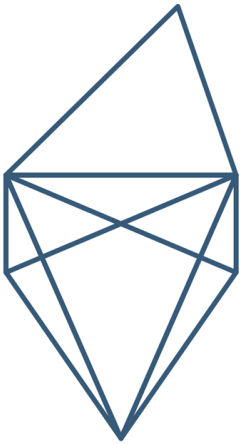
Automatización de módulos en Odoo y

desarrollo de modelos predictivos para páginas web

2ºDAM



Alumno: Alex Monllor Jerez

CIP FP BATOI - Tutor del proyecto: Antonio Vicente Santos Silvestre

AGRADECIMIENTOS

Primero de todo quiero agradecer a todos los compañeros de la empresa que me han ayudado a realizar el proyecto puesto que es muy complicado y se le debió dedicar muchas horas.

He tenido la suerte de contar con un equipo que no puso en ningún momento trabas a mi desarrollo y aprendizaje y me impulsaron a llegar a mis objetivos. La FCT es siempre muy costosa para una empresa ya que debe de dedicar tiempo a sus empleados para instruir a los nuevos alumnos que llegan, pero de esta manera, se han llegado a nuevos horizontes para todos.

Por último, quiero dar las gracias a los profesores que me tuvieron paciencia con mi cabezonería y mi forma de ser tan peculiar, que se esforzaron por entenderme y ayudarme en poder completar esta ardua tarea.

Índice

AGRADECIMIENTOS **1**
1-Introducción **4**
1.1-Motivo de selección 4
1.2-Descripción y objetivos 4
1.3-Posibles aplicaciones prácticas 4
1.4-Lugar de uso 4
1.5-Presentación del proyecto 4
2-Fundamentos generales teóricos y prácticos **5**
3-Presentación de la empresa **6**
4-Preparación del entorno **7**
4.1 Pasos para configurar google analytics 4 y obtener la api key 7
4.2-Tecnologías utilizadas 10
4.2.1-Entorno Virtual 10
4.2.2-Instalar importaciones 10
4.3-Verificar permisos 11
4.4-Exploración métricas 11
4.5-Descarga de datos 12
4.6-Subir CSV a BigQuery 13
5-Forma de entrenamiento y validación **13**
5.1-Forma de entrenamiento 13
5.1.1-Validación cruzada 16
5.1.2-Tipos comunes de validación cruzada 17
5.1.3-Ventajas de la validación cruzada 18
5.1.4-Limitaciones 18
5.2-Verificación del modelo 18
5.2.1- Conjunto de prueba (test set) 18
5.2.2- Datos reales de producción 19
5.2.3- Datos históricos no utilizados 19
6-Tipos de modelo **20**
6.1- Gráfico de Valores Reales vs. Predicciones 21
6.2- Gráfico de Distribución de Residuos 22
6.3-Matriz de correlación 24
7-Problemas detectados y soluciones **26**
7.1-Subajuste y Sobreajuste 26
7.1.1-Subajuste (Underfiting): 26
7.1.2-Sobreajuste (Overfiting): 26
7.2-Configuración del Pipeline 27
7.2.1-Importaciones necesarias: 27
7.2.2-Estructura del Pipeline: 27

7.3-Modelos que soportan y no soportan entrenamiento incremental 29
7.3.1-Modelos basados en árboles de decisión: 29
7.3.2-Modelos que sí soportan partial_fit: 30
7.4-Problemas y soluciones 31
7.4.1-¿Qué hacer si tu modelo no soporta partial_fit? 31
7.4.2-GridSearchCV 31
8-Hiperparametrizacion y opciones a usar 32
8.1-¿Qué es la hiperparametrización? 32
8.2-Comparativa de Metodologías antiguas 32
8.3-Comparativa Random Search VS Deep Research 33
8.3.1- ¿Qué es Random Search? 33
8.3.2- ¿Qué es Deep Research? 34
8.3.3-Comparativa: Random Search vs Deep Research 35
8.3.4-¿Cuándo Usar Cada Enfoque? 36
9-Odoo 37
9.1-Equipo necesario 37
9.2-Funcionalidad 37
9.2.1-Comprobar credenciales 37
9.2.2-Descarga e importación de datos 37
9.2.3-Sistema de entrenamiento 38
9.2.4-Integración y seguridad 38
10-Conclusiones 38
10.1-Resultados Obtenidos 38
10.2-Puntos Pendientes 39
10.3-Tiempo dedicado 39
10.4-Valoración personal 39
10.5-Conexión con las prácticas 39
10.6-Consejos destacables 39
11-Necesidades y sugerimientos de formación 40
12-Bibliografía 40
13-Recursos utilizados 40
14-Diccionario 41

1-Introducción

1.1-Motivo de selección

Las prácticas las elegí por el hecho de que me garantiza un proyecto novedoso que podía desarrollar durante el tiempo que trabajaba y me pareció una buena baza para ahorrar tiempo aun sabiendo lo costoso que iba a ser realizarlo. Además, viene perfecto para compaginarlo en el futuro con mis estudios Universitarios sobre IA. Aun sabiendo que no podría llegar a realizarse al completo el proyecto decidí aceptar el reto y realizar las prácticas con empeño.

1.2-Descripción y objetivos

El proyecto consiste en la documentación de metodologías de entrenamiento de Machine learning y el entrenamiento de un modelo predictivo con la mayor tasa de acierto posible. El modelo se focaliza en detectar el comportamiento de las personas al entrar a la página y cuando caigan al umbral de usuarios que no van a realizar una compra se le aplicará un pequeño descuento para intentar obtener una compra mínima aunque sea.

1.3-Posibles aplicaciones prácticas

El uso básico del modelo es la capacidad de detectar la posibilidad de que un cliente compre en nuestra PW, aunque el proyecto se compone de otros 2 núcleos que es todo el desafío de la creación de los archivos desde cero y la API para conectar el modelo con la PW final. Es verdad que en un futuro se le podría nombrar nuevas tareas como detección de ip malignas para vetar a usuarios indeseados o incluso la posibilidad de cambiar automáticamente el idioma de la página si detectase que el usuario es de un país distinto, pero es demasiado complicado para realizarlo de manera pronta y rápida.

1.4-Lugar de uso

Los archivos, modelos y API generados durante las prácticas se quedan en la empresa Inprofit para su uso personal y comercial exclusivo de sus clientes, aunque las malas lenguas ya dicen que las grandes compañías están empezando a integrar estos modelos en sus páginas.

1.5-Presentación del proyecto

Como ya he explicado en los puntos anteriores, durante todo este proyecto mínimo vamos a ver cómo funcionan los archivos y cómo comprender los resultados del entrenamiento del modelo. También habrá unas pequeñas guías de cómo se deben realizar algunos pasos por si el usuario decide trabajar de manera independiente se le explicaran algunos funcionamientos y también la parte más importante que tiene que ver con la api de Google Analytics pues se necesita para la descarga de datos. Que no será de gran ayuda si no tiene el usuario una PW con tráfico.

2-Fundamentos generales teóricos y prácticos

El proyecto realmente tiene un enfoque más dedicado a lo que deberían ser Asignaturas de DAW, pero aun asi se adaptó para que fuese apto para DAM:

1ºCurso

Programación

Es un corso vital aun si no supuse programar esta asignatura al menos te llega a enseñar la base para poder entender el código que hay ante ti, en el mundo actual en el que en las empresas la creación de código está empezando a ser dominado por la creación de la IA, la capacidad de en un solo vistazo entender que es lo que se está haciendo es de vital importancia. Por eso es una asignatura importante, necesaria y espléndida.

Entornos de desarrollo

Seguramente, habría tenido grandes problemas para trabajar durante los cursos sin esta asignatura impartida por Empar. La facilidad y la eficiencia son dotes necesarias para la creación y modificación de código y con las herramientas y técnicas que te enseñan es perfecto para reducir los errores al momento de trabajar y optimizar el tiempo en trabajos sencillos y monótonos. Además de que siempre hay que tener una copia de seguridad para poder trabajar tranquilo por lo que pueda suceder.

