GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Desarrollo de una plataforma de extracción de conocimiento a partir de datos arbitrarios “Machine Learning as a Service”.

Autor: Ignacio Divassón González

Director: Miguel Ángel Sanz Bobi

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Desarrollo de una plataforma de extracción de conocimiento a partir de datos arbitrarios "Machine Learning as a Service".

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2022/23 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Ignacio Divassón González Fecha: 31/ 05/ 2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Firmado por SANZ BOBI MIGUEL ANGEL - ***6599** el día 31/05/2023
con un certificado emitido por AC FNMT Usuarios

Fdo.: Miguel Ángel Sanz Bobi Fecha: ……/ ……/ ……

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Desarrollo de una plataforma de extracción de conocimiento a partir de datos arbitrarios “Machine Learning as a Service”.

Autor: Ignacio Divassón González

Director: Miguel Ángel Sanz Bobi

Madrid

Agradecimientos

A Miguel Ángel Sanz, mi tutor del TFG por ayudarme en el proyecto y en la elaboración de la memoria, estando siempre accesible y dispuesto a ayudar.

A mi familia por el apoyo recibido durante este proyecto.

DESARROLLO DE UNA PLATAFORMA DE EXTRACCIÓN DE CONOCIMIENTO A PARTIR DE DATOS ARBITRARIOS "MACHINE LEARNING AS A SERVICE".

Autor: Divassón González, Ignacio.
Director: Sanz Bobi, Miguel Ángel.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

La plataforma "Machine Learning as a Service" es una aplicación web con la capacidad de generar modelos de predicción, tanto de regresión como de clasificación para el usuario sin que éste tenga que programar ni tener conocimientos sobre algoritmos de aprendizaje automático. Simplemente tiene que subir datos a la aplicación web, y esta guía al usuario y proactivamente recomienda decisiones para el usuario a la hora de generar los mejores modelos predictivos para el conjunto de datos subido.

Palabras clave: Machine Learning as a Service, WebApp, Data Science, Big Data, Cloud Computing

1. Introducción

El mundo del Machine Learning ha evolucionado mucho en estos últimos años, desarrollando muchos y complejos algoritmos para regresión y clasificación [1]. El problema es que estos avances en Inteligencia Artificial y más concretamente en el mundo del Machine Learning no se han puesto de manera fácilmente accesible a disposición del público general no experto. Ni los usuarios corrientes ni, sobre todo, las PYMES, utilizan casi el *cloud computing* y el Big Data. Solo lo hacen el 8.73% y el 3,13% de las PYMES respectivamente [2]. La razón de esta falta de uso, la encontramos en el informe *Uso de tecnologías digitales por empresas en España* [3], que indica que la principal razón por la que las PYMES no utilizan la inteligencia artificial es la falta de conocimiento en la materia.

Este proyecto pretende que personas u organizaciones sin conocimientos de análisis de datos o de algoritmos de machine learning, puedan utilizar estas herramientas de predicción.

2. Definición del proyecto

Este proyecto es una aplicación web mediante la cual un usuario sin experiencia ninguna en programación o análisis de datos puede entrenar modelos de machine learning subiendo un conjunto de datos, y hacer predicciones sobre nuevos datos.

Esta plataforma tiene una interfaz sencilla, amigable y No-Code, en la que los usuarios autenticados pueden crear proyectos de machine learning, subiendo un conjunto de datos, con los que se entrenarán modelos de predicción.

Para cada proyecto, la plataforma guía al cliente a través de todas las fases necesarias en los proyectos de machine learning, proponiendo soluciones de manera proactiva para cada una de estas fases: preprocesado, visualización, codificación y particionado, búsqueda de hiperparámetros, entrenamiento, valoración y comparación de los modelos

entrenados, y finalmente, la predicción sobre nuevos datos. Además, la plataforma permite la creación de Ensembles de una manera muy sencilla e intuitiva.

3. Descripción de la plataforma

Esta aplicación web está desarrollada con el framework (o entorno de trabajo) de Django, y está compuesta de varias aplicaciones o módulos que realizan distintos servicios dentro de toda la plataforma.

Las aplicaciones que tiene este proyecto de MLaaS (Machine Learning as a Service) son:

I. Landing Page
II. Autenticación
III. CrearAbrirProyectos
IV. AnalizarDatos
V. CrearModelos
VI. VisualizarModelos
VII. Predicción

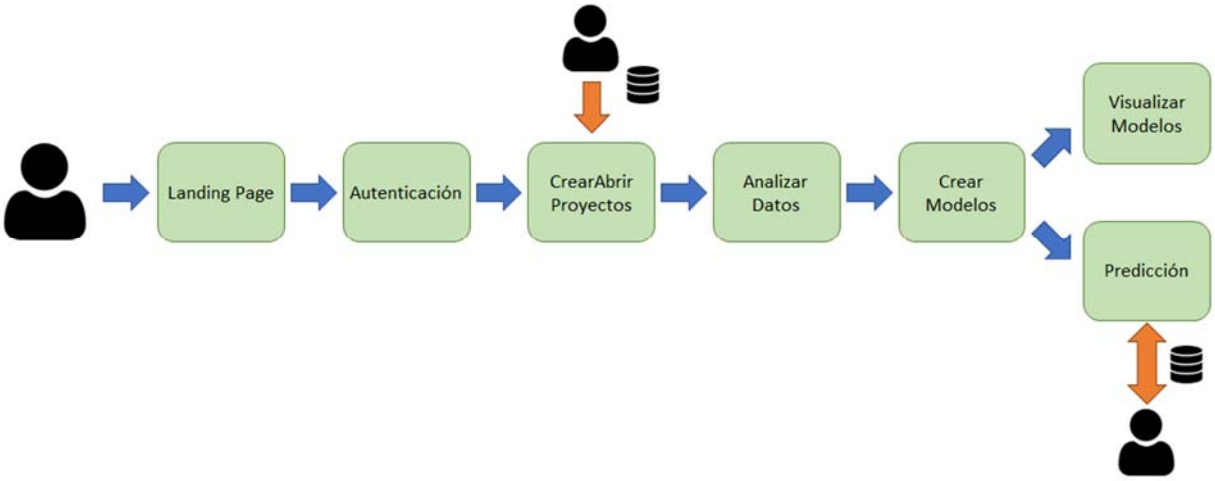


Ilustración 1 – Diagrama de las aplicaciones de la plataforma desarrollada

La primera aplicación, denominada Landing Page, se dedica exclusivamente a mostrar la página web de inicio, así como los archivos CSS, JavaScript e imágenes que se utilizan en los demás Templates HTML de Django en el conjunto de aplicaciones. Esto permite que la plataforma mantenga una imagen coherente en su totalidad.

La segunda aplicación, Autenticación, tiene como responsabilidad principal registrar a los usuarios en la base de datos, permitir que inicien sesión en la plataforma y asegurarse de que el proyecto al que un usuario intenta acceder realmente le pertenezca, brindando así un nivel de seguridad adicional.

La tercera aplicación, Crear Abrir Proyectos, ofrece a los usuarios la posibilidad de crear nuevos proyectos de machine learning vinculados a su cuenta. Para ello, el usuario solamente necesita un nombre y una base de datos, como un archivo Excel, por ejemplo. Además, esta aplicación permite al usuario visualizar todos los proyectos que ha creado y eliminar aquellos que desee.

La cuarta aplicación, Analizar Datos, integra las funciones de preprocesamiento y análisis de datos para los proyectos de machine learning. Esta aplicación guía al usuario a través de las etapas necesarias, sin permitir omitirse ninguna etapa, ofreciendo recomendaciones proactivas tras analizar los datos.

En la primera etapa, el usuario especifica la variable a predecir en el proyecto. Posteriormente, el usuario confirma si el problema es de clasificación o regresión, ofreciéndole una opción predeterminada basada en el análisis de la variable objetivo. Después, se ofrece al usuario poder cambiar los tipos de datos de las variables del proyecto. A continuación, la plataforma sugiere acciones para el preprocesamiento de datos, como eliminar columnas innecesarias y tratar los datos faltantes, proporcionando la aplicación recomendaciones basadas en el porcentaje de datos faltantes. Finalmente, se ofrece una herramienta de análisis exploratorio de los datos (EDA) que permite al usuario explorar las relaciones y distribuciones entre las variables mediante gráficos interactivos.

La quinta aplicación, Crear Modelos, es responsable de entrenar modelos de machine learning en un proyecto. Inicialmente, la plataforma ofrece al usuario entre dos modos de entrenamiento de los modelos, “Normal” y “Experto”. El modo “Normal” entrena automáticamente todos los modelos disponibles para el tipo de problema del proyecto, buscando automáticamente los mejores hiperparámetros para cada uno de los modelos. El modo “Experto” permite al usuario seleccionar qué porcentaje de datos quiere destinar a prueba y cuántos a entrenamiento, ratio que en el modo “Normal” es de un 30%-70% respectivamente. Además, en el modo “Experto” el usuario puede elegir qué modelos quiere entrenar, y para cada uno de ellos elegir que se busque automáticamente los mejores hiperparámetros como en el modo “Normal”, o utilizar unos hiperparámetros en concreto. Además, si se han entrenado al menos dos modelos, esta aplicación permite al usuario la creación de Ensembles de una manera sencilla para mejorar la precisión de éstos.

La sexta aplicación, Visualizar Modelos, se encarga de generar visualizaciones fáciles de comprender para que el usuario pueda ver y comparar cómo se comportan los modelos que ha entrenado en la plataforma. Además, se añaden explicaciones sobre las visualizaciones para los usuarios menos experimentados en visualizar el comportamiento de modelos de Machine Learning. Finalmente, tanto los modelos entrenados como los Ensembles se ordenan en función de su eficacia con el conjunto de prueba para que el usuario, de un vistazo, sepa cuál es el mejor modelo o Ensemble.

Finalmente, la séptima aplicación, Predicción, se encarga de predecir, con los modelos ya entrenados sobre un nuevo conjunto de datos que aporte el usuario, y descargar el archivo subido a la aplicación, junto con una columna extra que contiene la predicción del modelo entrenado.

4. Resultados

La plataforma desarrollada cumple con todos los objetivos del proyecto, y es capaz de generar modelos y realizar predicciones sobre datos arbitrarios sin necesitar ningún conocimiento de análisis de datos o de modelos predictivos, ni mucho menos saber programar.

Las ilustraciones que siguen muestran distintas partes de la plataforma web.

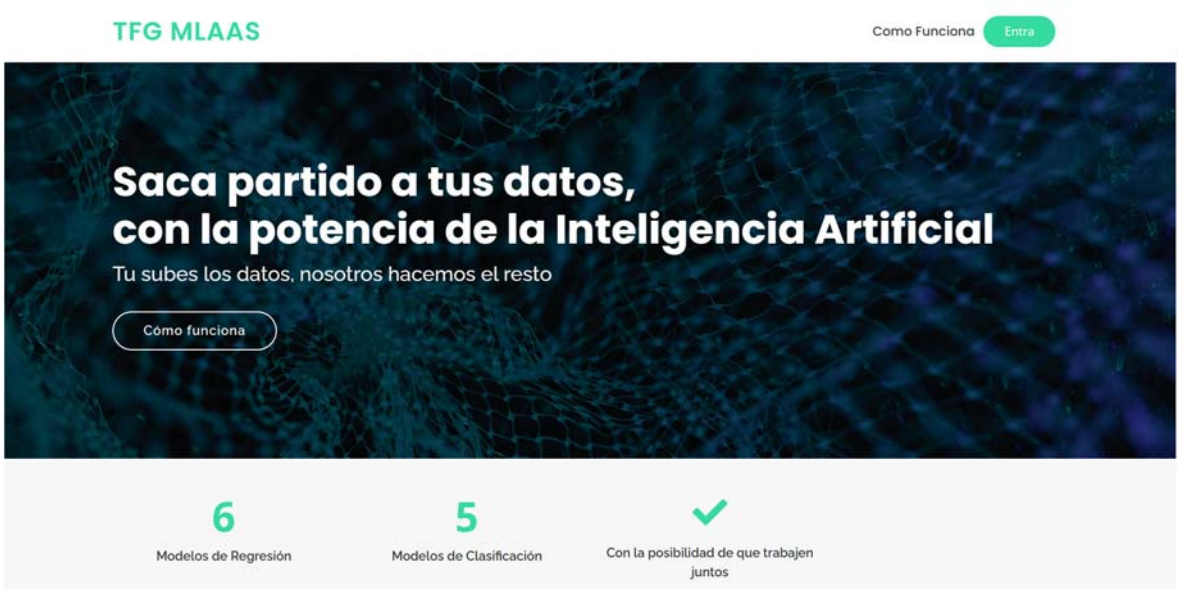


Ilustración 2 – Página web de inicio (Landing Page)

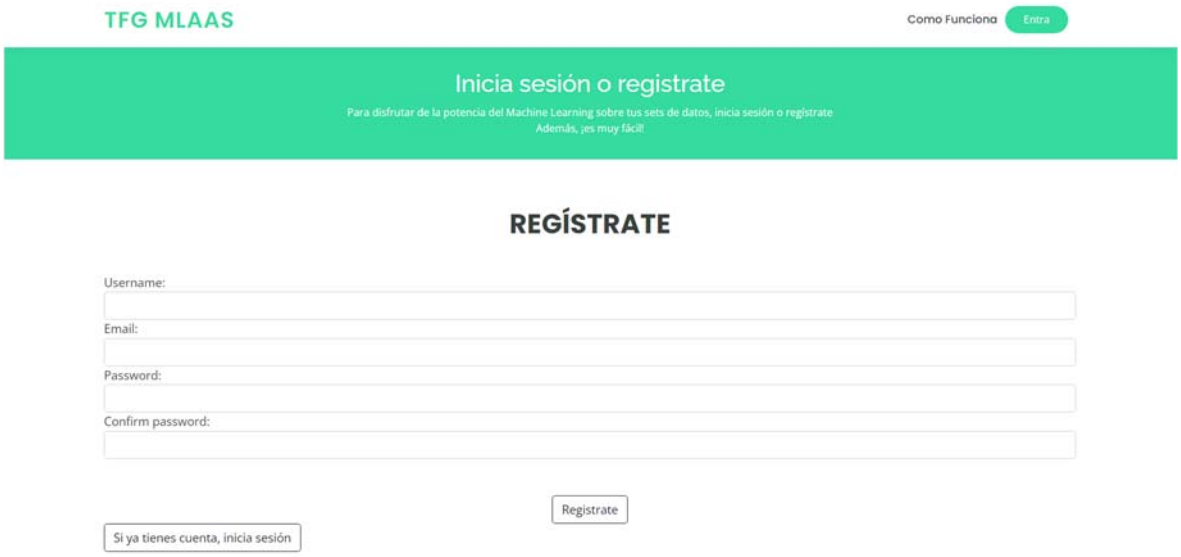


Ilustración 3 – Página web para darse de alta en la plataforma

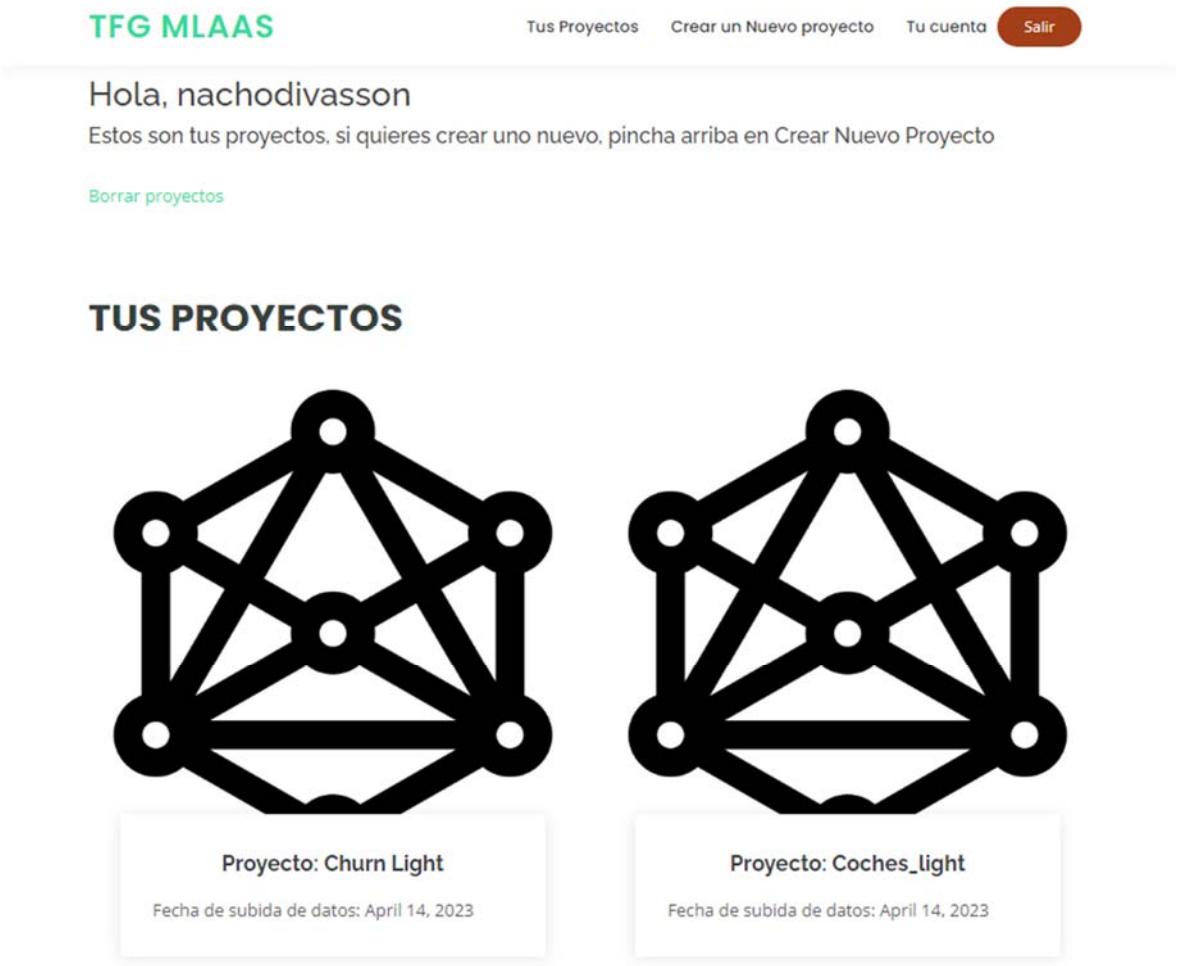


Ilustración 4 – Visión de los proyectos de un usuario

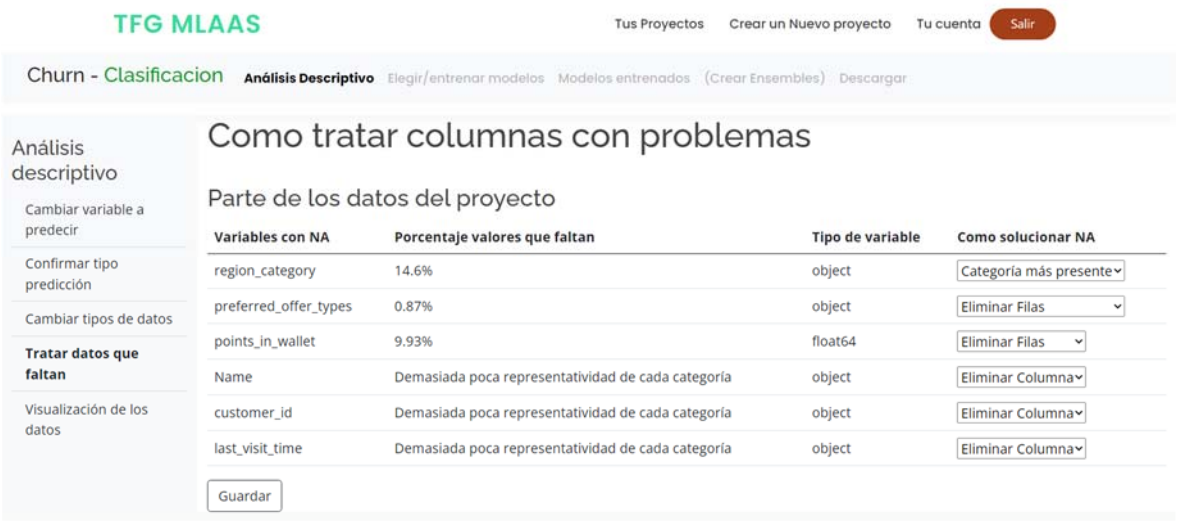


Ilustración 5 – La aplicación propone proactivamente tratar de distintas formas datos no existentes

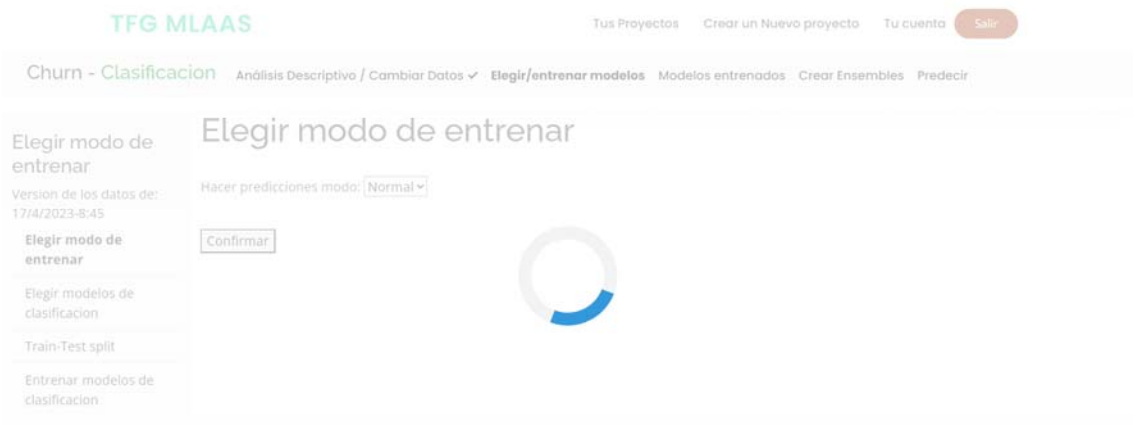


Ilustración 6 – La aplicación está entrenando todos los modelos en la forma de entrenamiento "Normal"

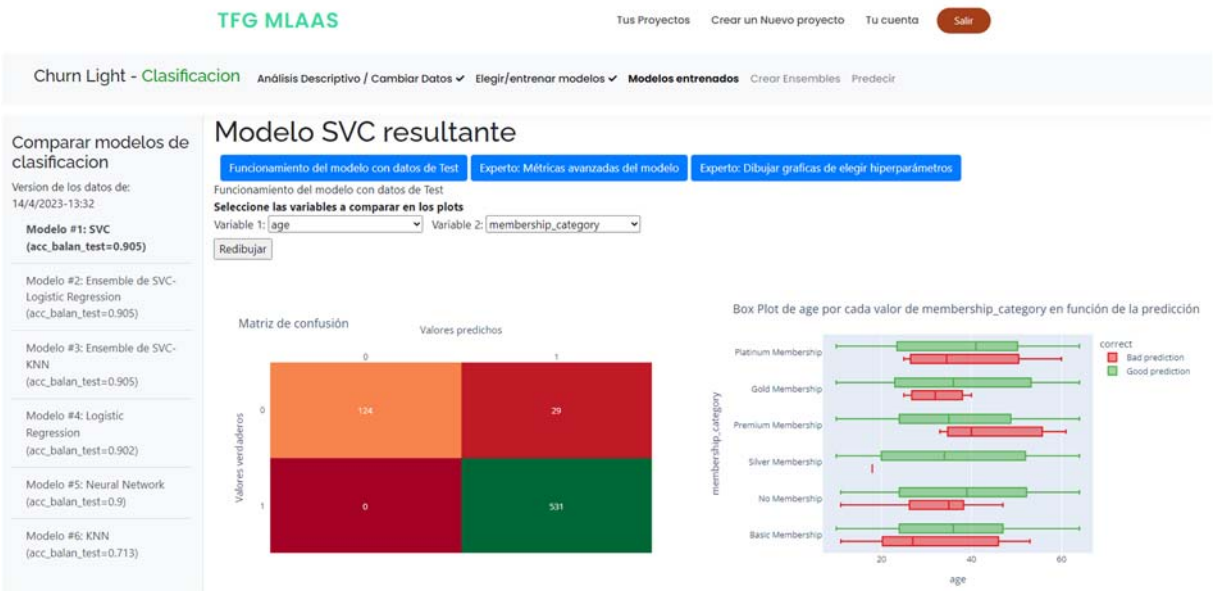


Ilustración 7 – Visualización del rendimiento de los modelos entrenados

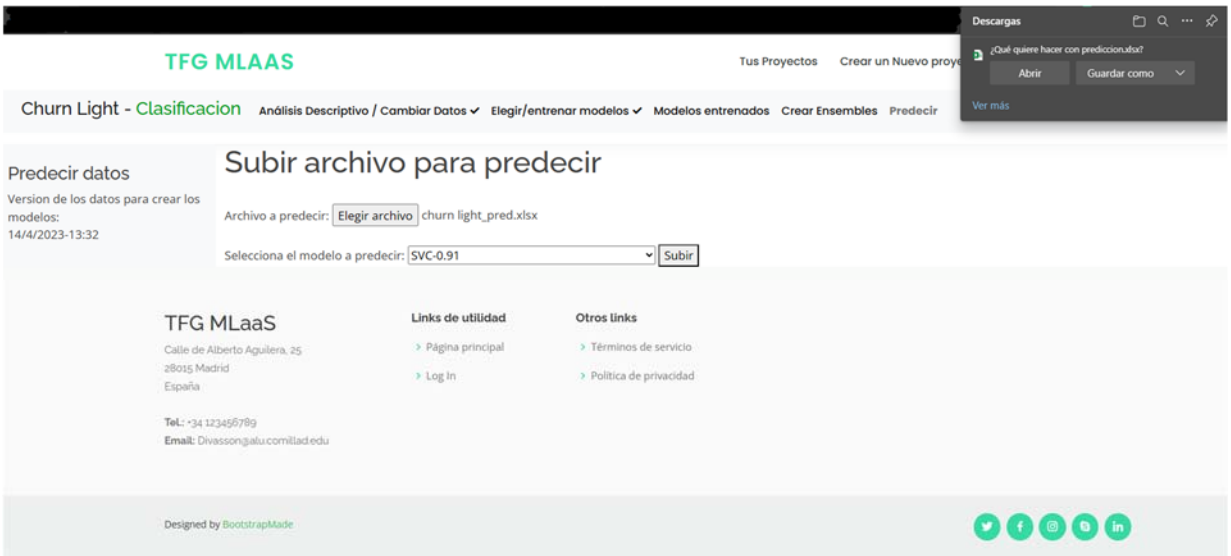


Ilustración 8 – Predicción sobre un nuevo conjunto de datos

5. Conclusiones

La plataforma desarrollada de Machine Learning as a Service (MLaaS), permite acercar el mundo del machine learning a todos los usuarios, incluso aquellos sin conocimientos de programación o análisis de datos.

Posee una interfaz intuitiva y amigable, guía a los usuarios en todas las etapas del proceso de crear modelos de machine learning y proporciona recomendaciones proactivas en cada toma de decisiones basadas en los datos para los usuarios con menos experiencia. Además, cuenta con herramientas de visualización interactiva para comprender y comparar los modelos entrenados. La plataforma también permite la creación de ensembles, combinando varios modelos ponderados y garantizando la privacidad de los proyectos mediante la autenticación de usuarios.

El proyecto es una solución útil para usuarios que deseen aprovechar el poder predictivo del machine learning sin que se requiera conocimientos especializados en programación o inteligencia artificial. Especialmente, puede ser de utilidad para PYMES y departamentos de grandes empresas que carecen de científicos de datos.

6. Referencias

[1] Aggarwal, K., Mijwil, M. M., Al-Mistarehi, A., Alomari, S., Gök, M., Alaabdin, A. M. Z., & Abdulrhman, S. H. (2022). Has the future started? the current growth of artificial intelligence, machine learning, and deep learning. Iraqi Journal for Computer Science and Mathematics, 3(1), 115-123.

[2] Ministerio de Asuntos Económicos y Transformación Digital. (2021). *Informe de digitalización de las PYMES 2021. un análisis comparado.* (13,47,73). Gobierno de España.
myyux?44| | | 3tsyxn3jx4xnyjx4tsyxn4knqjx475762
5>4irlnyfqnfhntsu~r jx7576fsfqpxnxhtr ufwfit3uik

[3] Observatorio Nacional de Tecnología y Sociedad. (2022). Uso de tecnología digitales por empresas en España. Ministerio De Asuntos Económicos Y Transformación Digital, Secretaría General Técnica, https://www.ontsi.es/sites/ontsi/files/2022-01/usotecnologiasdigitalesempresas2022.pdf

[4] Django. (2023). *Django*. Django Project. https://www.djangoproject.com/

[5] Bhat, I., & Data Bridge Market Research. (2021). *Machine learning as a service (MLaaS) market*. https://www.linkedin.com/pulse/machine-learning-service-mlaas-market-estimated-2028-likely-indu-bhat/

[6] Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130.

DEVELOPMENT OF A PLATFORM FOR KNOWLEDGE EXTRACTION FROM ARBITRARY DATA "MACHINE LEARNING AS A SERVICE".

Author: Divassón González, Ignacio.
Supervisor: Sanz Bobi, Miguel Ángel.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

The "Machine Learning as a Service" platform is a web application with the ability to generate predictive supervised machine learning models of regression and classification, for the user without the user having to code or have any knowledge of machine learning algorithms. The user simply uploads its data to the web application, and the application guides the user and proactively recommends decisions to generate the best predictive models for the uploaded dataset.

Keywords: Machine Learning as a Service, WebApp, Data Science, Big Data, Cloud Computing

1. Introduction

The world of Machine Learning has evolved a lot in recent years, developing many complex algorithms for regression and classification [1]. The problem is that these advances in Artificial Intelligence and more specifically in the world of Machine Learning have not been made easily accessible to the non-expert general public. Neither ordinary users nor, above all, SMEs, make almost any use of cloud computing and Big Data. Only 8.73% and 3.13% of SMEs do so respectively [2]. The reason for this lack of use, can be found in the report *Use of digital technologies by companies in Spain* [3], which indicates that the main reason why SMEs do not use artificial intelligence is the lack of knowledge on the subject.

