

**MBIT SCHOOL**

**Trabajo de Fin de Máster**

Médicos sin Fronteras, predicción de aumento de cuota de socios

**AUTORAS y AUTORES:** Marta Leonor Rodríguez Viejobueno, Miriam Macías Domínguez, Roberto Carballares Navarro, Marcial Lalanda González-Bueno

**TUTOR**: Javier Cózar del Olmo

Máster en Ciencia de Datos MDS-Oct22

Máster en Ingeniería de Datos MDE-Oct22

Contenido

[RESUMEN EJECUTIVO 7](#_Toc148878668)

[INTRODUCCIÓN, ANTECEDENTES Y MOTIVACIÓN 9](#_Toc148878669)

[ARQUITECTURA DE LA SOLUCIÓN 11](#_Toc148878670)

[Visión general del proceso y herramientas utilizadas 11](#_Toc148878671)

[Análisis Exploratorio de la Información Recibida 14](#_Toc148878672)

[Tabla de donaciones recurrentes 15](#_Toc148878673)

[Tabla de modificaciones de cuota 16](#_Toc148878674)

[Tabla de Contactos 17](#_Toc148878675)

[Tabla de Campañas 18](#_Toc148878676)

[Tabla de Oportunidades 19](#_Toc148878677)

[Tabla de Tareas 19](#_Toc148878678)

[Modelo basado en tabla de interacciones con los socios 20](#_Toc148878679)

[Selección de características 22](#_Toc148878680)

[Procesamiento y limpieza de datos 25](#_Toc148878681)

[Almacén de características (Feature Store) 29](#_Toc148878682)

[Elección y construcción del modelo 29](#_Toc148878683)

[Despliegue con MLFlow 32](#_Toc148878684)

[Modelo basado en información histórica de modificaciones de cuota 34](#_Toc148878685)

[Selección de características 37](#_Toc148878686)

[Procesamiento y limpieza de datos 38](#_Toc148878687)

[Elección y construcción del modelo 42](#_Toc148878688)

[PLAN DE TRABAJO 46](#_Toc148878689)

[Reunión con MSF para planteamiento del problema 46](#_Toc148878690)

[Disponibilidad de los datos 46](#_Toc148878691)

[Tratamiento y limpieza de datos 46](#_Toc148878692)

[Selección de principales Driver y segmentaciones de la muestra 47](#_Toc148878693)

[Desarrollo del modelo de predicción, análisis de resultados 47](#_Toc148878694)

[Elaboración de la memoria y Documentación del proyecto 47](#_Toc148878695)

[Defensa del proyecto 47](#_Toc148878696)

[RESULTADOS 48](#_Toc148878697)

[Modelo 1: Red Neuronal Profunda: 48](#_Toc148878698)

[Datasets Generados 48](#_Toc148878699)

[Modelos y rendimiento 48](#_Toc148878700)

[Modelo 2: Regresión Logística: 50](#_Toc148878701)

[Datasets Generados 50](#_Toc148878702)

[Modelo y rendimiento 51](#_Toc148878703)

[Aplicación 51](#_Toc148878704)

[IMPLANTACIÓN, MONETIZACIÓN Y RETORNO DEL PROYECTO 52](#_Toc148878705)

[CONCLUSIONES Y TRABAJO FUTURO 53](#_Toc148878706)

[Resumen de los hallazgos 53](#_Toc148878707)

[Limitaciones 53](#_Toc148878708)

[Posibles mejoras y trabajos futuros 54](#_Toc148878709)

[INFORMES DE ANÁLISIS EXPLORATORIO Y EXPERIMENTOS DE MODELADO 56](#_Toc148878710)

[Análisis preliminar tablas aportadas por MSF 56](#_Toc148878711)

[Profiling 56](#_Toc148878712)

[Preprocesado y limpieza 56](#_Toc148878713)

[PROTOTIPO 57](#_Toc148878714)

[CÓDIGO FUENTE Y BIBLIOGRAFÍA 62](#_Toc148878715)

#### Tabla de Ilustraciones

[Figura 1: Esquema general de la arquitectura utilizada 10](#_Toc148881293)

[Figura 2: Ciclo de Vida con MLFlow 12](#_Toc148881294)

[Figura 3: Precompute Serving Pattern 12](#_Toc148881295)

[Figura 4: Profiling de la variable msf\_closetype\_\_c de la tabla Task obtenido con Sweetviz 20](#_Toc148881296)

[Figura 5: Muestra tabla Task 20](#_Toc148881297)

[Figura 6: Muestra tabla Quota Modification 21](#_Toc148881298)

[*Figura* 7*: Resultado del Join de Task y Quota Modification con campos a NaN* 26](#_Toc148881299)

[Figura 8: Experimentos en MLFlow 31](#_Toc148881300)

[Figura 9: Modelos en MLFlow 32](#_Toc148881301)

[Figura 19: Disminución de la función de pérdida durante el entrenamiento 49](#_Toc148881302)

[Figura 20: Gráfica de la curva ROC 50](#_Toc148881303)

[Figura 21: Aplicación Streamlit con modelo Embebido 58](#_Toc148881304)

[Figura 22: Carga del modelo en la aplicación 58](#_Toc148881305)

[Figura 23: Widget para predicción sobre socio 59](#_Toc148881306)

[Figura 24: Lista de socios con más probabilidad de aumentar la cuota según el modelo 60](#_Toc148881307)

[Figura 25: Filtro para la lista de socios 60](#_Toc148881308)

[Figura 26: Widget con gráfica de número de socios por rango de probabilidad de aumentar cuota 61](#_Toc148881309)

[Figura 27: Web de la aplicación 62](#_Toc148881310)

# RESUMEN EJECUTIVO

El objetivo del proyecto es la predicción del Incremento de Cuotas Periódicas de los Socios de Médicos sin fronteras (MSF).

**Hipótesis de Trabajo**: A través de la información contenida en los conjuntos de datos proporcionados por MSF (información de socios y donaciones periódicas, historial de incrementos de cuotas e interacciones con socios), es posible modelar y predecir qué socios tienen mayor probabilidad de aumentar sus cuotas en un futuro cercano. Para abordar esta hipótesis, hemos implementado dos modelos predictivos: el primero, a través de una red neuronal profunda ,y el segundo, utilizando una regresión logística.

**Metodología y Desarrollo:** Para materializar lo expuesto en la hipótesis se procede a realizar un Análisis Exploratorio de Datos (EDA) con la inspección inicial de los tres conjuntos de datos para comprender su estructura, relaciones y posibles inconsistencias. A continuación se procede al preprocesamiento de datos y selección de características con el propósito de crear tanto un conjunto de datos para entrenamiento y validación como un conjunto de características que servirán como entrada para nuestros modelos, incluyendo una Red Neuronal Profunda (DNN) desarrollada utilizando Tensorflow y un modelo de Regresión Logística. Estos modelos se utilizarán para realizar predicciones y se presentarán a través de una pequeña página web implementada con la herramienta Streamlit.

**Conclusiones:** A pesar de algunas dificultades a la hora de determinar las características más relevantes y filtrar y cruzar la información de los distintos ficheros de datos proporcionados, se pudo obtener un conjunto de datos para entrenamiento y validación con una calidad que consideramos prometedora de cara a conseguir el objetivo planteado en la hipótesis. Tras probar con numerosas combinaciones de hiperparámetros, obtuvimos modelos con métricas de "accuracy", "precisión" y "recall" que indican que existe margen para optimizar aún más los procesos destinados a aumentar las cuotas de los socios. Tanto el modelo de Red Neuronal Profunda (DNN) como el modelo de Regresión Logística se implementaron sin problemas utilizando el framework MLFlow y se integraron en una aplicación Streamlit de fácil acceso.

# INTRODUCCIÓN, ANTECEDENTES Y MOTIVACIÓN

En este proyecto se va a tratar de desarrollar modelos capaces de predecir la probabilidad de aumento de cuota entre los socios donantes recurrentes. El propósito es identificar a aquellos socios que tienen una mayor propensión al aumento de cuota, lo que permitirá realizar acciones específicas de manera eficiente. Esto, a su vez, mejorará la efectividad de las campañas y garantizará una interacción más efectiva con los socios, evitando la sobrecarga de información y solicitudes innecesarias.

Para conseguir este objetivo, planteamos la creación de sendos modelos que permitan discernir cuáles de los socios actuales son más propensos a aumentar sus cuotas. Este modelo se basará en información histórica disponible hasta el momento y se implementará a través de dos enfoques: una Red Neuronal Profunda (DNN) desarrollada con Tensorflow y un modelo de Regresión Logística. Ambos modelos se utilizarán para evaluar la probabilidad de aumento de cuota en función de la información histórica de los socios. Este enfoque permitirá tomar decisiones más informadas y estratégicas en la gestión de los socios y las campañas, mejorando la eficiencia y la efectividad en las interacciones con los socios donantes recurrentes

Para ello se dispone de la base de datos aportada por Médicos Sin Fronteras (MSF a partir de este momento) que contiene información de cuotas, campañas, tareas e información variada sobre los socios con varios años de información acumulada.

Durante el presente documento se explicará cómo se han desarrollado estos modelos, y se explicarán las decisiones clave tomadas en la ejecución de este proyecto.

Como hemos comentado el enfoque es concentrar el estudio en los socios que tengan aportaciones periódicas activas y prever su comportamiento a la hora de elevar su contribución periódica tanto al ser contactado a través de los diferentes canales de comunicación disponibles en MSF, como telemarketing, iteraciones en persona, plataforma web, etc, como globalmente, tratando de conocer cuál es la probabilidad de que aumenten su cuota al año siguiente.

El primer enfoque está más centrado en un caso de uso más específico, ya que no pretende abordar el conjunto de datos en su totalidad para predecir una cantidad global de aumentos de cuota en un período específico, si no que busca determinar la probabilidad individual de que cada socio aumente su cuota si se establece un contacto en el momento actual. Sin embargo, una vez que hemos obtenido las probabilidades individuales, tenemos la capacidad de extrapolar esta información para obtener una visión global. Utilizando un umbral para identificar a los socios con mayor probabilidad de incrementar sus cuotas y considerando los aumentos promedio de cuota observados en el pasado para cada uno de ellos, podemos realizar una estimación del importe global que representaría para todos los socios. Esto nos permite una gestión más estratégica y focalizada, maximizando el impacto de las interacciones con los socios y mejorando la eficiencia de las campañas, lo que a su vez puede traducirse en un incremento significativo en el apoyo financiero a MSF.

El segundo planteamiento lo que pretende es predecir la posibilidad de que un socio aumente su cuota dentro del año siguiente conociendo su historial de años anteriores. Es un enfoque más general y destinado a conocer de forma más global el aumento de cuotas que se producirá en el siguiente año.

