimo_mes,
15.         "num_predicciones_realizadas_ultimo_mes":self.num_predicciones_realizadas_ultimo_mes,
16.     }

```

Posteriormente tenemos que crear una vista en la plataforma de control que devuelva esta información a la aplicación a paquetizar.

```

1. def descargar_datos_de_usuario(request):
2.     if request.method == 'POST':
3.         try:
4.             user = request.user
5.             datos_usuario = user.obtener_todos_datos()
6.             return Response(data=datos_usuario,
7.                             status=status.HTTP_200_OK)
8.         except:
9.             return Response(status=status.HTTP_400_BAD_REQUEST)
10.    return Response(status=status.HTTP_401_UNAUTHORIZED)

```

Finalmente, tendremos que crear una vista en la aplicación paquetizada, en la aplicación Autenticación, que recoja la petición del cliente de descargar los datos que guardamos de él, realice la petición a la aplicación de control, y posteriormente devuelva esta información en un Excel, para que el usuario pueda visualizarla fácilmente.

```

1. def obtener_datos_relacionados_con_cliente(request):
2.     user = request.user
3.     if not user.is_authenticated:
4.         return redirect('/index.html')
5.     context = {}
6.
7.     # Informar al servidor de control de la petición de datos del usuario
8.
9.     # obtener el JWT del usuario
10.    jwt_secret_access = request.user.get_jwt_secret_access()
11.    # enviar la petición al servidor de control MLaaS
12.    headers = {
13.        'Authorization': 'Bearer '+ jwt_secret_access,
14.    }
15.    url = url_aplicacion_control+"getDataFromUser"
16.    response = requests.post(url, headers=headers)
17.
18.    if response.status_code != 200:
19.        jwt_secret_refresh = request.user.get_jwt_secret_refresh()
20.        data = {

```

```

21.         'refresh': jwt_secret_refresh,
22.     }
23.     url = url_aplicacion_control+"refreshToken"
24.     response = requests.post(url, json=data)
25.     if response.status_code != 200:
26.         context['error'] = "Error al obtener el JWT del servidor de MLaaS"
27.         return render(request,"error.html",context)
28.         # Guardamos el JWT en la base de datos de Django local
29.
request.user.set_jwt_secret(response.json()["jwt_secret_access"],response.json()["jwt_secret_r
efresh"])
30.     request.user.save()
31.     # Reintentar informar al servidor de control MLaaS del borrado
32.     # obtener el JWT del usuario
33.     jwt_secret_access = request.user.get_jwt_secret_access()
34.     # enviar la petición al servidor de control MLaaS
35.     headers = {
36.         'Authorization': 'Bearer '+ jwt_secret_access,
37.     }
38.     response = requests.post(url, headers=headers)
39.     if response.status_code != 200:
40.         context['error'] = "Error al informar al servidor de MLaaS de la predicción"
41.         return render(request,"error.html",context)
42.
43.     # Ya se han descargado los datos del usuario en el servidor de control
44.
45.     dataframe_datos_descargados = pd.DataFrame(response.json())
46.
47.
48.     response = HttpResponse(content_type='application/ms-excel')
49.     response['Content-Disposition'] = 'attachment; filename="datos_usuario.xlsx"'
50.
51.     # write the DataFrame to the response object as an Excel file
52.     dataframe_datos_descargados.to_excel(response, index=False)
53.
54.     return response
55.

```

Con este código, el usuario accediendo a esta vista automáticamente se descargan los datos de los que dispone la aplicación sobre el cliente.

### 5.5.5 PAQUETIZACIÓN DE LA APLICACIÓN

Para paquetizar la aplicación que entrena modelos de machine learning de una manera fácil e intuitiva utilizaremos la herramienta *pyinstaller*, que instalaremos mediante el siguiente comando en el terminal:

```
pip install pyinstaller
```

A continuación crearemos un nuevo fichero de Python en la carpeta del proyecto llamado *runserver.py* que se encargará de iniciar el servidor de esta aplicación web en el ordenador

de nuestros clientes a la vez que también de abrir el navegador de los mismos para que puedan utilizar la plataforma en la dirección en la que se ejecuta, que es <http://localhost:8000/>.

```
1. import os
2. import sys
3. import webbrowser
4. import threading
5. from django.core.management import execute_from_command_line
6.
7. def open_browser():
8.     webbrowser.open_new("http://localhost:8000/")
9.
10. if __name__ == "__main__":
11.     os.environ.setdefault("DJANGO_SETTINGS_MODULE", "TFGTelecoMLaaS.settings")
12.
13.     # Empezamos el servidor de Django en un nuevo hilo
14.     timer = threading.Timer(2, open_browser)
15.     timer.start()
16.
17.     execute_from_command_line(["runserver.py", "runserver"])
18.
```

Para crear el paquete que se descargarán nuestros clientes con la herramienta *pyinstaller*, debemos introducir el siguiente comando donde indicamos a la herramienta *pyinstaller* la dirección de la plataforma, junto con el archivo que tiene que ejecutarla (*runserver.py*):

```
pyinstaller --onefile --add-data
"LandingPage/templates;LandingPage/templates" --add-data
"LandingPage/static;LandingPage/static" --add-data
"AnalizarDatos/templates;AnalizarDatos/templates" --add-data
"Autenticacion/templates;Autenticacion/templates" --add-data
"CrearAbrirProyectos/templates;CrearAbrirProyectos/templates" --
add-data "CrearModelos/templates;CrearModelos/templates" --add-
data "Prediccion/templates;Prediccion/templates" --add-data
"VisualizarModelos/templates;VisualizarModelos/templates" --add-
data "db.sqlite3;." runserver.py
```

Con este código, la herramienta *pyinstaller* nos crea un ejecutable llamado *runserver.exe*, que, simplemente haciendo doble clic en él, inicia el servidor y automáticamente abre el navegador con la dirección <http://localhost:8000/> para que los clientes puedan utilizar esta herramienta.