This project aims to enable people or organizations without knowledge of data analysis or machine learning algorithms to use these predictive tools.

2. Project definition

This project is a web application through which a user with no experience in programming or data analysis can train machine learning models by uploading a dataset and make predictions on new data.

This platform has a simple, friendly, and No-Code interface, in which authenticated users can create machine learning projects by uploading a dataset, with which prediction models will be trained.

For each project, the platform guides the client through all the necessary phases of machine learning projects, proactively proposing solutions for each of these phases: preprocessing, visualization, encoding and partitioning data, hyperparameter search, training, evaluation and comparison of the trained models, and finally, prediction on new data. In addition, the platform allows for the creation of Ensembles in a very simple and intuitive way.

3. Web application description

The web application has been developed with the Django framework and is composed of several applications or modules that perform different services within the entire platform.

The applications that this MLaaS (Machine Learning as a Service) project has are:

I. Landing Page
II. Authentication
III. CreateOpenProject
IV. AnalyzeData
V. CreateModels
VI. VisualizeModels
VII. Prediction

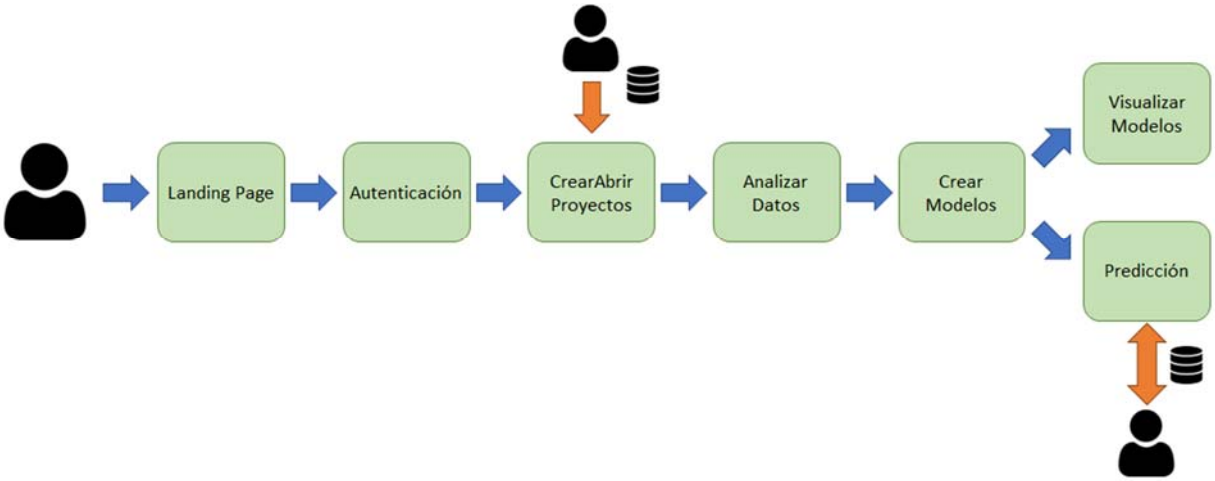


Illustration 1 - Diagram of the developed platform applications

The first application, called Landing Page, is dedicated exclusively to displaying the home page, as well as the CSS, JavaScript and images that are used in the other HTML Django Templates in the application suite. This allows the platform to maintain a consistent look and feel throughout.

The second application, Authentication, has the primary responsibility of registering users in the database, allowing them to log in to the platform, and ensuring that the project a user is trying to access belongs to them, thus providing an additional level of security.

The third application, Create Open Projects, offers users the ability to create new machine learning projects linked to their account. To do so, the user only needs a project name and a database, such as an Excel file. In addition, this application allows users to view all the projects they have created and to delete those they wish to delete.

The fourth application, Analyze Data, integrates the preprocessing and data analysis functions needed for machine learning projects. This application guides the user through the necessary steps, without allowing any step to be skipped, while offering proactive recommendations based on the project data.

In the first stage of this fourth module, the user specifies the target variable for the project. Subsequently, the user confirms whether the problem is a classification or regression problem, offering a default option based on the analysis of the target variable. Next, the user is offered by the platform the ability to change the data types of the project variables. Afterwards, the platform suggests actions for data preprocessing, such as eliminating unnecessary columns and dealing with missing data, with the application providing recommendations based on the percentage of missing data. Finally, an exploratory data analysis (EDA) tool allows the user to explore relationships and distributions between variables through interactive graphs.

The fifth application, Create Models, is responsible for training machine learning models on a project. Initially, the platform offers the user between two model training modes, "Normal" and "Expert". The "Normal" mode automatically trains all available models for the project problem type, automatically searching for the best hyperparameters for each of the models. The "Expert" mode allows the user to select what percentage of data he/she wants to allocate to testing and how much to training, a ratio that in the "Normal" mode is 30%-70% respectively. In addition, in the "Expert" mode the user can choose which models to train. For each of the selected models, the user can choose to automatically search for the best hyperparameters as in the "Normal" mode or use specific hyperparameters. In addition, if at least two models have been trained, this application allows the user to create Ensembles in an effortless way to improve trained models' accuracy.

The sixth application, Visualize Models, is responsible for generating easy-to-understand visualizations so that the user can see and compare how the models that have been trained on the platform behave with test data. In addition, explanations of the visualizations are added for users less experienced in visualizing the behavior of Machine Learning models. Finally, both the trained models and the Ensembles are ordered according to their effectiveness with the test set so that the user, at a glance, knows which is the best model or Ensemble.

Finally, the seventh application, Prediction, is in charge of using the trained models to predict on a new data set provided by the user, and downloading the file uploaded to the application, along with an extra column containing the prediction of the trained model.

4. Results

The developed platform meets all the objectives of the project and is able to generate models and make predictions on arbitrary data without requiring any knowledge of data analysis or predictive modeling, let alone programming skills.

The illustrations below show the different parts of the web platform.

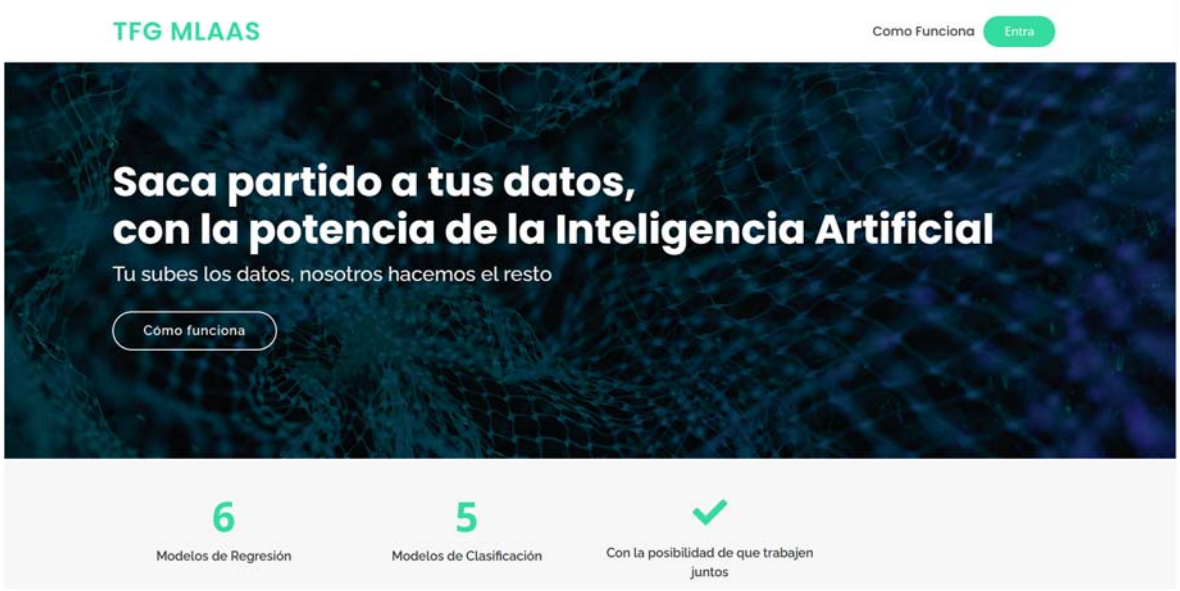


Illustration 2 - Landing Page

Illustration 3 - Web page to register on the platform

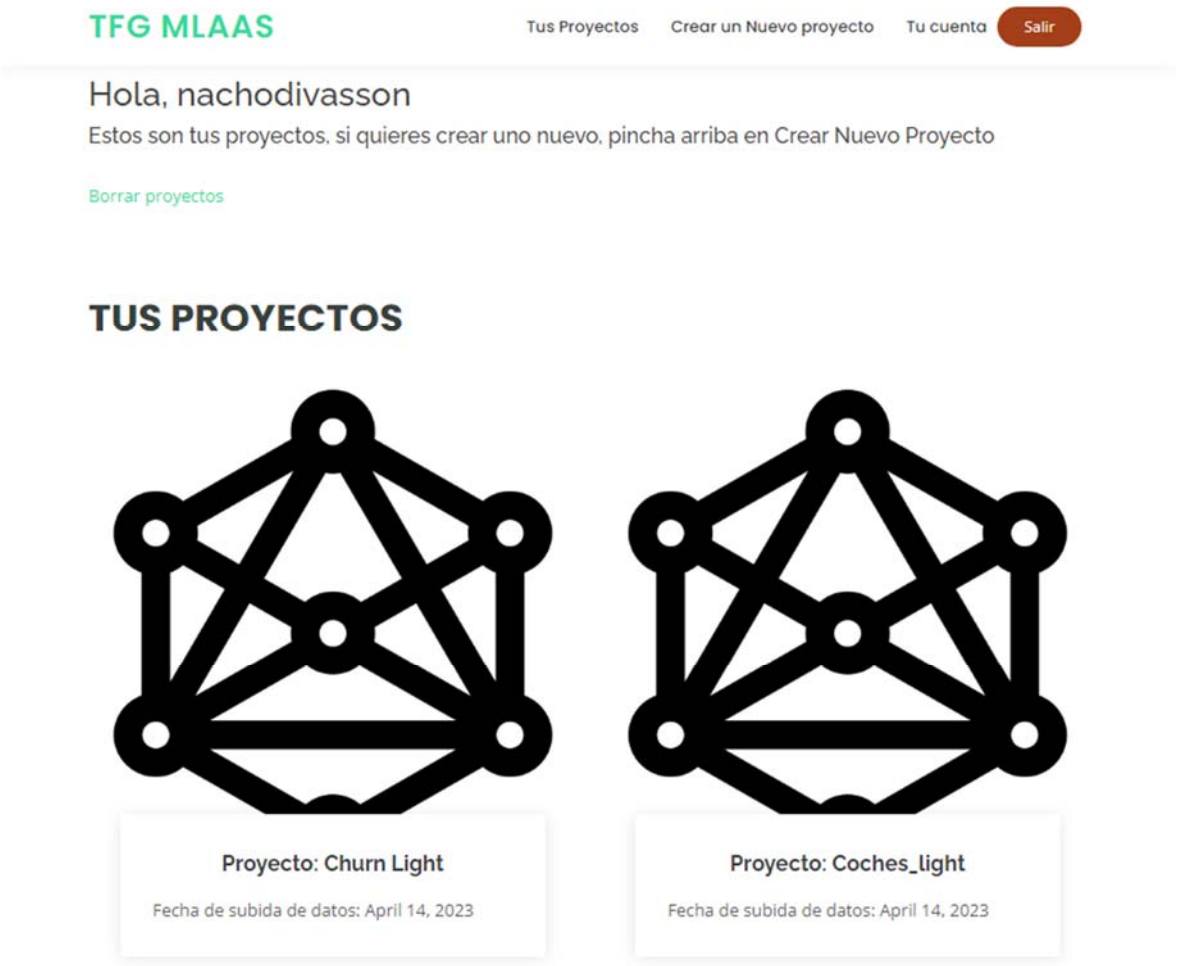


Illustration 4 - User's view of its projects

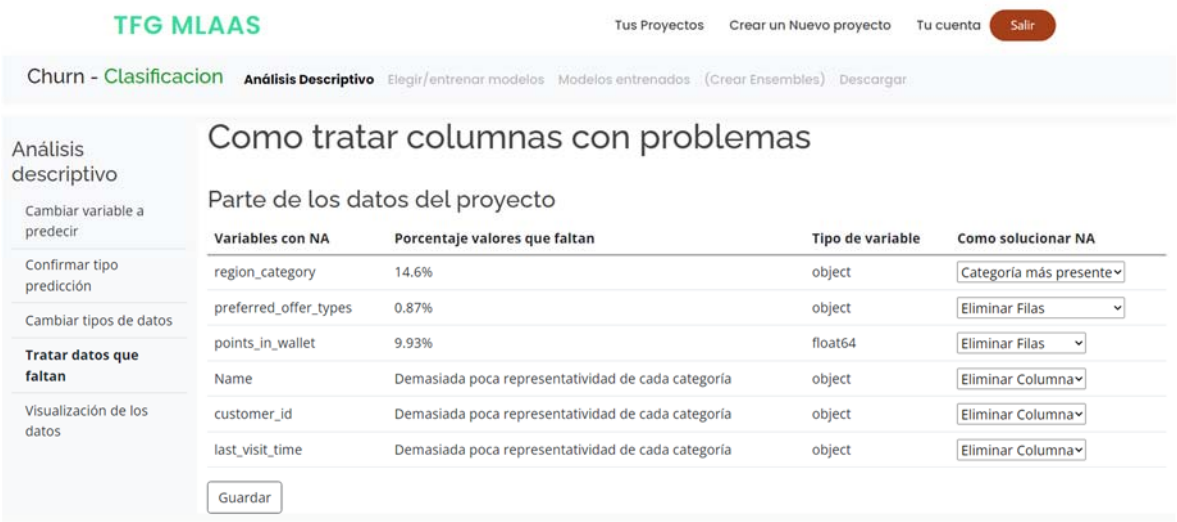


Illustration 5 - The application proactively proposes to process non-existing data in different ways

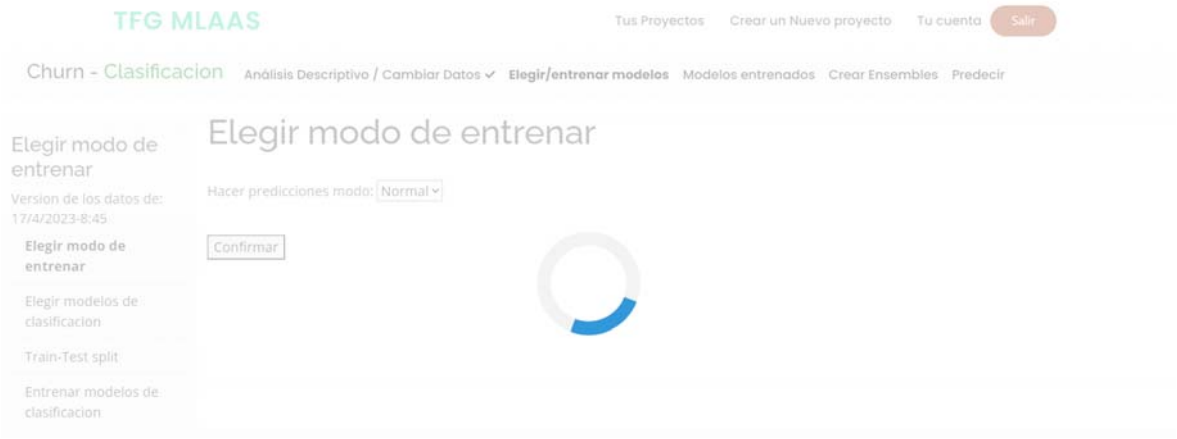


Illustration 6 - The application is training all models in the "Normal" training mode.

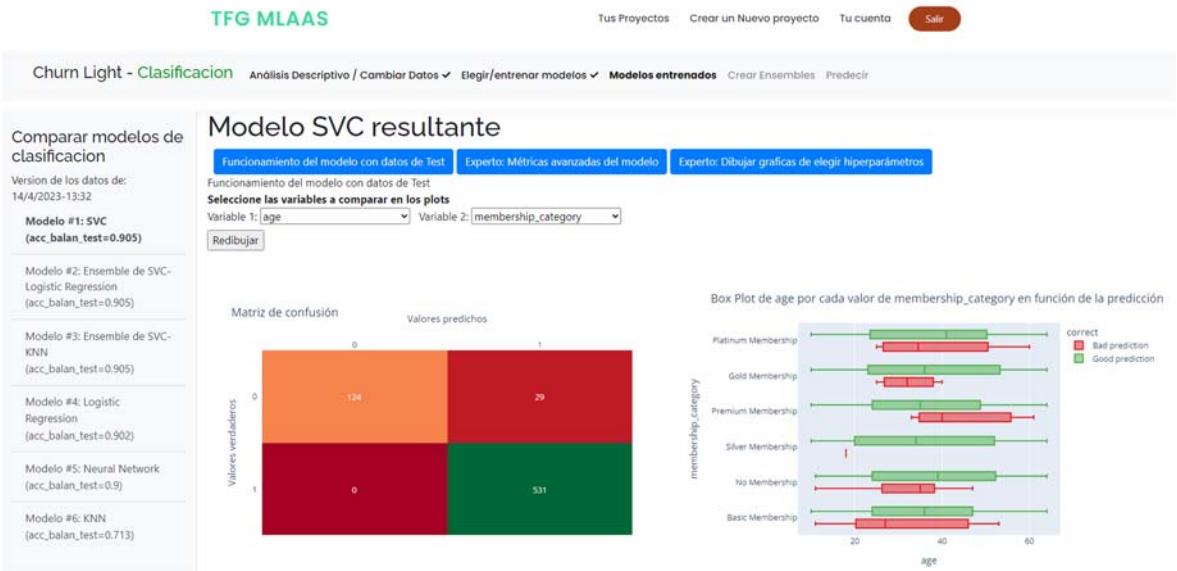


Illustration 7 - Visualization of the performance of the trained models

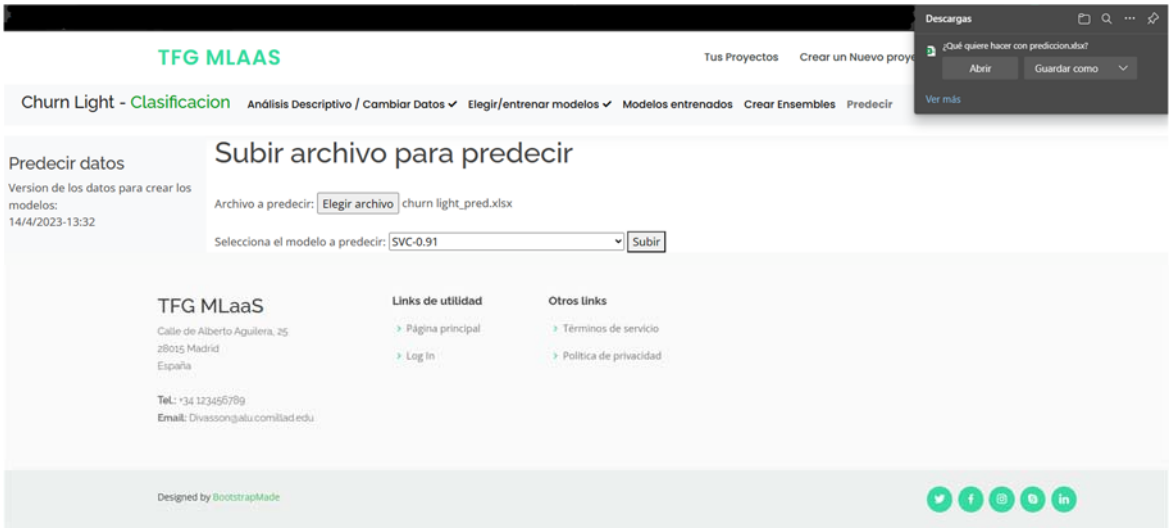


Illustration 8 - Prediction on a new data set

5. Conclusions

This Machine Learning as a Service (MLaaS) platform, which has been developed in this project, brings the world of machine learning to all users, even those with no programming or data analysis knowledge.

It has an intuitive and user-friendly interface, guides users through all stages of machine learning model training and provides proactive data-driven recommendations for every decision-making process for less experienced users. In addition, it features interactive visualization tools for understanding and comparing trained models. The platform also allows the creation of ensembles, combining several models, and ensures project privacy through user authentication.

Although there are some limitations, such as the lack of support for custom metrics and the lack of more complex models such as unsupervised learning, the project is a useful solution for users who want to take advantage of the predictive power of machine learning without requiring specialized knowledge in programming or artificial intelligence. In particular, it can be useful for SMEs and departments of large companies that lack data scientists.

6. Referencias

[1] Aggarwal, K., Mijwil, M. M., Al-Mistarehi, A., Alomari, S., Gök, M., Alaabdin, A. M. Z., & Abdulrhman, S. H. (2022). Has the future started? the current growth of artificial intelligence, machine learning, and deep learning. Iraqi Journal for Computer Science and Mathematics, 3(1), 115-123.

[2] Ministerio de Asuntos Económicos y Transformación Digital. (2021). *Informe de digitalización de las PYMES 2021. un análisis comparado.* (13,47,73). Gobierno de España.
myyux?44| | | 3tsyxn3jx4xnyjx4tsyxn4knqjx475762
5>4irlnyfqnfhntsu~r jx7576fsfqpxnxhtr ufwfit3uik

[3] Observatorio Nacional de Tecnología y Sociedad. (2022). Uso de tecnología digitales por empresas en España. Ministerio De Asuntos Económicos Y Transformación Digital, Secretaría General Técnica, https://www.ontsi.es/sites/ontsi/files/2022-01/usotecnologiasdigitalesempresas2022.pdf

[4] Django. (2023). *Django*. Django Project. https://www.djangoproject.com/

[5] Bhat, I., & Data Bridge Market Research. (2021). *Machine learning as a service (MLaaS) market*. https://www.linkedin.com/pulse/machine-learning-service-mlaas-market-estimated-2028-likely-indu-bhat/

[6] Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130.

Índice de la memoria

Índice de figuras

Índice de tablas

Capítulo 1. INTRODUCCIÓN

Vivimos en la edad de la abundancia de los datos. Generamos datos constantemente. Desde nuestras búsquedas en la red, al contenido que visualizamos en redes sociales, pasando por los mensajes que enviamos a nuestros amigos. Se estima que se crean 2.5 quintillones de bytes de datos cada día [1], y no parece que este crecimiento vaya a dejar de crecer. Pero almacenar estos datos solo tendría valor si se consigue extraer conocimiento de ellos. Si no, sólo ocuparían espacio en el disco duro donde estén guardados.

Extraer conocimientos de los datos significa que su posesión reporte algún valor al que lo guarda. Este valor añadido puede venir entre otras cosas por entender las causas de un evento, obtener más información sobre un cliente, o predecir un evento en el futuro.

Existen muchas formas de extraer conocimiento de los datos, pero sin duda, la más potente es utilizar algoritmos de Machine Learning (ML). Machine Learning es un campo dentro de la rama de las ciencias de la computación que permite a los ordenadores detectar patrones con el objetivo de poder predecir un evento en el futuro. Los algoritmos de Machine Learning, guiados por el diseñador, son capaces de crear modelos con el objetivo de que éstos representen lo más certeramente la realidad para poder predecir eventos con nuevos datos de entrada. Cuanto más precisos sean estos modelos creados, más conocimiento se estará extrayendo.

Los algoritmos de Machine Learning son tan potentes pues permiten extraer conocimiento de cantidades masivas de datos (Big Data en inglés) con algoritmos ya contrastados. El problema actual es que el uso de estos algoritmos de Machine Learning requiere de unos conocimientos de programación muy extensivos que no están al alcance de muchos.

Actualmente, las empresas contratan a científicos de datos para elaborar sus modelos de extracción de conocimiento con herramientas de Machine Learning. Esto representa una barrera de entrada para gran cantidad de empresas que, aun teniendo datos, o no tienen el

músculo financiero para permitirse contratar a un científico de datos, o la frecuencia con la que desean extraer información de sus datos es lo suficientemente baja como para afrontar ese gasto recurrente. Además, carecen de conocimiento sobre tratamiento de datos.

A la hora de aplicar algoritmos de Machine Learning, éstos se pueden ejecutar en local o en proveedores de servicios en la nube (cloud providers en inglés).

Para ejecutar estos algoritmos en local se necesita tener un entorno de desarrollo en el ordenador, y para que los algoritmos se entrenen se necesita codificarlos con algún lenguaje de programación, como por ejemplo Python o R.

La otra opción es utilizar las plataformas de la nube, como Apache Spark. Estas, aparte del coste de uso, son más complejas de utilizar porque el usuario aparte de programar los algoritmos de Machine Learning también necesita tener conocimientos sobre entornos de desarrollo en la nube, tales como la de determinar los recursos que se van a adjudicar a ese proceso.

Hoy en día, las grandes plataformas de la nube como Amazon, Google, y sobre todo Microsoft Azure, están simplificando sus sistemas para que el usuario no tenga que escribir código para entrenar los modelos de extracción de información y predicción. Aunque esto supone bajar la barrera de entrada para muchos usuarios potenciales, se sigue requiriendo conocimientos de Machine Learning pues el código se ha sustituido por una interfaz gráfica en forma de diagramas con cajas (módulos), pero no se ha automatizado.

Así que este proyecto pretende crear un entorno de desarrollo y entrenamiento de modelos de machine learning que sea gratis amigable y sencillo para acercar el mundo de Machine Learning a cualquier usuario independientemente de su conocimiento.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

2.1 INTRODUCCIÓN A LAS TECNOLOGÍAS UTILIZADAS

Esta plataforma de MLaaS va a utilizar diferentes tecnologías para llevar a cabo cada parte del proyecto. Estas tecnologías son las siguientes:

- **Django**: esta tecnología permite crear aplicaciones web, y es la base de este proyecto. La aplicación web creada será con la que interactúe el usuario y las distintas partes de la plataforma.
- **Algoritmos de machine learning**: a la hora de crear una plataforma de Machine Learning, es necesario implementar algoritmos de predicción. Los algoritmos de predicción utilizados son implementados con las librerías de keras y de scikit-learn.
- **Optuna**: esta tecnología permite elegir los mejores hiperparámetros de los algoritmos de machine learning mencionados anteriormente.
- **Plotly**: esta tecnología permite crear gráficos interactivos y sencillos para transmitir mejor la información al usuario inexperto.

2.2 DJANGO

La mejor forma para crear un servicio de "Machine Learning as a Service" como el propuesto, es con una aplicación web. Esta aplicación web idóneamente deberá estar implementada, como la mayoría de sus competidores, en algún proveedor de servicios en la nube, como puede ser de Amazon Web Services, para que los usuarios se conecten al servidor simplemente mediante una URL.

Para crear una aplicación web o plataforma, hay que elegir un framework de desarrollo de aplicaciones web que gestione todos los servicios web que componen la aplicación, desde las bases de datos donde se guardan distintos recursos, hasta cómo enrutar hasta las

diferentes vistas en función de la ruta URL, pasando por cargar las librerías utilizadas en esta aplicación web.

Existen muchos frameworks de desarrollo de aplicaciones web, pero Django [2] ha sido elegido por la rapidez que proporciona a las aplicaciones, su escalabilidad, lo seguro que son previendo por ejemplo ataques csrf [3], y finalmente por estar desarrollada en Python, que es el mismo lenguaje donde se van a desarrollar tanto las visualizaciones como sobre todo los algoritmos de Machine Learning.

Además, Django está actualmente siendo utilizado como el motor de aplicaciones web por infinidad de empresas como Pinterest o Instagram.

2.2.1 CREAR APLICACIONES Y CÓMO INICIAR EL SERVIDOR DJANGO

Una vez que se crea un proyecto de aplicación web con Django, este proyecto se puede dividir en diferentes aplicaciones que realizan diferentes servicios dentro de la aplicación web. Estas aplicaciones están interconectadas gracias al framework de trabajo de Django que, con este diseño modular, permite organizar mejor el proyecto, y, sobre todo, le aporta escalabilidad.

Una vez que ya esté lista la aplicación web construida con Django, es necesario arrancar el servidor que ejecutará la aplicación web. La forma en la que se arranca el servidor es introduciendo el siguiente comando en el directorio del proyecto donde se encuentra el archivo manage.py.

```
python manage.py runserver
```

Este comando arrancará el servidor en la dirección por defecto que es:

```
http://127.0.0.1:8000/
```

Una vez arrancado, podemos irnos a un navegador como Google Chrome, y poniendo esta dirección por defecto, accederemos a la aplicación web.

2.2.2 FUNCIONAMIENTO DE LAS APLICACIONES

Como se ha dicho antes, Django permite dividir el proyecto en diferentes aplicaciones que realicen distintas funciones para así dividir el proyecto entero en módulos.

La lógica de cada aplicación está en el archivo de views.py, que contiene funciones, también llamadas vistas por el framework de Django, que procesan cada una de las *requests* que hace el usuario a esa aplicación o módulo dentro del proyecto [4]. Estas vistas devuelven una respuesta web, que normalmente es un contenido HTML de Django llamado Template.

A los Templates, se les pueden pasar variables desde la vista en el archivo views.py para que el HTML acceda a ellas y proporcione una página web dinámica en función de los datos o variables que la aplicación le proporciona. Esta información se pasa al Template mediante un diccionario en Python que se llama *context* en el framework de Django.

2.2.3 TEMPLATES EN DJANGO

Los Templates en Django son archivos HTML que pueden tener acceso a las variables que la vista del archivo *views.py* les pasa, y, con estas variables, estos Templates permiten hacer bucles y condiciones if para modificar el contenido HTML y que se ajuste a los datos recibidos.

Además, los Templates en Django soportan la jerarquía de documentos, permitiendo que un Template herede de otro Template parte del código. La herencia en los Templates permite crear un documento base que contenga aspectos comunes en otros documentos, y definir qué partes pueden modificar otros Templates que hereden de él [5].

