KEEPCODING

DEBT ANALYTICS

TU SOCIO EN PREDICCIÓN DE PAGOS

Nuestra aplicación analiza el comportamiento financiero de los clientes, prediciendo la probabilidad de cumplimiento de sus pagos. Utilizando algoritmos avanzados y análisis de datos, ofrecemos predicciones precisas, ayudando a minimizar riesgos y maximizar oportunidades de negocio.

TFB KC Debt Analytics

XII Bootcamp BD, AI & ML 2023-2024

EQUIPO:

Ignacio Vicente

Javier García

Carolina Segado

Laura Aparicio

https://kcdebtanalytics.wordpress.com/

KEEPCODING
Debt Analytics

MEMORIA

Tabla de contenido

MEMORIA

MEMORIA

Contexto

KC Debt Analytics es una empresa que se dedica a la compra de carteras de deuda crediticia de bancos, caracterizadas por una alta variabilidad en la tipología de préstamos, generalmente en situaciones de estrés financiero. El objetivo principal es recobrar la mayor cantidad posible de dichas carteras. Para ello, la empresa dispone de una variedad de herramientas destinadas a maximizar la recuperabilidad de los préstamos.

Las carteras se reparten entre los gestores de negocio, quienes tienen autonomía para alcanzar acuerdos que permitan maximizar la recuperabilidad de los préstamos adquiridos. Estos acuerdos pueden incluir promesas de pago, nuevos calendarios de pago fraccionado y, en algunos casos, la posibilidad de una quita tras el cumplimiento del acuerdo.

Necesidades por cubrir

KC Debt Analytics busca implementar tecnologías de inteligencia artificial (IA) para optimizar la gestión de recobros. Con el objetivo de mejorar la productividad de los gestores, se propone construir una suite completa de herramientas que permitan el seguimiento de los acuerdos con simulaciones de probabilidades de recobro. Esto ayudará a optimizar la gestión de las deudas, minimizando riesgos y maximizando oportunidades de negocio, además de hacer previsiones más precisas del cumplimiento de objetivos.

Steackholders del proyecto:

- **Gestores:** Mejorar el seguimiento y la gestión de los acuerdos de pago para la consecución de sus objetivos.
- **Managers**: Posibilidad de realizar estimaciones basadas en modelos de predicción más precisos y fiables, y elaborar estrategias de gestión efectivas.
- **Departamento de inversiones:** Mejorar los análisis de comportamiento de posibles nuevas carteras a adquirir y desarrollar un plan de negocio optimizado gracias a estas previsiones de cumplimiento.

Hitos

Para este proyecto, se establecen los siguientes hitos:

- Creación de un sistema de base de datos en la nube simulando la arquitectura real de una empresa donde se recoja la información de todas las fuentes utilizadas.
- Sistema automatizado para recibir y almacenar información externa relevante para el modelo predictivo.
- Dashboard visual en PowerBI para analizar las carteras.
- Desarrollo de componentes visuales personalizados e integrados en PowerBI a través de de D3.
- Desarrollo de una demo en forma de web donde realizar simulacro de propuestas a clientes y evaluar la probabilidad de cumplimiento.

MEMORIA

Fase I: Definición del Dataset

Inicialmente se decide utilizar un dataset privado de una compañía real en lugar de uno público. Adicionalmente, a este conjunto de datos se añaden fuentes externas que proporcionan una visión completa de las operaciones crediticias y relaciones comerciales de la empresa.

- **Datos internos**: BBDD de deudas, de deudores, de procesos judiciales y/o concursales e histórico de acuerdos de pago.
- **Datos públicos**: Series históricas de defaults y datos de renta por código postal.

Se han importado ficheros anonimizados de una empresa real, y se ha creado la arquitectura MySQL en Google Cloud simulando la estructura de BBDD real de la empresa.

Una vez montada, se realizaron en Colab las consultas necesarias para obtener el dataset final con los datos requeridos de diversas fuentes. Este dataset se guardó en una tabla de hechos en Google Cloud Platform (GCP) y en un archivo .csv en Google Cloud Storage.

Finalmente, dichas queries se expresaron en forma de función en Python alojada dentro del servicio Cloud Functions que se ejecutaba mensualmente a través de un Cloud Scheduler.

Google function para bajar y transformar datos de Banco de España (es la function-2):

	Entorno	Nombre	Última implementación	Región	Recomendación	Activador	Tiempo de ejecución	Memoria asignada	Función ejecutada
☐	2nd gen	function-1	17 may 2024 13:47:29	us-central1		Bucket: keep_impagados	Python 3.12	256 MiB	hello_gcs
☐	2nd gen	function-2	8 may 2024 22:06:20	europe-west1		HTTP	Python 3.12	256 MB	hello_http
☐	2nd gen	Subidaficheros	17 may 2024 13:44:08	europe-west1		Bucket: keep_impagados	Python 3.12	256 MiB	hello_gcs
☐	2nd gen	subidaformsv3	17 may 2024 18:10:49	us-central1		HTTP	Python 3.12	256 MB	upload_spreadsheet
☐	2nd gen	Updated_CSV	17 may 2024 13:54:32	europe-west1		HTTP	Python 3.12	256 MB	hello_gcs
☐	2nd gen	uploadingformsv2	17 may 2024 17:55:21	europe-west1		–	Node.js 20	–	upload_spreadsheet

Y abajo, imagen del scheduler:

	Nombre	Estado de la última ejecución	Región	Estado	Descripción	Frecuencia	Destino
☐	lanzar_series_BDE	Aún no se ha ejecutado	europe-west1	Habilitado		0 9 5 * * (Europe/Madrid)	URL : https://europe-west1-sonic-airfoil-421707.cloudfunctions.net/function-2

01 preparación data: https://colab.research.google.com/drive/1vFwXIEXwddp44ihWXR-PyfXzeNztQdRV?usp=sharing

