ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

PROYECTO DE TITULACIÓN

Previo a la obtención del Título de:

Magister en Ciencia de Datos

TEMA:

SISTEMA DE RECOMENDACIÓN INTELIGENTE PARA EL MANTENIMIENTOA LA INFRAESTRUCTURA DE LA RED DE TELECOMUNICACIONES

Presentado por:

MARCO ANTONIO CAZAR IBARRA

GUAYAQUIL – ECUADOR

Año: 2023

RESUMEN

Hoy tecnologías de la información y la comunicación son el día a día, impulsando por la pandemia se aceleró la transformación digital en todo planeta. Ahora bien, así como esto ha impulsado el crecimiento del sector también ha generado retos, entre ellos mantener e incrementar la base de clientes, en este sector la tasa de rotación es bastante alta dependiendo del tipo servicio, ubicación, tecnología puede estar entre el 20 al 40% según (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022), convirtiéndolo en un factor crítico a ser tomado en cuenta.

La empresa “Xnet” tiene definidos planes enfocados en la retención del cliente, pero todos los esfuerzos siempre han estado enfocados en aspectos de mercadeo, con promociones, aumento de anchos de banda, servicios adicionales, etc., dejando por fuera temas técnicos.

Actualmente se tiene un grupo de personas del área de operaciones enfocadas en el mantenimiento de la infraestructura de la red de telecomunicaciones, estos esfuerzos son reactivos a la queja del cliente o la experiencia de los expertos que planifican los mantenimientos,aquí se ha pensado porque no unir estos esfuerzos al propósito de retención del cliente, qué talsi proactivamente se realizan estos mantenimientos en lugares que se sabe que los clientestienen alta probabilidad de cancelar sus servicios, si todo funciona como debe o se nota mejorascumpliendo con lo que el usuario espera, la tentación de buscar un nuevo proveedor disminuirá.

Por ello se plantea que utilizando la data que posee la empresa, se genere un algoritmo que permita predecir que clientes planean cancelar sus servicios, sobre ellos se extrae información de donde se encuentran ubicados físicamente, de que infraestructura dependen, que tecnología utilizan, etc. Con toda esta data se entrega un tablero interactivo al personal de operaciones que le sirve de recomendaciones para armar la planificación de sus mantenimientos,trabajo que, si o si se debe efectuar, pero esta vez en base a datos y apoyando el esfuerzo de retención de clientes

Este dashboard no solo presentará del detalle de la infraestructura que se sugiere ser parte del plan, sino que muestra detalles del porque se ha seleccionado dicha infraestructura, como, por ejemplo, cantidad de tickets, niveles de potencia, ubicación, etc.

ABSTRACT

Today information and communication technologies are everyday life, driven by the pandemic, digital transformation accelerated throughout the planet. Now, just as this has driven the growth of the sector, it has also generated challenges, including maintaining and increasing the customer base. In this sector, the turnover rate is quite high depending on the type of service, location, technology, it can be between 20 40% according to (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022), making it a critical factor to be taken into account.

The company “Xnet” has defined plans focused on customer retention, but all efforts have always been focused on marketing aspects, with promotions, increased bandwidth, additional services, etc., leaving out technical issues.

Currently there is a group of people from the operations area focused on maintaining the infrastructure of the telecommunications network. These efforts are reactive to the customer's complaint or the experience of the experts who plan the maintenance. Here we have thought about why not join these efforts to the purpose of customer retention, what if these maintenances are proactively carried out in places where it is known that customers have a high probability of canceling their services, if everything works as it should or improvements are noted, complying with what the user expects , the temptation to look for a new supplier will decrease.

For this reason, it is proposed that using the data that the company has, an algorithm is generated that allows predicting which clients plan to cancel their services, information about them is extracted from where they are physically located, what infrastructure they depend on, what technology they use, etc. With all this data, an interactive dashboard is delivered to the operations staff that serves as recommendations to plan their maintenance, work that must be carried out, but this time based on data and supporting the retention effort. customers

This dashboard will not only present details of the infrastructure that is suggested to be part of the plan, but also shows details of why said infrastructure has been selected, such as, for example, number of tickets, power levels, location, etc.