Así, se puede crear un Template que contenga el encabezado y el pie de página de todas las vistas dentro de una aplicación concreta, y otros Templates que hereden de éste y solo modifiquen cada una de las partes centrales de las distintas vistas. Esto aporta mucha modularidad y rapidez a la hora de crear Templates en Django.

2.2.4 ESTRUCTURA DE LOS MODELOS EN DJANGO

Finalmente, Django es un framework que soporta la programación orientada a objetos. Los objetos en Django se crean en los ficheros *models.py* dentro de cada aplicación.

Los objetos o modelos en Django pueden estar conectados entre sí, y tener relaciones entre ellos. Estos modelos se guardan en la base de datos que se ha especificado en la configuración del proyecto de Django, y cada uno de sus atributos son campos de la base de datos de ese objeto o clase.

2.3 ALGORITMOS DE MACHINE LEARNING

Existe una gran cantidad de algoritmos de Machine Learning. Estos se pueden dividir en aprendizaje supervisado, aprendizaje no supervisado, y aprendizaje por refuerzo. En esta plataforma sólo se han implementado algoritmos de aprendizaje supervisado, debido a que es lo que resulta más fácil e intuitivo para personas u organizaciones sin conocimiento sobre algoritmos de Machine Learning.

Se han implementado los siguientes algoritmos de machine learning utilizando las librerías de **Keras** y de **scikit-learn**.

2.3.1 REGRESIÓN LINEAL

La regresión lineal es utilizada en casos de regresión. Es el modelo de regresión más simple que existe pues no tiene hiperparámetros que elegir. Este modelo calcula los pesos que se le da a cada variable numérica independiente para crear una recta que prediga la variable dependiente, o variable a predecir, minimizando el error cuadrático medio (*mse* por sus siglas en inglés).

La regresión lineal no es enteramente un modelo de Machine Learning, pues como se ha dicho antes, no tiene ningún hiperparámetro. La regresión lineal es más un modelo estadístico que obtiene la línea que mejor se ajusta a la variable a predecir. En el proyecto se ha implementado con la librería scikit-learn [6].

2.3.2 REGRESIÓN LOGÍSTICA

La regresión logística se utiliza en casos de clasificación. Se ha implementado con la librería scikit-learn [7]. Los hiperparámetros más importantes de este modelo, e implementados en la plataforma son los siguientes:

- Solver: determina el algoritmo que utiliza el modelo para resolver el problema de optimización de la regresión logística.
  Los valores que se han utilizado en la plataforma son: "liblinear", "saga"
- C: su inverso determina la fuerza con la que se regulariza en el modelo.
  Los valores que se han utilizado en la plataforma son: valores decimales en el intervalo [0.001,10]
- Penalty: determina el tipo de regularización que se aplica en el modelo.
  Los valores que se han utilizado en la plataforma son: "l1", "l2", "none"

2.3.3 KNN

Este algoritmo de Machine Learning se utiliza tanto en regresión como en clasificación. Es un algoritmo simple que determina el valor de la salida en función de las observaciones que más cerca se encuentran en el conjunto de entrenamiento. Se ha implementado en el proyecto con la librería scikit-learn [8] y [9]. Los hiperparámetros del modelo más importantes e implementados en el modelo son los siguientes:

- N_neighbors: determina el número de observaciones que se tendrán en cuenta a la hora de hacer la estimación.
  Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo de [3, 40].
- Metric: determina cómo se computa la distancia con la que obtener qué observaciones son las que están más cerca de la nueva observación a ser predicha.
  Los valores que se han utilizado en la plataforma son: "euclidean", "manhattan", "minkowski".

- Weights: determina cómo se pondera el voto de cada una de las observaciones cercanas.
  Los valores que se han utilizado en la plataforma son: “uniform”, “distance”

2.3.4 ELASTICNETCV

Este algoritmo de Machine Learning se utiliza en problemas de regresión. Es un algoritmo muy similar al de la regresión lineal, solo que utiliza dos tipos de regularización, la L1, también llamada Ridge, y la L2, también llamada Lasso, con el objetivo de evitar el sobreajuste del modelo y mejorar su generalización. Se ha implementado con la librería scikit-learn [10]. Los hiperparámetros del modelo más importantes y los implementados en la plataforma son los siguientes:

- L1_ratio: determina qué combinación de la regularización L1 y L2 tiene el modelo.
  Los valores que se han utilizado en la plataforma son: valores decimales en el intervalo [0.0, 1.0].
- Eps: determina con qué precisión se debe alcanzar el mínimo del error del modelo con regularización.
  Los valores que se han utilizado en la plataforma son: valores decimales en el intervalo [0.001, 0.1].
- N_alphas: determina el número de valores que se utilizan en la búsqueda de hiperparámetros mediante la validación cruzada.
  Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo [50, 200].

2.3.5 SVC Y SVR

Estos algoritmos de vectores soporte, se utilizan en problemas de clasificación (Support Vector Classifier, o SVC) y regresión (Support Vector Regressor, o SVR). Son modelos de machine learning que buscan los hiperplanos que mejor separan las clases o que mejor se ajustan a los datos de entrenamiento utilizando vectores de soporte. Se ha implementado con

la librería scikit-learn [11] y [12]. Los hiperparámetros más importantes de ambos modelos y los implementados en la plataforma son los siguientes:

- Kernel: especifica el tipo de kernel que se va a utilizar en el modelo con el objetivo de transformar los datos en un espacio de mayor dimensión y así permitir la separación no lineal.
  Los valores que se han utilizado en la plataforma son: "poly", "rbf", "sigmoid", "linear".
- C: controla la penalización de los errores de clasificación del modelo.
  Los valores que se han utilizado en la plataforma son: valores decimales en el intervalo [0.001, 100].
- Gamma: determina cuánto influyen las observaciones cercanas al hiperplano a su definición.
  Los valores que se han utilizado en la plataforma son: "scale", "auto".
- Degree: determina el grado del polinomio de la función de kernel para la transformación de los datos.
  Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo [1, 7].

2.3.6 RANDOM FOREST

Estos algoritmos de Machine Learning se pueden utilizar tanto para regresión como para clasificación. Estos algoritmos combinan distintos árboles de decisión aleatorios mediante la técnica de Bootstrap [13] para producir una predicción robusta y precisa evitando el sobreajuste. Tanto el modelo de regresión como el de clasificación se han implementado con la librería scikit-learn, [14] y [15] respectivamente. Los hiperparámetros más importantes del modelo y los implementados en la plataforma son los siguientes:

- Criterion: determina la función de medida de calidad que se utilizará en el modelo para evaluar cada partición en cada árbol de decisión.

Los valores que se han utilizado en la plataforma son en clasificación son: “gini”, “entropy”. Por el contrario, en regresión se han utilizado los siguientes valores: “squared_error”, “absolute_error”, “friedman_mse”, “poisson”.

- N_estimators: determina el número de árboles de decisión que se construyen y que se combinan mediante la técnica de bootstrapping.
  Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo [50, 5000].
- Max_depth: determina la profundidad máxima de cada árbol de decisión. Este parámetro ayuda a luchar contra el sobreajuste de cada árbol de decisión.
  Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo [150, 1000].
- Min_samples_leaf: determina el número mínimo de muestras que debe haber en las hojas de decisión en los árboles que se combinan mediante boostrapping.
  Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo [1, 15].

2.3.7 REDES NEURONALES

Las redes neuronales sirven tanto para problemas de regresión como de clasificación. Actualmente existen muchísimas arquitecturas de redes neuronales, pero en este caso, por dar simplicidad y generalidad al problema, se ha elegido un modelo sequencial de Deep-Learning. Tanto el modelo de regresión como el de clasificación se han implementado con la librería keras, que bebe de tensorflow [16]. A continuación se exponen los hiperparámetros que se utilizan para entrenar estos modelos:

- Número de capas de deep learning: especifica cuántas capas de deep-learning tiene el modelo interconectadas completamente.
  Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo [3, 10].
- Número de neuronas en cada capa: determina cuántas neuronas tiene cada capa de deep-learning en el modelo.

Los valores que se han utilizado en la plataforma son: valores enteros en el intervalo [60, 200].

- Función de activación en cada capa: determina la función de activación que tiene cada capa de deep-learning en el modelo para dar no-linealidad.
  Los valores que se han utilizado en la plataforma son: “relu”, “selu”, “softmax”, “tanh”.
- Optimizador del modelo: especifica el algoritmo con el que el modelo intenta encontrar el mínimo error
  Los valores que se han utilizado en la plataforma son: “Adam”, “Nadam”.

2.4 OPTUNA

Este proyecto utiliza Optuna [17] para la búsqueda de los hiperparámetros de los modelos mencionados anteriormente. Optuna es una librería de código abierto de optimización de hiperparámetros que permite realizar búsquedas adaptativas de hiperparámetros en un espacio de alta dimensionalidad. Esto ayuda a encontrar los hiperparámetros óptimos de una forma más rápida que una simple búsqueda aleatoria, como implementa la famosa función GridSearch de la ya mencionada librería de scikit-learn.

2.5 PLOTLY

Este proyecto utiliza la librería de código abierto Plotly [18]. Plotly permite crear gráficos interactivos y personalizados con el objetivo de que puedan ser comprendidos por los usuarios de esta plataforma.

Capítulo 3. ESTADO DE LA CUESTIÓN

En los últimos años ha habido una revolución en el mundo de la Inteligencia Artificial (AI), en el que numerosas empresas se han sumado a esta revolución de la AI. Las empresas más grandes del planeta como Facebook (Meta) [19], Apple [20], Microsoft [21], Amazon [22], Google (Alphabet) [23], (conocidas como MAMAA por sus siglas en inglés) han sacado productos que o facilitan la utilización de Machine Learning por los usuarios, u ofrecen servicios basado en Machine Learning.

Además, esta revolución no parece de que se vaya a detener. OpenAI, la empresa que está detrás del famoso modelo de lenguaje, ChatGPT, en su paper *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models* [24] especifica que hasta el 80% de todos los trabajadores de Estados Unidos verán afectado su trabajo en un 10% por modelos de Machine Learning. Además, casi el 20% de los trabajadores verán afectado su trabajo en un 50% por estos modelos.

3.1 CAMBIO DE PARADIGMA DE VENDER PRODUCTOS A SERVICIOS

En los últimos años ha habido un cambio de paradigma que consiste en que muchas empresas están migrando de un modelo de negocio orientado a la venta de productos a uno orientado a la venta de servicios [25].

Este modelo de negocio está triunfando porque beneficia tanto a la parte que los ofrece como a la parte que los contrata.

La parte que ofrece estos servicios está beneficiada porque los servicios de subscripción ofrecen ingresos de forma continua mes a mes. Estos ingresos recurrentes son beneficiosos para las empresas ofertantes pues: A) son más estables que los ingresos derivaos de la venta de productos, pues el subscriptor al suscribirse al servicio acepta que esta suscripción se prolongue indefinidamente hasta que él tome la decisión de darse de baja de la misma; B)

permiten visualizar los ingresos a una mayor distancia en el tiempo, ayudando así en la planificación de recursos; y C) ayudan a cubrir gran parte de los costes fijos que supone una gran parte de los costes en el mundo de la informática, como puede ser el tiempo de cómputo por tener encendido el servicio.

La parte contratante de estos modelos de negocio orientados como servicios es beneficiada pues delega la implementación y el mantenimiento del servicio, que para ellos tendría un coste muy elevado como puede ser el personal técnico o el desarrollo en la propia compañía de los servicios que se pueden contratar. Además, desde el punto de vista del contratante de estos servicios, sólo se seguirá adelante con el contrato mientras éste proporcione un ahorro al contratante, lo que incentiva a contratar estos servicios.

Además, se espera que este "boom" en el modelo de negocio "Anything as a service" (XaaS) siga creciendo a un ritmo por encima de un 20% anualmente según la firma Fortune Business Insights [26].

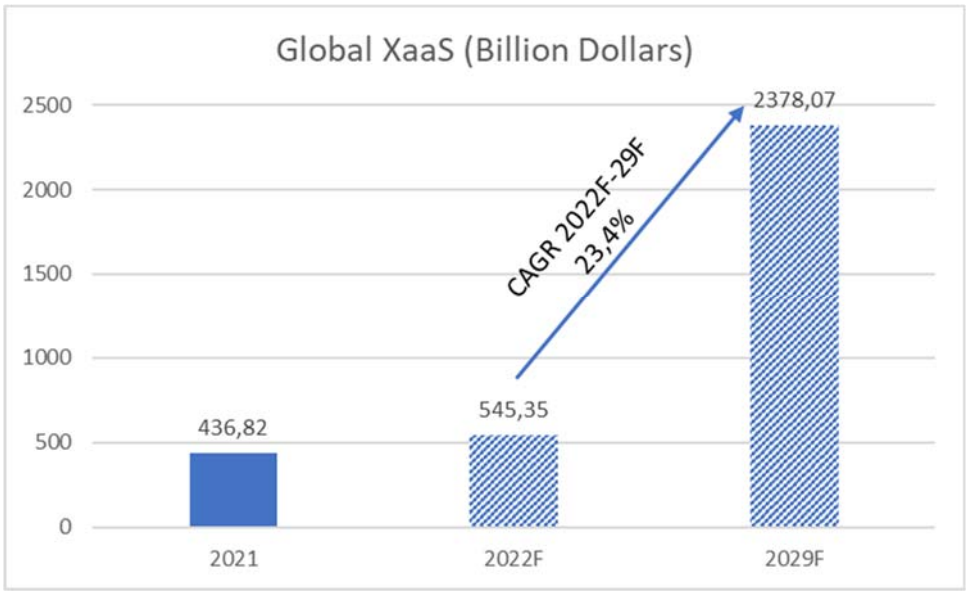


Figura 1. Tamaño de mercado "XaaS" en miles de millones de dólares 2021-2028F [26]

3.2 MACHINE LEARNING AS A SERVICE

3.2.1 SITUACIÓN DEL MERCADO

Dentro de la explosión de los modelos de negocio como un servicio ("as a service" en Inglés), el Machine Learning as a Service (MLaaS) es de los sectores que más se espera que crezca, aumentando más de un 40% en facturación según un estudio de Data Bridge Market Research [27].

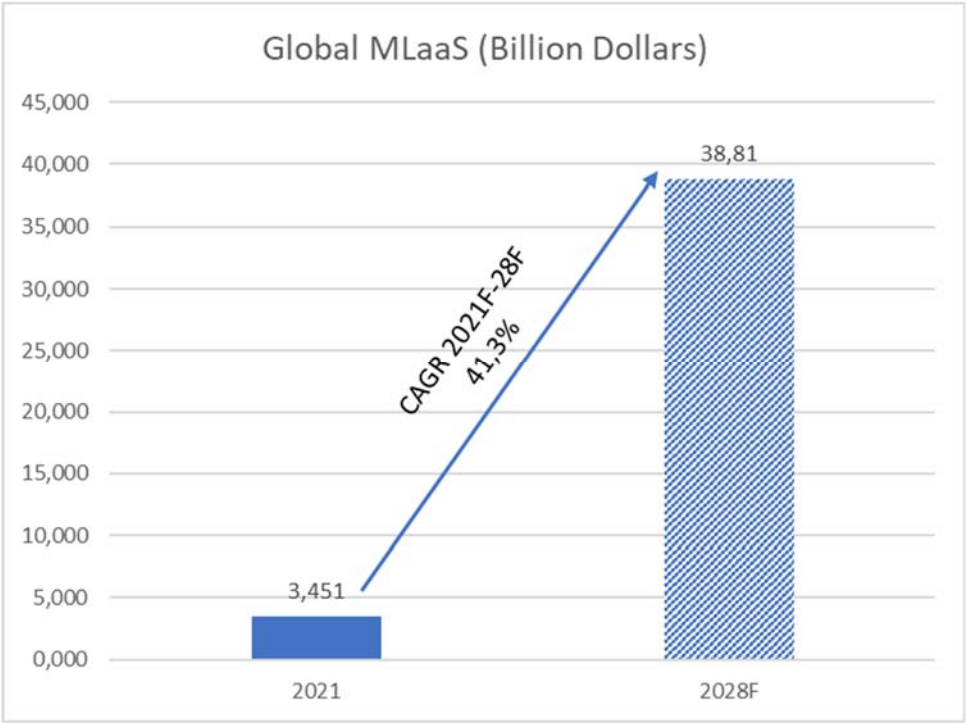


Figura 2. Tamaño de mercado MLaaS en miles de millones de dólares 2021-28F [27]

Esto nos lleva a la conclusión de que se espera que en el futuro haya mucha competencia en este sector, y a la vez mucha demanda por estos servicios.

Por esto, es el momento de crear una plataforma sencilla pero potente que permita a los usuarios, independientemente de su cualificación, acceder a estos servicios que van a estar tan demandados en el futuro. Además, el Machine Learning as a Service puede ahorrar mucho dinero a las empresas, a las que puede resultar muy costoso implementar ellas mismas

estas técnicas de machine learning contratando a un científico de datos (*data scientist*), que según Jobted, tiene de media un sueldo cercano a los 39.000€ brutos al año [28].

3.2.2 PRINCIPALES COMPETIDORES

Los principales competidores actuales del Machine Learning as a Service son empresas punteras tecnológicamente. Estas empresas, con su gran infraestructura de servidores montada en todo el planeta, sus equipos de científicos de datos e ingenieros, y su posición como líderes en el mercado, les permiten ofrecer muy buenos servicios a bajo coste.

Los principales competidores son:

- **Amazon Web Services (AWS)**: de todos los productos que ofrece AWS relacionados con el Machine Learning, Amazon SageMaker es el que más similitudes tiene con el proyecto a desarrollar. SageMaker [29] simplifica la forma en la que se entrenan los modelos, y automáticamente busca los hiperparámetros óptimos para el modelo a entrenar. Además, una vez está el modelo entrenado, se puede descargar para un uso fácilmente.

  SageMaker tiene muchos algoritmos potentes de Machine Learning como Linear Learner, Factorization Machines, XGboost, K-Means, PCA, Random Cut Forest; además de tener algoritmos de reconocimiento de imágenes; de series temporales; y de procesamiento del lenguaje.

  La forma de interactuar con este servicio de Amazon es a través de ficheros de trabajo (notebooks en inglés) que se conectan con el servidor de Amazon para correr y entrenar el código, como el que se ve en la Figura 3.

  Aunque SageMaker permite descargar ficheros de trabajo o *notebooks* parcialmente rellenos de código para según qué casuísticas con el fin de que el usuario tenga que escribir el mínimo código, es el usuario el que tiene que elegir el algoritmo de Machine Learning que quiera entrenar utilizando código en el notebook de SageMaker.

  Igualmente, es el usuario el que proactivamente tiene que elegir, con conocimientos de algoritmos de Machine Learning, el error que quiera minimizar. Esto aunque

ofrece mucha personalización, puede resultar complicado para usuarios no avanzados en el mundo de la programación.

Además, SageMaker no tiene una parte de preprocesamiento de datos, y AWS espera que el usuario lo realice en el notebook mediante la programación antes de entrenar sus potentes modelos.

También, SageMaker no te permite hacer visualizaciones de los datos limpios (EDA) que permita al usuario entender mejor cómo se comportan sus datos de una manera fácil y sencilla.

Asimismo, SageMaker no facilita la posibilidad de crear Ensembles, o de juntar los modelos predictivos entrenados en uno solo mejorando las predicciones. Esto debe hacerse como lo haría cualquier programador fuera de SageMaker.

Finalmente, los costes de utilizar estos ficheros de trabajo, cuesta, con 4 CPUs y 8 GB de RAM 0.242 USD por hora.

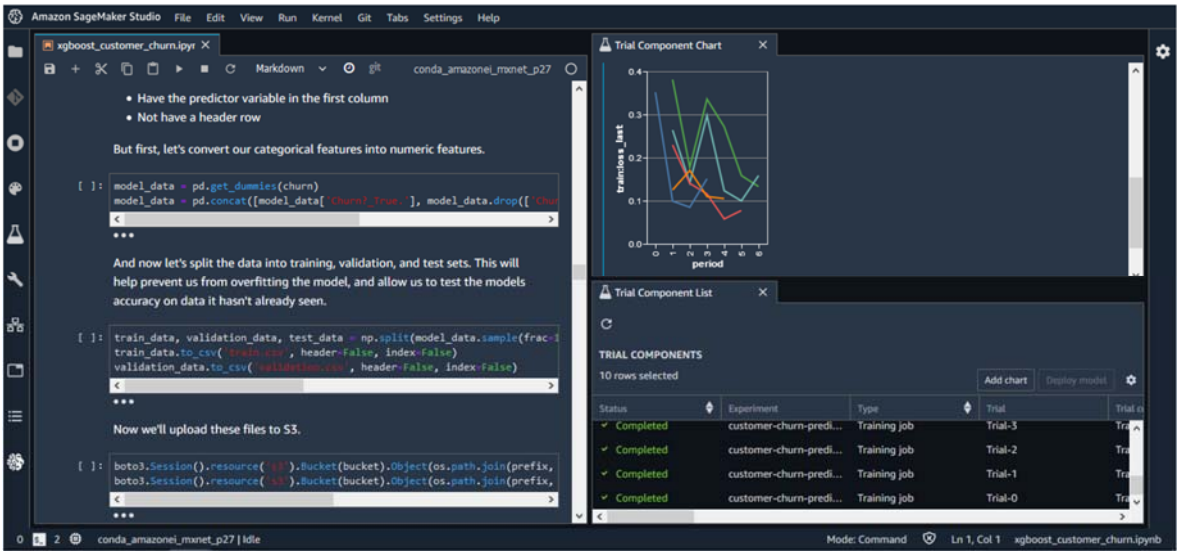


Figura 3. Fichero de trabajo en SageMaker [29]

- **Google:** VertexAI [30], lanzada en 2022, permite a los desarrolladores entrenar modelos de Machine Learning, proporcionando una solucion No-Code, es decir, sin necesidad de programación, en la que los usuarios de la plataforma pueden desplegar

"bloques" sobre un entorno de trabajo que esconden código tras de ellos. Así, se elimina el requisito de saber programar para el usuario inexperto.

El problema con la solución No-Code propuesta por Google en VertexAI es que el usuario tiene que saber qué etapas tiene la creación de modelos de predicción, como limpieza de datos, entrenamiento y búsqueda de hiperparámetros, y evaluación de los modelos. Así, Google requiere en VertexAI, dando versatilidad al usuario, que el usuario seleccione qué etapa quiere llevar a cabo en cada momento.

Además, VertexAI, al igual que SageMaker, no proporciona ningún EDA sobre los datos de entrenamiento, ni proactivamente ninguna una forma en la que limpiar los datos. No obstante, VertexAI, a diferencia de SageMaker, sí que permite crear Ensembles de los modelos ya entrenados.

Finalmente, en VertexAI se espera que el usuario final decida por adelantado qué modelos y algoritmos va a entrenar, lo que dificulta bastante la usabilidad por personas sin conocimientos técnicos.

VertexAI tiene un coste de 21.25 USD por hora por cada de cada CPU para entrenar modelos, y para predecir, 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.

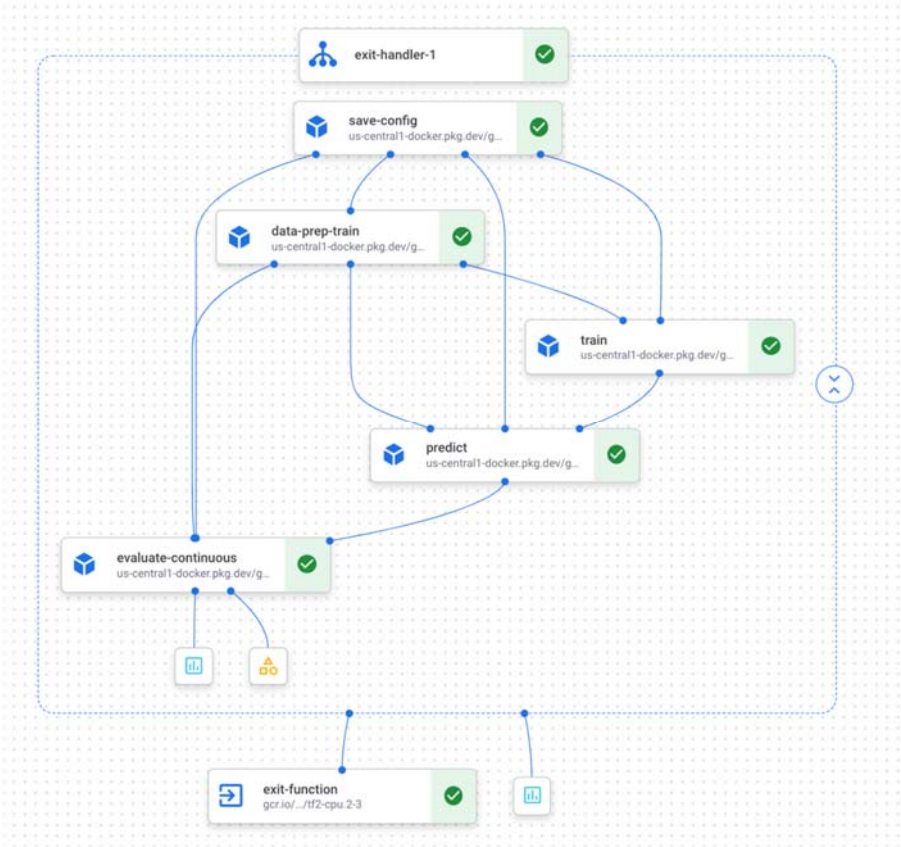


Figura 4. Entorno de trabajo en VertexAI [31]

- **Microsoft**: Microsoft Azure [32] es otro de los competidores identificado.

  Al igual que VertexAI de Google, Microsoft Azure también está basado en una solución No-Code que acerca la forma de crear modelos a la gente sin experiencia en el mundo de la programación. Esta solución No-Code también está basada en un sistema de bloques en la que el usuario elige qué partes del proceso de creación de modelos decide hacer (como el dividir los datos en datos de entrenamiento y datos de prueba). Además, el usuario, como en VertexAI, tiene que determinar qué modelos quiere entrenar por adelantado.

  También, Microsoft Azure no proporciona ninguna herramienta o interfaz de visualización interactiva de los datos, para realizar la función de EDA (Exploratory Data Analysis), con el objetivo de que el usuario entienda cómo se relacionan sus

datos. De la misma manera, Azure tampoco permite de una forma fácil la creación de Ensembles de una forma fácil y sencilla.

Finalmente, Azure tiene unos costes de 0.24 USD por hora cuando se utilizan 4 CPUs y 16 GB de RAM.

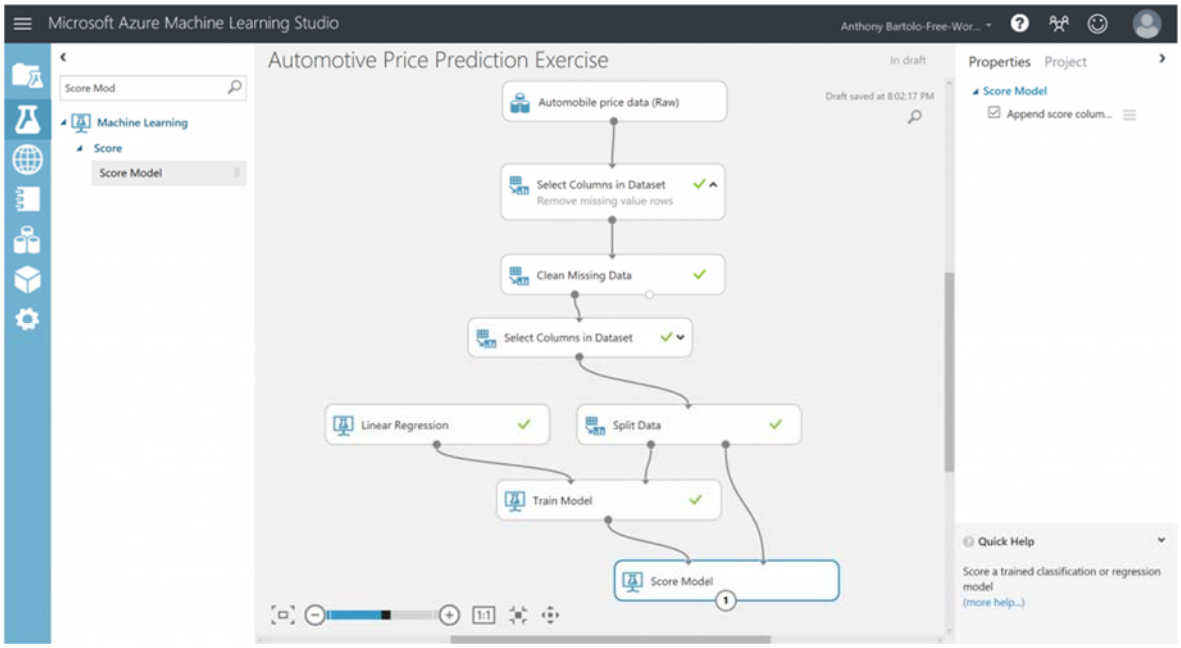


Figura 5. Entorno de trabajo en Microsoft Azure [33]

- **IBM**: IBM Watson ofrece muchos servicios de machine learning complejos como pueden ser el reconocimiento visual o el NLP (clasificación del lenguaje natural por sus siglas en inglés). No obstante, el servicio que más se parece al propuesto en este proyecto es IBM Watson Studio [34].

  Al igual que Azure y VertexAI, IBM Watson Studio es una solución No-Code basada en bloques que acerca el mundo del Machine Learning a los usuarios sin conocimientos de programación, pero que sí que poseen conocimientos de machine learning, pues su mecanismo basado en bloques contiene los mismos problemas que Azure y VertexAI.

  Una de las cosas que sí que implementa IBM Watson, es una interfaz para poder visualizar los datos subidos de una forma muy intuitiva (EDA).

IBM Watson no tiene un apartado de limpieza de los datos que ayude al usuario inexperto sobre cómo debería tratar los datos que le faltan en su base de datos.

IBM Watson Studio tampoco divide los datos automáticamente en datos de entrenamiento y datos de prueba a la hora de entrenar modelos de predicción.

Finalmente, los costes de IBM Watson Studio, por 4 CPUs y 16 GB de ram son 1.9 USD por hora.