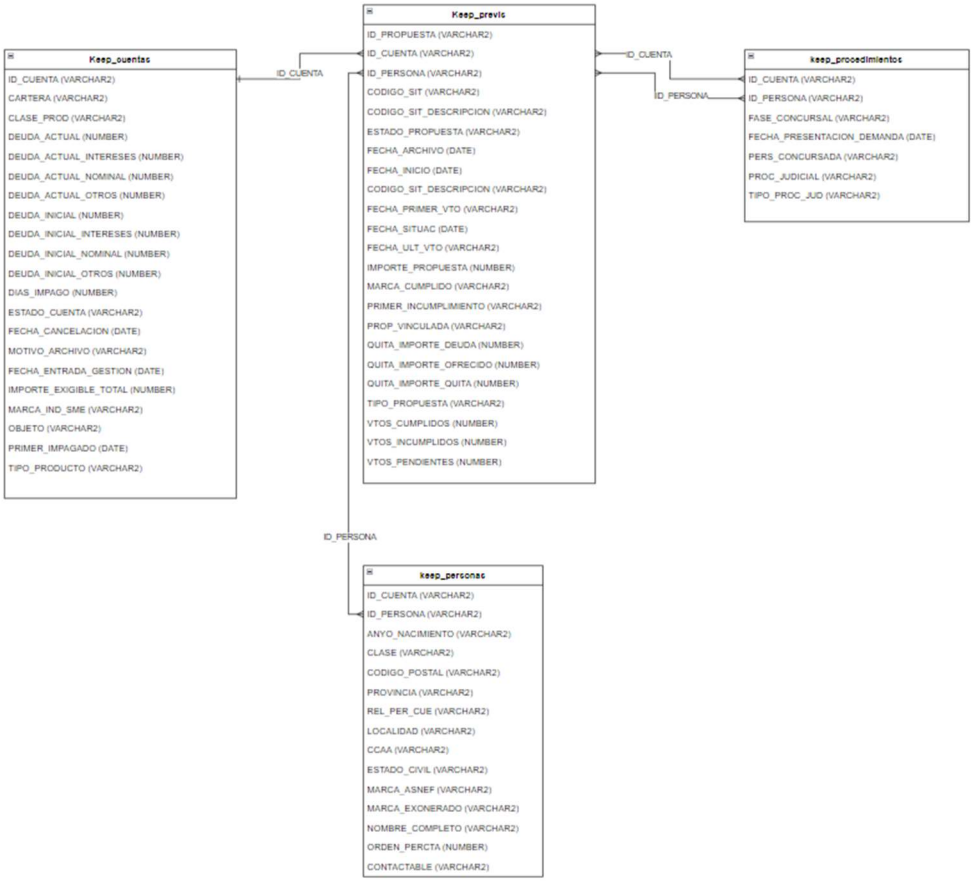
MEMORIA

A continuación, desglosamos ambas fuentes:

Fuentes de datos internas

Las fuentes de datos internas se componen de:

- **Histórico de operaciones crediticias**: Esta sección abarca el registro detallado de todas las transacciones financieras llevadas a cabo por la empresa, junto con sus atributos y características relevantes. *(Tabla: Keep_cuentas).*
- **Listado de clientes:** Incluye información sobre personas físicas y jurídicas con las que la empresa ha interactuado en el pasado, proporcionando una visión detallada de sus perfiles y relaciones comerciales. *(Tabla: Keep_personas).*
- **Registros judiciales y concursales:** Esta fuente de datos aborda los procedimientos judiciales y concursales relacionados con operaciones y entidades previamente mencionadas, tanto en curso como cerrados. *(Tabla: Keep_procedimientos).*
- **Propuestas acordadas**: Aquí se registra el histórico de las promesas y acuerdos de pago acordados entre la empresa y sus clientes, indicando el estado de estas (vigentes, cumplidas, incumplidas, etc.). *(Tabla: Keep_previs).*

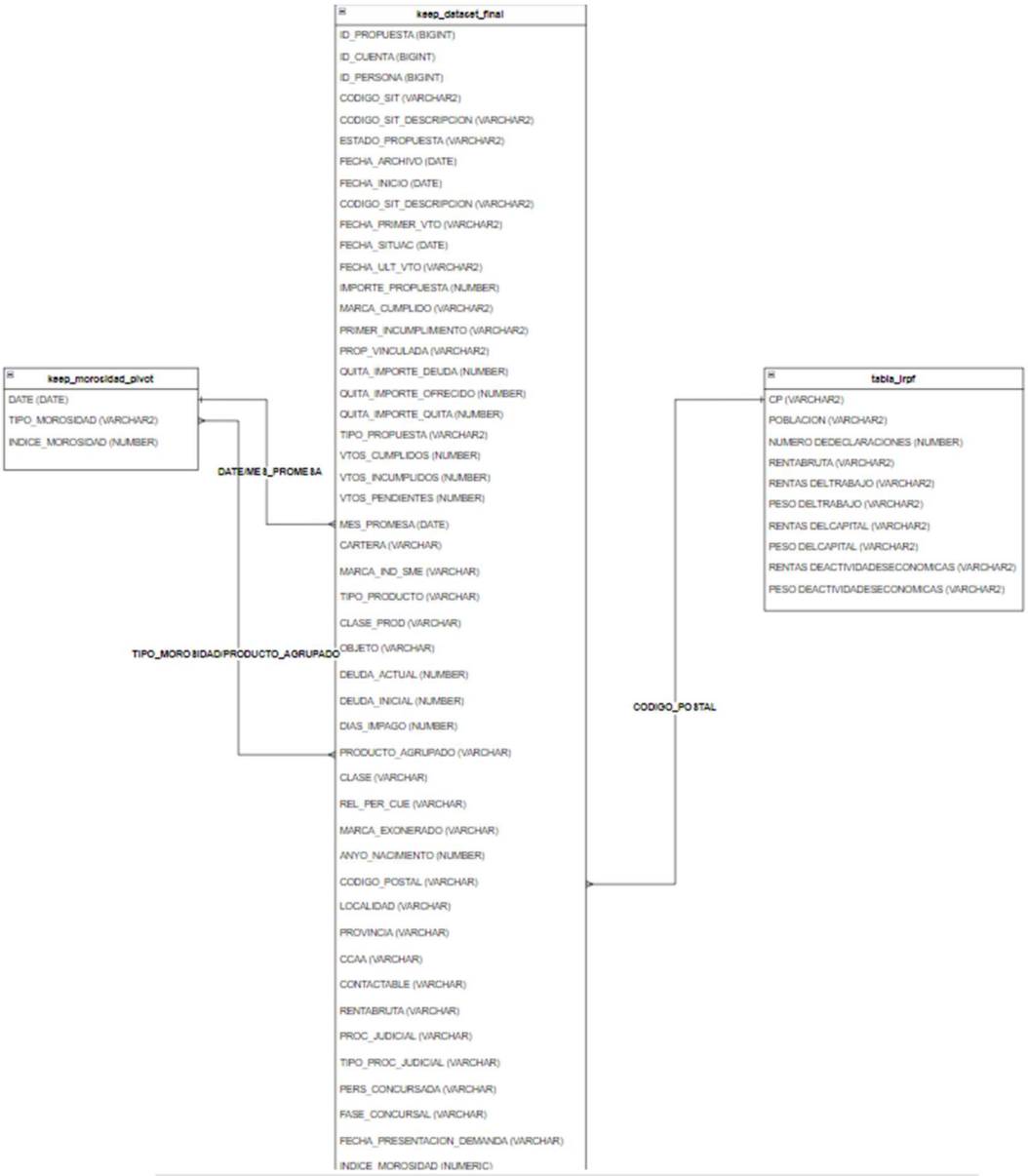


MEMORIA

Fuentes de datos externas

Además de las fuentes internas mencionadas, el conjunto de datos se complementa con dos fuentes públicas que ofrecen información adicional que nos podría ayudar a calcular la probabilidad de recobro de las operaciones:

- **Serie histórica de defaults**: Procedente del Banco de España, esta fuente proporciona datos históricos sobre defaults en España para distintos tipos de crédito, enriqueciendo el análisis con información relevante del mercado.
- **Datos de renta por código postal**: Obtenidos de la Agencia Tributaria Estadística de los declarantes del IRPF de los mayores municipios por código postal: 2021: Composición Renta Bruta Tramos de Base Imponible: Total (agenciatributaria.gob.es), estos datos ofrecen información detallada sobre la renta según declaraciones de IRPF, permitiendo analizar la situación económica a nivel geográfico y mejorar la predicción de cumplimiento.