En el primer caso, una vez desarrollado un modelo que se considere valioso, se podría medir su efectividad desde una perspectiva comercial, como por ejemplo, el número de interacciones que resultaran positivas durante un periodo determinado lanzando simultáneamente dos campañas de telemarketing sobre un número similar de socios, una que utilice el modelo desarrollado para estimar los socios que estarían más predispuestos a aumentar su cuota y otra que no dispusiera de esta asistencia. Otra métrica que se podría evaluar es la estimación global del importe de incrementos de cuota que se extrapolaría del resultado de la predicción del modelo sobre todos los socios y los aumentos medios de cuota que han hecho en el pasado y compararlo con el valor real que se obtenga en el periodo de estudio.

Con el segundo modelo se podría comparar la estimación con el resultado del año analizado para, comparando, comprobar su efectividad a la hora de predecir el aumento de ingresos esperado.

# ARQUITECTURA DE LA SOLUCIÓN

## Visión general del proceso y herramientas utilizadas

Como se deduce de lo comentado hasta ahora en los puntos anteriores este proyecto tiene un enfoque más orientado a ciencia que a ingeniería de datos. Médicos sin Fronteras nos ha proporcionado un conjunto de datos desde el que extraer la información relevante con la que construir un modelo que permita hacer predicciones acerca de la propensión que puedan tener los socios de aumentar sus cuotas periódicas. Un esquema de las herramientas utilizadas y la secuencia del plan de trabajo realizado se puede ver en la siguiente figura:

Diagrama

Descripción generada automáticamente

Figura 1: Esquema general de la arquitectura utilizada

Las tablas de campañas, contactos (socios y donantes), tareas (interacciones con los anteriores), donaciones periódicas, modificaciones de cuota y oportunidades (pagos concretos) se proporcionan directamente almacenadas en AWS S3, por lo que no es necesario implementar un proceso específico de ingesta. Utilizando las utilidades proporcionadas por el servicio AWS Glue como son los “crawlers”, que permiten la posterior catalogación de los datos, y las sesiones interactivas, para poder trabajar con notebooks de Jupyter se procede, por un lado, a realizar el análisis exploratorio de los datos (EDA) y, por otro, a la limpieza y preprocesamiento de estos. Para estas tareas se ha hecho uso extensivo de herramientas como AWS Athena, Pandas y Apache Spark. Como resultado de las mismas se obtiene un fichero con la selección de todas la características y la variable objetivo que entendemos son prometedoras para la construcción de un modelo de Machine Learning que pueda aportar valor. También preparamos un almacén de características (feature store) con un registro por cada uno de los socios resultado del filtrado realizado en busca de los susceptibles de amentar la cuota.

Con ayuda de pandas y algunas funciones de Scikit Learn se ultima la preparación para que los datos puedan ser utilizados para el entrenamiento y validación de los modelos. Estos son desarrollados tanto con Tensorflow como con Scikit Learn. Hemos utilizado tanto redes neuronales profundas (Deep Neural Network o DNN) por su capacidad de capturar con mayor precisión los patrones que se derivan de un número elevado de características y de las relaciones no lineales que puedan derivarse de ellas, como modelos de regresión logística para implementar los modelos.

Para poder gobernar el ciclo de vida de los modelos, desde los experimentos con diferentes hiperparámetros hasta su despliegue en producción, pasando por el registro de los experimentos seleccionados hacemos uso durante el proyecto del Framework MLFlow.

Diagrama

Descripción generada automáticamente

Figura 2: Ciclo de Vida con MLFlow

Con el modelo ya publicado en MLFlow nos quedará, por último, obtener las predicciones y distribuir el archivo con las mismas e incluso publicarlas con alguna herramienta de visualización. Dado que no es un proceso que requiera realizar las predicciones en tiempo real podemos utilizar el método de despliegue conocido como Precompute Serving Pattern, que obtiene las predicciones en un proceso Batch y las persiste en base de datos o en un fichero.

Diagrama

Descripción generada automáticamente

Figura 3: Precompute Serving Pattern

Por último, esta información puede ser leída y mostrada en una herramienta de visualización o en una página web como las que nos permite hacer el framework Streamlit. Si, por algún motivo fuera necesario conocer predicciones para algunos de los socios en tiempo real, también podría invocar al modelo desde la aplicación Streamlit, o bien integrando el modelo en la propia aplicación, o integrando el modelo en un API e invocando ésta desde la aplicación Streamlit.

## Análisis Exploratorio de la Información Recibida

Para poder plantear un modelo de predicción, lo primero es entender y analizar los datos de los que se dispone. Este es uno de los pasos más importantes a la hora de plantear un modelo, ya que la información disponible y como la trates será fundamental para el éxito o fracaso del modelo.

Por parte de MSF se ha recibido la siguiente información:

* **Tabla Donaciones recurrentes:** Esta tabla contiene información de las donaciones recurrentes, tanto actuales como pasadas. Existiendo un indicador de si la donación recurrente sigue en vigor o no, e información del momento de inicio y final de la misma.
* **Tabla Modificación de cuotas:** En esta tabla se dispone de información sobre las diferentes modificaciones que haya habido históricamente en las cuotas de los socios. Estas pueden ser tanto aumentos como disminuciones de cuotas, o también cambios en la periodicidad de los pagos.
* **Tabla Contactos:** En esta tabla se dispone de información a nivel contacto, como el género, el año de nacimiento, la fecha de entrada como donante/socio/etc. Además de variables como los importes totales donados, o el scoring como donante.
* **Tabla Campañas:** En esta tabla se recogen las diferentes campañas con las fechas en las que han sido realizadas, así como información útil para distinguirlas como el objetivo principal, el segmento al que van dirigidas o los canales por los que se lanzan.
* **Tabla Tareas:** En la tabla de Tareas están identificadas todas las interacciones que se han tenido con los socios/donantes de cualquier tipo, ya sean unidas a campañas o no, teniendo diferentes objetivos como peticiones económicas, informativas, de agradecimiento. Se dispone de las fechas en las que han sido realizadas y el resultado de las mismas.
* **Tabla oportunidades:** Recoge las aportaciones económicas tanto de socios como de donantes, también las herencias. así como las campañas a las que se liga, teniendo fechas de inicio y final si se consigue el objetivo, que se pude identificar a través del tipo. También dispone de una variable para identificar cada una de las etapas en las que se encuentra.

La descripción de los atributos en cada una de las tablas se puede consultar en el siguiente fichero Excel proporcionado por MSF.



En los siguientes apartados se van a realizar los análisis exploratorios de los datos contenidos en las tablas. Al plantearse análisis por variable se va a aportar un resumen de las conclusiones por tabla para posteriormente en el apartado de “Informes de análisis exploratorio y experimentos de modelado” aportar tanto los códigos realizados en pandas, análisis documentados individualmente por variable y profiling de Datasets realizado con Sweetviz.

### Tabla de donaciones recurrentes

Comenzamos con la tabla de donaciones recurrentes que, como se ha dicho almacena un registro por cada proceso de donación recurrente iniciado por el alta de un socio. Estas pueden estar activas o canceladas y un mismo socio puede haber cancelado y abierto múltiples donaciones recurrentes. Primero se ha observado en general la tabla, viendo:

* Tiene 21 campos
* Tiene 1.198.207 registros.
* No existen duplicados
* 8 variables categóricas
* 4 numéricas
* 9 de tipo texto

Como conclusión del análisis exploratorio de esta tabla se puede determinar que:

* Es una tabla con información histórica relevante. Se va a plantear un modelo que tiene como base estas donaciones recurrentes.
* Casi todas las variables están bien informadas, aunque hay algunas con vacíos son lógicos.
* Se consideran como más relevantes a incorporar en los posibles modelos:

|  |  |
| --- | --- |
| **LABEL** | **APINAME** |
| Número de registro | Id |
| Cuota anualizada | msf\_AnnualizedQuota\_\_c |
| Fecha de cancelación | msf\_CancelationDate\_\_c |
| Id Miembro | msf\_MemberId\_\_c |
| Fecha de alta de la cuota regular | npe03\_\_Date\_Established\_\_c |
| Periodicidad | npe03\_\_Installment\_Period\_\_c |
| Cantidad de cuotas pagadas | npe03\_\_Total\_Paid\_Installments\_\_c |
| Canal de entrada 1 inicial | msf\_LeadSource1\_\_c |
| Canal de entrada 1 actual | msf\_CurrentLeadSource1\_\_c |
| Contacto | npe03\_\_Contact\_\_c |

### Tabla de modificaciones de cuota

En este apartado se va a explicar el análisis exploratorio realizado sobre esta tabla. Una vez realizado se hará un resumen con las conclusiones generales de la tabla, así como del planteamiento inicial del uso de la misma para los siguientes pasos del proyecto.

Primero se ha observado en general la tabla, viendo:

* Tiene 16 campos
* Tiene 2.003.409 registros.
* No existen duplicados

Como conclusión del análisis exploratorio de esta tabla se puede determinar que:

* Es una tabla con información histórica relevante.
* Casi todas las variables están bien informadas, casi no se observan variables con nulos o vacíos.
* Es información que se considera consecuencia del objetivo del modelo. Es decir, si se consigue el aumento de cuota, se generará un nuevo registro en esta tabla.
* Se consideran como más relevantes a que se utilizarán para rehacer los valores pasados de las cuotas en cada momento histórico, además de unir información con otras tablas:

|  |  |
| --- | --- |
| **LABEL** | **APINAME** |
| Donación recurrente | msf\_recurringdonation\_\_c |
| Tipo de cambio | msf\_changetype\_\_c |
| Nueva cuota | msf\_newamount\_\_c |
| Nueva cuota anualizada | msf\_newannualizedquota\_\_c |
| Nueva periodicidad | msf\_newrecurringperiod\_\_c |
| Id. de contacto | msf\_contactid\_\_c |
| Fecha de modificación de cuota | msf\_changedate\_\_c |

### Tabla de Contactos

En este apartado se va a explicar el análisis exploratorio realizado sobre esta tabla. Una vez realizado se hará un resumen con las conclusiones generales de la tabla, así como del planteamiento inicial del uso de la misma para los siguientes pasos del proyecto.