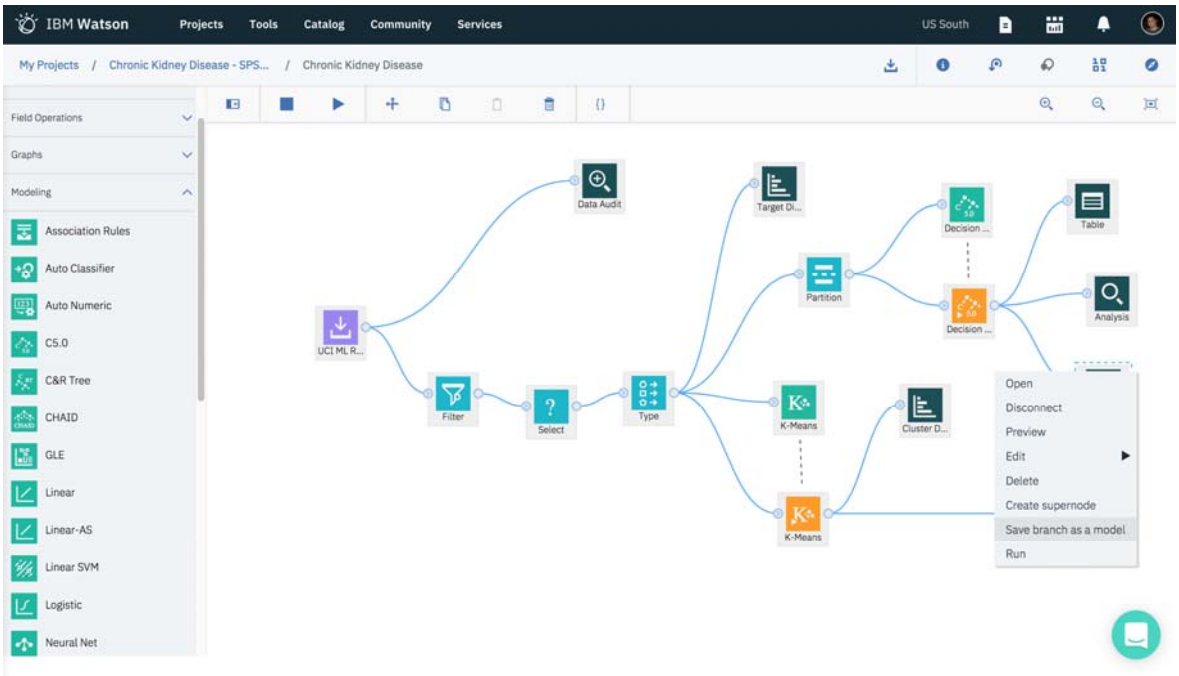


Figura 6. Entorno de trabajo en IBM Watson Studio [35]

- **WEKA**: WEKA [36] es un proyecto de la universidad de Waikato y a diferencia del resto de competidores, no se ejecuta en la nube, sino que se ejecuta en el propio ordenador del usuario. WEKA es una aplicación Java en la que se permite entrenar algoritmos de Machine Learning tanto de clasificación, como de regresión, y de clustering complejos. De esta manera, WEKA no es un servicio de Machine Learning as a Service, pues es gratuito y se ejecuta en los ordenadores de los usuarios que se hayan descargado este programa.

Su interfaz de usuario es muy pobre y no es intuitiva. Pese a eso, establece un orden al usuario a la hora de crear sus modelos de predicción que el resto de los competidores no hace, lo que acerca el proceso al usuario inexperto.

Además, proporciona proactivamente al usuario una visualización, aunque muy pobre, sobre los datos con los que se va a entrenar el modelo de predicción.

Uno de los problemas de WEKA es que el usuario tiene que ir creando uno a uno los modelos de predicción, y no permite hacer una comparación en conjunto de todos ellos para ver cuál es el mejor.

Finalmente, WEKA no tiene coste ninguno pues es una herramienta gratuita.

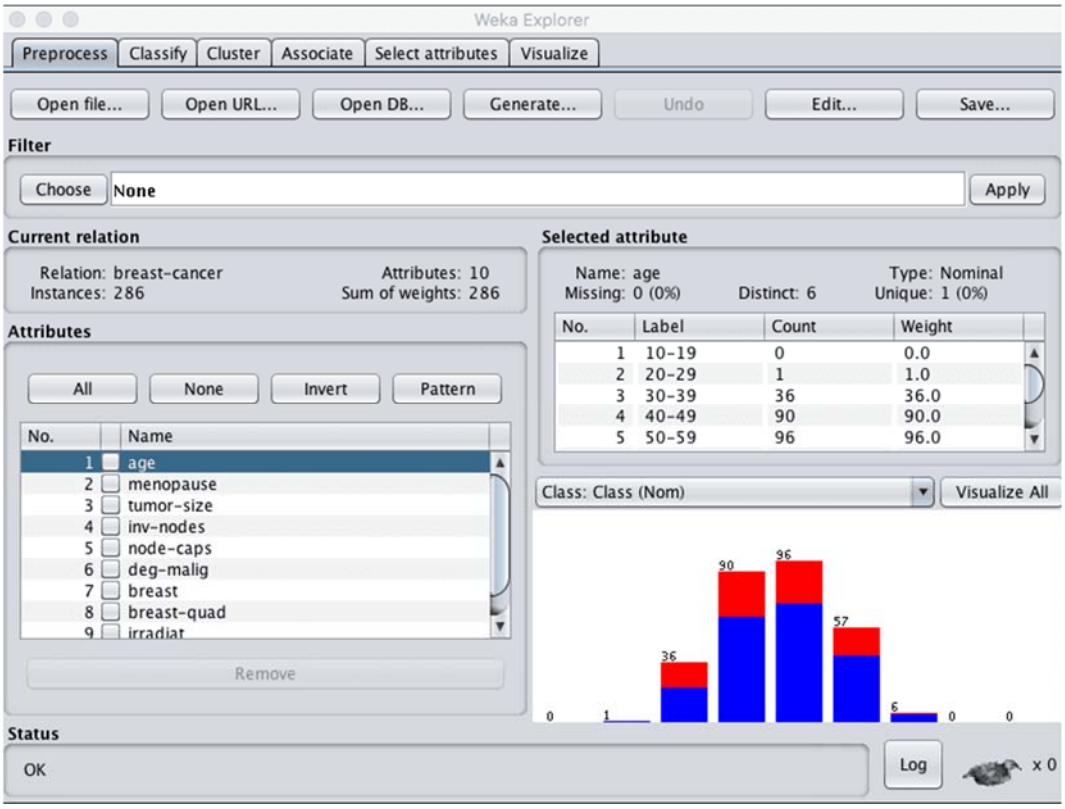


Figura 7. Entorno de trabajo en WEKA [37]

3.2.3 CONCLUSIÓN

Después de haber cubierto a los competidores que también ofrecen plataformas de Machine Learning as a Service podemos elaborar la siguiente tabla comparándolos.

	Amazon SageMaker	Google VertexAI	Microsoft Azure	IBM Watson Studio	WEKA	**Proyecto propuesto**
Uso sin conocimientos de programación	✗	✓	✓	✓	✓	✓
Uso sin conocimientos de Machine Learning	✗	✗	✗	✗	✗	✓
Facilidad de uso	✗	✓	✓	✗	✗	✓
Preprocesado de datos	✗	✗	✗	✗	✗	✓
EDA incorporado	✗	✗	✗	✗	✓	✓
Comparar diferentes modelos	✗	✗	✗	✗	✗	✓
Coste	✓	~	✓	~	✓✓	✓✓
Algoritmos implementados	✓✓	✓✓	✓✓	✓✓	✓✓	✓
Posibilidad de crear Ensembles	✗	✓	✗	✗	✗	✓

Tabla 1. Comparación entre los competidores identificados y la plataforma propuesta

Como se puede ver, el proyecto propuesto está destinado a los usuarios sin conocimientos de programación ni conocimientos de Machine Learning, de una manera fácil e intuitiva.

También el proyecto debe ayudar al usuario a preprocesar los datos, a visualizarlos con un EDA, a entrenar modelos sin ningún conocimiento de Machine Learning, poder compararlos, y ofrecer la posibilidad de crear ensembles.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Como se ha especificado en el apartado anterior, estamos en un momento en el que los servicios de Machine Learning as a Service van a experimentar un crecimiento por encima del 40% anualmente en los próximos 10 años.

Además, tras el estudio de la competencia, se ve en la tabla final del Estado de la Cuestión que existe la necesidad de una plataforma que permita utilizar algoritmos de Machine Learning para personas que no estén experimentadas en técnicas de machine learning y que contenga una interfaz intuitiva que ayude al usuario inexperto en el entrenamiento de los modelos de machine learning.

Aunque la mayoría de los competidores proporcionan soluciones No-Code, en ellos se requiere que el usuario sepa el orden en el que se debe organizar un proyecto de machine learning (preprocesado – visualización – entrenamiento - valoración). Además, los competidores identificados requieren que el usuario seleccione de antemano qué algoritmo de machine learning quiere entrenar de todos los que tienen disponibles. Esto, aunque facilita el uso de estas herramientas, no permite que usuarios no experimentados las utilicen.

Muchos usuarios que no conocen las fases por las que ha de pasar un proyecto de machine learning, necesitan inicialmente que se les indique qué fases y en qué orden tienen que realizarse. Pero no basta con pararse ahí, pues el usuario inexperto no tiene conocimientos sobre el cómo llevar a cabo estas fases del proceso de entrenamiento de modelos de machine learning.

Inicialmente, en la etapa de preprocesado, la plataforma debería proponer al usuario maneras para preprocesar sus datos en función de cómo se distribuyan estos, como, por ejemplo, qué hacer con los datos que faltan. Pese a que estas recomendaciones sobre el preprocesado de

los datos no serán mejores que las que realice un científico de datos, para el usuario inexperto, sí que puede ser una buena ayuda o aproximación a tal trabajo.

Aparte de la etapa de preprocesado, muchas plataformas actualmente no ofrecen visualizaciones interactivas de los datos (EDA, por las siglas en inglés de Exploratory Data Analysis), en las que el usuario pueda visualizar todos sus datos antes de entrenar los modelos de aprendizaje automático. Esta tarea es realizada por todos los científicos de datos para poder ver comportamientos de los datos que van a entrenar los modelos en la plataforma. Así, como esta herramienta no está implementada en los actuales competidores identificados, este proyecto espera implementarlo para ayudar a los usuarios inexpertos ofreciéndoles una herramienta interactiva para visualizar sus datos.

Otra de las etapas importantes del proceso de creación de modelos de machine learning es comparar varios modelos ya entrenados con un conjunto de datos de prueba para valorar cuál funciona mejor. Ninguna de las herramientas en el mercado actualmente permite comparar diferentes modelos de machine learning entrenados sobre el mismo conjunto de datos de una manera fácil y en un solo vistazo.

Asimismo, otra característica importante es la facilidad de uso y tener una interfaz de usuario amigable que facilite el entrenamiento de modelos de conocimiento y la predicción de nuevas observaciones. Esta interfaz amigable se da en algunos competidores identificados, como Vertex AI, Microsoft Azure, o IBM Watson Studio. No obstante, en estos casos esta interfaz no guía al usuario proponiendo siguientes etapas del proceso de creación de modelos.

Finalmente, los algoritmos de machine learning se comportan de forma muy distinta sobre un mismo conjunto de datos, prediciendo algunos datos mejor que otros. Una salida muy común de muchos científicos de datos es crear Ensembles, que es juntar diferentes modelos con distintas capacidades para crear modelos más robustos, en los que los errores de uno sean compensados con aciertos de otro modelo y viceversa. Como se ha visto antes en la tabla del capítulo 3, solo Vertex AI permite crear Ensembles con sencillez, y este proyecto pretende permitir crear Ensembles de una forma sencilla para el usuario inexperto.

Finalmente, como se ha visto anteriormente, se espera un crecimiento casi exponencial del uso de las técnicas de machine learning en el mundo empresarial, y aunque el precio de algunas empresas identificadas es muy competitivo, este proyecto tiene como objetivo brindar este servicio sin coste ninguno, alineándonos con los ODS de la ONU, favoreciendo el crecimiento económico de todo tipo de empresas, también las que no disponen de muchos recursos.

4.2 OBJETIVOS

Una vez conocido el actual estado del arte y el crecimiento esperado en el Machine Learning as a Service se puede enunciar que objetivo principal del trabajo que aquí se describe es el ***desarrollo de una plataforma de extracción de conocimiento a partir de datos arbitrarios "Machine Learning as a Service***".

Este objetivo, se puede dividir en los siguientes subobjetivos:

- La plataforma a desarrollar tendrá una interfaz de usuario sencilla e intuitiva. La experiencia de usuario debe estar muy cuidada para que los usuarios vean igual de intuitivo crearse una cuenta y acceder a la plataforma, que crear modelos de machine learning y realizar todas las fases asociadas a este proceso de aprendizaje automático.
- La creación de proyectos de machine learning asociados a un conjunto de datos debe ser muy sencilla, así como la visualización de todos los proyectos que tiene un usuario.
- Los usuarios no deberían acceder a los proyectos de otros usuarios, y por lo tanto garantizar su privacidad.
- Dentro de un proyecto de machine learning, el proceso a seguir y las fases a realizar debe estar estipulado por la plataforma mediante el cual el usuario tenga que recorrer cada paso necesario sin que tenga que decidir el usuario proactivamente qué fases completar en cada momento.
- La plataforma deberá recomendar al usuario proactivamente cómo limpiar sus datos si éstos contienen datos que faltan.

- El usuario debe poder visualizar los datos con los que va a entrenar los modelos de machine learning en el EDA de una manera intuitiva, interactiva y sencilla.
- Debe existir una forma de entrenar todos los modelos de forma automática para que el usuario no necesite elegir los modelos a entrenar por adelantado, ni necesite saber cómo entrenarlos. Así, la plataforma debe poder entrenar automáticamente todos los modelos, buscando óptimamente los mejores hiperparámetros para cada modelo.
- En el caso de que nos encontremos con un usuario más experimentado, la plataforma debe ofrecer la posibilidad de que este usuario elija los modelos que quiere entrenar, y si así lo desea, también los hiperparámetros de estos modelos.
- La plataforma, una vez entrenados algunos modelos de machine learning debe ofrecer al usuario una página para visualizar y comparar el comportamiento de sus modelos.
- Asimismo, la plataforma, con los modelos entrenados, debe ofrecer al usuario la posibilidad de crear Ensembles, es decir, juntar diferentes modelos de machine learning, de una forma fácil y automatizada.
- Finalmente, a la hora de predecir sobre un nuevo conjunto de datos, la plataforma debe devolver el archivo subido con las predicciones en otra columna del archivo, a la vez que trata los nuevos datos como los preprocesados (rellenando algunos datos, o obviando algunos otros por falta de datos)

Todos estos objetivos se pueden clasificar en las siguientes 7 categorías, que corresponden con las partes en las que se divide la plataforma.

I. Interfaz de usuario intuitiva y amigable
II. Autenticación de usuarios a la vez que se garantiza su privacidad
III. Creación y visualización de proyectos de entrenamiento de modelos
IV. Preprocesado y análisis de datos automáticos
V. Creación de modelos y de los ensembles
VI. Visualización y comparación de los modelos
VII. Predicción sobre nuevos datos

4.3 METODOLOGÍA

Existen muchas metodologías a la hora de desarrollar un proyecto. Estas metodologías ayudan a dar un enfoque estructurado a la planificación de las tareas a realizar, así como cómo ejecutarlas y cómo darlas por terminadas. Utilizando estas metodologías, es más fácil aumentar la predictibilidad de los resultados que tendremos en cada una de estas etapas, mientras se optimiza el tiempo que se invierte en su desarrollo. Otra de las características de las buenas metodologías de trabajo es que permiten realizar entregables de pequeñas partes independientes del resto del proyecto, y así, utilizar la retroalimentación continua para ir mejorando y puliendo cada una de las tareas realizadas.

Como se ha dicho anteriormente, existen muchas metodologías para llevar a cabo proyectos de programación, aunque las dos más conocidas y utilizadas son Waterfall y Agile. Waterfall tiene un enfoque más basado en el desarrollo lineal, en el que no se puede mover a una siguiente tarea hasta que ésta esté terminada completamente, pues esta metodología asume que las siguientes tareas dependen de la anterior. Por el contrario, Agile tiene una metodología mucho más flexible que Waterfall y asume que todas las tareas son entornos en constante evolución que pueden ser mejorados mediante la retroalimentación.

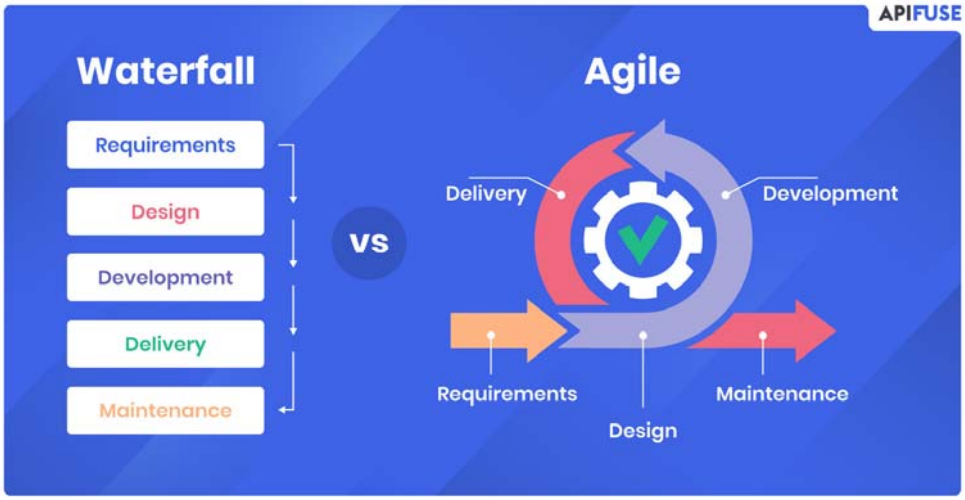


Figura 8. Comparación entre las metodologías Waterfall y Agile [38]

Como este proyecto tiene 7 grandes partes (mencionadas anteriormente en los objetivos), y muchas de estas partes son independientes entre ellas, se ha optado por una metodología Agile, en la que se van mejorando cada una de estas 7 grandes partes pudiendo trabajar paralelamente en varias partes del proyecto.

4.4 PLANIFICACIÓN

En la planificación de esta plataforma de Machine Learning as a Service orientada hacia usuarios no experimentados, se ha optado por un método Agile. El método Agile funciona con Sprints, que son iteraciones de cada una de las tareas en las que se divide el proyecto. Los diagramas Gantt de a continuación muestran la planificación propuesta para el proyecto. Desde la definición de la idea, hasta el desarrollo de la misma, pasando por la recopilación de información y la formación en las tecnologías implementadas durante el proyecto.

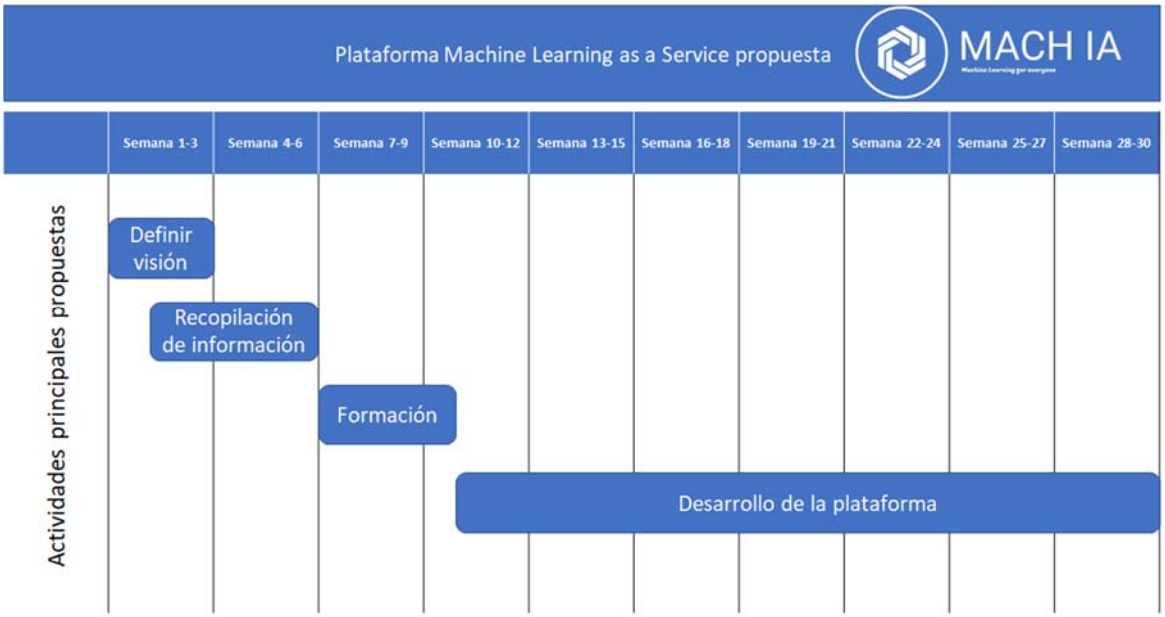


Figura 9. Cronograma general propuesto

Además, dentro del desarrollo de la plataforma podemos ver cómo se han planificado el desarrollo de cada una de las tareas a lo largo del tiempo.

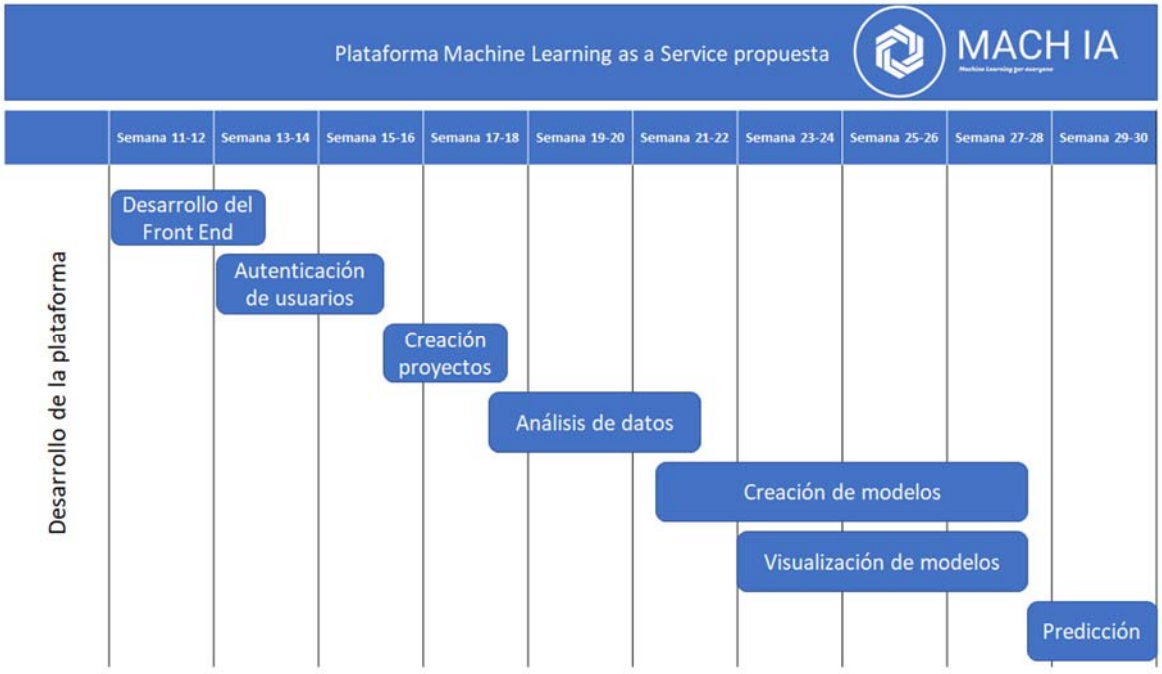


Figura 10. Cronograma del desarrollo de la plataforma

Capítulo 5. PLATAFORMA DESARROLLADA

Esta plataforma de Machine Learning as a Service orientada para los usuarios sin ningún conocimiento de análisis de datos, está desarrollada con el framework (o entorno de trabajo) web de Django. Django es un framework open-source basado en el lenguaje de programación Python. Como se ha dicho anteriormente, Django permite dividir un proyecto en muchas aplicaciones, facilitando la modularización la plataforma. La plataforma desarrollada tiene varias aplicaciones que realizan los distintos servicios dentro de toda la aplicación web. Éstas son:

I. Landing Page
II. Autenticación
III. CrearAbrirProyectos
IV. AnalizarDatos
V. CrearModelos
VI. VisualizarModelos
VII. Predicción

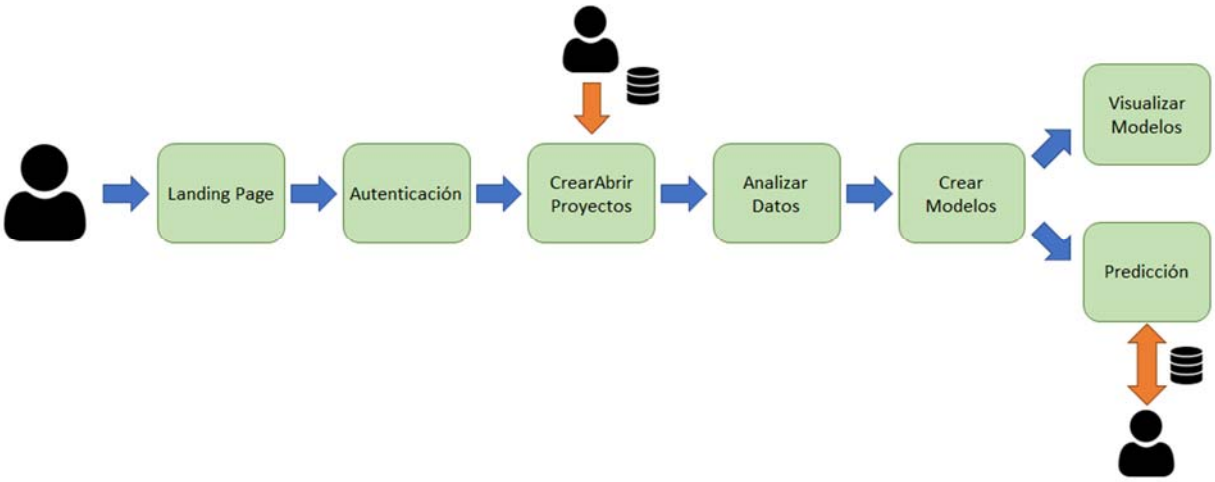


Figura 11. Diagrama de las aplicaciones de la plataforma desarrollada

Asimismo, esta plataforma utiliza bases de datos para almacenar la información tanto de los usuarios, como de los proyectos creados, como de los modelos de machine learning entrenados. La siguiente figura explica la relación entre las bases de datos de los objetos de esta aplicación.

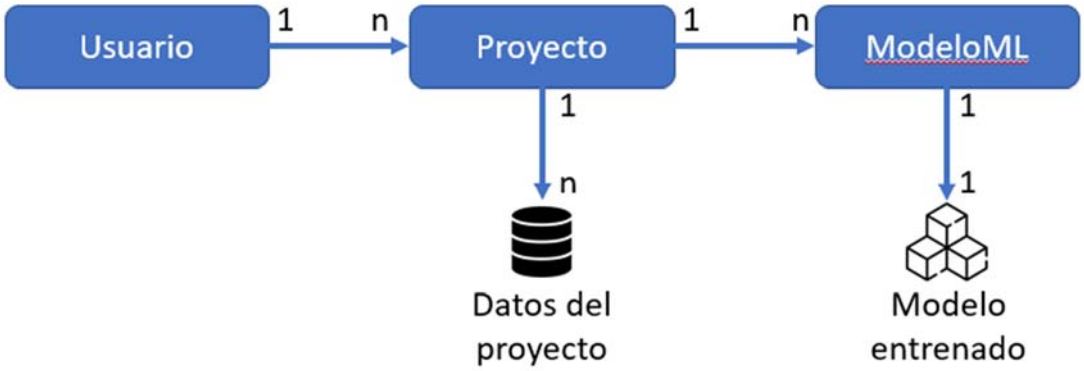


Figura 12. Diagrama de la relación entre las clases en la plataforma desarrollada

La clase "Usuario" está definida en la aplicación de Autenticación. Los usuarios tienen la capacidad de crear proyectos de Machine Learning, para los que se ha creado la clase "Proyecto" definida en la aplicación de *Crear Abrir Proyectos*. De esta manera, existe una relación entre los usuarios y los proyectos, donde un usuario puede tener múltiples proyectos de machine learning asociados.

Los proyectos de Machine Learning almacenan diversos datos en la base de datos, incluyendo el conjunto de datos con el que se entrenarán los modelos de machine learning proporcionado por el usuario al crear el proyecto, así como el tratamiento de los datos faltantes, la normalización de las variables continuas y la codificación de las variables categóricas.

Por último, un Proyecto de Machine Learning puede entrenar varios Modelos de aprendizaje automático, representados por la clase ModeloML, lo que establece una relación donde un proyecto puede tener múltiples modelos de machine learning. La clase "ModeloML" está definida en la aplicación de CrearModelos y guarda tanto el modelo entrenado como diversas métricas e información relacionada al mismo.

A parte de estas siete aplicaciones, el proyecto tiene una carpeta que controla el proyecto entero, llamada TFGTelecoMLaaS. A continuación, se explicará la utilidad de cada una de estas aplicaciones y su funcionamiento en más detalle.

Todo el código de la plataforma puede encontrarse en el siguiente repositorio de GitHub: https://github.com/Divasson/TelecoMLaaS

5.1 CARPETA DEL PROYECTO

La carpeta del proyecto, con nombre TFGTelecoMLaaS, une todas las aplicaciones que tiene la plataforma. Los dos archivos importantes que controlan y juntan todo el proyecto son *settings.py* y *urls*.py.