DEDICATORIA

A mis padres
Marco y Gladys

AGRADECIMIENTOS

A todos quienes apoyaron o contribuyeroncon
el desarrollo del presente trabajo

En especial a mi familia que ha sido un
apoyo fundamental

DECLARACION EXPRESA

“Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Yo Marco Antonio Cazar Ibarra doy consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Marco Antonio Cazar Ibarra

Autor

COMITÉ EVALUADOR

Dr. Christian Tutiven

PROFESOR TUTOR

Dr. Sergio Bauz

PROFESOR EVALUADOR

ABREVIATURAS

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

CAPÍTULO 1

1. PLANTEAMIENTO DE LA PROBLEMÁTICA

1.1. Descripción del problema

En el mundo actual se tiene una enorme cantidad de información que sirve para análisis como, por ejemplo:

- En medicina se puede utilizar modelos que trabajan con imágenes para extraer de ellas características que permitan identificar si un paciente presenta una determinada condición.
- En finanzas los datos pueden ayudar a detectar fraudes, o si una persona es elegible para un crédito.
- En seguridad cibernética se tiene modelos de IA que ayudan con el filtrado de datos en tiempo real para detectar amenazas.
- En la industria los datos proporcionados por los equipos se utilizan para generar modelos para mantenimiento predictivo.
- En el transporte los datos proporcionan información valiosa, para determinar causas frecuentes de accidentes, rutas de transporte que deben aumentar su frecuencia de funcionamiento, etc.
- En empresas de venta minorista se utiliza para predecir el stock, posicionar productos, personalizar ofertas.
- En el sector turístico la particularización de paquetes en base a consumos y preferencias, campañas publicitarias por segmento, promociones para fidelizar clientes, etc.
- En las empresas de telecomunicaciones, que generan inmensos flujos de dato con patrones de comportamiento del cliente, tendencias de consumo, horarios de uso, preferencia, quejas, etc. Se utilizan estos datos para segmentación de clientes, promociones focalizadas, optimización de precios, pronosticar la demanda, tasa de abandono, etc.

En el entorno globalizado en el que todos estamos conectados, viviendo la virtualidad de alguna manera como en los negocios, educación, salud, entretenimiento, etc.es imprescindible contar con empresas que briden la conectividad con los estándares que los usuarios requieren, es por ello por lo que el gobierno ecuatoriano en el 2019, lanzo el plan Ecuador digital cuyo objetivo era proporcionar acceso a estos servicios para el 98% de la población del país, tomando como pilares la conexión, la seguridad cibernética y la innovación (MINTEL, 2020).

Parte de esto consistía en habilitar puntos Wi-Fi gratuitos a lo largo del territorio nacional, como un primer paso para que la población esté conectada al mundo, pero por efecto de la pandemia este esfuerzo no se ha logrado concretar y aún se está trabajando en ello con el aporte de las empresas privadas.

La pandemia tuvo un enorme impacto en la sociedad en general obligándonos a dar grandes pasos hacia la transformación digital, esto impacto directamente a las empresas de telecomunicaciones con un gran incremento de nuevos suscriptores de banda ancha, debido a la nueva normalidad en donde lo virtual es parte de día a día y el Internet se convirtió en recurso indispensable para cualquier persona, según el índice de gobierno electrónico, los servicios en línea y telecomunicaciones pasaron de 72.9% y 36.9% a 82.9% y 44.8% respectivamente para el 2021 (MINTEL], 2020.).

Actualmente en el Ecuador tenemos 10.17 millones de usuarios de internet, el volumen de comercio electrónico alcanzó 2.3 mil millones, lo que supone un crecimiento de 700 millones con respecto al 2019 lo que significa un incremento del 43.75% (MINTEL, 2020).

Bajo este nuevo escenario, el usuario realmente verifica que es lo que está pagando y ahora es más exigente, no se tolera lentitud en la conexión, ni caídas en medio de una reunión o clase virtual, peor aún falta de respuesta del proveedor en los tiempos estipulados o falta de opciones en el servicio que requiere. Por lo que sin mayor problema toma la decisión de migrar a otro proveedor que cumpla con sus necesidades.

Este escenario ha complicado a los proveedores de telecomunicaciones que deben alcanzar la satisfacción del cliente, aquí un aspecto fundamental para mantener a un usuario contento con el servicio, es una infraestructura funcional, tolerante a fallos que soporte la enorme carga de tráfico que se maneja hoy en día, eso ha sido un verdadero problema, empresas cuyo servicio aparentemente funcionaban bien, ahora tienen quejas de sus clientes por inestabilidad en el servicio, eso por el aumento de clientes en sus redes, lo que nos da la noción de tecnologías obsoletas, poco o nulo dimensionamiento de recurso y falta de mantenimiento.