Primero se ha observado en general la tabla, viendo:

* Tiene 139 campos
* Tiene 1.803.419 registros.
* No existen duplicados

Como conclusión del análisis exploratorio de esta tabla se puede determinar que:

* Un tercio de los campos tienen algún tipo de valor a missing o nulo
* Variables a fecha de entrega de la tabla, sin histórico.
* Muchas variables se descartan por no tener casi discriminación.
* Se consideran como más relevantes a incorporar en los posibles modelos:

|  |  |
| --- | --- |
| **LABEL** | **APINAME** |
| Id. de contacto | Id |
| Antigüedad en años colaborador MSF | msf\_Seniority\_\_c |
| Año de mejor donación | npo02\_\_Best\_Gift\_Year\_\_c |
| Año de nacimiento | msf\_birthyear\_\_c |
| Campaña de entrada inicial | msf\_EntryCampaign\_\_c |
| Donación promedio | npo02\_\_AverageAmount\_\_c |
| Fecha alta MSF | msf\_BeginDateMSF\_\_c |
| Fecha de cobro de primera cuota de socio | msf\_DateFirstRecurringDonorQuota\_\_c |
| Género | Gender\_\_c |
| Lifetime Value (cont) | msf\_LTVCont\_\_c |
| Lifetime Value (desc) | msf\_LTVDesc\_\_c |
| Provincia | mailingstate |
| Queja por presión en Captación de fondos | msf\_PressureComplaint\_\_c |
| Recencia de socio (cont) | msf\_RecencyRecurringDonorCont\_\_c |
| Recencia de última aportación (cont) | msf\_RecencyTotalCont\_\_c |
| Recencia de última aportación (punt) | msf\_RecencyTotalScore\_\_c |
| Resumen percoms | msf\_PercomsSummary\_\_c |
| RFV de socio | msf\_RFVRecurringDonor\_\_c |
| Scoring RFV de socio | msf\_ScoringRFVRecurringDonor\_\_c |
| Scoring RV Total | msf\_ScoringRVTotal\_\_c |
| Total donado | npo02\_\_TotalOppAmount\_\_c |
| Total importe histórico fiscal | msf\_TotalFiscalOppAmount\_\_c |
| Última cuota anualizada | msf\_LastAnnualizedQuota\_\_c |
| Valor colaborador (cont) | msf\_ValueTotalCont\_\_c |
| Valor colaborador (desc) | msf\_ValueTotalDesc\_\_c |
| Valor máximo de donativo | msf\_MaximumDonorValue\_\_c |
| Valor promedio de donante | msf\_AverageDonorValue\_\_c |
| Vida útil | msf\_LifeTime\_\_c |

### Tabla de Campañas

En este apartado se va a explicar el análisis exploratorio realizado sobre esta tabla. Una vez realizado se hará un resumen con las conclusiones generales de la tabla, así como del planteamiento inicial del uso de la misma para los siguientes pasos del proyecto.

Primero se ha observado en general la tabla, viendo:

* Tiene 24 campos
* Tiene 11.501 registros.
* No existen duplicados

Como conclusión del análisis exploratorio de esta tabla se puede determinar que:

* Es una tabla con información histórica de las campañas.
* Es una tabla aunque no tenga registros a nulo, si tiene campos con bastante información a vacío.
* Es información que puede unificarse con los registros de los id campaña repartidos por el resto de tablas.
* De primeras no se incorporará información al modelo, ya que se ha analizado que en los registros del dataset para modelizar, casi todos tienen los mismos objetivos de campaña, por lo que si una variable no discrimina no va a aportar información al modelo.

### Tabla de Oportunidades

En este apartado se va a explicar el análisis exploratorio realizado sobre esta tabla. Una vez realizado se hará un resumen con las conclusiones generales de la tabla, así como del planteamiento inicial del uso de la misma para los siguientes pasos del proyecto.

Primero se ha observado en general la tabla, viendo:

* Tiene 20 campos.
* Tiene 31.188.223 registros.
* No existen duplicados.

Como conclusión del análisis exploratorio de esta tabla se puede determinar que:

* Es una tabla difícil de ligar con contactos o donaciones concretas.
* Aunque hay un tipo de tarea relacionada con las donaciones recurrentes, los diferentes stage en los que se encuentra tienen que ver con el cobro de las mismas y no con la posibilidad de un aumento de la misma.
* Se ha descartado el uso de la tabla, por lo que ninguna de sus variables será incorporada al modelo.

### Tabla de Tareas

En este apartado se va a explicar el análisis exploratorio realizado sobre esta tabla. Una vez realizado se hará un resumen con las conclusiones generales de la tabla, así como del planteamiento inicial del uso de la misma para los siguientes pasos del proyecto.

Primero se ha observado en general la tabla, viendo:

* Tiene 29 campos.
* Tiene 2.612.004 registros, relacionado con Aumento de Cuota.
* No existen duplicados.

Como conclusión del análisis exploratorio de esta tabla se puede determinar que:

* Es una tabla con pocas variables, pero relevantes para el modelo planteado.
* Se considera como la tabla principal con la información a incorporar en el modelo, siendo la base para un modelo de los planteados.
* Se han analizado las variables principales que pueden incorporarse al modelo y pocas de sus variables tienen registros a nulo o vacíos.
* Estos son las variables que van a seleccionarse para el dataset inicial del modelo:

|  |  |
| --- | --- |
| **LABEL** | **APINAME** |
| Id. de actividad | Id |
| Fecha de cierre | ActivityDate |
| Canal | msf\_Channel\_\_c |
| Tipo de cierre | msf\_CloseType\_\_c |
| Sentido | msf\_InboundOutbound\_\_c |
| Nombre | WhoId |

## Modelo basado en tabla de interacciones con los socios

Como se ha comentado en los puntos introductorios de este documento hemos planteado dos estrategias diferentes a la hora de tratar de predecir los aumentos de cuota de los socios. Sendas estrategias tienen como resultado un modelo reflejo de cada una de ellas. En este punto describimos el trabajo hecho para el desarrollo de un modelo basado en las interacciones que se producen entre MSF y cada uno de los socios.

El planteamiento adoptado para este modelo hace de la tabla Task su tabla maestra ya que contiene las interacciones que se han tenido desde MSF con sus socios tipificadas y calificadas en función del resultado según su temática y objetivo. Al realizar el EDA apreciamos que en esta tabla tenemos una columna “subject” que indica el tipo de tarea que se realiza, entre sus posibles valores tenemos precisamente “MSF Aumento de Cuota” con el que poder realizar un primer filtrado. También, como resultado del profiling de las tablas podemos ver que la columna “*msf\_closetype\_\_c*”, si bien contiene numerosos valores faltantes también contiene una información valiosa por cuanto que denota el resultado de la interacción con el socio. Es por esta razón que la utilizaremos como variable objetivo para el modelo resultante.

Gráfico, Gráfico de embudo

Descripción generada automáticamente

Figura 4: Profiling de la variable msf\_closetype\_\_c de la tabla Task obtenido con Sweetviz

Un ejemplo de parte de la información contenida en la tabla para uno de los socios sería la mostrada en la siguiente captura desde AWS Athena:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 5: Muestra tabla Task

Para ese mismo socio tenemos en la tabla QuotaModification los aumentos de cuota que ha ido haciendo a lo largo del tiempo como vemos en la siguiente captura:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura 6: Muestra tabla Quota Modification

El planteamiento consiste en combinar estas tablas, ejerciendo la de tareas el papel de tabla maestra, es decir, conservando todos los registros de la misma, y asociando a cada registro que corresponda el incremento de cuota que desencadenó dicha interacción. Lamentablemente no hay una forma directa de cruzar ambas tablas porque no hay ningún id que relacione los registros de una con los de la otra, por lo que tenemos que recurrir a otra vía. Básicamente la combinación (Join/Merge) se hará basándonos en las fechas de unos y otros. A cada tarea de tipo Aumento de Cuota que haya tenido resultado Positivo le asociaremos el registro de modificación de cuota cuya fecha de cambio esté dentro de un periodo de 60 días posterior al de la fecha de la tarea. De esta forma ligamos causalmente la tarea con su resultado (aumento de cuota). El plazo determinado puede perder recoger información de donaciones con periodicidad no mensual si la tarea se realiza con mucha antelación a la siguiente fecha de puesta al cobro de la cuota.

Para realizar estas combinaciones o joins haremos uso de la herramienta Apache Spark debido al gran tamaño de los ficheros. Utilizaremos el servicio de notebooks de AWS Glue que nos permite realizar las operaciones dentro de un notebook de Jupyter.

### Selección de características

Una vez realizado el Join entre las dos tablas anteriores seleccionamos las columnas que consideramos pueden proporcionar información relevante al modelo. Adicionalmente combinamos este dataset con las tablas de contactos y la de donaciones recurrentes para enriquecer el dataset con la información relevante que estas puedan contener. Así, de cada tabla nos quedamos con las siguientes columnas:

**Tareas (Task):**

Características Seleccionadas:

|  |  |
| --- | --- |
| Columna | Descripción |
| Whoid | Id del contacto vinculado a la tarea |
| Activitydate | Fecha en la que se ha actualizado el último Estado de la tarea |
| msf\_channel\_\_c | Canal por el que se ha vehiculizado la tarea |
| msf\_inboundoutbound\_\_c | Sentido de la tarea (Saliente = le contactamos nosotros; Entrante = contacta con nosotros |
| msf\_closetype\_\_c | Cierres aplicados únicamente a las tareas salientes (llamadas, emails, etc que enviamos) |

Filtros:

|  |  |
| --- | --- |
| Columna | Valor |
| subject | "MSF Aumento de Cuota" |
| isdelete | False |

**Modificaciones de Cuota (QuotaModification):**

Características Seleccionadas:

|  |  |
| --- | --- |
| Columna | Descripción |
| msf\_changedate\_\_c | Fecha en la que se ha realizado la modificación de cuota |
| msf\_leadsource3\_\_c | Canal mediante el que se ha realizdo la modificación de cuota (agrupación 3) |
| msf\_newamount\_\_c | Nueva cuota (no anualizada) tras la modificación de cuota |
| msf\_newannualizedquota\_\_c | Nueva cuota anualizada tras la modificación de cuota |
| msf\_newrecurringperiod\_\_c | Nueva periodicidad (si no ha cambiado la periodicidad aparecerá la que ya existía antes de la modificación) |

Filtros:

|  |  |
| --- | --- |
| Columna | Valor |
| isdeleted | False |

**Socios (Contac):**

La tabla de Socios o Contactos se combina con la de tareas filtrándola previamente para garantizar que estamos tratando con un socio y no un donante, también chequeamos que el contacto permite la comunicación por todas las vías o por varias de ellas. Las columnas seleccionadas de esta tabla son:

Características Seleccionadas:

|  |  |
| --- | --- |
| Columna | Descripción |
| msf\_seniority\_\_c | Antigüedad en años colaborador MSF |
| msf\_birthyear\_\_c | Año de nacimiento |
| msf\_entrycampaign\_\_c | Campaña de entrada inicial |
| npo02\_\_best\_gift\_year\_\_c | Año fiscal en que se ha realizado mayor importe total |
| npo02\_\_averageamount\_\_c | Media de importe de donativos que ha realizado |
| gender\_\_c | Género |
| msf\_ltvcont\_\_c | Valor de la suma de todas la aportaciones cobradas al colaborador (donativos y cuotas) |
| msf\_recencyrecurringdonorcont\_\_c | Número de días entre la fecha de la última cuota de socio cobrada y la fecha actual |
| msf\_recencytotalcont\_\_c | Número de días entre la fecha de cobro de la última aportación cobrada (cuota o donativo) y la fecha actual |
| msf\_recencytotalscore\_\_c | Puntuación asignada según recencia\_cont\_total:  0 o Nulo |
| msf\_rfvrecurringdonor\_\_c | RFV de socio |
| npo02\_\_totaloppamount\_\_c | Suma de importe del total de oportunidades cobradas |
| msf\_valuetotalcont\_\_c | Valor colaborador (cont) |
| msf\_valuetotaldesc\_\_c | Valor colaborador (desc) |
| msf\_lifetime\_\_c | Número de años (número entero, redondeo a la baja) desde fecha de cobro de primera aportación cobrada hasta fecha de cobro de última aportación cobrada. |
| msf\_pressurecomplaint\_\_c | Queja por presión en Captación de fondos |
| msf\_scoringrfvrecurringdonor\_\_c | Scoring RFV de socio |
| msf\_scoringrvtotal\_\_c | Scoring RV Total |
| msf\_averagedonorvalue\_\_c | Valor promedio de donante |
| msf\_percomssummary\_\_c | "Agrupación según los permisos de comunicación. Valores:  Nada,No captación de fondos,Sólo certificado fiscal,Todo,Varios" |
| msf\_totalfiscaloppamount\_\_c | Total importe histórico fiscal |
| msf\_maximumdonorvalue\_\_c | Valor máximo de donativo |

Filtros:

|  |  |
| --- | --- |
| Columna | Valor |
| msf\_isactiverecurringdonor\_\_c | 'Socio' |
| msf\_percomssummary\_\_c | 'Todo','Varios' |

**Donaciones recurrentes (RecurringDonation):**

La tabla de donaciones recurrentes se cruza con la de modificaciones de cuota para filtrar aquellas que están activas por un lado, y para recuperar algunos campo de interés por otro.