El archivo *urls.py* de la carpeta general del proyecto tiene como finalidad determinar qué aplicaciones se encargan de qué rutas URL. Así, este archivo redirige a cada una de las distintas aplicaciones las llamadas a según qué rutas URL. Posteriormente, serán los archivos *urls.py* de cada una de las diferentes aplicaciones del proyecto, los que determinen a qué función del archivo *views.py* de la misma aplicación redirigir cada una de las llamadas en función de su ruta URL. Finalmente, cada vista o función del archivo *views.py* devuelve una página web al usuario mediante los Templates de Django, a los que se les pasa información desde estas funciones, en lo que en Django se llama contextos.

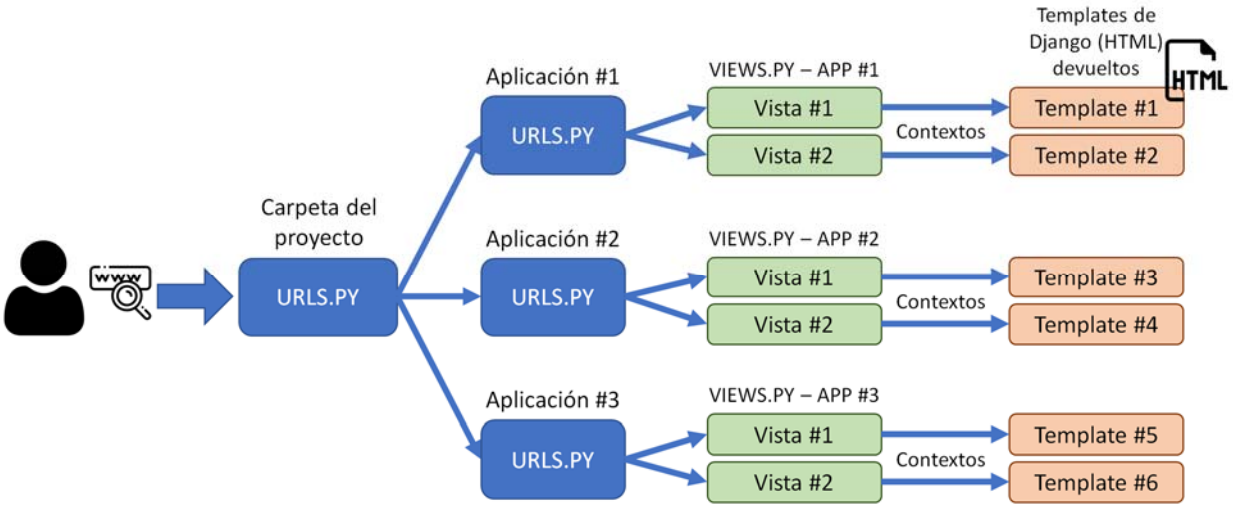


Figura 13. Funcionamiento de Django las vistas a mostrar en función de la URL

Este proyecto, siguiendo la configuración predeterminada de Django, se inicia en la dirección web http://127.0.0.1:8000/. De esta manera, las rutas URL que utilizará el archivo *urls.py* de la carpeta del proyecto son las que se agreguen justo después de la dirección web en la que se está ejecutando el proyecto.

En el archivo *settings.py*, en la variable global del proyecto *INSTALLED_APPS*, se listan los módulos o aplicaciones que pertenecen al proyecto y que por lo tanto deben cargarse. Además, se especifica en qué carpeta tiene que buscar los Templates de Django, que en este caso es la carpeta de Templates de la aplicación *Landing Page*. Finalmente, en este archivo se configura que la clase que va a utilizar esta plataforma para autenticar a los usuarios es la definida en la aplicación *Autenticación*, sobrescribiendo así a la clase de Usuario de Django por defecto.

5.2 LANDING PAGE

La aplicación *Landing Page* consiste en una página web de inicio sencilla. La ruta URL de esta aplicación es la dirección vacía, lo que significa que es la página predeterminada cuando

accedemos a nuestra plataforma. Al ingresar a esta aplicación, se muestra en pantalla la página web de inicio.

Tanto esta página web de inicio, como el resto de las páginas web, se han desarrollado utilizando un diseño descargado de https://bootstrapmade.com/. Todos los archivos necesarios para que este diseño tenga la misma apariencia y funcionalidad, como los archivos CSS y JavaScript, se encuentran almacenados en esta aplicación y se hacen referencia desde las demás aplicaciones, de modo que todas las páginas web creadas tengan una apariencia consistente.

Esta página de inicio simplemente contiene una descripción de la plataforma y los sencillos pasos que tiene que realizar un usuario en esta plataforma para poder sacarle todo el partido a la misma. Además, tiene un enlace que redirige al usuario para registrarse o iniciar sesión en la plataforma y acceder a sus proyectos. Este enlace dirige al usuario a la aplicación de *Autenticación*.

5.3 AUTENTICACIÓN DE USUARIOS

La aplicación *Autenticación* contiene toda la lógica para registrar y autenticar a los usuarios en la plataforma, además de contener las páginas web (HTML) o Templates de Django con los formularios tanto para crear usuarios como para iniciar sesión. La ruta URL de esta aplicación es "/authenticate", y por lo tanto todas las direcciones web con esta ruta URL serán tratadas por esta aplicación.

Dentro de la ruta URL de esta aplicación, ésta tiene tres funcionalidades. La primera funcionalidad, a través de la ruta URL vacía, permite a los usuarios iniciar sesión en la plataforma utilizando. La segunda permite a los usuarios registrarse mediante la ruta URL "/register". Finalmente, la tercera permite a los usuarios cerrar sesión en la plataforma a través de la ruta URL "/logout".

Una vez que un usuario crea una cuenta o inicia sesión, se redirige a la aplicación *Crear y Abrir Proyectos*, donde el usuario podrá ver, crear y borrar proyectos de machine learning.

Además, la plataforma puede verificar si hay un usuario ya autenticado en la misma sesión del navegador, lo que evita que el usuario tenga que iniciar sesión nuevamente en la plataforma.

En esta aplicación se ha creado una clase llamada "Usuario" que hereda de la clase "AbstractBaseUser" de Django. Cada usuario de esta nueva clase de usuarios tendrá asociados una lista de proyectos. Además, podemos almacenar más información para cada usuario como la última vez que inició sesión, y otros que ofrecen servicios adicionales, como el de administrador. Pero lo realmente importante, relacionar la clase "Usuario" con la clase "Proyecto", implica que podemos acceder a todos los proyectos de machine learning relacionados con un usuario y, al mismo tiempo, podemos asegurarnos de que ese usuario no tenga acceso a los proyectos de otros usuarios. Esto garantiza la privacidad de los usuarios, ya que solo ellos pueden acceder a sus propios proyectos de machine learning.

Además de garantizar la privacidad a la hora de acceder a los proyectos de cada usuario, el sistema también cuenta con seguridad a la hora de guardar la contraseña del usuario. La contraseña que se guarda es un hash de la contraseña introducida, y así, en el caso de que la base de datos de usuarios sea robada, no se podrá averiguar la contraseña del usuario pues esta se guarda en su forma Hash. Esto se consigue al heredar de la clase "AbstractBaseUser" de Django.

Para registrar a estos usuarios y permitirles iniciar sesión, se han creado formularios en Django. Estos formularios se pasan a los Templates HTML de Django a través del contexto, lo que permite al usuario crear su cuenta de usuario o iniciar sesión en la página web rellenando estos formularios.

5.4 *CREAR Y ABRIR PROYECTOS*

La aplicación *Crear y Abrir Proyectos* contiene la lógica para crear Proyectos de machine learning, visualizarlos y borrarlos. La ruta URL para acceder a esta aplicación es "/projects".

Cada Proyecto de machine learning está asociado a un conjunto de datos específico y contiene todos los modelos entrenados con ese conjunto de datos en particular. Al crear un proyecto, solo se requiere una fuente de datos y un nombre.

Cada uno de los proyectos creados guardan muchos datos en la base de datos por defecto en Django (SQLite3). La clase Proyecto guarda información, a través de sus atributos, que permiten el funcionamiento de la plataforma. Entre la información que guarda cada proyecto está la fuente de datos original que se utilizó para crear el proyecto, el nombre del proyecto, la variable a predecir, el tipo de predicción, el usuario al que pertenece este proyecto, los datos preprocesados, qué etapas se han realizado para redirigir al usuario a donde lo dejó por última vez, y las reglas que ha establecido el usuario para tratar y preprocesar los datos para futuras predicciones.

Toda esta información que guarda un proyecto se va añadiendo a la base de datos conforme el usuario va a avanzando en el proceso de preprocesado-análisis de datos, y el entrenamiento de los modelos de aprendizaje.

Una vez el usuario ha creado algún proyecto, la plataforma permite al usuario visualizar todos sus proyectos, ordenados en orden descendiente de creación, mostrando cuándo se subió la fuente inicial de datos, y así permite entrar al usuario en cada uno de ellos.

Finalmente, la plataforma permite también borrar los proyectos, pues al estar estos asociados a un conjunto de datos estático, si el usuario quiere añadir nuevos datos, tendrá que crear un nuevo proyecto, y por lo tanto el anterior quedará obsoleto. Al borrar un Proyecto también se borran todos los datos asociados con él, como modelos de machine learning o la forma en la que se deberían tratar los datos ausentes.

5.5 ANALIZAR DATOS

Esta aplicación contiene toda la lógica previa a entrenar los modelos de machine learning y se accede a ella mediante la ruta URL “/initProject” y a continuación, en la dirección web, se le pasa el ID del proyecto, aunque esto ocurre de forma automática al abrir el proyecto.

Antes de entrar en cada una de las cinco etapas o funcionalidades del análisis y el preprocesado de los datos, la aplicación confirma que el usuario que tenga la sesión iniciada sea el dueño del proyecto que se está abriendo. En el caso en el que un usuario esté intentando acceder a un proyecto que no le corresponde, se redirigirá al usuario a la aplicación de *Crear y Abrir Proyectos* para que seleccione un proyecto que le pertenezca.

La aplicación *Analizar Datos* guía al usuario por cinco etapas necesarias en el preprocesado de datos y el análisis de estos, ofreciendo cuando pueda recomendaciones de manera proactiva para ayudar y guiar al usuario inexperto. Además, la plataforma no permite al usuario saltarse ninguna de estas etapas, pues va registrando qué pasos lleva realizados, y cuando se cambia de vistas, y también de aplicaciones, la plataforma comprueba que se hayan hecho todas las etapas anteriores, y, si no, redirige al usuario a su última etapa realizada.

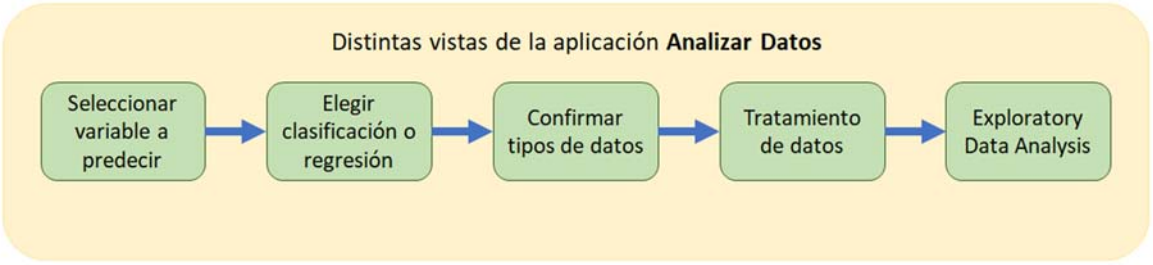


Figura 14. Estructura de las vistas y su orden de la aplicación Analizar Datos

5.5.1 SELECCIONAR VARIABLE A PREDECIR

Esta vista de la aplicación de *Analizar Datos*, tiene como objetivo que el usuario especifique qué variable quiere predecir del conjunto de datos que ha subido al proyecto. Se accede a esta vista mediante una ruta URL dentro de la ruta de Analizar Datos de "/analisisDescriptivo/seleccionarVariable".

Para elegir la variable a predecir, se ha creado dentro de esta aplicación un formulario *dropdown* de Django que se le pasa al Template con todas las columnas de la fuente de datos inicial del proyecto. Así, el usuario puede seleccionar qué variable quiere predecir en todo

el proyecto. La información de la variable a predecir se guarda como atributo del proyecto para poder consultar en las sucesivas etapas y aplicaciones cuál es la variable que se quiere predecir.

5.5.2 CONFIRMAR TIPO DE PREDICCIÓN

Tras elegir una variable como la variable a predecir en el proyecto, la aplicación *Analizar Datos* redirige al usuario a la vista *Confirmar Tipo de Predicción*. Esta vista, con ruta URL "/analisisDescriptivo/tipoPrediccion", pide al usuario que especifique el tipo de problema de machine learning que quiere solucionar en el proyecto. Las dos opciones posibles en el formulario *Dropdown* de Django son regresión y clasificación. Como es posible que muchos usuarios no sepan identificar si su problema es uno de regresión o de clasificación, la plataforma establece por defecto la mejor opción habiendo analizado la variable a predecir en los datos subidos por el usuario al proyecto. Así, el usuario simplemente tiene que confirmar el tipo de problema pinchando el botón de "confirmar" dejando la opción por defecto. Aun así, se permite que el usuario cambie el tipo de problema si los datos así lo permiten (pues no se puede hacer una regresión sobre una variable con cadenas de caracteres).

5.5.3 CONFIRMAR TIPOS DE DATOS

Una vez el usuario ha seleccionado o confirmado si el proyecto conlleva un problema de clasificación o de regresión, la plataforma redirige al usuario a la vista *Confirmar* Datos. Esta vista ofrece al usuario poder cambiar los tipos de datos que especifican cómo va a tratar la plataforma a cada una de las variables del proyecto. Esta vista, a la que se accede mediante la ruta URL dentro de la de Analizar Datos, de "/analisisDescriptivo/confirmarDatos", tiene como objetivo poder corregir algunos errores que pueda tener el usuario a la hora de crear la base de datos. Un posible caso de uso sería que una variable realmente categórica, cuyas categorías no tienen ningún orden, está codificada con números. Con esta funcionalidad el usuario podrá decirle a la plataforma que esa variable debe tratarla como una variable puramente categórica. Para simplificar las tareas a los usuarios inexpertos, la plataforma

selecciona por defecto los valores más comunes, inspeccionando los datos de cada variable, para que estos solo tengan que confirmar estos tipos de datos.

La forma en la que la plataforma propone a los usuarios seleccionar cómo debe tratarse cada columna es mediante unos formularios Django en formato *dropdown*. No obstante, aunque los usuarios puedan cambiar los tipos de datos, la plataforma antes de guardar esta información comprueba que se pueda hacer ese cambio en cada una de las variables, para así no propagar fallos a futuras etapas del proyecto.

5.5.4 TRATAMIENTO DE DATOS

Después de haber confirmado a la plataforma cómo debería tratar cada una de las variables que contenían los datos subidos por el usuario al crear el proyecto, la plataforma redirige al usuario a la vista de *Tratamiento de Datos*, a través de la ruta URL "/analisisDescriptivo/tratarNa". Esta vista analiza cada todo el conjunto de datos y propone al usuario maneras en las que preprocesar los datos.

Lo primero que comprueba la plataforma es, para todas las columnas del conjunto de datos excluyendo la variable a predecir, si debiera borrarse esa columna por no añadir ningún conocimiento extra al conjunto de datos. Con esta primera prueba la plataforma pretende sugerir al usuario eliminar variables que son identificativas de cada observación y que por tanto no van a añadir información adicional al modelo. Ejemplos de columnas que serían seleccionadas para ser borradas para crear modelos más simples sin ruido son: ID usuario, teléfono, email, o textos que contenga cada una de las observaciones.

Una vez se la plataforma ha seleccionado las columnas que son susceptibles de ser borradas, la plataforma comprueba los datos ausentes de cada una de las variables del conjunto de datos excluyendo la variable a predecir. Para cada una de estas variables que contengan datos ausentes, la plataforma calcula el porcentaje de datos que faltan y así sugiere varias acciones dependiendo de este valor.

I. Si faltan **menos del 20%** de datos, la plataforma recomienda eliminar esas observaciones

II. Si faltan entre el **20% – 40%** de los datos, la plataforma recomienda asignar estos valores al valor mediano si la variable es numérica y a la categoría más presente si es una variable categórica
III. Si faltan **más del 40%** de los datos, la plataforma recomienda eliminar esa columna

Estas son las recomendaciones que hace la plataforma al usuario para tratar los datos que faltan y las columnas con valores que aportan poco valor añadido. No obstante la plataforma, mediante formularios Django creados a medida en formato *dropdown*, permite al usuario tomar otras acciones.

- **Para columnas con poca representatividad**, la plataforma permite los valores de eliminar la columna o de mantenerla
- **Para columnas categóricas con valores ausentes**, la plataforma permite eliminar esos valores ausentes, eliminar la columna, o asignar a los valores ausentes la categoría más presente en el resto del conjunto de datos
- **Para las columnas numéricas con valores ausentes**, la plataforma permite eliminar estos valores ausentes, eliminar la columna, asignar a los valores ausentes el valor mediano del conjunto de datos, o asignarles el valor 0.

Gracias a esta vista, la plataforma propone al usuario de manera proactiva distintas acciones para tratar los datos, aunque el usuario siempre es el responsable de confirmar estos valores por defecto.

Lo que acepte el usuario en este paso se guarda en la configuración del proyecto y se aplicará en todas las futuras etapas del proyecto hasta que no se reescriba esta configuración. Esto incluye a los ficheros con los que el usuario quiera hacer predicciones, que se tratarán con estas mismas reglas.

5.5.5 VISUALIZACIÓN INTERACTIVA DE LOS DATOS

Una vez el usuario ha recorrido estos pasos se le ofrece la herramienta de visualización interactiva de los datos (EDA). Esta vista, a la que se accede con la ruta URL de "/analisisDescriptivo/visualizarDatos", permite al usuario entender mejor las distintas

relaciones entre sus variables y sus distribuciones de una forma interactiva. Para esto, esta vista, dentro de la aplicación de *Analizar Datos*, utiliza la librería Plotly para poder crear distintos gráficos para explicar mejor los datos.

Esta etapa de Exploratoy Data Analysis no está implementada en ninguna plataforma de manera nativa (solo parcialmente en algunos casos). Con esta herramienta, el usuario puede elegir dos variables en un formulario para comparar, y generar distintos gráficos en función de qué tipos de variables se estén comparando entre sí.

- Si las dos variables seleccionadas son una misma variable numérica, se dibuja un histograma de esa variable numérica
- Si las dos variables seleccionadas son una misma variable categórica, se dibuja un gráfico circular para visualizar las proporciones de cada una de las categorías.
- Si las dos variables seleccionadas son distintas y ambas son numéricas, se dibuja un gráfico de dispersión
- Si las dos variables seleccionadas son distintas y una de ellas es numérica y la otra es categórica, se dibuja un boxplot evidenciando la distribución de cada una de las categorías presentes en la variable categórica
- Si las dos variables seleccionadas son distintas y ambas son categóricas, se dibuja un gráfico de mosaico, donde poder ver la diferente proporción de una categoría en función de los valores de la otra.

5.6 *CREAR MODELOS*

Una vez el usuario ha pasado por todos los pasos de la aplicación de *Analizar Datos*, se le redirecciona a la aplicación *Crear Modelos*, a la que se accede mediante la ruta URL general de "/modelsProject". Esta aplicación contiene la lógica del entrenamiento de los modelos de predicción implementados.

Para ello, es necesario un apartado de que divida el conjunto de datos en datos de entrenamiento y datos de prueba, codifique las variables categóricas en variables independientes, y normalice las variables numéricas.

Posteriormente, es necesario que se elija qué modelos de machine learning se van a implementar, y elegir sus hiperparámetros y entrenarlos.

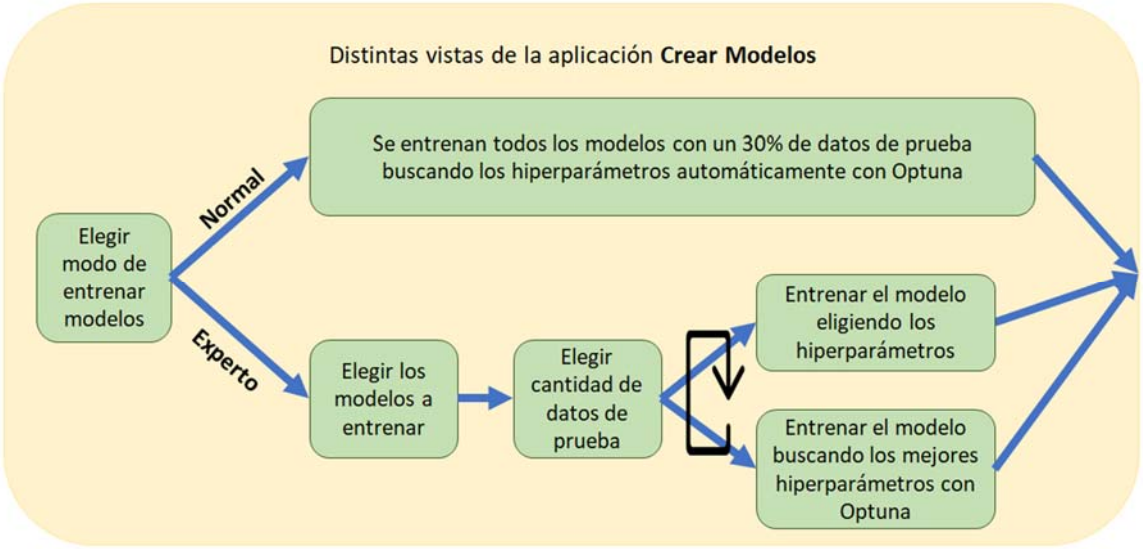


Figura 15 – Estructura de las vistas y su orden de la aplicación Crear Modelos

5.6.1 ELEGIR MODO DE CREAR MODELOS

Como esta plataforma está pensada para el usuario inexperto, la primera vista que se le ofrece al usuario dentro de esta aplicación, con la ruta URL vacía, es la de elegir el modo en el que quiere crear los modelos en esta aplicación.

Los dos modos para elegir son “Normal” y “Experto”. En el modo “Normal”, el usuario no tiene que hacer nada más en esta aplicación, pues la plataforma entrena todos y cada uno de los modelos de machine learning disponibles para el tipo de problema del proyecto (clasificación y regresión), buscando sus hiperparámetros de forma óptima como se

describirá más adelante. Así, el usuario no tiene por qué saber qué modelos de aprendizaje supervisado funcionarán mejor para su conjunto de datos, sino que la plataforma entrena todos los modelos y posteriormente los evaluará en la aplicación de visualización. Además, en el modo normal se divide el conjunto de datos en datos de entrenamiento y datos de prueba de manera automática, seleccionando el 30% de los datos para prueba y el 70% para entrenamiento.

Si el usuario selecciona el modo "Experto", la plataforma asume que tiene conocimientos de machine learning y por lo tanto puede seleccionar él el porcentaje de datos que se destinan a prueba y a entrenamiento a la vez que selecciona los modelos de machine learning a entrenar.

5.6.2 ELEGIR LOS MODELOS DE MACHINE LEARNING A ENTRENAR

Si el usuario ha seleccionado el modo "Experto", se le redirige a esta vista de esta aplicación a través de la ruta URL "/elegirModelos". En esta vista se permite al usuario elegir los modelos de machine learning que quiere entrenar mediante un formulario Django en el que se pueden seleccionar varios modelos. Esta información se guarda en el proyecto para que las futuras vistas o aplicaciones puedan saber los tipos de modelos que se tienen que entrenar o se han entrenado.

Si el usuario ha seleccionado el modo "Normal" en la vista anterior, automáticamente se le seleccionan todos los modelos de machine learning para el tipo de problema del proyecto (regresión o clasificación) y se guarda esta información en la clase de Proyecto.

Como se ha comentado anteriormente en el capítulo 2, se han implementado los siguientes modelos de machine learning para cada uno de los tipos de problemas que puede tener el proyecto:

- Para proyectos de regresión:
  - Regresión linear
  - Elastic Net
  - KNN

- SVR
- Random Forest
- Redes Neruronales Profundas (Deep Neural Networks)

- Para proyectos de clasificación:
  - Regresión logística
  - KNN
  - SVC
  - Random Forest
  - Redes Neruronales Profundas (Deep Neural Networks)

5.6.3 SELECCIONAR DATOS DE ENTRENAMIENTO Y DATOS DE PRUEBA

Si el usuario ha seleccionado el modo "Experto", una vez ha seleccionado los modelos de machine learning que quiere entrenar, se le redirige a esta vista a través de la ruta URL "/preprocesado". El objetivo de esta vista es que el usuario seleccione el porcentaje del conjunto total de datos del proyecto que se destinarán a entrenamiento y cuántos a valorar la eficacia del modelo. La plataforma recomienda un 30% de datos destinados a probar la eficacia del modelo, aunque el usuario puede cambiar este porcentaje.

Si el usuario ha seleccionado el modo "Normal" en la primera vista de esta aplicación, la plataforma asume un 30% de datos destinados a valorar el modelo, y un 70% destinados al entrenamiento del mismo. Al mismo tiempo, en ambos modos de creación de modelos, este porcentaje se guarda en la clase de Proyecto para que sea accesible en futuras etapas.

Además, una vez se selecciona este porcentaje de datos para entrenamiento y para prueba (tanto en el modo "Normal", como en el "Experto"), la plataforma codifica los datos para que los modelos de machine learning puedan entrenarse utilizándolos.

Así, dentro de esta vista, para ambos modos de creación de modelos, las variables categóricas se codifican en variables independientes, y las variables numéricas se normalizan en una función normal. Los parámetros de codificación tanto para las variables numéricas como para las categóricas, conseguidos con los datos de entrenamiento, se aplican a los datos de

prueba, y posteriormente se guardan en los datos del Proyecto para poder replicarlo en la fase de predicción con datos nuevos.

5.6.4 ENTRENAMIENTO DE LOS MODELOS

Una vez se ha dividido el conjunto inicial de datos en datos de entrenamiento y datos de prueba, y estos se han codificado para que puedan usarse para entrenar modelos de machine learning, se redirige al usuario a esta vista a través de la ruta URL "/entrenarModelos". Esta vista itera sobre todos los modelos seleccionados para entrenarlos. Toda la información de los modelos de machine learning se guardan en la clase creada en esta aplicación, *Crea Modelos*, llamada ModelosMachineLearning. Esta clase contendrá información acerca del modelo de machine learning, como por ejemplo el proyecto al que pertenece, el fichero del modelo entrenado, su métrica con el conjunto de prueba, y otras características propias de cada uno de los modelos implementados para garantizar que todos puedan funcionar juntos.

En la vista del entrenamiento de modelos, se muestra al usuario en el menú de la izquierda de la pantalla todos los modelos que ha seleccionado para entrenar, y se va iterando por cada uno de ellos para entrenarlo. La plataforma indica al usuario qué modelos ya se han entrenado con un *tick* al lado del nombre del modelo en este menú lateral en la parte izquierda de la pantalla.

Para cada modelo de machine learning se ofrece al usuario dos opciones. La primera es entrenar el modelo buscando automáticamente los mejores hiperparámetros, mientras que la segunda es que el propio usuario entrene el modelo insertando él mismo los hiperparámetros que él considere. Para buscar automáticamente los hiperparámetros, la plataforma utiliza la librería Optuna (descrita en el capítulo 2) que utiliza algoritmos de prueba y error para encontrar la configuración óptima que maximice el rendimiento del modelo.

Si el usuario no quiere buscar automáticamente los mejores hiperparámetros, se le redirige a la vista "/entrenarModelosConParams", donde puede seleccionar para cada modelo los hiperparámetros que desee para entrenar el modelo.

Así, un usuario puede elegir entrenar algunos modelos buscando automáticamente los hiperparámetros (opción por defecto), y otros modelos especificando él mismo los hiperparámetros a utilizar para entrenar el modelo.

En el caso de que un usuario que haya seleccionado el modo "Normal", todos los modelos se entrenarán utilizando la búsqueda óptima de hiperparámetros de Optuna.

Finalmente, cuando se ha terminado de entrenar cada modelo, antes de guardarlo en la clase ModelosMachineLearning, se calcula la efectividad del modelo con una métrica con el conjunto de prueba para poder comparar posteriormente los modelos, y se guarda esta métrica en la clase de ModelosMachineLearning.

La métrica elegida para problemas de regresión es la raíz del error cuadrático medio (RMSE por sus siglas en inglés). Esta métrica está elegida por su fácil interpretación al estar en las unidades de las que se están haciendo las predicciones. En problemas de clasificación la métrica elegida es la precisión ponderada por la cantidad de observaciones en cada clase (Balanced - Accuracy en inglés) por su fácil interpretación por usuarios inexpertos.

5.6.5 CREAR ENSEMBLES

En el caso de que el usuario haya entrenado al menos dos modelos de machine learning en un proyecto, se le permite crear Ensembles. Un Ensemble es una técnica que combina varios modelos de machine learning, ponderados en función de una métrica común para crear un modelo mucho más robusto.

Esta vista, a la que se accede mediante la ruta URL "/crearEnsemble", permite al usuario crear otro modelo de predicción, como un nuevo objeto de la clase ModelosMachineLearning, que combine los modelos que el usuario elija. Esta vista presenta al usuario un formulario de Django en el que el usuario tiene que seleccionar, de los modelos entrenados, más de uno, para poder juntarlos y que trabajen juntos. Los modelos en el formulario están ordenados en función de su eficacia (determinada por la métrica asociada a cada tipo de proyecto). Al crear el Ensemble, antes de guardarlo en la base de datos, se le calcula su métrica particular para valorarlo como otro modelo más.

5.7 VISUALIZAR MODELOS

Una vez el usuario ha entrenado algún modelo de machine learning, puede acceder a la aplicación de Visualizar Modelos a través de la ruta URL "/vizProject". Esta aplicación solo contiene una vista (a través de la ruta URL vacía) mediante la cual el usuario puede visualizar el comportamiento de los modelos y compararlos más fácilmente mediante gráficas. Tanto en regresión como en clasificación se ordena a los modelos en un menú a la izquierda de la pantalla, en función de su métrica, permitiendo ver fácilmente al usuario qué modelos funcionan mejor, y en qué orden.