La empresa "Xnet" que es líder en el mercado ecuatoriano con más de dos décadas de funcionamiento, armó una estructura organizacional que se encargue de darle solución a estos problemas, utilizando expertos técnicos que dan los lineamientos que se debe seguir en base a su propia experiencia, generalmente reactivos y enfocados reportes de problemas técnicos escalados por el cliente, generan planes de mantenimiento para que la red funcione de la maneraesperada.

1.2. Justificación

Dado el problema descrito, si la empresa "Xnet" únicamente se centran en ser reactivos en base a la experiencia de las personas cuyo enfoque principal es guiarse por los tickets de soporte, para determinar los lugares en los que se debe ejecutarse planes de mantenimiento y mejora de la infraestructura, confiando que con ello el servicio sea lo que el cliente espera, lo más probable es que con esta forma de enfrentar el problema en el corto plazo no se consiga los resultados deseados, que son la permanencia y fidelidad de sus clientes.

Para ello es necesario poder predecir quienes son los clientes con mayor probabilidad de abandonar la empresa y sobre ellos tomar las medidas necesarias para retenerlos, en este punto el enfoque actual es netamente de marketing con una perspectiva de precio y promociones (cambios de ancho de banda o servicios adicionales sin incrementar valores) lo cual es correcto, pero se está dejando de lado el componente técnico del servicio, por lo que se propone tomar la iniciativa y trabajar de manera proactiva tomando decisiones basadas en datos y no solo en la experiencia de las personas, teniendo como prioridad aquellos clientes que tienen la intención

de cancelar el servicio por problemas propios de la infraestructura, dotando a los encargados de una herramienta visual que les permita planificar el mantenimiento de la mejor forma que aporte con este objetivo empresarial.

1.3. Objetivos

1.3.1. Objetivo general

Mostrar de forma automática recomendaciones sobre el mantenimiento a la infraestructura de la red de telecomunicaciones, identificando las zonas en los cuales los clientes tienen una alta probabilidad de deserción.

1.3.2. Objetivos específicos

- Unificar los sets de datos de las diferentes fuentes de información en un único almacén de datos.
- Determinar las características relevantes, tomando en cuenta los clientes que han cancelado sus servicios.
- Definir la mejor técnica para predecir la probabilidad que un cliente cancele sus servicios.
- Desarrollar un modelo de predicción de deserción por mantenimiento que tenga una precisión mayor del 90%.
- Generar un cuadro de mando que muestre los clientes con probabilidad de deserción y la infraestructura de la que dependen, siendo una sugerencia las zonas en las que debería ejecutar los planes de mantenimiento.

1.4. Metodología

Como metodología se utilizará algoritmos de machine learning y Deep learning, variables demográficas, variables técnicas del servicio, detalles del comportamiento de pago del cliente e información del sistema transaccional.

Partiremos del set de datos de servicio, soporte, potencia y cancelaciones que contienen la información de los clientes, por cada uno de los módulos tenemos varios archivos, por lo que el paso inicial será concatenar los archivos por modulo y luego unimos todos los archivos en un solo set de datos final con el que se creará nuestro almacén de datos

Como siguiente paso se realizará un análisis exploratorio de los datos del set de datos final, identificando valores nulos, valores faltantes e información relevante que nos ayude a entender el dataset final.

Sobre este dataset limpio se ejecutarán técnicas de machine learning y deep learning que permitan identificar los clientes que tienen la intención de cancelar su servicio, entre estas técnicas tenemos:

- Aprendizaje supervisado.
- Aprendizaje semisupervisado.
- Aprendizaje no supervisado.
- Detección de anomalías.
- Detección de valores atípicos.

Sobre estos clientes ahora debemos identificar a aquellos cuya motivación para cancelar el servicio sea problemas técnicos con el servicio.

Con esta información determinamos la infraestructura de la que dependen dichos clientes y esto se muestra como sugerencia al personal encargado de la planificación de los mantenimientos, en forma de un cuadro de mando interactivo, en la Figura 1 podemos observar lo descrito en los párrafos anteriores y agrupados en cuatro fases que son la recolección de datos, la limpieza de los datos, la aplicación de modelos y finalmente la visualización.