Características Seleccionadas:

|  |  |
| --- | --- |
| Columna | Descripción |
| msf\_annualizedquota\_\_c | Valor estandarizado que aporta el socio, resultado de multiplicar el campo importe por un valor según la periodicidad (12 si es mensual, 6 si es bimensual, 4 si es trimestral, 2 si es semestral o 1 si es anual) |
| npe03\_\_total\_paid\_installments\_\_c | Número de oportunidades vinculadas con esta recurring donation que están con etapa Cobrada |
| npe03\_\_installment\_period\_\_c | Mensual, bimensual, trimestral, semestral o anual |

Filtros:

|  |  |
| --- | --- |
| Columna | Valor |
| npe03\_\_open\_ended\_status\_\_c | 'Open' |
| isdeleted | False |

### Procesamiento y limpieza de datos

Sobre el Dataset resultante de las combinaciones anteriores es necesario realizar aún distintas transformaciones para adecuar los datos al tipo de modelo que utilizaremos (DNN) y tratar de mejorar el aprendizaje en la medida de los posible.

Uno de los primeros pasos es crear una nueva columna en la que computamos agrupando por socio los días que han pasado respecto a la tarea anterior basándonos en la columna “activitydate” de la tabla de tareas. Lo hacemos así porque pensamos que es interesante que el modelo pueda interpretar el dato de los días transcurridos desde la última interacción con el socio. De hecho, esta información determina que se intente o no realizar la interacción si el tiempo transcurrido es menor de, por ejemplo, un año. Para poder hacer esto utilizamos las utilidades de ventanas que nos proporciona “Pyspark” y funciones como “lag” y “datediff”.

A continuación, hacemos lo propio con los días transcurridos desde el anterior cambio de cuota de cada socio. El motivo es que parece interesante tener esta información en todos los registros para luego rellenar información en campos que se hayan quedado en blanco al hacer alguno de los “Left Joins”. También añadiremos al registro la fecha del anterior cambio de cuota.

Por la misma razón también nos guardamos en los registros otra información que nos servirá para rellenar campos que quedan a null en los Joins como es el caso de las columnas “msf\_oldamount\_\_c”, “msf\_oldannualizedquota\_\_c” y “msf\_oldrecurringperiod\_\_c” que contendrán en cada registro los correspondientes valores “msf\_newamount\_\_c”, “msf\_newannualizedquota\_\_c” y “msf\_newrecurringperiod\_\_c” del registro anterior cronológicamente.

Como vemos en la imagen siguiente, al hacer un left join con la tabla Task como tabla izquierda, tenemos valores a nulos en aquellos registros de Quota Modification que no han hecho match.

Imagen que contiene interior, tabla, decorado, pastel

Descripción generada automáticamente

*Figura* 7*: Resultado del Join de Task y Quota Modification con campos a NaN*

Para rellenar la información que queda a nulos en los datos de importes de cada socio nos apoyamos en los registros que sí disponen de ella y la propagamos hacia adelante cronológicamente hasta que encontramos un registro con esa información informada. Repetimos el proceso hacia adelante a partir de este último con los datos que contenga. Como decimos esto se hace agrupando por socio.

También puede ocurrir que se nos quede información a nulos desde el primer registro que tengamos por lo que repetimos el proceso en la dirección contraria, ordenando los registros por activitydate en sentido descendente. Eso rellenará los nulos con los valores “old” que tuviéramos.

Ciertas columnas se rellenan utilizando la función coalesce de pyspark, que elige el primer valor no nulo entre sus argumentos. Esto ayuda a rellenar huecos en los datos de determinados campos. Por ejemplo, podemos rellenar el campo “msf\_newamount\_\_c” con el “msf\_oldamount\_\_c” si el primero está vacío.

En otros casos, como en “npe03\_\_total\_paid\_installments\_\_c” rellenaremos con el mismo valor que tenga cualquier otro de los registros del mismo socio.

Calculamos también la edad del socio para expresarla en años y no tener que lidiar con fechas que nuestra DNN no va a entender. Para ello creamos una columna “age” restando el año de nacimiento del año actual. En aquellos casos en los que la edad no está informada utilizamos la media de las edades de todo el dataset.

Los valores de las columnas de tipo categórico es necesario pasarlos a numérico por una de las dos vías siguientes. O bien se les asocia una numeración ascendente mapeando cada valor estableciendo de esa forma una suerte de asociación jerárquica, que no tiene mucho sentido en algunos casos, o se realiza una transformación de tipo “One Hot Encode” si no queremos establecer dicha jerarquía y que, creará una nueva columna por cada uno de los valores de la variable categórica.

En nuestro caso utilizaremos One Hot Encoding para las variables: “gender\_c” (Género) y “msf\_channel\_\_c” (El canal de comunicación) y asociaremos una numeración secuencial para datos como “msf\_valuetotaldesc\_\_c” (Valor colaborador), “msf\_newrecurringperiod\_\_c” (Periodicidad de la cuota), “msf\_pressurecomplaint\_\_c” (si el socio se ha quejado por presión de telemarketing) y “msf\_percomssummary\_\_c” (resumen de vías de comunicación permitidas).

En cuanto a nuestra variable objetivo trataremos cualquier interacción que no sea positiva o potencial como negativa. La potencial la transformaremos por tanto a positiva en caso de que se haya materializado la subida de cuota dentro de los dos siguientes meses.

Es importante en el conjunto de datos el número de filas de cada uno de los valores de la variable objetivo esté equilibrado. En este caso tenemos dos valores posibles 0: Negativo o 1: Positivo, por lo que utilizaremos la función “*resample*” de “*sklearn.utils*” para obtener un conjunto de datos balanceado.

Por último, borraremos del fichero datos que no puedan ser procesados por una DNN como fechas e Ids (si bien conservaremos el id del contacto para su posterior utilización) y ordenaremos las columnas de forma que el id del contacto sea la primera y la variable objetivo la última, simplemente por una cuestión de facilitar la visualización.

El dataset resultante de todas estas operaciones se vuelve a escribir en S3 en formato Parquet, sobrescribiendo cualquier dato existente con el mismo nombre. Este conjunto de datos es el que nos servirá para entrenar y evaluar nuestra modelo.

### Almacén de características (Feature Store)

Para este tipo de proyecto es útil crear un almacén de características que permitirá posteriormente tener preparada la información con la que alimentar a la red neuronal y obtener las predicciones. Este conjunto de datos tendrá las features ya preprocesadas para cada uno de los socios con potencial para incrementar su cuota periódica.

Para obtenerlo se conserva el registro más reciente cronológicamente de cada contacto (identificado por el su id) del fichero obtenido en el proceso descrito en el punto anterior.

El DataFrame filtrado se guarda también en S3 para tenerlo disponible a la espera de que tengamos un modelo con el que realizar las predicciones.

### Elección y construcción del modelo

Como se ha comentado en puntos anteriores el modelo elegido para tratar de capturar el patrón que permita discernir con mayor exactitud que socios son más propensos a aumentar su cuota es una red neuronal profunda (DNN). La justificación reside principalmente en que entendemos que el patrón que intentamos resolver es complejo, en el sentido de que, el hecho de que un socio decida o no aumentar su cuota puede obedecer a múltiples razones y, de hecho, ser diferente aunque el contexto en el que se produce la decisión sea similar. También tenemos muchos datos, lo que ayuda a que los modelos de DNN resultantes sean mas eficientes.

Para poder entrenar la red neuronal utilizaremos el conjunto de datos que obtuvimos en el apartado de Preprocesamiento y Limpieza de datos. Dado que vamos a utilizar el Framework MLFlow para realizar nuestros experimentos con diferentes modelos, por comodidad, descargamos el conjunto en una máquina local y utilizaremos las librerías Pandas y Scikit Learn para ayudarnos a preparar los conjuntos de entrenamiento y test. Separamos en primer lugar las columnas predictoras (quitando primero el id del socio) y la columna a predecir (msf\_closetype\_\_c). A continuación, utilizando la función “*train\_test\_split”* del paquete “*sklearn.model\_selection*” separamos el conjunto de datos en conjunto de entrenamiento (x\_train), variable predictora de entrenamiento (y\_train) y las correspondientes para validación y test (x\_test e y\_test). Con ello tenemos ya preparados los datos para entrenar el modelo.

Es entonces cuando utilizamos los Frameworks Tensorflow y MLFlow para comenzar a experimentar con los modelos.

Los hiperparámetros que utilizaremos de inicio y con los que jugaremos hasta encontrar el modelo que nos dé mejores resultados serán los siguientes: epochs, learning\_rate, número de layers (n\_layers) y número de nodos (nodes), optimizer y los.

Dado que nuestro problema es del tipo de clasificación binario la función de pérdida la mantendremos como '*binary\_crossentropy'* y el optimizador también lo fijaremos como ‘*adam*’. La función de activación para nuestra capa de salida será de tipo “*sigmoid* “, ya que buscamos que el resultado sea una probabilidad que nos indique de cuál de los dos valores estamos más cerca. La función de activación en las capas profundas de la DNN será la ReLU.

Utilizamos algunos de los Callbacks que nos proporciona Tensorflow para la fase de aprendizaje, en concreto, para detención temprana y para reducción progresiva de la tasa de aprendizaje, en concreto son:

**early\_stop**: Que detiene el entrenamiento tempranamente si no observa mejoras en la métrica de “accuracy” después de un cierto número de épocas.