5.7.1 CLASIFICACIÓN

Para los proyectos de clasificación, se proporcionan tres diferentes pantallas de gráficas. En primer lugar, están las gráficas para el usuario inexperto. Estas gráficas contienen una matriz de confusión, y una herramienta de visualización interactiva que permite ver al usuario de qué forma acierta o se equivoca su modelo. Como es posible que no todos los usuarios entiendan la matriz de confusión, se ha desarrollado un *tooltip* con una explicación de la matriz de confusión para que cada vez que un usuario pase por encima de la gráfica, aparezca la explicación de esta gráfica.

En segundo lugar, hay un apartado que contiene una curva ROC, junto con el AUC para cada categoría desarrollados para el usuario experto.

Finalmente, también para el usuario experto se ha creado una tercera pantalla que contiene la forma en la que se han elegido los hiperparámetros mediante Optuna, la importancia de cada uno de los hiperparámetros para este conjunto de datos, y finalmente los hiperparámetros elegidos. Esta pantalla no está disponible para los modelos que hayan sido creados sin Optuna (si el usuario ha introducido los hiperparámetros manualmente), o si el modelo es un Ensemble, y por lo tanto tiene a varios modelos detrás.

5.7.2 REGRESIÓN

Para los proyectos de regresión, se proporcionan 2 pantallas de gráficas.

En primer lugar, están las gráficas para el usuario inexperto. En esta pantalla está una gráfica que representa el valor predicho frente al valor real, junto con la recta y=x, que señalaría una predicción perfecta. Esta gráfica también contiene un *tooltip* que explica esta gráfica a los usuarios inexpertos. La segunda gráfica de esta pantalla es la curva de un histograma de los errores cometidos por este modelo, junto al dibujo de una normal. Esta gráfica, también acompañada de un *tooltip* que explica su funcionamiento, intenta mostrar al usuario el error que comete el modelo, enseñándole que cuanto más cercano sea a una normal (o incluso más puntiagudo cerca del cero) mejor será el modelo. Finalmente, en esta primera pantalla para el usuario inexperto se dibuja esta misma gráfica, solo que, superponiendo todos los modelos creados, para que así el usuario, en una sola gráfica, pueda comparar todos los modelos que ha entrenado.

La segunda pantalla, hecha para el usuario experto, como en el apartado de clasificación, contiene la forma en la que Optuna ha elegido los hiperparámetros para este modelo, junto con su importancia y los hiperparámetros finales elegidos. Al igual que en clasificación, esta pantalla solo está disponible para los modelos que hayan sido entrenados con la herramienta de Optuna y por ello quedan excluidos los Ensembles y los modelos que hayan sido entrenados con los hiperparámetros especificados por el usuario.

5.8 PREDICCIÓN

Esta última aplicación, disponible desde que se haya entrenado un modelo de machine learning y que se accede a través de la ruta URL "/prediction", permite al usuario hacer predicciones sobre nuevos conjuntos de datos. Esta aplicación, que solo tiene una vista, cuya ruta URL es la nula, pide al usuario que suba a la plataforma un nuevo conjunto de datos y seleccione el modelo de machine learning con el que quiere hacer la predicción (dentro de estos modelos podemos encontrar los Ensembles que hemos creado). Una vez el usuario ha seleccionado el modelo de machine learning, y subido el archivo, el usuario pulsa el botón de hacer predicción, y la plataforma le devuelve el conjunto de datos que subió el usuario, junto con una nueva columna que contiene la predicción de ese modelo sobre ese conjunto de datos.

Para realizarlo, la vista en primer lugar transforma los datos subidos como los datos iniciales. Esto incluye el borrado de algunas columnas, borrar observaciones con datos que faltan, o rellenar datos que faltan para el valor mediano, el valor 0, o la categoría más presente (dependiendo de la naturaleza del problema). Además, esto incluye también la normalización de las variables numéricas y el codificado de las variables categóricas conforme se hizo con los datos de entrenamiento en la aplicación de Crear Modelos.

Además, la plataforma devuelve exactamente los mismos datos que se subieron por el usuario. Así que para las observaciones que han sido borradas durante el preprocesado, la plataforma, en la columna que incluye las predicciones, inserta la frase “Faltan datos”, para indicar al usuario que sobre esa observación no se ha realizado ninguna predicción porque existen campos sin datos y que en la aplicación de AnalizarDatos se le informó que debían ser borrados.

Capítulo 6. CASO DE USO

Una vez finalizada la plataforma a continuación se evidencia un posible caso de uso real de la misma con imágenes sobre su funcionamiento tanto ara un problema de clasificación como de regresión

6.1 CLASIFICACIÓN

Imaginemos una empresa que utiliza un software de CRM (Customer Relationship Management) para mantener información de sus clientes, y clasificar la posibilidad de que éstos cesen o no la relación comercial que tienen con la empresa (Churn). Esta empresa podría utilizar la plataforma para poder predecir qué clientes podrían prescindir de sus servicios y también cuál s el uso de los mismos por parte de los que confían en la empresa con objeto de incrementar su grado de fidelidad. Se ha descargado un conjunto de datos de una plataforma de CRM de la página web http://kaggle.com para ofrecer un posible caso de uso de esta plataforma.

El primer paso que realizaría el usuario es visitar la página web de la plataforma

Figura 16 – Vista de la página de inicio (Landing Page)

Posteriormente, el usuario procede a registrarse en la plataforma, indicando un nombre de usuario (usuario_prueba en el ejemplo), su email, y una contraseña, como se indica en la siguiente figura.

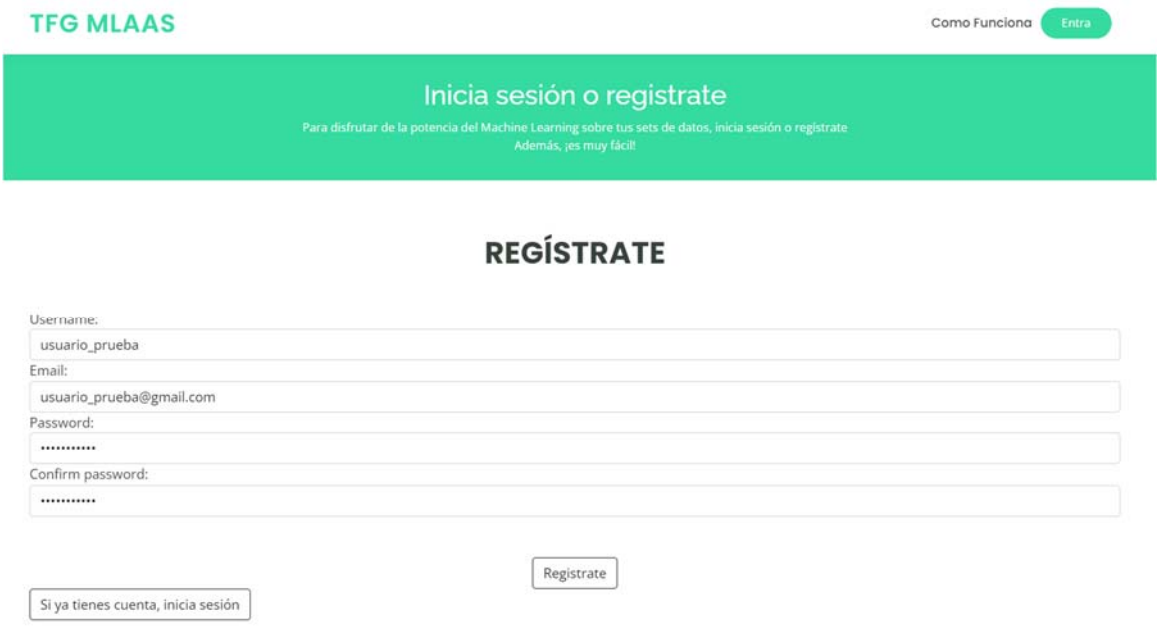


Figura 17. Vista de la página de registro (dentro de la aplicación Autenticación)

Una vez el usuario se ha registrado en la plataforma accede a la vista de los proyectos, donde la plataforma le indica que tiene que crear un nuevo proyecto.

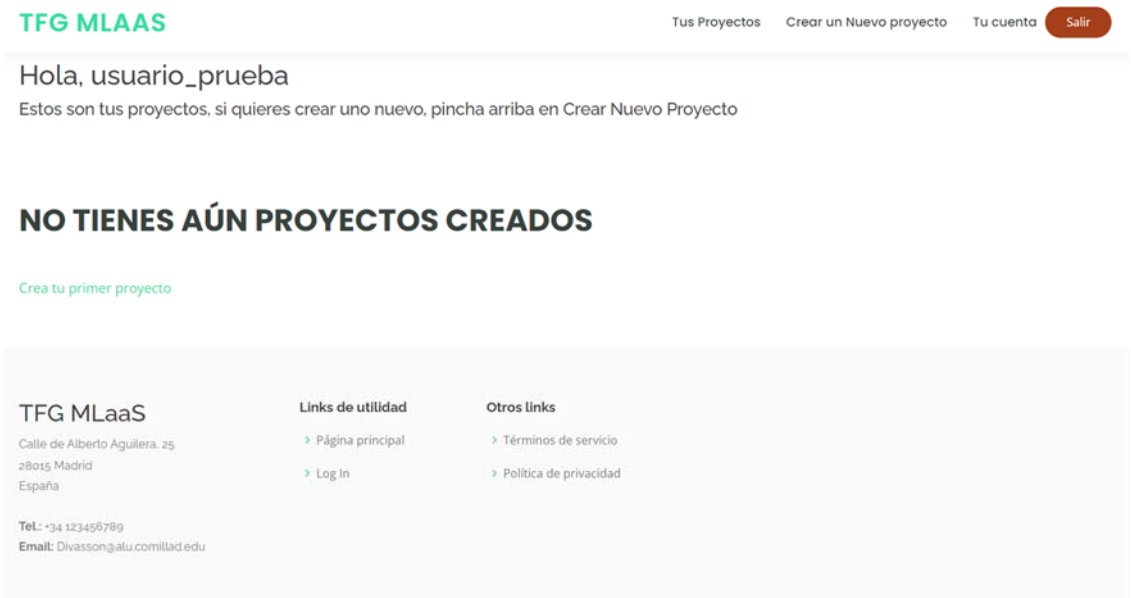


Figura 18. Vista de los proyectos del usuario (dentro de la aplicación Crear y Abrir Proyectos)

El usuario tras seleccionar *Crear un nuevo proyecto*, se encuentra con un formulario para crear un proyecto donde tiene que indicar el nombre del proyecto que está creando, así como subir el conjunto de datos a analizar en el proyecto.

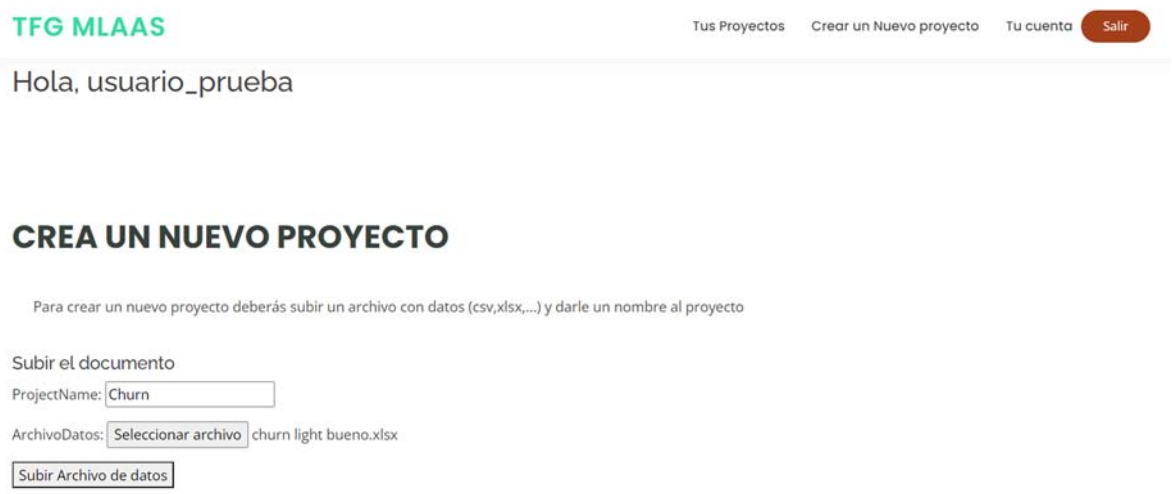


Figura 39. Vista del formulario de crear un proyecto (dentro de la aplicación Crear y Abrir Proyectos)

Una vez el usuario crea un nuevo proyecto, se le redirige a la vista anterior para visualizar los proyectos del usuario, mostrando este nuevo proyecto creado.

Los proyectos son mostrados en una tabla con dos columnas, ordenando los proyectos en función de la fecha de creación, teniendo como más reciente el primero de todos. Además, cada proyecto muestra el nombre asignado al proyecto, junto con la fecha de subida de los datos que se van a utilizar para entrenar los modelos de aprendizaje supervisado. Esta fecha es importante por si existen nuevas versiones de los datos más actualizadas, para que el usuario pueda crear nuevos proyectos con las fuentes de datos actualizadas.

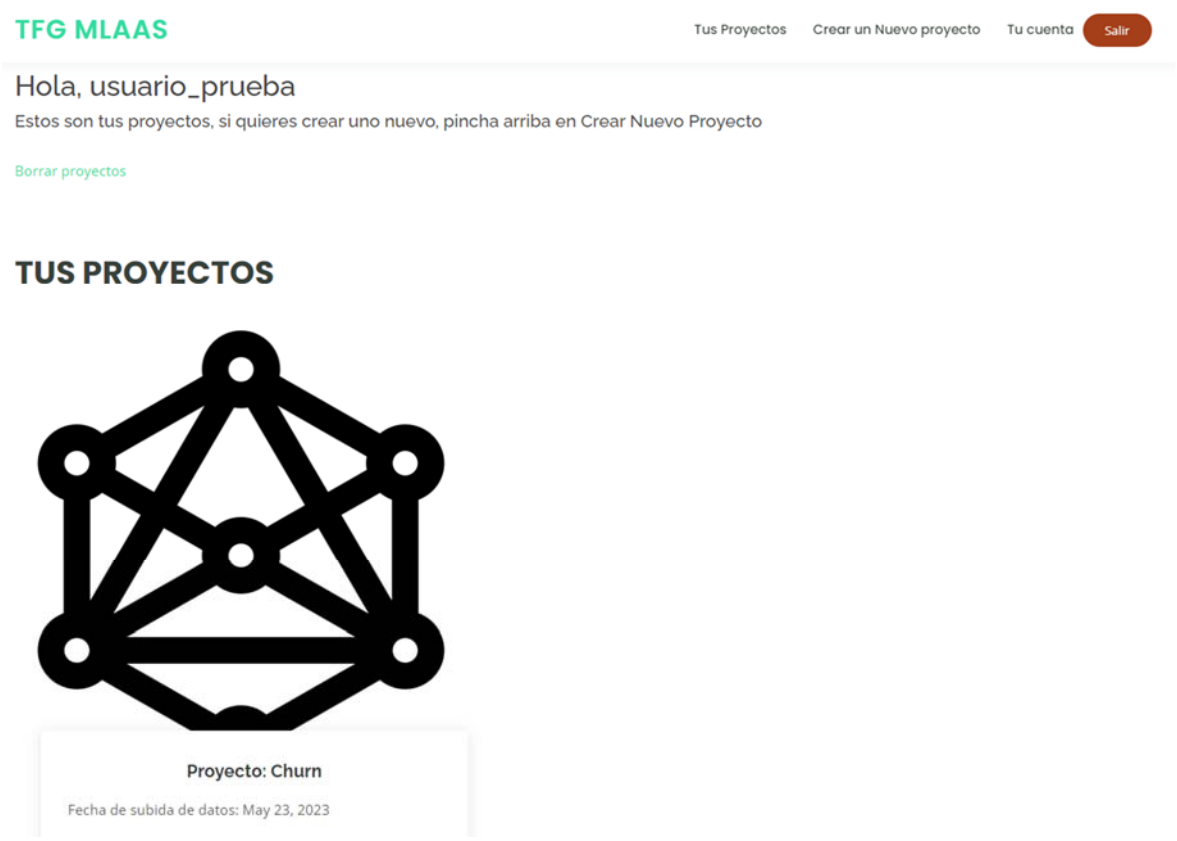


Figura 20. Vista de los proyectos del usuario (dentro de la aplicación Crear y Abrir Proyectos)

Una vez el usuario entra dentro del proyecto, se le redirige a la primera aplicación o etapa de todo el proceso que no haya realizado, pues la plataforma va guardando constantemente los pasos realizados por el usuario.

Como en este caso el usuario aún no ha realizado ningún paso porque es la primera vez que abre el proyecto, se le redirige a la primera aplicación, *Analizar Datos*, y dentro de ésta, a la primera vista, que es la de seleccionar la variable a predecir. Dentro de esta vista, el usuario se encuentra un formulario donde ha de indicar qué variable va a seleccionar como "target" u objetivo para predecir o clasificar en su caso). En este caso, el usuario selecciona la variable "Churn", y le selecciona la opción *enviar*.

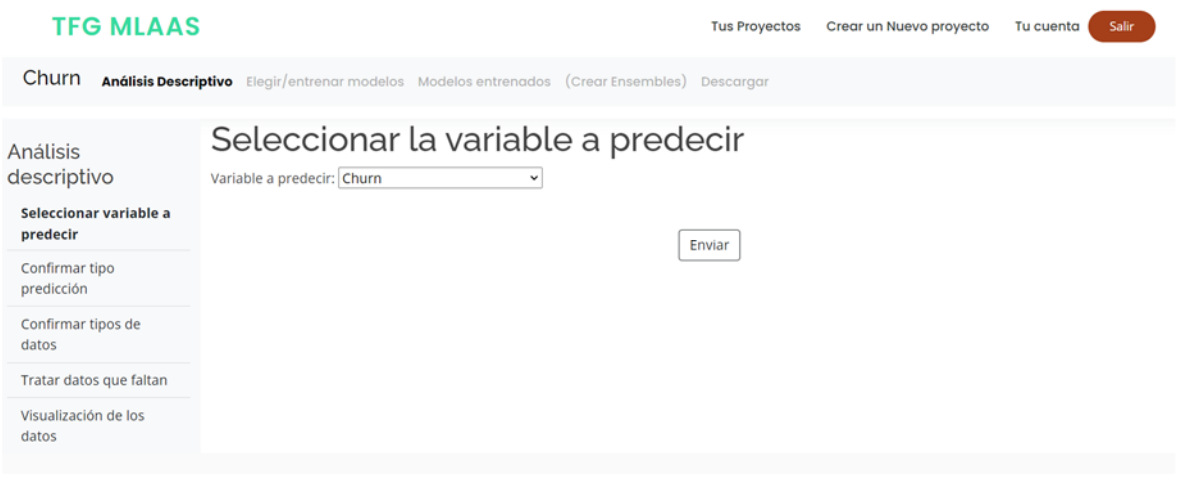


Figura 21. Vista de la selección de la variable a predecir del proyecto (dentro de la aplicación Analizar Datos)

Tras identificar la variable "Churn" como la variable objetivo, la plataforma redirige al usuario a la segunda vista de la aplicación *Analizar Datos*, que es la de seleccionar el tipo de algoritmo a usar. En esta vista el usuario debería simplemente confirmar el tipo de algoritmo, pues la plataforma analiza la variable target y establece por defecto la forma que más se adecúe a los datos analizados. En este caso, aunque la variable "Churn" solo contiene valores numéricos (ceros y unos), la plataforma recomienda seleccionar *Clasificación*.

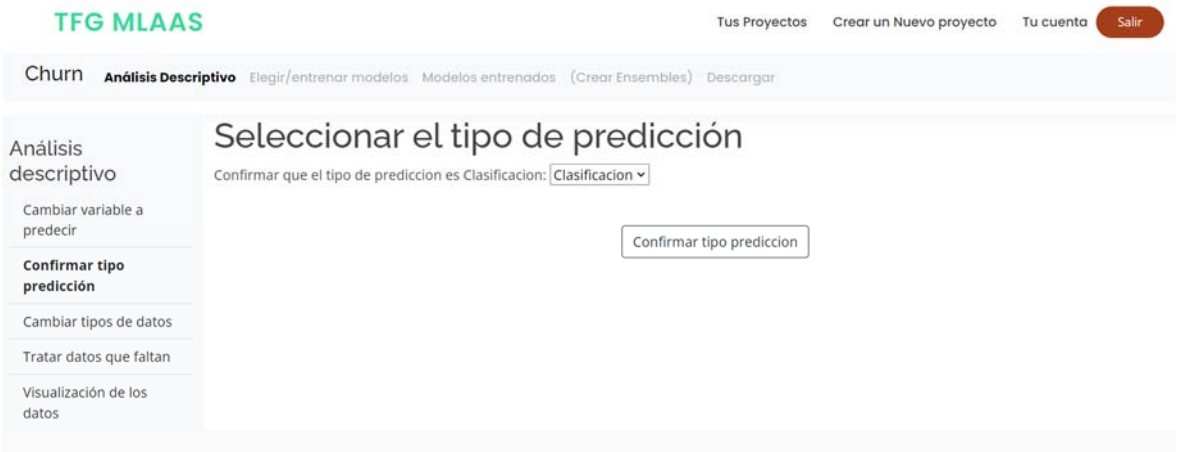


Figura 22. Vista de la selección del tipo de predicción (dentro de la aplicación Analizar Datos)

Tras confirmar el tipo de predicción, al usuario se le redirige a la vista de confirmar tipos de datos, dentro de la aplicación Analizar Datos. En esta vista se enseña el usuario una vista de

algunos datos subidos junto con un formulario sobre si quiere cambiar la forma en la que la plataforma trata a las columnas. En este caso el usuario no hace ningún cambio y clica el botón *Guardar y continuar* situado antes del formulario.

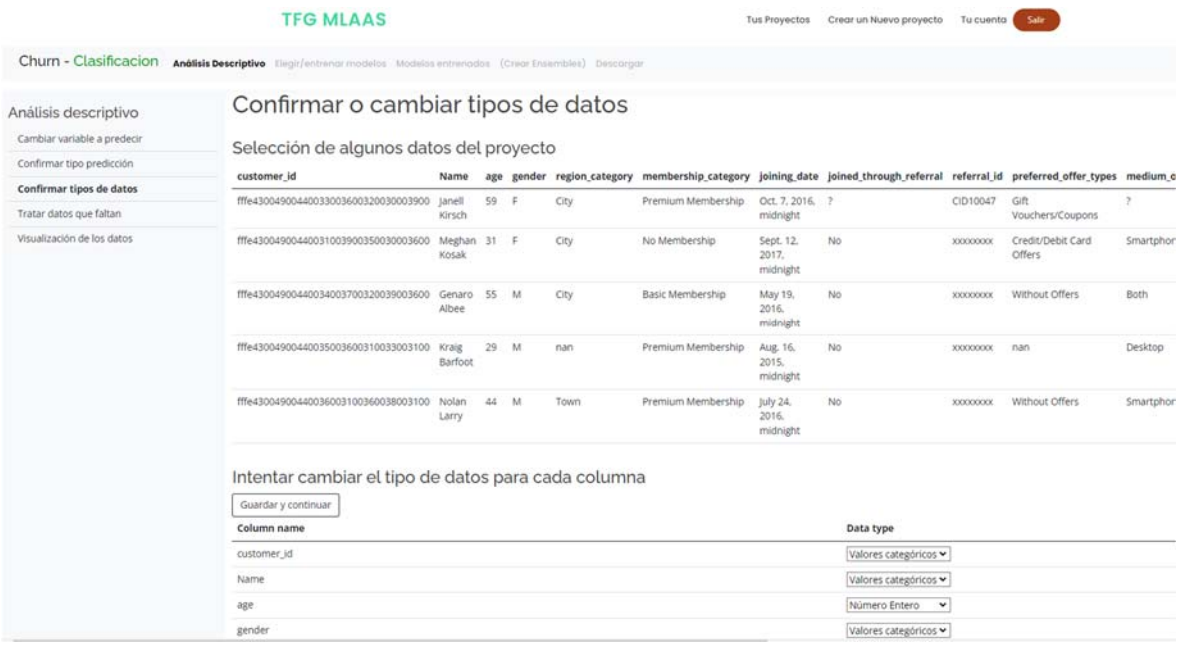


Figura 23. Vista de cambio de tipo de datos (dentro de la aplicación Analizar Datos)

Tras esta vista, se redirige al usuario a la vista de tratamiento de datos. En esta vista la plataforma enseña al usuario las columnas con algunos problemas, ofreciendo la lógica previamente mencionada para tratar estos datos. No obstante, el usuario puede decidir cambiar algunos patrones según considere. Tras hacer algunos cambios, como asignar los puntos de los usuarios en la plataforma de CRM que faltan a 0, el usuario guarda esta configuración, que se aplicará a todos los datos a partir de ahora.

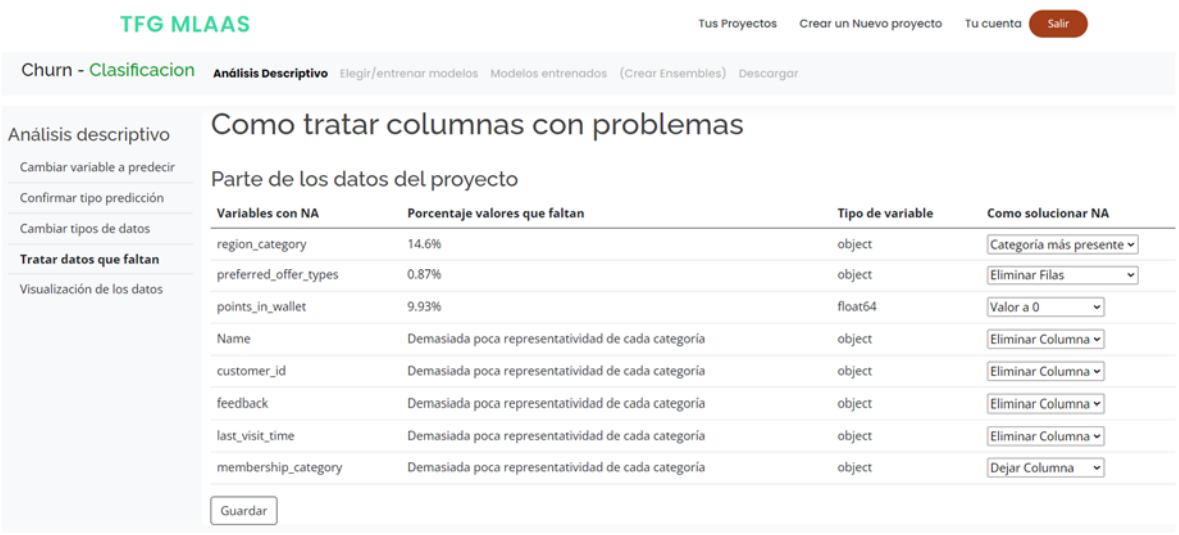


Figura 24. Vista de tratamiento y preprocesado de datos (dentro de la aplicación Analizar Datos)

Una vez el usuario ha elegido cómo tratar los datos, la plataforma le redirige a la vista que contiene visualizaciones interactivas para que éste pueda comprobar la relación entre sus datos. En este caso, el usuario está interesado en comprobar cómo se relacionan el medio por el que operan sus clientes, y si empezaron a ser clientes mediante un código de referencia.

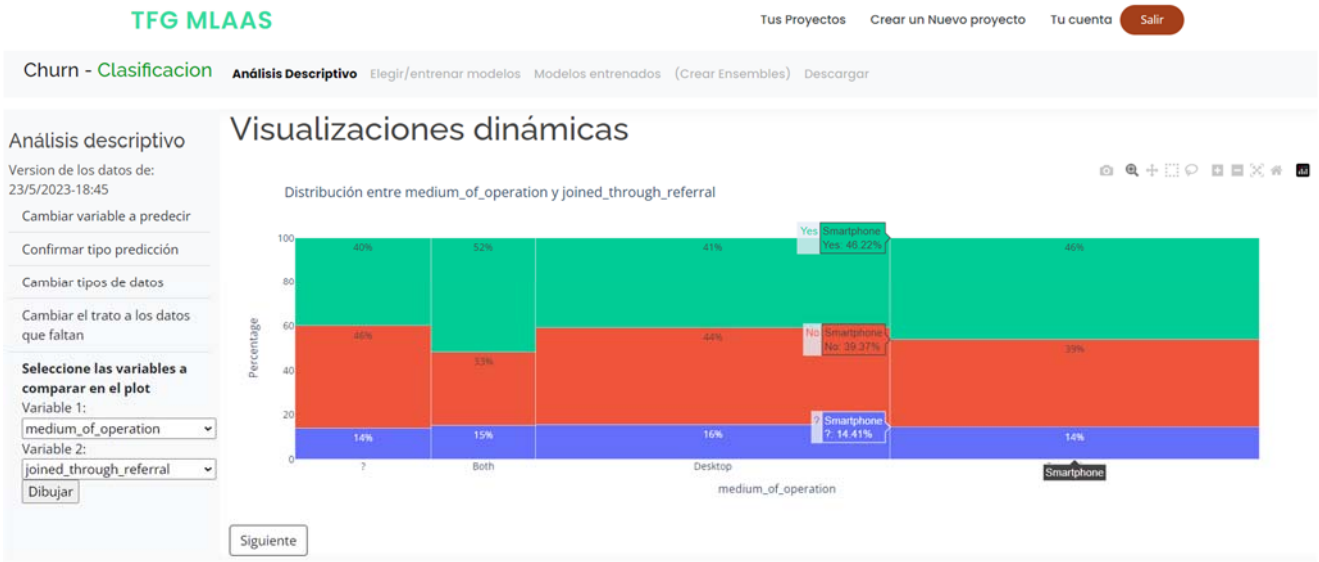


Figura 25. Vista de la interfaz de análisis de datos (dentro de la aplicación Analizar Datos)

Después de que el usuario analice los datos, sabiendo que siempre puede volver a esta interfaz para visualizarlos de forma interactiva, la plataforma le redirige a la aplicación cuyo

objetivo es entrenar modelos de machine learning (*Crear Modelos*). La plataforma inicialmente ofrece al usuario una vista con un formulario que pregunta si quiere seguir el modo "Normal", o por el contrario el modo "Experto". En este caso, el usuario ha elegido el modo "Normal", y por lo tanto la plataforma empieza a entrenar todos los modelos posibles habilitados en clasificación. Además, la plataforma le indica al usuario que se están entrenando los modelos mediante una rueda giratoria.