Con todos estos cambios realizados (creación de una nueva plataforma, conexión segura entre la aplicación a paquetizar y la nueva plataforma, informar a los clientes de la necesidad de cumplir con GDPR, permitir descargar los datos de usuario de los que dispone la aplicación, y finalmente paquetizar la aplicación para que se ejecute en los ordenadores de nuestros clientes) la aplicación cumple con la regulación GDPR, y es viable su comercialización.

## **Capítulo 6. MODELO DE NEGOCIO**

### **6.1 CAMBIO DE MODELO DE NEGOCIO**

Tras adaptarse a la legislación vigente, la nueva plataforma se ha transformado por completo, alejándose de la propuesta inicial y de los competidores identificados. Mientras que la versión anterior de la plataforma se basaba en servidores web y estaba orientada a usuarios sin experiencia, la versión actual se ejecuta en los ordenadores de los propios clientes, utilizando sus procesadores para entrenar y ejecutar los modelos predictivos. Esto implica un cambio en el modelo de negocio, ya que, en lugar de cobrar por tiempos de ejecución, se opta por cobrar a los clientes por el valor añadido que obtienen al poder extraer conocimiento de sus datos. La generación de modelos predictivos y, sobre todo, la realización de predicciones sobre nuevos datos son los aspectos más valiosos para los usuarios, por lo que la nueva plataforma se enfocará en cobrar por la creación de modelos predictivos y las predicciones sobre los nuevos datos.

### **6.2 FACTORES DIFERENCIADORES**

Los dos mayores factores diferenciadores que tiene esta nueva plataforma comparándola con los competidores actuales, es que está pensado y desarrollado especialmente para los usuarios inexpertos, que en este caso es la gran mayoría de la población; y que es compatible con la actual estricta regulación GDPR, aspecto que no está muy claro en el caso de los otros competidores.

La primera diferencia, la facilidad para utilizarlo por usuarios inexpertos, nos va a poder ayudar a comercializar este negocio en las pequeñas y medianas empresas en nuestro país. Con este negocio las empresas no necesitan realizar cursos preparatorios para utilizar la aplicación, sino que pueden utilizarla directamente. Esto ahorra a las PYMES recursos económicos, tanto de forma directa por no tener que comprar estos cursos para utilizar la

aplicación, como de forma indirecta por el tiempo que se ahorran las empresas en entrenar a sus trabajadores en esta herramienta.

La segunda diferencia, el cumplimiento de la GDPR, da a esta empresa una ventaja competitiva muy significativa frente a los competidores identificados. Si una empresa decidiese optar por contratar los servicios de un competidor, como puede ser Microsoft Azure, tendría que prescindir de todos los datos que ya tiene recopilados. Esto es debido a que no habría informado a los usuarios cuyos datos han sido recopilados que aceptan mandar su información a terceras empresas para que sean tratados (como se ha explicado en el capítulo 5.2). Además, estas empresas deberían ser muy cautas de que los datos de sus usuarios no salgan a servidores ubicados fuera de la Unión Europea, puesto que esto vulneraría la GDPR. Las PYMES deben tener muy en cuenta a la ley Europea de Protección de Datos por sus altas multas, en las que las infracciones leves conllevan multas de hasta 10 millones de euros o un 2% de la facturación anual de la empresa (la mayor de las dos); y las infracciones graves conllevan multas de hasta 20 millones de euros o un 4% de la facturación anual de la empresa (también la mayor de las dos) (Gallardo, 2018). Debido a esto, este cumplimiento de la ley GDPR es un factor diferenciador importante frente a las grandes tecnológicas.

Aparte de estas dos grandes diferencias con nuestros competidores, el uso de esta plataforma es muy sencillo. La instalación de la aplicación es fácil e intuitiva, puesto que la plataforma corre en el ordenador del cliente y solo necesita que el usuario le dé doble clic a la aplicación paquetizada para que se encienda. Además, la conexión con la aplicación de control se realiza automáticamente sin que el usuario tenga que intervenir en el proceso.

Por estas diferencias con los competidores identificados nuestros clientes potenciales son las PYMES puesto que, como se ha dicho en el capítulo 2, la mayoría no implementa técnicas de machine learning pese a que su uso les ayudaría en mejorar sus cifras de negocio, y existen incentivos gubernamentales para que las pequeñas y medianas empresas se digitalicen utilizando estas novedosas técnicas de machine learning.

### **6.3 ACCIONES DE MARKETING**

Para dar a conocer a esta nueva plataforma a sus clientes potenciales, se va a realizar las siguientes acciones de marketing, centrándose especialmente en los aspectos que nos diferencian de nuestros competidores.

