UNIVERSIDAD DE ANTIOQUIA

Modelo predictivo de Churn de clientes para el negocio de Telecomunicaciones

Autor(es)

Andrés Felipe Echeverri Giraldo

Universidad de Antioquia

Facultad de Ingeniería, Departamento de Ingeniería Electrónica y de Telecomunicaciones

Medellín, Colombia

2019

CONTENIDO

Resumen

El Churn (abandono) es uno de los más grandes problemas en el negocio de las telecomunicaciones. Dado que es mucho más costoso atraer clientes nuevos que retener los existentes se deben crear estrategias que permitan de manera proactiva predecir y prevenir el Churn, permitiendo a su vez la fidelización del cliente. En este trabajo se describe el ciclo de vida y procedimientos necesarios para elaborar un modelo predictivo de Churn. Como resultado se obtuvo una base de datos centralizada y un algoritmo de Machine Learning supervisado desarrollado en Python capaz de predecir hasta un 66% del Churn al mes siguiente.

1. Introducción

En la era digital el mercado de las telecomunicaciones se ha visto saturado por una alta oferta y demanda de servicios, esto obligó a las compañías a transformarse hacia un enfoque centrado en el cliente, y en pro de este se deben tomar acciones que lleven a las empresas un paso adelante. Parte de esta transformación requirió una reinvención de los métodos tradicionales de mercadeo y allí entra el Machine Learning a jugar un papel trascendental en para la estrategia comercial.

Una de las mayores preocupaciones dentro de una compañía de telecomunicaciones es la alta tasa de cancelaciones o abandono por parte de los clientes, esto genera un problema de capital dado que, por un lado, atraer un cliente nuevo implica una estrategia comercial de adquisición siendo necesario el uso de personal de ventas, herramientas de difusión y una fuerte puja de marketing, por el otro lado, la experiencia del área ha detectado que aquellos clientes que se retiran rara vez regresan. Como medida de acción se desea construir un modelo de Machine Learning que permita predecir aquellos clientes que podrían cancelar sus productos con la compañía, el cuál servirá de insumo para la elaboración de campañas de retención.

Con ayuda de técnicas de Machine Learning se busca que, a partir del historial del cliente se pueda predecir la propensión de estos hacía el abandono. Para lograr este objetivo es necesario determinar los motivos o causas que pueden conducir a un abandono, en esto podría verse involucrada información tal como hábitos de consumo, preferencias personales, y otro tipo de características que serán analizadas en el proceso de construcción del modelo. Para la implementación del modelo se utilizaron las siguientes técnicas supervisadas, árboles de decisión, random forest, máquinas de soporte vectorial y XGBoost, y con ayuda de estás se construyó un modelo que dé solución a las necesidades del negocio.

En este informe se detallan los pasos necesarios para la implementación de un modelo predictivo de abandono, inicialmente se definen los objetivos que se desearon alcanzar, seguidamente se detallan los conceptos importantes para comprender la implementación centrado en aquellos que fueron necesarios para el desarrollo de este problema en específico. Esta restricción es importante ya que el abanico de conceptos y opciones es demasiado extenso y no se desean ahondar en tanto detalle. Posteriormente se explica la metodología usada basada en el modelo CRISP-DM para minería de datos y finalmente se detallan los resultados obtenidos.

2. Objetivos

2.1 Objetivo General

Desarrollar un modelo de Churn de clientes usando técnicas de Machine Learning que permita, de manera proactiva, predecir el abandono de clientes.

2.2 Objetivos Específicos

- Construir una base de datos centralizada que incluya la mayor cantidad de información relevante que sea posible obtener acerca del cliente, este proceso es de vital importancia para poder estructurar la información que sustente las decisiones que se tomen en la implementación del modelo, adicionalmente será un insumo para el entrenamiento del modelo.
- Agrupar los clientes con un algoritmo de Clustering con el fin de encontrar patrones en sus hábitos de consumo, que permitan entender las necesidades y características de cada grupo.
- Definir la población objetivo para el modelo teniendo en cuenta las características que definen al cliente y la necesidad propia del negocio. Una población objetivo bien definida permite al modelo estar más ajustado al momento del entrenamiento.
- Evaluar y comparar el desempeño de al menos tres técnicas supervisadas de Machine Learning tales como árboles de decisión, random forest, máquinas de soporte vectorial, redes neuronales y XGBoost.
- Crear una estrategia de seguimiento, mantenimiento y despliegue del modelo para garantizar su funcionamiento en el día a día, y así también garantizar una buena interpretación de los resultados.

3. Marco Teórico

3.1 Machine Learning

El Machine Learning es un campo de las ciencias de la computación que se enfoca en aprender sobre un conjunto de datos [1]. En este sentido la palabra aprendizaje hace referencia a como los algoritmos son capaces de identificar patrones dentro de un conjunto de datos que representan algún suceso o evento. Actualmente este tema se ha extendido a un gran número de campos, desde el procesamiento del lenguaje natural hasta la robótica, En este informe

nos enfocaremos en una de sus aplicaciones, el mundo de los modelos predictivos el cuál es de relevancia para una gran cantidad de áreas, ya que se puede aplicar tanto para detección temprana del cáncer como para detección de fallos en una máquina industrial.