Fase II: Arquitectura y validación de datos

La arquitectura se fundamenta en la suite de Google Cloud Platform y Streamlit Community, creando un entorno empresarial simulado que comprende:

Dominio específico y correos electrónicos

Se estableció un dominio dedicado para la empresa *(@kcdebtanalytics.net)*, junto con direcciones de correo electrónico personalizadas para cada uno de los cuatro empleados, todas vinculadas a dicho dominio.

Google Cloud Platform (GCP)

- **Buckets de Google Storage**: Utilizados para almacenar los datos en formato .csv sin relación.
- **Google Functions**: Implementadas como funciones serverless para la descarga de datos del Banco de España e IRPF y la transformación de tablas a formato .csv, facilitando su almacenamiento y manipulación.
- **Triggers autoejecutables**: Programados según calendario para la descarga, transformación y almacenamiento automático de datos públicos.
- **Google Cloud SQL MySQL**: Se emplea como base de datos relacional para almacenar datos estructurados.
- **PowerBI**: Enlazado con Cloud SQL a través de una puerta de enlace para visualizar los datos.
- **Página web corporativa en WordPress**: Alojada en un hosting de WordPress y enlazada a una API desplegada en GCP para la captura de nuevos datos, facilitando la interacción con los usuarios.

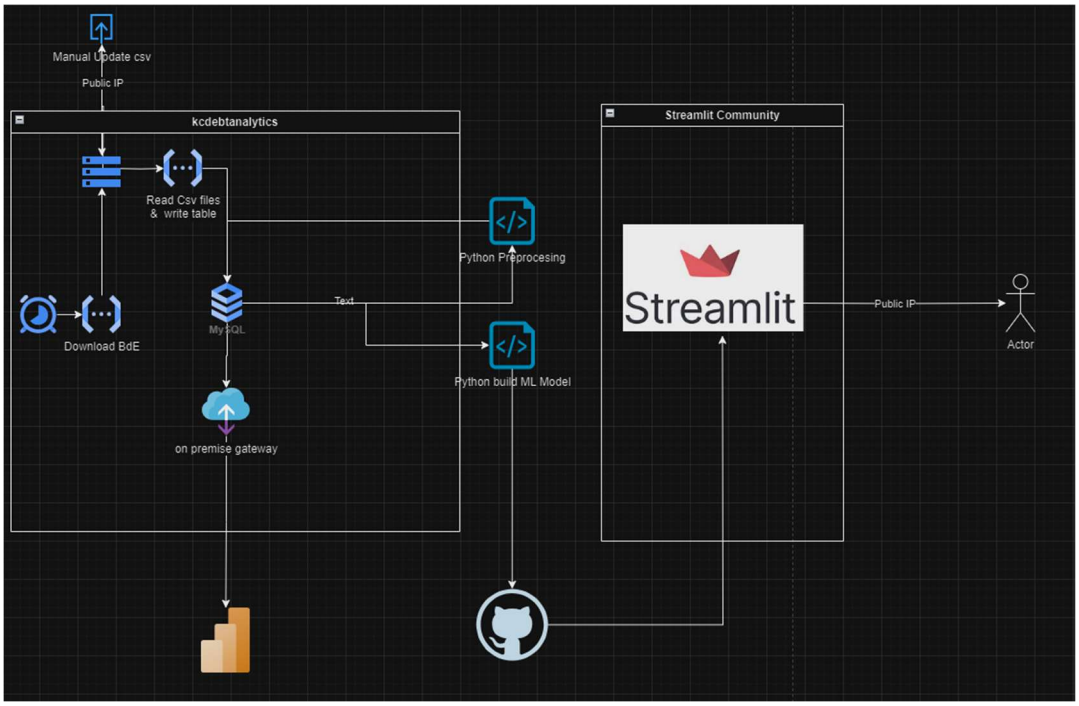
MEMORIA

Streamlit:

Para el despliegue real de la aplicación de predicción, se ha construido a través de Streamlit. Para ello ha sido necesario:

- **Github**: Repositorio de código de la aplicación de Streamlit.
- **Streamlit Community**: alojamiento de la aplicación de predicción.

A continuación, se representa la arquitectura a través de un esquema de todo el proceso desarrollado:

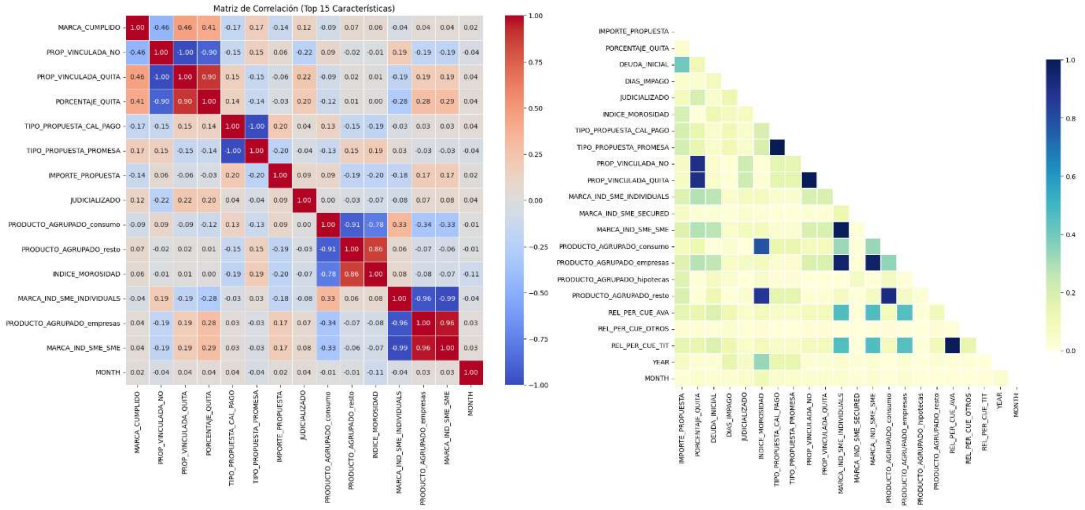


MEMORIA

Fase III: Análisis exploratorio

Durante esta fase, se realizaron varias tareas para profundizar en el análisis del dataset. Inicialmente, se revisaron las variables para detectar redundancias y se evaluaron las correlaciones existentes entre ellas. Esto condujo a la identificación de la necesidad de desarrollar nuevas variables que potencien el desempeño del modelo. Decidimos imputar valores a los campos vacíos en algunos casos utilizando la información disponible de otras variables y en otras los valores oportunos para esa variable, y determinamos que ciertas variables con una alta cantidad de datos faltantes, por ejemplo, aquellas relacionadas con fases concursales, deberían eliminarse para mantener la claridad del modelo.

Dado que muchas de las variables eran categóricas, se realizó una cuidadosa selección para evitar sobrecargar el modelo con demasiada complejidad. También se evaluaron las correlaciones entre las variables para garantizar que el modelo funcionara correctamente sin problemas de ejecución.



Utilizando PowerBI sobre la base de datos SQL desarrollada, se genera un dashboard que proporciona una visión detallada de las principales métricas de la cartera.

