

**MBIT SCHOOL**

**Trabajo de Fin de Master**

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 MCD-Oct22

Máster en Ingeniería de Datos MDE-Oct22

Contenido

[RESUMEN EJECUTIVO 4](#_Toc148625142)

[INTRODUCCIÓN, ANTECEDENTES Y MOTIVACIÓN 5](#_Toc148625143)

[ARQUITECTURA DE LA SOLUCIÓN 7](#_Toc148625144)

[Visión general del proceso y herramientas utilizadas 7](#_Toc148625145)

[Análisis Exploratorio de la Información Recibida 10](#_Toc148625146)

[Tabla de donaciones recurrentes 10](#_Toc148625147)

[Tabla de modificaciones de cuota 11](#_Toc148625148)

[Tabla de Contactos 11](#_Toc148625149)

[Tabla de Campañas 11](#_Toc148625150)

[Tabla de Oportunidades 12](#_Toc148625151)

[Tabla de Tareas 12](#_Toc148625152)

[Modelo basado en tabla de interacciones con los socios 12](#_Toc148625153)

[Selección de características 15](#_Toc148625154)

[Procesamiento y limpieza de datos 17](#_Toc148625155)

[Almacén de características (Feature Store) 21](#_Toc148625156)

[Elección y construcción del modelo 21](#_Toc148625157)

[Despliegue con MLFlow 24](#_Toc148625158)

[PLAN DE TRABAJO 26](#_Toc148625159)

[Reunión con MSF para planteamiento del problema 26](#_Toc148625160)

[Disponibilidad de los datos 26](#_Toc148625161)

[Tratamiento y limpieza de datos 26](#_Toc148625162)

[Selección de principales driver y segmentaciones de la muestra 27](#_Toc148625163)

[Desarrollo del modelo de predicción, análisis de resultados 27](#_Toc148625164)

[Elaboración de la memoria y Documentación del proyecto 27](#_Toc148625165)

[Defensa del proyecto 27](#_Toc148625166)

[RESULTADOS 28](#_Toc148625167)

[Datasets Generados 28](#_Toc148625168)

[Modelos y rendimiento 28](#_Toc148625169)

[Aplicación 30](#_Toc148625170)

[IMPLANTACIÓN, MONETIZACIÓN Y RETORNO DEL PROYECTO 31](#_Toc148625171)

[CONCLUSIONES Y TRABAJO FUTURO 32](#_Toc148625172)

[Resumen de los hallazgos 32](#_Toc148625173)

[Limitaciones 32](#_Toc148625174)

[Posibles mejoras y trabajos futuros 33](#_Toc148625175)

[Modelo basado en tabla de donaciones recurrentes 34](#_Toc148625176)

[INFORMES DE ANÁLISIS EXPLORATORIO Y EXPERIMENTOS DE MODELADO 38](#_Toc148625177)

[PROTOTIPO 39](#_Toc148625178)

[CÓDIGO FUENTE Y BIBLIOGRAFÍA 44](#_Toc148625179)

#### Tabla de Ilustraciones

[Figura 1: Esquema general de la arquitectura utilizada 8](#_Toc148625180)

[Figura 2: Ciclo de Vida con MLFlow 10](#_Toc148625181)

[Figura 3: Precompute Serving Pattern 10](#_Toc148625182)

[Figura 4: Profiling de la variable msf\_closetype\_\_c de la tabla Task obtenido con Sweetviz 14](#_Toc148625183)

[Figura 5: Muestra tabla Task 14](#_Toc148625184)

[Figura 6: Muestra tabla Quota Modification 15](#_Toc148625185)

[Figura 7: Resultado del Join de Task y Quota Modification con campos a NaN 20](#_Toc148625186)

[Figura 8: Experimentos en MLFlow 25](#_Toc148625187)

[Figura 9: Modelos en MLFlow 26](#_Toc148625188)

[Figura 10: Métricas resultantes del modelo 30](#_Toc148625189)