**reduce\_lr**: Reduce la tasa de aprendizaje si no hay mejoras en la “accuracy” después de un cierto número de épocas.

Para evaluar el modelo utilizaremos las métricas de “*accuracy*“, “precision” y “recall” tanto en el conjunto de entrenamiento como en el de Test, aunque serán las de éste último las que nos darán la verdadera medida de la calidad del modelo.

**Inicio del registro en MLflow**

Para hacernos más fácil el gobierno del ciclo de vida de los modelos con los que experimentar empleamos MLFlow que nos permite registrar toda la actividad y recoger las métricas y parámetros utilizados, así como registrar el modelo y salvarlo y publicarlo en “Producción” para su uso posterior. Los pasos que se irán dando serán:

* Para ello se inicia un registro (run) en MLflow con un “experiment\_id” específico y un nombre de ejecución "msf-quote-increase".
* A continuación, se registra cada hiperparámetro del modelo en MLflow para rastreo y referencia futura.
* Se crea un modelo secuencial con capas densas. Las capas densas se generan en función de los hiperparámetros definidos anteriormente. La última capa tiene una única neurona con una activación sigmoidal ya que se trata de un problema de clasificación binaria.
* Se compila el modelo especificando el optimizador, la función de pérdida y las métricas a utilizar
* Se entrena el modelo utilizando los datos de entrenamiento (x\_train, y\_train) los callbacks previamente definidos y los datos de validación (x\_test, y\_test)
* Se evalúa el modelo tanto en el conjunto de entrenamiento como en el de test para obtener las métricas de precisión, recall y exactitud o accuracy

Los modelos resultantes quedan registrados en MLFlow, lo que nos permite realizar distintos experimentos y recuperar modelos probados con anterioridad así como publicar y versionar aquellos que hayamos visto que tienen mayor interés. En la siguiente captura vemos como MLFlow registra los entrenamientos de los modelos y sus resultados.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Figura 8: Experimentos en MLFlow

### Despliegue con MLFlow

Aquellos modelos que han proporcionado mejores resultados pueden ser versionados y promocionados a dos posibles estados, Staging y Production. En este último el modelo ya estaría preparado para ser cargado desde cualquier aplicación y comenzar a hacer predicciones. En Python, la carga del modelo sería tan sencilla como ejecutar el siguiente comando: “mlfow.pyfunc.load\_model” y proporcionarle la uri del modelo.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

Figura 9: Modelos en MLFlow

Una vez tenemos el modelo preparado abordamos la implementación de un prototipo de aplicación con el framework Streamlit ya que nos permite desarrollar webs sencillas de forma muy rápida. La descripción de esta prueba de concepto se hace de forma más detallada en el apartado “Prototipo” de este documento.

## Modelo basado en información histórica de modificaciones de cuota

El planteamiento de este segundo modelo se basa en la necesidad de tener información de aumento de cuota para las donaciones recurrentes para una ventana de estimación concreta.

Es decir, como las tablas de donaciones recurrentes y modificación de cuota tiene históricos, se puede aprovechar que generar un modelo de regresión que utilice este histórico para enriquecer la información.

De esta manera, se van a realizar el dataset de entrada al modelo utilizando información de varias ventanas temporales. Utilizando la tabla de donaciones recurrentes se puede conocer cuáles de las donaciones estaban activas a cada cierre de año. Aunque existe más histórico, se va a utilizar información desde 2015 a 2021, ya que la información de 7 años parece suficiente y si fuera más amplia sería difícil de manejar en cuento a tiempos de ejecución.

Con lo cual, el planteamiento principal del modelo es poder tener un desempeño común en la observación de si esa donación va o no a aumentar cuota en el corto plazo. La variable objetivo de este modelo será si estas donaciones aumentan o no de cuota en los siguientes 12 meses de observación. De esta manera, ante cada una de las situaciones existentes en cada fin de año para cada donación, el modelo deberá predecir si cada una de esas cuotas aumentará o no antes del fin del año siguiente.

Para ello, para cada una de estas 7 tablas de donaciones recurrentes vivas a final de cada año desde 2015 a 2021, se le va generar esta variable objetivo, que tomará el valor de 1 en caso de que si aumente de cuota en el siguiente año y 0 en cualquier otro caso.

Para poder crear estas tablas anuales se realizarían los siguientes pasos:

* Se fija el cierre de año sobre el que se quiere crear la tabla. Se realizará con todos los años desde 2015 hasta 2021. Para la explicación se pondrá como ejemplo el 2015.

|  |  |
| --- | --- |
| Variable externa | Valor |
| FECHA | 31/12/2015 |

* Partiendo de la tabla de donaciones recurrentes se filtrarán las donaciones cuya fecha de alta de la donación regular (npe03\_Date\_Established\_\_c) sea menor o igual a 31 de diciembre de 2015. Habrá que dejar de considerar a las donaciones que se hayan dado de baja antes de fin de año, ya que no sean donaciones activas en ese momento. Para ello se eliminarán los registros que tengan una fecha de cancelación (msf\_CancelationDate\_\_c) informada (no vacía) que sea menor o igual a 31 de diciembre de 2015. Para facilitar la comprensión de los pasos a esta tabla la llamaremos TABLA1.

|  |  |
| --- | --- |
| **TABLA DONACIONES RECURRENTES** | |
| **LABEL** | **APINAME** |
| Número de registro | Id |
| Cuota anualizada | msf\_AnnualizedQuota\_\_c |
| Fecha de cancelación | msf\_CancelationDate\_\_c |
| Id Miembro | msf\_MemberId\_\_c |
| Fecha de alta de la cuota regular | npe03\_\_Date\_Established\_\_c |
| Periodicidad | npe03\_\_Installment\_Period\_\_c |
| Cantidad de cuotas pagadas | npe03\_\_Total\_Paid\_Installments\_\_c |
| Canal de entrada 1 inicial | msf\_LeadSource1\_\_c |
| Canal de entrada 1 actual | msf\_CurrentLeadSource1\_\_c |
| Contacto | npe03\_\_Contact\_\_c |

Filtros:

|  |  |
| --- | --- |
| Columna | Valor |
| npe03\_\_Date\_Established\_\_c | Igual o inferior a FECHA |
| msf\_CancelationDate\_\_c | Nula o superior a FECHA |

* Como el objetivo es predecir el aumento de cuotas, pero en la tabla de “*Modificación de cuota*”, además de aumentos hay otros tipos de cambios, se creará una tabla con los registros que pertenezcan a aumentos, sin considerar el resto de los cambios. Esta tabla “*Modificación Aumentos*”, se creará, como se ha comentado, a partir de la tabla total de “*Modificación de cuota*”, pero filtrando los aumentos a través del campo “*Tipo de Cambio*” (msd\_changetype\_\_c) cuando toma el valor “Increase”. Para facilitar la compresión del flujo de trabajo, a esta tabla la llamaremos TABLA2.

|  |  |
| --- | --- |
| **TABLA MODIFICACION DE CUOTA** | |
| **LABEL** | **APINAME** |
| Donación recurrente | msf\_recurringdonation\_\_c |
| Tipo de cambio | msf\_changetype\_\_c |
| Nueva cuota | msf\_newamount\_\_c |
| Nueva cuota anualizada | msf\_newannualizedquota\_\_c |
| Nueva periodicidad | msf\_newrecurringperiod\_\_c |
| Id. de contacto | msf\_contactid\_\_c |
| Fecha de modificación de cuota | msf\_changedate\_\_c |

Filtros:

|  |  |
| --- | --- |
| Columna | Valor |
| msf\_changetype\_\_c | Igual a “Increase” |

* El siguiente paso será la creación de la variable objetivo. Para ello se va a proceder a la unión de ambas tablas anteriores (tabla1 y tabla2) a través de una unión LEFT. Es decir que los registros de la tabla1 en ningún caso se pierden, existan o no en la tabla2. El flag de aumento de cuota (“*FLAG\_AUMENTO”*) que tomará el valor 1 en caso de que durante el siguiente año se haya realizado un aumento de cuota y 0 en cualquier otro caso. Para ello, utilizando la información de la tabla “*Modificación Aumentos*” creada en el paso anterior, se generará esta nueva variable “*FLAG\_AUMENTO”* como un 1 en caso de que la fecha de modificación de cuota (“msf\_changedate\_\_c) esté dentro del año siguiente al analizado. En el caso del ejemplo que se está explicando, tomará valor 1 en caso de que exista un registro en la tabla “*Modificación Aumentos*” que sea mayor a 31 de diciembre de 2015 y menor o igual a 31 de diciembre de 2016.

|  |  |
| --- | --- |
| Variable externa | Valor |
| FECHA\_MAS 1 | 31/12/2016 |

|  |  |
| --- | --- |
| **LABEL** | **APINAME** |
| Donación recurrente | msf\_recurringdonation\_\_c |
| Tipo de cambio | msf\_changetype\_\_c |
| Nueva cuota | msf\_newamount\_\_c |
| Nueva cuota anualizada | msf\_newannualizedquota\_\_c |
| Nueva periodicidad | msf\_newrecurringperiod\_\_c |
| Id. de contacto | msf\_contactid\_\_c |
| Fecha de modificación de cuota | msf\_changedate\_\_c |

Filtros:

|  |  |
| --- | --- |
| Columna | Valor |
| msf\_changedate\_\_c | Superior a FECHA e igual o inferior a FECHA\_MAS1 |

De esta manera, se dispone de esta información básica: Donaciones activas en el cierre de cada año desde 2015 a 2021 con los flag de aumento de cuota dentro del año natural siguiente.

Una vez realizados estos pasos, se unifica la información en un único dataframe.

### Selección de características

Una vez se tienen los registros principales de donaciones activas a cada cierre de año, y se tiene el flag de aumento que es la variable objetivo, se va a unir información del resto de tablas como variables input al modelo con las que se pueda predecir el objetivo.

De esta manera, tras haber comentado en el paso "Análisis Exploratorio de la Información Recibida” cuales de las variables tienen tanto calidad como potencial para el modelo, se van a incorporar:

De las tablas utilizadas para la unión, ya se conoce las variables seleccionadas, que se han ido manteniendo en las tablas. Del resto de tablas:

**Contactos:**

La tabla de contactos va a aportar información sobre el socio, tal como género o fecha de nacimiento.

Características Seleccionadas:

|  |  |
| --- | --- |
| **TABLA CONTACTOS** | |
| **LABEL** | **APINAME** |
| Id. de contacto | Id |
| Antigüedad en años colaborador MSF | msf\_Seniority\_\_c |
| Año de nacimiento | msf\_birthyear\_\_c |
| Campaña de entrada inicial | msf\_EntryCampaign\_\_c |
| Género | Gender\_\_c |
| Lifetime Value (cont) | msf\_LTVCont\_\_c |
| Provincia | mailingstate |
| Queja por presión en Captación de fondos | msf\_PressureComplaint\_\_c |
| Scoring RFV de socio | msf\_ScoringRFVRecurringDonor\_\_c |
| Valor promedio de donante | msf\_AverageDonorValue\_\_c |

### Procesamiento y limpieza de datos

Sobre el Dataset resultante de las combinaciones anteriores es necesario realizar aún distintas transformaciones para adecuar los datos al tipo de modelo que utilizaremos.