Recurso: PowerBI

El análisis exploratorio realizado a través de PowerBI incluye los siguientes elementos:

Para el desarrollo de los componentes personalizados, debido a las limitaciones que presentaba PowerBi, se ha decidido desarrollar 2 componentes de burbujas en D3 que muestren de forma visual las composiciones de la cartera.

Repositorio de componentes: https://github.com/JGMFC/Burbujas

Utilizando PowerBI sobre la base de datos SQL desarrollada, se genera un dashboard que proporciona una visión detallada de las principales métricas de la cartera. El análisis exploratorio realizado a través de PowerBI incluye los siguientes elementos:

MEMORIA

Tamaño de las carteras y KPI esencial

Gráficos de barras horizontales que muestra la composición de las carteras de crédito, identificadas por su nombre comercial, en sus dos variables más relevantes: (1) el % de cumplimiento de las propuestas de reorganización de deuda a nivel de importes y (2) el tamaño inicial de las carteras (en términos de importe inicial antes de la propuesta)

KPIs esenciales

A la izquierda se muestran los KPIs esenciales en tarjetas numéricas (1) el número de carteras, (2) el total de los activos gestionados con propuestas crediticias (3) el número de propuestas presentadas a contrapartidas (4) el importe agregado de las propuestas formuladas (5) el % de quita (aminoración de deuda) sobre el importe propuesto y (6) el % de cumplimiento de las propuestas de reorganización de deuda a nivel de importes. Estos KPIs se expresan tanto para el conjunto de la compañía como específicamente para cada una de las carteras, ya que se filtran tras la selección de estas.

Nube de propuestas vinculadas

Con el propósito de presentar la información en el PowerBI con una incrustación de gráficos construidos mediante D3 e incluidos como librerías de React, se presenta un gráfico de nubes dinámico donde se presentan el conjunto de operaciones crediticias por tamaño (representado en forma de burbuja) y con indicación de si viene acompañado de una propuesta de quita (representado en función del color).

Nube de segmentación

De igual forma, se ha construido mediante gráficos en D3, nubes que representan el tamaño de cada comunidad autónoma en el conjunto de la cartera creditica y colores que representan el tipo de crédito.

Evoluciones de tasas de impago en España

Con el fin de que sirva para la construcción de benchmarks contra los que comparar, en la parte central se refleja información histórica de las tasas de defaults en España para cada tipología de crédito.

Evolución del “performance” de la empresa

En la parte inferior se refleja un gráfico que señala el nivel de actividad comercial de la empresa. Dicho gráfico, refleja cuántas propuestas se formulan, cuántas quitas se ofrecen (ambos en forma de barras) combinado con un gráfico de línea donde se aprecia el % de cumplimiento en las propuestas que plantea la empresa. Toda la información aparece calendarizada por trimestre.

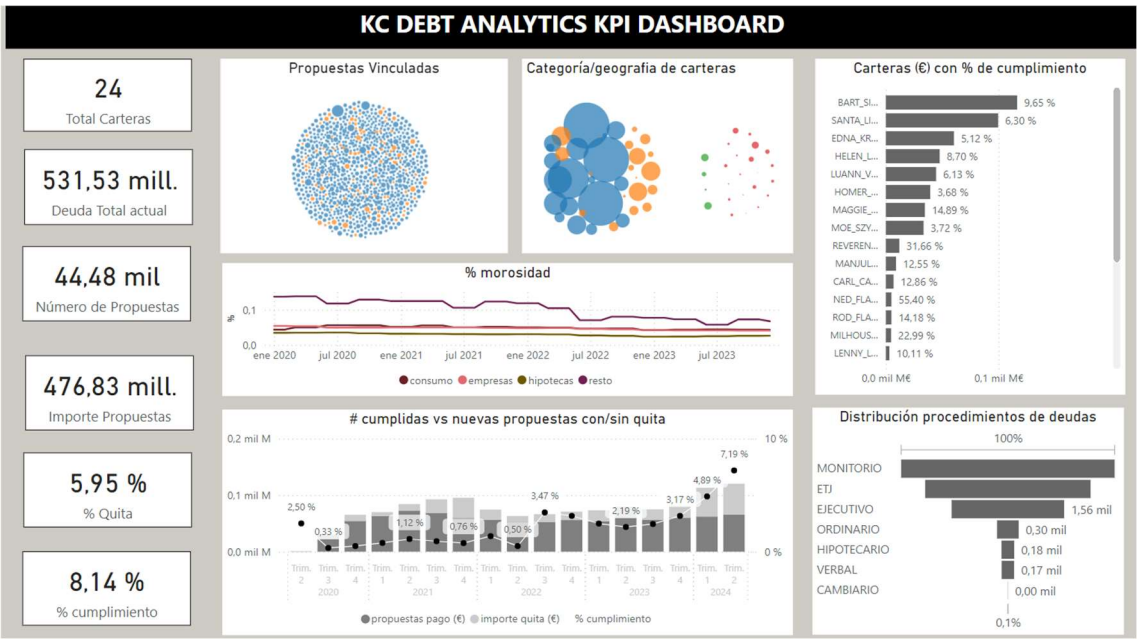
Gráficos de barras y líneas con el “performance” de la empresa

En la parte inferior se refleja un gráfico que señala el nivel de actividad comercial de la empresa. Dicho gráfico, refleja cuántas propuestas se formulan, cuántas quitas se ofrecen (ambos en forma de barras) combinado con un gráfico de línea donde se aprecia el % de cumplimiento en las propuestas que plantea la empresa. Toda la información aparece calendarizada por trimestre.

MEMORIA

Distribución del procedimiento de las deudas

En la parte inferior izquierda se refleja un gráfico tipo funnel en el que se refleja de forma jerarquizada: (a) el número de procedimientos judiciales vs no judiciales y (b) dichos procedimientos de qué tipología son.



Fase IV: Preprocesado

02 preprocesado:
https://colab.research.google.com/drive/1gfcdYGsWPS4Ka9JqIIjNPS6OEyzMeuVd?usp=sharing

El preprocesado de datos es una etapa crucial en el proyecto, compuesta por múltiples fases destinadas a garantizar la calidad y la utilidad de los datos para su posterior análisis y modelado. Las principales acciones realizadas durante esta fase incluyen:

Transformación de datos con identificadores únicos:

Se aplicó una transformación para asignar un identificador único a cada operación y cliente de manera concatenada, permitiendo una identificación clara y unificada de los registros en el conjunto de datos.

Depuración de outliers

Se llevó a cabo una exhaustiva depuración de valores atípicos que pudieran distorsionar el análisis y el modelado posterior.

Limpieza de valores vacíos

Se realizaron operaciones de limpieza para eliminar datos faltantes o NaN que podrían afectar la calidad de los análisis.

Determinación de umbrales de validez y eliminación de entradas erróneas

Se establecieron criterios de validez y se eliminaron entradas erróneas o inconsistentes, con un enfoque especial en la validación de códigos postales.

Creación de nuevas variables

Se crearon nuevas variables necesarias para el modelo, como por ejemplo la división de la fecha en mes y año, el cálculo de edad de los clientes, índice de los diferentes registros, una marca de judicializado en el momento de realizar el acuerdo, etc.

Conversión en variables binarias

Se realizaron conversiones adicionales para transformar variables relevantes en variables binarias, lo que facilita su manejo y su interpretación en los modelos predictivos.