Bases de datos

La manera en la que nos impartieron la asignatura fue un poco… especial, pero en el fondo sí que se llegó al objetivo que era comprender cómo funcionan los datos. Esto me ayudó sobre todo cuando trabajé con Google Analytics y las primeras consultas que hacía ya que ayudaban a la comprensión y el entendimiento más fluido de mi flujo de trabajo lo que hacía que no tardase tanto en seguir avanzando.

2ºCurso

Acceso a datos

Esta asignatura no voy a mentir, la tuve que recuperar y en mi ignorancia no sabía cuán importante era hasta que realice el proyecto. Trabajo con API 's, datos transaccionales y más partes de la asignatura. Fue en ese momento que me percate del verdadero valor de su importancia y de cómo se debía trabajar. La asignatura me enseñó mucho más de lo que esperaba y la verdad es que si que era mas importante de lo que creía.

Desarrollo de interfaces

Incluimos este apartado ya que fue la asignatura encargada de instruirnos en Python y la cual nos dio la base para poder trabajar con el proyecto. Gracias a los conocimientos básico se agilizo el aprendizaje del lenguaje y la construcción del proyecto

Programación de servicios y procesos

Esta asignatura no es de una relevancia real en el proyecto pero sí que es cierto que tiene su propio peso de relevancia. Al haber trabajado con los procesos subyacentes he podido optimizar algunos ligeros aspectos del proyecto mejorando su eficiencia

Sistemas de gestión empresarial
Esta asignatura es crucial para que el proyecto pueda desarrollarse, hay que tener muy en cuenta que sin ella no se podrían personalizar módulos de Odoo y es algo muy necesario para la adaptación del proyecto.

Gracias al trabajo realizado en el curso se pueden desarrollar módulos de un gran agrado visual lo que facilita el desarrollo y entrenamiento de los modelos predictivos, además de que aumenta la facilidad de la comprensión de cómo se deben realizar las tareas pudiendo alguien sin conocimientos, mínimamente comprender cómo se realizan y entrenan los modelos en unos sencillos pasos.

Además de la ayuda muy necesaria de haber trabajado con Python ya que fue el primer contacto con el lenguaje el cual sirvió para el desarrollo más fluido del proyecto.

3-Presentación de la empresa

Inprofit(Página Web) es una empresa especializada en marktech que combina marketing, tecnología y estrategia para ofrecer soluciones innovadoras. En el área de marketing destacan servicios como SEO, social media, neuromarketing y analítica digital.

En tecnología, impulsan la transformación digital con herramientas como inteligencia artificial, VR & AR, y soluciones IoT. Su enfoque estratégico incluye consultoría para CEOs, expansión de franquicias y estudios de viabilidad. Con un compromiso con la transparencia y la creatividad, buscan sorprender a sus clientes sin retenerlos forzadamente.

Actualmente, desarrollan módulos de IA para optimizar la retención de clientes potenciales, complementando su experiencia en posicionamiento SEO.

4-Preparación del entorno

4.1 Pasos para configurar google analytics 4 y obtener la api key

1. Crear una propiedad en Google Analytics 4 (GA4)

Antes de configurar la API, es necesario crear una propiedad en Google Analytics 4 si aún no la tenemos, estos son los pasos a seguir:

1. Accede a Google Analytics.
2. Inicia sesión con tu cuenta de Google.
3. Haz clic en el botón **Administrar** (es un icono de engranaje ⚙️) en la esquina inferior izquierda.
4. En la columna de la cuenta, selecciona tu cuenta o crea una cuenta.
5. En la columna de propiedades, haz clic en **Crear propiedad**.
    - Ingresa un nombre para la propiedad(PW o APP).
    - Configura la zona horaria y la moneda según tus necesidades.
6. Haz clic en **Siguiente** y completa la configuración de tu empresa.
7. Una vez creada la propiedad, configura la **corriente de datos** proporcionando la URL de tu sitio web o la aplicación correspondiente.

   En mi caso sería poner la URL de una de las WEBS de Inprofit, pero ya tenía dichos datos preparados cuando empecé el proyecto entonces solo tuve que encontrar los pasos a seguir para realizar las tareas.

2. Crear un proyecto en Google Cloud Console

1. Accede a Google Cloud Console.

2. Inicia sesión con la misma cuenta de Google que usas en Google Analytics.

3. Crea un nuevo proyecto o selecciona uno existente desde el menú desplegable en la parte superior.

   De la misma manera aquí ya están cargadas las bases de datos y se puede trabajar. Al principio este era el formato que se utilizaba para generar el modelo y ver los datos, pero resultaba muy costoso comprender la información por lo que se desestimó rápido este formato de entrenamiento, el único remanente que se quedó es el archivo que veremos a posterior de subir datos a BigQuery que se utilizaba para la generación de las tablas poder acceder a ellas con las consultas.

   El mayor problema es que el coste computacional es muy alto puesto que al crear los modelos y las consultas estas consumiendo diferentes recursos y la factura de un solo dia llegar a ser de 40 euros y un consumo de 6 TB por lo que se descartó el formato directamente y no se retomo.

3. Habilitar la API de Google Analytics Data

Google Analytics 4 utiliza la API de Google Analytics Data para generar informes.

1. Dentro del proyecto, ve al menú lateral izquierdo y selecciona:

   - **API y servicios > Biblioteca**.

2. En el buscador, escribe: **Google Analytics Data API**.

3. Selecciona la API y haz clic en **Habilitar**.

   De la misma manera la API en la empresa estaba activada y se corroboró que así fueran los pasos para acceder a ella. Es importante porque se necesitará para cuando se hagan las conexiones al **GA4** pueda autenticarse.

4. Crear credenciales para acceder a la API

Para interactuar con la API, debes configurar una cuenta de servicio:

1. Ve al menú lateral izquierdo y selecciona:
    - **IAM y administrador > Cuentas de servicio**.
2. Crea una nueva cuenta de servicio:
    - Asigna un nombre a la cuenta.
    - Elige un rol, como **Editor** o uno personalizado que tenga permisos suficientes.
3. Guarda la cuenta y descarga el archivo JSON que contiene la clave privada.

    Es de vital importancia que el archivo no se pierda ni se difunda puesto que con él, otro usuario podría acceder a las BD y más de nuestro GA4 y eso sería un grave problema de seguridad.

5. Configurar permisos en la propiedad de GA4

1. Accede a tu propiedad de GA4 desde Google Analytics.
2. Ve a la configuración de usuarios en **"Configuración de acceso a la cuenta"**.
3. Añade el correo electrónico de la cuenta de servicio creada anteriormente con permisos de **Analista** o superiores.

    Este paso es de vital importancia puesto que si no se cumple cuando se realicen las peticiones se declinaran ya que no contaremos con los permisos ni para descargar los datos ni para subirlos y por mucho que tengamos el .json no servirá de nada.

6. Obtener el ID de la propiedad en Google Analytics(Opcional)

1. Dentro de Google Analytics, accede a la propiedad que configuraste.
2. Ve a **Administrar** y selecciona la propiedad correspondiente.
3. Copia el **ID de la propiedad**, que será necesario para descargar y trabajar con los datos a través de la API.

Este paso puede obviarse ya que existe un archivo que nos generará todas las propiedades de nuestras BD que tenemos en el GA4. La única diferencia será que ya tendremos las ID y podremos realizar directamente las descargas.