Los algoritmos de Machine Learning se pueden clasificar en aprendizaje supervisado, semi supervisado y no supervisado. El aprendizaje supervisado se basa en la detección de patrones sobre los datos que permiten clasificar los datos con una etiqueta previamente definida. El algoritmo semi supervisado es relativamente similar al anterior pero también permite crear nuevas clasificaciones a datos sin etiquetar. Finalmente, el no supervisado se encarga de agrupar los datos según categorías definidas propiamente por el algoritmo, este último es altamente utilizado para generar conocimiento adicional de los datos [2].

Los algoritmos de clasificación son capaces de clasificar de manera automática nuevos eventos a partir del conocimiento de eventos pasados, para que esto sea posible se debe contar con información suficiente para identificar las características que permiten que este se ubique en una etiqueta u otra. Como se dijo previamente, el enfoque de este proyecto busca profundizar en el funcionamiento de un algoritmo de clasificación binario el cual determina si un cliente tiene propensión hacia el Churn a partir del conocimiento de las características de aquellos que ya han hecho Churn.

Los eventos están conformados por un conjunto de características y de observaciones. Una característica es una propiedad medible del fenómeno observado, y se define la observación como cada evento independiente que fue registrado. Para este caso cada observación representa a un cliente y las características es cada una de las variables que componen al cliente y que caracterizan sus necesidades y hábitos. Para poder clasificar a clientes sin etiqueta, de cada cliente en la base de entrenamiento se debe conocer con plena certeza si hizo o no Churn.

3.2 Técnicas

Para implementar un algoritmo de Machine Learning se pueden utilizar distintos métodos o técnicas de aprendizaje cuyo objetivo es abordar desde distintos enfoques un mismo problema de clasificación, aunque es posible utilizar un mismo método de aprendizaje para casi todos los problemas de clasificación

esto no sería una buena práctica ya que no se garantiza que las predicciones tengan un desempeño.

Se debe evaluar el desempeño del modelo para que se pueda confiar en sus predicciones, esto significa que el modelo debe ser capaz de generalizar a cualquier data nunca vista dentro de la misma categoría. Esto va de la mano con la elección de la técnica cuyo desempeño es más apropiado para resolver el problema. Las técnicas de aprendizaje supervisado se seleccionaron a partir de los siguientes criterios:

- Buen desempeño para problemas de clasificación binaria.

- Permiten el manejo de grandes bases de datos.

- Sencillas de interpretar

- Técnicas previamente utilizadas en el área para otros modelos

A continuación, se presenta un acercamiento a las técnicas seleccionadas que cumplieron con los criterios mencionados:

3.2.1 Decisión Tree: Un árbol de decisión como lo dice su nombre, es una estructura que se puede interpretar como un grafo en el que cada nodo representa una prueba sobre un atributo, cada rama representa el resultado del test y cada hoja es una categoría o etiqueta, en resumen, un árbol de decisión es un conjunto de reglas que permiten llegar a una conclusión. Los árboles de decisión tienen la ventaja de presentar alta exactitud, son buenos para encontrar relaciones no lineales lo que permite que se adapten a múltiples problemas.

3.2.2 Random Forest: El random forest consiste en un gran número de árboles de decisión, todos funcionan como una estructura en el que cada árbol arroja una predicción, la predicción resultante será aquella que más se repita, tiene la ventaja de que no se ve afectado por los errores individuales de cada árbol.

4.2.3 XGBoost: Es una extensión de los árboles de decisión pero que busca convertir aquellos árboles que son malos predictores en buenos predictores, esto se logra mediante un método iterativo, aumentando el peso de las observaciones más difíciles de clasificar y a su vez creando nuevos árboles para así obtener la predicción final. Tiene la ventaja de ser paralelizable haciéndolo mucho más rápido, adicionalmente se caracteriza por tener muy buen desempeño y con alta adaptabilidad.

3.2.4 Support Vector Machine: Las máquinas de soporte vectorial buscan encontrar un plano n-dimensional que permita distinguir perfectamente las clases de las observaciones, dado que existen muchos posibles hiperplanos que permitan separar las clases, el plano seleccionado será aquel que maximice la distancia al hiperplano de las observaciones, esto tiene la ventaja de permitir un amplio margen para la ubicación de nuevas observaciones y así no atraviesen la frontera.

3.3. Evaluación

Para identificar la técnica que mejor se adapta a la detección del Churn se debe hacer una evaluación de desempeño. Esto ayuda a encontrar el modelo que mejor representa la data y que garantiza el funcionamiento del modelo con los clientes sin categorizar. Para lograr esto se debe partir de un primer modelo tan simple como sea posible, al que corresponde un desempeño medido con ayuda de una métrica de optimización, los siguientes modelos serán mejores o peores en la medida que superen al modelo simple en esa métrica definida. Cada técnica posee un conjunto de parámetros que se pueden configurar de múltiples maneras a fin de ajustar la respuesta del modelo a la entrada de datos dada. A continuación, se hace una explicación a mayor detalle de estos términos claves para entender la evaluación.

3.3.1 Hiperparámetros: Las técnicas por sí solas podrían no ser suficientes para obtener una predicción confiable, cada técnica contiene un conjunto de parámetros que permite aproximar aún más la respuesta a la función que estamos buscando. Ajustar los hiperparámetros es un proceso manual, no estandarizado, y extenso ya que cada solución es especialmente adaptada al problema, cabe aclarar que el proceso puede ser complejo, sin embargo, cada problema contiene un grupo de características que podrían contribuir a reducir el espacio de búsqueda.

