Universidad ORT Uruguay
Facultad de Ingeniería

Anticipando la fuga de clientes: un caso real de una corporación de servicios para empresas

Entregado como requisito para la obtención del título de Máster en Big Data

Nazim Abisab – 278262

Yanina Lembo – 166877

Luis Oliari – 168379

Tutor: Alejandro Bianchi

2023

Declaración de autoría

Nosotros, Nazim Abisab, Yanina Lembo y Luis Oliari declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

1. La obra fue producida en su totalidad mientras realizábamos el Proyecto Final del Master en Big Data;
2. Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
3. Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
4. En la obra, hemos acusado recibo de las ayudas recibidas;
5. Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
6. Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Nazim Abisab	Yanina Lembo	Luis Oliari
30/03/2023	30/03/2023	30/03/2023

Dedicatoria

Este proyecto se lo dedicamos con mucha gratitud a todas las personas que nos han apoyado a lo largo de este proceso.

En especial a nuestras familias por estar presentes de forma incondicional en cada momento importante de nuestras vidas, que tanta paciencia nos han tenido y no nos han dejado nunca bajar los brazos.

Agradecimientos

Se dice que la mejor herencia que los padres pueden dejarnos son los estudios, legado por el cual estamos muy agradecido. Gracias a nuestros padres y familiares por habernos enseñado a nunca desistir, a luchar siempre por nuestros sueños, por estar siempre a nuestro lado, apoyándonos de manera incondicional.

Al ver el resultado logrado en este proyecto, solamente se nos ocurre una única palabra: ¡Gracias!

Gracias a nuestro tutor Alejandro Bianchi por su guía y consejos a lo largo del desarrollo de esta tesis. A la Universidad ORT por habernos permitido culminar esta gran etapa de nuestras vidas, por la orientación y apoyo a lo largo de toda la maestría.

Un agradecimiento especial a Ricoh LATAM y Dynamo Global por haber puesto a disposición los datos para el desarrollo de la tesis.

Gracias a nuestras empresas por habernos apoyado en el transcurso de la carrera y en particular en esta última etapa.

Finalmente queremos agradecerles a nuestros amigos y compañeros de trabajo, por sus palabras de aliento, compañía y ayuda en los momentos más difíciles.

Abstract

En toda empresa se cuenta como uno de los objetivos de negocio lograr la satisfacción del cliente, consolidar relaciones comerciales y evitar la fuga de estos. Con la ayuda de las tecnologías se pueden obtener estos objetivos de forma más ágil y eficiente.

Frente a esto, es que a lo largo de esta tesis se exponen conceptos teóricos de Machine Learning, la generación de algoritmos de aprendizaje supervisado y no supervisado que permiten predecir el comportamiento de los clientes, así como una agrupación de estos en base a características comunes, para una empresa real, que presta servicios a otras empresas (B2B).

En primer lugar, se realizó un entendimiento general de la empresa, la cual trabaja hoy en día principalmente con soluciones de Microsoft, tanto a nivel de ERP como CRM, de esta forma se decidió basar la arquitectura del proyecto en Microsoft Azure. Luego se investigaron técnicas de Machine Learning para abordar la problemática de fuga de clientes en una empresa de servicios, se desarrollaron y testearon algoritmos para predecir el *Churn*, se compararon los resultados de los diferentes algoritmos con la métrica de Curva de ROC. Luego de haber llegado a un algoritmo aceptable, se continúa aplicando técnicas de aprendizaje no supervisado para el Clustering de los clientes y en base a una combinación del *Churn* y el *cluster* de pertenencia, desarrollar conclusiones.

Se aplicó lenguaje de programación Python para los algoritmos de aprendizaje supervisado y no supervisado, *Power Query* para transformación inicial de variables del *dataset* y *Power BI* para la extracción de conclusiones de Clustering de clientes.

Los resultados obtenidos para la predicción fueron más que satisfactorios, alcanzando con el algoritmo Stacking un Cross Validation Score de 96.81% y un ROC Score de 90.05%. En el caso del Clustering, se crearon 3 grupos de clientes, número óptimo según el análisis realizado.

Como conclusión del trabajo, se realizaron recomendaciones a la empresa para mejorar el manejo y gobernanza de datos, la interpretación de resultados, las técnicas y algoritmos a aplicar.

Palabras clave

Chun Rate, Machine Learning, Clustering, aprendizaje supervisado, aprendizaje no supervisado, algoritmos estadísticos predictivos, Microsoft Azure.

Índice

1. Contexto empresarial 12

1.1. Evolución del negocio 12

2. Objetivos 14

2.1. Objetivos generales 14

2.2. Objetivos específicos 14

3. Marco teórico 16

3.1. Gestión de abandono de clientes (Churn Rate, fuga de clientes) 16

3.2. Machine Learning 20

3.3. Tipos de aprendizajes de Machine Learning 20

3.3.1 Aprendizaje supervisado 21

3.3.2 Aprendizaje no supervisado 21

3.4. Algoritmos de aprendizaje estadístico predictivo 21

3.4.1. Arboles de Decisión 22

3.4.2. Random Forest 25

3.4.3. Gradient Boosting Machine 28

3.4.4. Stacking 31

3.5. Ensayo del proyecto 33

3.5.1. Información de los datos 33

3.5.2. Análisis exploratorio de datos 34

3.5.3. Modelado 35

3.5.4. Conclusiones del ensayo 37

4. Desarrollo del proyecto 39

4.1. Arquitectura de la solución 39

4.1.1. Diagrama de la arquitectura elegida 40
4.1.2. Componentes 41
4.2. Dataset y problemática a resolver 43
4.2.1. Estructura del *dataset* 45
4.3. Análisis Exploratorio de los Datos 47
4.3.1. Carga y revisión de los datos 47
4.3.2. Análisis de variables 50
4.3.3. Transformaciones necesarias 53
4.4. Aprendizaje automático 55
4.4.1. Curva ROC 55
4.4.2. Particionado *Train/Test* 57
4.4.3. Modelos de Machine Learning 59
4.5. Predicción del Churn Rate 71
4.5.1. Planilla de resultados 72
4.6. Clustering de clientes 73
4.7. Conclusiones particulares 81
5. Análisis del resultado por predicción de Churn 91
6. Recomendaciones al cliente 94
7. Conclusiones 96
7.1. Lecciones aprendidas 96
7.2. Trabajo futuro 97
8. Glosario 98
9. Referencias bibliográficas 102
10. ANEXOS 109
10.1. Diagrama de Dataflow 109
10.2. Implementación en Microsoft Azure 110

Índice de Ilustraciones

Ilustración 1 - Conjuntos de datos para crear una vista de 360 de los clientes [1]..... 18
Ilustración 2 - Ciclo de retroalimentación [1] 19
Ilustración 3 - Tipos de aprendizaje automático [2] 20
Ilustración 4 - Diagrama de Árboles de Decisión..... 23
Ilustración 5 - Diagrama explicativo de XGBOOST [6] 29
Ilustración 6 - Comparación de rendimiento de modelos [6] 30
Ilustración 7 - Importación de librerías 33
Ilustración 8 - Lectura de los datos..... 34
Ilustración 9 - Matriz de confusión..... 36
Ilustración 10 - Vista esquemática de la arquitectura elegida y el flujo de datos..... 40
Ilustración 11 - Importación de librerías necesarias (1) 47
Ilustración 12 - Importación de librerías necesarias (2) 48
Ilustración 13 – Listado de tipos de variables 50
Ilustración 14 – Gráfico de distribución del Churn 51
Ilustración 15 – Gráfico de *outliers* de Equipos Arrendados, Incidentes y Reclamos ... 51
Ilustración 16 – Matriz de correlación entre variables 52
Ilustración 17 - Resultado gráfico sin valores nulos..... 54
Ilustración 18 – Aplicación de la técnica Smote 55
Ilustración 19 - Algoritmo de Árboles de Decisión..... 59
Ilustración 20 - Diagrama del Árboles de Decisión creado..... 60
Ilustración 21 – Curva ROC - Primer resultado - Árboles de Decisión 60
Ilustración 22 - Combinación de hiperparámetros - Árboles de Decisión 62
Ilustración 23 – Curva ROC - Segundo resultado - Árboles de Decisión 62
Ilustración 24 - Algoritmo de Random Forest..... 63
Ilustración 25 – Curva ROC - Primer resultado - Random Forest 63
Ilustración 26 - Combinación de hiperparámetros - Random Forest..... 64
Ilustración 27 - Modelo final de Random Forest..... 64
Ilustración 28 - Configuración del Modelo final de Random Forest..... 64
Ilustración 29 – Curva ROC - Segundo resultado - Random Forest 65

Ilustración 30 - Algoritmo Gradient Boosting Machine 66
Ilustración 31 – Curva ROC - Primer resultado - Gradient Boosting Machine 67
Ilustración 32 - Combinación de hiperparámetros - Gradient Boosting Machine..... 67
Ilustración 33 – Curva ROC - Segundo resultado - Gradient Boosting Machine 68
Ilustración 34 - Algoritmo LightGBM Classifier 68
Ilustración 35 – Curva ROC - Resultado LightGBM Classifier..... 69
Ilustración 36 – Curva ROC - Resultado Stacking 70
Ilustración 37 - Carga del conjunto de datos previo en una variable 71
Ilustración 38 – Ejecución del comando “predict” 71
Ilustración 39 - Carga de predicción en el conjunto de datos..... 71
Ilustración 40 - Almacenamiento del conjunto con predicción en un archivo ".csv" 71
Ilustración 41 - Importación librerías para Clustering..... 74
Ilustración 42 - Carga de datos en *dataframe* de pandas 74
Ilustración 43 - Codificación de variables categóricas 75
Ilustración 44 - Normalización de variables con Standardscaler 75
Ilustración 45 - K óptimo método del codo 76
Ilustración 46 - K óptimo método de Silhouette 77
Ilustración 47 - K-Means con K óptimo 77
Ilustración 48 - Etiqueta *cluster* en conjunto de datos original 78
Ilustración 49 - Clustering – Cantidad Equipos Arrendados vs Tiempo de Respuesta.. 78
Ilustración 50 – Cantidad Equipos Arrendados vs Tiempo de Respuesta y Categoría de facturación 79
Ilustración 51 - Resultado Clustering – Sparse Matrix 80
Ilustración 52 - Almacenamiento del resultado de Clustering en ".csv" 80
Ilustración 53 – Gráfico Total facturado por categoría de facturación..... 81
Ilustración 54 - Tasa de fuga de clientes 83
Ilustración 55 - Estructura por Churn por facturación y cliente 84
Ilustración 56 - Churn por *cluster*, facturación y clientes 84
Ilustración 57 - Cantidad clientes del *cluster* 2 85
Ilustración 58 - Churn por c*luster*, tipo de empresa y país de venta por facturación 86
Ilustración 59 - Churn por *cluster*, tipo de empresa y país de venta por clientes..... 87
Ilustración 60 - Pasos para la aplicación de un modelo de predicción de Churn 91

Índice de Tablas

Tabla 1 - Tabla comparativa de algoritmos Bagging, Boosting y Stacking [7] 32
Tabla 2 - Comparación de performance de Algoritmos 37
Tabla 3 - Particionado *Train* y *Test* 57
Tabla 4 - Comparación de resultados de pruebas realizadas 58
Tabla 5 – Planilla de resultados obtenidos 72
Tabla 6 - Facturación por predicción de Churn 82
Tabla 7 - Facturación por Churn 82
Tabla 8 - Predicción facturación de clientes que se fugaron 83
Tabla 9 - Situación general de clientes fugados 88
Tabla 10 - Situación en México de clientes fugados 89
Tabla 11 - Situación en Chile de clientes fugados 90
Tabla 12 - Resumen facturación por Churn y Predicción de Churn 92
Tabla 14 - Top 10 clientes con mayor facturación mensual con predicción del Churn . 93

1. Contexto empresarial

Ricoh Latin America (Ricoh LATAM) es una empresa ubicada en Weston, Florida, subsidiaria propiedad de Ricoh Latin America Inc.

La Sociedad es el centro de operaciones regional de atención al cliente, planeamiento, logística, configuración, transporte regional y distribución de equipos digitales y soluciones para el procesamiento de imágenes y documentos.

Ricoh LATAM se enfoca en una forma de trabajo ágil e inteligente a través de:

- Soluciones de flujo de trabajo de documentos.
- Equipo de impresión y de imagen para oficinas.
- Tecnologías de colaboración audiovisual.
- Soluciones de impresión de producción.
- Servicios de TI y soporte técnico.
- Soluciones específicas para la asistencia sanitaria, legal, educación superior y otras industrias.

1.1. Evolución del negocio

Ricoh LATAM comenzó su negocio siendo una empresa que ofrecía productos, servicios y soluciones, "*Commercial & Industrial Printing*", proponiendo soluciones de impresión para el mundo corporativo (Corporaciones, Empresas de ventas al por menor, Educación o Medianas Empresas, Editoriales y Centros de Copiado) e impresiones sobre prendas de vestir. También ha implementado Oficinas Inteligentes, en donde además del servicio tradicional de impresión, los dispositivos ofrecidos permiten acceder a sus documentos en la nube y compartirlos como, donde y cuando quiera, dando la posibilidad de impulsar la movilidad y trabajo en equipo. A su vez, ofrecen impresiones móviles, software para la administración de documentos (capturar, convertir, indexar y dirigir el contenido a flujos de trabajo electrónicos). En lo que respecta a suministros y accesorios, brindan

tinta, tóner, piezas y mantenimiento de equipos. Estas líneas de negocios son las que le permitieron a Ricoh LATAM consolidarse en el mercado y ser hoy una marca conocida y confiable.

Actualmente las actividades principales de Ricoh LATAM consisten en:

1. Adquisición y posterior comercialización de fotocopiadoras electroestáticas, duplicadores, faxes, equipos de videoconferencias, proyectores, entre otros, así como también abastece los accesorios, suministros y partes para su correcto funcionamiento y gestiona la logística del transporte regional.
2. Servicio de soporte (suministros y servicio técnico) para clientes finales Ricoh LATAM (*contact center* incorporado en el año 2018).
3. A partir del segundo semestre del año 2020 se comenzó con la centralización de otras actividades en Uruguay al incorporar un *Shared Service* Center de finanzas.
4. A comienzos de enero 2021 se comenzó a prestar servicio de soporte para clientes finales Ricoh en USA, por cuenta y orden de Ricoh USA Inc.
5. Durante el año 2022 se fueron incorporando nuevos servicios de *offshoring* para compañías Ricoh fuera de LATAM.
6. En 2022, Ricoh se embarcó en el proyecto de implementación de Microsoft D365 Finance & Operations, para sustituir su ERP existente (Microsoft AX 12). A su vez ya cuenta con Microsoft Sales como CRM y Microsoft Customer Insights para mejorar el entendimiento de sus clientes.

En base a la inversión realizada en 2022 para la migración de CRP y ERP, es que Ricoh LATAM brindó una red de 24.951 clientes distribuidos en 12 países, para la realización del presente proyecto en donde se pretende demostrar la utilidad de la aplicación de Inteligencia Artificial, para evitar la fuga de clientes.

Debido a que los datos proporcionados no representan el 100% de la cartera de clientes, y se cuenta con limitantes de disponibilidad de ciertos recursos, como personal, estrategias de marketing y operaciones, es que a lo largo del desarrollo del proyecto se deben tomar ciertos supuestos para el tratamiento de los datos, así como de las conclusiones y recomendaciones arribadas.

2. Objetivos

2.1. Objetivos generales

El objetivo general del proyecto es poder identificar grupos de clientes que sean más proclives a cancelar sus servicios con la empresa y poder tomar acciones preventivas para que eso no suceda, minimizando las pérdidas en la cartera de clientes y aumentando las ganancias del negocio en general.

La retención de los clientes es un tema de alta prioridad para la compañía, dado que su negocio apunta a ser un integrador de soluciones para otras empresas, mediante la comercialización de suscripciones recurrentes de diversos servicios.

En numerosos estudios se ha comprobado que el costo de adquisición de clientes es cinco (5) veces mayor al de retención. Los esfuerzos de marketing tradicionales a menudo se dirigen a captar clientes, ignorando el valor de mantener los clientes actuales.

2.2. Objetivos específicos

A efectos de lograr el objetivo general planteado en este proyecto, es que se definen cuatro (4) objetivos específicos:

1. **Mostrar el valor de los modelos de predicción, en particular Churn Rate.** Cualquier negocio que tenga clientes puede y debe usar la predicción de abandono para evitar que esta ocurra. Si bien generar nuevos clientes es importante para el crecimiento, garantizar que los clientes existentes permanezcan, es esencial para la longevidad de cualquier negocio exitoso. La capacidad de predecir cuándo es más probable que un cliente abandone, permite a las empresas considerar la rotación anterior no como una medida de fracaso, sino como una oportunidad de mejora.

   Asimismo, este análisis puede proporcionar claridad sobre la calidad del negocio, identificar que clientes están satisfechos y cuales no, permitiendo obtener una métrica que luego es comparable con la competencia y así poder tomar decisiones estratégicas.

2. **Desarrollar un algoritmo que satisfaga los requerimientos de la compañía.**

   Cada compañía tiene sus requerimientos específicos inherentes a factores tanto externos como internos. El rubro, el modelo de negocio, sus decisiones tecnológicas y hasta su cultura hacen que el problema a resolver sea único.

   Se debe realizar un adecuado análisis y selección de variables que se consideran relevantes para la compañía y su contexto, en base a la información que será proporcionada.

3. **Implementar el algoritmo en un mínimo producto viable.**

   Obtener una solución que satisfaga los mínimos requisitos planteados en el equipo de trabajo para que la solución sea adecuada para el contexto actual de la empresa.

4. **Elaborar conclusiones y lecciones aprendidas que puedan ser usadas por la empresa para la mejora continua del algoritmo y el producto.**

   A partir del trabajo realizado, Ricoh LATAM tendrá acceso a una propuesta base para monitorear su cartera de clientes y ajustar sus diferentes estrategias para mantenerlos y así potenciar su negocio. No obstante, esta es una primera versión de la cual se puede continuar optimizando el Mínimo Producto Viable (MVP) para obtener cada vez mejores resultados.

3. Marco teórico

3.1. Gestión de abandono de clientes (Churn Rate, fuga de clientes)

Churn, abandono o fuga de clientes, se define como el número o porcentaje neto de clientes perdidos en un determinado periodo de tiempo. En las empresas se puede interpretar como una medida de éxito o fracaso de sus políticas y procesos para la retención de clientes.

Ahora bien, el abandono de clientes puede tener diferentes matices que deben ser analizados al momento de buscar una solución basada en modelos de predicción. A continuación, se exponen los diferentes matices.

1) Abandono absoluto: el cliente cesa activamente su relación con la empresa.
2) Abandono presunto: el cliente puede simplemente dejar de interactuar con la empresa, aunque a nivel formal en documentos, no haya ningún cambio.
3) Abandono reactivo: en ocasiones, los clientes pueden reaccionar ante eventos o experiencias negativas específicas que desencadenan en el alejamiento de la empresa.
4) Abandono prospectivo: a veces la desvinculación es gradual y no necesariamente impulsada por un hecho específico, este es lo que se conoce como abandono prospectivo.

Por otro lado, a la hora de analizar el abandono de clientes, debemos tener presente que el tiempo sobre el que se mide y concluye que el cliente ha abandonado su suscripción, depende de la industria.

Por ejemplo, el tiempo que debe transcurrir en una empresa de telecomunicaciones para concluir que un cliente se ha perdido, no es el mismo tiempo que en una empresa de viajes.

Históricamente, la rotación de clientes resultaba en un número / porcentaje que las empresas conocían y sobre el cual poco podían hacer.

Ahora bien, hoy en día las cosas han cambiado, las empresas, a través de los avances tecnológicos tienen la capacidad de capturar una gran cantidad de datos que reflejan la experiencia y estado de sus clientes. El surgimiento de técnicas de inteligencia artificial y análisis de grandes volúmenes de datos ayuda aún más a aprovechar esta riqueza de datos para abordar el problema de abandono.

Con la Inteligencia Artificial y la Big Data, la gestión de abandono dispone de un conjunto de tecnologías que le permite identificar cuasi en tiempo real el comportamiento de los clientes, de forma tal que las empresas puedan tomar medidas proactivas para evitar que el abandono se efectivice. El estudio de los datos históricos sobre las experiencias de los clientes y como estas han respondido a estas experiencias, puede ayudar a desarrollar un modelo para predecir la rotación reactiva. Permite utilizarse para rastrear desencadenantes similares que experimentan los clientes actuales para determinar cómo es probable que reaccionen.

En síntesis, para un resultado adecuado, las empresas deben invertir en conocer gobernar y procesar los datos que describen a sus clientes y su comportamiento, lo que se conoce hoy en día como conocimiento 360 de los clientes.

A continuación, exponemos una imagen que proporciona algunos conjuntos claves de datos que permiten un conocimiento integral de los clientes:

Ilustración 1 - Conjuntos de datos para crear una vista de 360 de los clientes [1]

Ahora bien, analizando los datos que se obtienen de los clientes, y su comportamiento histórico, no son suficientes para realizar una adecuada gestión para disminuir la tasa de abandono, puesto que, para esto, se requiere un seguimiento de las acciones tomadas, para ajustar el mejor curso de acción y abordar el riesgo de deserción a nivel de clientes individuales.

Es por ello por lo que en general se establecen tres (3) pasos claves para atacar de forma eficaz el riesgo de abandono de clientes.

1. Determinar los clientes objetivo para la empresa. Cada cliente tiene un valor diferente para el negocio. Por lo tanto, el primer paso sería averiguar qué tan importante es cada cliente individual para la empresa y definir qué nivel de recursos dedicarle a este grupo de clientes.

2. El siguiente paso consta en determinar cuáles son las estrategias de tratamiento más efectivas para cada tipo de cliente. La estrategia de tratamiento generalmente consta de dos factores:
    (a) qué tipo de campaña o intervención responden mejor y
    (b) qué canales de comunicación responden mejor.

3. Ciclo de retroalimentación de prueba y aprendizaje. Este ciclo ajusta constantemente el modelo, para mejorar su rendimiento. [1]

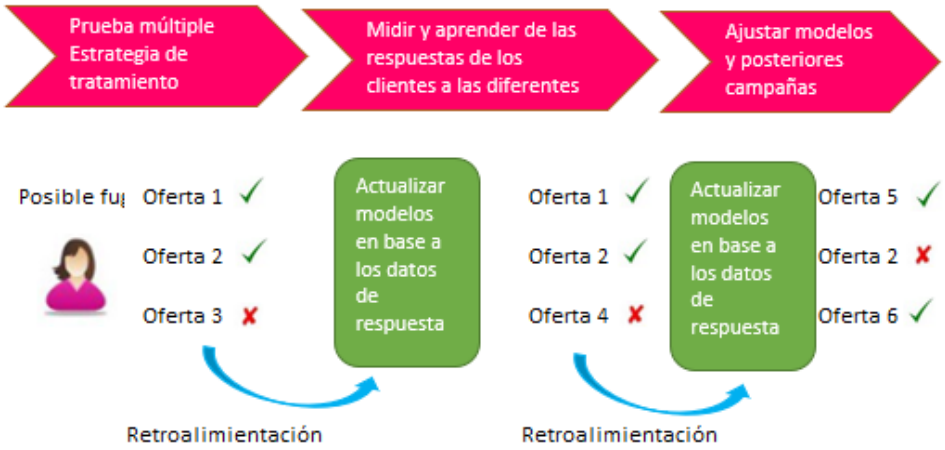


Ilustración 2 - Ciclo de retroalimentación [1]

Es debido a la importancia que se le da a lograr una adecuada gestión de clientes en las diferentes empresas y el potencial que brinda la Big Data, la Inteligencia Artificial, y dentro de esta, el aprendizaje automático, que en los próximos apartados se desarrollaran:

1. Diferentes tipos de aprendizajes de Machine Learning y algoritmos de aprendizaje estadístico predictivo.
2. Se expondrá un caso de uso para obtener una predicción de Chun Rate
3. Sugerir acciones para lograr la retención de los clientes en función de las características y grupos de estos.

3.2. Machine Learning

Machine Learning (ML) o aprendizaje automático es una de las disciplinas de la Inteligencia Artificial (IA), que proporciona la capacidad de aprender y mejorar de manera automática, a partir de la experiencia. Para que un modelo realice predicciones de manera robusta, necesita alimentarse de una gran cantidad de datos.

Una vez que se disponen de los datos, es posible comenzar un proceso de aprendizaje, a través de la aplicación de diferentes algoritmos, en donde se busca transformar los datos en información, analizar y explorar los datos en búsqueda de patrones ocultos, sobre las cuales basar la toma de diferentes decisiones.

3.3. Tipos de aprendizajes de Machine Learning

Es posible implementar Machine Learning de varias maneras, siendo las más difundidas, las técnicas de aprendizaje supervisado y no supervisado.

También es posible aplicar tipos de aprendizajes semi supervisados y por refuerzo.

Ilustración 3 - Tipos de aprendizaje automático [2]

Para el caso planteado se implementarán técnicas de aprendizajes supervisados, dado que se busca identificar grupos de clientes con una alta probabilidad de solicitar la baja del servicio (fuga de clientes, abandono o Churn Rate).

3.3.1 Aprendizaje supervisado

En lo que respecta a aprendizaje supervisado, está técnica consta de un etiquetado de datos, en donde se desarrollan diferentes modelos, que tienen como objetivo asignar una etiqueta al dato de salida, basado en los datos de entradas.

Estos modelos son entrenados con datos históricos conocidos, para aprender como son las asignaciones de las etiquetas y sus relaciones, para así poder aplicar ese proceso o algoritmo, a datos desconocidos y obtener resultados con cierta precisión.

Es muy común utilizar la técnica de aprendizaje supervisado para problemas de clasificación y regresión (predicción de variables).

3.3.2 Aprendizaje no supervisado

El aprendizaje no supervisado es una técnica de carácter exploratorio, el cual se utiliza más en problemas de Clustering, agrupamientos y asociación, para los cuales los datos con los que se dispone no cuentan con una etiqueta previa para entrenar al modelo.

Se utiliza para descubrir la estructura subyacente del conjunto de datos. [3]

3.4. Algoritmos de aprendizaje estadístico predictivo

Los modelos predictivos, se basan en métodos matemáticos para pronosticar variables futuras. Esto se realiza a través de un proceso iterativo, en donde de un conjunto de datos histórico, se toma una parte de estos como conjunto de entrenamiento, y sobre ellos se desarrolla un modelo capaz de predecir, se realizan pruebas sobre estos y luego se validan para determinar su precisión y así identificar el mejor conjunto de algoritmos apto para generar mejores predicciones sobre los datos futuros desconocidos. La validación se torna un aspecto importante en el uso de estos modelos, dado que algunos algoritmos tienden a sobreajustar el modelo a los datos de entrenamiento y no logran generalizar

adecuadamente, volviendo al modelo creado, solo efectivo para los datos con los que fue entrenado.

Estos modelos predictivos se dividen en dos, modelos de clasificación y de regresión.

Los modelos de clasificación se basan en reconocer los distintos patrones y en base a ellos estimar la pertenencia a una clase u otra. Para estimar la pertenencia a la clase, se basa en la cercanía entre las variables.

En tanto los modelos de regresión, son utilizados para predecir un valor. Se busca determinar la relación de variables dependientes con respecto a las variables explicativas.

En lo que respecta a algoritmos de predicción para la fuga de clientes, es muy frecuente ver que se utilicen algoritmos de Árbol de Decisión, Random Forest, Boosting-GBM, Stacking. Aunque el algoritmo que mejor resultados obtiene, depende del caso, su contexto y el conjunto de datos con el que se cuenta. [4]

Si bien es posible utilizar para este tipo de problemas, algoritmos menos complejos, como ser Regresión Lineal o Naive Bayes, estos suelen obtener menores resultados de precisión, por lo que no se desarrollaran en este trabajo de investigación y aplicación para el caso real de Ricoh LATAM.

3.4.1. Arboles de Decisión

Los Árboles de Decisión son una secuencia de condiciones que permiten dividir los datos de manera iterativa (un nodo tras otro, esencialmente) hasta que se logra asignar cada dato a una etiqueta. Estos pertenecen a una clase de algoritmos de aprendizaje automático supervisado, que se utilizan tanto en el modelo predictivo de clasificación (predice resultados discretos) como en la regresión (predice resultados numéricos continuos). Los cuales se construyen a partir de solo dos (2) elementos: nodos y ramas.

Estructura de Árbol de Decisión:

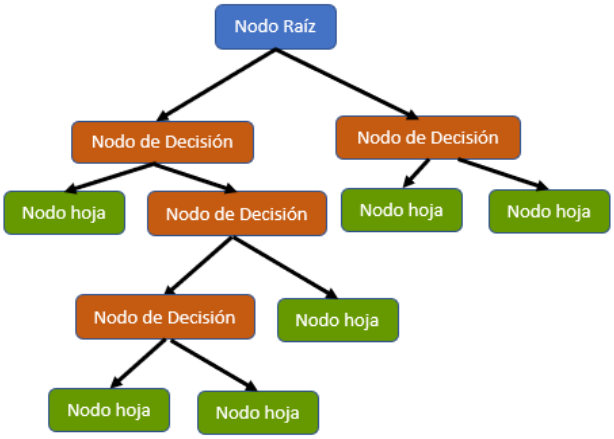


Ilustración 4 - Diagrama de Árboles de Decisión

Los **nodos raíz** contienen una función para dividir con mayor precisión los datos a nivel primario en los **nodos de decisión.**

Los **nodos de decisión** son los nodos donde se evalúan las variables.

Los **nodos hoja**: son los nodos finales en los cuales se realiza la predicción.

Esta técnica tiene los siguientes parámetros configurables para obtener mejores resultados a la hora de entrenar y predecir el modelo:

1. *Max depth*: profundidad máxima del árbol.
2. *Min samples split*: número mínimo de muestras para dividir un nodo.
3. *Min samples leaf*: número mínimo de muestras para cada nodo de hoja.
4. *Max leaf nodes:* el número máximo de nodos hoja en el árbol
5. *Max features*: número máximo de características que se evalúan para dividir en cada nodo. Este parámetro es válido para algoritmos que aleatorizan las características consideradas en cada división, es decir no siempre se va a ajustar.

A continuación, se expone como afecta cada parámetro definido, en el resultado del modelo.

1. *Max depth*: aumentar la profundidad máxima disminuirá el sesgo y aumentará la varianza.
2. *Min samples split*: aumentar la división mínima de muestras aumenta el sesgo y disminuye la varianza.
3. *Min samples leaf*: aumentar la hoja de muestras mínimas aumenta el sesgo y disminuye la varianza.
4. *Max leaf nodes*: la disminución del nodo de hoja máxima aumenta el sesgo y disminuye la varianza.
5. *Max features*: la disminución de las características máximas aumenta el sesgo y disminuye la varianza.

Como ventajas de este algoritmo, se destaca:

- Son simple de entender, interpretar y visualizar.
- Pueden manejar datos numéricos y categóricos, así como también pueden manejar problemas de múltiples salidas.
- Los árboles de decisión requieren relativamente poco esfuerzo por parte de los usuarios en la etapa de preparación de datos.
- Las relaciones no lineales entre parámetros no afectan el rendimiento del árbol.

Como desventajas del algoritmo se tiene:

- Sobreajuste: Los Árboles de Decisión tienden al sobreajuste muy rápidamente. Si se los deja crecer sin un mecanismo de poda o un mecanismo de corrección después de que el árbol haya sido entrenado, pueden dividirse tantas veces que cada hoja es una muestra. Esto significa que literalmente aprendieron cómo se ve el conjunto de datos de entrenamiento, pero no logra generalizar bien para predecir en datos desconocidos.
- No es robusto ante la introducción de datos nuevos. Un pequeño cambio en los datos de entrenamiento puede resultar en un árbol completamente diferente.

- Los Árboles de Decisión de clasificación tienden a favorecer la predicción de la clase dominante en conjuntos de datos con desequilibrio de clases.
  Las características que se dividen con mayor frecuencia y que están más cerca de la parte superior del árbol, se consideran las más importantes.