4.2-Tecnologías utilizadas

4.2.1-Entorno Virtual

Se creará un entorno virtual con python donde realizaremos todas las pruebas con el comando:

python -m venv mi_venv

Al realizar el VE en windows pueden ocurrir algunos problemas como que la ejecución de los scripts esté desactivada en dicho caso se deberá de acceder al PowerShell en modo Administrador y ejecutar el siguiente comando:

Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

Y con este otro se devuelve la política a su estado original

Set-ExecutionPolicy -Scope Process -ExecutionPolicy LocalMachine

COMANDO PARA ACTIVAR EL ENTORNO

.\mi_venv\Scripts\Activate

4.2.2-Instalar importaciones

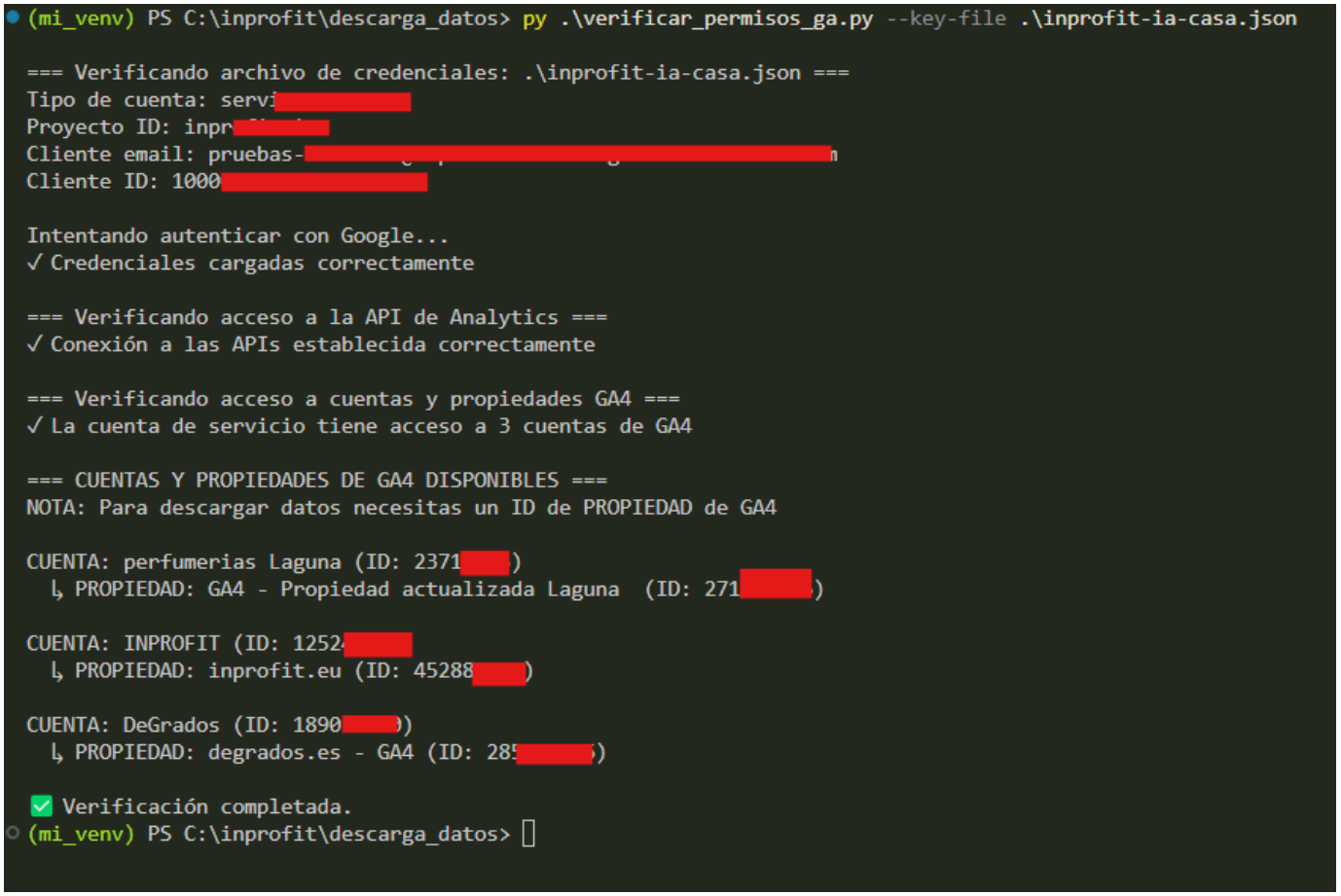
El archivo requirements tiene todas las importaciones necesarias para el funcionamiento de todo el proyecto, con solo 1 comando instalaremos todas las dependencias(suponiendo que tengamos instalado python ya python en nuestro equipo):

Linux
sudo apt update
sudo apt install python3

Windows
Descargar de la Página Web(Enlace)
pip install -r .\requirements.txt

4.3-Verificar permisos

Primero se verifican los permisos con el archivo .json que nos hemos descargado previamente de GA4, podremos ver que si se autentifica y nos muestra los ID de las tablas que tenemos en nuestro GA4. Se recuerda que el número que necesitamos para descargar los datos es el de propiedad.



4.4-Exploración métricas

Antes de ponernos a extraer los datos, vamos a extraer las métricas. GA4 tiene la posibilidad de extraer hasta un total aproximado de 90 métricas y 400 dimensiones. ¿Pero que son estas 2 características?:

Métricas:Son valores numéricos y los que normalmente vamos a trabajar ya que panda utiliza estos valores. Van desde los usuarios activos las acciones que realizan como los clicks de ratón y las conversiones que es el valor numérico de su comportamiento que oscila entre 0 y 1.

Dimensiones: Son los atributos que nos identifican información no numérica sobre el usuario y sus comportamientos. Este es desde su nacionalidad y dispositivo hasta las páginas en las que estuvo de nuestra web y las horas en las que accede.

4.5-Descarga de datos

El archivo realiza una conexión con GA4(Google Analytics) para obtener los datos de la BD de la PW de la empresa especificada para así poder tenerlos de forma local en formato CSV para dárselos al archivo que se encargará de entrenar al modelo. Le especificaremos ID property de donde queremos que nos descargue los datos ya que ese es lo que utiliza para identificar de todas las BD que podemos llegar a tener.

Guarda las listas completas de métricas y dimensiones en archivos CSV para análisis posterior.

Proporciona una interfaz CLI para que el usuario especifique:

Nombre	Funcion
key-file	Ruta del archivo de credenciales
property-id	ID de la propiedad de donde vamos a descargar
start-date	Fecha inicial de donde va a comenzar a descargarse los datos
end-date	Fecha final de donde va a finalizar la descarga de datos
modelo	Tipo de métricas a utilizar (Revisar el archivo de métricas)
output	Nombre de salida del archivo CSV, si no recibe ninguno se le dará uno por defecto.

En resumen, este script es una herramienta útil para explorar y analizar las métricas y dimensiones disponibles en una propiedad de Google Analytics 4, además de permitir la exportación de estos datos para su uso en análisis o modelos predictivos.

El archivo realiza diferentes llamadas a la API de manera que así se puedan extraer todas las métricas/dimensiones que quiera el usuario sin causar un problema en el formato. Tampoco existe límite de registros.

El funcionamiento normal es realizar 1 sola llamada lo que llega a tener un tope de información de la que se puede descargar. Con una fórmula matemática se modifica para que pueda realizar varias llamadas en un solo comando consiguiendo así toda la descarga de datos. El problema ocurría cuanto mayor era el lapso de tiempo y cuanto mayor es el impacto de la web en internet pues recibía más usuarios.

4.6-Subir CSV a BigQuery

Si se quieren realizar consultas desde Bigquery o alojar los datos puesto ya que ya han sido procesados. Así se puede tener la información específica en la nube si lo desea.

5-Forma de entrenamiento y validación

5.1-Forma de entrenamiento

Nombre del parámetro	Función del parámetro
archivos	Nombre del csv a ingresar de donde se va a utilizar los datos para entrenar al modelo
objetivos	Nombre de las columnas de métricas y dimensiones objetivas
modelo salida	Nombre que va a tener el modelo que se está entrenando
salida	Ruta que donde se van a generar el modelo y los informes para ver su información
incremental	Formato de aprendizaje donde se selecciona el mejor modelo
batch-size	Tamaño del lote de entrenamiento(es multiplicado por 10 para reajustar un poco los valores)
epochs	Número de veces que se va a realizar el entrenamiento

(EJEMPLO DE RESULTADO DE EJECUCIÓN DE UN ANTIGUO MODELO DE SOLO 1 C.O.)
=== Evaluación del modelo ===

Resultados de validación cruzada:
R² promedio (CV): 0.9549 ± 0.0165
RMSE promedio (CV): 0.1515 ± 0.0231

Resultados en conjunto de validación:
R²: 0.9476
RMSE: 0.1759
MSE: 0.0309
Informe generado: resultados\informe_modelo.html

AVISO:
Si se entrena el modelo de forma incremental no se podrá realizar de otra manera ya que las metodologías que este implementa son incompatibles con el formato normal y viceversa.