3.3.2 Métricas: Para poder seleccionar el mejor modelo primero se debe contestar la pregunta ¿mejor respecto a qué?, para esto existen dos sucesos claves.

- Definir un modelo simple
- Definir la métrica de optimización

El modelo simple consiste en un modelo que no es bueno para generalizar y que solamente identifica de manera semi aleatoria o por tendencia de repetición, esto a fin de tener un desempeño de base que se buscará superar. Las métricas permiten cuantificar el desempeño del modelo y también dependen del

problema a resolver, no centraremos en aquellas utilizadas para problemas de clasificación binario ya que se intenta predecir el abandono [3].

- Accuracy: Mide la ratio de predicciones correctas del total de predicciones.

- Precision: Mide la proporción de abandonos correctamente detectados entre todos los predichos.

- Recall: Mide la proporción de abandonos correctos detectados contra el total de abandonos reales.

- Curva ROC: Mide la capacidad para discriminar casos positivos de casos negativos. Tiene un área bajo la curva (AUC) entre 1 y 0, una curva ROC con AUC igual 1 es un clasificador perfecto.

Para este problema en particular se utilizó el Recall como métrica de optimización, dado que se priorizó la detección de la mayor cantidad posible de abandonos [4].

4. Metodología

La metodología está basada en el modelo CRISP-DM el cual describe el ciclo de vida de un proyecto de análisis de datos.

4.1 Recolección de datos

En este paso se buscaron todas las fuentes de información disponibles en el área, sin embargo, solo se seleccionaron aquellas fuentes con cifras oficiales y cuya periodicidad de actualización fuera menor o igual a un mes. Como cada observación debe representar un único cliente para lograr su correcto perfilamiento, por lo tanto, toda la información recolectada debió ser agregada a este nivel.

Calidad de los datos

Una vez se ha recolectado la información de las distintas fuentes se debió validar que esta pueda cumplir con su propósito, ya que no existe una estandarización en cuanto a calidad se refiere, parte del proceso también consistió en definir los atributos que deben obedecer estos datos para garantizar un buen modelamiento, estos atributos son:

Una vez se ha recolectado la información de las distintas fuentes se debió validar que esta fuese capaz de cumplir con su propósito [5], ya que no existe una estandarización en cuanto a calidad se refiere, parte del proceso también

consistió en definir los atributos que deben obedecer estos datos para garantizar un buen modelamiento, estos atributos son:

Actualización: Se debió tener pleno conocimiento de la periodicidad de actualización de cada fuente, esto es indispensable puesto que de esta periodicidad se desprende la periodicidad de ejecución misma del modelo. Dado que solamente se seleccionó información con una periodicidad de actualización menor o igual a un mes, fue posible garantizar un periodo de ejecución del modelo a nivel mensual.

Coherencia: Fue necesario garantizar que todos los datos del cliente estuvieran dentro de los límites máximos y mínimos de cada variable definidos por la empresa, así mismo se garantizó que las variables categóricas estuviesen estandarizadas.

Accesibilidad: Como el modelo implementado se debe correr de manera periódica, todo dato utilizado debe ser posible de obtener dentro del periodo definido, en consecuencia, se descartó toda información que fuera sensible a desaparecer, aquellas que pudieran no estar disponibles por problemas en las fuentes o que por limitaciones de gobierno de información no se pueda utilizar.

Relevancia: La compañía almacena grandes cantidades de información de múltiples indoles, sin embargo, esto tiene sus limitantes ya que utilizar la información sin discriminar puede generar sobre costos en el tiempo de ejecución y de procesamiento. Toda la información utilizada fue estrictamente seleccionada a partir del conocimiento del negocio y del entendimiento del cliente.

Nota: Dado que el procesamiento de la información es el proceso más importante se debe sustentar cada decisión en base al conocimiento de los datos, el conocimiento del negocio y con apoyo del personal encargado del gobierno de los datos.

Exploración de los datos

Previamente se hizo una depuración de los datos, con el fin de descartar datos que pudieran no tener calidad suficiente o que no brindaban suficiente información. La siguiente etapa se centró en un análisis a mayor profundidad de los datos seleccionados buscando obtener información adicional que pudiera no estar explicita en estos.

Se construyó una visualización (para esto se utilizó la herramienta Tableau Desktop) con las agrupaciones de los datos, sus distribuciones, agrupamientos y tendencias, esto permitió crear nuevas variables a partir de las existentes, y

transformar algunas existentes para que su información fuese más concluyente. Otra de las ventajas del análisis exploratorio es que permitió observar los rangos de valores que tomaban las variables, así como sus tipos de datos y si contenían datos nulos.

Definición del público objetivo:

Teniendo claras las necesidades de la compañía y el problema a resolver, se delimitó el alcance del problema a un público objetivo formado por dos conjuntos de clientes. Un conjunto formado por clientes cuyo estado de abandono es conocido y otro sobre el cual se desea identificar su propensión al abandono. Este proceso se hizo con completo acompañamiento de expertos del área y con un previo conocimiento de la información disponible para alcanzar a este público.

Modelamiento:

Una vez se completaron los pasos anteriores se garantizó que la data se encontraba en condiciones óptimas para el entrenamiento, lo que permite seguir a la etapa de modelamiento. Para esta etapa primero se particionó la data en entrenamiento y prueba, esto permite tener un grupo de control para evaluar el desempeño del modelo. El siguiente paso es crear un modelo tonto o básico a partir del cual se podrá definir si existe o no existe una mejora en el desempeño del modelo, en otras palabras, es el modelo que vencer.