- **Posicionamiento SEO (Search Engine Optimization):** para que las PYMES puedan encontrar a esta empresa en los buscadores web vamos a crear una página web con los siguientes atributos:
  - Una Landing Page sencilla y atractiva que tenga una interfaz fácil de ser navegada para los usuarios y así estar priorizados en mayor medida por buscadores (Shenoy & Prabhu, 2016).
  - A la hora de diseñar esta página web, se tendrá en cuenta las palabras clave (Yalçın & Köse, 2010) que pueden utilizar nuestros potenciales clientes para que nos encuentren. Estas palabras, como “digitalizar mi empresa”, “utilizar datos CRM”, o “machine learning para PYMES”, se utilizarán en esta página web para conseguir aparecer, de forma orgánica, más alto en las búsquedas.
  - Se obtendrá referencias cruzadas desde medios de comunicación para aumentar la optimización SEO (Zhang & Cabage, 2017). Concediendo entrevistas en medios de comunicación sobre esta novedosa plataforma, habrá referencias desde estos periódicos a nuestra página web, lo que mejorará el SEO de la página web.
  - Finalmente se optimizará la página web técnicamente (Arias, 2013) para mejorar la rapidez de carga, que sea accesible para todo tipo de usuarios (aportando descripción de las imágenes insertadas en la página), permitiendo que pueda ser legible en teléfonos móviles, y finalmente creando una lista con las páginas web de la empresa (XML Sitemap en inglés).
  
- **Participación en ferias de PYMES:** se participará en ferias tanto de PYMES como de empresas nacientes (startups) para potenciar esta plataforma entre nuestros

posibles clientes. Ejemplos de estas ferias son: E-Show Barcelona, CYL-HUB Segovia, FAME INNOWA, BeDIGITAL, IOT Solutions World Congress, Madrid Platform y Forinvest Valencia.

- **Fondos gubernamentales de digitalización:** otra acción de marketing sería la de conseguir la aprobación de proyecto financiado por los fondos gubernamentales en favor de la digitalización por parte del Ministerio de Asuntos Económicos y Transformación Digital. Con esta aprobación, podremos comercializar más fácilmente nuestro producto a las PYMES, puesto que su coste será parcial o totalmente subsanado por subvenciones gubernamentales. Además, para financiarnos, intentaremos conseguir financiación de los fondos ENISA y así poder hacer frente a costes como estas acciones de marketing o a los gastos administrativos.
- **Potenciar el boca-oreja (referrals con descuento):** se potenciará el boca-oreja ofreciendo descuentos tanto para los nuevos clientes que entren en la plataforma recomendados (mediante un código de referencia) por usuarios ya presentes en la plataforma como para los clientes que les han recomendado. Así, los nuevos clientes se convertirán en embajadores de la nueva plataforma intentando recomendarla a compañeros de profesión. Tanto al nuevo cliente como al antiguo que recomienda la plataforma se les aplicará un descuento de un 50% en su factura en los próximos 6 meses.

Además, el ingreso medio por usuario, el churn y el CAC son los KPIs más importantes que se van a monitorizar para guiar las acciones de marketing.

## **6.4 VALORACIÓN**

### **6.4.1 PRECIO DE LA PLATAFORMA Y TAMAÑO DE MERCADO**

A la hora de establecer el precio de la plataforma, deberemos tener en cuenta el uso medio de la plataforma y los costes de nuestros competidores para que el coste que represente esta plataforma a nuestros clientes sea inferior a la de nuestros competidores.

Analizando los costes de nuestros competidores, como se vio en la tabla de comparación de competidores, el competidor que ofrecía un menor coste era Microsoft Azure. Este competidor, para la especificación base de 8 Gb de RAM y 4 CPUs, similar a la de un portátil base con procesador Intel core i5 (la última generación es el i9), factura al mes a las compañías cerca de 82€ por esa configuración base (<https://azure.microsoft.com/en-us/pricing/details/machine-learning/>).

Para poder ser competitivos, podemos intentar facturar a nuestros clientes 50€ mensuales, un 39% inferior al mejor precio en el mercado actual. No obstante, como cobraremos por uso, se monitorizará el uso que se le dé a la plataforma por parte de nuestros clientes para encontrarnos por debajo del precio de mercado y poder crecer con rapidez.

Para dar datos sobre los precios del uso de la plataforma, necesitamos conocer el uso medio que harían nuestros clientes. Pero, dada la opacidad del mercado no disponemos de cifras de las bases de datos de las empresas o del uso que den éstas, como para dar precios cerrados que representen de media estos 50€.

Si asumimos que una empresa media que contrate esta plataforma tenga una base de datos de clientes o posibles clientes de 2500, que prediga sobre esta base de manera mensual y que vuelva a entrenar todos los modelos predictivos cada 2 meses; a los dos meses esta persona estaría entrenando 6 modelos predictivos (obviando los Ensembles) con 2500 observaciones, y realizando predicciones sobre un total de 5000 observaciones.

Para poder facturar los 50€ mensuales según el uso medio mencionado anteriormente, la empresa cobrará a sus usuarios:

- 5€ por cada modelo entrenado
- 0.014€ por cada predicción realizada

De esta manera, estaremos cobrando a las empresas por debajo del precio de mercado. Además, hay que recalcar que, aunque se vaya a facturar esta cantidad de 50€ mensuales a los clientes, a las compañías pueden repercutirle una cantidad muy inferior por las subvenciones gubernamentales en favor de la digitalización que se esperan conseguir.