Antes se tenía que volver a ejecutar el comando para ver qué resultado había dado el modelo, así que guarde los datos dentro de un json.


```
{
  "cv": {
    "r2_mean": 0.6794102149471815,
    "r2_std": 0.1060305243125954,
    "rmse_mean": 0.18944349346609674,
    "rmse_std": 0.011272085932592673,
    "mse_mean": 0.036015897137910785,
    "mse_std": 0.004160077819017926
  },
  "validation": {
    "r2": 0.7643959386670585,
    "rmse": 0.22076906306351532,
    "mse": 0.048738979205942404
  }
}
```


El primer bloque nos muestra el resultado predictivo que tiene el modelo y está dividido en 3 valores:
(La mejor opcions es que se acerquen lo máximo posible a 1,0,0)
Valores en los que se considera que el modelo ya es bastante bueno:
(0.9,0.4,1.6)/(0.7,0.2,0.4)

Fórmula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Donde:

- y_i son los valores reales (observados),
- $\hat{y}_i$ son los valores predichos por el modelo,
- $\bar{y}$ es la media de los valores reales.

R2: Tasa de acierto de el modelo

(Que tan preciso es, este valor se ve reducido si los datos con los que es entrenado no tienen correlación(Matriz de correlación))

Fórmula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Donde los términos son los mismos que para el MSE.

Interpretación:

- **RMSE bajo:** El modelo tiene predicciones más cercanas a los valores reales.
- **RMSE alto**: Indica que las predicciones están alejadas de los valores reales en promedio.

rmse: Tasa de error promedio del modelo(Hace referencia a todas las predicciones que ha realizado pero se ha equivocado porque la relación resultó ser distinta si el valor no superará 2 se podría considerar apto para un R2 del 0,8)

Fórmula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Donde:

- y_i son los valores reales (observados),
- $\hat{y}_i$ son los valores predichos por el modelo,
- n es el número total de observaciones.

mse: Tasa de error cuadrática del modelo(suele ser una exponente de 2 para la tasa de errores, lo que suele ser 4 veces mayor a rsme, si este valor no se incrementa de forma disparada entonces el modelo está apto. Hay casos en los que puede hasta ser menos que la tasa de error unitaria llegando a darnos cuenta de que realmente el modelo es capaz de encontrar falsos positivos y obviarlos para el conteo final de los datos y parámetros.)

Todos los valores cuentan con 2 campos ya que el primero es su valor base y el segundo es su oscilación ya sea positiva o negativa que no se podría revisar hasta que se probará y validará. Por eso se entrena el modelo y validar en el acto dejando así el segundo bloque que son los datos que se verían con el modelo en ejecución.

5.1.1-Validación cruzada

Acabamos de ver y explicar cómo se leería la validación de nuestro modelo ahora toca explicar cómo la realiza y otras opciones que podrían implementarse en su lugar.

KFold es utilizado para una validación cruzada más fuerte

Valoración de la validación cruzada:

El conjunto de datos disponible se divide en múltiples subconjuntos llamados **pliegues** o **folds**.

- Una parte de los datos se utiliza para entrenar el modelo (conjunto de entrenamiento), y otra parte se utiliza para evaluarlo (conjunto de validación), como se explicó anteriormente.

2. **Iteraciones del modelo**:

   - El proceso se repite varias veces, cada vez utilizando un pliegue diferente como conjunto de validación y los restantes como conjunto de entrenamiento. Logrando así encontrar todas las combinaciones posibles para la mejor interacción de los datos y unos resultados más precisos.

3. **Promedio de resultados**:

   - Los resultados obtenidos en cada iteración (como la precisión, el error cuadrático medio, etc.) se promedian para obtener una estimación general del rendimiento del modelo.

5.1.2-Tipos comunes de validación cruzada

1. **k-Fold Cross-Validation**:
    - Los datos se dividen en kkk pliegues de tamaño aproximadamente igual.
    - El modelo se entrena kkk veces, usando cada pliegue como conjunto de validación una vez y los k−1k-1k−1 restantes como conjunto de entrenamiento.
    - Es el método más común y versátil. Es el utilizado en el proyecto para la validacion del modelo.
2. **Leave-One-Out Cross-Validation (LOOCV)**:
    - Cada observación en el conjunto de datos se utiliza como conjunto de validación una vez, y el resto como conjunto de entrenamiento.
    - Es una variante extrema de k-Fold donde kkk es igual al número de observaciones.
3. **Stratified k-Fold**:
    - Similar a k-Fold, pero los pliegues se crean de forma que la distribución de clases en los pliegues sea similar a la del conjunto completo.
    - Útil para problemas de clasificación con clases desbalanceadas.
4. **Time Series Split**:
    - Diseñada para datos secuenciales, como series temporales.
    - Asegura que el conjunto de validación siempre esté en el "futuro" en comparación con el conjunto de entrenamiento para respetar la estructura temporal.

5.1.3-Ventajas de la validación cruzada

- **Mejor evaluación del rendimiento**: Permite estimar cómo se comportará el modelo con datos desconocidos.
- **Menor sesgo**: Todos los datos participan tanto en el entrenamiento como en la validación.
- **Evitar el sobreajuste**: Ayuda a identificar modelos que funcionan bien solo en el conjunto de entrenamiento.

5.1.4-Limitaciones

- **Costo computacional**: Requiere entrenar el modelo varias veces, lo cual puede ser costoso para modelos complejos o conjuntos de datos muy grandes.
- **Dependencia de los datos**: Si los datos no son representativos o tienen errores, las estimaciones pueden ser incorrectas.

En resumen, la validación cruzada es una herramienta fundamental para construir modelos más robustos y confiables.

5.2-Verificación del modelo

Al momento de entrenar al modelo se separan los datos para realizar las tareas de entrenamiento, validación y verificación del modelo. Lo que garantiza que en solo un paso se entrene, se valore los resultados aproximados y también se ven en resultados reales. Teniendo en cuenta la explicación del punto anterior vamos a ver qué valores se utilizan para la verificación.

5.2.1- Conjunto de prueba (test set)

Una parte del conjunto de datos original que se reserva exclusivamente para la verificación final del modelo.

Al dividir los datos originales, se separa un porcentaje (por ejemplo, 20%) como conjunto de pruebas. Este conjunto no se utiliza durante el entrenamiento ni la validación cruzada.

Lo que permite evaluar el modelo en datos que no ha visto antes, simulando su rendimiento en producción al mismo tiempo que genera el modelo.

5.2.2- Datos reales de producción

Son los datos recopilados del entorno real donde se implementará el modelo.

Se recopilan datos nuevos después de que el modelo ha sido entrenado y validado.
Estos datos reflejan las condiciones reales en las que el modelo operará.
Logrando proporcionar una evaluación más realista del rendimiento del modelo.

5.2.3- Datos históricos no utilizados

Estos datos históricos son los que no se incluyeron en el conjunto de entrenamiento ni en la validación.
Se seleccionan registros de un período de tiempo diferente al utilizado para el entrenamiento.
Lo que permite evaluar el modelo en datos que tienen patrones similares pero no idénticos.

6-Tipos de modelo

Diferencias clave entre los modelos

Conversión y Engagement tiene diferentes métricas y dimensiones, con la opción de todo se puede forzar a que el modelo recoja datos de ambas y los distribuya.