Figura 26. Vista del entrenamiento de los modelos según el modo "Normal" (dentro de la aplicación Crear Modelos)

Una vez terminan de entrenarse todos los modelos con los mejores hiperparámetros, se redirige al usuario a la aplicación de Visualizar Modelos, que permite al usuario comparar estos modelos ya entrenados. No obstante, el usuario puede elegir crear algunos Ensembles pinchando en el botón del menú superior. En esta vista, dentro de la aplicación de Crear Modelos, se ofrece al usuario el siguiente formulario para que éste seleccione todos los modelos que quiere analizar.

Figura 27. Vista de la forma en la que crear Ensembles (dentro de la aplicación Crear Modelos)

Una vez ya se han entrenado todos los modelos, y se han elegido los modelos que se quieren analizar, se redirige al usuario a la vista para visualizar los modelos y compararlos. En la parte izquierda, se pueden observar los modelos entrenados ordenados según su precisión ponderada en función de la cantidad de observaciones en cada clase (balanced - accuracy en inglés) con el conjunto de prueba. Aunque en este proyecto se ha elegido esta métrica, existen otras ampliamente difundidas en el mundo del machine learning.

En esta primera pantalla, el usuario puede ver dos gráficas. La gráfica de la izquierda es una matriz de confusión, que enseña al usuario cómo acierta y cómo se equivoca el modelo. A esta gráfica le acompaña una descripción de qué es lo que representa una matriz de confusión. A su izquierda, hay una gráfica que demuestra, en función de las variables de los datos subidos, cómo acierta y cómo se equivoca el modelo. Esta gráfica es una gráfica interactiva porque el usuario elige qué variables quiere comparar en el gráfico. En este caso se han comparado las variables “past_complaint” y “region_category”.

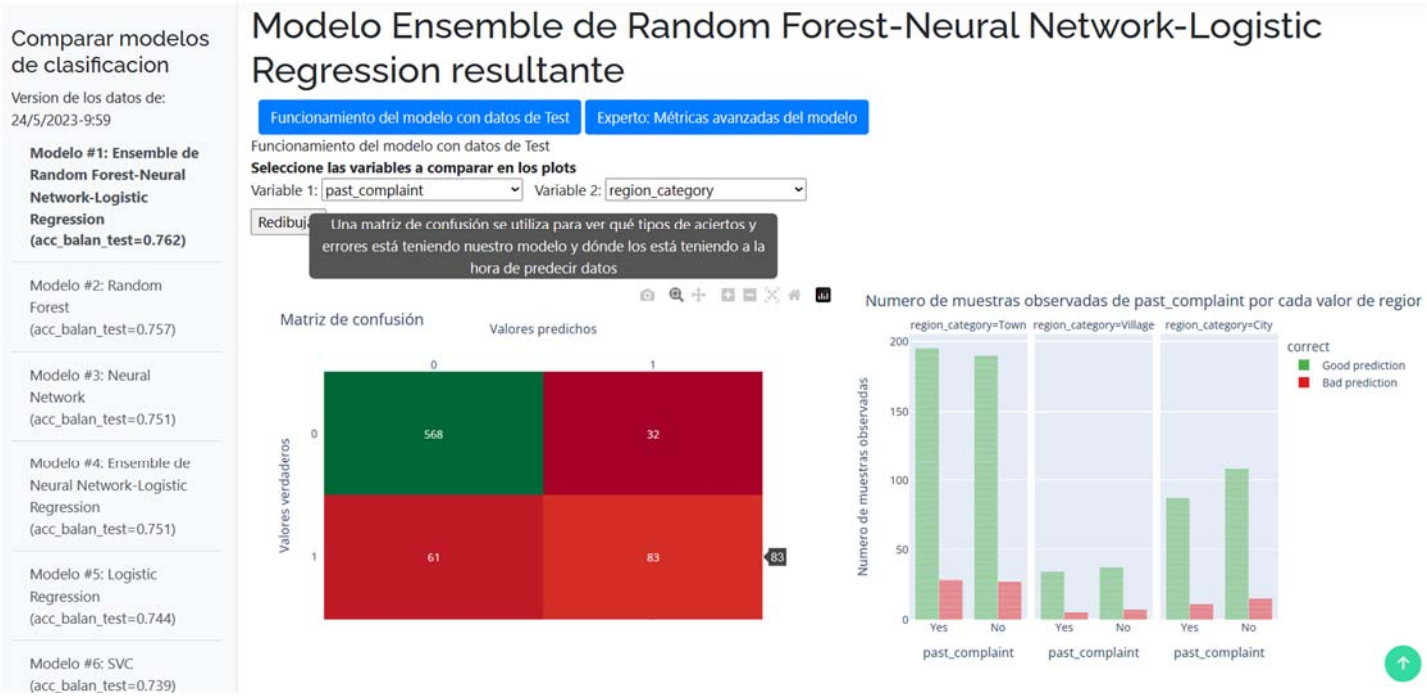


Figura 28. Vista primera con matriz de confusión y gráfica interactiva para valorar modelos (dentro de la aplicación Visualizar Modelos)

Si el usuario es un usuario experto, podrá clicar en las siguientes dos pantallas. La primera contiene una curva ROC junto con el área debajo de la curva, o AUC por sus siglas en inglés, para cada una de las clases de la variable a predecir.

Figura 29. Vista segunda con curva ROC para valorar modelos (dentro de la aplicación Visualizar Modelos)

La última pantalla contiene las gráficas con información sobre cómo se han elegido los hiperparámetros con la librería Optuna. Contiene gráficas que explican el proceso, la importancia de cada uno de los hiperparámetros, y finalmente, los hiperparámetros utilizados.

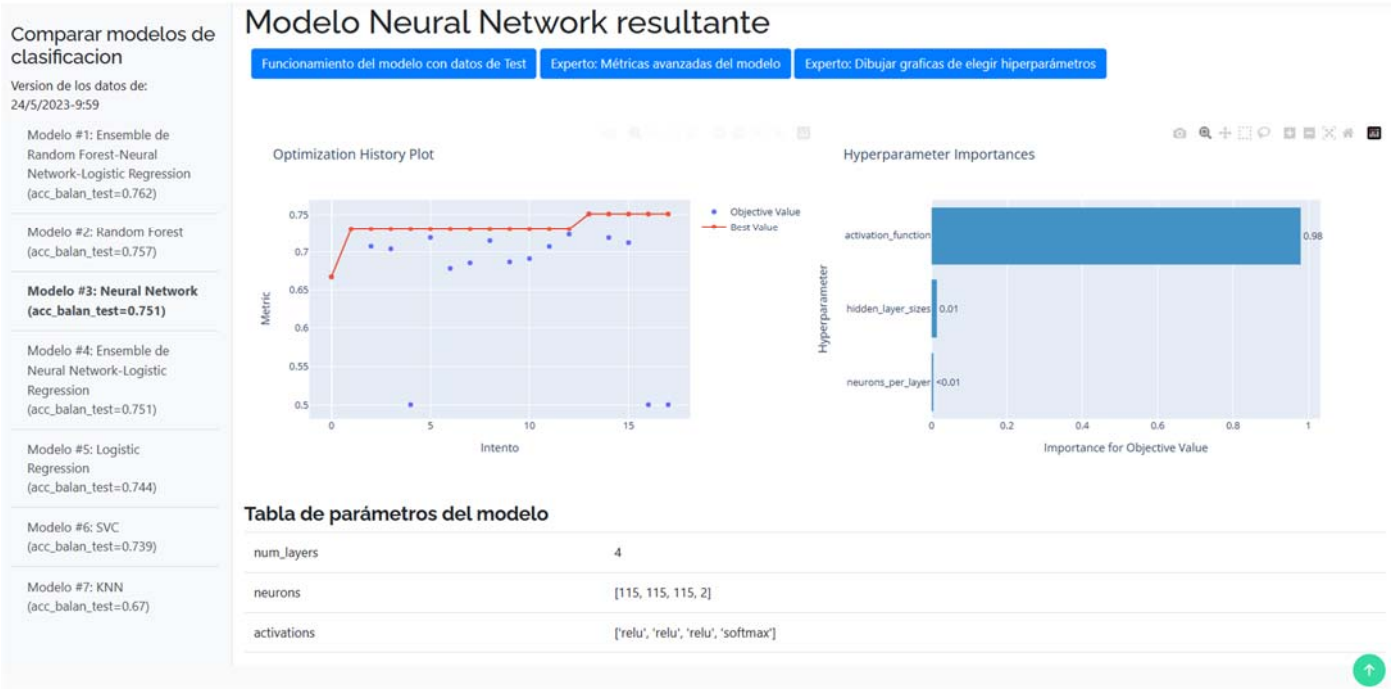


Figura 30 – Vista tercera con cómo se han elegido los hiperámetros (dentro de la aplicación Visualizar Modelos)

Por último, el usuario puede subir un nuevo conjunto de datos para utilizar estos modelos entrenados y predecir sobre este nuevo conjunto de datos. En esta vista, la plataforma pide al usuario subir el archivo que quiere usar, junto con el modelo que quiere utilizar para predecirlo. En este caso el usuario va a subir unos datos de los que tiene etiqueta para comprobar la eficacia de esta plataforma.

Una vez el usuario pincha en subir, la plataforma trata los datos subidos, hace la predicción, y devuelve el fichero subido, solo que con una columna extra con la predicción.

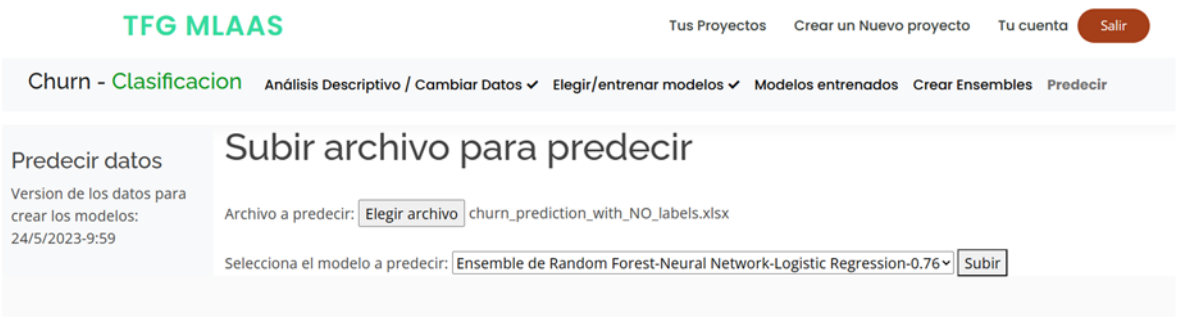


Figura 31. Forma en la que el usuario puede predecir un nuevo conjunto de datos (dentro de la aplicación Predicción)

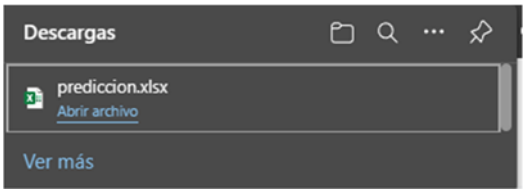
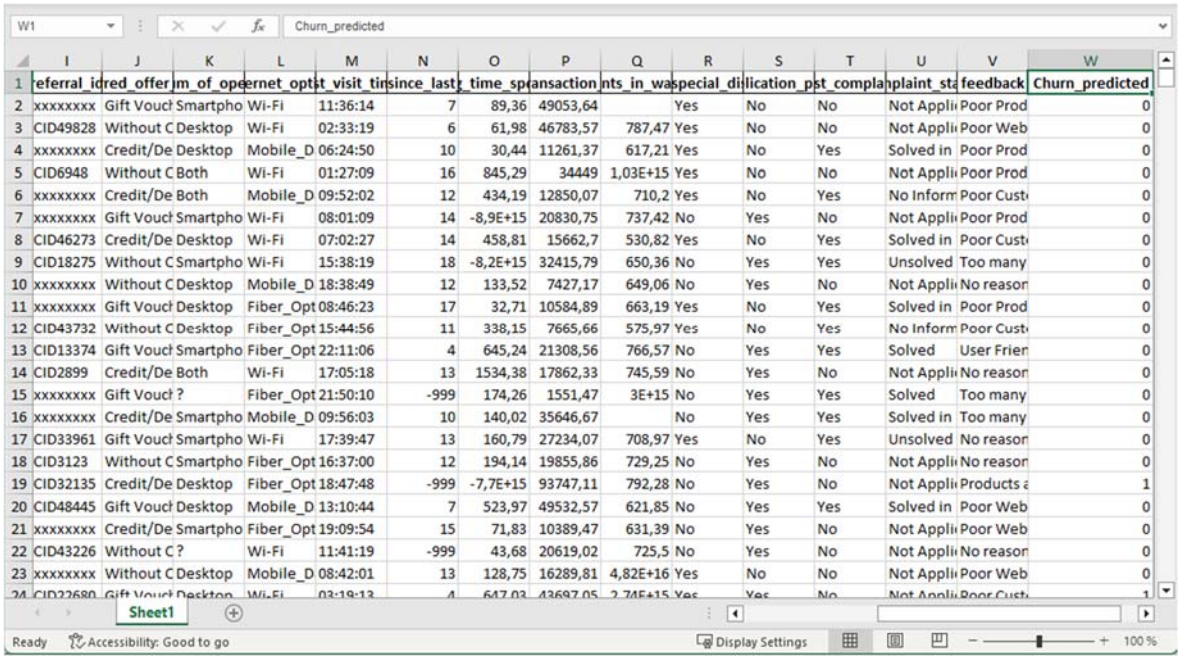


Figura 32. La plataforma automáticamente descarga el fichero subido con la predicción (dentro de la aplicación Predicción)



	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	referral_id	red_offer	m_of_ope	ernet_opt	st_visit_tir	since_last	_time_spe	ansaction	nts_in_wa	special_di	lication_pa	st_compla	plaint_sta	feedback	Churn_predicted
2	xxxxxxxx	Gift Vouc	Smartpho	Wi-Fi	11:36:14	7	89,36	49053,64		Yes	No	No	Not Appli	Poor Prod	0
3	CID49828	Without C	Desktop	Wi-Fi	02:33:19	6	61,98	46783,57	787,47	Yes	No	No	Not Appli	Poor Web	0
4	xxxxxxxx	Credit/De	Desktop	Mobile_D	06:24:50	10	30,44	11261,37	617,21	Yes	No	Yes	Solved in	Poor Prod	0
5	CID6948	Without C	Both	Wi-Fi	01:27:09	16	845,29	34449	1,03E+15	Yes	No	No	Not Appli	Poor Prod	0
6	xxxxxxxx	Credit/De	Both	Mobile_D	09:52:02	12	434,19	12850,07	710,2	Yes	No	Yes	No Inform	Poor Cust	0
7	xxxxxxxx	Gift Vouc	Smartpho	Wi-Fi	08:01:09	14	-8,9E+15	20830,75	737,42	No	Yes	No	Not Appli	Poor Prod	0
8	CID46273	Credit/De	Desktop	Wi-Fi	07:02:27	14	458,81	15662,7	530,82	Yes	No	Yes	Solved in	Poor Cust	0
9	CID18275	Without C	Smartpho	Wi-Fi	15:38:19	18	-8,2E+15	32415,79	650,36	No	Yes	Yes	Unsolved	Too many	0
10	xxxxxxxx	Without C	Desktop	Mobile_D	18:38:49	12	133,52	7427,17	649,06	No	Yes	No	Not Appli	No reason	0
11	xxxxxxxx	Gift Vouc	Desktop	Fiber_Opt	08:46:23	17	32,71	10584,89	663,19	Yes	No	Yes	Solved in	Poor Prod	0
12	CID43732	Without C	Desktop	Fiber_Opt	15:44:56	11	338,15	7665,66	575,97	Yes	No	Yes	No Inform	Poor Cust	0
13	CID13374	Gift Vouc	Smartpho	Fiber_Opt	22:11:06	4	645,24	21308,56	766,57	No	Yes	Yes	Solved	User Frien	0
14	CID2899	Credit/De	Both	Wi-Fi	17:05:18	13	1534,38	17862,33	745,59	No	Yes	No	Not Appli	No reason	0
15	xxxxxxxx	Gift Vouc	?	Fiber_Opt	21:50:10	-999	174,26	1551,47	3E+15	No	Yes	Yes	Solved	Too many	0
16	xxxxxxxx	Credit/De	Smartpho	Mobile_D	09:56:03	10	140,02	35646,67		No	Yes	Yes	Solved in	Too many	0
17	CID33961	Gift Vouc	Smartpho	Wi-Fi	17:39:47	13	160,79	27234,07	708,97	Yes	No	Yes	Unsolved	No reason	0
18	CID3123	Without C	Smartpho	Fiber_Opt	16:37:00	12	194,14	19855,86	729,25	No	Yes	No	Not Appli	No reason	0
19	CID32135	Credit/De	Desktop	Fiber_Opt	18:47:48	-999	-7,7E+15	93747,11	792,28	No	Yes	No	Not Appli	Products a	1
20	CID48445	Gift Vouc	Desktop	Mobile_D	13:10:44	7	523,97	49532,57	621,85	No	Yes	Yes	Solved in	Poor Web	0
21	xxxxxxxx	Credit/De	Smartpho	Fiber_Opt	19:09:54	15	71,83	10389,47	631,39	No	Yes	No	Not Appli	Poor Web	0
22	CID43226	Without C	?	Wi-Fi	11:41:19	-999	43,68	20619,02	725,5	No	Yes	No	Not Appli	No reason	0
23	xxxxxxxx	Without C	Desktop	Mobile_D	08:42:01	13	128,75	16289,81	4,82E+16	Yes	No	No	Not Appli	Poor Web	0
24	CID22680	Gift Vouc	Desktop	Wi-Fi	03:19:13	4	647,03	43697,05	2,74E+15	Yes	Yes	No	Not Appli	Poor Cust	1

Figura 33. Captura con el Excel descargado con la columna predicha

Finalmente, el usuario comprueba que el modelo Ensemble predice con una precisión ponderada (balanced - accuracy) de un 86% sobre estos datos de los que tenía etiqueta.

6.2 REGRESIÓN

Imaginemos ahora un caso de uso de regresión en el que un usuario quiere predecir los precios de coches de segunda mano. Otra vez, como en el caso anterior, hemos descargado un dataset de la página web https://kaggle.com con diversos datos sobre coches de segunda mano y sus precios.

Inicialmente el usuario crea el proyecto para predecir el precio de los vehículos de segunda mano introduciendo en el formulario de la página web un nombre del proyecto, en este caso "Precio Coches", y el fichero con el conjunto de datos.

Figura 34. Vista de creación del proyecto de regresión para predecir el precio de coches de segunda mano

Una vez creado el proyecto, nos aparece en la vista de todos los proyectos asociados al usuario que tiene la sesión iniciada, junto con el resto de los proyectos (como el de Churn mencionado en el capítulo 6.1).

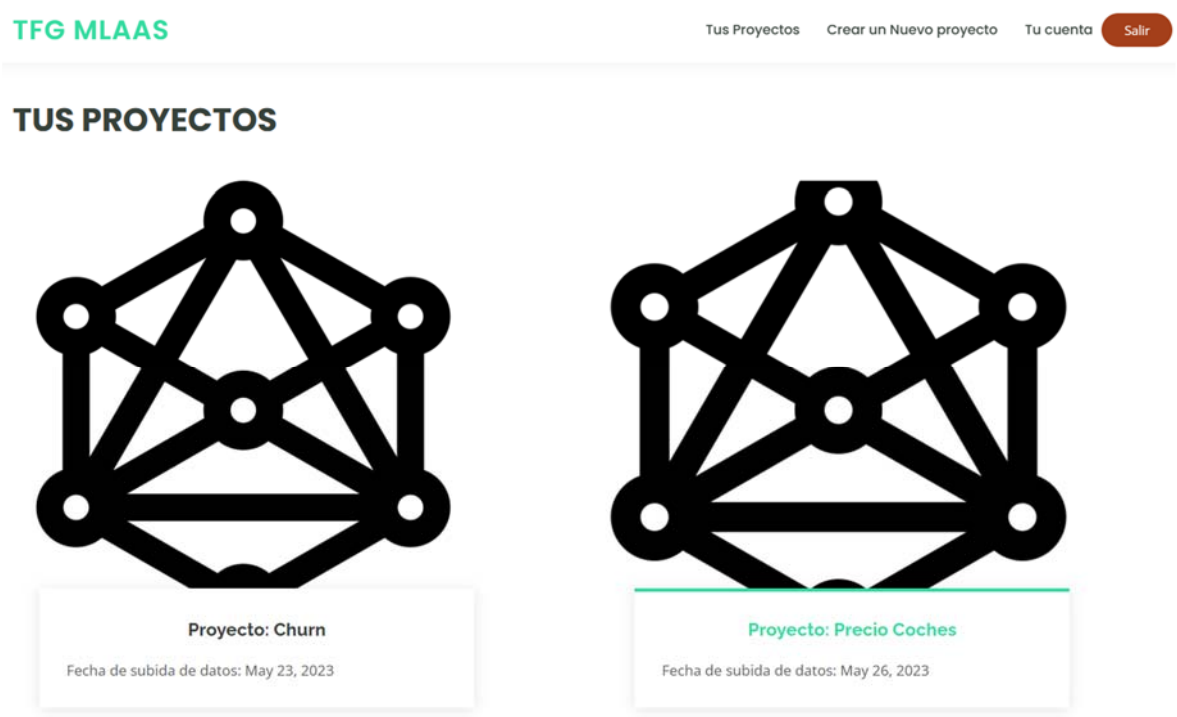


Figura 35. Vista de los proyectos creados por el usuario dado de alta en la aplicación

Una vez el usuario selecciona el proyecto "Precio Coches", se le redirige a la primera parte de las fases del entrenamiento de modelos de machine learning, el preprocesado y el análisis descriptivo. En la primera vista, la plataforma pide al usuario que seleccione qué variable quiere predecir en este proyecto, a lo que el usuario selecciona la variable "Price" del formulario con formato *dropdown* de esta primera vista.

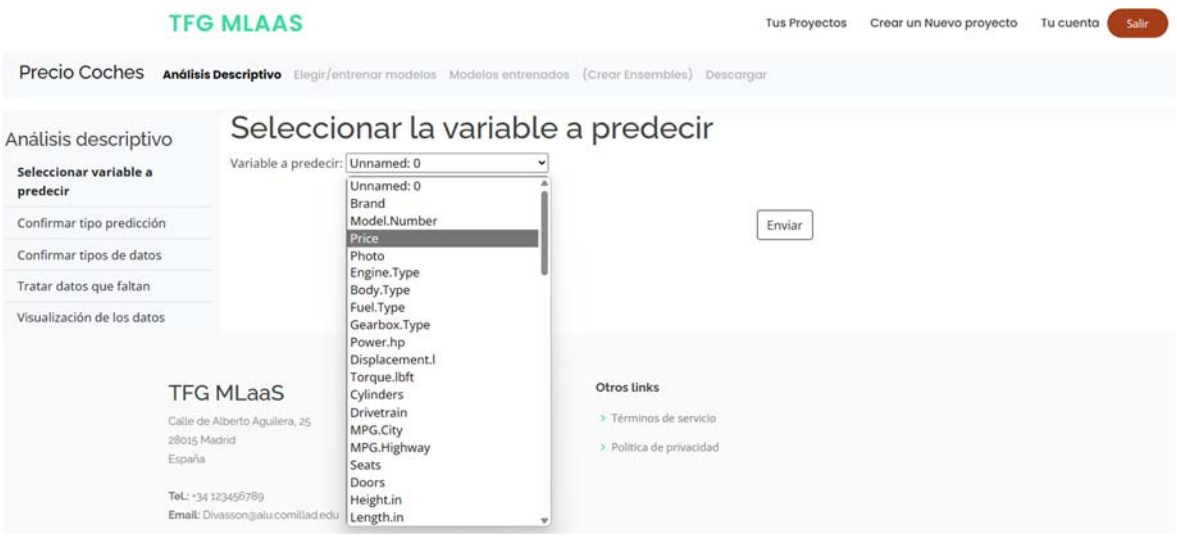


Figura 36. Vista de la selección de la variable a predecir en el proyecto "Precio Coches"

Tras seleccionar la variable a predecir, la plataforma solicita al usuario que confirme que el proyecto se trata de un problema de regresión, puesto que la aplicación ha analizado la distribución de la variable a predecir. Ante este formulario, el usuario confirma que es un problema de regresión y no de clasificación.

Figura 37. Vista de la confirmación del tipo de problema a solucionar en el proyecto "Precio Coches"

Después de confirmar que este proyecto es un proyecto de regresión, la plataforma pregunta al usuario si quiere cambiar la forma en la que va a tratar los datos subidos a la plataforma. Un ejemplo es la columna "Unnamed: 0", que en el conjunto de datos descargados no tenía nombre, y es simplemente un contador que asigna un número a cada observación. La aplicación, tras analizar esta variable, ha concluido que pese a estar representada con

números es una variable categórica, y por lo tanto, en la siguiente vista, tratar datos, se espera que recomiende eliminarla por no aportar información.

El usuario guarda esta configuración tras inspeccionar cómo va a tratar la plataforma su conjunto de datos, seleccionando el botón de "Guardar y continuar".

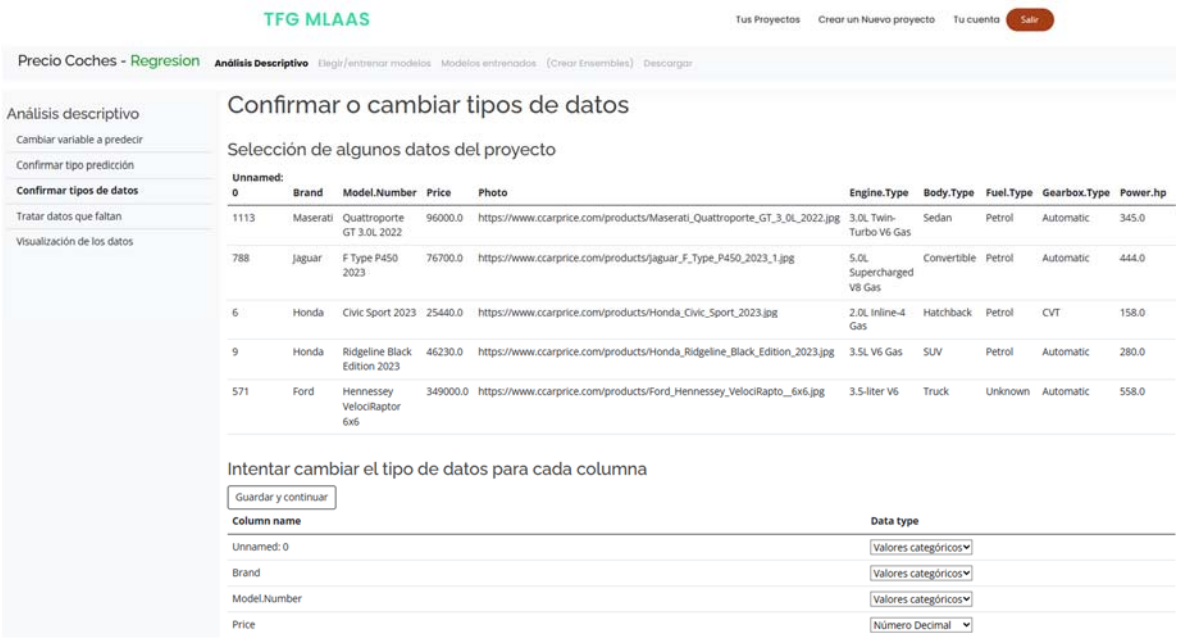


Figura 38. Vista de la confirmación de los tipos de datos

A continuación, la plataforma guía al usuario a la vista donde se le recomienda cómo tratar, tanto los datos que les faltan, como algunas columnas que el programa identifica que podrían ser susceptibles de ser borradas por la información que añaden. Entre las columnas que se notifica al usuario como susceptibles de ser borradas aparece la columna "Unnamed: 0", pues cada observación tiene un valor distinto de esta columna. El usuario acepta la configuración propuesta por la plataforma, y selecciona continuar con el proceso de preprocesado y análisis descriptivo de los datos.

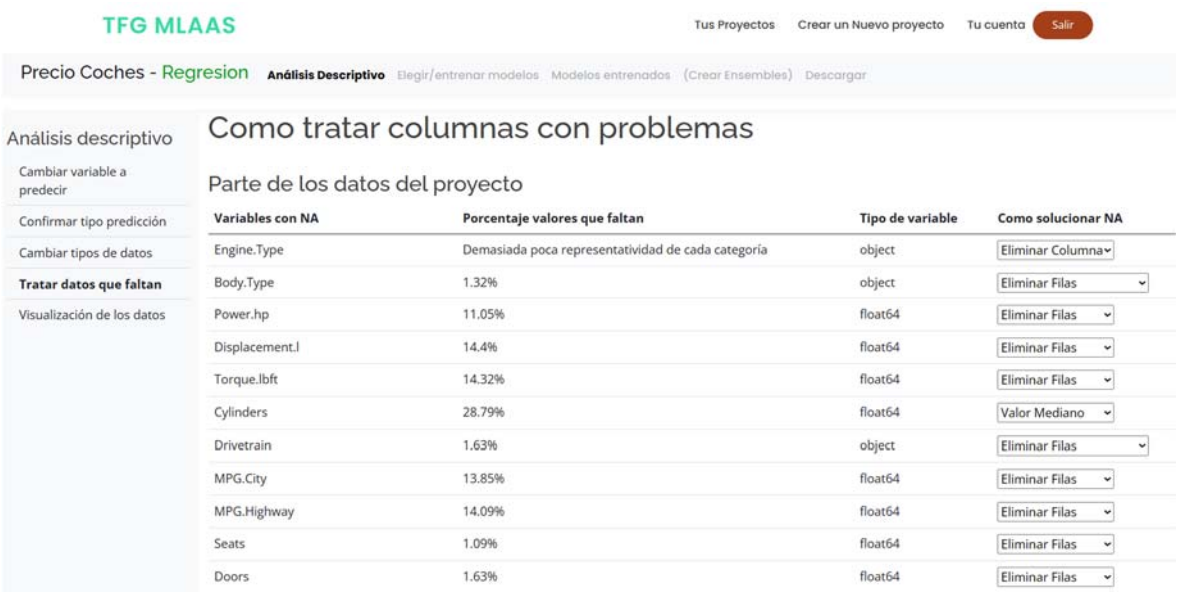


Figura 39. Vista de las recomendaciones sobre cómo tratar el conjunto de datos en el proyecto "Precio Coches"

La última etapa de la fase de preprocesado y análisis de datos es la visualización de estos. En esta vista, el usuario ha seleccionado comparar las variables "Marca" y "Precio" en la herramienta interactiva, donde se representa, un *boxplot* de la distribución del precio para cada marca.