Como forma de contrarrestar estas desventajas, se puede establecer un número mínimo de entradas de entrenamiento para usar en cada hoja del árbol, aplicar técnicas de poda, lo que implica eliminar las ramas que hacen uso de características que tienen poca importancia. De esta manera, reducimos la complejidad del árbol.

A modo de resumen, se exponen algunas soluciones que podemos implementar para evitar el sobreajuste:

- Reducir la profundidad máxima.
- Aumentar la división mínima de muestras.
- Equilibrar los datos para evitar el sesgo hacia las clases dominantes.
- Disminuir el número de características.

3.4.2. Random Forest

Random Forest (RF) es una técnica de aprendizaje automático supervisada basada en Árboles de Decisión. La cual, sirve tanto para problemas de regresión como de clasificación.

Esta técnica se basa en la combinación de diferentes Árboles de Decisión seleccionados de forma aleatoria. Estos árboles toman diferentes porciones de datos y sobre ellos realizan una predicción, la cual al combinarlas posteriormente unos errores se compensan con otros (realiza un promedio) y así se logra una predicción que generaliza mejor.

La aleatoriedad en la selección de los árboles es lo que permite disminuir la correlación entre estos, mejorando así el desempeño del modelo a la hora de predecir.

La idea es que el aprendizaje en conjunto se desempeñe mejor que el aprendizaje individual.

Esto lo logra utilizando la técnica de Bagging, puesto que combina de forma aleatoria modelos individuales de Árboles de Decisión para formar el bosque. Cada árbol se construye usando una muestra aleatoria de registros y cada división se construye usando una muestra aleatoria de predictores.

Esta técnica es uno de los mecanismos que se pueden utilizar para abordar el problema de sobreajuste del algoritmo de Arboles de Decisión.

RF generalmente no es propensos a sobre ajustarse porque la selección aleatoria de funciones y el Bagging tienden a promediar cualquier ruido del modelo.

La adicción de más árboles no provoca el sobreajuste, ya que el proceso de aleatorización continúa promediando el ruido, de hecho, a mayor cantidad de árboles, generalmente se reducen el sobreajuste. Sin embargo, es posible que el algoritmo sobreajuste, si los Árboles de Decisión aleatorios tienen una varianza extremadamente alta.

El algoritmo de RF utiliza como parámetros configurables para el entrenamiento:

- *Mtry:* consta de la cantidad de predictores seleccionados aleatoriamente. Es decir, el número de predictores considerados en cada división. Por defecto, se emplea como valor la raíz cuadrada del número total de predictores disponible (redondeado a la baja).
- *Ntree:* número de árboles
- *Sampe_size:* número de muestras aleatorias
- *Node_size:* tamaño de los nodos (número de observaciones en el nodo final).
- *Num estimator*: números de Árboles de Decisión en RF
- *Max features*: máximo número de características que se evalúan para dividir en cada nodo.
- *Max_depth*: número máximo de niveles en cada Árbol de Decisión.
- *Min_samples_split*: número mínimo de puntos de datos colocados en un nodo antes de que el nodo se divida.
- *Min_samples_leaf*: número mínimo de puntos de datos permitidos en un nodo hoja.
- *Bootstrap*: métodos para muestra de puntos de datos (con y sin remplazo).

Es posible decir que la técnica de RF se ejecuta en dos (2) etapas, en primer lugar, se crea un bosque aleatorio combinando N Árboles de Decisión, y en segundo lugar se procede a realizar la predicción de cada árbol de la primera etapa.

Para ejecutar esta técnica pueden numerarse cinco (5) pasos:

1. Seleccionar K datos aleatorios del conjunto de entrenamiento.
2. Construir los Árboles de Decisión asociados con los datos seleccionados (Subconjuntos).
3. Definir el número N para los Árboles de Decisión que desea construir.
4. Repetir los pasos 1 y 2.
5. Para nuevos datos, obtener las predicciones de cada Árbol de Decisión y asignar los nuevos datos a la categoría que gana la mayoría de los votos. [5]

A continuación, se exponen algunas ventajas y desventajas de esta técnica.

Ventajas:

- En general, funciona bien, aún sin modificar los valores por defecto de los parámetros.
- Tiene un buen funcionamiento tanto para problemas de clasificación como de regresión.
- Reduce significativamente el sobreajuste, al utilizar múltiples árboles aleatorios.
- Se mantiene estable al agregar nuevas muestras, ya que al utilizar cientos de árboles sigue prevaleciendo el promedio de sus votaciones.

Desventajas:

- Insume un mayor costo de creación y ejecución, que un solo Árbol de Decisión.
- En algunos datos de entrada "particulares" RF también puede caer en sobreajuste.
- En función de cómo se configuren sus parámetros, puede requerir mucho tiempo para el entrenamiento.
- No obtiene buenos resultados en conjuntos de datos pequeños.
- Es complejo poder interpretar la gran cantidad de árboles creados en el bosque, si se busca comprender y explicar a un cliente su comportamiento.

3.4.3. Gradient Boosting Machine

El Gradient Boosting Machine (GBM) es una técnica que entrena varios modelos individuales en forma secuencial y escalonada. Cada modelo aprende de los errores cometidos por el modelo anterior.

Para la creación del modelo predictivo utiliza la técnica de Boosting y así minimizar gradualmente los errores de los modelos simples, empleando el método del gradiente en la función de pérdida. Es decir, se basa en múltiples predictores débiles (Árboles de Decisión) para crear un predictor fuerte. Específicamente incluye una función de perdida que calcula el gradiente del error con respecto a cada característica y luego crea interactivamente nuevos Árboles de Decisión que minimiza el error actual.

GBM puede ejecutarse en cinco (5) pasos:

1. Entrenar un Árbol de Decisión.
2. Aplicar el Árbol de Decisión recién entrenando para predecir.
3. Calcular el residual de Árbol de Decisión, reemplazar los "y" por los errores residuales (nuevo "y").
4. Repita el paso 1, hasta alcanzar la cantidad de árboles que se configuran para el entrenamiento.
5. Realizar la predicción final.

Los principales hiperparámetros que se pueden ajustar en los modelos GBM, adicionales a los hiperparámetros de árboles, son:

- *Loss function*: función de perdida para calcular el error. Es un método para evaluar que tan bien un algoritmo especifico modela los datos otorgados.
- *Learning rate*: la velocidad a la que los árboles nuevos corrigen / modifican el predictor existente.
- *Num estimator*: El número total de árboles a producir para la predicción final.

A diferencia de los algoritmos de Bagging y RF, los algoritmos Boosting tienden al sobreajuste si la profundidad es muy alta a pesar de que este tiende a ocurrir lentamente, ante esto, es posible intentar reducir el sobreajuste al reducir el hiperparámetro de "*learning rate*", o reducir del tamaño de la submuestra.

Dentro de lo que es la familia de los algoritmos de Gradient Boosting, existen otras técnicas más robustas como ser XGBoost, el cual se basa en una implementación optimizada del algoritmo de árboles aumentados, maneja más tipos de datos, relaciones y distribuciones que otros algoritmos de árboles aumentados. Optimiza los recursos (*software y hardware*) para obtener resultados superiores utilizando menos recursos informáticos en el menor tiempo posible.

A continuación, se expone una imagen para ilustrar las razones de porque este algoritmo es eficiente a la hora de ejecutar su implementación.

Ilustración 5 - Diagrama explicativo de XGBOOST [6]

En cuanto a resultados obtenidos, se han realizado diversos estudios con *dataset* de prueba, para los cuales se ha seleccionado uno para exponer una comparativa de la predicción a la que se arriba y el tiempo de entrenamiento insumido.

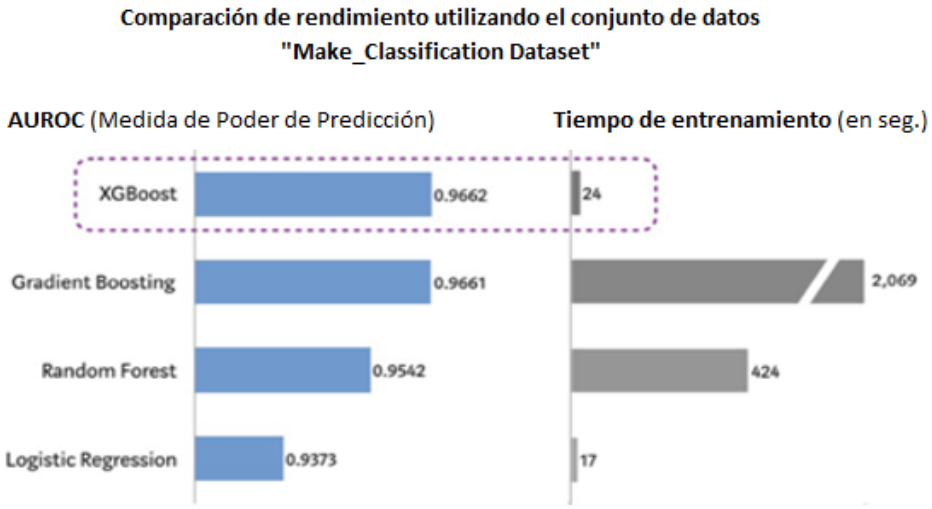


Ilustración 6 - Comparación de rendimiento de modelos [6]

Ensamble es un aprendizaje muy poderoso que no solo puede usarse para clasificación, sino que puede usarse para regresión. Aunque generalmente funcionan en métodos de árbol, también se pueden usar para métodos lineales.

Es importante tener presente, que no solo se debe elegir el algoritmo correcto, sino también la configuración correcta de algoritmos ante la realidad bajo entrenamiento, para obtener adecuados resultados.

A continuación, se exponen algunas ventajas y desventajas de esta técnica.

Ventajas:

- Son capaces de seleccionar predictores de forma automática.
- Se pueden utilizar en problemas de regresión y clasificación.
- Los árboles pueden, en teoría, manejar tanto predictores numéricos como categóricos sin tener que crear variables *dummy*.
- Al tratarse de métodos no paramétricos, no es necesario que se cumpla ningún tipo de distribución especifica.

- Por lo general, requieren mucha menos limpieza y preprocesamiento de los datos en comparación con otros métodos de aprendizaje estadístico (no requieren estandarización).
- No se ven muy influenciados por *outliers*.
- Si para alguna observación, el valor de un predictor no está disponible, a pesar de no poder llegar a ningún nodo terminal, se puede conseguir una predicción empleando todas las observaciones que pertenecen al último nodo alcanzado.
- Son muy útiles en la exploración de datos, permiten identificar de forma rápida y eficiente las variables (predictores) más importantes.
- Tienen buena escalabilidad, pueden aplicarse a conjuntos de datos con un elevado número de observaciones.

Desventajas:

- Al combinar múltiples árboles se pierde la interpretabilidad que tienen los modelos basados en un solo árbol.
- Cuando tratan con predictores continuos, pierden parte de su información al categorizarlas en el momento de división de los nodos.
- No son capaces de extrapolar fuera del rango de los predictores observado en los datos de entrenamiento.

3.4.4. Stacking

Stacking es un método de ensamble, como Boosting y Bagging, el cual consta en realizar una combinación de varios submodelos predictivos para mejorar el rendimiento de la predicción. Aprende como combinar mejor las predicciones de los submodelos y crear un nuevo modelo con dichas combinaciones.

Este permite utilizar cualquier modelo de aprendizaje automático para aprender cómo combinar mejor las predicciones de los miembros contribuyentes. El modelo que combina las predicciones se denomina metamodelo, mientras que los miembros del conjunto se denominan modelos base.

Como **ventajas**, si se aplica correctamente permite generar un modelo que sea eficiente ante datos no conocidos. Se puede utilizar tanto en escenarios supervisados como no supervisados y el rendimiento de este método de ensamblado aumenta cuanto más diversos sean los modelos.

Como **desventaja** se puede comentar que insume mucho tiempo computacional y que es difícil de configurar para obtener buenos resultados.

En los métodos de Stacking, los diferentes modelos débiles se ajustan de forma independiente entre sí y se entrena un metamodelo, para predecir los resultados en función de los resultados devueltos por los modelos base.

A continuación, se expone una breve comparativa de las técnicas de ensamblado comentadas anteriormente.

	Bagging	Boosting	Stacking
Partición de los datos en subconjuntos	Aleatorio	Mayor preferencia a muestras mal clasificadas	Varios
Objetivo	Minimizar la varianza	Aumentar la fuerza predictiva	Ambos casos
Método donde se utiliza	Subespacio aleatorio	Descenso de gradeinte	Combinación
Función para combinar modelos individuales	Peso promedio	Voto mayoritario	De regresión

Tabla 1 - Tabla comparativa de algoritmos Bagging, Boosting y Stacking [7]

3.5.Ensayo del proyecto

Para el desarrollo de este proyecto, antes de obtener los datos reales, se realizó un experimento preliminar de un conjunto de datos públicos obtenidos de una compañía de telecomunicaciones.

3.5.1. Información de los datos

En primera instancia se instalaron algunas librerías para el trabajo. Se comenzó con las librerías más comunes y principales para este tipo de análisis y a medida que fue avanzando en el trabajo, se fueron agregando algunas otras que resultaban adecuadas e interesantes para el proyecto. Algunos de los ejemplos de librerías más utilizadas son: Matplotlib, Seaborn y Pandas.

Posteriormente, cuando se procedió a la etapa de creación del modelado, para obtener mejores resultados, se incorporó la librería de Sklearn.

```
# importar las librerias necesiarias:
import warnings
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
%matplotlib inline
pd.options.display.float_format = '{:.2f}'.format
warnings.filterwarnings('ignore')
```

Ilustración 7 - Importación de librerías

En segunda instancia se cargó un conjunto de datos de prueba que provienen en un formato de tabla CSV sobre la empresa Telecom. Los *dataframe* son excelentes para representar datos reales: las filas corresponden a instancias (ejemplos, observaciones, entre otros) y las columnas corresponden a características de estas instancias.

La lectura de los datos se realizó con "*read_csv*". Con el objetivo de visualizar su importación y contenido, se observaron las primeras 3 líneas con el código "*head*":

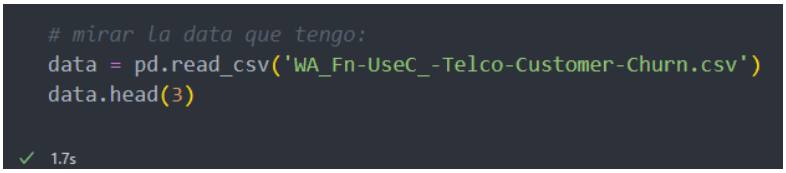


Ilustración 8 - Lectura de los datos

Para continuar con la exploración del conjunto de datos, se ejecutaron los comandos:

- "*data.shape*" para obtener la dimensión de los datos.
- "*data.columns*" para observar las columnas de los datos .
- "*sample*" con el cual se logran visualizar diez (10) elementos sin repetición del conjunto de datos.
- "*describe*" para ver la descripción de las variables y detectar valores nulos en los datos.

Por otro lado, para un uso más eficiente de los datos, se analizaron las variables de tipo categóricas, para aplicarles así, una transformación a tipo numérica.

Se creó una copia del conjunto de datos original y se etiquetó la codificación de los datos de texto ("*Object"*) a numéricos ("*Int32"*). Se ejecutó el comando "describe" para ver los nuevos datos.

3.5.2. Análisis exploratorio de datos

Aquí lo que se realizó fue dividir las columnas numéricas de las categóricas. Resultando que la variable objetivo está desbalanceada.

En esta etapa, se comenzó con una normalización de los datos y se analizó la matriz de correlación para identificar que variables se encuentran altamente correlacionadas y no aportaban a la generación del modelo. Finalmente, se ejecutó el *chi-squared-Test,* con el cual se identificaron variables que no eran relevantes para el análisis.

Luego, para contrarrestar el desbalanceo de los datos se aplicó el método SMOTE el cual es uno de los métodos de sobremuestreo más utilizados para resolver el problema del

desequilibrio. Su objetivo es equilibrar la distribución de clases aumentando aleatoriamente los ejemplos de clases minoritarias al replicarlos.

3.5.3. Modelado

En lo que respecta al modelado, se particionaron los datos en *train* y *test*. Para ello se determinó como porción de datos de *train* el 80% de estos y el restante 20% se designó para *test*.

Luego de determinada la partición de los datos, se procedió con la aplicación de diferentes algoritmos, como ser "Xgboost Classifier", el cual resultó en un *accuracy* de 83%, siendo este un resultado aceptable.

También se instrumentó la matriz de confusión para complementar el resultado del algoritmo e interpretar el mismo para el caso planteado, donde se observaron los falsos positivos resultantes, que representan los clientes que realmente se retiraron y se predijo que no se iban. Este resultado se corresponde con el peor escenario a obtener, dado que la empresa no realiza ninguna acción para retener a estos clientes, que considera que no se van a retirar. Con este algoritmo se obtuvo un resultado de 6,62%, siendo este el menor resultado obtenido en comparación con otros algoritmos que se ejecutaron en el ensayo del proyecto.

True Positive = TP

False Negative = FN

False Positive = FP

True Negative = TN

Accuracy = 821+896/ (821+896+219+134) = 83%

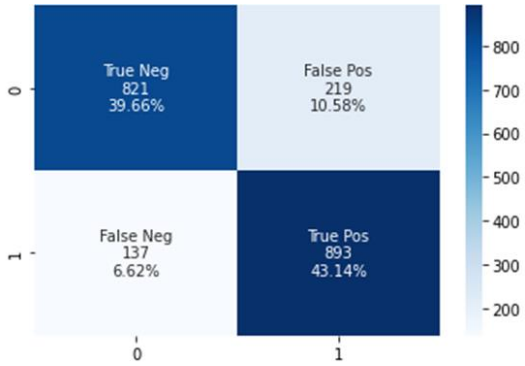


Ilustración 9 - Matriz de confusión

Al ejecutarse el algoritmo de “LightGBM Classifier” no se observaron mayores mejoras a lo anteriormente comentado.

Luego se aplicó Random Forest y Árbol de Decisión, de estos dos últimos se obtuvo un *accuracy* un poco más bajo que los anteriores un 78% y un 76% respectivamente, y los falsos positivos en la matriz de confusión fueron levemente superiores. En lo que respecta a los falsos negativos el que más se incrementó es el modelo de Árbol de Decisión.

Por último, se aplicó un algoritmo de ensamblado de Stacking, que combina los cuatro (4) algoritmos mencionados anteriormente (XGBClassifier, LightGBMClassifier, RF Classifier, Árboles de Decisión Classifier), con el cual no se observaron mejoras sustanciales de los resultados frente a los obtenidos al ejecutar los modelos individuales. El *accuracy* obtenido es de 83%, al igual que con XGBClassifier.

En resumen, el que mejor performance obtuvo en las distintas métricas fue Stacking que es un algoritmo de ensamble que combina varios submodelos, en este caso los cuatro anteriores.

Sr. No.	ML Algorithm	Cross Validation Score	ROC AUC Score	F1 Score (Churn)
1	XGBClassifier	90.17%	82.63%	83%
2	LightGBMClassifier	90.33%	82.87%	83%
3	RandomForestClassifier	85.69%	79.12%	80%
4	DecisionTreeClassifier	84.29%	76.53%	79%
5	Stack of All 4 Classifiers	90.88%	83.01%	83%

Tabla 2 - Comparación de performance de Algoritmos

3.5.4. Conclusiones del ensayo

Algunas impresiones de los resultados obtenidos del conjunto de datos del pre-proyecto de telecomunicaciones son:

- Como público objetivo, quizás se puedan generar tres tipos de clientes: tercera edad, los que viven en pareja y los que viven solos.
- Se visualizó que el número de clientes de la tercera edad es bajo, pero el límite inferior de pagos mensuales es más alto que el de los otros clientes. Por lo tanto, se intuye que los clientes de tercera edad están dispuestos a pagar más dinero por el servicio que los clientes que viven solos o en pareja.
- Para tener una base sólida de clientes la empresa necesita crear una entrada fácil y accesible para sus servicios. Para la permanencia de los primeros seis (6) meses, debe centrarse ampliamente en "*OnlineSecurity*", "*OnlineBackup*", "*deviceProtection*" y "*TechSupport*" ya que este período es el más crítico e incierto para los clientes.
- Otra de las conclusiones a la que se puede arribar es que se debe poner fin al pago por cheque electrónico debido a su alta rotación y centrarse en la trasferencia bancaria y tarjeta de crédito.

Este *dataset* de telecomunicaciones cuenta con grandes oportunidades para adelantarse en el problema comercial del mundo real y puede tratarse con las técnicas de ciencia de datos.

Los conocimientos de Análisis Exploratorio de Datos (EDA) resultaron muy valiosos para comprender la eficiencia de los sistemas existentes y en la elaboración de planes y medidas para contrarrestar los problemas o evitar un ciclo infinito de mejora. El EDA se trasladó al caso real de la tesis con algunas modificaciones, dependiendo del conjunto de datos disponible. Por ejemplo, se utilizó el análisis SMOTE para equilibrar los datos, así como combinaciones de submuestreo y sobre muestreo.

En cuanto al rendimiento del modelo todos preformaron de forma similar, quizás Xgboost Classifier fue el que mejor resultado registro de los modelos individuales. Tal vez con el ajuste de hiperparámetros y la detección de *outliers*, es posible que se siga mejorando el modelo.

Para profundizar en las bases de las conclusiones mencionadas, se recomienda revisar el código “Pre-Proyecto”.

4. Desarrollo del proyecto

4.1. Arquitectura de la solución

La arquitectura es una abstracción del sistema que describe cierta parte de su estructura, omitiendo ciertos detalles irrelevantes para el modelado bajo diferentes puntos de vista. [8]

No diseñar una adecuada arquitectura lógica, puede implicar riesgos en los esquemas físicos de las organizaciones. Su diseño se hace en base a objetivos (requisitos), los cuales son aquellos prefijados para el sistema, pero no sólo los de tipo funcional, sino también el mantenimiento, disponibilidad, la auditoría, flexibilidad, seguridad, escalabilidad e interacción con otros sistemas de información.

Por otro lado, se deben tener en cuenta las restricciones del contexto y del dominio de la aplicación, ya que estas son las que determinan cuando una arquitectura es más recomendable de implementar, ya que unas resultan más eficientes con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de s*oftware* de tres capas para implementar sistemas en tiempo real.

En base a lo expuesto, se procede a detallar la arquitectura elegida para abordar los objetivos planteados en el proyecto. Se utilizaron herramientas del ecosistema de Microsoft ya implementados por la empresa, por las siguientes razones:

- Simple integración con la arquitectura actual de la empresa
- Seguridad avanzada
- Gobernanza
- Facilidad de administración
- Disponibilidad
- Escalabilidad
- Portabilidad

En lo que respecta a la arquitectura de la solución, se presentó una restricción de alcance, dado que la empresa ya se encontraba utilizando Microsoft Azure, por lo que la solución

propuesta debe enmarcarse en el ecosistema ya existente. No obstante, los requerimientos destacados de disponibilidad, escalabilidad, compatibilidad e integración con otros sistemas y seguridad se ven abordados con Azure.

Para el caso de estudio se tomaron datos tanto del CRM (Microsoft Customer Insights) como del ERP (Microsoft D365 Finance & Operations) de Ricoh, almacenados en Azure Data Lake. Luego de implementada la arquitectura, los datos serían procesados en *batches* cada 24hs.

4.1.1. Diagrama de la arquitectura elegida

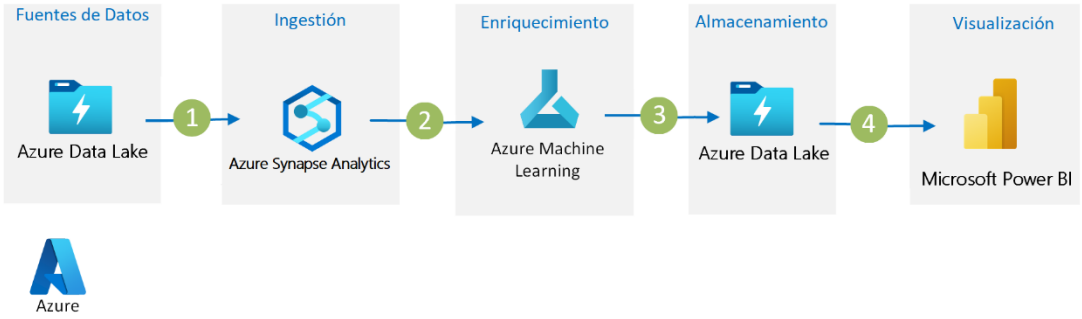


Ilustración 10 - Vista esquemática de la arquitectura elegida y el flujo de datos

1) **Azure Synapse Analytics** extrae, transforma y carga los datos en *batches* con comandos SQL desde el **Azure Data Lake Storage** de Ricoh.
2) Se utiliza **Azure Machine Learning** para enriquecer los datos con un modelo de aprendizaje automático.
3) Los datos resultantes del modelo son almacenados en **Azure Data Lake Storage**. También proporciona memoria *caché* para entrenar el modelo de aprendizaje automático.
4) Los datos obtenidos pueden visualizarse en *dashboards* de **Power BI,** que proporciona un panel con representaciones gráficas interactivas que usan datos almacenados en **Azure Synapse Analytics** para impulsar decisiones sobre las predicciones.

4.1.2. Componentes

Azure Data Lake Storage es un servicio para crear *data lakes*, ofreciendo una forma fácil de administrar cantidades masivas de datos. Destaca por su rendimiento, fácil administración y seguridad.

Sus funcionalidades más importantes incluyen: acceso compatible con Hadoop, escalabilidad de forma natural (hasta varios exabytes con niveles de latencia medios), alta compatibilidad con otros servicios de Azure.

Azure Synapse Analytics es un servicio de análisis ilimitado que reúne la integración de datos, el almacenamiento de datos empresariales y el análisis de macrodatos posee amplia libertad para manipular los datos para obtener diferentes tipos de consultas, usando opciones sin servidor o dedicadas, a gran escala. Azure Synapse combina todo esto para ofrecer una experiencia unificada para ingerir, explorar, preparar, transformar, administrar y servir datos con el fin de satisfacer las necesidades inmediatas de inteligencia empresarial y aprendizaje automático.

Las funcionalidades más importantes del servicio incluyen:

- Tareas de integración, exploración y almacenamiento de datos
- Análisis de macrodatos y aprendizaje automático desde un único entorno unificado
- Exploración de datos relacionales y no relacionales directo del *data lake*
- Crear procesos ETL/ELT en un entorno visual sin necesidad de escribir código
- Usar motores de Apache Spark y SQL totalmente integrados
- Análisis y registros de telemetría, uso de diferentes lenguajes (T-SQL, KQL, Python, Scala, Spark SQL y .Net)
- Integración profunda de Azure Machine Learning, Azure Cognitive Services y Power BI.

Azure Machine Learning es un servicio de nivel empresarial para el ciclo de vida del aprendizaje automático. Ayuda a los científicos de datos a preparar los datos, crear y

entrenar modelos, desplegar en producción y monitorear cientos de modelos en una misma plataforma.

Otras funcionalidades que destacan son: uso de *jupyter notebooks* para trabajar de forma colaborativa, incluye de *frameworks open-source* de forma *out-of-the-box* y manejo de *pipelines*, permitiendo la capacidad de trazabilidad, auditabilidad y gobernanza.

Power BI es una plataforma de visualización unificada y escalable que proporciona *dashboards* interactivos, permitiendo la toma de decisiones basado en datos.

Con sus más de quinientos (500) conectores de datos, usando tanto datos estructurados como no estructurados, permite generar reportes para análisis de negocios en diferentes plataformas.

Las capturas de los pasos que se siguieron para la implementación en Azure, así como la forma en la que interactúan los componentes, se encuentran en la sección de Anexos (10.2. Implementación en Azure)

4.2.Dataset y problemática a resolver

La empresa compartió para la realización de este proyecto los siguientes *datasets*:

Accounts: Contiene 129.555 registros de cuentas del CRM. Entre ellas se encuentran, clientes potenciales, clientes activos y ex – clientes.

Entre las variables disponibles, se decidió descartar por falta de registros a las variables: Cliente VIP (variable binaria, menos de 20 registros que indica si el cliente es de gran importancia para la empresa), Programa de Retención de Cliente (variable que indica si el cliente actualmente recibe beneficios para mantener el servicio; menos de 200 registros), Número de Empleados (variable con menos de 100 registros, que indica el tamaño de la empresa medido en cantidad de empleados), Clasificación de Cartera (variable categórica que describe 4 tipos de clientes según su importancia financiera (Clientes Standard, Premium, Gold o Atrasados); cuenta con menos de 200 registros y solo es utilizada por algunas subsidiarias), Cantidad de Campañas Activas (Enumera la cantidad de campañas publicitarias en la que está participando el cliente, solamente cuenta con 7 registros), entre otras.

Por otro lado, también hay variables que fueron descartadas por su poca relevancia al problema de estudio, entre ellas: Método preferido de Contacto (vía de comunicación entre la empresa y el cliente) y última Fecha de Modificación en el CRM (fecha en la cual se realizó una actualización de los datos del cliente en la base de datos).

Las variables que se decidió conservar son: CustomerId, Fecha de la Última Factura Emitida, País de la Venta, Cantidad de Equipos Arrendados, Vertical y Antigüedad del cliente, las cuales se describen en el apartado de “Estructura del dataset”.

Primero se filtró de la lista aquellos registros que no contaban con fecha de última factura, correspondientes a registros de clientes potenciales, obteniendo así 24.951 registros. Luego, se creó una nueva columna con la fecha de extracción de la data y se restó a la columna “fecha de la última factura”, obteniendo así la cantidad de días desde la última factura. Por política de la empresa, una cuenta se considera inactiva tras un año de inoperatividad, así que se creó una nueva columna “Churn”, categórica, que etiqueta con

0 a los registros cuya cantidad de días desde la última factura es menor a 365 y 1 a los demás casos.

Actividades: Contiene 178.731 registros de diferentes actividades realizadas por el equipo de ventas de Ricoh. Esto incluye: citas, llamadas de teléfono, correo electrónico, tareas, sesiones y cartas. Estas actividades cuentan con el CustomerId, fecha de la acción e ID de Acción (obsoleto). La variable correspondiente a acciones fue dividida en columnas *dummy* y luego agrupadas por CustomerId, consiguiendo 24.348 registros de la cantidad de acciones por cliente en los últimos 24 meses.

Casos creados: Contiene 533.536 registros de diferentes acciones referidas a casos de soporte creados en los últimos 24 meses, su título, cliente, su tiempo de respuesta y su fecha. El título del registro contiene más información del caso, tipo (reclamo o incidente) y subtipo (Instalación, hardware, redes, entre otros), así que se dividió y extrajo en nuevas columnas. Se decidió descartar el subtipo porque no resultaba relevante para el problema de estudio. Luego se crearon columnas *dummy* para las variables de reclamos e incidentes y según el tiempo de respuesta se etiquetaron de forma binaria en base a si se habían cumplido a tiempo o no. Finalmente se agruparon por CustomerId, consiguiendo 14.900 registros correspondientes a la cantidad de incidentes y reclamos por cliente en los últimos 24 meses, Tiempo de respuesta promedio de esos casos y la Fracción de Casos solucionados a tiempo.

Scores: Contiene 2.756 registros de diferentes encuestas agrupadas por CustomerId y su fecha de realización. Se cuenta con información de 6 encuestas diferentes que miden aspectos como satisfacción, conexión, consistencia, lealtad y probabilidad de futura recomendación. Luego, se creó una nueva variable promediando las encuestas completadas por cliente y se clasificaron según su puntuación en "Negativas", "Indiferentes" y "Positivas".

Contratos NO autorrenovados: Contiene 89.935 registros de contratos con fecha de finalización en los últimos 24 meses y un período de renovación mayor o igual a 1 mes, correspondientes a contratos factibles a autorrenovarse. Las variables que contiene incluyen CustomerId, ID de Contrato, Fecha de Finalización del contrato, Fecha de la Próxima Renovación Automática, Período de Renovación y Contrato NO Autorrenovado

entre otras. Ésta última es una variable categórica que etiqueta con 1 los contratos que no fueron autorrenovados y 0 en caso contrario. Luego, se realizó la suma los registros de Contrato NO Autorrenovado y la cuenta del total de contratos, agrupados por un mismo CustomerId. Para finalizar se calculó el ratio de: Contratos No Autorrenovados sobre Total de Contratos.