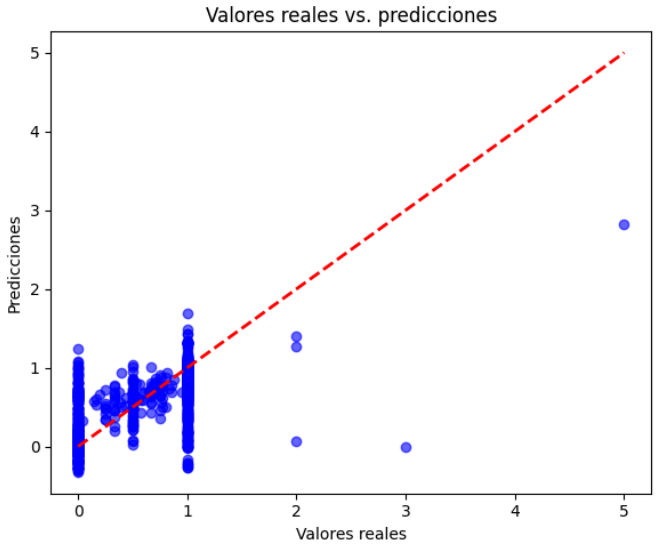
Nombre	Conversión	Engagement
Definicion	Acción especifica que un usuario realiza en un sitio web o aplicación, alineada con un objetivo(Comprar, suscribirse, etc)	Nivel de interacción, interés o participación de los usuarios en el contenido o plataforma que se está evaluando
Objetivo principal	Lograr que el usuario realice una acción beneficiosa para el negocio	Fomentar relaciones a largo plazo mediante interacciones regulares y significativas
Métricas Principales	-Tasa de conversion -Número de ventas -Descargas -Registros -Formularios completos	-Tasa de clics(CTR) -Likes, comentarios, compartido -Tiempo en la página -Frecuencia de interacción -Reacción en la redes sociales
Importancia estratégica	Crucial para medir el éxito de las campañas de los objetivos específicos del negocio.	Fundamental para construir relaciones sólidas en relación con la marca

Resumen de diferencias entre: Validación VS Verificación

Aspecto	Validación (validacion_modelo)	Verificación (verificacion_modelo)
Propósito	Evaluar el rendimiento del modelo	Comprobar la robustez del modelo
Datos utilizados	Datos de validación (parte del conjunto original)	Datos completamente nuevos o reales
Enfoque	Ajuste de hiper parámetros, selección de modelo	Pruebas finales antes de producción
Resultados esperados	Métricas de rendimiento (R^2, RMSE, etc.)	Confirmación de robustez y consistencia

6.1- Gráfico de Valores Reales vs. Predicciones

Este gráfico muestra una comparación entre los valores reales (eje X) y las predicciones del modelo (eje Y).
Una línea diagonal roja (línea de identidad) indica la referencia ideal donde las predicciones coinciden exactamente con los valores reales.



Qué buscar:

Distribución cercana a la línea de identidad: Si los puntos están cerca de la línea, significa que el modelo está haciendo buenas predicciones.

Desviaciones sistemáticas:
Si los puntos se desvían consistentemente hacia arriba o abajo, puede indicar un sesgo en el modelo y por ende no se está encontrando patrones idénticos que refuercen el aprendizaje sobre las columnas objetivo descritas.

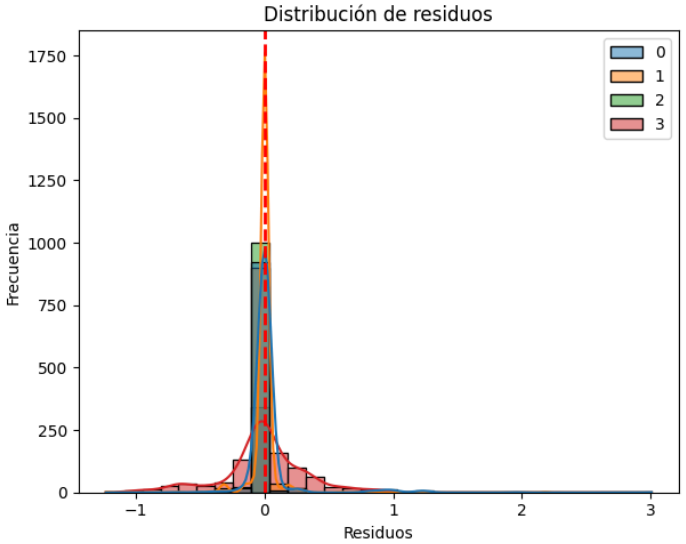
Patrones específicos:
Si hay patrones claros **como una curva**, puede ser señal de que el modelo no está capturando correctamente la relación entre las variables.

6.2- Gráfico de Distribución de Residuos

Descripción:

Este gráfico muestra la distribución de los residuos, que son las diferencias entre los valores reales y las predicciones **(residuo = valor_real - predicción)**.
Una línea vertical en 0 indica el punto donde los residuos son nulos (predicción perfecta).

Qué buscar:



Distribución centrada en 0:
Si la distribución está centrada en 0, significa que el modelo no tiene un sesgo sistemático.
Simetría:
Una distribución simétrica indica que los errores están distribuidos de manera uniforme.
Cola larga o asimetría:
Si hay colas largas o asimetría, puede ser señal de que el modelo tiene problemas para predecir ciertos valores.
Picos o multimodalidad:
Si hay múltiples picos, puede indicar que el modelo está funcionando de manera diferente para distintos subconjuntos de datos.
Ejemplo de interpretación:
Valores reales vs. predicciones:

Distribución de residuos:

Una distribución con forma de campana centrada en 0 es ideal.
Si los residuos tienen una cola larga hacia un lado, el modelo podría estar subestimando o sobreestimando ciertos valores.
Conclusión:
Ambos gráficos son herramientas complementarias:
El gráfico de valores reales vs. predicciones te ayuda a entender cómo de precisas son las predicciones del modelo.
El gráfico de distribución de residuos te ayuda a identificar patrones en los errores del modelo y posibles problemas de sesgo o varianza.

6.3-Matriz de correlación

Cálculo de la matriz de correlación:

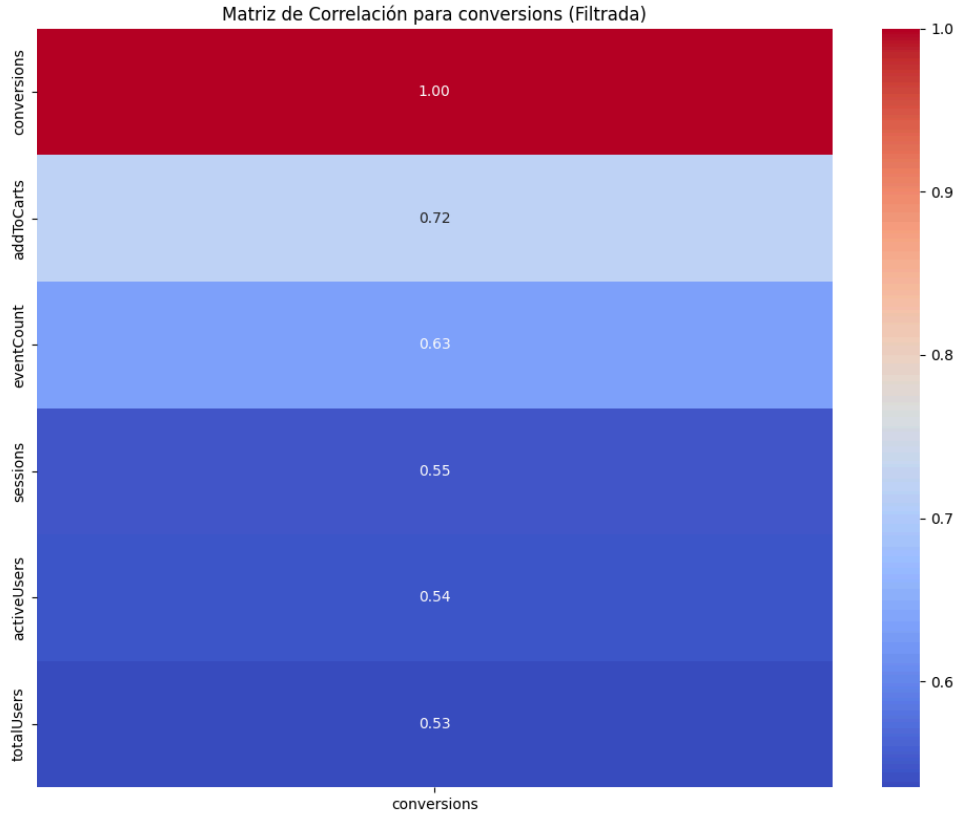
```
# Convertir a DataFrame para correlación
df_real_pred = pd.DataFrame(
    np.column_stack([np.array(y_val), np.array(y_pred)]),
    columns=y_val_cols + y_pred_cols
)
corr_matrix = df_real_pred.corr(method='pearson')
```