[Figura 11: Disminución de la función de pérdida durante el entrenamiento 30](#_Toc148625190)

[Figura 12: Gráfica de la curva ROC 31](#_Toc148625191)

[Figura 13: Aplicación Streamlit con modelo Embebido 40](#_Toc148625192)

[Figura 14: Carga del modelo en la aplicación 40](#_Toc148625193)

[Figura 15: Widget para predicción sobre socio 41](#_Toc148625194)

[Figura 16: Lista de socios con más probabilidad de aumentar la cuota según el modelo 42](#_Toc148625195)

[Figura 17: Filtro para la lista de socios 42](#_Toc148625196)

[Figura 18: Widget con gráfica de número de socios por rango de probabilidad de aumentar cuota 43](#_Toc148625197)

[Figura 19: Web de la aplicación 44](#_Toc148625198)

# 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 implementando un modelo predictivo con una red neuronal profunda.

**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 las que obtener tanto un fichero para entrenamiento y validación como un almacén de características que se utilizarán como entradas para un modelo implementado con una Red Neuronal Profunda (DNN) usando Tensorflow. Con éste se realizan las predicciones y se publicarán en una pequeña página web desarrollada 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 juego de datos para entrenamiento y validación con una calidad que entendemos prometedora de cara a conseguir el objetivo planteado en la hipótesis. Tras probar con numerosas combinaciones de hiperparámetros, el mejor modelo obtenido arrojaba unas métricas de “accuracy”, “precisión” y “recall” que sugieren un cierto margen de optimización de los procesos destinados a tratar de aumentar la cuota de los socios. El modelo se pudo productivizar sin problemas usando el framework MLFlow y embeberlo en una pequeña aplicación Streamlit.

# INTRODUCCIÓN, ANTECEDENTES Y MOTIVACIÓN

En este proyecto se va a desarrollar un modelo con el que poder predecir la probabilidad de aumento de cuota de los socios donantes recurrentes. De esta manera se podrá saber cuáles de ellos serán más propensos que otros, y realizar las tareas necesarias sobre las personas estadísticamente más propensas, consiguiendo que las campañas sean más eficaces y no sobrecargando de información y peticiones a aquellos que apenas tenga probabilidades de aumento de cuotas, mejorando se esa manera también la interacción con los socios.

Para conseguir este objetivo, se va a plantear la creación de un modelo que permita distinguir cuales de los socios actuales tiene más probabilidades de realizar un aumento de cuota, basándose en la información histórica disponible hasta este momento.

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 ha desarrollado este modelo, y los pasos, decisiones que se han ido siguiendo a lo hora de llevar a cabo este proyecto.

Como hemos comentado el enfoque es concentrar el estudio en los socios que tengan aportaciones periódicas activas y tratar de predecir su comportamiento a la hora de elevar su contribución periódica ante un eventual contacto por alguno de los canales de comunicación (Telemarketing, Persona a Persona, Web, etc) de los que dispone MSF. Es un enfoque más estratégico en el sentido que no trata globalmente el Dataset para predecir una cantidad global debida a todos los aumentos de cuota que puedan producirse en un futuro inmediato concreto, si no que trata de averiguar la probabilidad individual por socio de que aumentara su cuota si fuera contactado en este momento. No obstante, una vez obtenidas las probabilidades individuales es posible extrapolar y obtener información global utilizando un umbral para determinar el número de socios con más probabilidad de incrementar su cuota y, calculando los aumentos medios de cuota realizados en el pasado cada uno de ellos, realizar una estimación del importe global que supondría para todos los socios.

La principal posibilidad que se abre con este planteamiento es la de optimizar los procesos de captación de fondos por la vía de aumento de cuotas que se hagan vía campañas de telemarketing o similares.

Una vez desarrollado un modelo que se considere que aporta cierto valor, una de las métricas para evaluar su comportamiento por parte de negocio podría ser, 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.

# 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 : 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 del modelo. Este es desarrollado con Tensorflow y el tipo de algoritmo que se ha elegido para la tarea es una red neuronal profunda (Deep Neural Network o DNN) porque entendemos que es tipo de modelo que puede 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.