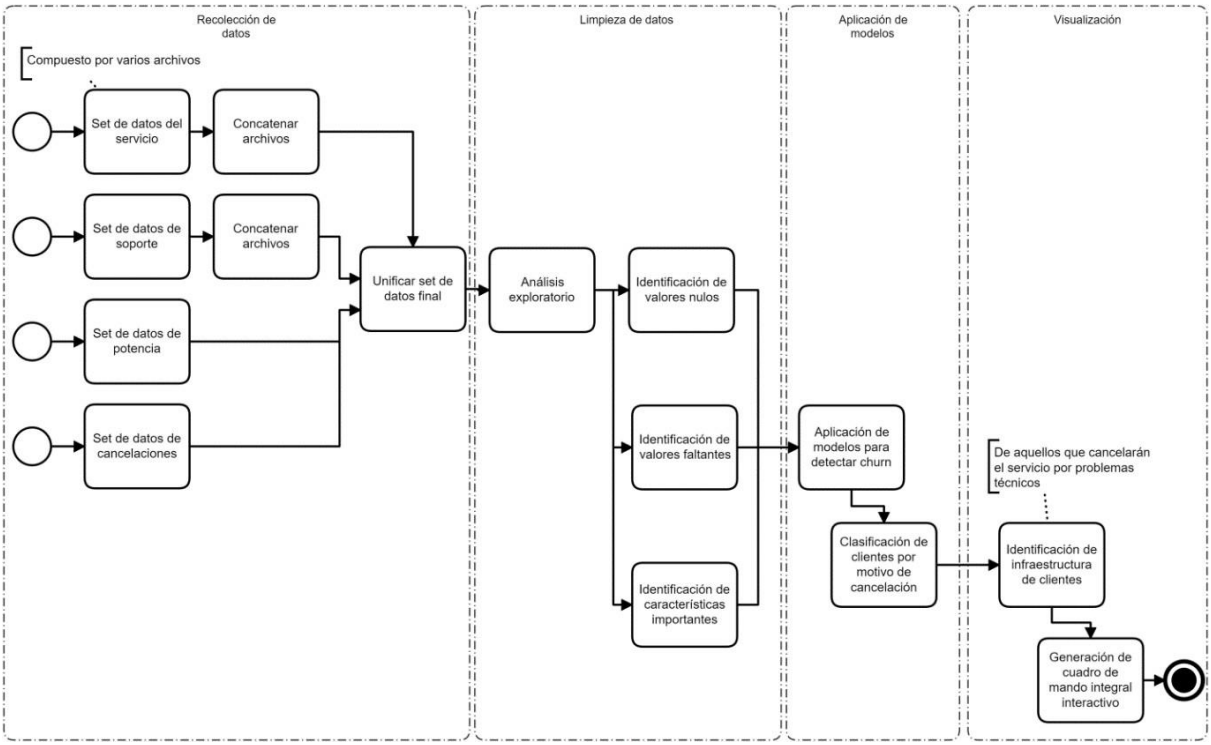


Figura 1: Esquema de las fases del proyecto

1.5. Resultados esperados

Se proporcionará un cuadro de mando que mostrará de forma automática la sugerencia de los sectores en los que se debería ejecutar mantenimientos en base a las predicciones realizadas por el algoritmo. Este cuadro de mando estará disponible para el personal de operaciones encargado de ejecutar esta tarea.

Este aplicativo estará levantado en el host local dentro de la infraestructura interna de una empresa de telecomunicaciones.

1.6. Dataset

La empresa “Xnet” cuenta con su propio sistema ERP desarrollado a la interna mismos que tienen data demográfica del cliente, servicios, facturación, pagos, soporte y parámetros de red, los diferentes módulos del sistema tienen su propio esquema en la base de datos, por lo cual la data no será un solo archivo sino la suma de varios documentos xls con diferente data que nos entregará los recursos que necesitaos para la ejecución del análisis.

A continuación, se presenta el detalle de los diferentes sets de datos que se han obtenidodesde el sistema.

Módulo de servicios: contiene información que nos indica quien es nuestro cliente, que servicio ha contratado, fecha de activación del servicio, cuál es su forma de pago, donde se encuentraubicado y el estado de su servicio.