Visualización de la matriz de correlación:

Se utiliza sns.heatmap de Seaborn para generar un gráfico de calor.
Se anotan los valores de correlación en el gráfico.

```
# Graficar heatmap de la matriz de correlación (sin guardar CSV)
heatmap_path = os.path.join(directorio_salida, 'matriz_correlacion.png')
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Matriz de correlación (valores reales vs predicciones)')
plt.tight_layout()
plt.savefig(heatmap_path)
plt.close()
```

La matriz de correlación se guarda como matriz_correlacion.png en el directorio especificado (resultados por defecto).



Salida informativa:

Se imprime la ubicación del archivo generado para facilitar su localización.

Pandas busca correlación entre columnas numéricas por lo que hacer que las columnas tipo String o varchar pasen a ser codificación numérica supone un reto a tener en cuenta.

Codifica columnas categóricas:

Convierte las columnas categóricas a valores numéricos usando astype('category').cat.codes. Esto asigna un código numérico único a cada categoría.
Incluye columnas categóricas en la matriz de correlación:

Una vez convertidas en numéricas, las columnas categóricas pueden ser incluidas en el cálculo de la matriz de correlación.
Mantiene las columnas categóricas:

No elimina las columnas categóricas, sino que las transforma temporalmente para el análisis.

7-Problemas detectados y soluciones

7.1-Subajuste y Sobreajuste

7.1.1-Subajuste (Underfiting):

Los resultados de validación cruzada son muy pobres, mostrando un mal desempeño del modelo que no ha identificado patrones relevantes en los datos. Esto puede deberse a:

- Datos muy repetitivos que no contienen suficiente información para abordar problemas complejos.
- Uso de un modelo inadecuado para el problema planteado.
- Entrenamiento insuficiente del modelo, que limita su capacidad para capturar relaciones subyacentes.

El modelo presenta un bajo rendimiento tanto en pruebas como en entrenamiento, siendo incapaz de generalizar más allá de los datos proporcionados.

7.1.2-Sobreajuste (Overfiting):

Los resultados de validación cruzada son excelentes, pero el desempeño en el conjunto de validación es significativamente peor, lo que indica un sobreajuste a los datos de entrenamiento. Esto puede ser causado por:

- Características en los datos que no están bien relacionadas con la variable objetivo.
- Presencia de ruido en los datos, que interfiere en la capacidad del modelo para generalizar.
- Diferencias en la distribución de los datos de entrenamiento y validación.

Para mitigar el sobreajuste, se recomienda aplicar regularización ajustando los hiperparámetros del modelo.

7.2-Configuración del Pipeline

7.2.1-Importaciones necesarias:

- **GradientBoostingRegressor**: Algoritmo de regresión basado en árboles de decisión.
- **Pipeline**: Permite encadenar varios pasos de procesamiento.
- **StandardScaler**: Estandariza características (media = 0, desviación estándar = 1).
- **SimpleImputer**: Maneja valores faltantes.
- **MultiOutputRegressor**: Predice múltiples variables objetivo.

7.2.2-Estructura del Pipeline:

1. **SimpleImputer**:
    - Rellena valores faltantes con la mediana para garantizar datos completos.
2. **StandardScaler**:
    - Estandariza características para mejorar el rendimiento del modelo.

3. **MultiOutputRegressor** con **GradientBoostingRegressor**:

    - **n_estimators=300**: Número de árboles del modelo. Incrementar este valor puede mejorar el rendimiento, pero debe combinarse con otros parámetros para evitar sobreajuste.

    - **learning_rate=0.05**: Controla la contribución de cada árbol. Un valor más bajo mejora la estabilidad pero requiere más árboles.

    - **max_depth=4**: Limita la profundidad de los árboles, reduciendo la complejidad del modelo para evitar sobreajuste.

    - **min_samples_split=5**: Mínimo número de muestras requerido para dividir un nodo, lo que previene divisiones demasiado específicas.

    - **min_samples_leaf=3**: Mínimo número de muestras que debe tener una hoja, útil para evitar fluctuaciones causadas por datos atípicos o ruido.

    - **subsample=0.8**: Proporción de datos utilizada en cada árbol, ayudando a reducir el sobreajuste al introducir variaciones menores en los conjuntos de entrenamiento.

    - **random_state=42**: Asegura reproducibilidad al mantener constantes las particiones aleatorias. Aunque es ajustable, se recomienda usar un valor fijo durante pruebas y optimización.

    - **n_jobs=-1**: Utiliza todos los núcleos de CPU disponibles para acelerar el entrenamiento.

7.3-Modelos que soportan y no soportan entrenamiento incremental

Los modelos que no soportan partial_fit son aquellos que no están diseñados para entrenamiento incremental. Estos modelos requieren que todos los datos estén disponibles al mismo tiempo para entrenarse y no pueden actualizarse con nuevos datos sin volver a entrenarse desde cero. Algunos ejemplos comunes incluyen:

Modelos que no soportan partial_fit:

7.3.1-Modelos basados en árboles de decisión:

GradientBoostingRegressor / GradientBoostingClassifier
RandomForestRegressor / RandomForestClassifier
DecisionTreeRegressor / DecisionTreeClassifier
ExtraTreesRegressor / ExtraTreesClassifier
Estos modelos construyen árboles completos durante el entrenamiento y no pueden modificar los árboles existentes ni agregar nuevos datos sin reconstruir todo el modelo.

Modelos de ensamble:

AdaBoost
BaggingClassifier / BaggingRegressor
VotingClassifier / VotingRegressor
Estos modelos combinan múltiples estimadores entrenados previamente, lo que hace que no puedan actualizarse de manera incremental.

Modelos de redes neuronales en scikit-learn:

MLPClassifier / MLPRegressor (Multi-Layer Perceptron)
Aunque las redes neuronales pueden entrenarse de manera incremental en otras bibliotecas como TensorFlow o PyTorch, las implementaciones de scikit-learn no soportan partial_fit.

Modelos de optimización global:

SVC / SVR (Support Vector Machines)
KernelRidge
GaussianProcessRegressor / GaussianProcessClassifier
Estos modelos suelen requerir cálculos globales sobre todo el conjunto de datos, lo que los hace incompatibles con el entrenamiento incremental.

7.3.2-Modelos que sí soportan partial_fit:

Modelos lineales:

SGDClassifier / SGDRegressor (Stochastic Gradient Descent)
PassiveAggressiveClassifier / PassiveAggressiveRegressor
Perceptron
Ridge (con solver='sag' o solver='saga')
Modelos Naive Bayes:

MultinomialNB
BernoulliNB
GaussianNB
Clustering:

7.4-Problemas y soluciones

7.4.1-¿Qué hacer si tu modelo no soporta partial_fit?

Si necesitas entrenamiento incremental pero tu modelo no lo soporta, puedes considerar:

1. Cambiar a un modelo compatible con partial_fit.
2. Entrenar desde cero con los datos combinados (aunque esto puede ser costoso en términos de tiempo y recursos).
3. Usar bibliotecas especializadas como TensorFlow, PyTorch o XGBoost, que ofrecen soporte para entrenamiento incremental en ciertos casos.