Entre ellas, está el tratamiento a ciertas variables para que el modelo pueda trabajar sobre ellas como son:

* EDAD: Se creará la variable edad como la diferencia entre el año de la fecha actual, en este caso sería 2023, y la variable “msf\_birthyear\_\_c” que corresponde con el año de nacimiento.
* PROVINCIA: Como la variable “mailingstate” variable tiene más de 1000 posibles valores, se ha realizado un tratamiento de la misma para poder asignar a las 5 más comunes dentro del conjunto de datos que son: Madrid, Cataluña, Pais Vasco, Valencia y Sevilla, Se creará también la categoría Resto para la que no pertenezca a ninguna de las anteriores. Si identificará cada provincia de la siguiente manera creadas son:
  + Madrid: si la variable “mailingstate“, transformada a minúsculas, tome alguno de estos valores 'madrid|madird'.
  + Barcelona: si la variable “mailingstate“, transformada a minúsculas, contenga alguno de estos valores 'bar.\*ona|lle.\*da|le.\*da|tarr.\*|gi.\*ona|ge.\*ona|catal.\*|bad.\*ona|ter.\*ona|ter.\*sa'.
  + Sevilla: si la variable “mailingstate“, transformada a minúsculas, contenga alguno de estos valores ''sev.\*'.
  + Pais Vasco: si la variable “mailingstate“, transformada a minúsculas, contenga alguno de estos valores 'pa.\*co|gu.\*coa|g.\*k|vi.\*y|bi.\*y|vi.\*k|bi.\*k|alava.\*|alaba.\*|san.\* se|dono.\*|vit.\*'.
  + Valencia: si la variable “mailingstate“, transformada a minúsculas, contenga alguno de estos valores 'va.\*ia'.
  + Resto: en caso de que no sea ninguno de los anteriores.

A continuación, después de generar las variables deseadas se van a eliminar las variables que no se van a considerar para el modelo, que son:

|  |  |
| --- | --- |
| **VARIABLES DESCARTADAS** | |
| msf\_campaigninfluence\_\_c | msf\_contactid\_\_c |
| msf\_memberid\_\_c | msf\_changetype\_\_c |
| npe03\_\_contact\_\_c | msf\_changedate\_\_c |
| msf\_campaigninfluence\_\_c' | msf\_recurringdonation\_\_c |
| msf\_FirstCampaignEntryRecurringDonor\_\_c | npe03\_\_recurring\_donation\_campaign\_\_c |
| Mailingstate | Id |
| Isdeleted | msf\_cancelationdate\_\_c |
| msf\_EntryCampaign\_\_c | npe03\_\_date\_established\_\_c |
| msf\_FirstCampaingColaboration\_\_c | msf\_currentquotamodification\_\_c |
| quo\_id | msf\_birthyear\_\_c |

ANALISIS VARIABLES NUMERICAS:

Posteriormente se procede al análisis cuantitativo de cada una de las variables numéricas. Por un lado, se realiza un análisis de los valores medios, máximos, mínimos, etc a través de un describe:

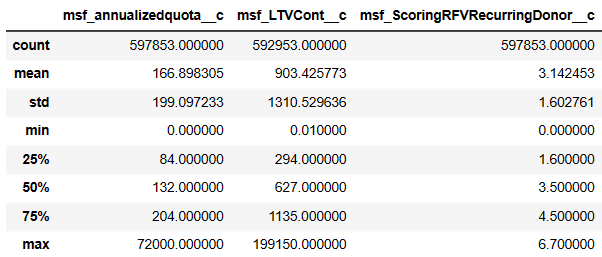


Figura 10: Análisis distribución cuantitativa variables numéricas, ejemplo para 3 variables

También se analiza la distribución de las variables a través de gráficas, un ejemplo para una de las variables. El análisis completo se encuentra en el código “Modelo2\_TFM” adjuntado en el apartado “Informes de análisis exploratorio y experimentos de modelado”, subapartado: “Preprocesado, limpieza y Modelo histórico de modificaciones de cuota”:

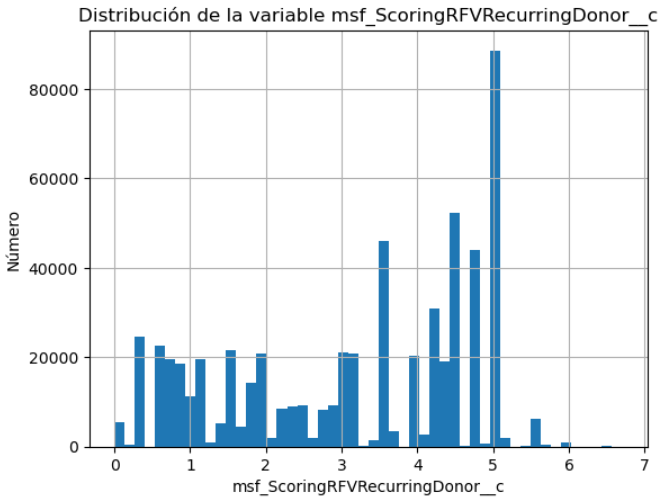


Figura 11: Análisis distribución de la variable msf\_ScoringRFVRecurringDonor\_\_c

Después de estos análisis, se analizan las variables finales, cuales tiene nulos o vacíos y se procede a realizar un tratamiento para los mismos. Se comprueba que solo 3 de las variables tienen algún tipo de nulo, y se decide sobrescribir estos nulos con el valor de la mediana de las variables, para no distorsionar la muestra. Estas tres variables eran:

* Edad
* msf\_LTVCont\_\_c
* msf\_AverageDonorValue\_\_c

Por último, se analiza si pueden existir algún tipo de distorsión en las mismas por la existencia de valores atípicos.

ANALISIS VARIABLES CATEGORICAS:

Para las variables categóricas se sigue procedimientos similares, pero adaptados a este tipo de variables. También se analiza su distribución gráficamente:

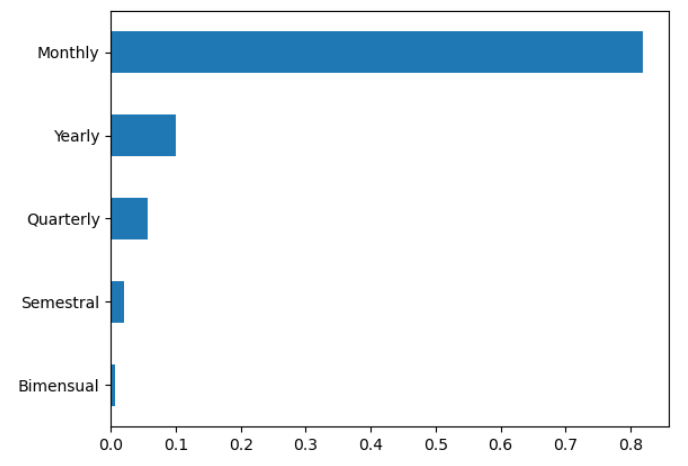


Figura 12: Análisis distribución de la variable npe03\_\_Installment\_Period\_\_c

Una vez analizados los resultados se procede a tratar alguna de ellas, como género o canal que tienen valores parecidos en diferentes categorías. Generando las variables agrup\_gender, agrup\_currentleadsource, agrup\_leadsource.

Por último, se procede a la conversión en variable booleanas de las mismas a través del procedimiento get\_dummies.

|  |  |
| --- | --- |
| **VARIABLES TRANFORMADAS EN BOOLEANAS PARA CADA CATEGORIA** | |
| npe03\_\_installment\_period\_\_c | msf\_PressureComplaint\_\_c |
| agrup\_gender | msf\_leadsource3\_\_c |
| msf\_newrecurringperiod\_\_c | Provincia2 |
| agrup\_currentleadsource | agrup\_leadsource |

Mediante una matriz de correlaciones vemos la relación entre las variables independientes y la variable objetivo.

Por último, detectamos que nuestro conjunto de datos presenta un desequilibrio significativo en la distribución de la variable objetivo, esta desproproción puede llevar a un modelo sesgado y poco preciso, por lo realizamos la técnica de oversampling (SMOTE dentro del paquete imblearn.over\_sampling) para generar datos adicionales en la clase minoritaria de la variable objetivo para que nuestro modelo aprenda de manera más eficiente y mejore su capacidad.

Este conjunto de datos es el que nos servirá para entrenar y evaluar nuestro modelo.

### Elección y construcción del modelo

Como se mencionó en secciones anteriores, hemos optado por emplear un modelo de regresión logística para abordar la tarea de identificar con mayor precisión qué socios son más propensos a aumentar sus cuotas. La decisión de un socio de incrementar su cuota puede estar influenciada por diversas razones, y de hecho, puede variar incluso en situaciones similares. Esta elección se basa en la idoneidad de la regresión logística para modelar variables dependientes binarias, como es el caso de nuestra variable objetivo, y en su capacidad para proporcionar interpretaciones claras de los coeficientes en términos de probabilidades. Tras un análisis minucioso de varios modelos, concluimos que la regresión logística es la opción más adecuada para nuestro estudio.

Comenzaremos creando nuestro modelo de regresión logísitica utilizando el conjunto de datos que hemos procesado y limpiado previamente. Este conjunto de datos contiene tanto las variables predictoras como la variable objetivo que estamos tratando de predecir, en nuestro caso, flag\_aumento.

Para poder entrenar y evaluar nuestro modelo, primero dividimos nuestro conjunto de datos en dos partes: entrenamiento y prueba. Utilizamos la función “*train\_test\_split”* del paquete “*sklearn.model\_selection*” para lograrlo. El conjunto de entrenamiento (X\_train) contiene las variables predictoras de entrenamiento, mientras que el conjunto de prueba (X\_test) contiene las variables para validación y pruebas. También separamos la variable objetivo en dos conjuntos correspondientes Y\_train e Y\_test.

Para garantizar que nuestras variables tengan la misma escala y eliminar cualquier sesgo que pueda existir en los datos, aplicamos “StandardScaler” que es una técnica de normalización disponible en el paquete “sklearn.preprocessing”, con la normalización aseguramos que las variables tengan una media de 0 y una desviación estándar de 1.

Con todo esto ya estamos listos para entrenar el modelo que se ajusta a los datos de entrenamiento y aprende a hacer predicciones basadas en las variables predictoras.

Una vez que el modelo está entrenado, lo utilizamos para hacer predicciones en el conjunto de validación, lo que nos permite evaluar qué tan bien se comporta nuestro modelo en los datos “no vistos”.