Como se mencionó antes, se creó un primer modelo simple cuya predicción sirvió de punto de comparación con las demás técnicas implementadas. Seleccionar las técnicas requirió entender que la data presentaba problemas de desbalanceo, en consecuencia, se debieron seleccionar unas técnicas sobre otras. Posteriormente se definieron un grupo de métricas respecto a la cuales se medirá el desempeño del modelo, esto será importante para determinar la técnica que brinda la mejor solución al problema.

Evaluación:

Luego de la construcción de todos los modelos y con las métricas definidas, se procedió a seleccionar el modelo cuyo desempeño fue aparentemente superior en esas métricas respecto a los otros. La técnica seleccionada contiene un conjunto de parámetros que permiten afinar aún más la respuesta del modelo. Estos parámetros se conocen como hiperparámetros (un hiperparámetro es una característica ajustable que rige la forma en como el modelo aprende) y afectan directamente el proceso de entrenamiento, a su vez afectando el desempeño, adicionalmente se evaluó la alternativa de incluir un sobre muestreo o submuestreo para disminuir el impacto del desbalanceo de la data.

Antes de proceder al despliegue, se debió evaluar a profundidad el modelo seleccionado, esto implicó una revisión de los procedimientos y pasos seguidos para garantizar que no existan sesgos inducidos en la solución. En este punto se podría considerar adicionar procesos que puedan ayudar a obtener un mejor desempeño, para lo que sería necesario regresar a la etapa de análisis de datos.

Automatización:

Hasta ahora se ha implementado el algoritmo a partir de distintos criterios y pruebas, pero una vez se definieron claramente las variables a usar, el modelo a implementar, la métrica y el público objetivo, es importante definir una metodología o protocolo de actuación el cual rige el despliegue del modelo implementado. Este protocolo documenta el paso a paso para la ejecución, incluyendo las fuentes necesarias, la periodicidad de entrenamiento y despliegue, y el manejo de la información suministrada por el modelo.

Adicionalmente se debe pasar por una etapa de automatización, que como su nombre lo indica, implicó la automatización de todo proceso manual dentro o fuera del algoritmo que pueda hacer no sostenible la ejecución del modelo a través del tiempo, esto implica automatizar consultas, definir estándares de codificación, parametrizar nombres e información temporal o periódica.

Despliegue:

Es la última etapa del ciclo de vida de un proyecto de Machine Learning y consiste en la divulgación de los resultados del modelo. Para la primera ejecución del modelo se tomó una fuente de datos de clientes sin etiquetar del mes de agosto y el modelo les asignó su respectiva propensión predicha de abandono. Cabe recordar que el ciclo de vida de un proyecto de Machine Learning es bidireccional y que a largo plazo podría ser necesario replantear algunos procedimientos, en pro de mantener el modelo en ejecución a través del tiempo.

5. Implementación

Antes de abordar el problema primero se definió qué es Churn y su impacto en el negocio. Se entiende como Churn a la pérdida total del cliente, lo que afecta directamente los ingresos de la empresa y adicionalmente genera un crecimiento de la competencia, en este tipo de negocio usualmente los abandonos se convierten en migraciones de operador. Teniendo claro por qué representa una necesidad la detección del Churn la solución se estructura en cuatro sprints con los

siguientes entregables: Base de datos centralizada, Modelo predictivo, automatización y Leads.

5.1 Centralización de información:

Primero se debió construir una base de datos de entrenamiento, para esto se definieron las variables que la constituyen y el medio para capturarlas y procesarlas. Con ayuda de la herramienta SQL Server 2012 se pudo explorar las bases de datos de la compañía, en las cuales se documentan grandes cantidades de información transaccional. Dado que todos los recursos provienen de diferentes fuentes, se debió construir cada consulta por separado buscando así mantener la integridad de los datos. Tener cada consulta de manera independiente implica una menor densidad de datos a depurar lo que permite observar más fácilmente problemas de duplicidad de datos, ayuda a la detección de datos nulos o atípicos y a otros problemas producto del almacenamiento manual de la información.

Para la empresa como para el modelo es importante centralizar la información ya que facilita el entendimiento del cliente y de sus necesidades, facilita la ingesta de información para las distintas áreas y adicionalmente permite que el ingreso de nuevos clientes sea más rápido y organizado. Existen distintas maneras de consolidar los datos en una tabla, se puede hacer con uniones y concatenaciones en Excel, en SQL Server o mediante ETL'S en Integration Services Provider, por facilidad e integrabilidad con el desarrollo del algoritmo se optó por realizar este proceso en Python creando así un perfilamiento 360 del cliente. La librería Pandas de Python permite un manejo rápido y eficiente de las tablas o data frames (marcos de datos) gracias a la vectorización de los bucles, Python ha cambiado los vectores por matrices lo que hace más eficiente las operaciones con grandes cantidades de datos.

Esta construcción se debe acompañar de un análisis exploratorio de la base a construida que permita tanto comprender mejor los datos como complementarlos con información no explicita. Con ayuda de Tableau (un software de visualización de datos) se exploran las distintas variables, y se extrae de ellas información tal como rangos, valores atípicos y nulos, media, moda, y distribuciones. Con esta información será posible entender mejor como abordar el problema y adicionalmente permitió crear variables nuevas a partir de las existentes que aportan otra perspectiva de los datos al modelo.