7.4.2-GridSearchCV

Implementar **GridSearchCV** para optimizar los hiperparámetros puede ayudar a mejorar la tasa de predicción y reducir los errores. Esto se debe a que **GridSearchCV** explora exhaustivamente todas las combinaciones de hiperparámetros en el espacio definido, lo que permite encontrar la configuración óptima para el modelo.
Beneficios:
Mejor ajuste del modelo:

Al probar todas las combinaciones posibles, es más probable que el modelo se ajuste mejor a los datos, lo que puede mejorar la precisión de las predicciones.
Reducción de errores:

Encontrar los hiperparámetros óptimos puede reducir el error de predicción (como el MSE o RMSE), ya que el modelo estará mejor configurado para capturar las relaciones en los datos.
Mayor estabilidad:

Un modelo con hiperparámetros optimizados tiende a ser más robusto y generalizar mejor en datos no vistos.

Consideraciones:
Tiempo de ejecución: El uso de **GridSearchCV** puede ser computacionalmente costoso, especialmente si el espacio de búsqueda es grande.

Datos de calidad: La mejora en la predicción también depende de la calidad de los datos. Es importante que los datos estén bien preprocesados y representen correctamente el problema.

8-Hiperparametrizacion y opciones a usar

8.1-¿Qué es la hiperparametrización?

La hiperparametrización es el proceso de ajustar los hiperparámetros de un modelo de aprendizaje automático para mejorar su rendimiento. Los hiperparámetros son configuraciones que no se aprenden directamente del conjunto de datos durante el entrenamiento, sino que se establecen antes de entrenar el modelo. Ejemplos de hiperparámetros incluyen:

La profundidad máxima de un árbol de decisión (max_depth).
La tasa de aprendizaje (learning_rate) en modelos como GradientBoostingRegressor.
El número de estimadores (n_estimators) en un modelo de ensamble.
Parámetros de regularización como alpha y l1_ratio en ElasticNet.
El objetivo de la hiperparametrización es encontrar la combinación de valores que maximice el rendimiento del modelo en un conjunto de validación, evitando tanto el sobreajuste como el subajuste.

8.2-Comparativa de Metodologías antiguas

Ahora vamos a explicar metodologías antiguas que se descartaron por su dificultad de funcionamiento y aprendizaje pero que se tuvieron en cuenta durante la creación del proyecto.

¿Qué es mejor: --random-search o --optimizar?
Ambas opciones (--random-search y --optimizar) son métodos para realizar la hiperparametrización, pero tienen diferencias clave:

1. --random-search (RandomizedSearchCV):
Cómo funciona: Busca los mejores hiperparámetros seleccionando aleatoriamente combinaciones de valores dentro de un rango definido. Se evalúa un número limitado de combinaciones (n_iter).

Ventajas:
Es más rápido que GridSearchCV cuando hay muchos hiperparámetros o rangos amplios.
Puede encontrar buenas combinaciones sin probar todas las posibilidades.
Desventajas:
No garantiza que se prueben todas las combinaciones posibles.
Puede ser menos exhaustivo si el número de iteraciones (n_iter) es bajo.
2. --optimizar (GridSearchCV):
Cómo funciona: Busca los mejores hiperparámetros probando todas las combinaciones posibles dentro de un rango definido.
Ventajas:
Es exhaustivo y garantiza que se prueben todas las combinaciones.
Es útil cuando el espacio de búsqueda es pequeño y bien definido.

Desventajas:
Es mucho más lento que RandomizedSearchCV si hay muchos hiperparámetros o rangos amplios.
Puede ser ineficiente si algunas combinaciones no son relevantes.
¿Cuál es mejor?
Depende del contexto:

Usa --random-search si:

Tienes un espacio de búsqueda grande (muchos hiperparámetros o rangos amplios).
Quieres optimizar el tiempo de búsqueda.
No necesitas probar todas las combinaciones posibles.
Usa --optimizar si:

Tienes un espacio de búsqueda pequeño y bien definido.
Quieres asegurarte de probar todas las combinaciones posibles.
No te importa que el proceso sea más lento.
Recomendación general:
En la mayoría de los casos, --random-search es más eficiente, especialmente si no tienes una idea clara de los valores óptimos de los hiperparámetros. Sin embargo, si ya tienes un rango reducido y quieres ser exhaustivo, --optimizar (**GridSearchCV**) puede ser más adecuado.

8.3-Comparativa Random Search VS Deep Research

Seguimos explicando metodologías utilizadas y retiradas del proyecto porque los datos obtenidos no eran lo que se esperaba. Si es verdad que Deep Research es muy potente y consigue que la tasa de error sea mínima superando rara vez el 1,5 en los parámetros de error. Su tasa de acierto es muy escasa llegando como máximo a 0,3 lo que lo convierte en un método muy complejo de utilizar por gente inexperta.

Vamos a ver en más profundidad las 2 metodologías para comprender sus puntos de vista y funcionamiento.

8.3.1- ¿Qué es Random Search?

Random Search explora aleatoriamente combinaciones de hiperparámetros dentro de un espacio definido. Es simple y efectivo, especialmente cuando solo algunas combinaciones específicas producen buenos resultados.

Ventajas de Random Search

1. **Eficiencia en espacios grandes**:

- Es más eficiente que Grid Search en espacios de hiperparámetros grandes, ya que no evalúa todas las combinaciones posibles, solo una muestra aleatoria.

2. **Foco en hiperparámetros importantes**:
    - Estudios han demostrado que solo unos pocos hiperparámetros suelen influir significativamente en el desempeño del modelo. Random Search tiene más probabilidades de encontrar buenos valores para esos hiperparámetros importantes, ya que no gasta recursos evaluando combinaciones exhaustivas de todos.
3. **Fácil de implementar**:
    - Es más rápido y sencillo que enfoques más avanzados como Bayesian Optimization o Deep Research.
4. **Paralelización**:
    - Es fácil de paralelizar, ya que las evaluaciones de las combinaciones son independientes entre sí.

8.3.2- ¿Qué es Deep Research?

El modo **Deep Research** es un enfoque más sistemático e iterativo que combina análisis detallados, inteligencia contextual y aprendizaje para encontrar los mejores hiperparámetros.

Ventajas de Deep Research

1. **Exploración más profunda**:
    - No se limita a una muestra aleatoria; utiliza heurísticas y análisis detallados para explorar regiones prometedoras del espacio de hiperparámetros.
    - Puede aplicar técnicas como:
        - **Bayesian Optimization**: Modela las relaciones entre los hiperparámetros y la métrica objetivo para explorar áreas con mayor probabilidad de mejora.
        - **Algoritmos genéticos**: Evoluciona combinaciones de hiperparámetros basándose en resultados anteriores.

2. **Adaptabilidad dinámica**:
    - Aprende de iteraciones previas para ajustar el enfoque en tiempo real, priorizando áreas más prometedoras del espacio.
3. **Mayor precisión**:
    - Es más probable que encuentre la combinación óptima en problemas complejos o con interacciones no lineales entre hiperparámetros.
4. **Aprovechamiento de datos históricos**:
    - Puede incorporar resultados de experimentos previos o conocimiento del dominio para guiar la búsqueda.
5. **Optimización contextual**:
    - Profundiza en cómo ciertos hiperparámetros afectan el modelo y ajusta dinámicamente las estrategias en función de los resultados.

8.3.3-Comparativa: Random Search vs Deep Research

Aspecto	Random Search	Deep Research
Facilidad de Implementación	Muy fácil (bibliotecas como `scikit-learn`)	Más compleja (requiere frameworks avanzados como `Optuna` o `Ray Tune`)
Eficiencia en espacios grandes	Buena, pero no sistemática	Excelente, especialmente en problemas no triviales
Exploración en profundidad	Limitada a la aleatoriedad	Muy profunda, adaptándose al espacio
Adaptabilidad	No adaptativa	Dinámica y basada en resultados previos
Recursos computacionales	Menos intensivo	Más intensivo
Aplicación óptima	Ideal para problemas simples o recursos limitados	Ideal para problemas complejos o recursos abundantes

8.3.4-¿Cuándo Usar Cada Enfoque?