Así, este ingreso medio por usuario de 50€ mensual se traduce en 600€ de ingreso medio anual (average revenue per user).

El público objetivo son las pequeñas y medianas empresas nacionales. El número total de pequeñas y medianas empresas según el INE (2022) es de 3.425.943 sociedades. No obstante, si se obvian las empresas sin trabajadores puesto que es difícil que éstas tengan datos suficientes como para utilizar la plataforma, y solo se tiene en cuenta las empresas con trabajadores, el número de empresas sería de 1.483.624 sociedades.

Uniando el número de empresas y el ingreso medio esperado por cada una de ellas, se obtiene un tamaño de mercado anual de 890.174.400€.

#### **6.4.2 VALORACIÓN DE LA EMPRESA POR USUARIO**

Para valorar la empresa en función de los usuarios, se necesita calcular el valor del tiempo de vida del cliente (Customer Lifetime Value o CLTV en inglés), junto al coste de adquisición de nuevos clientes (Customer Acquisition Cost o CAC en inglés).

Para calcular el CLTV, tendremos que considerar la facturación media anual a los clientes, el crecimiento de los ingresos por usuario, los costes medios de explotación, la duración máxima de vida del cliente, el porcentaje de clientes que nos abandonan cada año (churn rate), y la tasa a la que se descuentan estos ingresos futuros.

Se asume una vida máxima de cliente de 15 años, siguiendo el ejemplo de Damodaran cuando analiza empresas de suscriptores como Netflix, Amazon, y Uber (Damodaran, 2018).

	Años														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Ingresos</b>	<b>600 €</b>	<b>660 €</b>	<b>726 €</b>	<b>799 €</b>	<b>878 €</b>	<b>966 €</b>	<b>1.015 €</b>	<b>1.065 €</b>	<b>1.119 €</b>	<b>1.175 €</b>	<b>1.233 €</b>	<b>1.252 €</b>	<b>1.271 €</b>	<b>1.290 €</b>	<b>1.309 €</b>
Crecimiento de los ingresos		10%	10%	10%	10%	10%	5%	5%	5%	5%	5%	1,50%	1,50%	1,50%	1,50%
Costes de explotación	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €
Resultado por usuario	600 €	660 €	726 €	799 €	878 €	966 €	1.015 €	1.065 €	1.119 €	1.175 €	1.233 €	1.252 €	1.271 €	1.290 €	1.309 €
Probabilidad de supervivencia	100%	86%	74%	64%	55%	47%	40%	35%	30%	26%	22%	19%	16%	14%	12%
<b>Resultado esperado</b>	<b>600 €</b>	<b>568 €</b>	<b>537 €</b>	<b>508 €</b>	<b>481 €</b>	<b>455 €</b>	<b>410 €</b>	<b>371 €</b>	<b>335 €</b>	<b>302 €</b>	<b>273 €</b>	<b>238 €</b>	<b>208 €</b>	<b>182 €</b>	<b>158 €</b>
Tasa de impuestos	30%														
Resultado después de impuestos	420 €	397 €	376 €	356 €	336 €	318 €	287 €	259 €	234 €	212 €	191 €	167 €	146 €	127 €	111 €
Tasa de descuento	25%														
Resultados descontados	336 €	254 €	192 €	146 €	110 €	83 €	60 €	44 €	31 €	23 €	16 €	11 €	8 €	6 €	4 €
<b>CLTV</b>	<b>1.325,32 €</b>														
CAC															
Boca oreja	300,00 €														
2.5x media de damodaran	318,78 €														
<b>CAC Suma</b>	<b>618,78 €</b>														
<b>Valor Actual Neto</b>	<b>706,54 €</b>														

Tabla 2. Valor por usuario de la empresa Mach IA

La facturación media anual a nuestros clientes es de 600€, y como los modelos se entrenan en los ordenadores de nuestros clientes, los gastos de explotación son nulos, por lo que se obtiene un resultado por usuario medio anual de 600€ el primer año.

A partir de entonces, se asume una tasa de crecimiento de facturación por usuario entre el segundo año y el sexto de un 10%, bajando hasta el 5% entre el séptimo año y el undécimo y reduciéndose hasta el 1.5% en el año duodécimo hasta el decimoquinto año.

Para calcular la probabilidad de supervivencia se tendrá en cuenta la probabilidad de abandono de los clientes (churn rate en inglés). Suponiendo que el churn rate es del 15%, un número razonable comparado con el de Uber 5%; Amazon 4%; Netflix, 7.5% (Damodaran, 2018). La probabilidad de supervivencia se irá reduciendo al churn rate de forma anual. Esta probabilidad de supervivencia se multiplica por el resultado por usuario, dando lugar al resultado anual esperado.

Tras descontar el impuesto de sociedades (30%), este resultado se descuenta a una tasa de descuento del 25% (tasa que utilizan comúnmente los fondos de Venture Capital), obteniendo que el CLTV es de 1.325,32 €.

No obstante, a esta cifra hay que descontarle el CAC, puesto que vamos a invertir en marketing para conseguir clientes. Como se ha comentado antes, se potenciará el boca-oreja mediante un descuento de un 50% en los siguientes 6 meses tanto a los usuarios nuevos como a los que recomienden la plataforma. Como el ingreso anual medio esperado es de 600€, el coste por cada adquisición de usuario es de 300€ que se materializa en 150€ de media tanto para el recomendador como para el recomendado. Además, a estos 300€ por usuario, vamos a sumarle un coste de marketing general por usuario, que será 2.5 veces la media de Uber, Amazon y Netflix según Damodaran (2018), 318,78€. Sumando ambas cantidades, obtenemos un CAC total de 618,78 €.