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 : 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 : 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.

### Tabla de donaciones recurrentes

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 21 campos
* Tiene 1.198.207 registros.
* No existen duplicados

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

### 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 16 campos
* Tiene 2.003.019 registros.
* No existen duplicados

### 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

### 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 XXXXX campos
* Tiene XXXXXX registros.
* No existen duplicados

### 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 XXXXX campos
* Tiene XXXXXX registros.
* No existen duplicados

## Modelo basado en tabla de interacciones con los socios

El planteamiento adoptado para este segundo 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 : 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 : 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 : 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\_\_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\_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\_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" |
|  |  |
|  |  |
|  |  |
|  |  |

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 : 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 : 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 : 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.

# 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.

## 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 del modelo de predicción, análisis de resultados

Durante las primeras semanas del mes de octubre se realizan los experimentos de construcción, entrenamiento y evaluación de los modelos implementados con Redes Neuronales Profundas con los Datasets de features obtenidos en las etapas anteriores. 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. Inicialmente se plantearon dos modelos diferentes, uno de los cuales se completó y el otro quedó pendiente de llevar a cabo por problemas de tiempo. Éste último se describirá en la sección de trabajo futuro.

## 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

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.

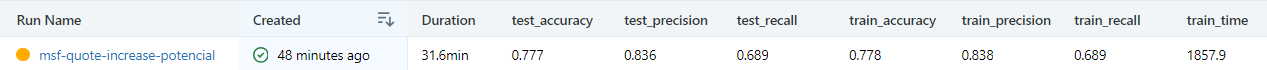


Figura : Métricas resultantes del modelo

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 : 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 : 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.

## 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, el modelo resultante obtenido podría 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 el modelo se ha probado de forma local pero para poder explotarlo en un entorno de producción con todas las garantías entendemos que podría 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 este modelo 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.

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 un modelo predictivo que una vez entrenado eso sí era necesario desplegar, si bien como hemos visto en puntos anteriores el framework de despliegue usado ha sido MLFlow en entorno local. Si 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 las 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.

### Modelo basado en tabla de donaciones recurrentes

Aparte del planteamiento seguido en el proyecto para el modelo implementado también barajamos otro basado en la creación de tablas de donantes anuales, concretamente en los cierres de cada año a partir de 2015. Una vez se tuviera estas tablas anuales con los donantes activos en ese momento, se crearía el Flag de aumento de cuota en caso de que durante el siguiente año haya o no realizado un aumento en la cuota. De esta manera se pretende conseguir dar el mismo intervalo de tiempo a todos los donantes recurrentes en la previsión, y de esta manera saber que la predicción de aumento en el modelo es en ese mismo intervalo de tiempo. Es decir, que este Flag, será la variable objetivo en el modelo de predicción a desarrollar en los siguientes pasos del proyecto.

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. Para la explicación se pondrá como ejemplo el 2015. Pero se realizará con todos los años desde 2015 hasta 2021.
* 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.
* De este filtro, 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 baja (msf\_CancelationDate\_\_c) informada (no vacía) que sea menor o igual a 31 de diciembre de 2015.
* 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”.
* El siguiente paso es crear el flag de aumento de cuota (“*FLAG\_AUMENTO\_CUOTA”*) 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 informacion de la tabla “*Modificación Aumentos*” creada en el paso anterior, se generará esta nueva variable “*FLAG\_AUMENTO\_CUOTA”* 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.

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.