- **Usa Random Search si**:
    - Tienes restricciones de tiempo o recursos computacionales.
    - Tu modelo tiene relativamente pocos hiperparámetros importantes o el espacio de búsqueda no es muy grande.
    - Necesitas una solución rápida y "suficientemente buena".
- **Usa Deep Research si**:
    - El modelo es complejo y depende de múltiples hiperparámetros interrelacionados.
    - Tienes acceso a más recursos computacionales.
    - Buscas maximizar el rendimiento del modelo en problemas críticos.
    - Deseas una búsqueda más inteligente y sistemática.

9-Odoo

El apartado de Odoo va a consistir en adaptar los archivos antiguos del proyecto para tener una beta visual de la creación de un modelo predictivo, ya sea para el emprendimiento autodidacta del usuario o un muestreo express para convencer al cliente de comprar el servicio/producto.

9.1-Equipo necesario

1. Vamos a tener Odoo en un docker compose el cual albergará un contenedor con base de datos y otro con odoo.

   Comenzamos realizando el comando:

   odoo scaffold ai_train_model

   Con esto se generará la estructura básica del proyecto

2. Los archivos con los que hemos trabajado con anterioridad

3. GitHub con el que haremos el control de versiones

9.2-Funcionalidad

9.2.1-Comprobar credenciales

Primero tenemos el archivo que se encarga de la gestión de las credenciales. Realiza el almacenamiento de forma segura y comprueba la validación para corroborar que son válidas, después de todo el proceso muestra los ID de las tablas y los property.

Con la vista el usuario es capaz de interactuar de manera gráfica con las credenciales incluso pudiendo llegar a tener varias si así dispone de archivos de credenciales. Pudiendo llegarse a ver diferentes credenciales aplicadas lo que sirve también de sistema de almacenaje local.

Por último todo es realizado con el archivo de verificación de permisos de fondo que se mencionó con anterioridad.(verificación)

9.2.2-Descarga e importación de datos

Comenzamos explicando el funcionamiento del modelo. Gestiona la descarga de datos como en el archivo de descargas a que está conectado internamente del cual recoge algunos métodos. El problema recae en que Odoo corta la conexión al momento de la descarga por tiempo y tamaño de datos lo que resulta imposible la descargarla incluso al mínimo. Para solventar el problema se habilitó un wizard en el cual se puede subir archivos. Con este parche temporal se puede solventar el problema de la descarga siendo una importación.

Como la intención es sólo didáctica/mostrativa se puede arreglar a posterior, este apartado se quedará para cosas que faltaron o no se llegaron a realizar.

9.2.3-Sistema de entrenamiento

Este apartado se divide en 2 puntos que son el MV y los útiles que se han utilizado con anterioridad.

A diferencia de la explicación anterior, aquí el error no nos importa tampoco el rate de predicción, solo queremos que se le muestre al cliente el funcionamiento de los formatos sencillos de cómo se entrena el modelo y cómo debe realizarse la lectura de los datos obtenidos.

Al terminar el entrenamiento se ejecutará un controller que nos mostrará un informe sobre el resultado del modelo y la posibilidad de descargarlo con todos los archivos que hemos visto con anterioridad.

9.2.4-Integración y seguridad

Existe un archivo csv que es el que proporciona la seguridad al módulo aunque en un principio está totalmente habilitado se podría entregar a los clientes con algunas restricciones.

Por parte de la navegación existe una vista encargada de dicha función mostrando un pequeño manu con el cual nos podremos desplazar por los 3 puntos necesarios para el trabajo.

10-Funcionamiento de Odoo y el modelo predictivo

Odoo
Modelo predictivo

11-Conclusiones

11.1-Resultados Obtenidos

Hemos llegado a obtener un modelo predictivo funcional que nos ayuda a poder detectar clientes que no tienen intención de comprar en la página web a la que la api esté conectada. Además de tener un módulo de Odoo el cual es capaz de predecir el porcentaje de posibilidades de que se tramite un lead en el CRM y otro módulo encargado de generar los modelos predictivos.

11.2-Puntos Pendientes

No se han quedado puntos pendientes en el proyecto como tal de manera de que no se haya completado una parte, pero sí que es verdad que siempre se puede mejorar el formato de entrenamiento y el de mejora de los modelos.

En cambio en la parte de Odoo si que se necesitaría mejor arreglar la descarga de datos.

11.3-Tiempo dedicado

El tiempo dedicado ha sido desorbitado, se le empleó el tiempo del proyecto y también el de las prácticas para poder llegar a terminarlo, el problema fue que no se sabía si se iba a llegar a tiempo y por ende la duda de si se llegara a tener proyecto para la empresa.

11.4-Valoración personal

La verdad es que el alumno trabajó de manera ardua para poder aprender todos los conceptos necesarios en un tiempo muy corto en comparación al que realmente se necesita por lo que tampoco se han consolidado los conceptos de manera firme pero si se han entendido sus funcionamientos básicos para saber cual es el desempeño y funcionalidades de las tecnologías con las que trabaja el proyecto.

11.5-Conexión con las prácticas

El proyecto tiene conexión directa con las prácticas ya que es la empresa la que me ha prestado el código para realizar el proyecto. En inprofit utilizaran el proyecto para su uso privado y el comercial de sus clientes.

11.6-Consejos destacables

El proyecto podría haber tardado menos si ser más sencillo de comprender si se hubiera limitado a unas tecnologías en específico pero en su día se probaron tantas de manera exponencial que el modelo llegaba a entrenarse de muchas maneras distintas y eso era contraproducente puesto que los datos entrenaba al modelo de tantas maneras que no servia, era ineficiente a si que se redujo a una versión en la que se entrena a un formato normal y uno incremental si los datos son demasiado grandes.

12-Necesidades y sugerimientos de formación

En realidad el machine learning es bastante importante porque se necesita muchísimo cuando se trabaja con modelos automáticos de predicciones y computaciones de grandes dimensiones. Pero a la vez es muy complicado poder impartirlo, por lo que a mi punto de vista prioriza la creación de API y un mejor entendimiento por encima de otra cosa de mi proyecto y para la mejora de los alumnos de DAM. Puesto que he sufrido mucho en entender cómo funcionaban las API por no haber dedicado tiempo suficiente durante el curso y aun sigo sin ser capaz de entender correctamente cómo es que realizan las cosas en su totalidad completa y me han tenido que ayudar para poder llegar.

13-Bibliografía

Habilitar la lectura de Scripts: Enlace
Curso de funcionamiento de BigQuery: Enlace
Video explicativo funcional de cómo trabajar con BigQuery: Enlace
Curso de aprendizaje de las bases de python y entrenamiento de modelos: Enlace
SGDRegressor: Enlace
Cross_val_score: Enlace
Kfold:Enlace
Pipeline:Enlace
GradientBoostingRegressor: Enlace
StandardScale: Enlace
SimpleImputer: Enlace
MultiOutputRegressor: Enlace
matplotlib: Enlace
Numpy: Enlace
seaborn: Enlace
Sklearn.metrics: Enlace
clone: Enlace
Bayesiana: Enlace
partial_fit: Enlace
tempfile: Enlace
Subprocess: Enlace
Base64: Enlace
Chardet: Enlace

14-Recursos utilizados

1. Portatil personal
2. GA4 de la empresa
3. Google Cloud de la empresa
4. Entorno virtual
5. WordPress de la empresa
6. Página web de pruebas de la empresa
7. Docker Compose para tener el Odoo en local
8. Odoo

15-Diccionario

Hiperparametro: Parámetro utilizado para controlar el proceso de aprendizaje
CO: Columna Objetivo
GA4: Google Analytics 4
PW: Pagina web
MV: Modelo vista