Existen múltiples formas de adquirir experiencia de los datos, una forma es encontrar relaciones por medio de un algoritmo de clustering el cual permite analizar datos sin etiquetas, esta estrategia segmenta los datos en grupos que

presentan ciertas similitudes, pero el costo de implementación radica en la correcta selección de las variables implicando así un análisis exploratorio previo. El análisis exploratorio se realizó en conjunto con expertos del área y permitió encontrar una segmentación natural de los clientes basada en la distribución de sus variables. Finalmente se determinó que la información obtenida era suficiente para caracterizar el perfil del cliente y por tanto no se optó por implementar el algoritmo de clustering.

Como se mencionó antes, en la data existen problemas de distintos indoles que se deben afrontar como son los datos nulos, rangos demasiado amplios, o variables categóricas las cuales el modelo no es capaz de procesar. A continuación, se muestran las estrategias que se usaron para pulir los datos de entrenamiento.

5.2 Imputación:

La imputación es el procedimiento mediante el cual se asignan valores a los datos nulos en base a las características, distribuciones y tendencias de las demás observaciones [6], esto es importante porque los datos nulos pueden crear grandes sesgos en las observaciones. La imputación en el modelo se pudo lograr gracias al conocimiento de los agrupamientos naturales de la data (descubiertos en el análisis exploratorio), para esto se usó la función groupby para formar los grupos y la función agg (agregación) para realizar cálculos sobre los grupos tales como sum (suma), mean (media), median (mediana) entre otras. Dado que el objetivo de la imputación es poder conservar las observaciones a costo de crear estimaciones, para minimizar el impacto de las estimaciones en el patrón de los datos se crea una variable adicional que permite al modelo identificar si hubo imputación, así el modelo considerará las imputaciones antes de determinar el peso de una observación.

5.3 Codificación:

Las bases de datos usualmente contienen variables categóricas pero los algoritmos de Machine Learning solamente son capaces de interpretar valores numéricos, para que el modelo sea capaz de interpretar estos datos, es esencial codificar las variables. Para realizar una buena codificación que ayude a mejorar el desempeño del modelo, se debe entender qué clase de data se está manejando para poder utilizar el codificador más adecuado a esta [7].

Dos de los principales tipos de data son:

nominal (grupos sin ningún orden)

ordinal (grupos con un orden definido y discreto).

Para codificar data ordinal se utilizan codificadores ordinales los cuales asignan un valor continuo a cada cadena, para codificar data nominal existen múltiples estrategias como one hot encoding, binary, hashing entre otras. Para la base de datos centralizado no hubo data ordinal que codificar, la estrategia para codificar la data nominal fue la siguiente:

La codificación se realizó con el método de One Hot Encoding que consiste en separar cada categoría como una característica para las observaciones (con valores entre cero y uno), se decidió utilizar este método ya que las categorías no eran tan grandes (no superaban las 50 categorías por característica) lo que garantizaba la eficiencia del One Hot Encoding.

Todo el procesamiento anteriormente mencionado es necesario para poder comenzar la etapa de modelamiento, se seleccionan las técnicas a implementar que puedan dar respuesta al problema, las más usadas para problemas de clasificación son Random Forest, Support Vector Machine (SVM), XGBoost y Decision Tree, dado que las máquinas de soporte vectorial son muy lentas para problemas con grandes cantidades de observaciones se opta por utilizar las otras técnicas mencionadas.

Para identificar cuál técnica tiene el mejor desempeño se necesita un contexto o punto de comparación, por esto se debe implementar un modelo simple, que es básicamente un modelo que no aprende mucho de la data, la funcionalidad de este radica en que su predicción puede ser uniformemente aleatoria o predecir el valor más frecuente que ha visto en la data de entrenamiento. Al medir la exactitud del modelo simple se encuentra que la precisión es bastante alta, esto se debe a que la data está desbalanceada en una escala de 150:1 aproximadamente, por esta razón se debe descartar esta métrica dado que no permite identificar los errores o problemas que presenta el modelo.

En consecuencia, se debe elegir una métrica más acorde al tipo de problema y data que se posee, para realizar esta selección se debe usar la matriz de confusión la cual funciona como herramienta de diagnóstico, la matriz de confusión permite conocer la forma en que fueron clasificados los datos con cuatro categorías, verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. En este caso nos centramos en minimizar los casos de abandono, se considera un Verdadero positivo a la correcta clasificación de

un cliente con propensión de abandono por otro lado el falso negativo es aquel cliente que siendo propenso no pudo ser identificado. Ya que el principal objetivo del modelo es predecir y prevenir el abandono, se toma como métrica de optimización el recall, este permite conocer la proporción de datos relevantes seleccionados, en otras palabras, el recall mide la proporción de clientes propensos al Churn que fueron detectados.