* A esta tabla de información básica, se le añade información disponible tanto en estas como en el resto de las tablas, que serán las variables input al modelo. De esta manera, tras haber comentado en el paso anterior, cuales de las variables tienen tanto calidad como potencial para el modelo, se van a incorporar:
  1. De la tabla Donaciones Recurrentes:
     1. Cuota anualizada “*msf\_AnnualizedQuota\_\_c*”.
     2. Canal de entrada inicial “*msf\_LeadSource1\_\_c*”.
     3. Canal de entrada actual “*msf\_CurrentLeadSource1\_\_c*”
     4. Periodicidad “*npe03\_\_Installment\_Period\_\_c*”.
     5. Campaña inicial “*npe03\_\_Recurring\_Donation\_Campaign\_\_c*”.
     6. Se creará una variable denominada “*Age\_donacion*” que se calculará como la diferencia en MESES de la fecha de la tabla (31 de diciembre de 2015 en el ejemplo que se está explicando” y la variable fecha de alta de la donación “*npe03\_\_Date\_Established\_\_c*”.
  2. De la tabla modificación de cuota se van a añadir variables que no se utilizarán como input al modelo (ya que esta tabla es el objetivo) pero servirán para la unificación de informacion con otras tablas:
     1. Diferencia de importe anualizada “*msf\_changeannualizedquota\_\_c*”. Esta servirá para crear un valor de cuota anual previo al cambio. Ya que en la tabla de donaciones recurrentes solo está la última existente “*msf\_AnnualizedQuota\_\_c*”. Por lo que restándole a esta ultima el importe de la diferencia de cuota se podrá conocer el importe de la cuota inicial que se denomirará “quota\_initial” y que en caso de que no se haya modificado la cuota será igual a “*msf\_AnnualizedQuota\_\_c*” de la tabla de donaciones recurrentes.
     2. Fecha de modificación de cuota “*msf\_changedate\_\_c*”. Esta fecha no será input del modelo, pero se necesitará para unir tareas, oportunidades y campañas que sean anteriores a la fecha del cambio y que por lo tanto hayan podido influir en el aumento de la cuota.
  3. De la tabla Contactos: Como es una tabla que no guarda histórico, para todas las tablas anuales creadas al final del paso 5 se la añadirá la misma información sobre la información disponible en Contactos.
     1. Género “*Gender\_\_c*”.
     2. Lifetime value: “*msf\_LTVCont\_\_c*”.
     3. Provincia “*mailingstate*”.
     4. Queja por presión en captación de fondos “*msf\_PressureComplaint\_\_c*”.
     5. Scoring RFV de socio donante “*msf\_ScoringRFVRecurringDonor\_\_c*”
     6. Valor promedio de donante “*msf\_AverageDonorValue\_\_c*”.
     7. Se creará una variable denominada “*Age*” que se calculará como la diferencia en años de la fecha de la tabla (31 de diciembre de 2015 en el ejemplo que se está explicando” y la variable fecha de nacimiento de esta tabla de contactos “*msf\_birthyear\_\_c*”.
  4. De la tabla Oportunidad: Como en la tabla hay fechas sobre las acciones realizadas, se añadirán a las tablas anuales en función de las fechas de la oportunidad estén dentro del año de desempeño que se observa en cada una de ellas. Se centrará en unir con las oportunidades que sean sobre la donación recurrente. Por lo que la tabla total se filtrará con las oportunidades cuyo tipo “*Type*” sea “*Recurrent Donation*”. Una vez se tienen filtradas las oportunidades relacionadas con las donaciones recurrentes, en el año de observación (en el caso de ejemplo las oportunidades entre el 1 de enero de 2016 y 31 de diciembre 2016 o fecha de modificación de cuota si el donante la ha mejorado) se unirá:
     1. Programa al que pertenecía el contacto “msf\_Program\_\_c”
     2. Segmento colaborador. “msf\_MailingSegment\_\_c”
     3. Campaña inicial “CampaignId”. Solo se considera esta variable campaña, ya que la campaña de modificación solo estará disponible para los que hayan aumentado, lo que quiere decir que no va a ayudar a distinguir quien va a aumentar o no en el futuro, ya que es posterior al momento del aumento.

# 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 Jupyterlab en local. A continuación, adjuntamos estos notebooks renderizados en HTML o PDF

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

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Figura : 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 : 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 : Web de la aplicación

# 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.

# ANEXO A – ANÁLISIS EXPLORATORIO DETALLADO