Categorización mediante "One Hot Encoding"

Las variables categóricas se convirtieron en variables dummy utilizando la técnica de "One Hot Encoding".

Reducción de dimensionalidad

Se llevó a cabo una reducción de dimensionalidad mediante la eliminación de variables altamente correlacionadas o redundantes.

Fase V: Modelado

03 Primer entrenamiento ML:
https://colab.research.google.com/drive/1gfcdYGsWPS4Ka9JqIIjNPS6OEyzMeuVd?usp=sharing

04 Entrenamiento ML con selección características:
https://colab.research.google.com/drive/1tshYUHS11M6rRFMRDNYG9FzyHeWUzrwg?usp=sharing

05 Entrenamiento modelos con SMOTE y selección características:
https://colab.research.google.com/drive/1ihXiQawI3ghlnvl1kW_lwHuiqQYDi-mz?usp=sharing

06 Modelo final RF con SMOTE:
https://colab.research.google.com/drive/1SvKFBEueIXAwtbkUzOOFgSBntwobjSsi?usp=sharing

Se abordó la fase de modelado con dos enfoques distintos, con el objetivo de seleccionar el modelo que mejor se ajustara a los datos y proporcionara una explicación sólida de la variable dependiente binaria, que predice la probabilidad de cumplimiento de una propuesta de recobro a un prestatario.

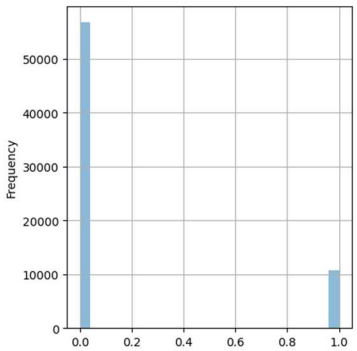
Modelos de Machine Learning/Regresión Logística

Considerando la naturaleza del proyecto, se optó por explorar modelos de *machine learning*. Se identificó que un modelo de *regresión logística* sería adecuado para este problema, ya que permite predecir la probabilidad de que el prestatario cumpla con la propuesta presentada. Se espera que este enfoque proporcione una explicación clara e interpretable de los factores que influyen en el cumplimiento de las propuestas de recobro.

Se entrenaron los siguientes modelos:

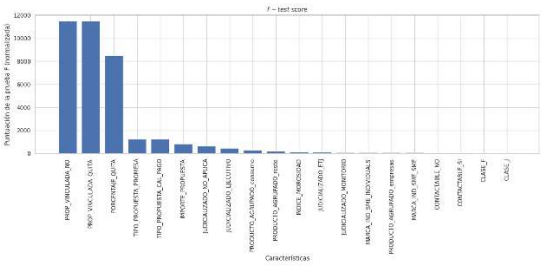
- *Regresión logística*
- *Árbol de decisión*
- *Random Forest Classifier*

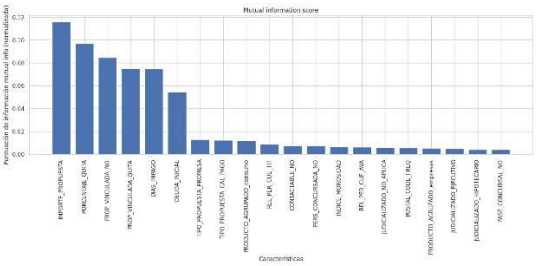
Vimos durante el análisis exploratorio que el dataset estaba muy desbalanceado, no sólo la variable objetivo sino también algunas de las variables.



MEMORIA

Se hizo selección de características en función de las más relevantes vistas en los modelos y se volvieron a entrenar.



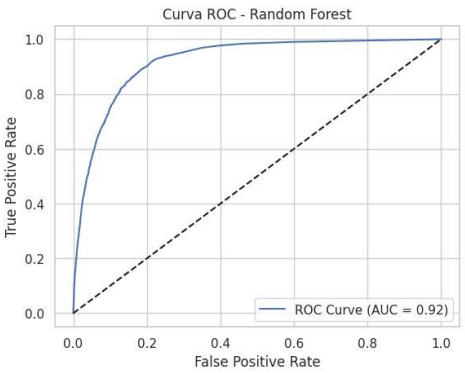


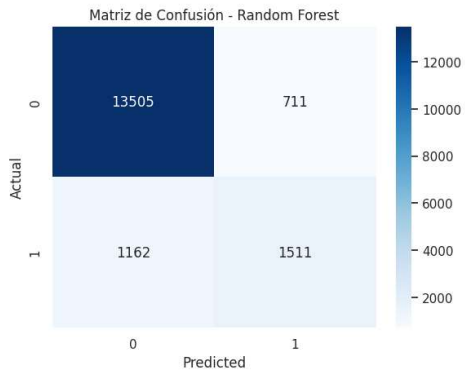
Debido al desbalanceo de las variables en el dataset, además de usar *stratify* en la división train/test, también entrenamos *Regresión Logística* y *Random Forest* aplicando la función *SMOTE* a los datos de entrenamiento, que crea datos sintéticos para las variables desbalanceadas aumentando la frecuencia de las características con menos concurrencia.

Random Forest sin SMOTE

	precision	recall	f1-score	support
0	0.92	0.95	0.94	14216
1	0.68	0.57	0.62	2673
accuracy			0.89	16889
macro avg	0.80	0.76	0.78	16889
weighted avg	0.88	0.89	0.88	16889

AUC: 0.9211169380501275

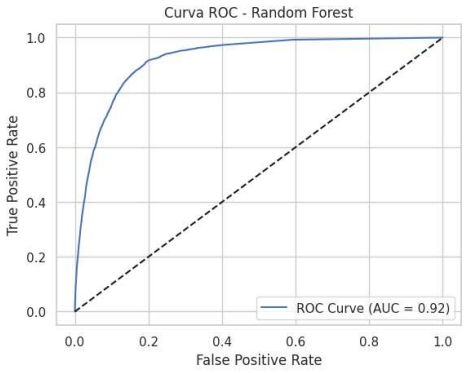


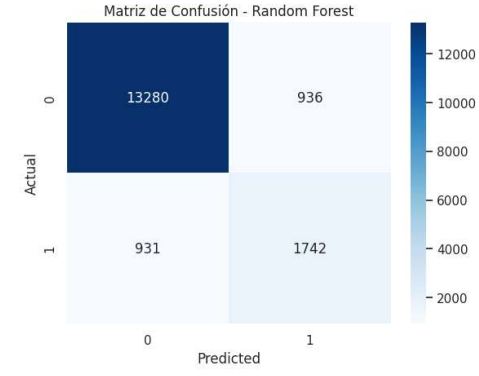


Random Forest con SMOTE

	precision	recall	f1-score	support
0	0.93	0.93	0.93	14216
1	0.65	0.65	0.65	2673
accuracy			0.89	16889
macro avg	0.79	0.79	0.79	16889
weighted avg	0.89	0.89	0.89	16889

AUC: 0.9231050632210515





MEMORIA

Modelos de Deep Learning

03.1Primer entrenamiento con DeepLearning

https://colab.research.google.com/drive/1VZ2pzTWplDy3Xt_lK-vU9VXFkBalCEgj?usp=sharing