6. Resultados

El desarrollo del proyecto se implementó en Python por su alta compatibilidad con distintas herramientas de análisis de datos, y por su simpleza, robustez y velocidad para el tratamiento de datos. Las librerías utilizadas en este proyecto son:

Base de Datos Centralizada	
Pyodbc	Esta librería permite un acceso simple a bases de datos de SQL.
Pandas	Esta librería permite en manejo rápido y eficiente de estructuras de datos
Numpy	Está librería facilita el trabajo con arreglos numéricos.
XGBoost	Esta librería permite implementar el modelo de clasificación XGBoost
Implementación del modelo a partir de Scikit-Learn	
La librería scikit-learn es una de las librerías de Python más utilizadas, la cual permite la implementación de una gran variedad de algoritmos supervisados y no supervisados. Para la elaboración de este proyecto se aprovechó la gran gama de paquetes que scikit-learn provee, que facilitan significativamente la transformación de los datos y la implementación de múltiples técnicas de aprendizaje.	
from sklearn.model_selection import train_test_split	Esta función permite particionar la data de entrenamiento en prueba y entrenamiento.
from sklearn.preprocessing import StandardScaler	Esta función permite estandarizar un conjunto de datos de manera que su media sea 0 y su varianza 1.

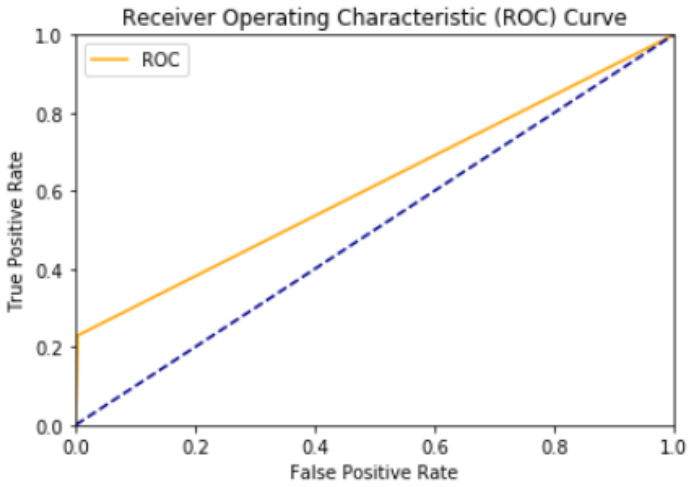
from sklearn.ensemble import RandomForestClassifier	Esta función permite implementar la técnica de clasificación Random Forest
from sklearn.metrics import [roc_curve, roc_auc_score, precision_score, recall_score]	El módulo de métricas permite obtener distintas calificaciones de desempeño
from sklearn.tree import DecisionTreeClassifier	Esta función permite implementar la técnica de clasificación árbol de decisión

Tabla 1. Librerías usadas en la implementación del modelo.

La primera técnica implementada fue el árbol de decisión ya que es la más simple de las técnicas seleccionadas, esta técnica es buena para problemas de clasificación binarios con data desbalanceada. El resultado que se obtuvo se puede observar en la tabla 2.

Métrica	**Tasa**
Recall	0.23
Presicion	0.18
AUC	0.61

Tabla 2. Desempeño Modelo basado en la técnica Decision Tree (simple).



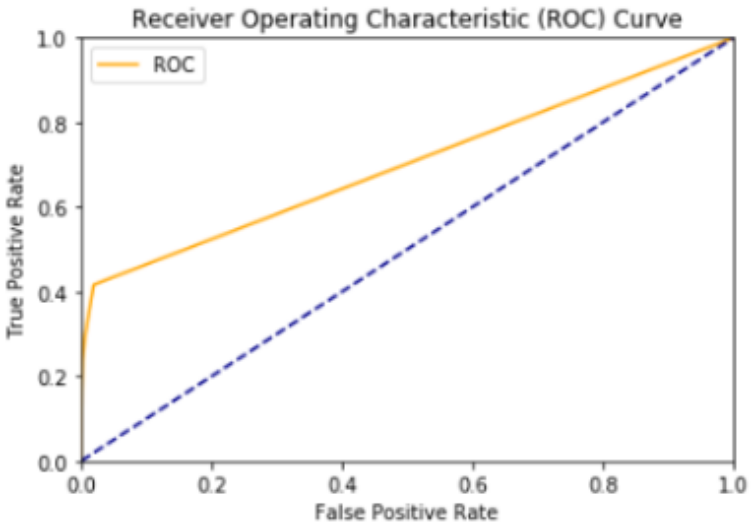
Gráfica 1. Curva ROC Modelo basado en la técnica Decision Tree (simple).

No es posible afirmar si estos resultados son malos o buenos ya que no existe otro modelo con el cual comparar, sin embargo, si se considera que anteriormente no existía un método para predecir el abandono se concluye que se obtuvo una ganancia significativa para el negocio.

En segundo lugar, se implementó y evaluó el desempeño de la técnica Random Forest, el resultado obtenido se puede observar en la tabla 3.

Métrica	**Tasa**
Recall	0.07
Precision	0.54
AUC	0.70

Tabla 3. Desempeño Modelo basado en la técnica Random Forest (simple).



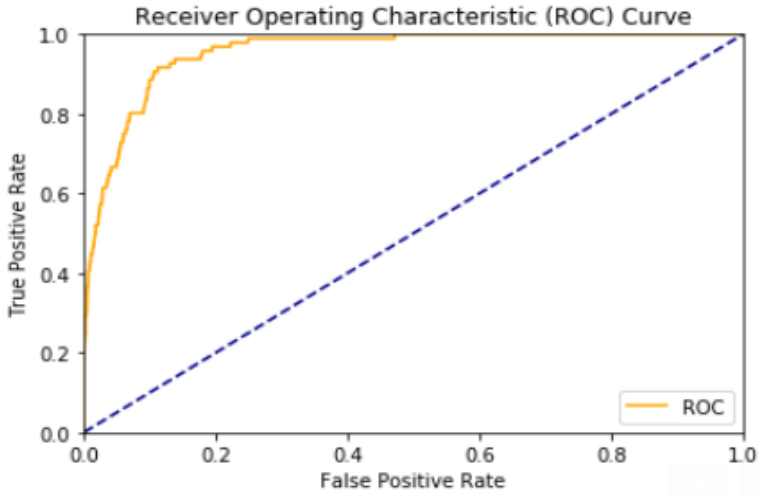
Gráfica 2. Curva ROC Modelo basado en la técnica Random Forest (simple).