Facturación: Contiene 22.593 registros de montos facturados agrupados por CustomerId. Las variables que contiene incluyen los Montos Facturados en los últimos 12 y 36 meses. A partir de esto, se creó una nueva variable para obtener el promedio mensual de facturación. Se tomaría el Monto Facturado en los últimos 12 meses en caso de tener un valor mayor a 0, de caso contrario se utilizaría la correspondiente a 36 meses, para obtener el promedio mensual. A partir de ésta, se procedió a crear otra variable, de tipo categórica (Categoría de Facturación), para clasificar a los clientes por cuartiles: "Facturación muy baja", "Facturación baja", "Facturación alta", "Facturación muy alta".

Luego, se combinaron los *datasets* haciendo un *left join* con los registros de CustomerId de Accounts. Obteniendo así un conjunto de datos con 15 variables de aproximadamente 25.000 clientes diferentes.

4.2.1. Estructura del *dataset*

Con respecto a las variables a ser utilizadas en los diferentes momentos del análisis, se conformó un conjunto de datos que contiene las siguientes variables, siendo que cada una de ellas es más relevante en diferentes momentos del análisis.

Variables que fueron incluidas en el conjunto de datos para el proyecto:

1. **CustomerId:** variable numérica que identifica al cliente único.
2. **País de Venta**: variable categórica que representa la subsidiaria de Ricoh que generó el registro en el CRM.
3. **Vertical**: variable categórica que indica el tipo de empresa, como ser: servicios públicos, instituciones educativas, servicios profesionales, entre otros.
4. **Meses desde última factura:** variable numérica que indica la cantidad de meses que han transcurridos desde que se emitió la última factura al cliente.

5. **Cantidad de equipos arrendados:** variable numérica que indica la cantidad de los equipos por un mismo cliente.
6. **Antigüedad del cliente**: Variable numérica que indica la cantidad de años que el cliente trabaja con la empresa.
7. **Casos abiertos en los últimos 12 meses:** Variable numérica que indica la cantidad de casos de quejas que fueron abiertos en los últimos doce meses.
8. **Casos Abiertos:** Variable numérica que indica la cantidad de casos de quejas que están abiertas en este momento.
9. **Cantidad de comunicaciones en los últimos 24 meses:** variable numérica que indica los contactos que se realizaron con el cliente por distintos medios de comunicación, como ser: llamada, mail, cita, entre otros.
10. **Encuestas:** variable categórica para orientar la satisfacción del cliente. Siendo estas "Negativas", "Indiferentes" y "Positivas".
11. **Renovación automática de contrato**: variable numérica que indica la fracción de contratos en lo que se canceló la auto renovación, en los últimos 24 meses.
12. **Categoría de facturación**: variable categórica que representa los montos facturados en categorías de "Facturación muy baja", "Facturación baja", "Facturación alta", "Facturación muy alta", en base a la distribución de los datos.
13. **Cantidad de incidentes**: variable numérica que indica la cantidad de veces en los últimos 24 meses en los que técnicos de Ricoh detectaron anomalías.
14. **Cantidad de reclamos**: variable numérica que indica la cantidad de veces en los últimos 24 meses en los que el cliente solicitó acción correctiva por parte de Ricoh.
15. **Tiempo promedio de respuesta**: variable numérica que indica el promedio de tiempo en solucionar los reclamos de los clientes.
16. **Fracción a tiempo**: variable numérica que indica la fracción de reclamos solucionados a tiempo según el acuerdo de servicio de Ricoh con el cliente.
17. **Churn:** variable categórica que indica si el cliente permanece en la empresa o no.

4.3. Análisis Exploratorio de los Datos

4.3.1. Carga y revisión de los datos

Para el desarrollo del proyecto se utilizó el lenguaje de programación Python en el editor de código Visual Studio Code**.**

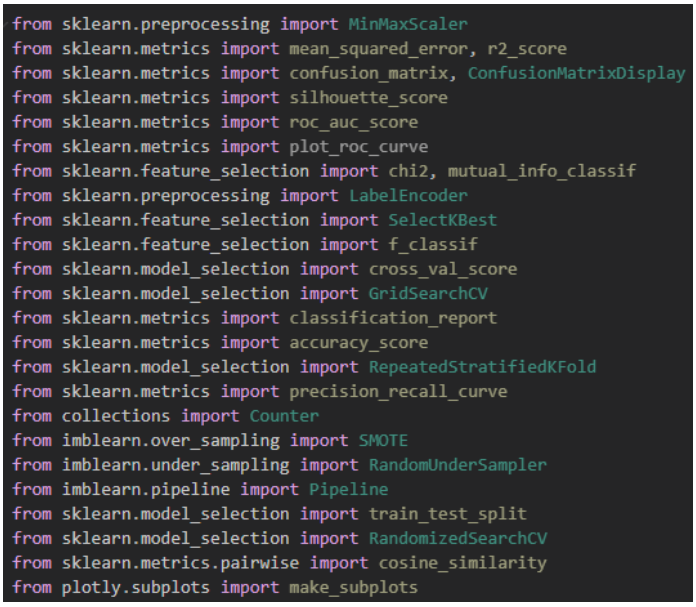
Para hacerse de los datos en el lenguaje de programación, realizar una revisión de los datos y posteriormente desarrollar modelos de Machine Learning, se aplicaron ciertas librerías de uso común en este tipo de trabajos, siendo algunas de ellas: Pandas (librería especializada en el manejo y análisis de estructuras de datos), Matplotlib (especializada en la creación de gráficos), Seaborn (especializada en visualización de datos basada en Matplotlib), Numpy (especializada en el cálculo numérico), Sklearn (librería útil de *Machine Learning* proporciona algoritmos de aprendizaje supervisado y no supervisado) y librerías propias de los algoritmos explorados. Asimismo, se investigaron e implementaron librerías como ser: Pickle (para transformar los modelos en una cadena de bytes única que puede ser guardada en un archivo) y Folium (herramienta de visualización que permite la concepción de mapas interactivos).

A continuación, se exponen algunas líneas de código para importación de librerías aplicadas en el proyecto.

```
# importacion de las librerias necesiarias:

import imblearn
import pickle
import folium
import os
import warnings
import seaborn as sns
import pandas as pd
import numpy as np
import xgboost as xgb
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import imblearn
%matplotlib inline
pd.options.display.float_format = '{:.2f}'.format
warnings.filterwarnings('ignore')
```

Ilustración 11 - Importación de librerías necesarias (1)



```
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import silhouette_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import plot_roc_curve
from sklearn.feature_selection import chi2, mutual_info_classif
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.metrics import precision_recall_curve
from collections import Counter
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import RandomUnderSampler
from imblearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics.pairwise import cosine_similarity
from plotly.subplots import make_subplots
```

Ilustración 12 - Importación de librerías necesarias (2)

En lo que respecta a la importación de los datos, se importó un archivo CSV, generado previamente en Power Query, utilizando la librería “Pandas”, a través del comando:

df = pd.read_csv (“[nombre del archivo]”)

Una vez que se importó el conjunto de datos en Python, se ejecutó una serie de comandos para visualizar el mismo y así tomar conocimiento de la dimensión del dataset, su estructura y contenido general (cantidad y denominación de columnas), tipo de variables, valores únicos de algunas variables (por ejemplo, tipos de empresas) y valores nulos.

Algunos de los comandos que se ejecutaron inicialmente son:

df.shape() para observar la dimensión del dataset.

Df.info() permite visualizar la información del dataset.

Df.column() expone las columnas que tiene el dataset.

Df.rename() se utiliza para renombrar columnas.

Df.isnull() una forma de observar si el dataset cuenta con valores nulos.

Como parte del análisis exploratorio se observaron valores únicos dentro de una misma columna, esto fue realizado en el proyecto con el comando ***Df[Nombre_columna].unique()***.

Por otro lado, fue realizado en esta etapa un análisis de los valores medios de las variables del dataset con respecto a la variable objetivo (“Churn”). En el proyecto aquí descripto, fue posible observar, por ejemplo, que el valor medio de la variable “meses desde la última factura” era mayor en el caso de los clientes que finalizan la relación comercial, con respecto a los clientes que se quedaban; situación que parece coherente, debido a que la empresa informa que la etiqueta Churn se coloca en función del tiempo de inactividad en base a la facturación del cliente. Asimismo, se visualizó que los valores medios de la "cantidad de incidentes reportados” era mayor en los que se quedaban que en los que se iban, situación que puede denotar una inconformidad en cuanto al servicio, dado que se seguían generando incidentes. En lo que respecta a “tiempo promedio de respuesta” el valor medio para los clientes que se quedaban era mayor que de los clientes que se iban, situación similar pasa con “cantidad de incidentes”.

En cuanto a la variable “cantidad de reclamos” el valor medio era mayor en los clientes que se quedaban que en lo que se iban. Lo que permite intuir que los clientes que se quedan tienen un vínculo más fluido con la empresa ya sea al realizar reclamos y/o reportando incidentes.

4.3.2. Análisis de variables

El conjunto de datos contaba con variables de tipo *Integers* (números enteros), *Floating* (números que tienen residuos, es decir decimales) y *Object* (texto que describe la variable, Ej. País de Venta = Brasil).

Seguidamente, se expone como se distribuyeron las variables del *dataset* utilizado por tipo de variable.

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24951 entries, 0 to 24950
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   País de Venta                   24951 non-null  object
 1   Vertical                        24951 non-null  object
 2   Meses desde última Factura      24951 non-null  int64
 3   Cant de equipos arrendados      24951 non-null  int64
 4   Antiguedad_de_Clientes          24951 non-null  float64
 5   Casos abiertos en ult 12 meses  24951 non-null  int64
 6   Casos abiertos                  24951 non-null  int64
 7   Cant_de_comunic_últimos_24_meses 24951 non-null  int64
 8   Encuestas                       24951 non-null  object
 9   Ren_aut_de_contrato             24951 non-null  float64
 10  Categoría_de_facturación        24951 non-null  object
 11  Cantidad_de_Incidentes          24951 non-null  int64
 12  Cantidad_de_Reclamos            24951 non-null  int64
 13  Tiempo promedio de respuesta    24951 non-null  float64
 14  Tiempo_de_respuesta             24951 non-null  float64
 15  Churn                           24951 non-null  int64
dtypes: float64(4), int64(8), object(4)
```

Ilustración 13 – Listado de tipos de variables

Continuando con la exploración de los datos, para el posterior desarrollo de modelos de Machine Learning, se estudió la distribución de la variable objetivo Churn, lo que permitió descubrir que el *dataset* se encontraba desbalanceado, ya que los clientes que se fueron eran aproximadamente el 21% y los que se quedaron representaban un 79% del conjunto. Se puede decir que se presentaba una relación 4 a 1 de clientes que se quedan en relación a los que se van, lo que provocaría, sin una técnica de balance de datos, que

las predicciones fueran sesgadas hacia los clientes que no abandonan debido al nivel predominante de datos para esta categoría.

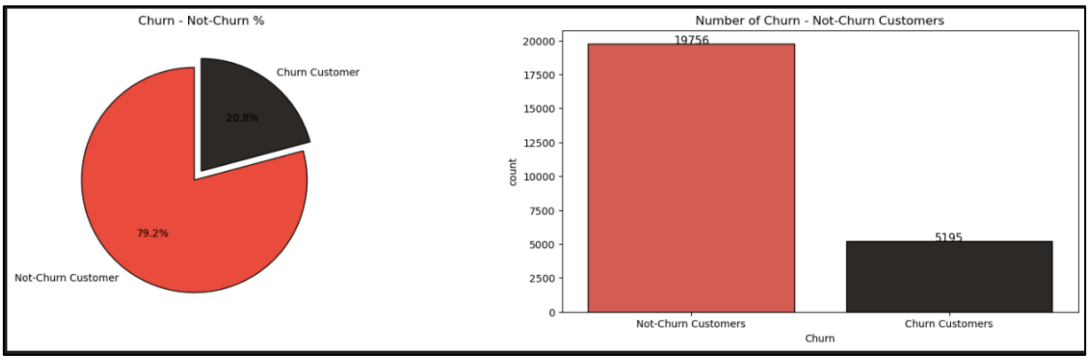


Ilustración 14 – Gráfico de distribución del Churn

Como parte del análisis de variables, se ejecutaron líneas de comando para detectar *outliers* en algunas variables que resultan de interés para el desarrollo de este proyecto.

Ilustración 15 – Gráfico de *outliers* de Equipos Arrendados, Incidentes y Reclamos

Si bien se detectaron varios *outliers* en las variables “Equipos Arrendados”, “Incidentes” y “Reclamos”, se consideró adecuado mantenerlos en el conjunto de datos, debido a que una mayor cantidad de equipos arrendados puede denotar que se trata de un cliente

importante para la empresa, así como también, a mayor cantidad de equipos, mayor cantidad de reclamos e incidentes, cuando estos se registran por equipos afectados.

Por último, en lo que respecta a la exploración de las variables del *dataset*, se estudió la correlación de las variables con respecto a la variable objetivo. Con esto se obtuvo que la variable "Meses desde la última factura" se encontraba altamente correlacionada con la variable objetivo, lo cual hace sentido, debido a que la etiqueta Churn la empresa la asigna en función del tiempo desde que no se emiten facturas por servicios al cliente. Situación que da indicios de que dicha variable debe ser extraída del conjunto para el desarrollo de los modelos de predicción, por la fuerte correlación con la variable objetivo que potencia la posibilidad de distorsiones en el análisis.

A continuación, se expone la matriz de correlación resultante:

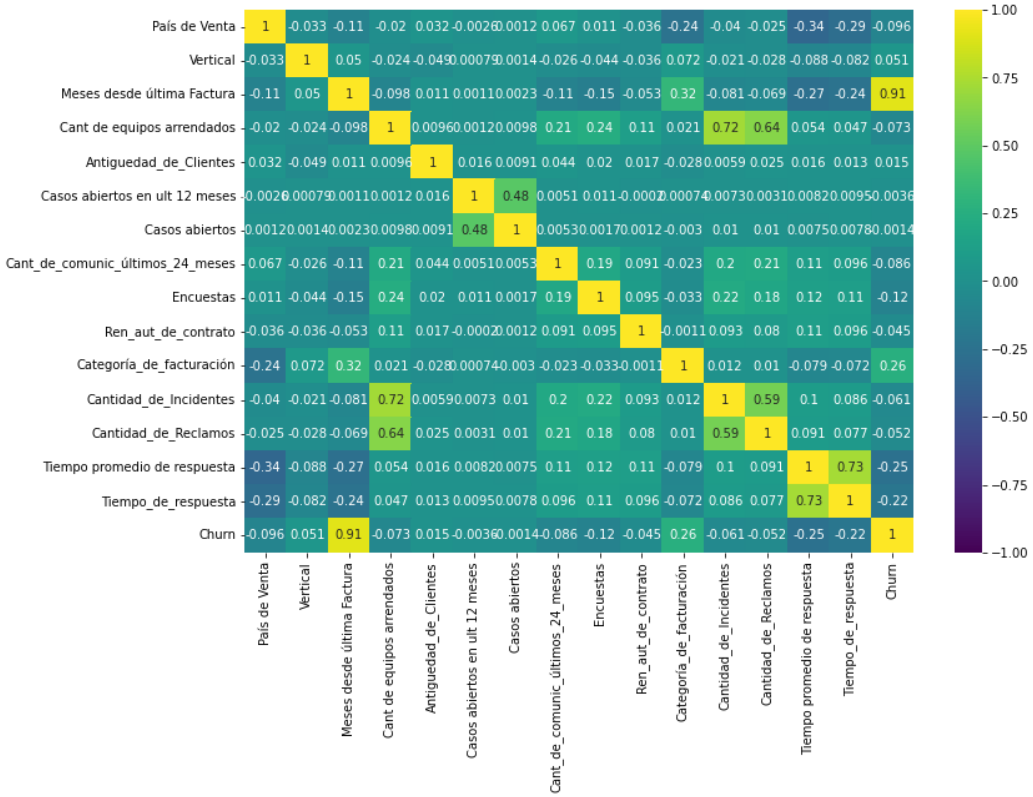


Ilustración 16 – Matriz de correlación entre variables

Adicionalmente se observó una fuerte correlación entre tiempo promedio de respuesta y respuesta (0.73), lo cual indica coherencia de los datos; entre cantidad de equipos arrendados, con cantidad de incidentes y reclamos (0.72 y 0.64 respectivamente), lo que

indica cierta confirmación a lo mencionado de que los incidentes y reclamos se abren por equipo y no por cliente en sí.

Finalmente se observó una correlación de 0.59 entre reclamos e incidentes.

4.3.3. Transformaciones necesarias

Tras consolidar todos los diferentes *datasets* y ejecutar una exploración del conjunto de datos unificado y sus variables, se presentaron algunos valores N/A en varias de las columnas. La mayoría de los casos se correspondían a que no todos los clientes del *dataset Accounts* están incluidos en los demás *datasets*, los otros casos simplemente son faltantes de los datos extraídos originalmente.

En el caso de la columna "Categoría de Facturación", se etiquetaron los clientes en cuarto grupos iguales: "Facturación muy baja", "Facturación baja", "Facturación alta", "Facturación muy alta", en base a la distribución de los datos de facturación de los últimos 36 meses. Los valores N/A y los montos menores a 0 fueron categorizados como "Facturación muy baja".

Las columnas "Tiempo de respuesta promedio" y "Fracción a tiempo", debieron ser complementadas con datos generados a partir de herramientas informáticas (Power Query). Se generaron datos aleatorios siguiendo una distribución normal creada a partir de los datos originales del *dataset*.

Para la columna "Encuestas", se promediaron las encuestas no-nulas y se categorizaron en "Negativas" para los valores menores a 50, "Indiferentes" para los valores entre 50 y 75 y "Positivas" para las mayores a 75.

La columna "Antigüedad" se generó a partir de la diferencia entre la fecha de creación del contacto en el CRM a la fecha de descarga de los datos.

En lo que respecta a las columnas: Cantidad de equipos arrendados, Cantidad de incidentes, Cantidad de reclamos, Renovación automática de contrato y Cantidad de comunicaciones en los últimos 24 meses; se reemplazaron los valores N/A por 0.

Para las columnas: País de Venta, Vertical y Encuestas; luego de indagaciones con la empresa y análisis del conjunto de datos, se concluyó que, para complementar los datos faltantes, es adecuado utilizar la funcionalidad que reemplaza los valores N/A por el valor de la fila superior, incluida en Power Query.

El diagrama del Dataflow, donde de detallan las transformaciones que debieron ser realizadas, se encuentra en la sección Anexos (10.1 – Dataflow).

Posteriormente a las transformaciones, se verificó que no hubiera valores nulos de forma gráfica, como la que se expone a continuación, el resultado es un gráfico completamente negro que indica que el dataset importado no cuenta con valores nulos.

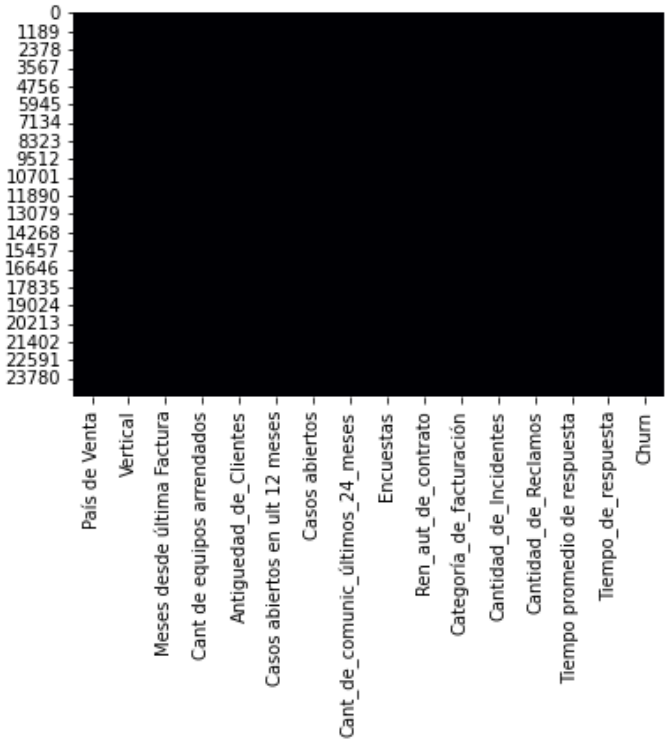


Ilustración 17 - Resultado gráfico sin valores nulos

Asimismo, se aplicó la técnica de ANOVA TEST para detectar variables poco significativas, siendo el resultado la detección de 4 variables no significativas.

En última instancia, como parte de las transformaciones y supuestos tomados, como se menciona en el apartado anterior, el *dataset* proporcionado se encontraba notoriamente

desbalanceado, por lo que se debió aplicar alguna técnica de balanceo, de forma tal de mitigar su efecto en la posterior predicción y lograr óptimos resultados.

Es por ello que, en este proyecto, se implementó la técnica de SMOTE, la cual consta de aumentar las muestras minoritarias de la variable de destino a las muestras mayoritarias.

```
over = SMOTE(sampling_strategy=1)

f1 = df2.iloc[:, :11].values
t1 = df2.iloc[:, 11].values

f1, t1 = over.fit_resample(f1, t1)
Counter(t1)
```

```
Counter({0: 19756, 1: 19756})
```

Ilustración 18 – Aplicación de la técnica Smote

4.4. Aprendizaje automático

4.4.1. Curva ROC

Es una herramienta que permite evaluar el rendimiento de los clasificadores binarios. Esta indica de manera visual la relación entre la precisión y la sensibilidad de un modelo, sirviendo para comparar el rendimiento de distintos modelos de clasificación.

La curva de ROC representa el “True Positive Rate” (TPR) en el eje “y” y el “False Positive Rate” (FPR) en el eje “x”. Cada punto en la curva ROC representa una configuración diferente del umbral de decisión del modelo. A medida que el umbral de decisión se desplaza hacia la derecha (es decir, se vuelve menos estricto), aumenta la tasa de falsos positivos y la tasa de verdaderos positivos.

Idealmente, un modelo de clasificación binario que es perfecto tendría un TPR del 100% y un FPR del 0%. Esto se traduciría en un punto en la esquina superior izquierda de la curva ROC. Un modelo cuyo rendimiento es completamente aleatorio tendría una curva ROC que es una línea diagonal, desde la esquina inferior izquierda hasta la esquina superior derecha.

La curva ROC también puede ser utilizada para comparar el rendimiento de diferentes modelos de clasificación binaria. En general, un modelo con una curva ROC que se encuentra más cerca de la esquina superior izquierda se considera mejor que un modelo con una curva ROC que se encuentra más cerca de la línea diagonal.

4.4.2. Particionado *Train/Test*

Previo a la exploración de los hiperparámetros en los diferentes modelos, se realizó una exploración del particionado del *dataset* en *Train* y *Test*, en los siguientes intervalos:

PARTICIÓN
Train 80 -Test 20
Train 70 -Test 30
Train 75 -Test 25
Train 85 -Test 15
Train 90 -Test 10

Tabla 3 - Particionado *Train* y *Test*

Si bien se realizaron análisis con una partición del 90% - 10%, se entendió que el tamaño de Test podría ser muy bajo para verificar el resultado obtenido con datos desconocidos, ajustar los parámetros y seleccionar los algoritmos adecuados.

En las pruebas realizadas inicialmente con los modelos bases, los mejores resultados se obtuvieron con una partición de Train=90% y Test=10%, seguido de las particiones de Train=85% y Test=15% y Train=80% y Test=20%.

Tal como se comentó inicialmente, para evitar caer en sobreajuste, se descartó el particionado Train=90% y Test=10%, para las demás pruebas y se concluyó que, dado los resultados obtenidos y que no se observaron diferencias significativas, es conveniente continuar con el entrenamiento con la partición más frecuentemente utilizada de Train 80% y Test 20% en este tipo de análisis.

A continuación, se exponen algunas de las pruebas realizadas que reflejan lo anteriormente detallado:

Algoritmo	Dataset	Manejo de desbalance	PARTICIÓN	CV Score	ROC ACU Score
RF	Completo	SMOTE	Train 70 - Test 30	95,593	89,978
	Completo	SMOTE	Train 75 - Test 25	95,754	89,583
	Completo	SMOTE	Train 80 - Test 20	95,86	89,52
	Completo	SMOTE	Train 85 - Test 15	95,876	89,742
	Completo	SMOTE	Train 90 - Test10	95,9	90
Algoritmo	**Dataset**	**Manejo de desbalance**	**PARTICIÓN**	**CV Score**	**ROC ACU Score**
Xgboost Classifier	Completo	SMOTE	Train 70 -Test 30	94,643	87,523
	Completo	SMOTE	Train 75 -Test 25	94,654	87,346
	Completo	SMOTE	Train 80 -Test 20	94,69	87,16
	Completo	SMOTE	Train 85 -Test 15	94,661	87,600
	Completo	SMOTE	Train 90 -Test 10	94,6	87,9
Algoritmo	**Dataset**	**Manejo de desbalance**	**PARTICIÓN**	**CV Score**	**ROC ACU Score**
LightGBM Classifier	Completo	SMOTE	Train 70 -Test 30	94,642	87,549
	Completo	SMOTE	Train 75 -Test 25	94,667	87,437
	Completo	SMOTE	Train 80 -Test 20	94,691	87,246
	Completo	SMOTE	Train 85 -Test 15	94,679	87,701
	Completo	SMOTE	Train 90 -Test 10	94,653	87,955

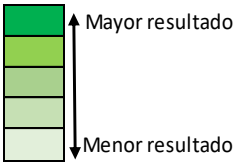


Tabla 4 - Comparación de resultados de pruebas realizadas

4.4.3. Modelos de Machine Learning

4.4.3.1. Árbol de Decisión

Como primer algoritmo de clasificación, se desarrolló Árbol de Decisión.

La finalidad de éste es clasificar un elemento en una categoría o tomar una decisión basada en una serie de características o variable.

Para este desarrollo, se utiliza la librería Sklearn, con los siguientes criterios en primera instancia:

```
#train the model
classifier_dt = DecisionTreeClassifier(random_state = 1000,max_depth = 4,min_samples_leaf = 1)
classifier_dt.fit(X_train, y_train)
classifier_dt.score(X_test, y_test)
```

Ilustración 19 - Algoritmo de Árboles de Decisión

- *Random state* = 1000 (para el control de la aleatoriedad del estimador).
- *Max_depth* = 4 (profundidad del árbol).
- *Min_samples_leaf* = 1 (Número mínimo de muestras para pertenecer al nodo hoja).

Posteriormente se graficó el árbol generado, lo que es importante para poder visualizar el algoritmo que se está aplicando.

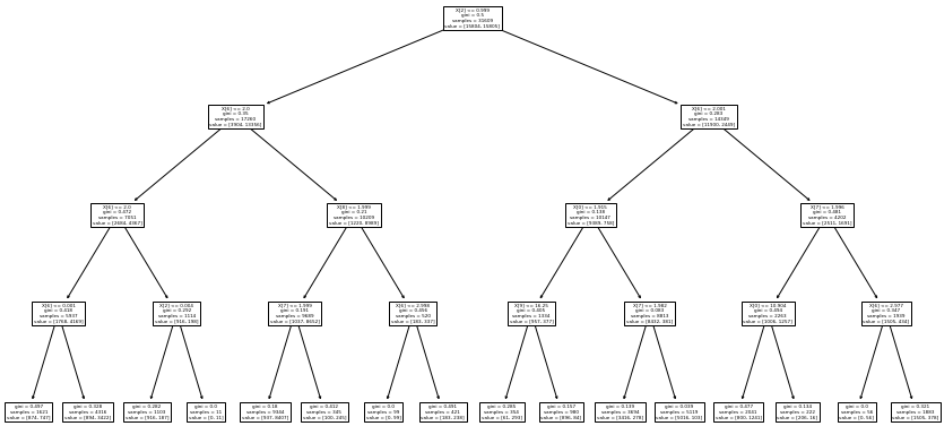


Ilustración 20 - Diagrama del Árboles de Decisión creado

Con la primera prueba del algoritmo ejecutado se logró obtener buenos resultados, que se visualizan claramente con el análisis de la curva de ROC.

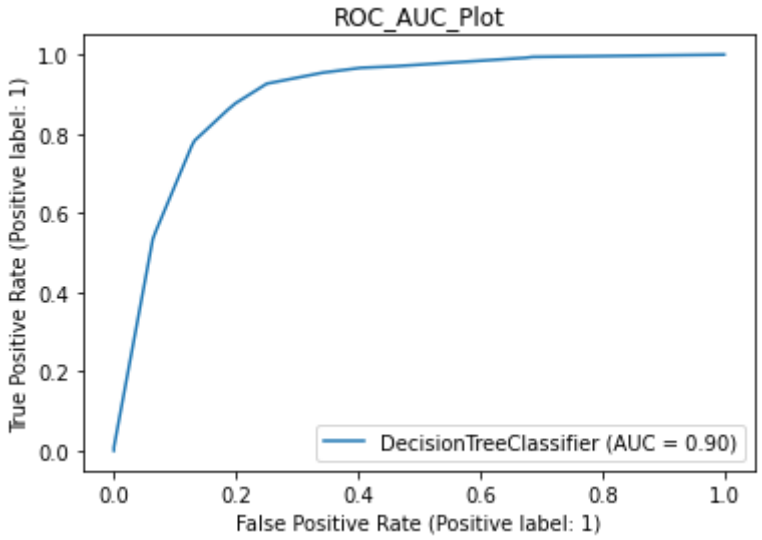


Ilustración 21 – Curva ROC - Primer resultado - Árboles de Decisión

Esta primera prueba se obtuvo una *performance* en "*Cross Validation*" de 90.50% y ROC_AUC de 84.13%.

Continuando con el desarrollo del algoritmo de Árboles de Decisión, se continuó mejorando los hiperparámetros del mismo. Para ello se intentó ver cuál es la profundidad óptima que se puede aplicar.

A lo largo de varias pruebas, se concluyó que es adecuado aplicar una profundidad de 5, porque desde ahí no se observaban mejoras sustanciales del modelo, así es como se comienza a incurrir la posibilidad de caer en sobreajuste.

En lugar de ir probando los diferentes hiperparámetros para aplicar al Árbol de Decisión, es posible aplicar técnicas de búsqueda de parámetros, como lo es GridSearchCV, la cual fue aplicada en este proyecto.

La mejor combinación de los hiperparámetros resultante fue:

```
print('Best Criterion:', clf_GS.best_estimator_.get_params()['criterion'])
print('Best max_depth:', clf_GS.best_estimator_.get_params()['max_depth'])
print('Best min_samples_split:',
      clf_GS.best_estimator_.get_params()['min_samples_split'])
print('Best min_samples_leaf:',
      clf_GS.best_estimator_.get_params()['min_samples_leaf'])
print('Best Number Of Components:',
      clf_GS.best_estimator_.get_params()['max_features'])
clf_GS.best_estimator_.get_params()
```

[64] ✓ 0.1s Python

```
Best Criterion: entropy
Best max_depth: 5
Best min_samples_split: 20
Best min_samples_leaf: 5
Best Number Of Components: 6
```

Ilustración 22 - Combinación de hiperparámetros - Árboles de Decisión

Una vez obtenida la mejor combinación de hiperparámetros, se procedió a ejecutar el comando de la curva Roc, tal como se expone a continuación:

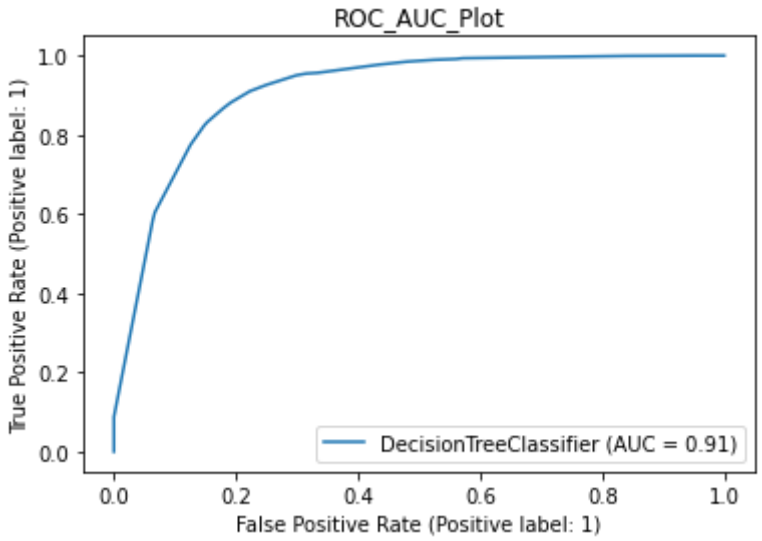


Ilustración 23 – Curva ROC - Segundo resultado - Árboles de Decisión

Al aplicar los resultados obtenidos del *GridSearchCV*, se logró obtener una leve mejora en el resultado de la Curva de Roc en "*Cross Validation*" 91.65% (inicialmente 90.50%) y ROC_AUC de 84.82% (84.13% en primera instancia).

4.4.3.2. Random Forest

Se implementó Random Forest con un random_state de 1.107, para controlar la aleatoriedad de arranque de las muestras utilizadas en la construcción del árbol. Para ello se ejecutó el siguiente comando:

```
classifier_rf = RandomForestClassifier(random_state=1107)

# random_state = controla la aleatoriedad de arranque de las muestras utilizadas en la
construcción del árbol.
```

Ilustración 24 - Algoritmo de Random Forest

Cuando se observó la *performance*, caso quedó evidencia una mejoría en los resultados, en comparación con los modelos anteriores de Árboles de Decisión, ya que en "Cross Validation" obtuvo un resultado de 95.86% y en ROC_AUC de 89.69%.

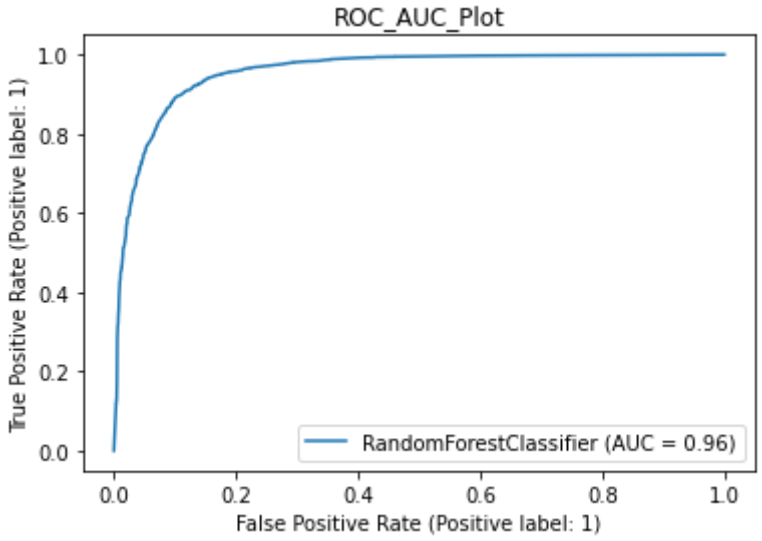


Ilustración 25 – Curva ROC - Primer resultado - Random Forest

Debido a la mejoría observada con la primera prueba en Random Forest, es que se continuó el trabajo con el algoritmo para optimizarlo a través de la mejora de los hiperparámetros del modelo.

Investigamos hiperparámetros para seleccionar el mejor bosque.