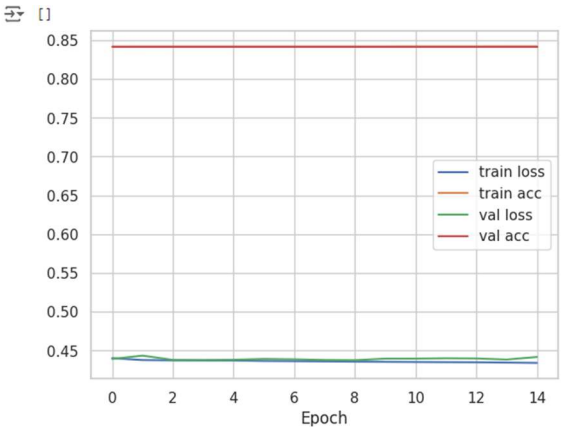
Además del enfoque *de machine learning* tradicional, se exploraron dos modelos de *deep learning* muy similares. Un primero de clasificación y un segundo de regresión. En ambos casos se implementó un modelo secuencial sencillo de redes neuronales de tres capas, con una preselección de características explicativas, una primera capa de 64 neuronas, una segunda capa de 32 neuronas, una tercera capa de 8 neuronas y con una última capa de tipo sigmoide, que inicialmente mostró promesa para el problema en cuestión.

```
Input size: 70
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_20 (Dense)            (None, 64)                4544

 dense_21 (Dense)            (None, 32)                2080

 dense_22 (Dense)            (None, 8)                 264

 dense_23 (Dense)            (None, 2)                 18

=================================================================
Total params: 6906 (26.98 KB)
Trainable params: 6906 (26.98 KB)
Non-trainable params: 0 (0.00 Byte)
```

También se probaron modelos de regresión. No obstante, el resultado se descartó por resultar más difícil de interpretar frente a un modelo de clasificación ordinario para la naturaleza del problema que era fundamentalmente binaria (cumple o no cumple)

El resultado del modelo, en términos de precisión tanto en test como el validación fue del 84%, no cualitativamente diferente de los modelos de machine learning.

Al final, se descartó la utilización de modelos de deeplearning porque la explicación de la relevancia de cada característica resultaba más intuitiva con los modelos de machine learning y porque la relación de variables explicativas y volumen de datos empleados parecía que encajaba mejor con un modelo de machine learning.



Decisión final

Tras la realización de pruebas comparativas, se decidió retirar el modelo de *deep learning* debido a su menor capacidad explicativa y a las limitaciones encontradas con algunas variables dependientes. En su lugar, se optó por el modelo de *Random Forest* de clasificación con SMOTE, que demostró ser más adecuado para el problema y proporcionar una explicación más clara de los resultados.

Este modelo ofreció una mejor curva ROC y mostró una precisión similar a otros modelos evaluados. Sin embargo, el árbol de decisión presentaba un ligero overfitting, lo cual también ocurría al aumentar el rango del parámetro *'max_depth'* en la regresión logística utilizando GridSearch. Por otro lado, la matriz de confusión del Random Forest mostró menos errores, lo que destacó su eficacia en la clasificación correcta de las observaciones.

Conclusiones y lecciones aprendidas

Suposiciones Iniciales (erróneas)

Se esperaba que una base de datos robusta, combinando datos públicos y privados fuera la piedra angular del modelo predictivo, sin embargo, se encontraron limitaciones significativas que mermaron la capacidad predictiva de los modelos:

- La falta de datos privados de los individuos más allá de su participación en procesos concursales, ya que dicha información era confidencial y no podía ponerse a disposición de una parte del equipo.
- La necesidad de anonimizar los datos personales para cumplir con regulaciones como GDPR, lo que limitaba el uso de técnicas como NLP, ya que no se podía predecir a partir del nombre de la persona ni combinar con información en las redes sociales.
- La anonimización dificultaba el acceso a fuentes externas de información, como registros de impagados o datos financieros de las compañías u otros indicadores indirectos de la capacidad de pago.

Se intentaron compensar estas limitaciones con datos de renta promedio por código postal a partir de la Base de Datos del INE. Pero no funcionó porque el INE no publica datos de renta para códigos postales de menos de 1.000 individuos. Por lo tanto, el empleo de la renta promedio obligaba a limitar sustancialmente la BBDD original, por lo que finalmente fue una variable descartada.

Durante el análisis y entrenamiento de los modelos

Al realizar los primeros análisis exploratorios:

- Vimos que nuestra variable objetivo que era la marca de cumplimiento de la promesa (cumplida o incumplida) estaba muy desbalanceada, cosa que no esperábamos. Es por este motivo que probamos *SMOTE* y lo que vimos es que la precisión bajaba un poco respecto a los modelos sin *SMOTE*. Sin embargo, si vemos las matrices de confusión y el recall, los modelos predicen mucho mejor el cumplimiento, baja un poco la precisión en los incumplimientos, pero la mejora en la predicción de la clase minoritaria lo compensa.
- Teníamos muchas variables categóricas y esto nos podía dificultar mucho a la hora de hacer los encoders, eliminamos todas las variables redundantes y simplificamos algunas como, por ejemplo, las de REL_PER_CUE (tipo de relación de la persona con la cuenta) y agrupamos las categorías minoritarias en un 'OTROS'.
- Vimos muchos errores en los datos por ejemplo en las fechas de nacimiento y otros datos que se introducen manualmente en el sistema.

Desarrollo web y API:

Proyecto Final del Bootcamp: Integración de Tecnologías para una Solución Predictiva Dinámica

En nuestro proyecto final del bootcamp, hemos elegido una solución integral que fusiona la visualización gráfica con la interacción directa a través de una plataforma web y una API. Optamos por estas tecnologías para ofrecer una experiencia de usuario accesible y dinámica, adecuada para usuarios técnicos y no técnicos por igual.

Desarrollo Web y Visualización de Datos

La interfaz principal del proyecto es una página web diseñada en WordPress, que nos permite gestionar y presentar la información de forma eficiente. Hemos enriquecido la web con visualizaciones interactivas que simplifican la comprensión del modelo de negocio y facilitan la navegación a través de la información presentada. Estas visualizaciones no solo ilustran los datos, sino que también enganchan al usuario al permitirle explorar diferentes aspectos del análisis.

Interacción con el Modelo Predictivo

Paralelamente, desarrollamos una API usando Streamlit, que está integrada en la página web. Esta API brinda a los usuarios la posibilidad de interactuar directamente con nuestro modelo predictivo. En una demo interactiva, los usuarios pueden introducir datos específicos y obtener predicciones en tiempo real. Esta funcionalidad no solo muestra la aplicabilidad práctica de nuestro modelo, sino que también subraya su capacidad para generar resultados precisos a partir de los datos introducidos por el usuario.

Detalle Streamlit: app-imjnrygri297w4gpblrwpm.streamlit.app

Repositorio: **GitHub - JGMFC/Streamlit**

Automatización y Actualización de Datos

Una vez que el proyecto se implemente en producción, el sistema se configurará para actualizar automáticamente los datos esenciales, como información sobre deudas, tipos de productos y otros parámetros críticos. Esta actualización continua alimenta el modelo predictivo, asegurando que las predicciones y análisis siempre estén basados en la información más reciente. Esta automatización no solo mejora la fiabilidad, sino también la precisión de los resultados, ofreciendo *insights* valiosos y oportunos a los usuarios finales.