Finalmente, si al CLTV le restamos el coste de adquirir clientes (CAC), se obtiene el valor actual neto por usuario de 706,54 €.

Para calcular el valor de esta empresa, tendremos que descontar el valor añadido que aportarán los nuevos clientes que esperamos conseguir durante los primeros diez años, restándole el valor después de impuestos de los gastos generales (overhead expenses) en cada año. Finalmente, deberemos tener en cuenta el valor final (terminal value) al terminar el año 10.

En primer lugar, se ha supuesto que el primer año se obtendrán 100 clientes, que irán creciendo anualmente durante los primeros diez años hasta llegar al 6.6% del total de PYMES con trabajadores en España. A partir del décimo año, se asume un crecimiento a largo plazo del 2%, como el crecimiento esperado del mercado a largo plazo. Este valor es necesario para calcular el valor terminal de la empresa.

	Años									
	1	2	3	4	5	6	7	8	9	10
<b>Clientes</b>	<b>100</b>	<b>600</b>	<b>3.000</b>	<b>12.000</b>	<b>24.000</b>	<b>38.400</b>	<b>61.440</b>	<b>73.728</b>	<b>88.474</b>	<b>97.321</b>
Crecimiento anual		500%	400%	300%	100%	60%	60%	20%	20%	10%
Nuevos clientes	100	500	2.400	9.000	12.000	14.400	23.040	12.288	14.746	8.847
CLTV - CAC	706,54 €									
Value added by new users	70.654 €	353.270 €	1.695.697 €	6.358.862 €	8.478.483 €	10.174.180 €	16.278.687 €	8.681.967 €	10.418.360 €	6.251.016 €
After tax overhead expenses	12.000 €	12.000 €	50.000 €	50.000 €	100.000 €	300.000 €	500.000 €	700.000 €	700.000 €	1.000.000 €
Tasa de crecimiento LP	2,0%									
Terminal Value										23.287.114 €
Tasa de descuento	25%									
Valor descontado	46.923 €	218.413 €	842.597 €	2.584.110 €	2.745.461 €	2.588.457 €	3.309.031 €	1.339.152 €	1.304.376 €	3.064.258 €
<b>Valor de la empresa</b>	<b>18.042.778 €</b>									

*Tabla 3. Valoración de la empresa Mach IA, llegando a un 6.5% del mercado*

Teniendo en cuenta el valor añadido por los nuevos usuarios de cada año (CLTV), el valor terminal de la empresa, junto con los gastos generales (overhead expenses), y una tasa de descuento del 25%, obtenemos los valores actuales esperados que generarán los futuros usuarios a lo largo de la vida de la empresa, o el valor de la empresa misma. Asumiendo esos valores, el valor de la empresa es de 18 millones de euros.

No obstante, si conseguimos llegar a un 13.3% del total de PYMES con trabajadores en España, aumentando los gastos generales reflejando este crecimiento en clientes, y manteniendo el 2% de crecimiento a largo plazo a partir del décimo año, obtenemos un valor de la empresa de 34.5 millones de euros.

	Años									
	1	2	3	4	5	6	7	8	9	10
<b>Clientes</b>	<b>100</b>	<b>600</b>	<b>3.000</b>	<b>12.000</b>	<b>36.000</b>	<b>72.000</b>	<b>115.200</b>	<b>149.760</b>	<b>179.712</b>	<b>197.683</b>
Crecimiento anual		500%	400%	300%	200%	100%	60%	30%	20%	10%
Nuevos clientes	100	500	2.400	9.000	24.000	36.000	43.200	34.560	29.952	17.971
CLTV - CAC	706,54 €									
Value added by new users	70.654 €	353.270 €	1.695.697 €	6.358.862 €	16.956.966 €	25.435.449 €	30.522.539 €	24.418.031 €	21.162.294 €	12.697.376 €
After tax overhead expenses	12.000 €	12.000 €	50.000 €	50.000 €	100.000 €	700.000 €	1.400.000 €	1.500.000 €	1.700.000 €	2.000.000 €
Tasa de crecimiento LP	2,0%									
Terminal Value										47.440.538 €
Tasa de descuento	25%									
Valor descontado	46.923 €	218.413 €	842.597 €	2.584.110 €	5.523.691 €	6.484.250 €	6.107.439 €	3.845.008 €	2.612.185 €	6.242.511 €
<b>Valor de la empresa</b>	<b>34.507.126 €</b>									

*Tabla 4. Valoración de la empresa Mach IA, llegando a un 13.3% del mercado*

## Capítulo 7. CONCLUSIÓN

En este proyecto se ha desarrollado una plataforma para solucionar el problema del acceso de las pequeñas y medianas empresas a las capacidades predictivas de los modelos de machine learning.

Para solucionar la baja utilización de las técnicas predictivas propias del machine learning que están haciendo las PYMES en España, cuyo uso ayudaría en gran medida a estas empresas, mejorando su rendimiento, relación con clientes y aumentando facturación, se ha utilizado una aplicación web como punto de partida, desarrollada por el autor de este mismo trabajo, la cual facilita la creación de modelos predictivos de forma sencilla. Esta plataforma ya desarrollada guía al usuario a lo largo de todas las etapas del proceso, ofreciendo recomendaciones inteligentes para la toma de decisiones en cada fase, lo que la hace idónea para usuarios inexpertos en machine learning como las PYMES.