```
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start=200, stop=2000, num=10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num=11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Criterio
criterion = ['gini', 'entropy']
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               # 'max_features': max_features, # Son muy pocas variables por lo cual no vale
               la pena aplicarlo
```

Ilustración 26 - Combinación de hiperparámetros - Random Forest

Luego del proceso de investigación de configuración de hiperparámetros a utilizar, se aplicó la mejor combinación de ellos obtenida, fue nombrada “Modelo final de Random Forest”.

```
# usando la mejor combinación de hiperparámetros, estimo modelo final
best_RF = RandomForestClassifier(**best_param)
```

Ilustración 27 - Modelo final de Random Forest

El mejor modelo obtenido en base a la investigación fue:

```
best_RF=RandomForestClassifier(random_state=1107, n_estimators=100, min_samples_split=2,
min_samples_leaf=2, max_depth=100, criterion='entropy', bootstrap=True)
```

Ilustración 28 - Configuración del Modelo final de Random Forest

Y los resultados obtenidos fueron:

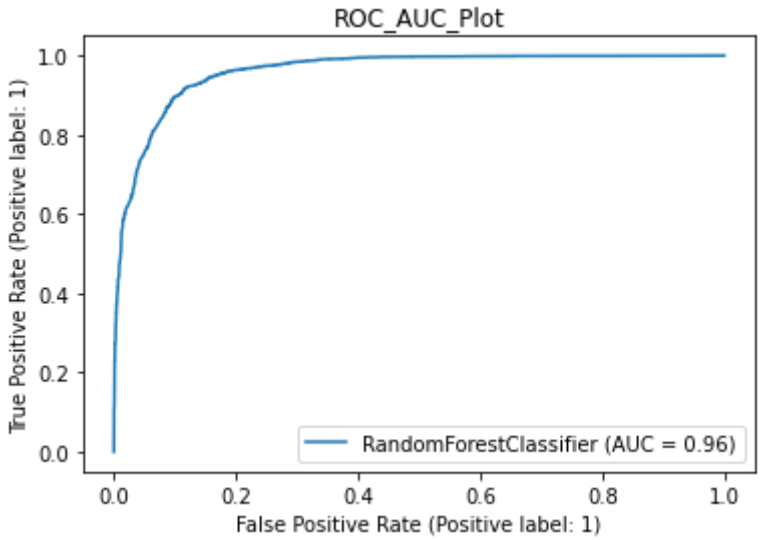


Ilustración 29 – Curva ROC - Segundo resultado - Random Forest

Como es posible observar, con los mejores hiperparámetros los resultados fueron similares a la primera prueba en Random Forest. Incurrir en costos computacionales para la búsqueda y la aplicación de un modelo más complejo en cuanto a sus parámetros, no mejora de manera significativa el modelo.

4.4.3.3. Gradient Boosting

XG BOOST (extreme Gradient Boosting) es un algoritmo de aprendizaje automático supervisado utilizado principalmente para problemas de clasificación y regresión. Es una extensión de algoritmo Gradient *Boosting*, que es un método de ensamblaje de Árboles de Decisión.

XGBoost funciona entrenando un conjunto de modelo de Árboles de Decisión débiles y combinándolos para formar un modelo de ensamble más fuerte. Cada árbol se entrena para predecir la variable objetivo utilizando las características del conjunto de datos y los errores residuales del árbol anterior.

El primer modelo de XGBoost que fue aplicado es el siguiente:

```
# Fit a Gradient Boosting model
# primer modelo que aplicamos:
classifier_xgb = xgb.XGBClassifier(
    n_estimators=1000, learning_rate=0.01, max_depth=3, random_state=0)
#classifier_xgb.fit(X_train, y_train)
#y_pred = classifier_xgb.predict(X_test)
#accuracy_score(y_test, y_pred)
```

Ilustración 30 - Algoritmo Gradient Boosting Machine

Los parámetros que se configuraron fueron:

- *n_estimators*= 1000 - número de etapas de refuerzo a realizar. El aumento del gradiente es bastante resistente al sobreajuste, por lo que un gran número generalmente da como resultado un mejor rendimiento.
- *learning_rate* = 0,01 - tasa de aprendizaje que determina cuanto se ajustan los pesos de los árboles en cada iteración. Un valor bajo como el aplicado, significa que los árboles aprenden lentamente, lo que puede resultar en un mejor rendimiento a largo plazo, pero también puede hacer que el algoritmo sea más lento en converger.
- *Max_depth*=3 - profundidad máxima de los estimadores de regresión individuales. La profundidad máxima limita el número de nodos en el árbol. Se ajusta este parámetro para obtener un mejor rendimiento; el mejor valor depende de la interacción de las variables de entrada. Si es “ninguno”, los nodos se expanden hasta que todas las hojas sean puras o hasta que todas las hojas contengan menos de min_samples_split samples. Si es *Integers*, los valores deben estar en el rango [1, infinito)
- *Random_state*=0 - controla la semilla aleatoria dada a cada estimador de árbol en cada iteración de impulso. Además, controla la permutación aleatoria de las funciones en cada división, así como la división aleatoria de los datos de entrenamiento para obtener un conjunto de validación.

La primera prueba con este algoritmo arrojó un buen resultado, tanto en Cross Validation Score, con 94.61% como en ROC_AUC Score de 87.59%. Si bien el resultado fue bueno, no fue el mejor de los obtenidos.

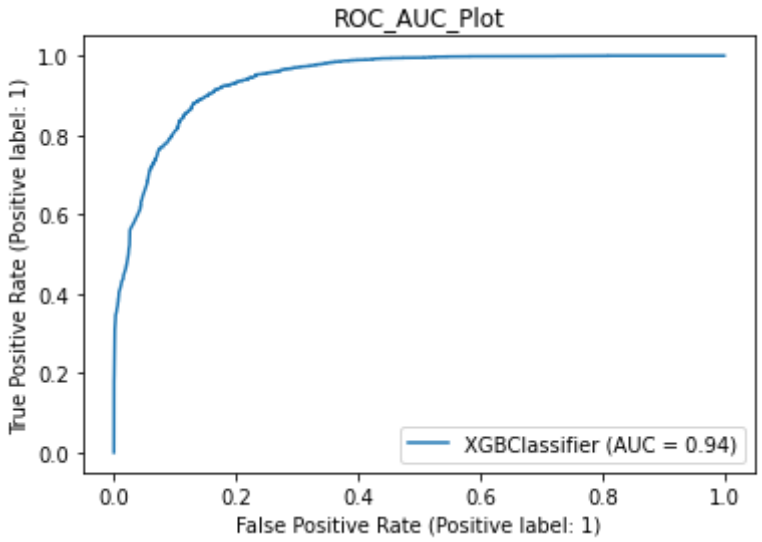


Ilustración 31 – Curva ROC - Primer resultado - Gradient Boosting Machine

A partir del resultado anteriormente, se continuó con la mejora del algoritmo a través de los hiperparámetros.

```
classifier_xgb_1 = xgb.XGBClassifier()
parameters = {
    "eta": [0.05, 0.10, 0.15, 0.20, 0.25, 0.30],
    "max_depth": [3, 4, 5, 6, 8, 10, 12, 15],
    "min_child_weight": [1, 3, 5, 7],
    "gamma": [0.0, 0.1, 0.2, 0.3, 0.4],
    "colsample_bytree": [0.3, 0.4, 0.5, 0.7]
}

grid = GridSearchCV(classifier_xgb_1,
                    parameters, n_jobs=4,
                    scoring="neg_log_loss",
                    cv=3)


grid.fit(X_train, y_train)
```

Ilustración 32 - Combinación de hiperparámetros - Gradient Boosting Machine

Con búsqueda expuesta en la Ilustración 32, se observó una mejora del resultado, dado que la curva de Roc en Cross Validation Score obtuvo un 96.49%, y en ROC_AUC Score dio 89.62%.

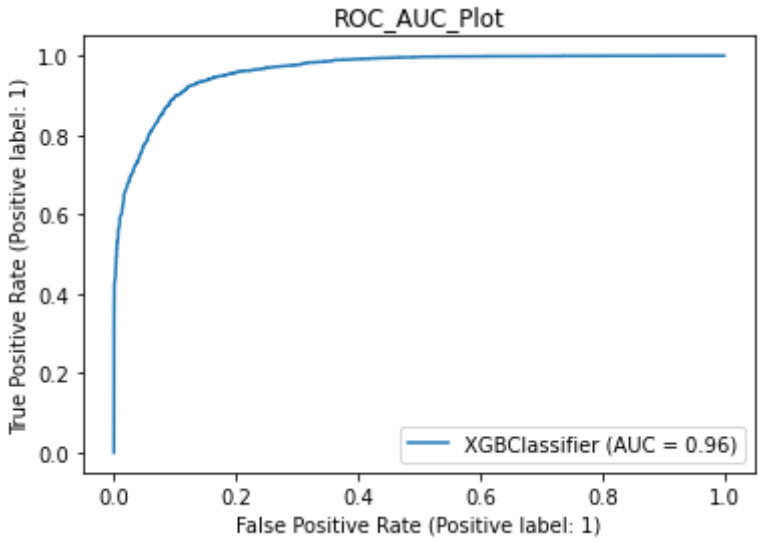


Ilustración 33 – Curva ROC - Segundo resultado - Gradient Boosting Machine

Por otro lado, dentro de lo que es Gradient Boosting, existe LightGBM Classifier, que es un marco de trabajo de aprendizaje automático de código abierto diseñado para el aprendizaje basado en árboles. Fue desarrollado por Microsoft y utiliza un algoritmo de aumento del gradiente que se basa en la frecuencia de la característica para construir un árbol de decisión. LightGBM se diferencia de otros algoritmos de aprendizaje basados en árboles en que utiliza un enfoque de partición vertical que divide los datos en columnas en lugar de filas. Esto permite una mayor eficiencia en el tiempo de entrenamiento y una mayor precisión en los conjuntos de datos grandes.

En la aplicación de la predicción del Churn para clientes de Ricoh, la utilización del algoritmo permitió obtener un buen resultado generado.

La configuración realizada y el resultado obtenido se exponen en las próximas 2 ilustraciones:

```
# LightGBM Classifier:
classifier_lgbm = LGBMClassifier(learning_rate= 0.01,max_depth = 3,n_estimators = 1000)
```

Ilustración 34 - Algoritmo LightGBM Classifier

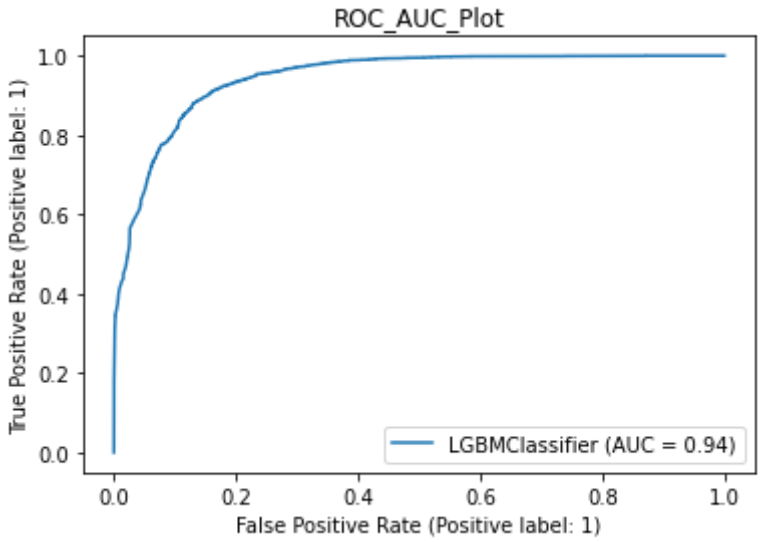


Ilustración 35 – Curva ROC - Resultado LightGBM Classifier

En Cross Validation se visualiza un resultado de 94.63% y en ROC_AUC Score un 87.56%. El cual no fue mejor al inicialmente obtenido en este modelo, pero igualmente reporta un resultado aceptable para el caso.

4.4.3.4. Stacking

En el proyecto para Ricoh, los modelos combinados fueron Árbol de Decisión, Random Forest, XGB Boost, y LightGBM. Dentro de estos, los modelos con mejores resultados.

Dicha combinación dio como resultados Cross Validation Score 96.81% y ROC_AUC Score: 90.05%.

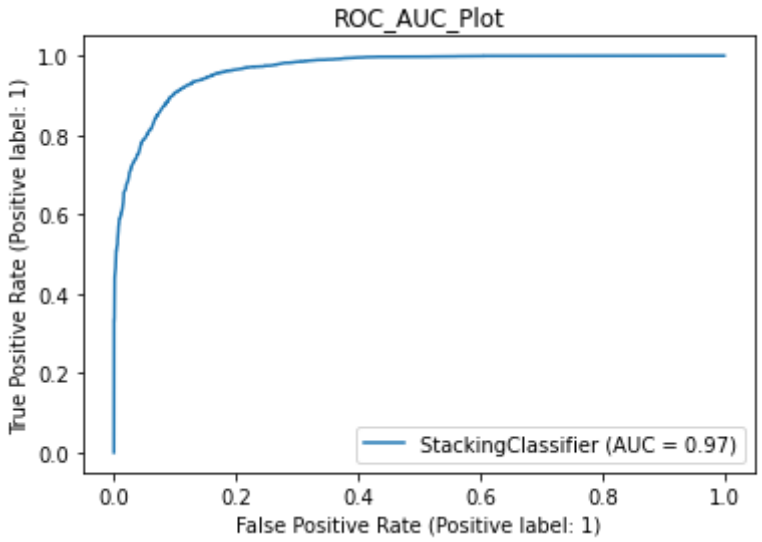


Ilustración 36 – Curva ROC - Resultado Stacking

En conclusión, el modelo que arroja el mejor resultado es Stacking, pero no se diferenció mucho con los resultados obtenidos con Random Forest y XGBoost. Por otro lado, los modelos que reportaron los peores resultados son Árbol de Decisión y el LightGBM.

Se aconseja como trabajos futuros, la investigación y continuidad de la mejora de hiperparámetros en LightGBM, y así obtener una optimización de su *performance*.

4.5. Predicción del Churn Rate

Luego de entrenar el *dataset* con varios modelos y seleccionar el que mejor performance obtuvo (Stacking), se procedió con la carga del conjunto de datos previo en una variable, llamada “data”:

```
data = df2.drop(['Churn'], axis=1)
```

Ilustración 37 - Carga del conjunto de datos previo en una variable

Se crearon las etiquetas que se desean predecir en una variable separada llamada, en nuestro caso, “*labels*”.

Se utilizó el modelo entrenado para hacer predicciones en el conjunto de datos anterior. Esto se ejecutó con el comando “*predict”* del modelo.

```
predicted_labels = stack.predict(data)
```

Ilustración 38 – Ejecución del comando “predict”

Se agregaron las predicciones al conjunto de datos anterior.

```
df_3 = pd.DataFrame(data)
df_3['predicted_labels'] = predicted_labels
```

Ilustración 39 - Carga de predicción en el conjunto de datos

Finalmente se guardó el conjunto de datos con las predicciones en un archivo “.csv”.

```
df_3.to_csv('predict_best.csv', index=False)
```

Ilustración 40 - Almacenamiento del conjunto con predicción en un archivo ".csv"

4.5.1. Planilla de resultados

Algoritmo	Manejo de desbalance	Seteo	CV Score en %	ROC ACU Score en %
Árboles de Decisión	SMOTE	random_state=1000, max_depth=4, min_samples_leaf=1 Demás parámetros por defecto	90,5	84,13
Árboles de Decisión_dt_1	SMOTE	random_state=1000, criterion='entropy', max_depth=5, min_samples_leaf=5, min_samples_split=20 Demás parámetros por defecto	91,65	84,82
Random Forest	SMOTE	n_estimators: int = 100, criterion: str = "gini", max_depth: Any | None = None, min_samples_split: int = 2, min_samples_leaf: int = 1, min_weight_fraction_leaf: float = 0, max_features: str = "sqrt", max_leaf_nodes: Any | None = None, min_impurity_decrease: float = 0, bootstrap: bool = True, oob_score: bool = False, n_jobs: Any | None = None, random_state: Any | None = None, verbose: int = 0, warm_start: bool = False, class_weight: Any | None = None, ccp_alpha: float = 0, max_samples: Any | None = None	95,86	89,69
Random Forest_best_RF	SMOTE	n_estimators: 1200, min_samples_split: 5, min_samples_leaf: 2, max_depth: 90, criterion: 'gini', bootstrap: False Demás parámetros por defecto	96,24	89,9
XGBoost	SMOTE	n_estimators=1000, learning_rate=0.01 max_depth=3, random_state=0 Demás parámetros por defecto	94,61	87,59
XGBoost_xgb_1	SMOTE	"eta": [0.05, 0.10, 0.15, 0.20, 0.25, 0.30], "max_depth": [3, 4, 5, 6, 8, 10, 12, 15], "min_child_weight": [1, 3, 5, 7], "gamma": [0.0, 0.1, 0.2, 0.3, 0.4], "colsample_bytree": [0.3, 0.4, 0.5, 0.7] Demás parámetros por defecto	96,49	89,62
Light GBM	SMOTE	learning_rate= 0.01, max_depth = 3, n_estimators = 1000 Demás parámetros por defecto	94,63	87,56
Staking	SMOTE	dt_1, best_RF, xgb_1 Light GBM	96,81	90,05

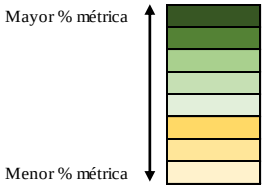


Tabla 5 – Planilla de resultados obtenidos

4.6. Clustering de clientes

El objetivo de la aplicación de Clustering es identificar las características en común en las observaciones que integran cada grupo o *cluster* y en base a esto, poder definir recomendaciones.

Las entradas al modelo de Clustering fueron un conjunto de variables numéricas normalizadas y el resultado final una etiqueta para cada observación que corresponde al *cluster* asignado.

Para realizar el Clustering se utilizó K-means, que es un algoritmo de agrupamiento no supervisado que toma un grupo de puntos no etiquetados e intenta agruparlos en un numero “K” de *clusters*, donde cada punto del grupo es similar entre sí.

Se utilizó este algoritmo por su popularidad y que es ampliamente utilizado debido a su simplicidad, escalabilidad, eficiencia y flexibilidad. A su vez, tiene asegurada la convergencia.

La “K” en K-means denota la cantidad de *clusters* que se desea tener al final. Por ejemplo, si K=4, se tendrán 4 *clusters* en el conjunto de datos.

Uno de los casos más comunes de utilización de *clusters* es la segmentación de clientes, lo cual se corresponde con el proyecto desarrollado.

Para la aplicación del algoritmo de clusterización, se aplicaron los siguientes pasos:

1) Importar las bibliotecas necesarias:

```
# 1) Importar las bibliotecas necesarias:

import imblearn
import pickle
import folium
import os
import warnings
import seaborn as sns
import pandas as pd
import numpy as np
import xgboost as xgb
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import imblearn
import statistics as stat
import gap_statistic
%matplotlib inline
pd.options.display.float_format = '{:.2f}'.format
warnings.filterwarnings('ignore')
```

Ilustración 41 - Importación librerías para Clustering

2) Cargar datos en un *dataframe* de pandas:

```
# 2) Cargar los datos en un DataFrame de pandas:

df = pd.read_csv("C:/Users/Administrador/Desktop/respaldo disco/BIGDATA/Postgrado ORT/TESIS
FINAL/Python_tesis/Proyecto/Completo 20230121.csv", encoding='latin-1')
df.head(3)
```

Ilustración 42 - Carga de datos en *dataframe* de pandas

3) Codificar las variables categóricas con el uso de LabelEncoder:

LabelEncoder es una clase de la biblioteca Scikit-learn de Python que se utiliza para convertir etiquetas categóricas en valores numéricos. En otras palabras, el LabelEncoder asigna un número a cada categoría única en una columna de datos categóricos.

Para poder usar tantas variables categóricas en nuestro modelo de clusterización, fue necesario codificarlas previamente, por lo que se utilizó Label Encoder. Se prefirió utilizar este método sobre *one-hot-encoding* por la cantidad de categorías y variables a ser codificada, a pesar de que las variables no sean ordinales.

```
# 3) Codificar las variables categóricas utilizando LabelEncoder:

le = LabelEncoder()

df2 = df1.copy(deep=True)
text_data_features = [i for i in list(
    df1.columns) if i not in list(df1.describe().columns)]

print('Label Encoder Transformation')
for i in text_data_features:
    df2[i] = le.fit_transform(df2[i])
    print(i, ' : ', df2[i].unique(), ' = ',
          le.inverse_transform(df2[i].unique()))
```

Ilustración 43 - Codificación de variables categóricas

4) Normalizar las variables con el uso de StandardScaler:

Lo que se hizo fue utilizar la técnica de StandardScaler la cual es útil en el procesamiento de datos para normalizar y escalar características numéricas, lo que puede mejorar el rendimiento y la precisión de muchos algoritmos de aprendizaje automático.

Al obtener solamente las variables numéricas, se aplicó la normalización de estas. El objetivo de la normalización es transformar los datos para que tengan una distribución normal con una media de cero y una desviación estándar de uno. Esto ayuda a garantizar que las variables no estén sesgadas y tengan la misma escala.

```
# 4) Normalizar las variables utilizando StandardScaler:
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df2)
data_scaled
```

Ilustración 44 - Normalización de variables con Standardscaler

5) Determinar del número óptimo de *clusters* con el método del codo:

El método Elbow o Codo, consiste en ejecutar el algoritmo de C*lustering* con diferentes valores de K (número de *clusters*) y graficar la suma de las distancias cuadradas de cada punto al centro de su *cluster* más cercano. El punto en el gráfico donde la curva tiene una forma de "codo" o donde la reducción de la distancia entre los puntos y su centro de *cluster* se desacelera significativamente, es ahí en donde está considerado el número óptimo de *clusters*.

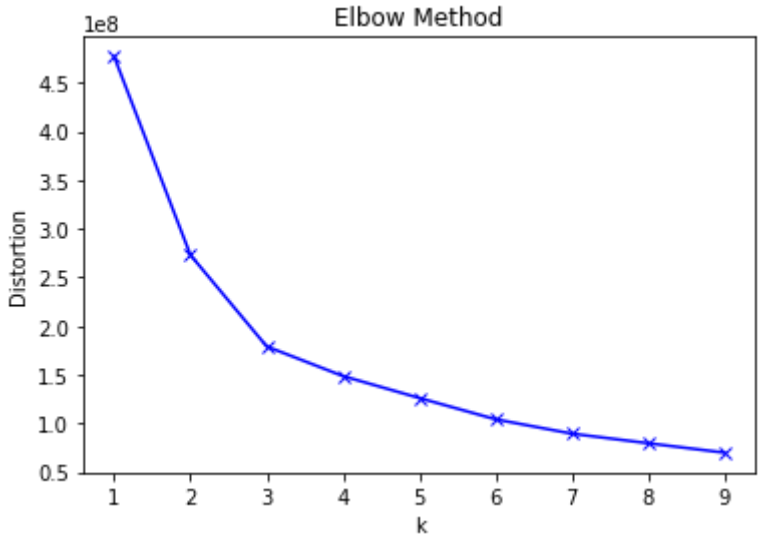


Ilustración 45 - K óptimo método del codo

El método Silhouette es una medida de cuán similar es un punto a su propio *cluster* en comparación con otros *clusters*. Este método implica ejecutar el algoritmo de Clustering con diferentes valores de K y graficar el coeficiente de Silhouette para cada uno. El número óptimo de *clusters* se corresponde con el valor de K que tiene el coeficiente de Silhouette más alto.

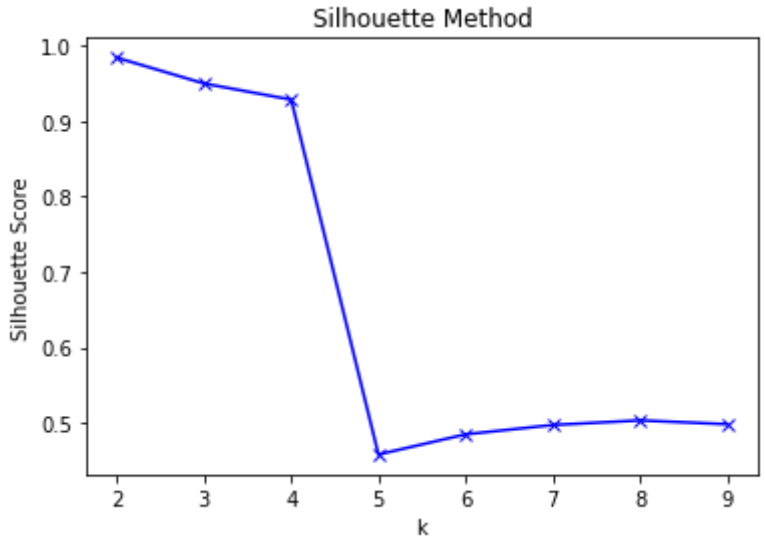


Ilustración 46 - K óptimo método de Silhouette

Se aplicaron estas dos (2) técnicas para comparar y seleccionar el K optimo.

Es importante tener en cuenta que no hay una única técnica que funcione en todos los casos. Estas son algunas de las disponibles para determinar el número óptimo de *clusters*

Dado el resultado obtenido, se decidió recurrir al método del Codo para determinar la cantidad de *clusters* óptimos.

Con el grafico del Codo, se concluyó que el número de *clusters* óptimo se encuentra en K=3. En base a ellos se definieron planes de acción para 3 grupos de clientes más ajustados a la realidad de los mismos.

6) Realizar del C*lustering* con el uso de K-Means con el número óptimo de *clusters*:

```
# Elegir el número de clusters optimo:

kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
# kmeans = KMeans(n_clusters=3)
0.0s
```

Ilustración 47 - K-Means con K óptimo

7) Agregar las etiquetas de *cluster* al conjunto de datos original:

```
df2['Cluster'] = kmeans.labels_
df2
```

Ilustración 48 - Etiqueta *cluster* en conjunto de datos original

8) Resultados Obtenidos:

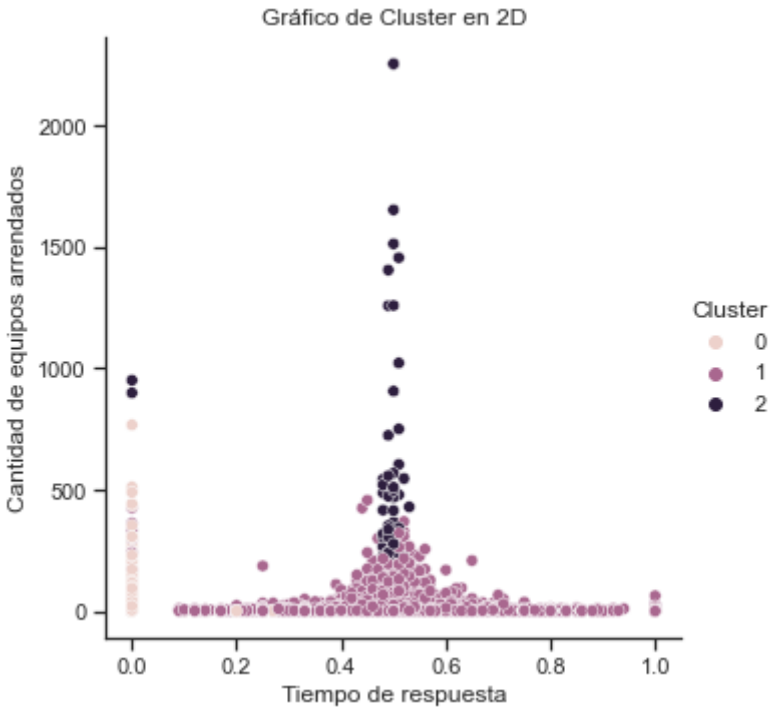


Ilustración 49 - Clustering – Cantidad Equipos Arrendados vs Tiempo de Respuesta

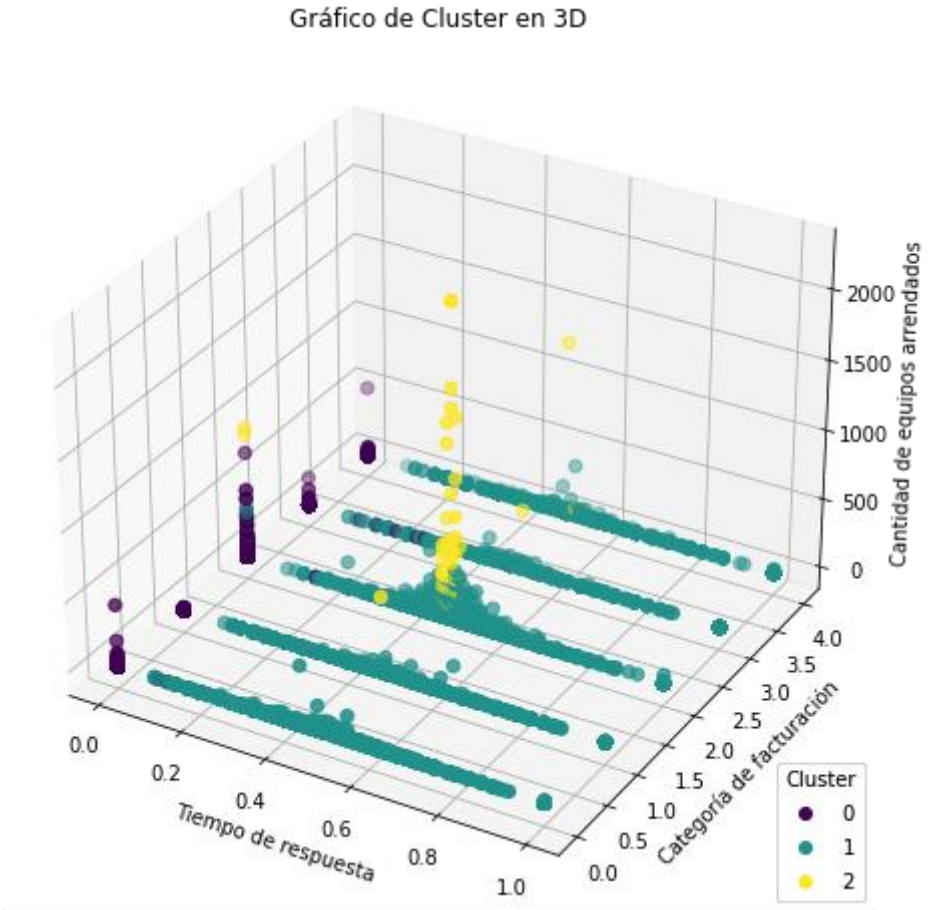


Ilustración 50 – Cantidad Equipos Arrendados vs Tiempo de Respuesta y Categoría de facturación

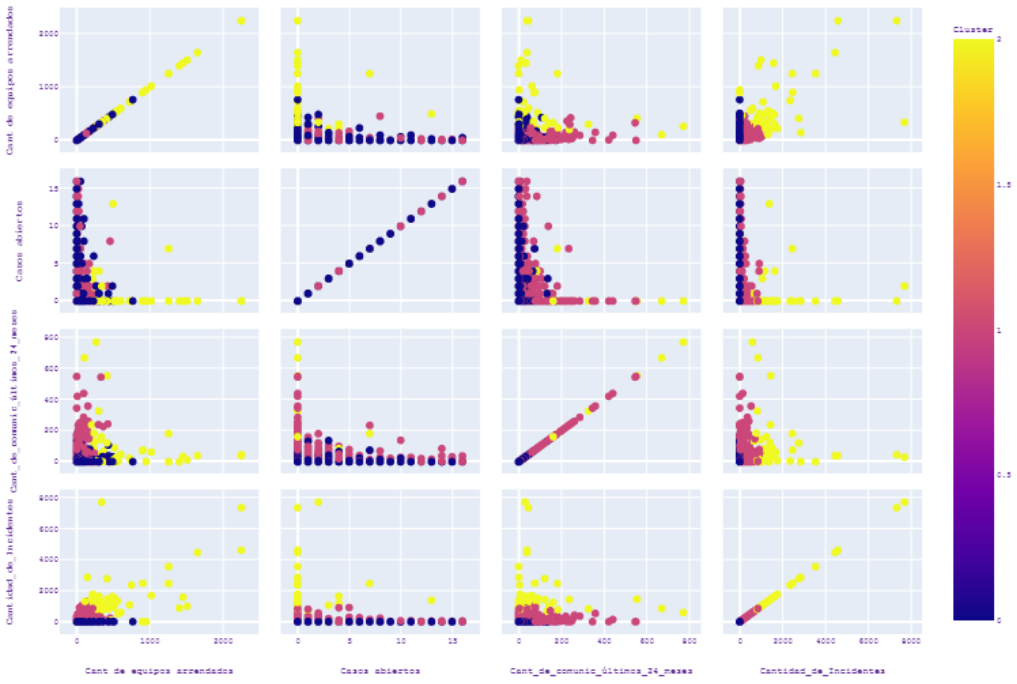


Ilustración 51 - Resultado Clustering – Sparse Matrix

9) Pasar a un CSV los resultados de *cluster*:

```
#·9)·pasar·a·un·csv·los·cluster:·
df2.to_csv('resultados_cluster_final.csv', index=False)
```

Ilustración 52 - Almacenamiento del resultado de Clustering en ".csv"

4.7. Conclusiones particulares

Una vez que se arribó a los resultados (tanto de Churn Rate como de Clustering), se volvió a realizar un análisis exploratorio para poder desarrollar conclusiones de los modelos aplicados, así como de la situación actual de la empresa.

Para mejorar la visualización de la analítica de datos practicada, se realizaron *dashboards* interactivos con la herramienta Power BI.

Para comenzar con el análisis, se realizó el clásico grafico circular, con el cual observar la relevancia de cada categoría de facturación (en dólares) en el conjunto de los datos.

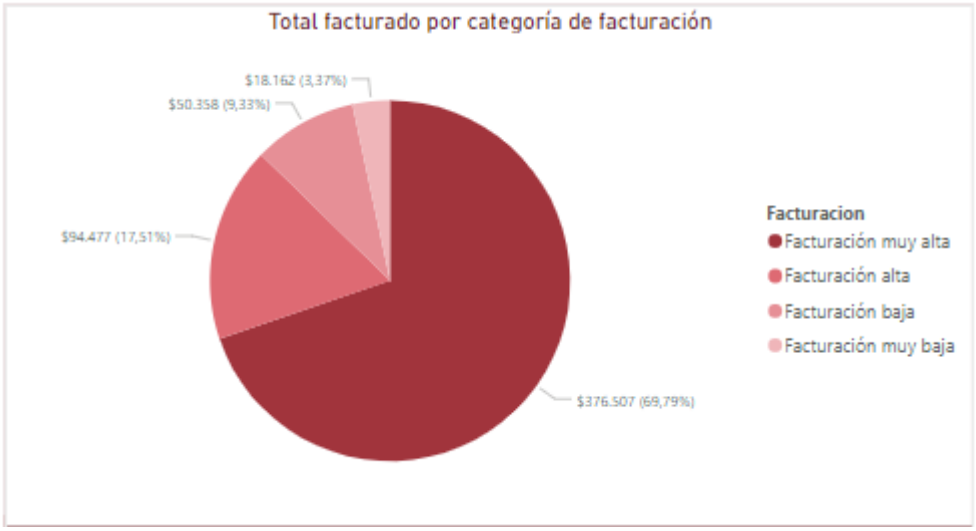


Ilustración 53 – Gráfico Total facturado por categoría de facturación

Seguidamente, para los resultados de la predicción de la variable objetivo (*Churn*), se realizaron gráficos de Matriz, similares a una tabla dinámica, pero que además de agrupar filas, también permiten agrupar columnas.

Esto se utilizó para observar cuanto facturaron los clientes que se **predijeron que se iban** a ir, cuanto facturaron los que **efectivamente se fueron** y finalmente cuanto facturaron los que se **predijeron que se iban**, y **realmente se fueron.** Para el ultimo cuadro se incorporó la funcionalidad de un segmentador, que permite filtrar los datos de las demás visuales.

Facturación abierta por predicción de Churn

Facturacion	0	1	Total
Facturación muy alta	25.188.852	151.228	25.340.080
Facturación alta	1.124.235	81.608	1.205.843
Facturación baja	218.766	63.756	282.522
Facturación muy baja	17.755	23.154	40.909
Total	**26.549.608**	**319.746**	**26.869.354**

Tabla 6 - Facturación por predicción de Churn

En esta matriz se observa el total de facturación mensual por categoría de facturación. El total facturado se situó en los UDS 26.869.354, representado por clientes que son categorizados como facturación muy alta, para los cuales en su mayoría se predijo que iban a mantener la relación comercial con la empresa, en tanto los que facturan menos, son los que tienen una mayor tendencia a dejar de utilizar el servicio. En este caso se predijo que en estos clientes hay un poco más del 50% de fuga.

Facturación abierta por Churn

Facturacion	0	1	Total
Facturación muy alta	24.963.573	376.507	25.340.080
Facturación alta	1.111.366	94.477	1.205.843
Facturación baja	232.164	50.358	282.522
Facturación muy baja	22.747	18.162	40.909
Total	**26.329.850**	**539.504**	**26.869.354**

Tabla 7 - Facturación por Churn

Esta matriz es similar a la anterior, diferenciándose que, en lugar de exponer la predicción, se expone la realidad de los clientes, en cuanto al Churn, con la cual se llega a las mismas conclusiones generales, de que los clientes cuya facturación es alta y representan la mayor proporción del *dataset*, un gran porcentaje continuó con la relación comercial, en tanto para los casos en donde la facturación es muy baja, se observó una relación de 44% aproximadamente, que finalizan la relación comercial.

Churn

0	1

Predicción de la Facturación ante clientes que se fugaron

Facturacion	0	1	Total
Facturación muy alta	226.105	150.402	376.507
Facturación alta	31.976	62.501	94.477
Facturación baja	6.117	44.241	50.358
Facturación muy baja	722	17.440	18.162
Total	**264.920**	**274.584**	539.504

Tabla 8 - Predicción facturación de clientes que se fugaron

En cuanto al efecto en la facturación de los clientes que se predijeron que se fueron y efectivamente lo hicieron, se observa que se lograron predecir UDS 274.584, de un total de UDS 539.504 mensuales, como se aprecia en la tabla anterior.

En lo que respecta a este resultado, se realiza un análisis más adelante en: “Análisis de resultados obtenidos del Churn”.

Para finalizar con la exploración de la estructura general, se calcula y expone en gráficos de tarjetas, la tasa de Churn actual de la empresa, tanto en termino de cantidad de fuga de clientes como en términos monetarios, a través de la facturación.

Ilustración 54 - Tasa de fuga de clientes

Del cuadro anterior, se puede interpretar que si bien la empresa tiene una tasa relativamente alta de fuga de clientes medida en unidades (19%), estos clientes no son de los que más peso tienen en los ingresos de la empresa, ya que, si se observa en términos monetarios, la tasa se sitúa en el 2% del total de facturación.

Por otro lado, luego de un análisis integral de lo ya expuesto, se puede concluir que si bien la empresa no ha aplicado técnicas de *Machine Learning* para el monitoreo y predicción de comportamiento de sus clientes, se intuye que cuenta con medidas para aquellos clientes categorizados como “Facturación muy alta”, dado que la tasa de Churn en esa categoría es menor y son los que representan la mayor parte de la facturación.

En segunda instancia para la extracción de conclusiones, se analiza en particular la estructura por Churn y *cluster* al que pertenecen los clientes. Este análisis se realiza tanto en términos monetarios como de cantidad y con gráficos circulares y de barras apiladas.

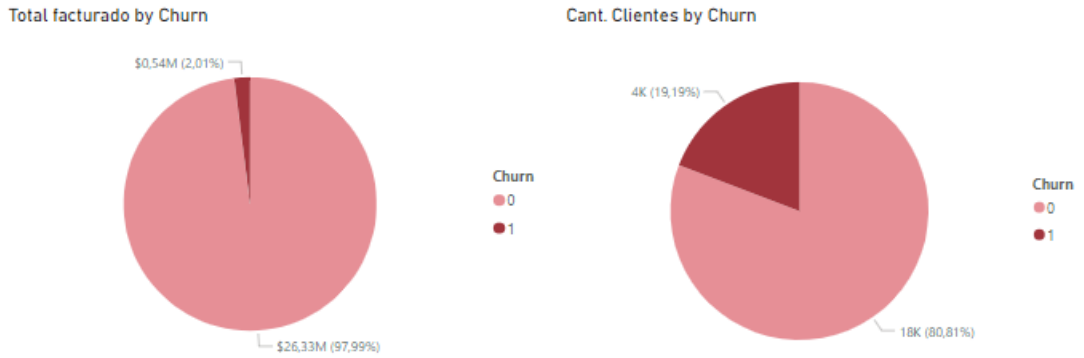


Ilustración 55 - Estructura por Churn por facturación y cliente

Si se observan los gráficos en conjunto, en el de la derecha se ve que el peso en cantidad es mayor que en el de la izquierda que es expresado en valores monetarios. Lo que permite interpretar, que puede haber muchos clientes que se van, pero son de los que representan un menor ingreso para la empresa. Es posible ver que los de menor peso económico son los que más se van, es decir, aquellos que tienen menos subscripciones o por menor valor.

Quizás lo interesante de esto es que por parte de la empresa se pueda analizar si estos clientes son necesarios mantener ya que el costo de mantenerlos puede llegar a ser mayor que el precio que ellos pagan por el servicio.

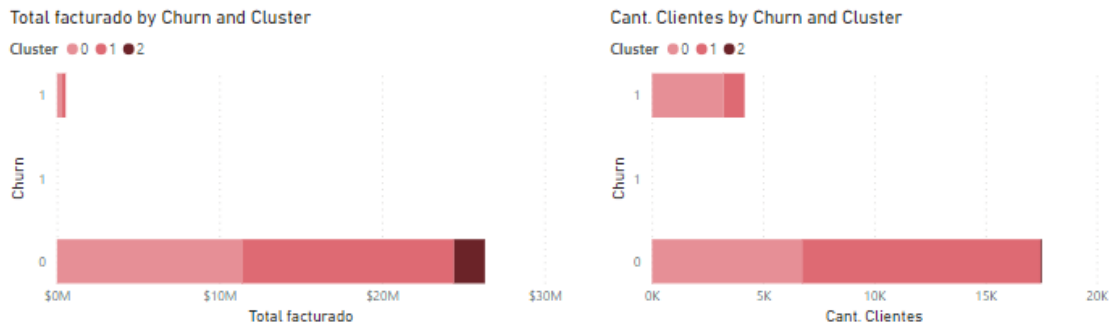


Ilustración 56 - Churn por *cluster*, facturación y clientes

En estos gráficos se puede observar que en cantidad de clientes (gráfico de la derecha), el *cluster* 2 cuenta con muy pocas empresas, pero si lo comparamos con el gráfico de la

izquierda (términos monetarios), este comienza a ganar relevancia. Lo que permite concluir que, dentro de este *cluster*, se encuentren clientes cuya facturación sea alta. Por lo que, si bien en la actualidad no se han ido clientes de este *cluster*, la empresa debe monitorearlos y cuidar de que no se vayan.

Si solo se tuviera en cuenta el análisis realizado en el Churn, los clientes que más se van son los que les facturan menos dinero. Pero siempre es bueno tener monitoreado a los clientes importantes, para que tengan un servicio adecuado y que se encuentren conformes con el mismo.

A continuación, se expone el grafico de cantidad de clientes por Churn y *cluster*, se evidencia la cantidad de clientes dentro del c*luster* 2.

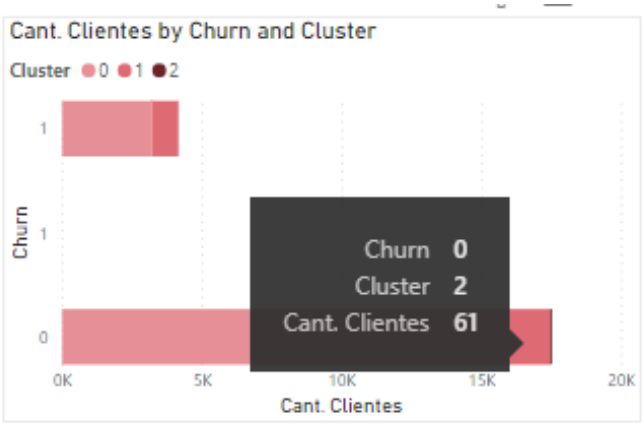


Ilustración 57 - Cantidad clientes del *cluster* 2

Como parte del análisis, se realizaron diferentes tipos de *dashboards* para observar con otra óptica algunas de las variables del *dataset*, que se consideran relevantes como lo son: países y tipo de empresas (columna “Vertical”) por *cluster* en los clientes que realmente se fueron tanto en términos de cantidades y monetarios.

A continuación, se exponen las imágenes para el análisis mencionado, en términos monetarios

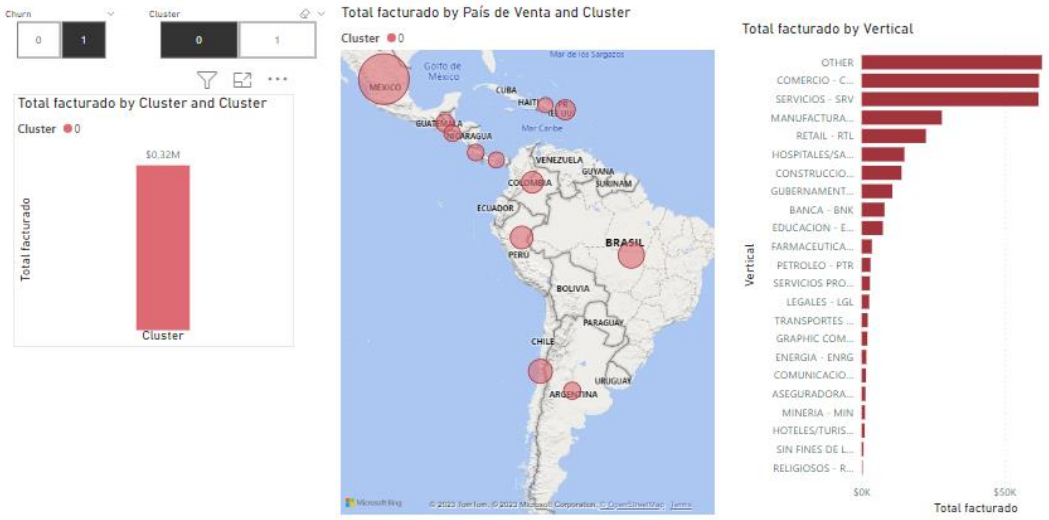


Ilustración 58 - Churn por c*luster*, tipo de empresa y país de venta por facturación

Con estas representaciones, se puede observar que dentro del *cluster* 0, se perdieron clientes por 0,32 millones aproximadamente, de los cuales en su mayoría pertenece al país de México (0,15 millones aproximadamente), seguido de Chile, Brasil y Colombia. En los rubros de empresas en que más se perdió facturación, fueron de *other*, comercio, servicios, manufactura y *retail*, tal como se puede observar en el último grafico de la imagen.

Se realizó nuevamente el análisis anterior, pero esta vez en términos de cantidad de clientes, las conclusiones a las que se llega son:

Ilustración 59 - Churn por *cluster*, tipo de empresa y país de venta por clientes

En términos de cantidades de clientes se puede observar que dentro del c*luster* 0 de los clientes que se fueron, la mayoría pertenecen a Chile, Guatemala, Costa Rica y Puerto Rico. En los rubros en que más se perdieron clientes fueron de *other*, comercio, servicios y manufactura.

Finalmente, se realizó un análisis de las variables de equipos arrendados, cantidad de reclamos e incidentes, de los clientes perdidos, por país. Para ilustrar estos análisis, se expone el total del resultado y luego la visualización para dos países de los observados en los análisis, los que más fugas tienen.

Se expone a continuación el resultado de la situación general de los clientes que se han fugado.

Churn
- ○ 0
- ◉ 1

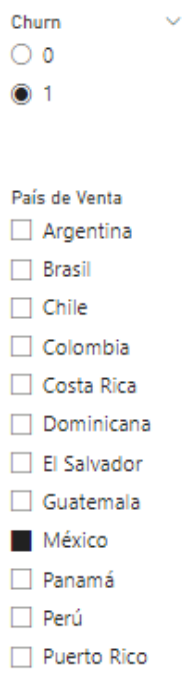
País de Venta
- ☐ Argentina
- ☐ Brasil
- ☐ Chile
- ☐ Colombia
- ☐ Costa Rica
- ☐ Dominicana
- ☐ El Salvador
- ☐ Guatemala
- ☐ México
- ☐ Panamá
- ☐ Perú
- ☐ Puerto Rico

Vertical	Cant de equipos arrendados	Cantidad de Reclamos	Catidad de Incidentes
OTHER	1.025	664	1.072
COMUNICACIONES - CMT	490	50	11
COMERCIO - CMR	281	689	507
MANUFACTURA - MNF	228	191	202
SERVICIOS - SRV	198	684	692
GUBERNAMENTAL - GSV	138	494	143
ASEGURADORAS - INS	103	8	7
EDUCACION - EDU	91	126	57
PETROLEO - PTR	90	9	12
ENERGIA - ENRG	74	10	15
BANCA - BNK	51	34	24
HOSPITALES/SALUD - HSP	35	101	160
TRANSPORTES - TRN/LOGISTICAS	28	55	80
BIENES RAICES-REALESTATE	25	13	18
GRAPHIC COMMUNICATIONS-GC	25	75	136
MINERIA - MIN	19	43	17
HOTELES/TURISMO - HTL	16	14	42
RETAIL - RTL	12	16	85
LEGALES - LGL	11	47	29
AGENCIAS DE PUBLICIDAD - PUB	8	5	8
FARMACEUTICAS - PHA	8	51	66
SERVICIOS PROFESIONALES	8	20	22
CONSTRUCCION - CNSTR	5	16	5
RELIGIOSOS - REL	3	6	23
SIN FINES DE LUCRO - NPS	2	12	9
AGRICULTURA-AGRO	1	3	11
COOPERATIVAS-COOP	0	1	7
INTERCOMPANIES - INT	0	0	2
OFICINAS PROFESIONALES - OFPRO	0	0	0
Total	**2.975**	**3.437**	**3.462**

Tabla 9 - Situación general de clientes fugados

Aquí se observa que para los clientes que se perdieron, el rubro *other* es el que cuenta con la mayor cantidad de equipos arrendados, a su vez es el que tiene una mayor cantidad de incidentes. Por otro lado, los rubros de empresa que más reclamos tienen son: comercio (689) y servicios (684).

Se expone este mismo análisis para el caso de México:



Vertical	Cant de equipos arrendados	Cantidad de Reclamos	Catidad de Incidentes
MANUFACTURA - MNF	162	0	0
SERVICIOS - SRV	108	0	0
ASEGURADORAS - INS	101	0	0
COMERCIO - CMR	92	0	0
ENERGIA - ENRG	47	0	0
PETROLEO - PTR	47	0	0
BANCA - BNK	26	0	0
MINERIA - MIN	13	0	0
OTHER	9	0	0
HOTELES/TURISMO - HTL	8	0	0
TRANSPORTES - TRN/LOGISTICAS	3	0	0
EDUCACION - EDU	2	0	0
COMUNICACIONES - CMT	1	0	0
FARMACEUTICAS - PHA	1	0	0
HOSPITALES/SALUD - HSP	1	0	0
LEGALES - LGL	1	0	0
CONSTRUCCION - CNSTR	0	0	0
RETAIL - RTL	0	0	0
SERVICIOS PROFESIONALES	0	0	0
SIN FINES DE LUCRO - NPS	0	0	0
Total	**622**	**0**	**0**

Tabla 10 - Situación en México de clientes fugados

En este caso se observa que no se han registrado incidentes ni reclamos, y que los rubros con mayor cantidad de equipos arrendados son: manufactura, servicios y aseguradoras.

Esto da a pensar que la situación de fuga de clientes en este país no viene asociada a reclamos e incidentes no atendidos debidamente, fallas de los equipos o insatisfacciones de algún tipo, sino que otras cuestiones son las que están por detrás, cuestiones que sería adecuado investigar para analizar y atacar eficientemente.

En la siguiente tabla se expone la Situación de Chile:

Churn
- ○ 0
- ◉ 1

País de Venta
- ☐ Argentina
- ☐ Brasil
- ■ Chile
- ☐ Colombia
- ☐ Costa Rica
- ☐ Dominicana
- ☐ El Salvador
- ☐ Guatemala
- ☐ México
- ☐ Panamá
- ☐ Perú
- ☐ Puerto Rico

Vertical	Cant de equipos arrendados	Cantidad de Reclamos	Catidad de Incidentes
COMUNICACIONES - CMT	426	36	0
SERVICIOS - SRV	14	194	104
COMERCIO - CMR	7	58	37
MANUFACTURA - MNF	6	8	21
AGENCIAS DE PUBLICIDAD - PUB	2	3	1
HOSPITALES/SALUD - HSP	2	9	42
BANCA - BNK	1	3	0
EDUCACION - EDU	1	13	10
ENERGIA - ENRG	1	0	0
GUBERNAMENTAL - GSV	1	18	1
MINERIA - MIN	1	26	12
AGRICULTURA-AGRO	0	0	0
ASEGURADORAS - INS	0	2	0
CONSTRUCCION - CNSTR	0	6	1
FARMACEUTICAS - PHA	0	2	15
GRAPHIC COMMUNICATIONS-GC	0	1	0
HOTELES/TURISMO - HTL	0	0	0
LEGALES - LGL	0	0	4
OTHER	0	51	7
RETAIL - RTL	0	0	0
SERVICIOS PROFESIONALES	0	1	0
SIN FINES DE LUCRO - NPS	0	0	0
TRANSPORTES - TRN/LOGISTICAS	0	1	1
Total	**462**	**432**	**256**

Tabla 11 - Situación en Chile de clientes fugados

El rubro con mayor cantidad de equipos arrendados se corresponde a Comunicaciones, no obstante, no es el rubro con mayor cantidad de reclamos, así como también se observa que no se han registrado incidentes en esta vertical.

En cambio, la vertical de negocio con mayor cantidad de reclamos e incidentes es servicios.

5. Análisis del resultado por predicción de Churn

En la realidad de las empresas, aplicar un modelo de predicción de Churn, lleva antes una adecuada consideración de factores claves y cálculos para determinar su aplicación o no.

Como parte del proceso, se considera adecuado realizar una serie de pasos que no necesariamente siguen un flujo secuencial de aplicación.

Ilustración 60 - Pasos para la aplicación de un modelo de predicción de Churn

A continuación, describimos las fases del proceso:

1) Identificar los costos de Churn: es decir, determinar cuánto cuesta perder a un cliente debido a Churn. Esto puede incluir la pérdida de ingresos, el costo de adquirir un nuevo cliente, el costo de los descuentos y promociones para retener a los clientes, entre otros costos indirectos.
2) Estimar la tasa de Churn actual: Esta tasa se calcula al dividir el número de clientes que se pierden en un período determinado por el número total de clientes al comienzo del mismo período.
3) Calcular el impacto de reducir la tasa de Churn: si se implementa un modelo de predicción de Churn, se puede esperar una reducción de su tasa, dado que se detectarían clientes potenciales de irse y en consecuencia poder aplicar acciones de retención. Aquí se debería lograr estimar cuánto se puede reducir la tasa de Churn e inferir el impacto en los costos identificados en el primer punto.
4) Calcular los costos asociados del modelo de predicción de Churn, costos de implementación (*software, hardware*), contratación de personal adicional para su aplicación, mejora y análisis de resultado, costos de personal para definir y ajustar las estrategias de *marketing* y operaciones que se adicionen por esta implementación, costos propios de marketing y operaciones, entre otros. El autor Thomas H. Davenport dice que, por cada dólar invertido en el algoritmo, se debe estimar 100 dólares para su implementación y posterior soporte [9].

5) Finalmente compara los beneficios y costos, los beneficios de reducir la tasa de Churn con el costo de implementar el modelo. Si los beneficios superan los costos, entonces la empresa tiene la justificación financiera adecuada para presentar ante el directorio e implementar el modelo.

A pesar del resultado de los cinco (5) pasos anteriores para evaluar la aplicabilidad del proyecto, es importante tener en cuenta que la implementación de un modelo de predicción de Churn no es una garantía de éxito y que existen muchas variables (internas y externas) que pueden afectar los resultados. Lo cual se debe dejar en claro a los directores, para que estos puedan tomar objetivamente la decisión.

Desde el alcance de este proyecto, para reflejar el beneficio de este en la empresa al implementar el algoritmo de predicción de forma preventiva, es que se detalla a continuación un resumen en donde el algoritmo predijo que el cliente iba a dejar de utilizar el servicio y efectivamente lo hizo.

Tal como se ha mencionado anteriormente, es esperable que la aplicación del algoritmo y las de medidas de acción para retener a clientes implica un costo para la empresa, este sería menor que el de afrontar la pérdida total de un cliente y a la vez, la captación de un nuevo cliente.

Si bien no se indica la "ganancia" real neta para la empresa, se expone el rango promedio de facturación de los clientes que se predijeron que se iban y efectivamente se fueron.

Del conjunto de datos, efectivamente se fueron clientes por una facturación promedio mensual de U$S 539.500 aprox. y el algoritmo predijo correctamente clientes por un importe mensual de U$S 274.600 aprox. (51% de los que realmente se fueron).

Total Facturación Churn	Total Facturación Predicción Churn
USD 539.504	USD 274.584

Tabla 12 - Resumen facturación por Churn y Predicción de Churn