Campo	Descripción del campo
ID_SERVICIO	Identificador único del servicio del cliente
AÑO	Año de activación del servicio
MES	Mes de activación del servicio
DÍA	Día de activación del servicio
DIAS_ACTV_CANCEL	Antigüedad del cliente
REGIÓN	División geográfica del Ecuador definida por la empresa
OFICINA	Oficina de venta de la que depende el cliente
JURISDICCION	Oficina técnica de la que depende el cliente
CLIENTE	Nombre del cliente
LOGIN	Identificad único del punto geográfico del cliente
PLAN	Nombre del plan contratado por el cliente
ES_VENTA	Bandera que indica si realmente es una venta o una cortesía
PRECIO_VENTA	Valor que se le factura mensualmente al cliente
PORCENTAJE_DESCUENTO	Porcentaje promocional que se descuenta del valor total de la factura (cuandoaplique)
VALOR_DESCUENTO	Valor promocional que se descuenta del valor total de la factura (cuando aplique)
FECUENCIA_PRODUCTO	Indica si el plan se factura mensual, trimestral, semestral o anual
USR_VENDEDOR	Asesor comercial que concreto la venta del plan al cliente
TIPO_NEGOCIO	Identifica si es un cliente residencial, una profesional o una pequeña/ mediana empresa
FORMA_PAGO	Indica la forma de pago del cliente
BANCO	Indica el banco del cliente (cuando aplique)
CUENTA	Indica el tipo de cuenta (cuando aplique)
PROVINCIA	Provincia en la que se encuentra el punto geográfico del cliente
CANTON	Cantón en el que se encuentra el punto geográfico del cliente
PARROQUIA	Parroquia en la que se encuentra el punto geográfico del cliente
SECTOR	División geográfica de la parroquia definida por la empresa
CONECTOR	Splitter del que se conecta el cliente (multiplexor)
CONTENEDOR	Elemento pasivo que contiene el Splitter
EQUIPO_CORE	Equipo core del que depende el servicio del cliente
INTERFACE_CORE	Puerto del equipo core del cual depende el cliente
ESTADO_SERVICIO	Estado del servicio del cliente
LONGITUD	Longitud del punto del cliente
LATITUD	Latitud del punto del cliente
INTERFACE_CPE	Puerto del equipo terminal del cliente
GRUPO	Corresponde al lote de facturas que se generan en diferentes instantes del mes
FACTURA_INSTALACION	Campo que indica si al cliente se le cobra o no la instalación del servicio
VALOR_INSTALACION	Valor que se le debe facturar al cliente por la instalación del servicio
DISTANCIA_PROMEDIO_UM	Recorrido en metros de las el contenedor hasta el punto físico del cliente
TECNOLOGIA	Marca de los equipos con los que se entrega el servicio
CANAL	Medio por el cual el cliente contrato el servicio
COMPORTAMIENTO_PAGO	Categoría que indica que tan cumplido es el cliente con relación al pago de susfacturas
SCORE	Categoría que califica al cliente en base a su comportamiento de pagos, quejas, requerimientos y demás interacciones con la empresa

Tabla 1: Módulo de servicios

Módulo de soporte: Contiene información de los soportes reportados por los clientes, indicado tiempo que tomó la solución, tiempo que dependió del cliente.

Campo	Descripción del campo
ID_TICKET	Identificador único del ticket
NUMERO_TICKET	Número de ticket conformado por año, mes, día y un contador
TIPO_TICKET	Identifica si es individual, masivo o reportado por el ente de control
LOGIN	Identificad único del punto geográfico del cliente
NUMERO_CONTRATO	Secuencial por oficina de facturación
FE_APERTURA	Fecha en la que se abre el ticket
FE_CIERRE	Fecha en la que se cierra el ticket
ESTADO_TICKET	Estado del ticket
MODELO_ELEMENTO_CLIENTE	Modelo del equipo en el sitio del cliente
TIEMPO_CLIENTE	Tiempo del caso que le corresponde al cliente
TIEMPO_EMPRESA	Tiempo del caso que le corresponde a la empresa
TIEMPO_SOLUCION	Tiempo que tomo la solución
TIEMPO_TOTAL	Tiempo total del ticket

Tabla 2: Modulo soporte

Nivel de potencia: Modulo que muestra el nivel de potencia del enlace del cliente.

Campo	Descripción del campo
DBM	Intensidad con la que llega la señal al equipo terminal en el sitio del cliente
LOGIN_PUNTO	Identificad único del punto geográfico del cliente

Tabla 3: Nivel de potencia

Motivo de cancelación: Modulo que muestra el motivo por el cual el cliente ha cancelado el servicio.