En este proyecto se han realizado diversas modificaciones en la plataforma para asegurar el cumplimiento de la Ley Europea de Protección de Datos (GDPR por sus siglas en inglés). Para cumplir con GDPR, se ha optado por paquetizar la aplicación ya desarrollada con el objetivo de que esta aplicación se ejecute en los ordenadores de los clientes (las PYMES) y así solucionar el problema de cesión de los datos a otras entidades a la hora de extraer información de ellos. Además, se ha desarrollado una nueva aplicación web cuya funcionalidad será simplemente facturar a los clientes por el uso que le den a la aplicación de crear modelos predictivos y realizar predicciones, sin que esta aplicación tenga que recibir ningún dato personal, ni para entrenar los modelos de machine learning ni para generar predicciones. Por otro lado, se ha establecido la forma en la que la aplicación paquetizada se comunicará con la aplicación de control para garantizar que se monitorice el uso que el cliente hace de la plataforma, mientras se mantiene la seguridad autenticando esta comunicación entre ambas plataformas y así prevenir ataques de suplantación de identidad.

También, se ha modificado la aplicación paquetizada para garantizar que los clientes confirmen que, en la recopilación de los datos, se ha informado a los usuarios de los que se extraen los datos, que su información podría ser procesada. La última modificación que se ha realizado en la plataforma es añadir la posibilidad de la descarga de los datos del usuario de la plataforma, así como el borrado de los usuarios, para cumplir con la ley GDPR.

Para comercializar y promocionar esta aplicación, se han definido diversas acciones de marketing. Estas son: el posicionamiento SEO, la participación en ferias de PYMES, conseguir la aprobación subvenciones en favor de la digitalización de las empresas para nuestra plataforma, y finalmente, potenciar el boca-oreja ofreciendo descuentos en la plataforma tanto a los nuevos clientes como a los que la recomiendan.

Finalmente, mediante el método de valoración de empresas por usuario, se comprueba que es viable esta empresa ya que asumiendo un ingreso medio anual de 600€ obtenemos un CLTV de 1.325,32 €, que, descontando el CAC, nos deja un valor actual neto por usuario de 706,54 €. Con una penetración dentro de 10 años de un 6.6% del total de las PYMES con trabajadores, esta empresa valdría 18 millones de euros. No obstante, con una penetración de un 13.3% en 10 años, esta empresa valdría 34.5 millones de euros.

En conclusión, es viable la comercialización de este negocio que ayuda a las PYMES a extraer conocimiento de sus datos, generando modelos predictivos de una forma fácil y sencilla, tanto desde el punto de vista legal, como desde el punto de vista económico.

## Capítulo 8. BIBLIOGRAFÍA

- Albrecht, J. P. (2016). How the GDPR will change the world. *Eur.Data Prot.L.Rev.*, 2, 287.
- Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- Arias, M. A. (2013). *Marketing digital. posicionamiento SEO, SEM y redes sociales*. IT Campus Academy.
- Bessen, J. E., Impink, S. M., Reichensperger, L., & Seamans, R. (2020). GDPR and the importance of data to AI startups. *NYU Stern School of Business*,
- Blanco, M. (2020). *Las pymes también se pueden beneficiar del Big Data*. Think Big. <https://empresas.blogthinkbig.com/big-data-pymes/>
- California consumer privacy act of 2018, 1798.100, (2018). [https://leginfo.ca.gov/faces/codes\\_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5#content\\_anchor](https://leginfo.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5#content_anchor)
- Candelon, F., Carlo, R. C. d., Bondt, M. D., & Evgeniou, T. (2021, -09-01T04:00:00Z). AI regulation is coming. *Harvard Business Review*, <https://hbr.org/2021/09/ai-regulation-is-coming>
- Chimakurthi, V. N. S. S. (2021). Strategic growth of everything-as-a-service (XaaS) business model transformation. *Engineering International*, 9(2), 129-140.
- Chinnamgari, S. K. (2019). *R machine learning projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5*. Packt Publishing Ltd.
- Clark, J., & Perrault, R. (2023). *Artificial intelligence index report 2023*. Stanford University Human-Centered Artificial Intelligence. [https://aiindex.stanford.edu/wp-content/uploads/2023/04/HAI\\_AI-Index-Report\\_2023.pdf](https://aiindex.stanford.edu/wp-content/uploads/2023/04/HAI_AI-Index-Report_2023.pdf)
- Comisión Europea. (2020). On artificial intelligence—a european approach to excellence and trust. [https://commission.europa.eu/system/files/2020-02/commission-white-paper-artificial-intelligence-feb2020\\_en.pdf](https://commission.europa.eu/system/files/2020-02/commission-white-paper-artificial-intelligence-feb2020_en.pdf)
- Comisión Europea. (2021). 2030 digital compass: The european way for the digital decade. *COM (2021) 118 Final*,
- Comisión Europea. (2022). *Digital economy and society index (DESI)*. European Commission. <https://digital-strategy.ec.europa.eu/en/policies/desi>
- Damodaran, A. (2018). Going to pieces: Valuing users, subscribers and customers. *Subscribers and Customers (may 23, 2018)*,
- Data Bridge Market Research. (2021). *Machine learning as a service (MLaaS) market – global industry trends and forecast to 2028*. Data Bridge Market Research. <https://www.databridgemarketresearch.com/reports/global-machine-learning-service-mlaas-market>

- Data Bridge Market Research. (2022). *Spain machine learning as a service market – industry trends and forecast to 2029*. <https://www.databridgemarketresearch.com/reports/spain-machine-learning-as-a-service-market>
- Domo. (2018). Data never sleeps 6.0. Domo, 1. [https://www.domo.com/assets/downloads/18\\_domo\\_data-never-sleeps-6+verticals.pdf](https://www.domo.com/assets/downloads/18_domo_data-never-sleeps-6+verticals.pdf)
- Eurostat. (2023). *E-commerce statistics for individuals*. [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=E-commerce\\_statistics\\_for\\_individuals](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=E-commerce_statistics_for_individuals)
- Fatouretchi, M. (2019). *The art of CRM: Proven strategies for modern customer relationship management*. Packt Publishing Ltd.