Para ilustrar lo anteriormente comentado, se expone una muestra de 10 clientes, los cuales son de la actividad Servicios, Comercio, Manufactura, Hospitales, Industria Gráfica y 2 de ellos están catalogados como otras verticales de actividad, ahora bien, se puede observar también que estos clientes pertenecen 4 a Colombia, 3 son de Brasil, y los restantes 3 corresponden a Chile, Argentina y Guatemala.

A continuación, se expone la imagen correspondiente a estos clientes:

CustomerId	Facturacón promedio mensual	País de Venta	Vertical	Meses desde última Factura	Antiguedad	Cant com en ult 24 meses	Encuestas	Catidad de Incidentes	Cantidad de Reclamos	Tiempo de respuesta promedio	Fracción reclamos a tiempo	Churn	Predicción de Churn
119b875c6b13a38b1cdc09920083233b	8.826.00	Colombia	HOSPITALES/SALUD - HSP	16	168	73	Indiferente	0	0	0	0	1	1
35dc9ce9318d6b6a83bbe1df80cacdb3	8.787.00	Brasil	OTHER	19	154	10	Indiferente	2	0	75	5	1	1
490f7cad9bff9073abbe0e79a06283c7	8.575.00	Colombia	SERVICIOS - SRV	20	168	59	Indiferente	168	132	5384	49	1	1
d1e835e5b89270a83054ecac391c6c44	7.610.00	Colombia	SERVICIOS - SRV	19	169	0	Indiferente	0	2	375	5	1	1
c1398d49dbbfc5aea94c924f46408b51	5.713.00	Brasil	COMERCIO - CMR	18	152	14	Indiferente	3	0	5867	67	1	1
d6cd205a822f25f3170c53040e4077b7	5.207.00	Guatemala	GRAPHIC COMMUNICATIONS-GC	15	154	6	Indiferente	13	10	5483	61	1	1
d426eada0c6ba22af1b6d32263dab4f0	4.982.00	Colombia	SERVICIOS - SRV	16	168	2	Indiferente	10	10	4745	5	1	1
35edfd30559b57ea58d362317c9303ec	4.514.00	Chile	SERVICIOS - SRV	14	169	0	Indiferente	1	18	4974	58	1	1
eb4e276924c3ef4b03f367e743064ac1	4.108.00	Brasil	OTHER	13	153	13	Negativa	16	0	5269	62	1	1
b13c610874c94108345a64c7b38a7b14	4.005.00	Argentina	MANUFACTURA - MNF	17	159	2	Indiferente	6	22	4736	36	1	1

Tabla 13 - Top 10 clientes con mayor facturación mensual con predicción del Churn

6. Recomendaciones al cliente

- Es importante definir claramente los objetivos y las metas que se quieren alcanzar con el análisis de Churn Rate y Clustering, es decir si lo que se busca es bajar la tasa de abandono de clientes, o identificar patrones de comportamiento de los clientes para ofrecer productos y servicios más relevantes o alguno otro objetivo.

- La gobernanza de datos es un tema importante dentro de las empresas. Es esencial para garantizar la calidad, integridad y la seguridad de los datos de la empresa. Como medidas a recomendar, se destacan: establecer políticas y estándares de datos claros, contratar a alguien que lidere el proyecto y se encargue de la gobernanza, desarrollar un catálogo de datos y asignar responsables de los mismos en cada región y sucursal, para así lograr y mantener una homogeneidad, implementar medidas de seguridad de los datos (encriptación, autenticación y autorización) para lograr protegerlos, capacitar al personal de la empresa sobre políticas y estándares de datos así como de la gestión de los datos.

- Poder identificar otras variables dentro de la información que cuenta la empresa que puedan ayudar a mejorar el análisis de Churn Rate. Que las mismas puedan identificar factores que influyen en la cancelación del servicio, como por ejemplo comportamiento del cliente, la satisfacción del cliente.

- Se recomienda aplicar estos modelos, tanto de Churn Rate como de Clustering, de forma anticipada, ya que estos le permitirán realizar mejores gestiones sobre los datos de los clientes con los que cuentan y tomar acciones a tiempo.

  Es importante hacer un seguimiento del Churn Rate mes a mes para monitorear y conocer el comportamiento de los clientes. Con el Clustering se puede agrupar clientes con características similares y analizar su comportamiento. Quizás es importante agregar algunas técnicas como el procesamiento de lenguaje natural, para poder tomar decisiones de manera anticipada y lograr mantener clientes que son importantes para la empresa.

- Es relevante cuando se aplican estas técnicas realizar mejoras continuas, con revisiones regulares de los resultados y ajustes de las estrategias y técnicas utilizadas para optimizar la retención del cliente y el rendimiento empresarial.

- Otra recomendación importante es la interpretación de los resultados. Los clientes en los que se prediga que se van a ir de la empresa se le puede realizar encuestas o entrevistas para recopilar información sobre la experiencia del cliente y los motivos que lo llevaron a cancelar los servicios.

- Al analizar la variable vertical (rubro en el cual opera el cliente) hay una categoría residual que se llama "*Other*" en la cual se ve que pertenecen muchos de los clientes que se fueron de la empresa. Por lo tanto, se recomienda dejar reflejado el verdadero rubro de la empresa para poder tomar decisiones más ajustadas con los datos obtenidos.

- Como los que facturan menos son los que tienen una mayor tendencia a dejar de utilizar el servicio. Se recomienda a la empresa analizar los casos en que los ingresos que generan estos clientes son menores que los costos que lleva mantenerlos dentro de la empresa. Sin perder de vista las características cualitativas de los clientes como por ejemplo en ciertas ocasiones sirve tener clientes con los que la empresa pierde porque atraen otros tipos de clientes en los cuales la empresa gana dinero (clientes estratégicos).

7. Conclusiones

7.1. Lecciones aprendidas

En base al análisis expuesto en esta tesis, se concluye que se ha realizado un adecuado proceso de investigación y aplicación de los datos, así como de diferentes modelos que permiten determinar una adecuada hipótesis ante un problema de predicción bajo el aprendizaje supervisado, así como también la categorización y agrupación de los datos bajo el aprendizaje no supervisado.

En base lo aplicado, se destaca como lecciones aprendidas, la importancia de contar con una adecuada gobernanza de los datos, ya que si los datos no guardan cierto criterio y uniformidad, la aplicación de los algoritmos requiere la realización de ciertos supuestos, una mayor limpieza y preparación de los datos, que puede resultar en la necesidad de omitir gran parte de estos, porque su falta de coherencia genera que las diferentes variables no aporten valor en los algoritmos y los resultados que se obtienen no se condicen posteriormente con la realidad, llevando esto a la toma de decisiones erróneas.

Al considerar el ámbito de aplicación de un trabajo de este estilo, es de suma importancia contar con el involucramiento temprano y constante de la empresa para llegar a conseguir el éxito del proyecto.

En esta tesis, tras las primeras reuniones con la empresa, se tuvo una demora en la entrega de los datos, situación que se presentó como una debilidad, pero que se logró transformar en fortaleza, dado que, se optó por investigar sobre problemas similares, con lo cual se generó un trabajo de pre-proyecto como base para luego aplicarlo en los datos proporcionados. Gracias a esto, al correr el código con los datos de Ricoh, no se encontraron dificultades significativas, tanto en la limpieza del *dataset* como en la aplicación de los distintos modelos.

Finalmente, como subproducto de los algoritmos aplicados, es posible extraer información para análisis de mejoras en diferentes áreas y procesos de negocio, como puede ser el caso de detectar ciertas características susceptibles de mejoras en lo que respecta a la atención de reclamos.

7.2. Trabajo futuro

En esta sección se presentan las posibles líneas de investigación a seguir, tomando como base los aportes realizados en esta tesis.

Se entiende que se puede seguir profundizando en las técnicas de este tipo y en diferentes algoritmos y librerías para lograr mejores resultados, así como también llegar a la generación de un modelo prescriptivo con el cual además de predecir la fuga de clientes, arroje las acciones a realizar por parte de la empresa para lograr retener al cliente, en función de su tipo y características.

Asimismo, se puede profundizar en el método redes neuronales convolucionales para la extracción de características de los clientes.

En lo que respecta a los datos en sí, se considera adecuado trabajar en la implementación de una gobernanza de datos, que permita la obtención y mantenimiento de una base consistente, uniforme y rica para la extracción de conclusiones para la toma de decisiones gerenciales, tal como se indica en las recomendaciones al cliente.

Para la toma de decisiones, se considera adecuado que la empresa realiza un estudio y seguimiento de la tasa de Churn, a efectos de poder disminuir la tasa actual, y detectar en qué momento y/o en que regiones es que se dispara más. Así como también, realizar un análisis detallado de costo-beneficio de la aplicación del código, con aplicación no nos referimos únicamente a la corrida del código, sino a interpretar sus resultados, definir los planes de acción para evitar la fuga de clientes y aplicarlos.

Finalmente, a futuro Ricoh debería evaluar la posibilidad de aprovechar otras fuentes de datos para la toma de decisiones. Una opción podría ser aprovechar los correos recibidos por los clientes, comentarios en las redes sociales y comentarios en la página web, para realizar un análisis de sentimiento. Esto puede ayudar a identificar problemas de forma temprana e identificar a los clientes insatisfechos para tomar acciones. Otra opción para tener en cuenta es utilizar datos de uso de las impresoras, ya sea para realizar mantenimiento predictivo o identificar fallas de forma temprana, reduciendo reclamos e incidentes.

8. Glosario

Accuracy: exactitud. Es la fracción de predicciones que el modelo realizó correctamente [10].

Algoritmo: Conjunto de instrucciones a seguir para resolver un problema concreto. En Inteligencia Artificial hace referencia a los procesos de entrenamiento de modelos de Aprendizaje automático [11].

Análisis de sentimiento: En procesamiento de lenguaje natural, se refiere a las técnicas que permiten deducir, a partir de un texto libre, el sentimiento que el autor tenía al escribirlo [11].

Árboles de decisión: Algoritmo de aprendizaje supervisado no paramétrico, que se utiliza tanto para tareas de clasificación como de regresión. Tiene una estructura de árbol jerárquica, que consta de un nodo raíz, ramas, nodos internos y nodos hoja. [12]

Azure: La plataforma compuesta por más de 200 productos y servicios en la nube diseñados para ayudarle a dar vida a nuevas soluciones que permitan resolver las dificultades actuales y crear el futuro [13] .

B2B: *Business-to-Business* es un modelo de negocio que consiste en los servicios que una compañía entrega a otra con el objetivo de mejorar las ventas de los productos y bienes que ofrece. Es decir, una transacción comercial entre empresas [14].

Bagging: también conocido como agregación bootstrap, es el método de aprendizaje por conjuntos que se usa comúnmente para reducir la varianza dentro de un conjunto de datos ruidoso. En el empaquetado, se selecciona una muestra aleatoria de datos en un conjunto de entrenamiento con reemplazo, lo que significa que los puntos de datos individuales se pueden elegir más de una vez [15].

Big Data: almacenamiento, procesamiento y gestión de un gran conjunto de datos o combinaciones de conjuntos de datos, que pueden ser tanto estructurados, como no estructurados [11].

Boosting: método de aprendizaje por conjuntos que combina una serie de aprendices débiles en un aprendiz fuerte para minimizar los errores de entrenamiento. En el boosting, se selecciona una muestra aleatoria de datos, se ajusta a un modelo y luego se entrena secuencialmente [16].

Churn Rate: o Tasa de abandono de clientes es una métrica que mide el número de clientes y suscriptores que han dejado de seguir a una compañía en un período de tiempo [17].

Cluster: grupo de objetos similares entre sí, pero distintos con respecto a los objetos de otros grupos [11].

Clustering: técnica de data mining o minería de datos [11].

CRM: *Customer Relationship Management*, se refiere al conjunto de prácticas, estrategias comerciales y tecnologías enfocadas en la relación con el cliente [18].

Dashboard: herramienta de gestión de la información que monitoriza, analiza y muestra de manera visual los indicadores de desempeño, métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico [19].

Dataframe: datos tabulares bidimensionales, potencialmente heterogéneos, con ejes etiquetados (filas y columnas) [20].

Data Lake: repositorio de almacenamiento que alberga el dato en estado crudo para ser consumido por la empresa en el momento que quiera. La información que se almacena en el Data Lake procede de diversas fuentes de datos, por lo que guarda datos de todo tipo, estructurados y no estructurados [11].

Dataset: o conjunto de datos hace relación a los contenidos de una única tabla de bases de datos donde cada columna representa una variable y cada fila representa a un valor determinado del conjunto de datos [11].

Dato: Unidad mínima de información con una semántica definida [11].

ERP: La planificación de recursos empresariales, también conocida como ERP, es un sistema que ayuda a automatizar y administrar los procesos empresariales de distintas áreas: finanzas, fabricación, venta al por menor, cadena de suministro, recursos humanos y operaciones. Los sistemas ERP desglosan los silos de datos e integran la información obtenida en los diversos departamentos, de esta forma, ayudan a los directivos a extraer conocimientos, optimizar operaciones y mejorar la toma de decisiones [21].

Formulas DAX: recopilación de funciones, operadores y constantes que se pueden usar para calcular y devolver uno o varios valores [22].

Gobernanza de datos: la gobernanza de datos empresariales abarca las políticas y procedimientos que se implementan para garantizar que los datos de una organización sean precisos y que se manejen correctamente cuando se ingresan, almacenan, manejan, acceden y eliminan [23].

Gradient Boosting Machine (GBM): técnica de aprendizaje automático utilizado para el análisis de la regresión y para problemas de clasificación estadística, el cual produce un modelo predictivo en forma de un conjunto de modelos de predicción débiles, típicamente árboles de decisión. Construye el modelo de forma escalonada y los generaliza permitiendo la optimización arbitraria de una función de perdida diferenciable [24].

Inteligencia Artificial: conjunto de sistemas o combinación de algoritmos, cuyo propósito es crear máquinas que imitan la inteligencia humana para realizar tareas y pueden mejorar conforme la información que recopilan [25].

Machine Learning: o aprendizaje automático, se engloba dentro de la rama de la inteligencia artificial y hace referencia al entrenamiento de las máquinas generalizando comportamientos que ofrecen los datos [11].

Outlier: valor atípico, se trata de una observación que es numéricamente distante del resto de los datos [26].

Portabilidad: Es la propiedad de un programa o una aplicación informática que le permite funcionar bajo diferentes sistemas. Cuando el programa informático es portable puede ser utilizados en diferentes tipos de equipos [27].

Python: lenguaje de programación interpretado y de propósito general, de código abierto [11].

Random Forest: método de aprendizaje automático que se basa en la combinación de varios árboles de decisión. Normalmente alcanzan niveles de precisión más altos que los árboles por separado. Pueden abordar tareas tanto de clasificación automática (elegir una categoría entre varias posibles), como de regresión (estimar un valor numérico) [11].

Sklearn: biblioteca que contiene una gran variedad de herramientas eficientes para el aprendizaje automático y modelado estadístico, incluyendo clasificación, regresión, agrupación, y reducción de dimensionalidad [28].

Stacking: apilamiento de modelos. Técnicas que permiten entrenar una segunda capa de modelos a partir de las predicciones de los modelos individuales generando, en general, mejores predicciones, más robustas y con menos sesgos [11].

Variable: La variable es una característica, cualidad o propiedad observada que puede adquirir diferentes valores y es susceptible de ser cuantificada o medida en una investigación [29].

Variables dummy: variables que representan un atributo con dos o más niveles o categorías diferentes [30].

9. Referencias bibliográficas

[1] S. Prakash, “AI 101: Understanding Customer Churn Management,” Junio 2018. [En línea]. Available: https://towardsdatascience.com/ai-101-understanding-customer-churn-management-514416c17643.

[2] Algotive, “Machine Learning: ¿Qué es el aprendizaje automático y cómo funciona?,” Marzo 2022. [En línea]. Available: https://www.algotive.ai/es-mx/blog/machine-learning-que-es-el-aprendizaje-autom%C3%A1tico-y-c%C3%B3mo-funciona.

[3] J. A. Sanchez, “¿Cómo aprenden las máquinas? Machine Learning y sus diferentes tipos,” Agosto 2020. [En línea]. Available: https://datos.gob.es/es/blog/como-aprenden-las-maquinas-machine-learning-y-sus-diferentes-tipos.