El segundo modelo incrementó la precisión en 2 pp pero disminuyó el recall en 0.7 pp. Como se desea obtener el modelo que máximiza el recall y con una buena relación de predicción (curva ROC), se descarta este modelo.

Finalmente se implementó la técnica XGBoost la cual es altamente utilizado en la actualidad por su versatilidad, adaptabilidad y velocidad, el resultado obtenido se puede observar en la tabla 4.

Métrica	**Tasa**
Recall	0.11
Presicion	0.65
AUC	0.96

Tabla 4. Desempeño Modelo basado en la técnica XGboost (simple).



Gráfica 3. Curva ROC Modelo basado en la técnica XGBoost (simple).

Respecto al primer modelo, XGBoost incrementó la precisión en 2.6 pp y disminuyó el recall en 0.5 pp. XGBoost ofrece un hiperparámetro que permite un mejor acercamiento a problemas de data desbalanceada ya que reconfigura el peso de la clase minoritaria, y como el abandono tiene una baja presencia en el data set se decidió repetir la prueba con estos ajustes.

La primera estrategia que se implementó se llama oversampling. El oversampling multiplica los datos de la clase minoritaria (de manera aleatoria) para que el modelo tenga una proporción de 1's mayor sobre los cuales aprender [8], la proporción final fue de 10:1 (cabe aclarar que el oversampling se realiza sobre la data de entrenamiento y nunca sobre el set de prueba).

El ajuste de los hiperparámetros es relativamente simple, basta con ir haciendo un barrido de pequeños saltos en cada uno e ir comparando su desempeño hasta que se encuentre un punto óptimo de costo-beneficio. La configuración resultante del proceso de ajuste se puede observar en la tabla 6.

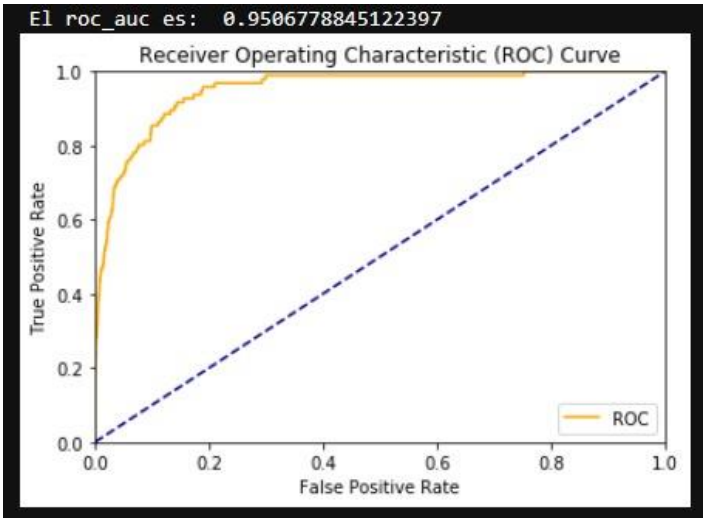
Hiperparámetro	**Valor**
Scale_pos_weight	30
N_estimators	300
Max-depth	2

Tabla 5. Ajuste de Hiperparámetros Modelo basado en la técnica XGboost (optimizado).

El modelo resultante tiene una taza de efectividad de 2:1 en la detección del abandono, esto significa que es capaz de detectar correctamente un 66.6% de los casos de abandono. El desempeño obtenido se puede observar en la tabla 6.

Métrica	**Tasa**
Recall	0.66
Presicion	0.05
AUC	0.95

Tabla 6. Desempeño Modelo basado en la técnica XGboost (optimizado).



Gráfica 4. Curva ROC Modelo basado en la técnica XGboost (optimizado).

La nueva prueba mejora el recall en 1.9 pp respecto al modelo simple y la curva ROC mejora en 0.58 pp. Dado que esta nueva técnica mejora tanto el recall como el AUC se seleccionada esta implementación como el mejor modelo. La curva ROC de la gráfica 4 evidencia la gran capacidad del modelo para detectar los abandonos con una baja tasa de error.

El proyecto finaliza con el despliegue y entrega del modelo de Machine Learning desarrollado. El despliegue está dado por 3 secciones que son:

Automatización:

Se parametrizó y automatizó la carga de todos los insumos que alimentan la base de datos centralizada, a fin de hacer replicable y sostenible el modelo a través del tiempo. La automatización se hizo en Python con la creación de scripts cuya entrada son parámetros de tiempo y que se encargan de manera automática de la creación tanto de la base de datos como del modelo.

Leads:

Dado que el modelo detecta la propensión de los clientes hacia el abandono, parte de la entrega consiste en el exporte del top 1 de clientes de propensión, este insumo servirá de punto de partida a otro tipo de modelos de retención que están por fuera del alcance de este proyecto.