- Forcier, J., Bissex, P., & Chun, W. J. (2008). *Python web development with django*. Addison-Wesley Professional.
- Fortune Business Insights. (2022). *Everything as a service market size & growth by 2028*. <https://www.fortunebusinessinsights.com/everything-as-a-service-xaas-market-102096>
- Furman, J., & Seamans, R. (2019). AI and the economy. *Innovation Policy and the Economy*, 19(1), 161-191.
- Gallardo, M. (2018). GDPR: Infracciones y sanciones. *Expansión*. <https://www.expansion.com/especiales/2018/GDPR/infracciones-y-sanciones.html>
- Ganapathy, A. (2020). Everything-as-a-service (XaaS) in the world of technology and trade. *American Journal of Trade and Policy*, 7(3), 91-98.
- Ghimire, D. (2020). Comparative study on python web frameworks: Flask and django.
- Gobierno de España. (2021a). *Plan de digitalización de PYMES 2021-2025*. Madrid. [https://espanadigital.gob.es/sites/espanadigital/files/2022-06/210127\\_plan\\_digitalizacion\\_pymes.pdf](https://espanadigital.gob.es/sites/espanadigital/files/2022-06/210127_plan_digitalizacion_pymes.pdf)
- Gobierno de España. (2021b). *Plan de digitalización de PYMES 2021-2025*. [https://espanadigital.gob.es/sites/espanadigital/files/2022-06/210127\\_plan\\_digitalizacion\\_pymes.pdf](https://espanadigital.gob.es/sites/espanadigital/files/2022-06/210127_plan_digitalizacion_pymes.pdf)
- Gobierno de España. (2021c). *Plan nacional de capacidades digitales*. <https://www.lamoncloa.gob.es/temas/fondos-recuperacion/Documents/16062021-Componente19.pdf>
- Gobierno de España. (2022). *España digital 2026*. [https://portal.mineco.gob.es/RecursosArticulo/mineco/ministerio/ficheros/EspanaDigital\\_2026.pdf](https://portal.mineco.gob.es/RecursosArticulo/mineco/ministerio/ficheros/EspanaDigital_2026.pdf)
- Google Trends. (2023). *Interés a lo largo del tiempo en la Inteligencia Artificial*. <https://trends.google.es/trends/explore?date=all&geo=ES&q=%2Fm%2F0mkz&hl=es>

- Hansen, E. B., & Bøgh, S. (2021). Artificial intelligence and internet of things in small and medium-sized enterprises: A survey. *Journal of Manufacturing Systems*, 58, 362-372.
- Hervás, L. V. (2022). ¿Cómo son las pymes en España? Cinco Días.  
[https://cincodias.elpais.com/cincodias/2022/06/24/pyme/1656070303\\_778210.html](https://cincodias.elpais.com/cincodias/2022/06/24/pyme/1656070303_778210.html)
- INE. (2022). *Empresas por CCAA, actividad principal (grupos CNAE 2009) y estrato de asalariados. (39372)*. Madrid:  
<https://www.ine.es/jaxiT3/Datos.htm?t=39372#!tabs-tabla>
- Janssen, N. (2022). The data science talent gap: Why it exists and what businesses can do about it. *Forbes*  
<https://www.forbes.com/sites/forbestechcouncil/2022/10/11/the-data-science-talent-gap-why-it-exists-and-what-businesses-can-do-about-it/>
- Kindström, D. (2010). Towards a service-based business model—Key aspects for future competitive advantage. *European Management Journal*, 28(6), 479-490.
- Lay, G., Schroeter, M., & Biege, S. (2009). Service-based business concepts: A typology for business-to-business markets. *European Management Journal*, 27(6), 442-455.  
<https://10.1016/j.emj.2009.04.002>
- Li, L., Lin, J., Luo, W., & Luo, X. R. (2023). Investigating the effect of artificial intelligence on customer relationship management performance in e-commerce enterprises. *Journal of Electronic Commerce Research*, 24(1), 68-83.  
[http://www.jecr.org/sites/default/files/2023vol24no1\\_Paper5.pdf](http://www.jecr.org/sites/default/files/2023vol24no1_Paper5.pdf)
- Margalló, N., Ginés, A., & Rojo, D. (2018). *Estado del mercado laboral en españa.* ().InfoJobs & ESADE.  