[4] P. R. d. l. Santos, “Tipos de aprendizaje en Machine Learning: supervisado y no supervisado,” Diciembre 2021. [En línea]. Available: https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/.

[5] javatpoint.com, “Random Forest Algorithm,” javatpoint, 2011-2021. [En línea]. Available: https://www.javatpoint.com/machine-learning-random-forest-algorithm.

[6] V. Morde, “XGBoost Algorithm: Long May She Reign!,” Abril 2019. [En línea]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

[7] stats.stackexchange.com, “Bagging, boosting and stacking in machine learning,” Octubre 2017. [En línea]. Available:

https://stats.stackexchange.com/questions/18891/bagging-boosting-and-stacking-in-machine-learning.

[8] ISO.org, “Organisation internationale de normalisation, “Systems and software engineering: architecture description.,” ISO/IEC/IEEE 42010,” Diciembre 2011. [En línea]. Available: https://www.iso.org/standard/50508.html.

[9] T. C. R. &. T. H. Davenport, “Getting Serious About Data and Data Science,” *MIT Sloan Management Review*, 2020.

[10] Google Developers, [En línea]. Available: https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=es-419. [Último acceso: Marzo 2023].

[11] Piperlab, [En línea]. Available: https://piperlab.es/glosario-de-big-data/. [Último acceso: Marzo 2023].

[12] IBM, “¿Qué es un árbol de desición?,” [En línea]. Available: https://www.ibm.com/es-es/topics/decision-trees. [Último acceso: Marzo 2023].

[13] Microsoft Azure , “Cloud Computing Dictionary,” [En línea]. Available: https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-azure/. [Último acceso: Marzo 2023].

[14] DocuSign, “B2B: Business-to-Business,” [En línea]. Available: https://www.docusign.mx/blog/b2b. [Último acceso: Marzo 2023].

[15] IBM, “¿Qué es Bagging?,” [En línea]. Available: https://www.ibm.com/es-es/topics/bagging?mhsrc=ibmsearch_a&mhq=Bagging. [Último acceso: Marzo 2023].

[16] IBM, “¿Qué es boosting?,” [En línea]. Available: https://www.ibm.com/es-es/topics/boosting?mhsrc=ibmsearch_a&mhq=Gradient%20Boosting%20Machine. [Último acceso: Marzo 2023].

[17] Paula Canal, “Churn Rate: Qué es y cómo se calcula,” [En línea]. Available: https://www.iebschool.com/blog/que-es-churn-rate-marketing-digital/#:~:text=%C2%BFQu%C3%A9%20es%20el%20Churn%20Rate,un%20largo%20per%C3%ADodo%20de%20tiempo. [Último acceso: Marzo 2023].

[18] Salesforce, “¿Qué es un CRM?,” [En línea]. Available: https://www.salesforce.com/mx/crm/#:~:text=CRM%20es%20la%20sigla%20utilizada,la%20relaci%C3%B3n%20con%20el%20cliente. [Último acceso: Marzo 2023].

[19] D. Ortiz, “¿Qué es un dashboard y para qué se usa?,” 2022. [En línea]. Available: https://www.cyberclick.es/numerical-blog/que-es-un-dashboard. [Último acceso: Marzo 2023].

[20] Data Science Team, “Pandas DataFrame,” [En línea]. Available: https://datascience.eu/es/programacion/python-pandas-dataframe/. [Último acceso: Marzo 2023].

[21] Microsoft Dynamics, “Definición de ERP,” [En línea]. Available: https://dynamics.microsoft.com/es-es/erp/define-erp/#:~:text=Definici%C3%B3n%20de%20ERP,suministro%2C%20recursos%20humanos%20y%20operaciones. [Último acceso: Marzo 2023].

[22] Microsoft Support, “Fundamentos de DAX,” [En línea]. Available: https://support.microsoft.com/es-es/office/tutorial-r%C3%A1pido-aprenda-los-fundamentos-de-dax-en-30-minutos-51744643-c2a5-436a-bdf6-c895762bec1a#:~:text=%C2%BFQu%C3%A9%20es%20DAX%3F,ya%20est%C3%A1%20en%20un%20modelo. [Último acceso: Marzo 2023].

[23] SAP Insights, “¿Qué es la gobernanza de datos?,” [En línea]. Available: https://www.sap.com/latinamerica/insights/what-is-data-governance.html#:~:text=La%20gobernanza%20de%20datos%20refiere,y%20seguridad%20de%20los%20datos%E2%80%93. [Último acceso: Marzo 2023].

[24] J. H. Freidman, “Greedy Function Approximation: A Gradient Boosting Machine,” *Annals of Statistics*, 2021.

[25] Wikipedia, “Inteligencia Artificial (IA),” [En línea]. Available: https://es.wikipedia.org/wiki/Inteligencia_artificial. [Último acceso: Marzo 2023].

[26] Wikipedia, “Valor atípico,” [En línea]. Available: https://es.wikipedia.org/wiki/Valor_at%C3%ADpico#:~:text=En%20estad%C3%ADstica%2C%20tales%20como%20muestras,valores%20at%C3%ADpicos%20ser%C3%A1n%20frecuentemente%20enga%C3%B1osas. [Último acceso: 2023 Marzo].

[27] Cultural S.A., de *Diccionario de Informática*, Madrid, Editorial Cultural, 1999, p. pp. 254.

[28] A. Torres, “Free Code Camp,” [En línea]. Available: https://www.freecodecamp.org/espanol/news/aprendizaje-automatico-en-python-las-principales-caracteristicas-nuevas-de-scikit-learn-que-debes-saber/#:~:text=Scikit%2Dlearn%20es%20uno%20de,agrupaci%C3%B3n%2C%20y%20reducci%C3%B3n%20de%20dimensionalidad. [Último acceso: Marzo 2023].

[29] A. E. Oyola-García, “Scielo,” [En línea]. Available: http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S2227-47312021000100016#:~:text=La%20variable%20es%20una%20caracter%C3%ADstica,entre%20dos%20valores%2C%20como%20m%C3%ADnimo. [Último acceso: Marzo 2023].

[30] J. Parra, “Introducción a las variables ficticias o variables dummy,” [En línea]. Available: https://www.javierparra.net/ecoknowmic/introduccion-a-las-variables-ficticias-o-variables-dummy/#:~:text=Las%20variables%20ficticias%2C%20variables%20dummy,c

omprender%20porqu%C3%A9%20utilizamos%20estas%20variables. [Último acceso: Marzo 2023].

[31] R. Yin, Case study research: design and methods, Beverly Hills, CA, EEUU, 2009.

[32] W3Schools, “Machine Learning,” 1999-2022. [En línea]. Available: https://www.w3schools.com/python/python_ml_getting_started.asp.

[33] J. A. Rodrigo, “Gradient Boosting con Python,” Octubre 2020. [En línea]. Available: https://www.cienciadedatos.net/documentos/py09_gradient_boosting_python.html.

[34] R. N. d. I. y. E. C. (RENIEC), “Arquitectura fisica y logica,” Setiembre 2015. [En línea]. Available: https://es.slideshare.net/ceasr/arquitectura-fisica-y-logica-53333784.

[35] G. Çavdar, “Average Customer Acquisition Cost (CAC) By Industry,” Febrero 2022. [En línea]. Available: https://hockeystack.com/blog/average-customer-acquisition-cost-by-industry/.

[36] L. Chen, “Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained,” Enero 2019. [En línea]. Available: https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725.

[37] T. Deshpande, “Telco Churn: EDA|CV Score (85%+)| F1 Score (80%+),” Setiembre 2022. [En línea]. Available: https://www.kaggle.com/code/tanmay111999/telco-churn-eda-cv-score-85-f1-score-80.

[38] M. D. P. Inilupu, “Data Science con R,” Junio 2020. [En línea]. Available: https://bookdown.org/dparedesi/data-science-con-r/aprendizaje-supervisado.html.

[39] Y. Kashnitsky, “Topic 1. Exploratory Data Analysis with Pandas,” Mayo 2021. [En línea]. Available: https://www.kaggle.com/code/kashnitsky/topic-1-exploratory-data-analysis-with-pandas/notebook.

[40] W. Koehrsen, “Random Forest Simple Explanation,” Diciembre 2017. [En línea]. Available: https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d.

[41] M. Kuhn, “Random Forest,” Marzo 2019. [En línea]. Available: https://topepo.github.io/caret/train-models-by-tag.html#Random_Forest.

[42] M. Mazzeschi, “https://pythonkai.org/2021/12/20/machine-learning-for-beginners-project-4-decision-tree-classifier/,” Diciembre 2021. [En línea]. Available: https://pythonkai.org/2021/12/20/machine-learning-for-beginners-project-4-decision-tree-classifier/.

[43] K. Moller, “How to calculate churn rate: Definition and formulas,” Abril 2020. [En línea]. Available: https://www.zendesk.com/blog/customer-churn-rate/.

[44] Aprender Machine Learning, “Random Forest, el poder del Ensamble,” Junio 2019. [En línea]. Available: https://www.aprendemachinelearning.com/random-forest-el-poder-del-ensamble/.

[45] D. Radečić, “Master Machine Learning: Decision Trees From Scratch With Python,” Abril 2021. [En línea]. Available: https://towardsdatascience.com/master-machine-learning-decision-trees-from-scratch-with-python-de75b0494bcd.

[46] S. Rao, “Decision Tree Classification: Explain It To Me Like I’m 10,” Marzo 2016. [En línea]. Available: https://pub.towardsai.net/decision-tree-classification-explain-it-to-me-like-im-10-59a53c0b338f.

[47] A. Ravanshad, “Gradient Boosting vs Random Forest,” Abril 2018. [En línea]. Available: https://medium.com/@aravanshad/gradient-boosting-versus-random-forest-cfa3fa8f0d80.

[48] F. Revert, “Fine-tuning XGBoost in Python like a boss,” Agosto 2018. [En línea]. Available: https://towardsdatascience.com/fine-tuning-xgboost-in-python-like-a-boss-b4543ed8b1e.

[49] G. Seif, “A Beginner’s guide to XGBoost,” Mayo 2019. [En línea]. Available: https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7.

[50] R. Shaw, “The 10 Best Machine Learning Algorithms for Data Science Beginners,” Junio 2019. [En línea]. Available: https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/#:~:text=The%20first%205%20algorithms%20that,are%20examples%20of%20supervised%20learning..

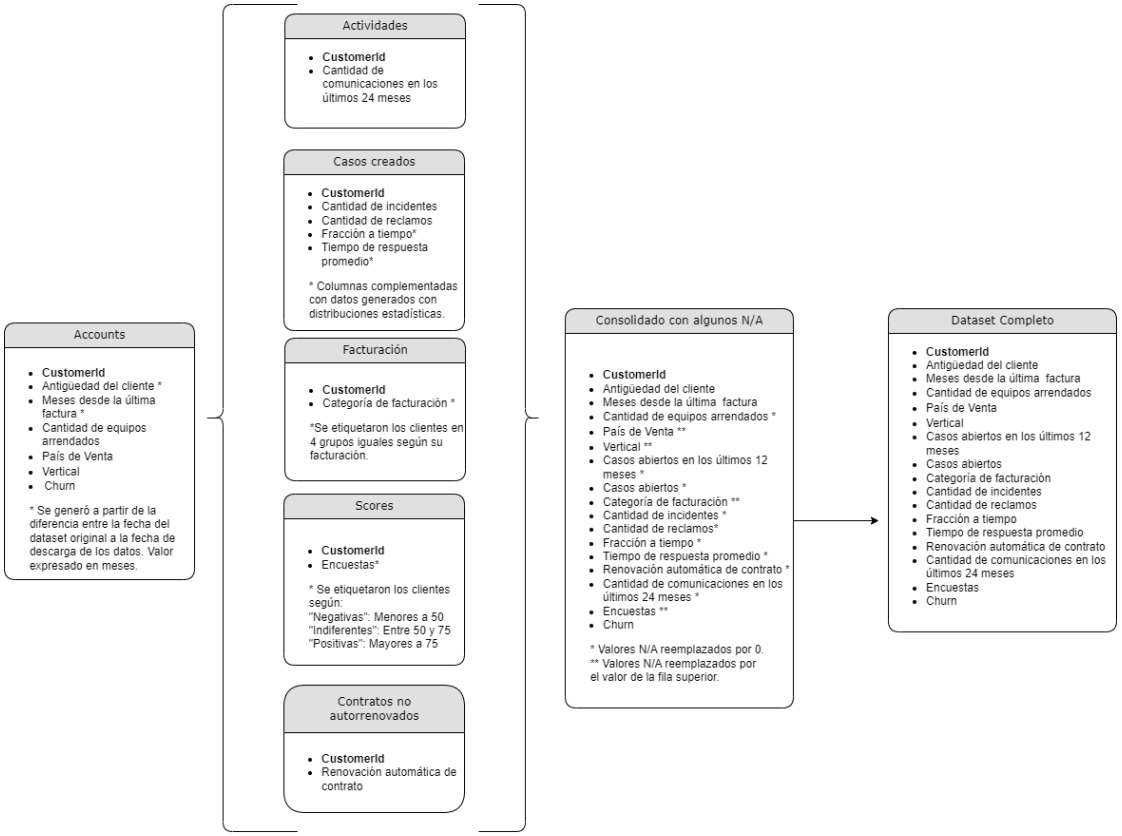
[51] T. Yiu, “Understanding Random Forest,” Junio 2019. [En línea]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[52] S. Brown, The Value Matrix Approach, Creating Wealth And Success By Reaching Your Personal And Business Goals, USA, 2004.

[53] Microsoft, “Optimize marketing with machine learning,” [En línea]. Available: https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/optimize-marketing-with-machine-learning. [Último acceso: Marzo 2023].

10. ANEXOS

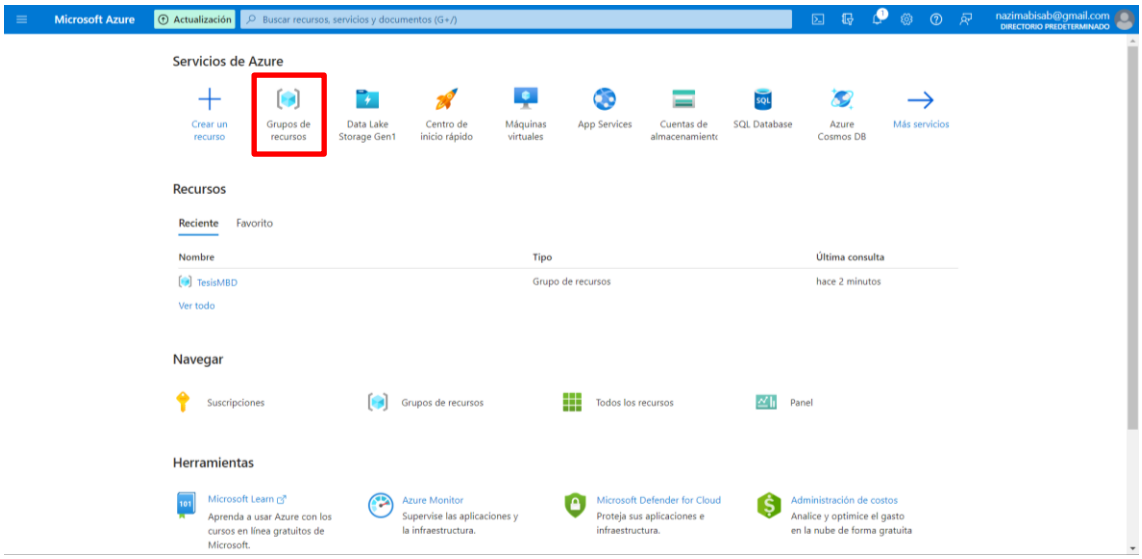
10.1. Diagrama de Dataflow

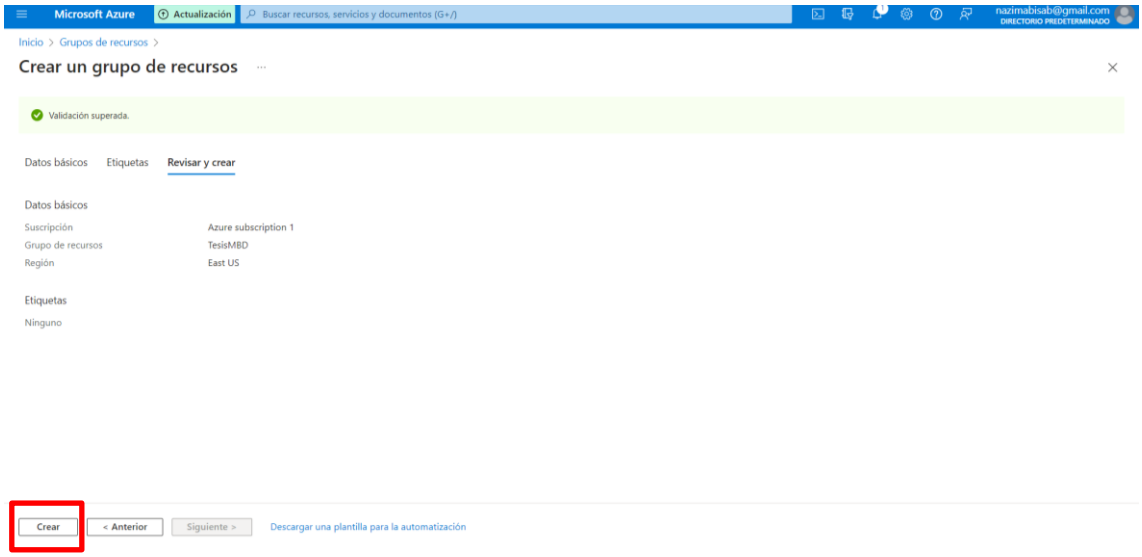


10.2. Implementación en Microsoft Azure

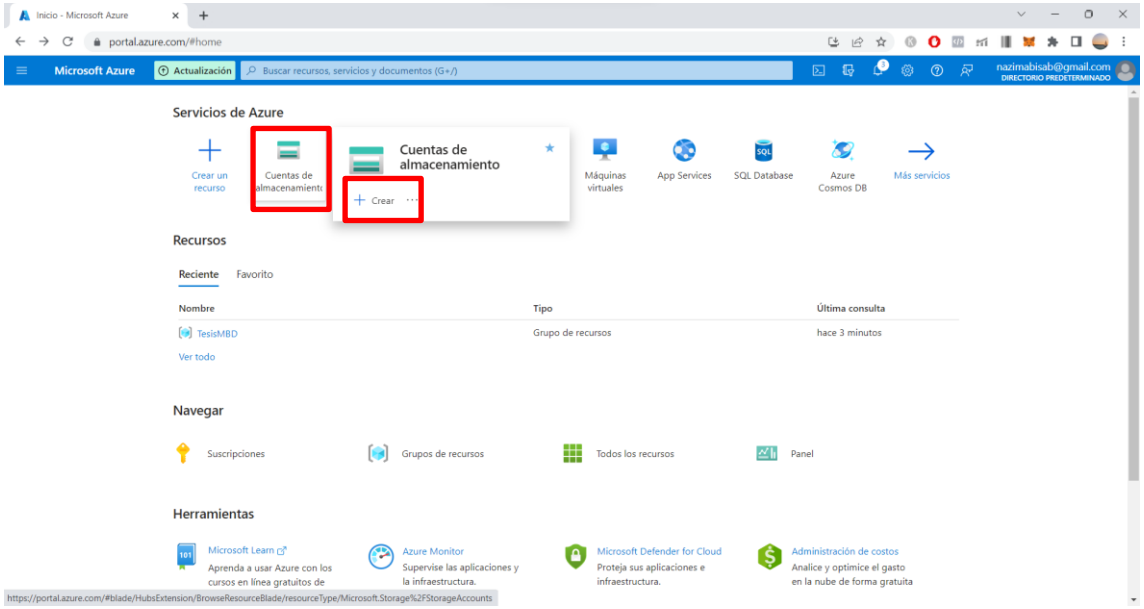
1- Acceder a Microsoft Azure

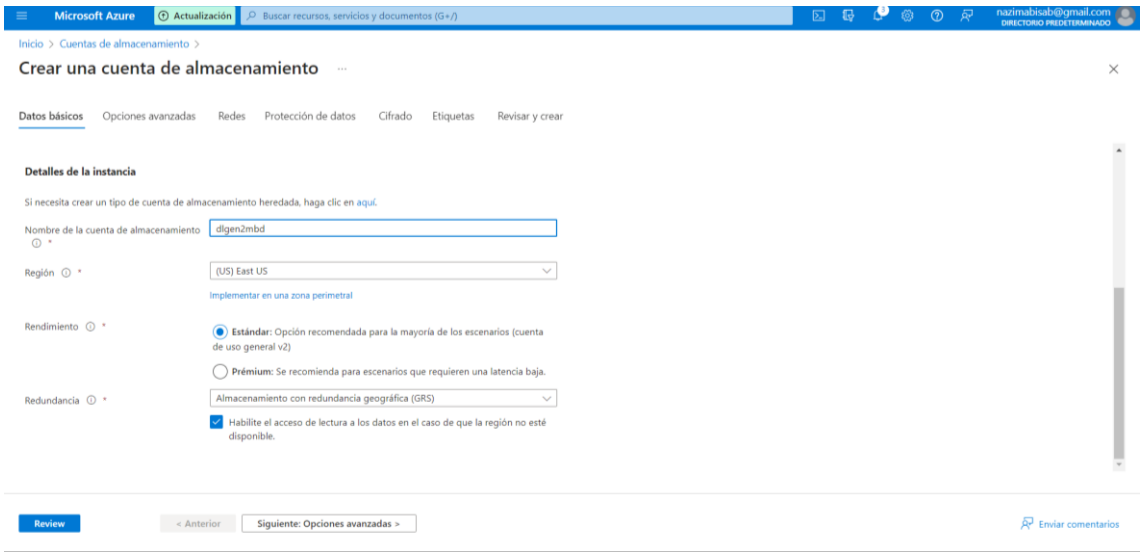
2- Crear un grupo de recursos

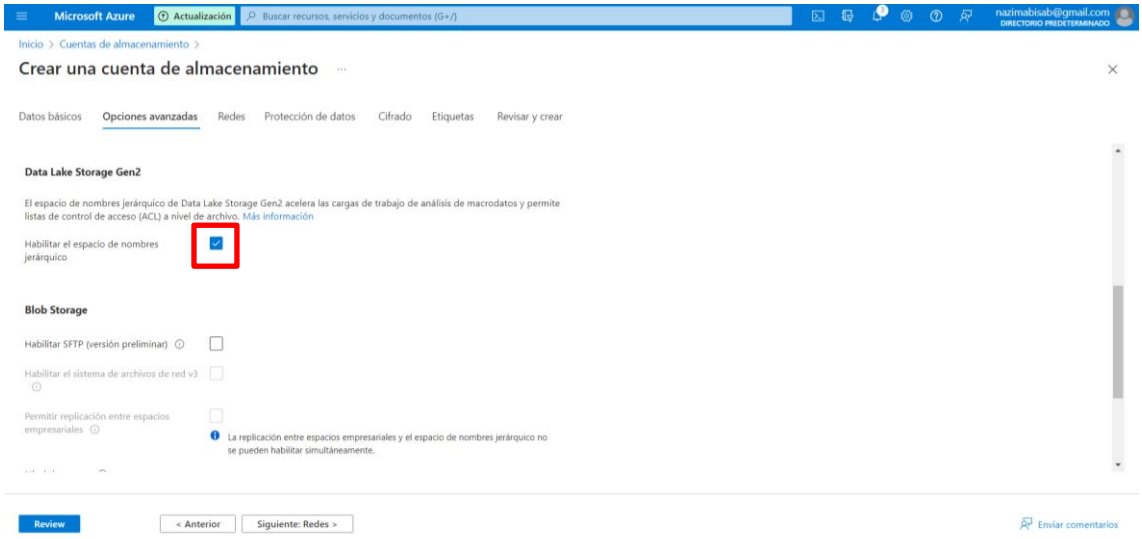




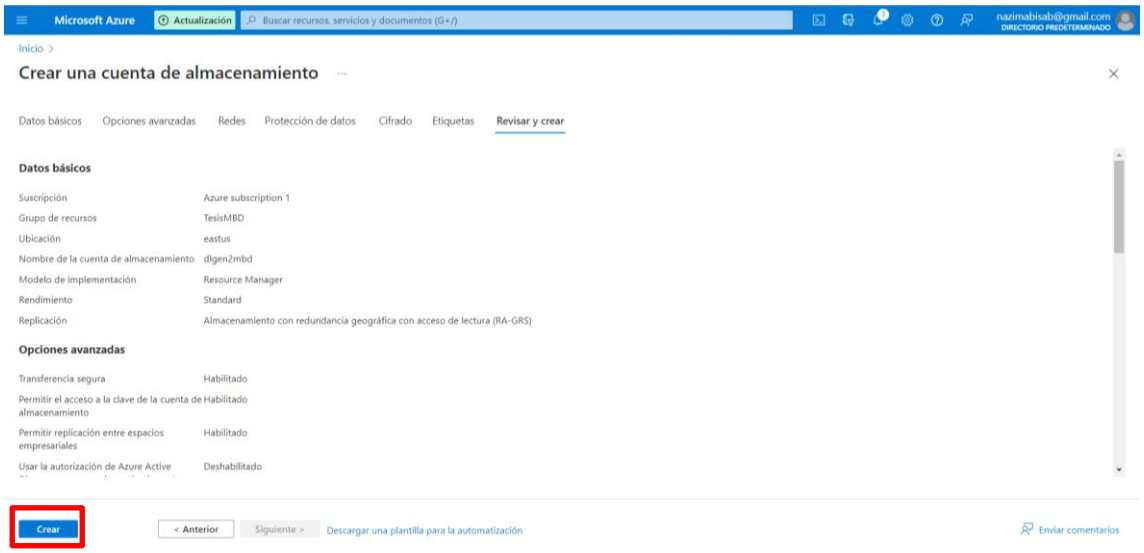
3- Crear cuenta de almacenamiento para levantar un Azure Data Lake compatible con generación 2

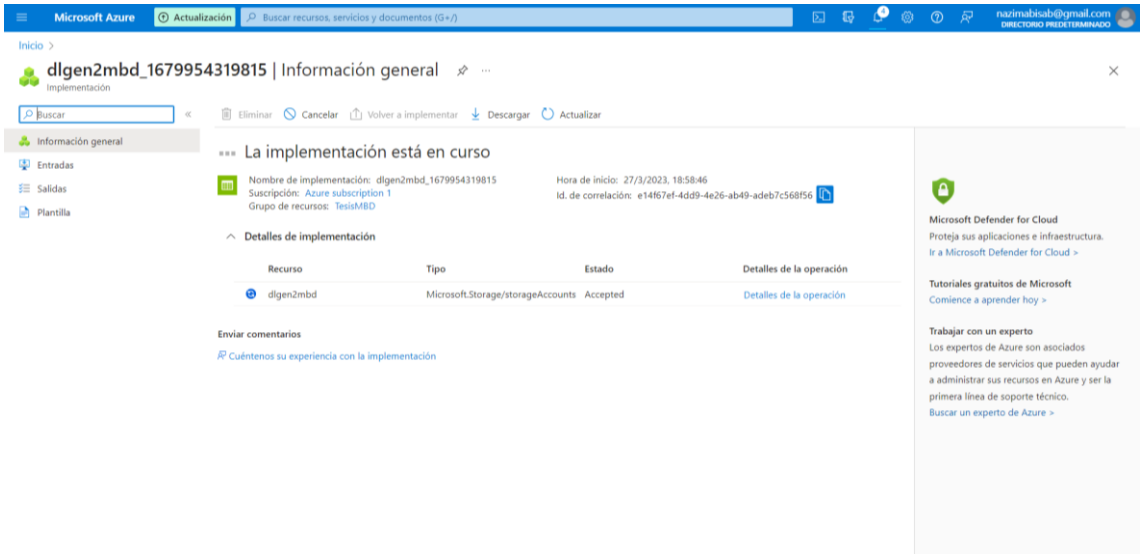




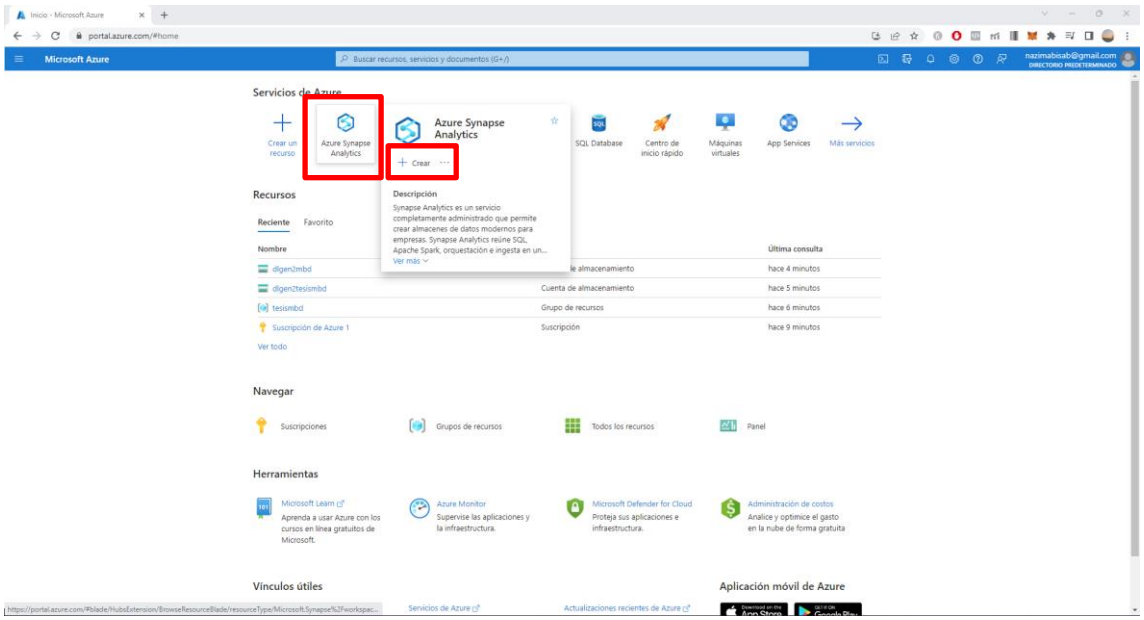


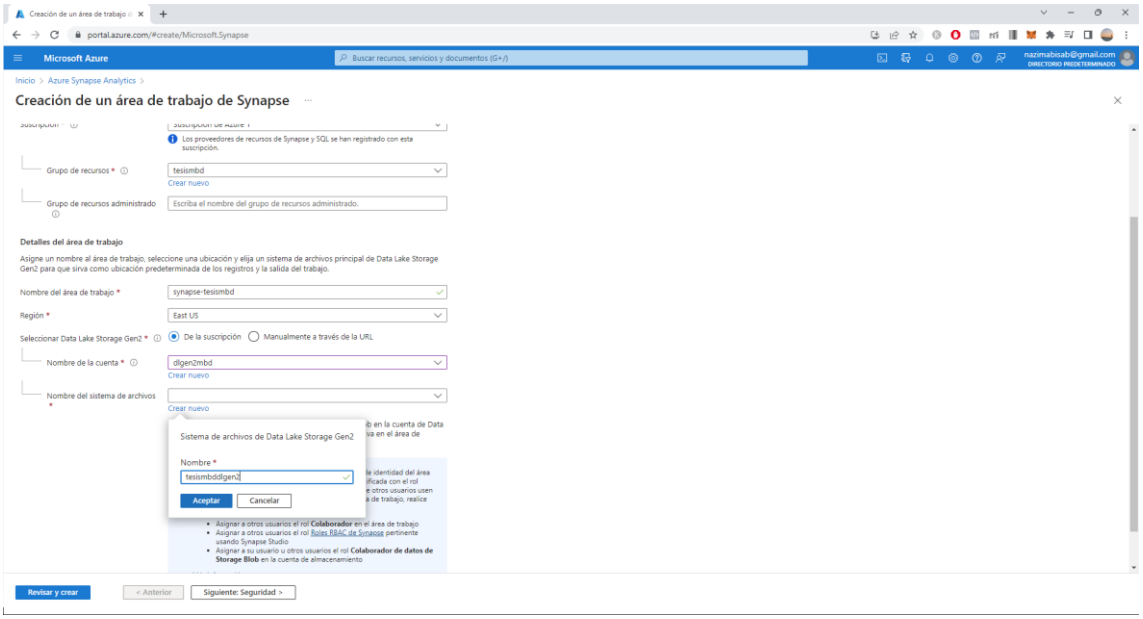
Se debe habilitar el espacio de nombres jerárquico para poder usar Data Lake Storage gen2, necesario para trabajar con Synapse.

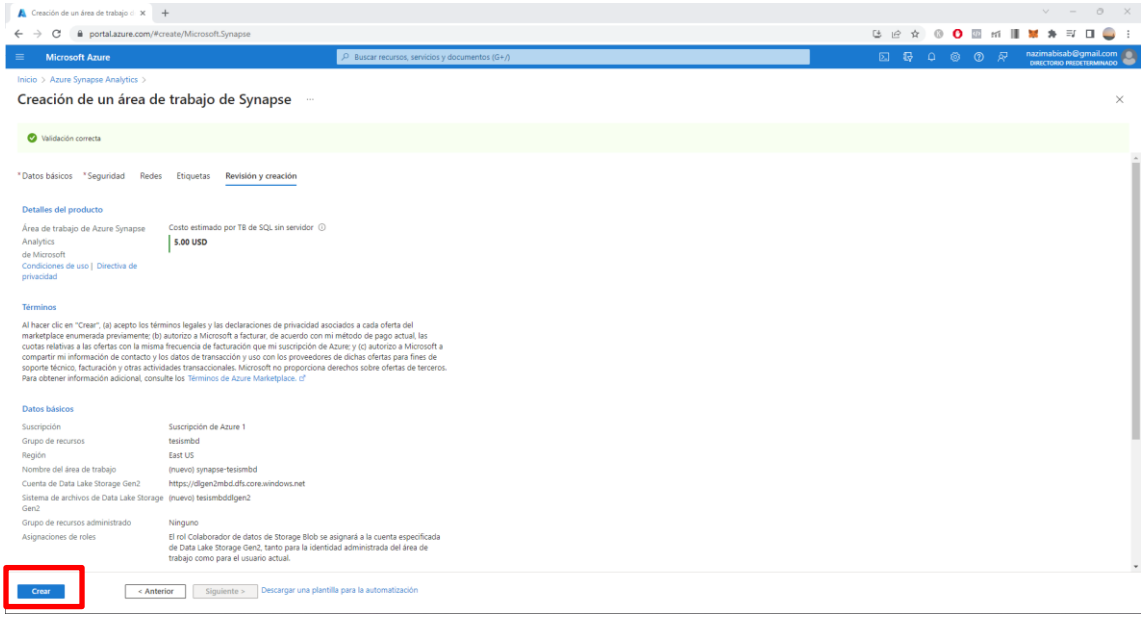




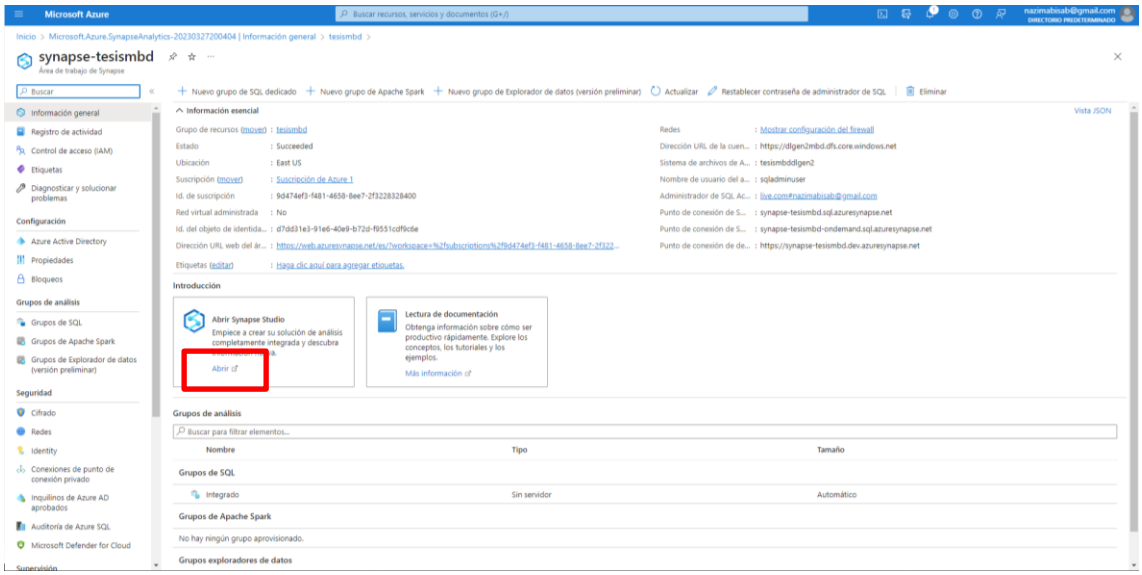
4- Crear espacio de trabajo en Azure Synapse

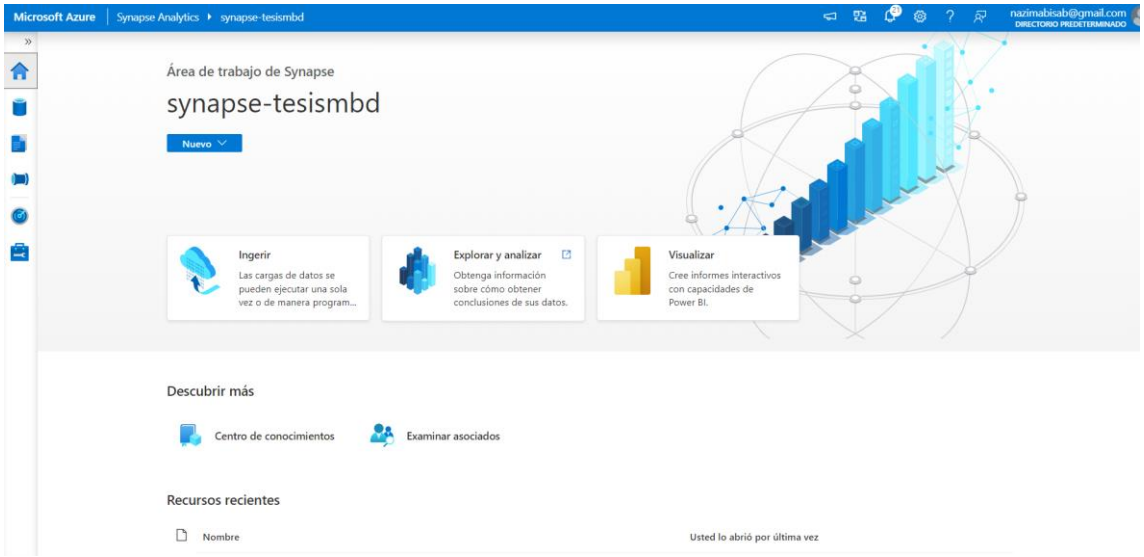



Creación de un área de trabajo de Synapse
Grupo de recursos *
tesismbd
Crear nuevo
Grupo de recursos administrado
Escriba el nombre del grupo de recursos administrado.
Detalles del área de trabajo
Nombre del área de trabajo *
synapse-tesismbd
Región *
East US
Seleccionar Data Lake Storage Gen2 *
De la suscripción
Manualmente a través de la URL
Nombre de la cuenta *
dlgen2mbd
Crear nuevo
Nombre del sistema de archivos *
Crear nuevo
Sistema de archivos de Data Lake Storage Gen2
Nombre *
tesismbddlgen2
Aceptar
Cancelar
Revisar y crear
< Anterior
Siguiente: Seguridad >


Creación de un área de trabajo de Synapse
Validación correcta
Datos básicos
Seguridad
Redes
Etiquetas
Revisión y creación
Detalles del producto
Área de trabajo de Azure Synapse Analytics de Microsoft
Condiciones de uso | Directiva de privacidad
Costo estimado por TB de SQL sin servidor
5.00 USD
Términos
Datos básicos
Suscripción
Suscripción de Azure 1
Grupo de recursos
tesismbd
Región
East US
Nombre del área de trabajo
(nuevo) synapse-tesismbd
Cuenta de Data Lake Storage Gen2
https://dlgen2mbd.dfs.core.windows.net
Sistema de archivos de Data Lake Storage Gen2
(nuevo) tesismbddlgen2
Grupo de recursos administrado
Ninguno
Asignaciones de roles
El rol Colaborador de datos de Storage Blob se asignará a la cuenta especificada de Data Lake Storage Gen2, tanto para la identidad administrada del área de trabajo como para el usuario actual.
Crear
< Anterior
Siguiente >
Descargar una plantilla para la automatización

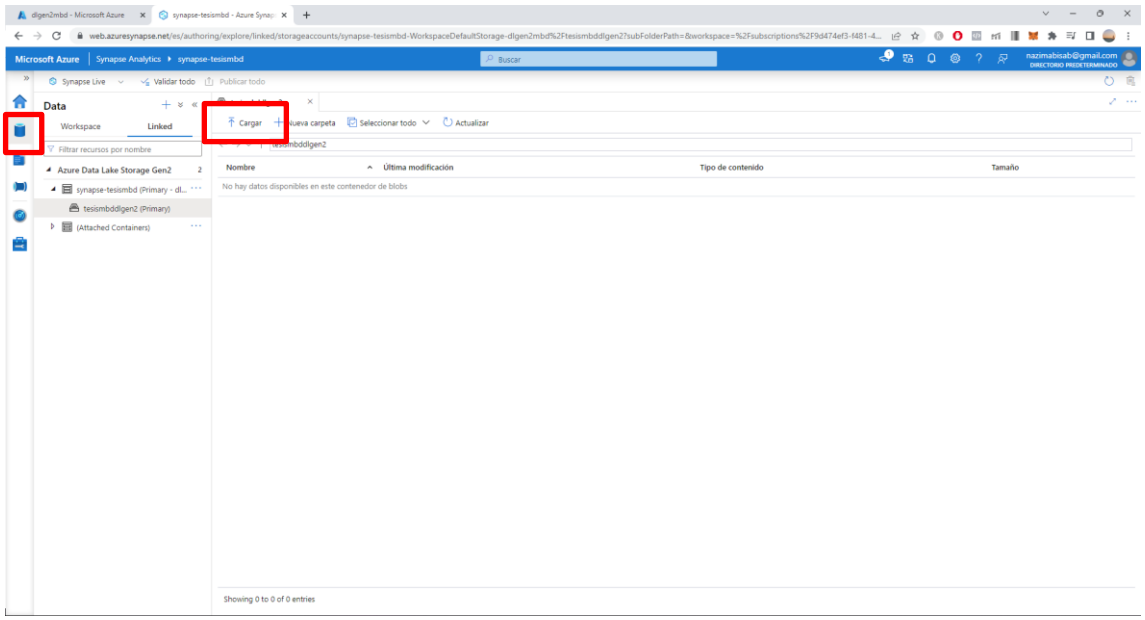
5- Acceder a Azure Synapse Studio

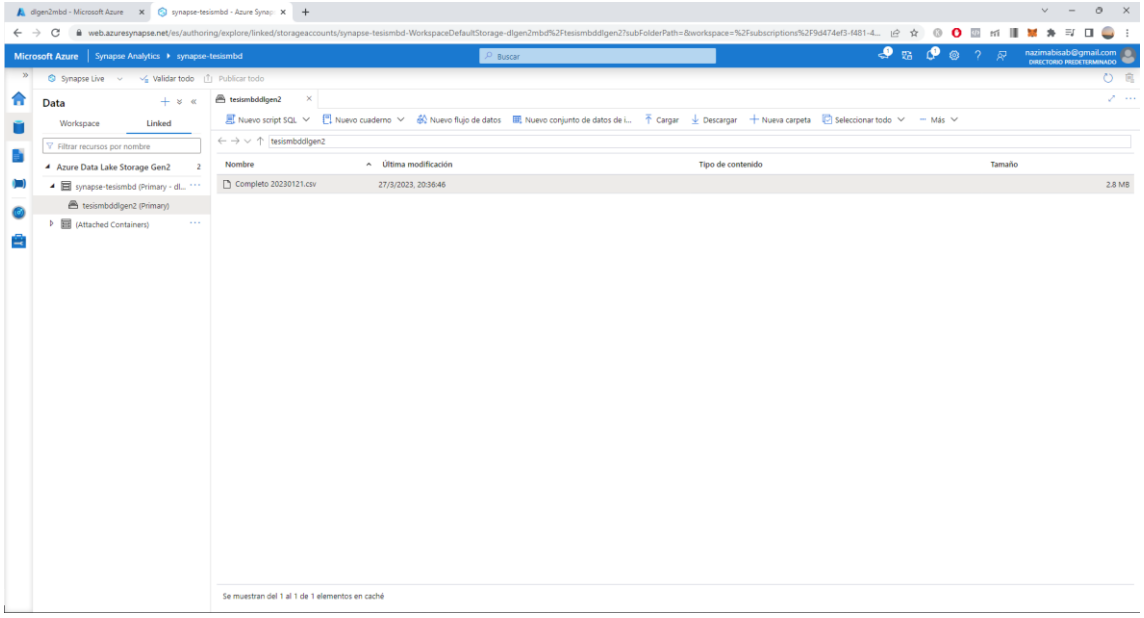




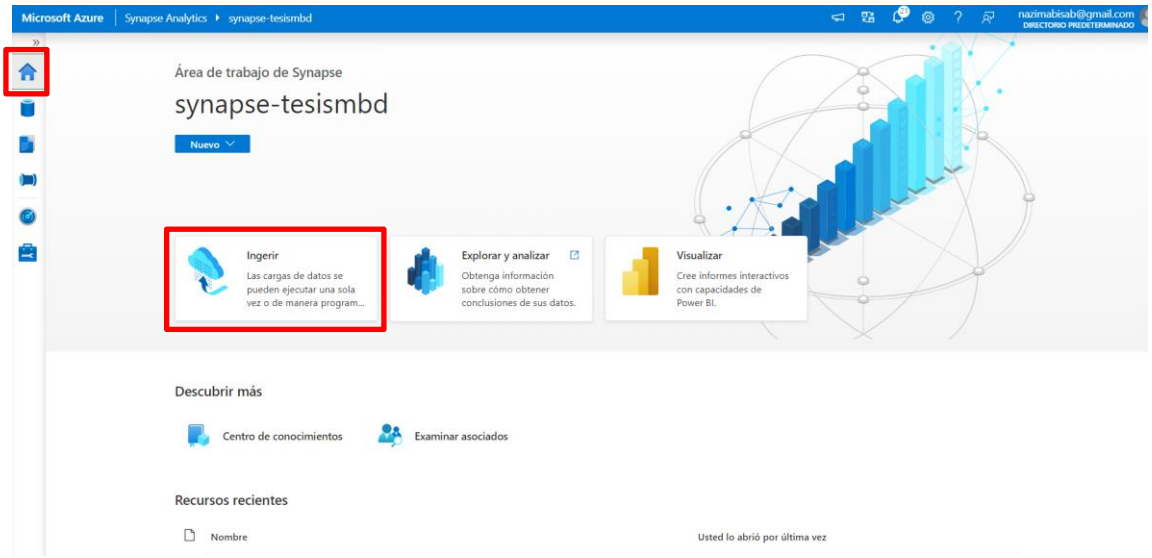
6- Carga de datos en Data Lake

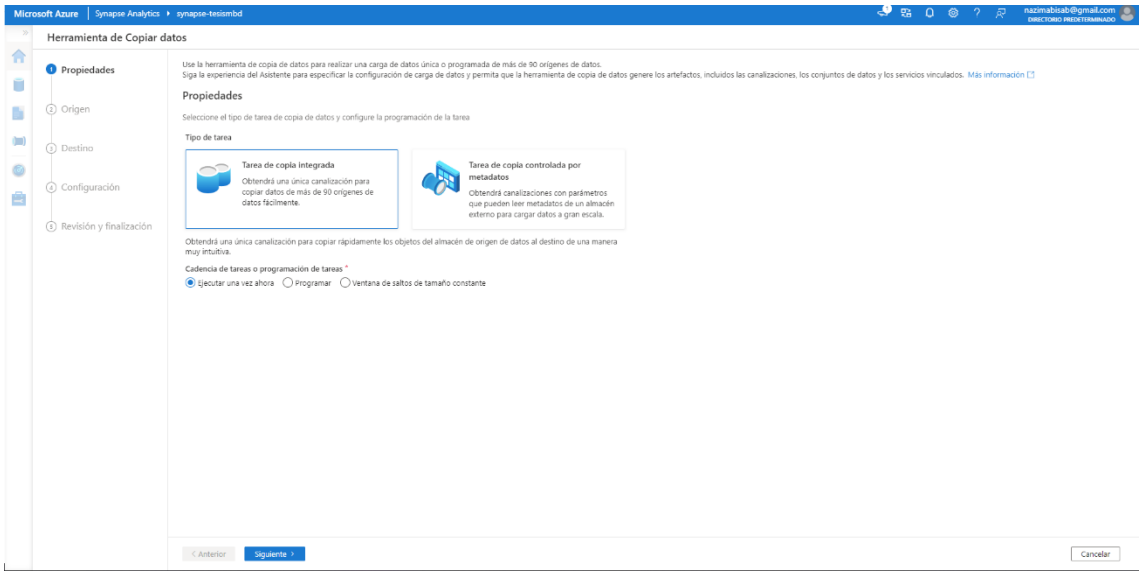
A modo de ejemplo, se cargará directamente el Dataset ya transformado y consolidado al Data Lake.

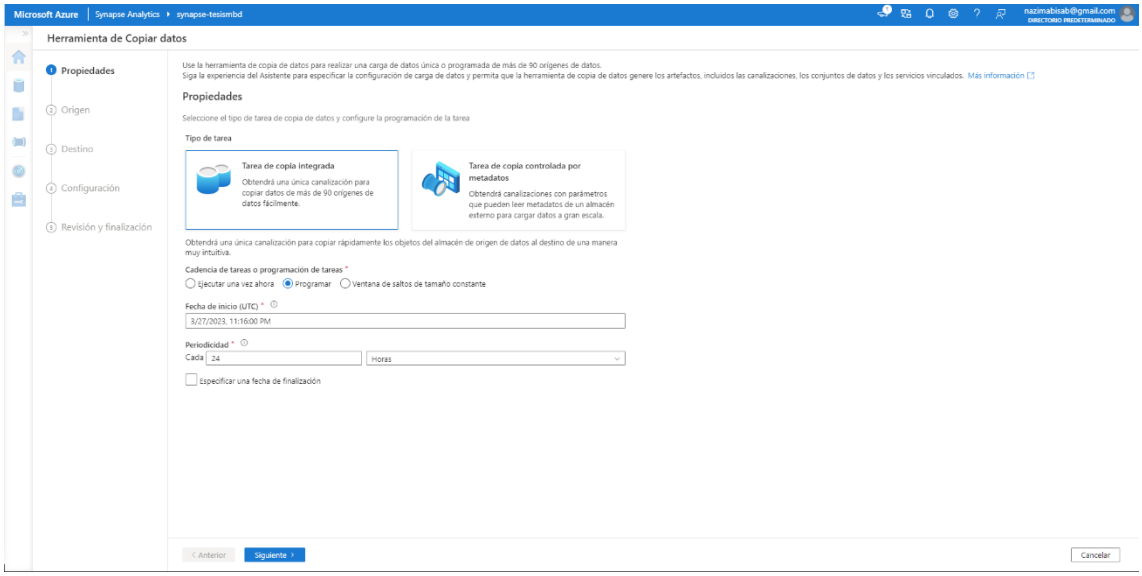




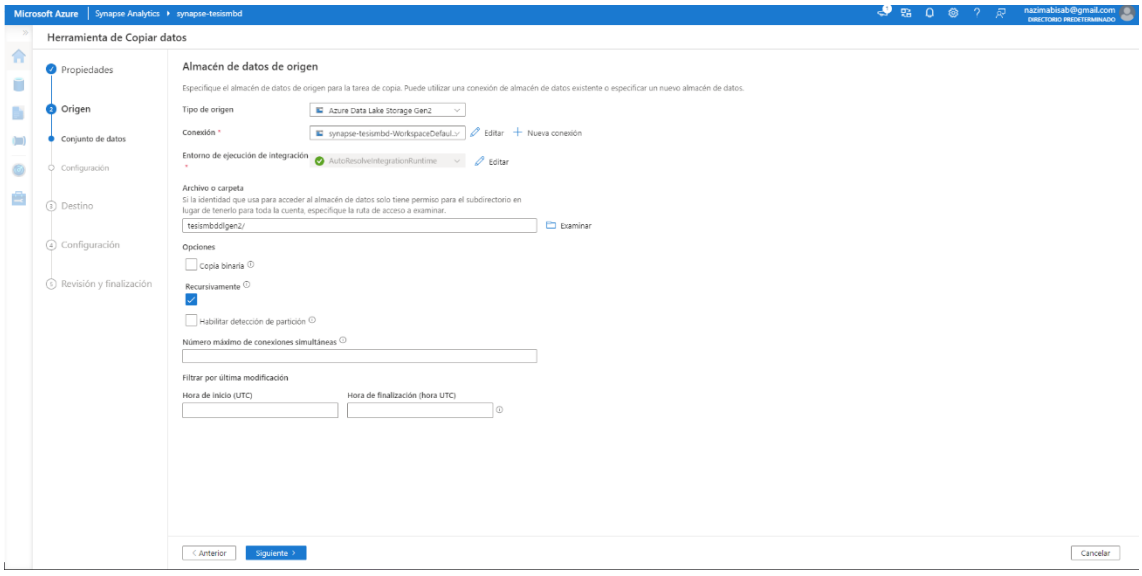
7- Ingestión de datos

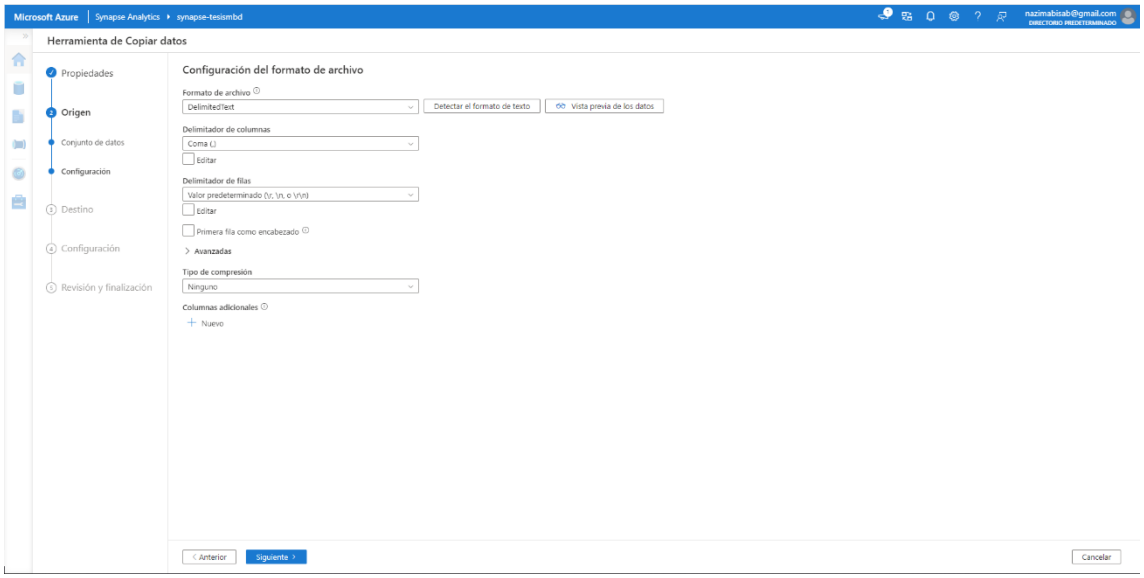


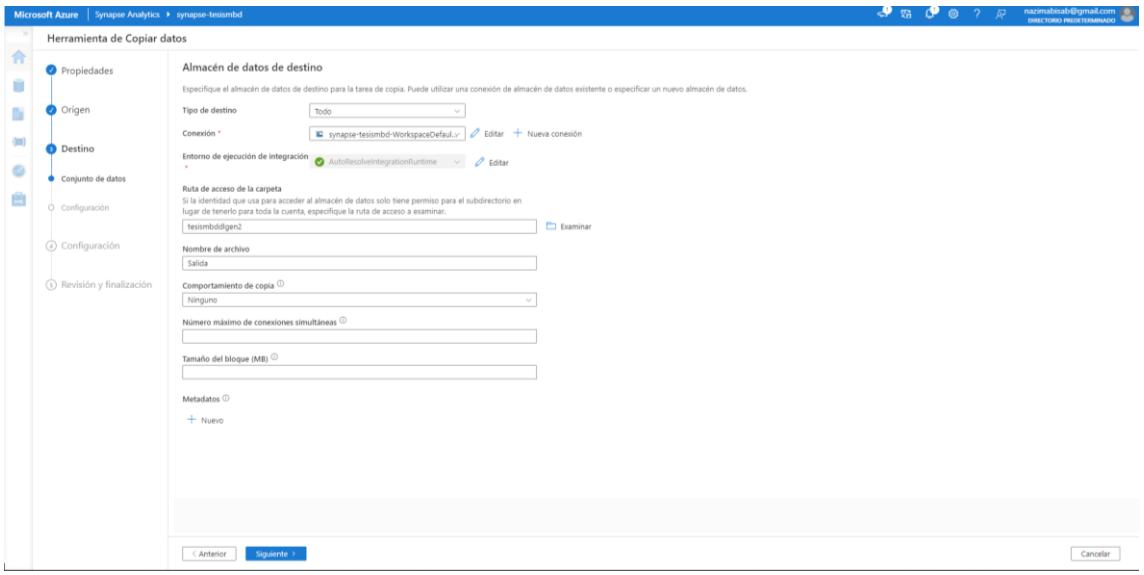


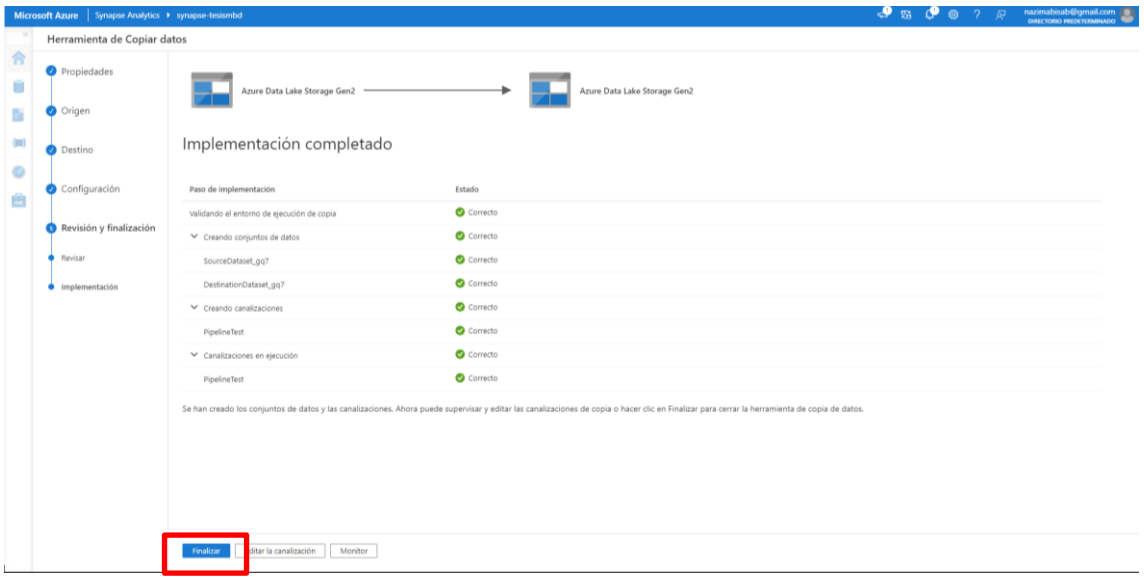


Se puede ejecutar tanto una carga manual como una carga programada en Batch, donde se debe completar la frecuencia y la hora de la carga. Para este ejemplo, se realizará solamente la carga manual.

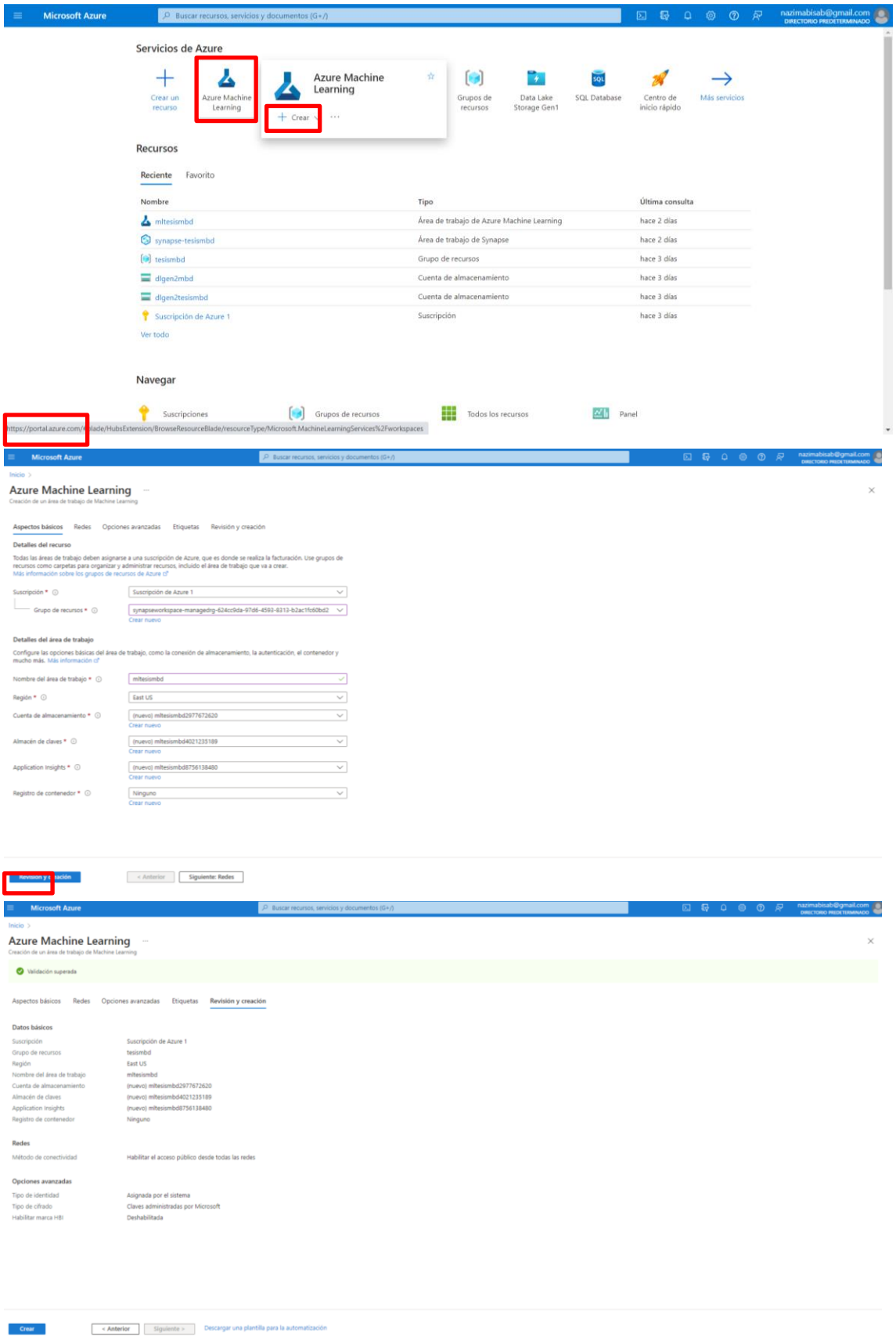



Microsoft Azure
Synapse Analytics
synapse-tesismbd
Herramienta de Copiar datos
Propiedades
Origen
Conjunto de datos
Configuración
Destino
Configuración
Revisión y finalización
Configuración del formato de archivo
Formato de archivo
DelimitedText
Detectar el formato de texto
Vista previa de los datos
Delimitador de columnas
Coma (,)
Editar
Delimitador de filas
Valor predeterminado (\r, \n, o \r\n)
Editar
Primera fila como encabezado
Avanzadas
Tipo de compresión
Ninguno
Columnas adicionales
Nuevo
Anterior
Siguiente
Cancelar

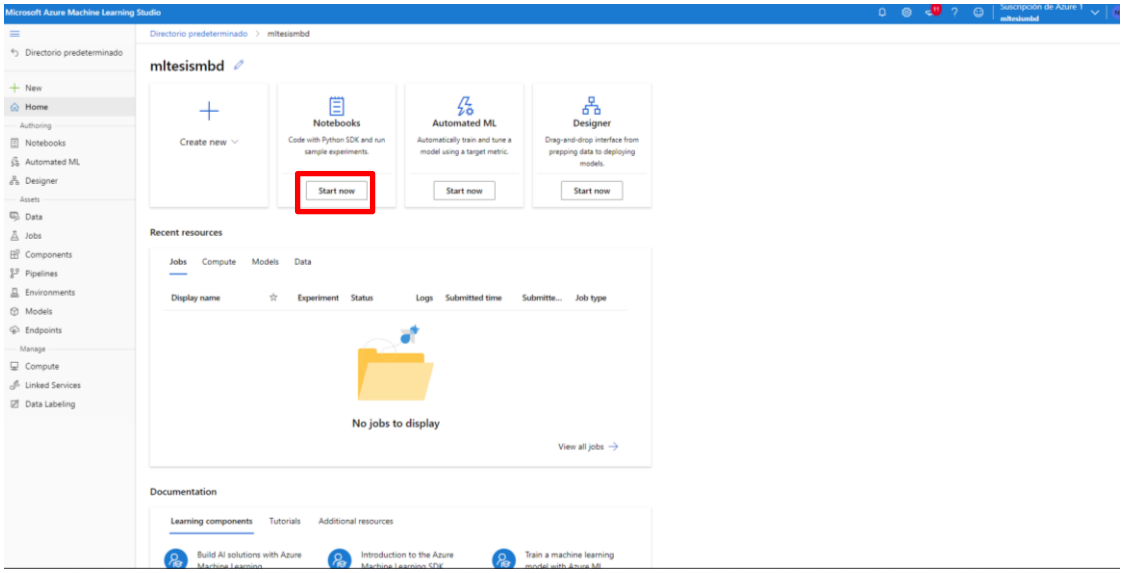

Microsoft Azure
Synapse Analytics
synapse-tesismbd
Herramienta de Copiar datos
Propiedades
Origen
Destino
Conjunto de datos
Configuración
Configuración
Revisión y finalización
Almacén de datos de destino
Tipo de destino
Todo
Conexión
Editar
Nueva conexión
Entorno de ejecución de integración
AutoResolveIntegrationRuntime
Editar
Ruta de acceso de la carpeta
tesismbddlgen2
Examinar
Nombre de archivo
Salida
Comportamiento de copia
Ninguno
Número máximo de conexiones simultáneas
Tamaño del bloque (MB)
Metadatos
Nuevo
Anterior
Siguiente
Cancelar

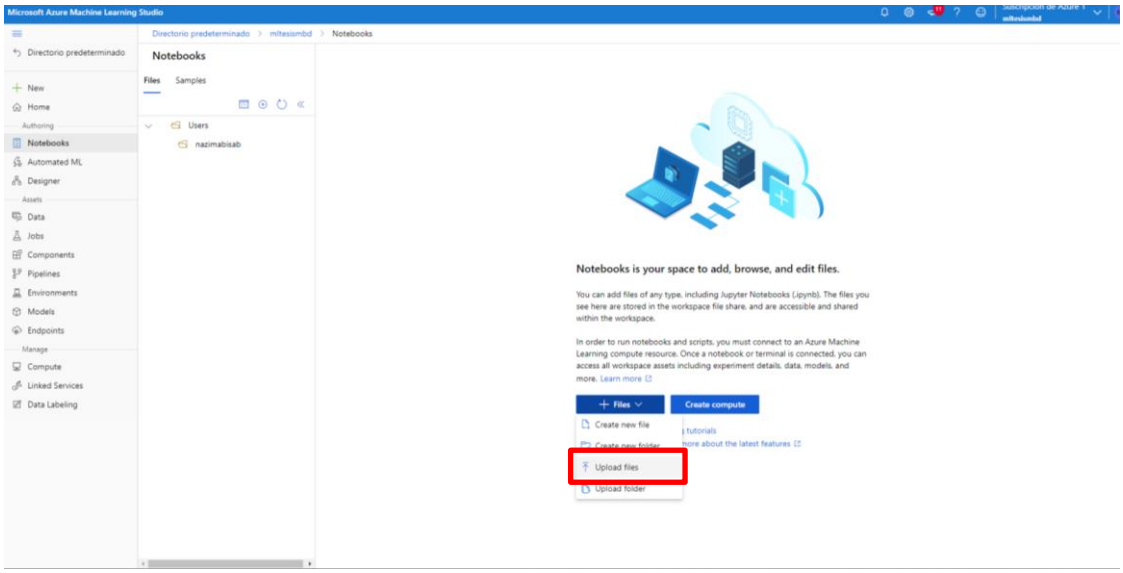

Microsoft Azure
Synapse Analytics
synapse-tesismbd
Herramienta de Copiar datos
Propiedades
Origen
Destino
Configuración
Revisión y finalización
Revisar
Implementación
Azure Data Lake Storage Gen2
Azure Data Lake Storage Gen2
Implementación completado
Paso de implementación
Estado
Validando el entorno de ejecución de copia
Correcto
Creando conjuntos de datos
Correcto
SourceDataset_gq7
Correcto
DestinationDataset_gq7
Correcto
Creando canalizaciones
Correcto
PipelineTest
Correcto
Canalizaciones en ejecución
Correcto
PipelineTest
Correcto
Se han creado los conjuntos de datos y las canalizaciones. Ahora puede supervisar y editar las canalizaciones de copia o hacer clic en Finalizar para cerrar la herramienta de copia de datos.
Finalizar
Editar la canalización
Monitor

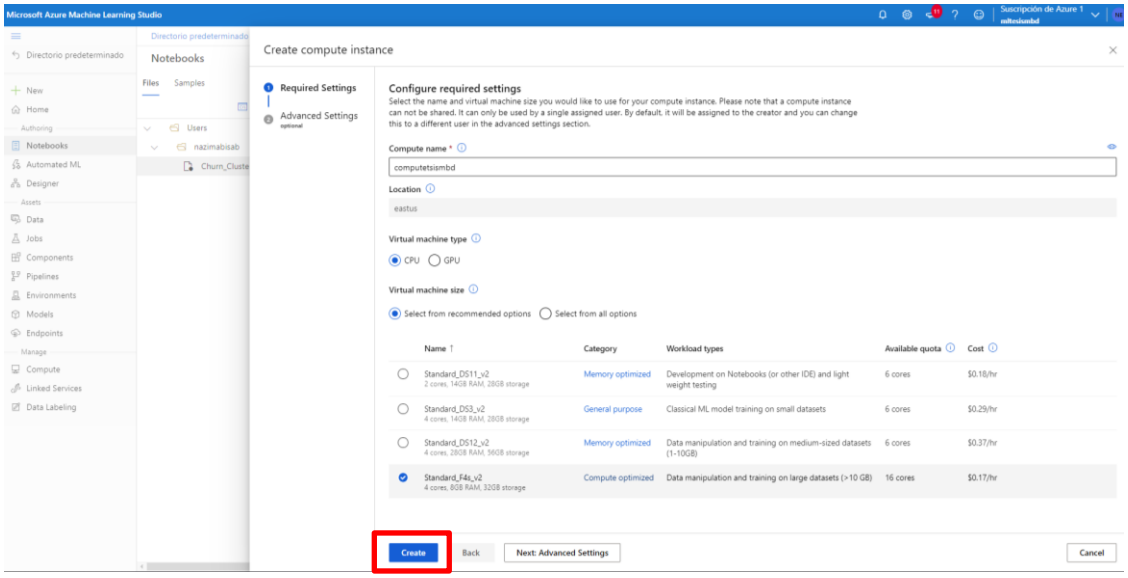
8- Crear servicio de Azure Machine Learning



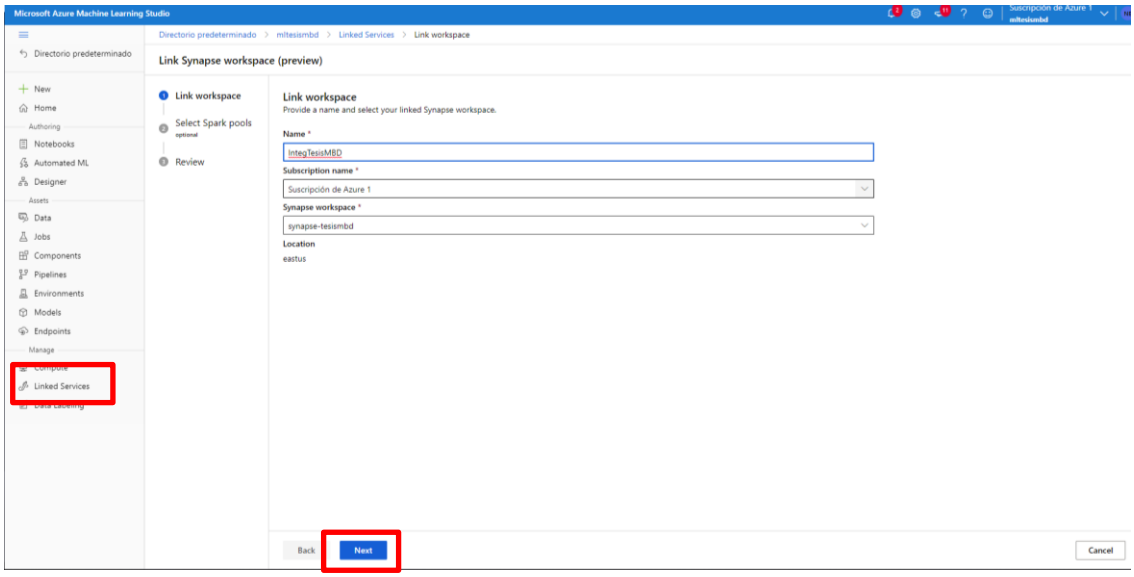
9- Carga del Notebook de predicción de Churn y asignación de recursos desde Azure Machine Learning Studio



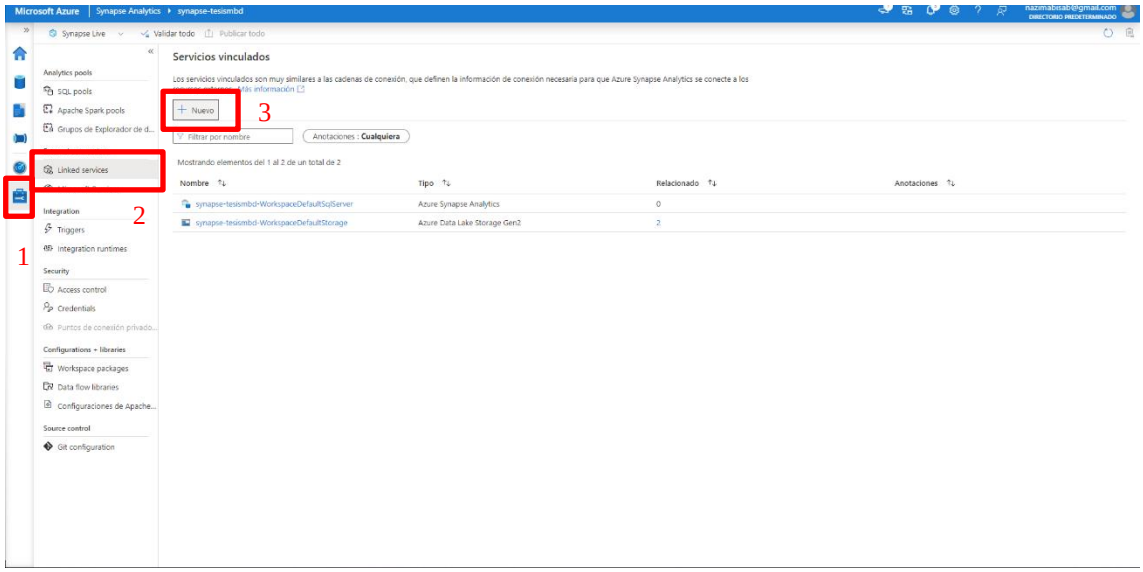


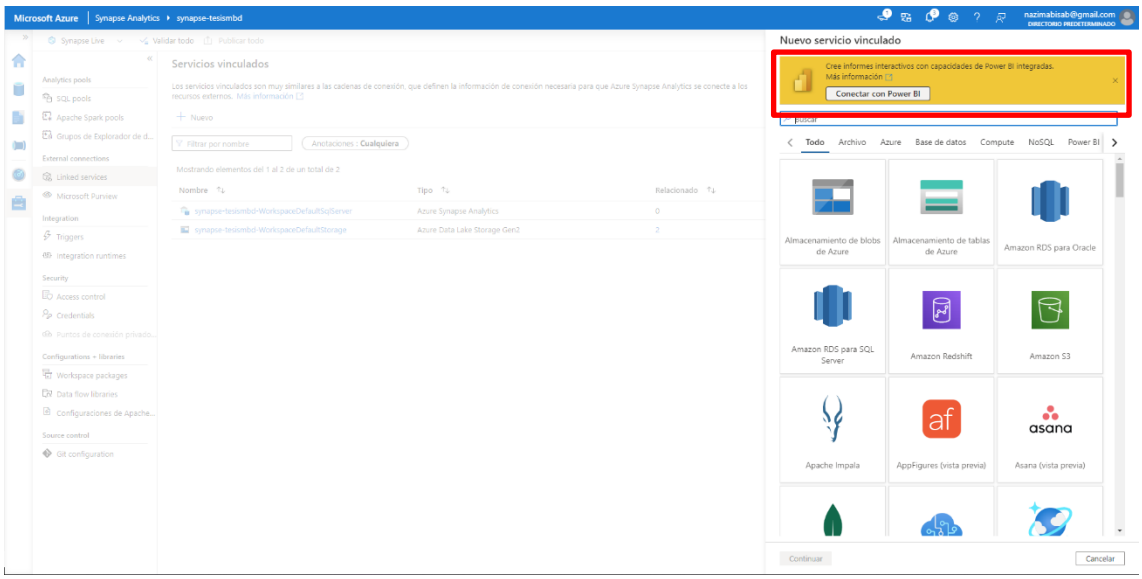

Microsoft Azure Machine Learning Studio
Directorio predeterminado
Notebooks
Files
Samples
Users
nazimabisab
Churn_Cluste
New
Home
Authoring
Notebooks
Automated ML
Designer
Assets
Data
Jobs
Components
Pipelines
Environments
Models
Endpoints
Manage
Compute
Linked Services
Data Labeling
Create compute instance
Required Settings
Advanced Settings
optional
Configure required settings
Select the name and virtual machine size you would like to use for your compute instance. Please note that a compute instance can not be shared. It can only be used by a single assigned user. By default, it will be assigned to the creator and you can change this to a different user in the advanced settings section.
Compute name *
computetsismbd
Location
eastus
Virtual machine type
CPU
GPU
Virtual machine size
Select from recommended options
Select from all options
Name
Category
Workload types
Available quota
Cost
Standard_DS11_v2
2 cores, 14GB RAM, 28GB storage
Memory optimized
Development on Notebooks (or other IDE) and light weight testing
6 cores
$0.18/hr
Standard_D53_v2
4 cores, 14GB RAM, 28GB storage
General purpose
Classical ML model training on small datasets
6 cores
$0.29/hr
Standard_DS12_v2
4 cores, 28GB RAM, 56GB storage
Memory optimized
Data manipulation and training on medium-sized datasets (1-10GB)
6 cores
$0.37/hr
Standard_F4s_v2
4 cores, 8GB RAM, 32GB storage
Compute optimized
Data manipulation and training on large datasets (>10 GB)
16 cores
$0.17/hr
Create
Back
Next: Advanced Settings
Cancel

10- Vinculación de Azure Machine Learning Studio con Synapse

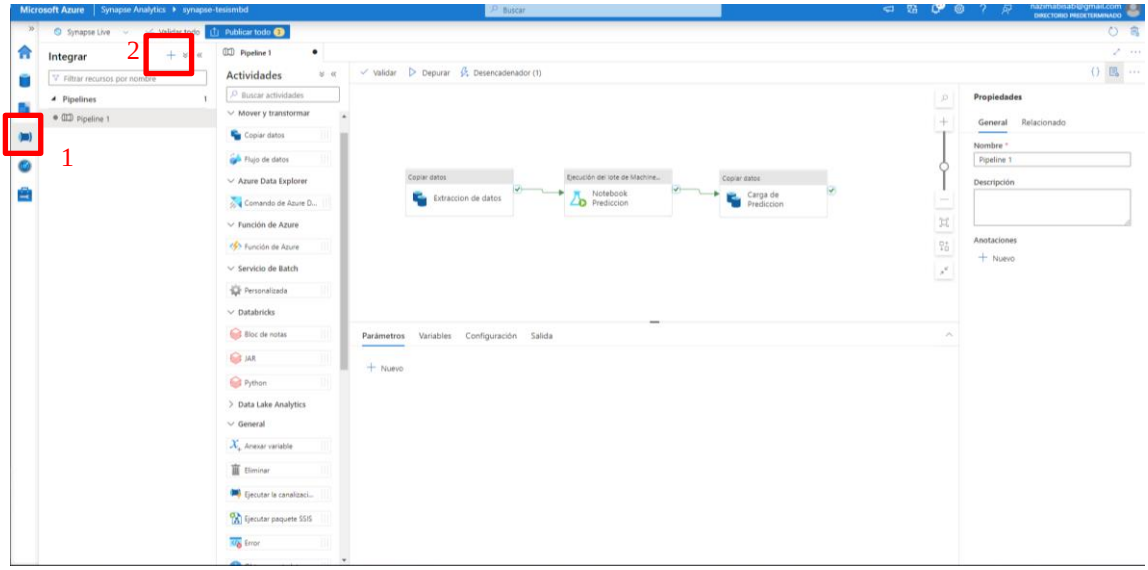


11- Vincular servicio de Power BI





12- Creación del Pipeline



Desde aquí se puede acceder a la predicción que está almacenada en el Azure Data Lake y crear un Dashboard con el Power BI Vinculado en el paso 12.