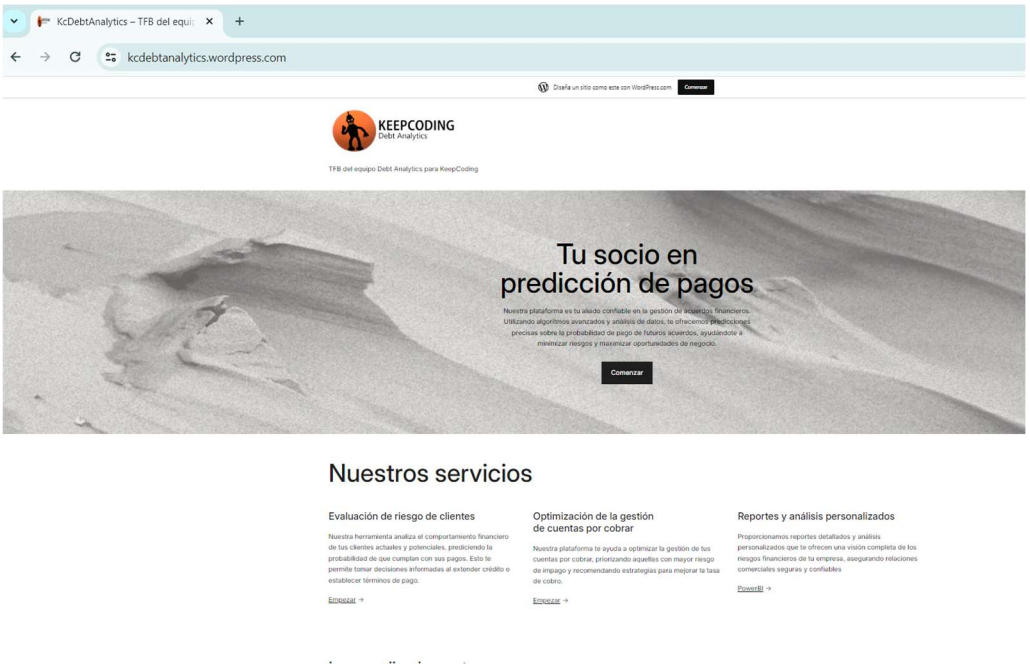
Implementación de la API:

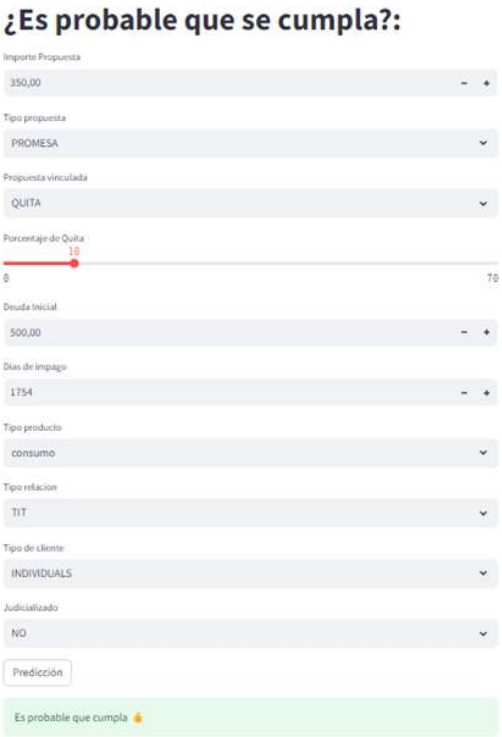
Para la implementación de la API, se utilizaron los siguientes archivos:

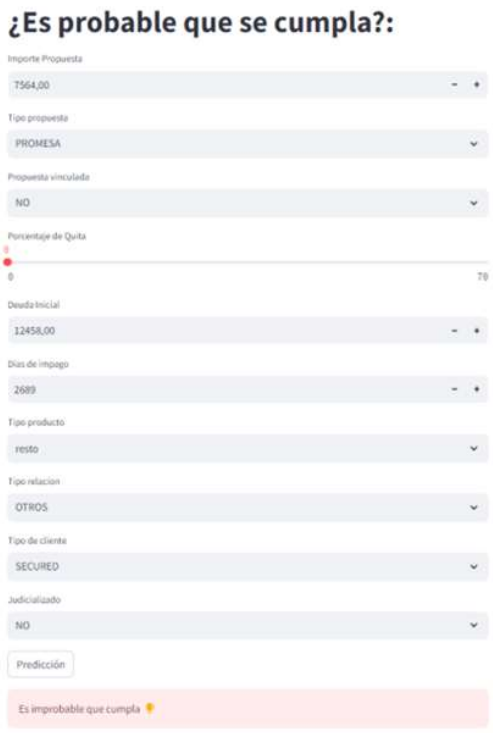
- **app.py:** Archivo principal de Streamlit que contiene la lógica de la interfaz de usuario y la integración con el modelo.
- **utils.py:** Contiene funciones auxiliares, incluyendo la función de procesamiento de datos necesarios para preparar la entrada del modelo.
- **Model_smote_rf. joblib:** Archivo que almacena el modelo de machine learning (Random Forest) entrenado y guardado en formato .joblib.

MEMORIA

- **Requirements.txt**: Especifica todas las dependencias y bibliotecas necesarias para ejecutar el proyecto, asegurando la compatibilidad y reproducibilidad del entorno.
- **Dockerfile**: Define cómo se debe construir la imagen del contenedor, facilitando la implementación del proyecto en cualquier entorno compatible con Docker.







Problemas encontrados:

Integración de Power BI con Google Cloud Platform:

La elección de una arquitectura basada en Google demostró ser acertada en términos de resolución de problemas de incompatibilidad. Surgieron dificultades al intentar integrar Google Forms y Power BI con Google Cloud Platform (GCP). Power BI, una herramienta nativa de Azure, presentó problemas de compatibilidad y adaptación a la arquitectura de GCP. Esto requirió una reconfiguración y ajustes adicionales para lograr una integración fluida entre las plataformas.

La decisión de utilizar PowerBI como herramienta de visualización fue inicialmente beneficiosa debido a la familiaridad del equipo, pero se encontraron desafíos de compatibilidad con la arquitectura elegida que finalmente se pudieron solventar.

Incompatibilidad de versiones de scikit-learn al cargar el modelo desde la API:

El script de la API llamaba a un modelo guardado en formato .joblib. Sin embargo, la versión de scikit-learn utilizada en Google Colab resultó ser incompatible cuando se intentó montar la API localmente usando Visual Studio Code (VSC). Para solucionar este problema, fue necesario rehacer el archivo .joblib utilizando otra versión más actual de scikit-learn en VSC, asegurando la compatibilidad entre el entorno de desarrollo y el de producción.

Problemas con permisos para usuarios en Google Cloud Platform:

Durante la configuración de GCP, surgieron problemas relacionados con los permisos de los usuarios. Algunos usuarios no podían acceder a ciertos recursos necesarios para el desarrollo y despliegue del proyecto. Esto fue debido a configuraciones inadecuadas en las políticas de identidad y acceso (IAM) de GCP. Para resolver este problema, se revisaron y ajustaron los roles y permisos asignados a cada usuario, asegurando que todos tuvieran los accesos necesarios sin comprometer la seguridad del sistema. Se crearon roles personalizados cuando fue necesario para cumplir con los requisitos específicos del proyecto.