[https://nosotros.infojobs.net/wp-content/uploads/2019/05/Informe\\_Mercado\\_Laboral\\_InfoJobs\\_ESADE\\_2018.pdf](https://nosotros.infojobs.net/wp-content/uploads/2019/05/Informe_Mercado_Laboral_InfoJobs_ESADE_2018.pdf)
- Menéndez Pujadas, Á, & Mulino Ríos, M. A. (2022). Evolución económica reciente de las pymes españolas y de su acceso a la financiación externa, de acuerdo con la encuesta semestral del banco central europeo. *Boletín Económico/Banco De España [Artículos]*, N.3, 2022,
- Ministerio de asuntos económicos y transformación digital. (2022). *El gobierno lanza la invitación a las comunidades autónomas para que presenten proyectos estratégicos de digitalización dentro del programa RETECH del plan de recuperación.*
- Programa UNICO-banda ancha, (2022). [https://portal.mineco.gob.es/es-es/ministerio/estrategias/Paginas/06\\_Programa\\_UNICO\\_Banda\\_Ancha.aspx](https://portal.mineco.gob.es/es-es/ministerio/estrategias/Paginas/06_Programa_UNICO_Banda_Ancha.aspx)
- Observatorio Nacional de Tecnología y Sociedad. (2021). Informe de digitalización de las pymes 2021. Un análisis comparado. *Ministerio De Asuntos Económicos Y Transformación Digital, Secretaría General Técnica*,  
<https://www.ontsi.es/sites/ontsi/files/2021-09/digitalizacionpymes2021 analisiscomparado.pdf>
- Observatorio Nacional de Tecnología y Sociedad. (2022). *Uso de tecnología digitales por empresas en España*. Madrid: Ministerio de Asuntos Económicos y Transformación Digital.

<https://www.ontsi.es/sites/ontsi/files/2022-01/usotecnologiasdigitalesempresas2022.pdf>

- Pantoja, A. (2023). Roles y funciones en ciencia de datos.  
<https://www.linkedin.com/pulse/roles-y-funciones-en-ciencia-de-datos-antonio-pantoja-c/?originalSubdomain=es>
- Penteo. (2022). *Market trends - penteo data & AI*.  
<https://www.penteo.com/market-trends-data-ia/>
- Pérez Callejo, M., Aznar Alarcón, J. P., Margalló, N., Rojo, D., & García, A. (2023). *Estado del mercado laboral en España*. Infojobs.  
<https://nosotros.infojobs.net/wp-content/uploads/2023/03/Informe-Anual-InfoJobs-Esade-2022.pdf>
- Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking*. "O'Reilly Media, Inc."
- REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos, (2016).  
<https://www.boe.es/doue/2016/119/L00001-00088.pdf>
- Reglamento (UE) nº 651/2014 de la comisión de 17 de junio de 2014 por el que declaran determinadas categorías de ayudas compatibles con el mercado interior en aplicación de los artículos 107 y 108 del tratado (texto pertinente a efectos del EEE). (2014).  
<https://www.boe.es/doue/2014/187/L00001-00078.pdf>
- Reinhold, M., & Reinhold, S. (2014). Data-driven intelligence for SME e-business: A marketing and sales perspective. *Marketing Review St.Gallen*, 31, 52-59.
- Rentero, A. (2023). *Uso del dato en España por parte de las empresas*. Silicon.  
<https://www.silicon.es/uso-del-dato-en-espana-por-parte-de-las-empresas-2474122>
- Secretaría de Estado de Digitalización e Inteligencia Artificial. (2022). *Estrategia nacional de inteligencia artificial*. Gobierno de España.  
[https://portal.mineco.gob.es/RecursosNoticia/mineco/prensa/noticias/2020/201202\\_np\\_eniav.pdf](https://portal.mineco.gob.es/RecursosNoticia/mineco/prensa/noticias/2020/201202_np_eniav.pdf)
- Secretaría de Estado de Telecomunicaciones e Infraestructuras Digitales. (2022). *Cobertura de banda ancha en España*. Madrid: Gobierno de España.  
[https://avancedigital.mineco.gob.es/banda-ancha/cobertura/Documents/Informe\\_Cobertura\\_BA\\_2022.pdf](https://avancedigital.mineco.gob.es/banda-ancha/cobertura/Documents/Informe_Cobertura_BA_2022.pdf)
- Shenoy, A., & Prabhu, A. (2016). *Introducing SEO*. Apress.Mumbai.Luettavissa:  
<https://Haagahelia.Finna.Fi/Record/nelli21.3710000000765483.Luettu>, 22, 2021.
- Valentini, G., & Masulli, F. (2002). (2002). Ensembles of learning machines. Paper presented at the *Neural Nets: 13th Italian Workshop on Neural Nets, WIRN VIETRI 2002 Vietri Sul Mare, Italy, may 30–June 1, 2002 Revised Papers 13*, 3-20.
- Yalçın, N., & Köse, U. (2010). What is search engine optimization: SEO? *Procedia-Social and Behavioral Sciences*, 9, 487-493.

- Zaeem, R. N., & Barber, K. S. (2020). The effect of the GDPR on privacy policies: Recent progress and future promise. *ACM Transactions on Management Information Systems (TMIS)*, 12(1), 1-20.
- Zaki, M. (2019). Digital transformation: Harnessing digital technologies for the next generation of services. *Journal of Services Marketing*,
- Zhang, S., & Cabage, N. (2017). Search engine optimization: Comparison of link building and social sharing. *Journal of Computer Information Systems*, 57(2), 148-159.