Para evaluar la calidad del modelo utilizaremos las siguientes métricas de calidad en el conjunto de prueba:

* F1 Score

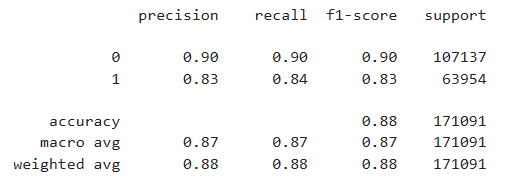


Figura 13: Resultado F1 Score del modelo de regresión logística

* Matriz de confusión

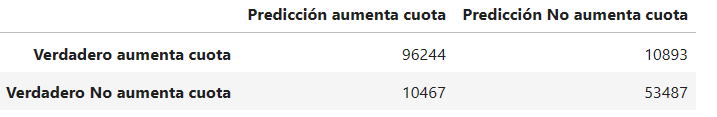
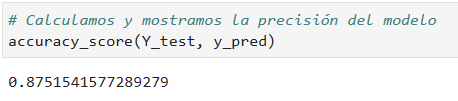


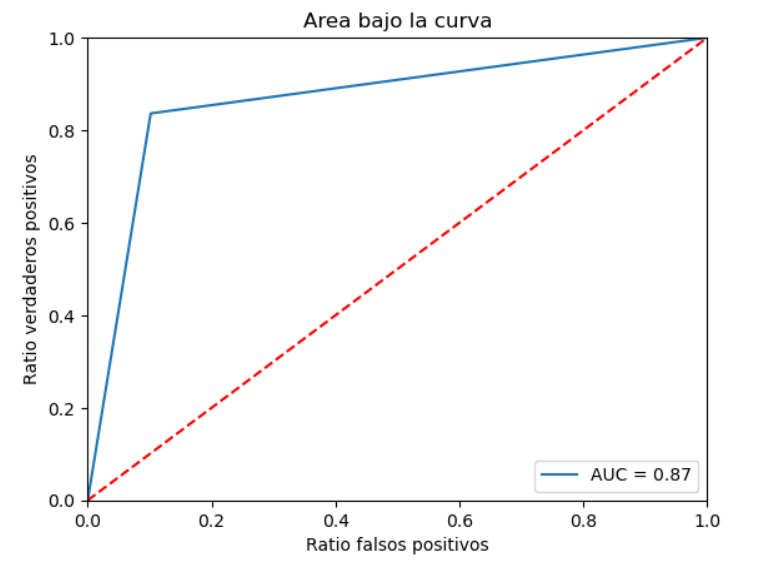
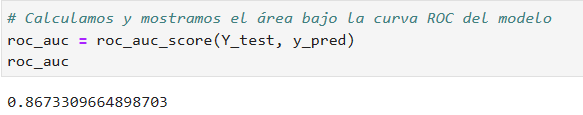
Figura 14: matriz de confusión del modelo de regresión logística

* Accuracy



*Figura 15: resultado del accuracy del modelo de regresión logística*

* Curva ROC



*Figuras 16 y 17: resultado del ROC y su área en el modelo de regresión logística*

**Otros análisis:**

Se ha documentado el modelo que finalmente se propone, pero se han analizado más variables que al final se ha decidido descartar por diversos motivos.

* Tabla Campañas: Se ha analizado la posibilidad de añadir el objetivo de la campaña a las diferentes variables de la campaña existentes en las tablas: campaña de entrada como socio, campaña de entrada como donante recurrente, campaña actual de donante recurrente, etc. Pero este objetivo no tenía muchas opciones concentrándose casi todos los valores en un mismo valor, por lo que se decidió descartarlas.
* Tabla modificaciones de cuota, además de las variables ya comentadas, se ha analizado la posibilidad de incluir en el modelo variables como numero de aumentos de cuota en el pasado, numero de disminuciones de cuota, también los importes de estas modificaciones pasadas. Se probaron modelos con ellas y no mejoraban los modelos planteados sin ellas, así que al ser variables difíciles crear y que no aportaban mejoría en la predicción decidieron descartarse.
* Tabla donaciones recurrentes, se planteó realizar un campo de meses desde inicio de la donación, utilizando la fecha de inicio de la cuota contra cada una de las fechas desde 2015 a 2021. Pero de la misma manera que la anterior, los modelos con esta variable no mejoraban predicción al modelo. Por lo que se ha intentado mantener pocas las variables que no sean directas de las tablas existentes para que sea entendible y fácil de modificar.
* Tabla de contactos: En esta tabla el principal problema para el modelo planteado es que las variables son a última fecha, lo que, ante el planteamiento de uso de información a ventanas históricas estáticas, no permite usar algunas de las variables disponibles.

Al igual que en el caso del primer modelo, también registramos los experimentos realizados con MLFlow, y promocionamos el mejor modelo al staging de producción para poder después desplegarlo en un servidor e invocarlo desde, por ejemplo, una aplicación, un api, etc.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# PLAN DE TRABAJO

## Reunión con MSF para planteamiento del problema

Podemos afirmar que el plan de trabajo del proyecto comienza con la reunión mantenida con el equipo de MSF para que nos explicaran en contenido de los conjuntos de datos que nos iban a proporcionar y las relaciones existentes entre estos. Esta reunión tuvo lugar en el 27 de junio y en ella se pudo recabar información para entender de objetivo y la utilidad del proyecto para MSF.

## Disponibilidad de los datos

El 26 de agosto los datos estuvieron disponibles en los buckets de AWS S3 de nuestros laboratorios, lo cual permitió comenzar a realizar las primeras tareas de exploración y comenzar a profundizar en el conocimiento de la información.

## Tratamiento y limpieza de datos

Durante el mes de septiembre se profundiza en las tareas del análisis exploratorio de los datos, identificando los datos de cada Dataset que nos podrían aportar información para el problema concreto que abordamos, cuales no tendrían relevancia y cuales nos servirían de filtro para eliminar aquella información que no tuviera que ver con el objetivo a conseguir.

Esta tarea no es independiente de la siguiente que se va a comentar por cuanto muchas de las variables que se eligen dependen también del planteamiento que se haya realizado para resolver el problema.

Durante este paso, se ha tenido que profundizar en cada tabla y variable para entender el sentido de las mismas y como se relacionaban entre ellas, ya que sin este entendimiento el planteamiento de un dataset inicial hubiera sido muy dificil.

La calidad de la base de datos, sobre todo relacionada con la tabla Contactos ha hecho que se pierda tiempo e información que podría ser relevante pero que hace muy difícil su uso, como puede ser la gran cantidad de nulos en la fecha de nacimiento, o los más de 1000 valores posibles para el campo de provincia.

## Selección de principales Driver y segmentaciones de la muestra

Durante el mes de septiembre y parte de octubre se establecen dos planteamientos para sendos modelos, uno que parte del conjunto de datos de Tareas o Task que recoge las interacciones que se han tenido con los socios y que se toma como tabla maestra por decirlo de alguna forma. El otro parte en su lugar de la tabla de donaciones recurrentes.

## Desarrollo de los modelos de predicción, análisis de resultados

Durante las primeras semanas de octubre, se llevaron a cabo los experimentos para construir, entrenar y evaluar los modelos implementados con Redes Neuronales Profundas y Regresión Logística, utilizando los conjuntos de datos de características obtenidos en etapas anteriores del proyecto. Estos experimentos son registrados con MLFlow para tener disponible el histórico de los trabajos realizados y el almacenamiento de los modelos para su posterior utilización. También se comienza a trabajar en el documento de la memoria del proyecto. Se plantearon dos modelos diferentes, siendo ambos tratados en este documento.

## Elaboración de la memoria y Documentación del proyecto

Aunque durante las semanas anteriores ya se ha ido esbozando el documento es en la tercera semana de octubre cuando se vuelcan los esfuerzos en la redacción de la memoria dada la fecha de entrega establecida (22/10/2023)

## Defensa del proyecto

La defensa del proyecto está prevista para el primer fin de semana de noviembre.

# RESULTADOS

## Modelo 1: Red Neuronal Profunda:

A continuación, se comentan los resultados obtenidos tras los distintos experimentos realizados con los modelos de tipo DNN, centrándonos en el que mejores métricas a alcanzado de todos ellos.

## Datasets Generados

Derivados de las fases de análisis exploratorio, selección y preprocesamiento de los datos se han obtenido diversos Datasets. Estos son los siguientes:

* **Inicial**: Contiene los datos resultantes del Filtrado, Join de ficheros y Selección de características de cada uno de los ficheros procesados.
* **Procesado:** En este caso ya se han hecho las transformaciones y generación de columnas calculadas necesarias y borrado las columnas que ya no son necesarias. Este dataset contiene el id del socio y la variable objetivo, que se omitirán para usarlo como fichero de entrenamiento del modelo. La variable objetivo se separará para actuar como el target del modelo.
* **Feature Store:** Para obtener este conjunto de datos se selecciona la fila con la fecha de actividad más reciente de cada uno de los ids de socio, de forma que tendremos la versión más actual de los datos de cada socio junto con sus características ya procesadas y listas para realizar predicciones sobre las mismas.
* **Predicciones:** Por último, y derivado de realizar las predicciones sobre feature store, tendremos un dataset conteniendo únicamente dos columnas, una con los ids de socio y otra con la predicción correspondiente a cada id.

## Modelos y rendimiento

El modelo que mejor rendimiento ha obtenido ha sido una red neuronal con los siguientes hiperparámetros:

|  |  |
| --- | --- |
| **Hiperparámetro** | **Valor** |
| epochs | 100 |
| learning\_rate | 0.01 |
| n\_layers | 5 |
| Nodes | [32, 64, 32, 16, 8] |
| Optimizer | 'adam' |
| Los | 'binary\_crossentropy' |

Durante el entrenamiento también se utilizaron los Callbacks “*early\_stop”* y “*reduce\_lr”* descritos en un punto anterior. Las métricas obtenidas se pueden ver en la siguiente captura de MLFlow.



Figura 18: Métricas resultantes del modelo DNN

Los valores en el conjunto de validación son muy similares a los del conjunto de entrenamiento, por lo que vemos que no hemos tenido “overfitting” si bien es cierto que resulta llamativo, en principio, que sean tan parecidos. En el gráfico que muestra la función de pérdida (loss) en función de las épocas vemos que esta va descendiendo gradualmente tanto para el conjunto de entrenamiento como para el de test y ambas se mantienen cerca una de la otra a pesar del comportamiento errático de la pérdida de validación en algún momento.

Gráfico, Histograma

Descripción generada automáticamente

Figura 19: Disminución de la función de pérdida durante el entrenamiento

También hemos obtenido la gráfica del área bajo la curva ROC para verificar visualmente que los resultados obtenidos difieren al menos de forma significativa del puro azar.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Figura 20: Gráfica de la curva ROC

Algunos de los experimentos se realizaron con datos normalizados pero no ofrecieron, aparentemente, resultados mejores que los que utilizaban los datos sin normalizar.