Problemas de despliegue en Google Cloud Platform:

Inicialmente, se utilizó una combinación de Streamlit y Flask, donde Flask era llamado por una petición POST de la API desarrollada en Streamlit. Sin embargo, el servidor Flask denegaba constantemente el acceso, después de innumerables intentos, crear nuevas reglas de fireware, habilitar diferentes puertos, etc. Lo que causó una pérdida significativa de tiempo. Finalmente, se encontró una solución utilizando únicamente Streamlit, alojado en GitHub, lo que simplificó el despliegue y solucionó los problemas de acceso.

Lecciones aprendidas

Se pueden extraer las siguientes lecciones del proyecto:

- Ha sido fundamental realizar reuniones diarias, a menudo, sólo para revisar el estado del proyecto. Estas reuniones han resultado ser muy productivas y han garantizado el cumplimiento de los plazos establecidos para cada tarea, así como la aportación de distintos enfoques por parte de todos los miembros.
- De la misma forma, nos fue de mucha utilidad tener un espacio de trabajo en Trello donde ir poder ir trabajando de forma conjunta anotando cualquier cambio y adjuntando todo tipo de archivos y enlaces.
- La importancia de dimensionar adecuadamente los proyectos y considerar la complejidad técnica de la infraestructura desde el principio.
- La necesidad de construir sistemas homogéneos y compatibles, eligiendo cuidadosamente la arquitectura que mejor se adapte a las necesidades del proyecto.
- Reconocimiento de las limitaciones impuestas por la anonimización de datos en el contexto del análisis de big data y machine learning.

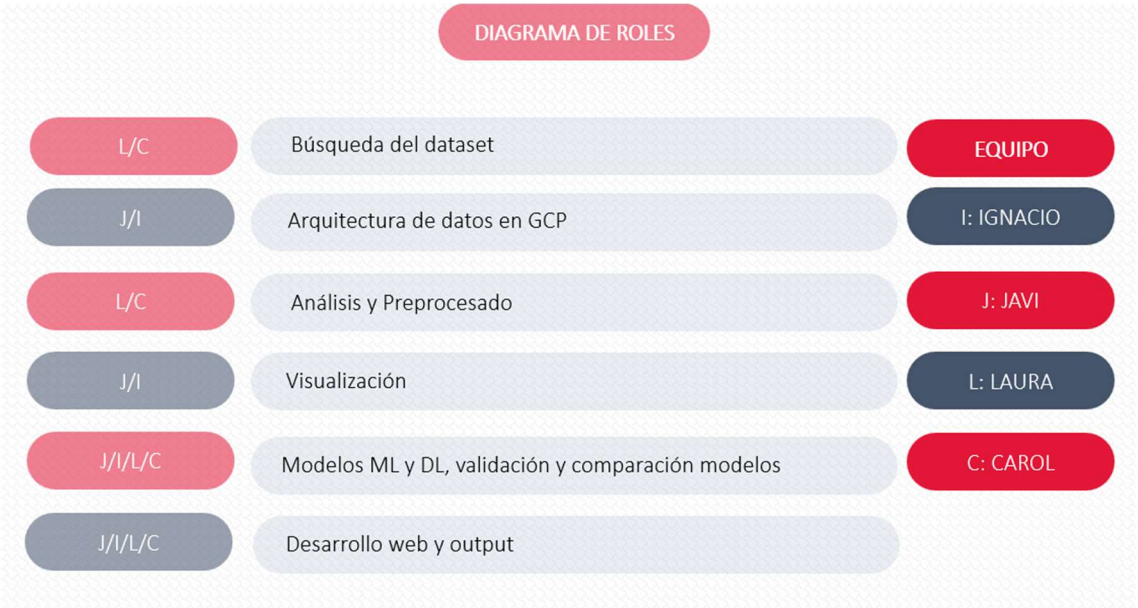
En resumen, el equipo logró implementar una estructura profesional para la ingestión, análisis y modelado de datos en un tiempo récord. Sin embargo, se identificaron posibles vías de mejora de cara a futuro:

- Ingestas adicionales de datos sobre las propuestas, como firmas físicas de los clientes, registro conversaciones telefónicas, emails intercambiados con el cliente.
- Empleo de modelos predictivos sobre los nuevos datos, como un modelo de NLP sobre las conversaciones telefónicas con el cliente, o sobre escritos/emails/solicitudes o un modelo de *deep learning* sobre el estilo de firma de documentos de la contrapartida.
- Migrar, una vez estabilizado el dashboard en PowerBI en soluciones visualmente más customizables de desarrollo web, como D3, para mejorar la experiencia del usuario de la suite de gestión y predicción de promesas sobre préstamos.

MEMORIA

Distribución de tareas en el equipo

Diagrama de roles



Cronograma

MEMORIA

Recursos

- **Drawio**: Esquemas Fuentes Drawio
- **Trello:** https://trello.com/b/TOVHEMBd/cuadro-de-mando
- **Datos de renta por código postal:** Estadística de los declarantes del IRPF de los mayores municipios por código postal: 2021: Composición Renta Bruta Tramos de Base Imponible: Total (agenciatributaria.gob.es)
- **Github para Streamlit:** https://github.com/JGMFC/Streamlit
- **Web:** https://kcdebtanalytics.wordpress.com/
- **API:** https://app-imjnrygri297w4gpblrwpm.streamlit.app/
- **Github para D3:** https://github.com/JGMFC/Burbujas
- **PowerBI:** https://app.powerbi.com/reportEmbed?reportId=51f9fcda-4e0f-4c3e-b432-5849fb9fd361&autoAuth=true&ctid=47556885-bfe6-4fef-bd77-2c9f940656ac
- **Colabs Utilizados:**

    - 01 Preparación data: https://colab.research.google.com/drive/1vFwXIEXwddp44ihWXR-PyfXzeNztQdRV?usp=sharing
    - 02 Análisis prepocesado: https://colab.research.google.com/drive/1gfcdYGsWPS4Ka9JqIIjNPS6OEyzMeuVd?usp=sharing
    - 03 Primer entrenamiento ML: https://colab.research.google.com/drive/1gfcdYGsWPS4Ka9JqIIjNPS6OEyzMeuVd?usp=sharing
    - 03.1 Primer entrenamiento con DeepLearning: https://colab.research.google.com/drive/1VZ2pzTWplDy3Xt_lK-vU9VXFkBalCEgj?usp=sharing
    - 04 Entrenamiento ML con selección características: https://colab.research.google.com/drive/1tshYUHS11M6rRFMRDNYG9FzyHeWUzrwg?usp=sharing
    - 05 Entrenamiento modelos con SMOTE y selección características: https://colab.research.google.com/drive/1ihXiQawI3ghlnvl1kW_lwHuiqQYDi-mz?usp=sharing
    - 06 Modelo final RF con SMOTE: https://colab.research.google.com/drive/1SvKFBEueIXAwtbkUzOOFgSBntwobjSsi?usp=sharing