Documentación:

Dado que el resultado final es un insumo de la compañía, todo proceso debió ser debidamente documentando para garantizar el seguimiento, mantenimiento y correcto aprovechamiento del recurso generado. En esta documentación se incluye la metadata de la base de datos centralizada, bases de datos utilizadas y archivos adicionales requeridos con su respectiva fecha de actualización y área encargada, documentación de los scripts, declaración de parámetros de entrada y un paso a paso para la ejecución.

7. Conclusiones

En este trabajo se presentó el procedimiento necesario para implementar un algoritmo de machine learning capaz de predecir de manera proactiva el abandono de clientes con el fin abordar una problemática que significa grandes pérdidas tanto en capital como en esfuerzo para las compañías de telecomunicaciones, los modelos predictivos maximizan la efectividad de las campañas de retención y a su vez permiten un enfoque más centrado en el cliente y sus necesidades.

A la hora de afrontar un problema de Machine Learning es necesario tener claro que este no ofrece soluciones perfectas, no siempre es posible implementar un algortimo de Machine Learning y que el tiempo y costo de implementación puede ser muy alto si no se tiene claro el alcance del proyecto. Para que la implementación sea exitosa es preferible definir sprints ya que estos ayudan a tener claras las etapas del desarrollo, en este sentido el modelo CRISP-DM es altamente recomendable de utilizar ya que segmenta de manera concisa los pasos necesarios para un proyecto de este tipo.

En el mundo del Machine Learning no existen fórmulas mágicas ni guías que determinen la mejor solución a un problema, en este sentido se puede afirmar que gran parte de la complejidad radica en la toma de decisiones que estén debidamente sustentadas en el conocimiento del área, del problema y de los resultados obtenidos.

Una de las partes más importantes en todo proyecto de minería de datos es precisamente la recolección, transformación y exploración de los datos, esta etapa puede parecer transparente ya que no hace parte del despliegue del modelo, sin embargo, se tiene una relación de esfuerzo de 70 a 30 respecto a la etapa de modelamiento y de ella depende en gran parte el desempeño del modelo. Un modelo implementado con una data escasa o de mala calidad dificultará el tuneo

del modelo, adicionalmente podría ocasionar overfitting/underfitting o sesgos en la salida, teniendo como consecuencia predicciones poco confiables.

Definir la población objetivo exige ir más allá de definir la variable dependiente en el set de datos, implica conocer el contexto del negocio, conocer los datos, las necesidades del público y la periodicidad de actualización, un público bien definido acota el problema y permite enfocar los esfuerzos hacia ese objetivo, adicionalmente ayuda a seleccionar la métrica a optimizar que maximice el desempeño, como resultado se obtendrá un modelo más ajustado y confiable. En este proyecto con base en el conocimiento del negocio y de los datos se definió un modelo cuyo objetivo es detectar de manera proactiva el abandono al mes siguiente.

El desempeño de un modelo va a depender de múltiples cosas como lo son el tipo de problema, la técnica usada, la métrica, los hiperparámetros y hasta el data set, por lo tanto, no es factible reducir el modelamiento a una única estrategia. Para obtener un modelo con buen desempeño primero se debió definir que se considera bueno, para solucionar este interrogante se creó un modelo simple cuyo desempeño sirvió de punto de partida, las siguientes implementaciones buscaron obtener el modelo que maximizaba la detección del abandono respecto al modelo simple, finalmente se seleccionó la técnica XGBoost gracias a su alta adaptabilidad, a su robustez y a su capacidad de trabajar con datos desbalanceados (esta afirmación está sustentada en la evaluación de las métricas de desempeño de la sección anterior).

8. Referencias Bibliográficas

1. ¿Qué es exactamente Machine Learning?. Articulo electrónico. Viviana Márquez. Oct 29, 2018. URL: https://medium.com/latinxinai/qu%C3%A9-es-exactamente-machine-learning-77441201a65b

2. Supervised and Unsupervised Machine Learning Algorithms. Articulo electrónico. Jason Brownlee. Mar 16, 2016. URL: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

4. Performance Metrics for Classification problems in Machine Learning. Articulo electrónico. Mohammed Sunarsa. Nov 11, 2017. URL: https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b

3. Cutting the Cord: Predicting Customer Churn for a Telecom Company. Articulo electrónico. Brenner Heintz. Nov 2, 2018. URL: https://towardsdatascience.com/cutting-the-cord-predicting-customer-churn-for-a-telecom-company-268e65f177a5

5. Caballero Muñoz Reja, Ismael & Gómez Carretero, Ana Isabel & Gualo Cejudo, Fernando & Merino García, Jorge & Rivas García, Bibiano & Piattini Velthuis, Mario Gerardo. Calidad de datos. Ediciones de la U. Bogotá. Ra-Ma Editorial, 2019. ISBN 9789587920048

6. ¿Qué hacer cuando tenemos valores faltantes? Artículo electrónico. Jun 30, 2017. Federico Zomeño Breitenstein. URL: https://conocemachinelearning.wordpress.com/tag/imputacion/

7. Smarter Ways to Encode Categorical Data for Machine Learning. Exploring Category. Articulo electrónico. Sep 11, 2018. Jeff Hale.URL: Encoders.https://towardsdatascience.com/smarter-ways-to-encode-categorical-data-for-machine-learning-part-1-of-3-6dca2f71b159

8. Dealing with Imbalanced Data. A guide to effectively handling imbalanced datasets in Python. Articulo electrónico. Tara Boyle. Feb 4, 2019. URL: https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18