## Modelo 2: Regresión Logística:

A continuación, se comentan los resultados obtenidos tras los distintos experimentos realizados con diferentes modelos de clasificación, centrándonos en el que mejores métricas a alcanzado.

## Datasets Generados

Derivados de las fases de análisis exploratorio, selección y preprocesamiento de los datos se han obtenido diversos Datasets. Estos son los siguientes:

* **Inicial**: Contiene los datos resultantes del Filtrado, Join de ficheros y Selección de características de cada uno de los ficheros procesados, lo hemos llamado df\_union.
* **Procesado:** En este apartado tendremos dos, uno con las transformaciones y generación de columnas calculadas necesarias y borrado las columnas que ya no son necesarias (hacemos referencia el dataframe df\_inicial) y por otro lado se crea otro como resultado de aplicar oversampling a nuestro conjunto desbalanceado (df\_balanceada)
* **Predicciones:** Por último, y derivado de realizar las predicciones , tendremos un dataset conteniendo únicamente dos columnas, una con los ids de socio y otra con la predicción correspondiente a cada id.

## Modelo y rendimiento

Tal y cómo aparecen recogidas en el experimento de MLFlow las métricas obtenidas con este segundo modelo se pueden observar en la siguiente figura:



La curva ROC del modelo la podemos ver también reflejada en el siguiente gráfico:Gráfico, Gráfico de líneas

Descripción generada automáticamente

## Aplicación

Para presentar visualmente las predicciones hechas por el modelo sobre los Datasets, así como permitir la interacción con el mismo de forma on-line se ha construido una aplicación web con Streamlit que será descrita en el aparatado “Prototipo”.

# IMPLANTACIÓN, MONETIZACIÓN Y RETORNO DEL PROYECTO

Cómo ya se ha comentado en puntos anteriores, los modelos resultante obtenidos podrían ser utilizado por MSF para tratar de optimizar sus procesos de telemarketing o captación de fondos por la vía de aumentar las cuotas de socios. Ello repercutiría en una mayor eficiencia a la hora de utilizar los recursos y minimizar los costes en Call Center y la presión sobre los socios.

En nuestro caso los modelos se ha probado de forma local pero para poder explotarlos en un entorno de producción con todas las garantías entendemos que podrían desplegarse en cualquiera de las plataformas en Cloud habilitadas para ello, como AWS Sagemaker, Azure ML, Google Vertex AI, etc o, también en una instalación On-Premise de un servidor de MLFlow (o herramienta similar) en su propio data center.

Cómo se comentó en la introducción seguramente la mejor forma de comprobar si con estos modelos podemos optimizar el proceso de aumento de cuotas de socio es por la vía de comparar los resultados de dos campañas similares que tengan dicho objetivo en la que una es asistida por el modelo y la otra no. Si hubiera una diferencia significativa entre ambas en favor de la que utiliza el modelo tendríamos la constatación de su aportación.

# CONCLUSIONES Y TRABAJO FUTURO

## Resumen de los hallazgos

De lo visto en el apartado de resultados, la sensación obtenida tras el trabajo es que los datos de los que dispone MSF tienen el potencial de optimizar los procesos relacionados con la captación de nuevos fondos por la vía del aumento de cuotas de socios por los distintos canales de interacción de los que disponen. También pueden proporcionar una mayor perspectiva sobre la cantidad global resultante de aumentos de cuota que puede esperarse en un determinado año.

Es cierto que los resultados de los modelos obtenidos son modestos pero, en todo caso, algo mejores que el puro azar, lo que significa una ganancia en cualquier caso. Esto sugiere que, empleando más tiempo y recursos los modelos podrían mejorarse y obtener resultados, ahora sí, más significativos.

El trabajo ha tenido un enfoque principalmente de ciencia de datos ya que no era necesario hacer una labor previa de ingesta (los ficheros estaban proporcionados directamente por MSF y almacenados en AWS S3) y el objetivo obtener modelos predictivos que una vez entrenados, eso sí, será necesario desplegar, si bien, como hemos visto en puntos anteriores el framework de despliegue usado ha sido MLFlow en entorno local. Sí hemos trabajado con muchas herramientas vistas durante el Master como son AWS S3, Glue, Athena, Spark, Pandas, Scikit Learn, Tensorflow, MLFLow etc, por lo que sí que los trabajos realizados se corresponden en gran medida con los contenidos del master.

## Limitaciones

Quizás la principal limitación que hemos tenido para el estudio es el tiempo, ya que la tarea de encontrar el mejor modelo requiere de un trabajo iterativo continuo en el que se prueba, no sólo distintos tipos de modelos e hiperparámetros, sino que también se añaden o quitan features y cada una de ellas, a su vez, puede requerir de distintos tipos de preprocesamiento. Otra dificultad es la naturaleza temporal de los laboratorios de AWS que si haces un uso intensivo de los mismos te dejan sin saldo con rapidez y te ves obligado a trasladar los archivos, notebooks etc a otro lugar para poder trabajar con ellos en el siguiente laboratorio. Por la misma razón a veces prefieres hacer procesamientos en la máquina local y no hacer uso de algunos de los servicios de AWS por no reducir el saldo y no tener que reconfigurar los servicios si se acaba.

## Posibles mejoras y trabajos futuros

Entre las cosas que se han quedado en el tintero por falta de tiempo o porque implican una mayor complejidad podríamos mencionar los siguientes:

Si bien en alguno de los ficheros de MSF hay alguna variable que indica si determinada campaña se realizó en un momento de emergencia mediática en lo que respecta a las tareas o interacciones con el usuario no hemos visto que dicha información figure actualizada para la fecha de cada una de estas interacciones. Es por ello qué sería interesante poder enriquecer el dataset con ese indicador, el cual se podría obtener de algún sistema de análisis de sentimiento que recorriera las webs de los principales medios de comunicación. Entendemos que es una variable muy relevante y que sería muy positivo poder incorporarla.

También surgen nuevas ideas para modelos más estratégicos, como por ejemplo realizar un estudio más específico sobre los socios que aún no han hecho ningún aumento de cuota.

En el modelo finalmente obtenido en el proyecto no se ha utilizado la tabla de oportunidades de la que seguramente también se puede añadir información que pueda resultar útil. Es por ello que en una eventual segunda fase se podría cruzar esta tabla con los conjuntos de datos obtenidos para añadir características que puedan ser relevantes.

Una tarea que quedó también pendiente es hacer el entrenamiento y despliegue de los modelos en el servicio AWS Sagemaker. No se ha llegado a hacer por un lado por la sencillez que aporta MLFLow trabajando en local y, por otra, por lo comentado en el punto de limitaciones por cuanto podía implicar disminuir rápidamente el tiempo de vigencia del laboratorio. Es cierto que podíamos haber simplemente desplegado el modelo en un endpoint de Sagemaker pero una vez comenzado el trabajo con MLFLow ya perdía un poco el sentido.

Otra parte interesante que no se ha podido explorar por falta de tiempo es la correlación de los aumentos de cuotas con las situaciones económicas. Ya que, en principio, pueda tener sentido que ante una situación de bonanza económica generalizada la población esté dispuesta a ser más generosa que en otros momentos donde todos sufrimos y tememos las consecuencias que pueda tener la crisis económica en nuestro propio día a día. Por ejemplo, en el momento actual, ante un escenario inflacionista, pueda parecer lógico que las personas se encuentren cada vez en situaciones más ajustadas en su economía personal y por lo tanto sean menos propensas a realizar un aumento en su cuota, o incluso piensen en bajarla.

# INFORMES DE ANÁLISIS EXPLORATORIO Y EXPERIMENTOS DE MODELADO

Los procesos descritos durante el presente documento se han realizado fundamente por la vía de ejecución de Notebooks dado que el proyecto tenía un enfoque principalmente de ciencia de datos. Dichos notebooks se han ejecutado tanto en herramientas en Cloud como AWS Glue interactive sessions, como con Jupyter, Jupyterlab en local. También se han utilizado herramientas como Sweetviz para realizar un profiling de los datos. A continuación se adjuntan en los Reports y Notebooks obtenidos renderizados en HTML.

## Análisis preliminar tablas aportadas por MSF



## Profiling



## Preprocesado y limpieza



## Construcción Modelos



# PROTOTIPO

Para permitir la visualización de las predicciones realizadas sobre el almacén de características, así como la posibilidad de poder realizar predicciones llamando directamente al modelo hemos desarrollado la aplicación muy sencilla, embebiendo el modelo dentro de la propia aplicación Streamlit.

Diagrama

Descripción generada automáticamente

Figura 21: Aplicación Streamlit con modelo Embebido

El prototipo consta de un único archivo .py (app\_msf\_embedded\_layered.py) con su fichero de requirements.txt para poder crear un entorno de ejecución propio con sus dependencias. También contiene una carpeta data en la que se almacenarán un fichero con las features calculadas para cada socio (id + features) así como otro fichero con las predicciones ya obtenidas para cada uno de los ids de socio (id + predicción).

El modelo se carga al comienzo de la aplicación obteniéndolo de MLFLow:

Texto

Descripción generada automáticamente

Figura 22: Carga del modelo en la aplicación

También se cargan el feature store y el fichero con las predicciones por Id. Con el primero se podrá realizar la predicción on-line con el modelo para cada id de socio elegido. El widget constará de un campo de entrada de texto en el que poder teclear el Id del socio y un botón que le permitirá hacer la predicción de las posibilidades de aumento de cuota del mismo. El aspecto de esta parte del código será el siguiente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 23: Widget para predicción sobre socio

El fichero con los ids de socio y las predicciones se utiliza en el prototipo para mostrar, de entrada, una lista o ranking con los veinte socios más propensos a aumentar su cuota según el valor de la predicción. Se podrá, no obstante, realizar una búsqueda sobre todo el conjunto de datos (no sólo los veinte mencionados) que permita localizar cualquier socio que sea de interés.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 24: Lista de socios con más probabilidad de aumentar la cuota según el modelo

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Figura 25: Filtro para la lista de socios

El prototipo también incluye un pequeño gráfico de barras en el que se muestran el número de socios que hay en distintos rangos de probabilidad según el valor de su predicción.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Figura 26: Widget con gráfica de número de socios por rango de probabilidad de aumentar cuota

El aspecto final componiendo los diferentes widget en dos columnas y añadiendo el logo sería el que se muestra a continuación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 27: Web de la aplicación

El código de la aplicación está en el siguiente fichero:



# CÓDIGO FUENTE Y BIBLIOGRAFÍA

El código fuente del proyecto (Notebooks, Streamlit, Reports) se puede encontrar en el siguiente repositorio de GitHub: <https://github.com/mlalandag/mbit_msf>

En cuanto a la bibliografía se han utilizado principalmente todos los recursos proporcionados por los profesores del Máster así como la inestimable ayuda de ChatGPT para plantearle consultas sobre operaciones con Pyspark o Pandas que resultaba más sencillo preguntarle que buscar entre los recursos mencionados.