Figura 40. Vista de interfaz de visualización EDA comparando la marca y el precio

Una vez el usuario ha hecho todas las visualizaciones que considere con esta herramienta de visualización de datos, la plataforma redirige al usuario a la vista de entrenar modelos. El usuario, como quiere entrenar todos los modelos de machine learning, buscando automáticamente los hiperparámetros para todos ellos, elige el modo de entrenamiento "Normal". Una vez confirmado el modo de entrenamiento "Normal", la plataforma informa al usuario, mediante la rueda giratoria que se están entrenando los modelos de machine learning.

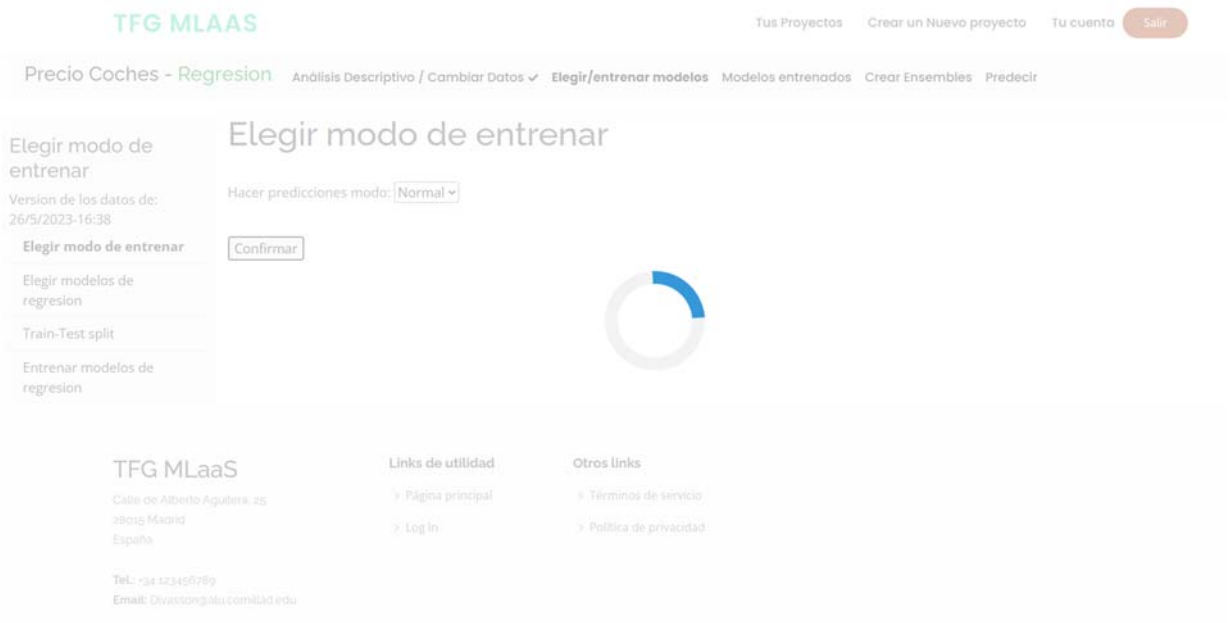


Figura 41. Vista del entrenamiento según el modo "Normal"

Una vez se entrenan todos los modelos de machine learning habilitados en la plataforma (pues el usuario eligió el modo "Normal"), el usuario decide intentar hacer modelos más robustos aprovechando la técnica del Ensemble. Para ello, el usuario se va al apartado destinado para crear Ensembles y selecciona algunos modelos para juntarlos y así intentar mejorar la precisión. Uno de los Ensembles que creará el usuario entre otros será juntando el modelo de Random Forest, la regresión lineal, y la red neuronal, como se ve en la Figura 41.

Figura 42 – Vista para crear Ensembles

A continuación, el usuario se desplaza a la pantalla para valorar los modelos de machine learning. En esta pestaña puede ver en la parte izquierda todos los diferentes modelos y Ensembles entrenados, ordenados por orden descendiente de la raíz del error cuadrático medio (RMSE) con el conjunto de prueba. El usuario así ve que el modelo que mejor funciona es el Ensemble que junta el modelo de Random Forest, y la regresión lineal, con una métrica inferior a los 12.000. Además, en este menú lateral izquierdo puede ver cómo los Ensembles tienen errores inferiores que los modelos de los que se componen.

Si el usuario quiere ver cómo se comporta este Ensemble, puede hacerlo inspeccionando las gráficas de la vista de valoración y comparación de modelos. En la primera gráfica, que dibuja la dispersión de los valores predichos junto frente a los valores reales con el conjunto de prueba, el usuario puede observar cómo se predice el modelo, y cómo de bien se ajusta a la predicción perfecta, definida por la línea dibujada y=x. En la gráfica de la derecha, el usuario puede ver la distribución del error de la predicción del modelo, viendo que, aunque los errores no están centrados en el 0 (caso óptimo), la predicción de este modelo está bastante centrada.

Finalmente, en la gráfica inferior puede ver la distribución de todos los errores de todos los modelos entrenados. En esta gráfica puede ver cómo el modelo que tiene un error más parecido es el modelo de Random Forest, aunque la distribución del error del Ensemble es más puntiagudo sobre su media que el del Random Forest.

Las capturas de a continuación representan en primer lugar la pantalla general (figura 43), y a continuación capturas con más zoom de las distintas partes de esta pantalla, mostrando con más detalle el listado a la izquierda de la pantalla (figura 44), las dos gráficas superiores (figura 45) y la gráfica inferior (figura 46).

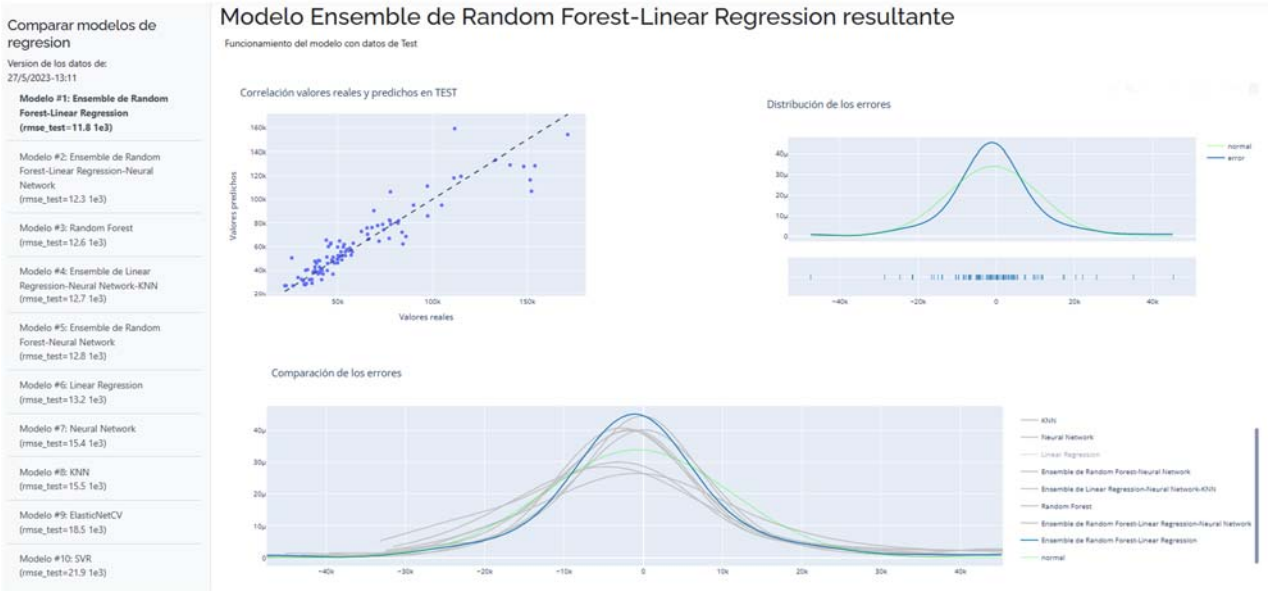


Figura 43. Vista general de la pantalla de valorar modelos

Figura 44. Vista del listado a la izquierda de la pantalla de valoración de modelos ordenando los modelos entrenados

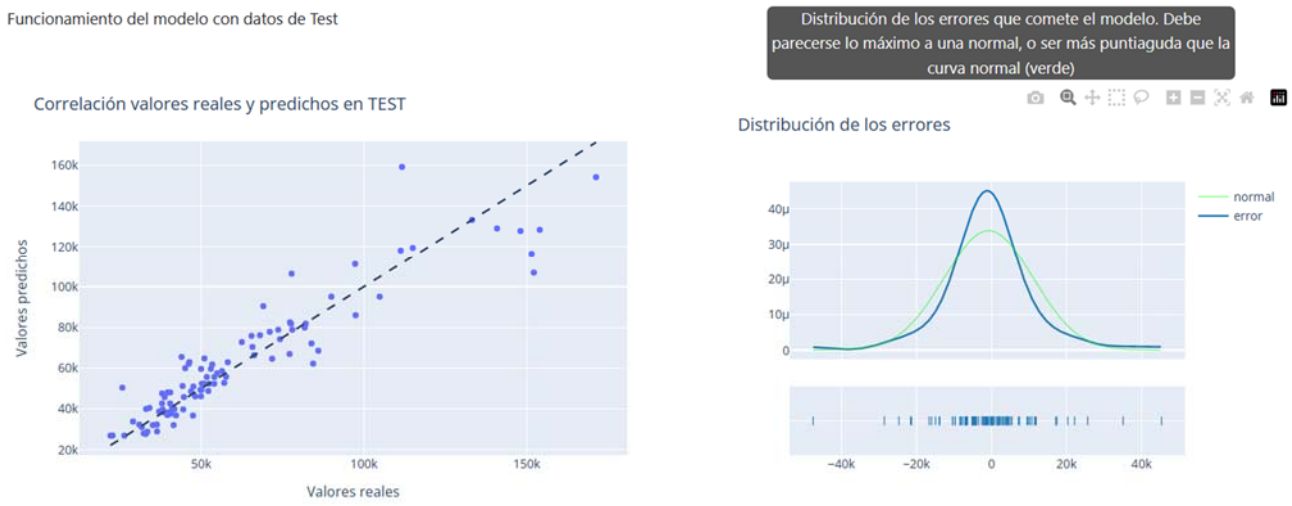


Figura 45. Vista de las dos gráficas superiores de la página de valoración

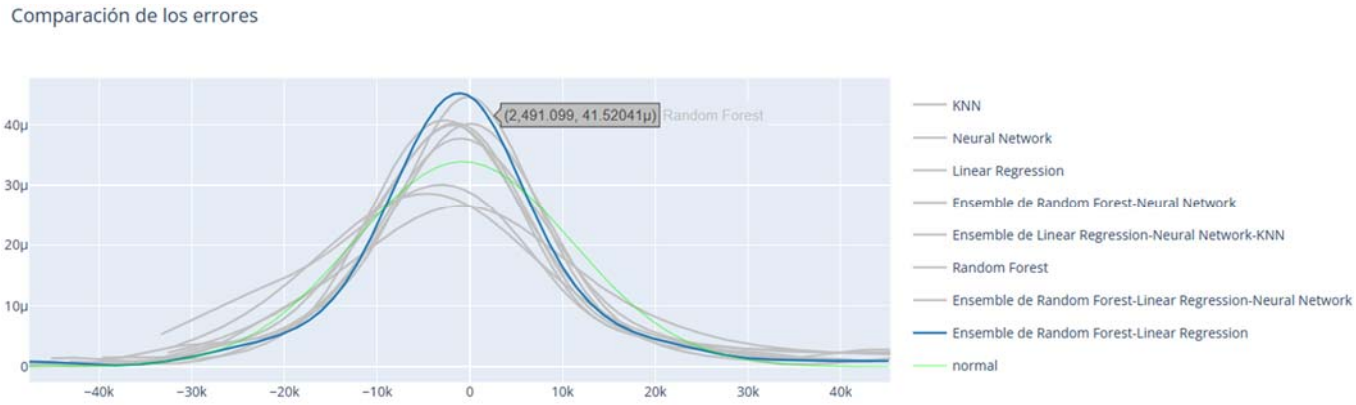


Figura 46. Vista de la gráfica inferior de la vista de comparar modelos

Una vez el usuario ha comparado los modelos que se han entrenados, se dispone a probar cómo funciona la herramienta de predicción, subiendo unos datos nuevos que hasta ahora no había visto la plataforma. Para ello, el usuario se desplaza hasta la página de predicción, selecciona el modelo con el menor error, el Ensemble de Random Forest y la regresión lineal, y selecciona un conjunto de datos que no había visto la plataforma sin etiqueta para que ésta realice la predicción.

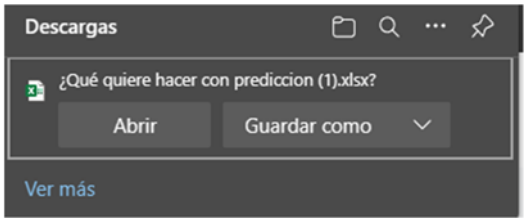


Figura 47. Descarga del archivo que contiene la predicción

Tras inspeccionar la predicción, el usuario ve que el error medio es de -158,5€, y tiene un RMSE de 9.130€, y que el modelo predice muy bien sobre nuevos datos. El usuario, utilizando Excel, hace la siguiente gráfica para ver cómo se distribuyen los errores del modelo con estos nuevos datos.

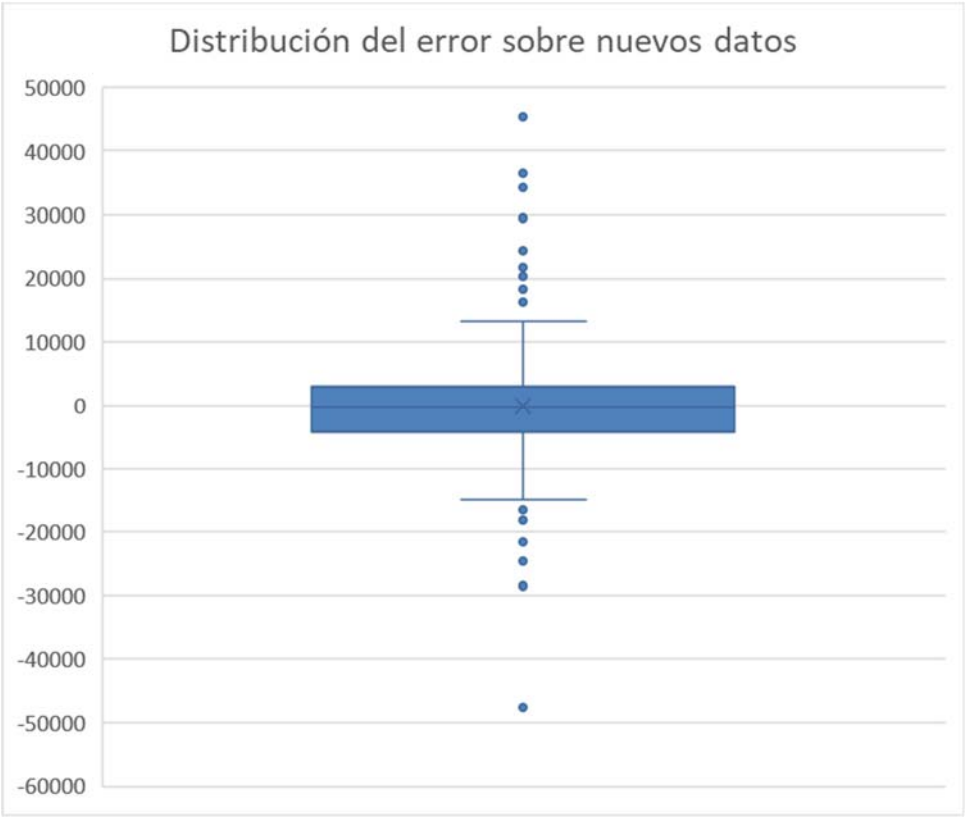


Figura 48. Box-plot del error en la predicción

Capítulo 7. CONCLUSIONES

En este proyecto se ha desarrollado una plataforma de extracción de conocimiento a partir de datos arbitrarios, conocido comercialmente como "Machine Learning as as Service" (MLaaS).

La mayoría de los actuales competidores con plataformas de MLaaS proporcionan soluciones en las que el usuario no tiene que escribir líneas de código para entrenar modelos de machine learning (soluciones No-Code). Aunque estas soluciones ofrecen tanto potencia como personalización, carecen de una interfaz que guíe al usuario en el proceso de entrenar modelos de machine learning, y no ofrecen recomendaciones proactivas sobre cómo tratar los datos o entrenar los modelos. Estas herramientas están diseñadas para brindar una gran ayuda a los científicos de datos, pero no se han diseñado de manera intuitiva para su uso generalizado por parte de la población en general.

La plataforma desarrollada aquí pretende acercar el mundo del machine learning a todos los usuarios, independientemente de sus conocimientos de programación o de las fases de un proceso de explotación y análisis de datos. Para conseguirlo, la plataforma los guía a través de todas las fases necesarias para el entrenamiento de modelos y algoritmos de aprendizaje supervisado, proponiéndoles de manera proactiva formas en las que realizar cada fase de las que se compone un proyecto de machine learning, en función del conjunto de datos.

El proyecto se ha subido a un repositorio en GitHub en internet (https://github.com/Divasson/TelecoMLaaS), junto con una sencilla guía de instalación para que cualquier usuario pueda perfectamente utilizar esta plataforma desarrollada, ejecutándola en sus propios ordenadores.

La plataforma desarrollada tiene las siguientes **ventajas**:

- Tiene una interfaz intuitiva, simple y amigable, que ayuda a los usuarios inexpertos a utilizar la plataforma.

- Contiene un proceso de autenticación de usuarios mediante el cual un usuario no puede acceder a los proyectos de los demás y viceversa, garantizando así privacidad.
- Guía al usuario por todas las fases necesarias dentro de un proceso de creación de modelos de machine learning, sin dejar que un usuario se salte ninguna. De esta manera un usuario inexperto no necesitará saber las etapas que tiene que realizar, puesto que la plataforma le obliga a pasar por todas ellas.
- En las situaciones en las que el usuario tiene que tomar decisiones, la plataforma propone al usuario una solución por defecto, después de analizar los datos de los que depende cada decisión. De esta manera, en el caso en el que un usuario no tenga conocimientos sobre qué decisión tomar en una de las etapas del entrenamiento de modelos de predicción, dejando la opción por defecto se asegura una solución como mínimo aceptable.
- El usuario inexperto no necesita conocer el funcionamiento de los algoritmos de machine learning, seleccionando por adelantado qué modelo quiere entrenar. La plataforma entrena todos los modelos habilitados de machine learning para que posteriormente el usuario elija el mejor de todos los modelos entrenados.
- La plataforma tiene la capacidad de seleccionar los mejores hiperparámetros para cada uno de los modelos de machine learning, utilizando la librería Optuna. De esta forma, el usuario no tiene por qué saber los hiperparámetros con los que entrenará el modelo, sino que la plataforma los elije automáticamente.
- Este proyecto incluye una herramienta de visualización interactiva de datos (EDA), que permite a los usuarios entender cómo se comportan y relacionan sus datos, con gráficos que varían para visualizar mejor estos datos.
- Este proyecto incluye una vista para valorar y comparar varios modelos de machine learning en una sola página web, gracias a métricas conjuntas para ordenarlos y gráficas interactivas para comprobar su éxito en la predicción sobre el conjunto de prueba.
- Esta plataforma permite a los usuarios juntar, de una manera muy sencilla, varios modelos de machine learning de una forma ponderada para que el Ensemble resultante sea un modelo más robusto que los modelos que lo crean.

- A la hora de realizar una predicción sobre un nuevo conjunto de datos, la plataforma, que procesa los datos como se procesaron los datos de entrenamiento del conjunto de datos del proyecto, devuelve exactamente el mismo fichero que el usuario subió, solo que con una columna extra conteniendo la predicción. Si alguna observación tuviese que ser borrada por faltarle algún dato, seguirá estando en el fichero descargable, solo que en la columna que contiene la predicción aparecerá "Faltan Datos".

No obstante, este proyecto tiene los siguientes **inconvenientes** que podrían ser resueltos en **trabajos futuros**:

- Actualmente, la plataforma solo soporta las métricas *RMSE* y *Balanced Accuracy* para regresión y clasificación respectivamente. Estas métricas, aunque son las más utilizadas, no son aptas para todos los problemas, y el usuario debería poder elegir qué métricas quiere utilizar a la hora de entrenar y valorar los modelos de machine learning.

  Para solucionar este inconveniente, deberá añadirse a la plataforma, en el apartado de entrenamiento, que en el modo "Experto" el usuario pueda elegir la métrica que quiere para entrenar sus modelos.
- Otro de los inconvenientes de esta plataforma es que no está subida en ningún servidor en internet. Así que actualmente, para utilizarla, los usuarios tienen que ejecutar esta aplicación web desde su propio ordenador, después de descargar el proyecto de GitHub.

  Para solucionarlo, se deberá primero crear un contenedor de la aplicación, y posteriormente subir ese contenedor a algún servidor en la nube, como puede ser Amazon Web Services.
- En esta solución solo se han implementado algoritmos de machine learning de aprendizaje supervisado. Aunque este tipo de machine learning es el que más fácilmente puede entenderse por usuarios sin conocimientos de inteligencia artificial, no se han implementado algoritmos de aprendizaje no supervisado con técnicas tan positivas como el clustering.

En un futuro se deberá añadir técnicas de aprendizaje no supervisado como clustering o detección de outliers.

- Los algoritmos de machine learning implementados solo funcionan para datos tabulares en los que no importa el orden. Aunque estos datos representan una gran parte de los datos que pueden tener almacenados una empresa, existen otros algoritmos de machine learning que podrían ser introducidos en la aplicación.

  En un futuro habría que implementar algoritmos para series temporales o para análisis de textos como redes LSTM (Long Short Time Memory) para series temporales, o modelos Word2Vec para textos.

Aunque se propone seguir mejorando esta plataforma añadiendo más personalización para los usuarios y añadiendo modelos más complejos, la desarrollada cumple con el objetivo de acercar el machine learning a los usuarios más inexpertos, dándoles la posibilidad de crear, de una manera sencilla, modelos de predicción con datos arbitrarios.

Como conclusión, este proyecto puede ayudar a muchos usuarios que deseen aprovechar el poder predictivo de los algoritmos de machine learning, pero que no tengan conocimientos de programación ni de machine learning para llevarlo a cabo, como por ejemplo las PYMES en España o incluso departamentos de grandes empresas que no dispongan de científicos de datos.

Capítulo 8. BIBLIOGRAFÍA

[1] Marr, B. (2018). *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read.* Forbes. https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/

[2] Django. (2023). *Django.* Django Project. https://www.djangoproject.com/

[3] Django. (2023). *Django - Cross Site Request Forgery protection¶*. Django Project. https://docs.djangoproject.com/en/4.2/ref/csrf/

[4] Django. (2023). *Django Documentation Initiation.* Django Project. https://docs.djangoproject.com/en/4.2/intro/tutorial01/

[5] Django. (2023). *Django Templates*. Django Project. https://docs.djangoproject.com/en/4.2/ref/templates/language/

[6] Scikit-Learn. *LinearRegression*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[7] Scikit-learn. *LogisticRegression*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[8] Scikit-learn. *KNeighborsClassifier*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[9] Scikit-learn. *KNeighborsRegressor*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

[10] Scikit-learn. *ElasticNetCV*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.linear_model.ElasticNetCV.html

[11] Scikit-learn. *SVC*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.svm.SVC.html

[12] Scikit-learn. *SVR*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.svm.SVR.html

[13] edX - 2U Company. Bootstrapping. https://www.mastersindatascience.org/learning/machine-learning-algorithms/bootstrapping/

[14] Scikit-learn. *RandomForestClassifier*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[15] Scikit-learn. *RandomForestRegressor*. scikit-learn. https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[16] Team, K. *Keras documentation: About Keras*. https://keras.io/about/

[17] Optuna Team. *Optuna*: A hyperparameter optimization framework. https://optuna.readthedocs.io/en/stable/

[18] Plotly Team. *Plotly*. https://plotly.com/python/

[19] Meta, Facebook. *ML applications*. https://engineering.fb.com/category/ml-applications/

[20] Apple. (2022). *Machine Learning - Core ML - Create ML.* https://developer.apple.com/machine-learning/

[21] Microsoft. *Microsoft Azure*. https://azure.microsoft.com/es-es/free/machine-learning/search/

[22] Amazon Web Services. *Machine Learning en AWS* . https://aws.amazon.com/es/machine-learning/

[23] Google Cloud. *Soluciones de IA y aprendizaje automático* . https://cloud.google.com/solutions/ai?hl=es

[24] Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130.

[25] Jacob, F., & Ulaga, W. (2008). *The transition from product to service in business markets: An agenda for academic inquiry. Industrial Marketing Management*, 37(3), 247-253. https://10.1016/j.indmarman.2007.09.009

[26] Fortune Business Insights. (2021). *Everything as a service market size & growth by 2028.* https://www.fortunebusinessinsights.com/everything-as-a-service-xaas-market-102096

[27] Bhat, I., & Data Bridge Market Research. (2021). *Machine learning as a service (MLaaS) market*. https://www.linkedin.com/pulse/machine-learning-service-mlaas-market-estimated-2028-likely-indu-bhat/

[28] Jobted. Sueldo del Data Scientist en España . https://www.jobted.es/salario/data-scientist

[29] Amazon Web Services. *Amazon SageMaker*. https://www.amazonaws.cn/en/sagemaker/

[30] Google, A. I. *Introduction to Vertex AI* . https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform

[31] Contino.Google Vertex AI: A Powerful Tool to Solve Your Machine Learning Woes. Contino | Global Transformation Consultancy. https://www.contino.io/insights/google-vertex-ai

[32] Microsoft Azure. *Azure Machine Learning - ML as a Service | Microsoft Azure.* https://azure.microsoft.com/en-us/products/machine-learning

[33] Bartolo, A. (2019). *Step-By-Step: Getting Started with Azure Machine Learning. TECHCOMMUNITY.MICROSOFT.COM.* https://techcommunity.microsoft.com/t5/ITOps-Talk-Blog/Step-By-Step-Getting-Started-with-Azure-Machine-Learning/ba-p/331327

[34] IBM Watson. (2022). *IBM Watson Studio - Información general*. https://www.ibm.com/es-es/cloud/watson-studio

[35] Budgeting Solutions. *IBM watson studio*. https://www.budgetingsolutions.co.uk/technologies/ibm-watson-studio/

[36] WEKA. *Machine Learning Project at the University of Waikato in New Zealand.* https://www.cs.waikato.ac.nz/ml/index.html

[37] Brownlee, J. (2016). *A tour of the weka machine learning workbench*. https://machinelearningmastery.com/tour-weka-machine-learning-workbench/

[38] Ryan Houma. (2020). Agile vs. waterfall methodology: Making A choice. http://apifuse.io/blog/agile-vs-waterfall-methodology/

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

El proyecto de Machine Learning as a Service desarrollado nace con la vocación de reducir la alta barrera de entrada de las herramientas de predicción de Machine Learning, democratizando un servicio muy importante para las empresas con pocos recursos, facilitando su toma de decisiones.

Así, este proyecto inicialmente no cobrará por su servicio, para no dificultar el acceso a estas herramientas a quien no tiene recursos económicos. Por lo tanto, el proyecto se alinea con el octavo objetivo de desarrollo sostenible, que es el de trabajo decente y crecimiento económico pues favorece que las empresas sin tanta capacidad económica puedan competir con grandes empresas.

Figura 49. Icono del octavo objetivo de desarrollo sostenible. Fuente: ONU

Además, este trabajo se alinea con el noveno objetivo de desarrollo sostenible de la ONU: Industria, Innovación e Infraestructura, al desarrollar una solución que acerca el complejo e innovador mundo de la inteligencia artificial a todos los estratos de la población, independientemente de sus conocimientos sobre el tema.

Al eliminar las barreras de entrada y hacer que la creación de modelos de predicción sea accesible para todos los usuarios, se promueve la innovación y el crecimiento de la industria

tecnológica. Además, al simplificar y automatizar el proceso de construcción de modelos, se reduce el tiempo y los recursos necesarios, lo que resulta en una infraestructura más eficiente y sostenible. Este proyecto facilita la adopción de tecnologías avanzadas, promoviendo así la modernización de la industria y el impulso de la innovación en los sectores que utilicen esta herramienta.

Figura 50. Icono del noveno objetivo de desarrollo sostenible. Fuente: ONU

Finalmente, esta plataforma desarrollada se alinea estrechamente con el décimo objetivo de desarrollo sostenible de la ONU: Reducción de las desigualdades. A través de esta plataforma accesible, se abren nuevas oportunidades derribando barreras para que personas de todos los estratos sociales puedan beneficiarse de la tecnología del aprendizaje automático. Al proporcionar una interfaz intuitiva y simplificada, se elimina la necesidad de conocimientos técnicos avanzados a la vez que se ofrecen interfaces distintos para usuarios con diferentes niveles de habilidad. Con esto, el proyecto desarrollado ayuda a reducir la brecha digital y las desigualdades de acceso a la tecnología, permitiendo que todos los usuarios puedan beneficiarse de las ventajas y oportunidades que brinda el Machine Learning en la toma de decisiones.

Figura 51. Icono del décimo objetivo de desarrollo sostenible. Fuente: ONU