Campo	Descripción del campo
LOGIN	Identificad único del punto geográfico del cliente
MOTIVO_CANCELACION	Motivo por el cual el cliente cancelo el servicio

Tabla 4: Modulo de cancelación

CAPÍTULO 2

2. MARCO TEÓRICO Y ESTADO DEL ARTE

2.1. Fundamentos del problema

2.1.1. Churn

(Peiró, s.f.) nos dice que el churn es una métrica que indica la cantidad de clientes quehan dejado de consumir los productos y/o servicios que la empresa ofrece.

Se debe considerar que para que toda empresa crezca o se mantenga en el tiempo nosolo debe adquirir clientes nuevos todos los días, sino que deben mantener una buena relacióncon los actuales. Y por ello necesitan conocer la cantidad de clientes que han decidido cancelaro abandonar la empresa, este es un dato valioso para cualquier organización (Peiró, s.f.). (Peiró, s.f.) menciona que la fórmula para calcular esta tasa es:

$$Churn = \frac{cantidad\ de\ clientes\ que\ cancelaron\ el\ servicio\ en\ el\ mes}{cantidad\ de\ clientes\ al\ inicio\ del\ mes} x100$$

(Peiró, s.f.) recomienda que, una vez obtenido este valor, ahora el objetivo es mejorarlo y paraello recomienda:

- Entender los motivos por los cuales el cliente tomó la decisión de abandonar la empresa,una buena opción es contactar al cliente e indagar por qué (Peiró, s.f.).
- Se debe conocer a los clientes, comportamiento, preferencia lo que pueda indicarnos quetan susceptible son a dejar la empresa (Peiró, s.f.).
- Armar un equipo que se dedique a este análisis, buscando opciones y propongaalternativas rápidas a la organización (Peiró, s.f.).
- No olvidar escuchar la voz del cliente, si nos dice algo es por una razón y nos ayudara amejorar o entender que estamos por el camino correcto (Peiró, s.f.).

Ahora bien, lo descrito en los párrafos anteriores lo indican de forma general, pero ¿qué pasa con el churn en las empresas de telecomunicaciones?

Según (Xu T, Ma Y, Kim K, 2021) estamos en un escenario de fuerte competencia, y un creciente número de empresas en este sector, por este motivo es extremadamente fácil para el cliente buscar una nueva alternativa que cubra su necesidad, precios competitivos, servicios adicionales, cobertura, estabilidad, etc.

(Xu T, Ma Y, Kim K, 2021) indican que la tasa de abandono en este sector está entre el 20% al 40%, valores bastante altos, además nos proporciona otro dato interesante ya que indica que retener a un cliente existente cuesta entre 5 a 10 veces menos que lo requerido para conseguir uno nuevo (García, D.L., Nebot, À. & Vellido, A., 2017).

Continuando con las estadísticas (Xu T, Ma Y, Kim K, 2021) menciona que predecir quienes son los clientes que abandonarán la empresa es 16 veces menos que lo que cuesta conseguir un nuevo cliente. Conseguir una reducción del 5% en el churn representa beneficios para la organización entre el 25% al 85% (Kotler, P., & Armstrong, G. ,1994), esto nos da una clara idea de la importancia de esta métrica en el sector de las telecomunicaciones.

Como una ventaja de las empresas de telecomunicaciones es que su día a día es la tecnología, por lo que generalmente tienen enormes cantidades de datos dentro de sus sistemas, infraestructura, personal capacitado y con los perfiles para generar modelos de aprendizaje automático, que sirvan de entrada para el equipo de Marketing y los directores definirán los planes de acción para reducir esta tasa (Amin, A., Al-Obeidat, F., Shah, B., Adnan, A., Loo, J., & Anwar, S. ,2019), pudiendo tratarse de promociones, beneficios, mejoras en el servicio (mayor ancho de banda por el mismo precio o un valor mínimo).

Ya que tenemos claro que es el churn y porque de su importancia, es necesario tenernociones sobre el mundo en que se desenvuelven las actividades de la empresa y es por ellopor lo que en la siguiente sección entenderemos como es la infraestructura que utiliza para servir a sus clientes.

2.1.2. Infraestructura de las redes de telecomunicaciones

La empresa "Xnet" en su constante búsqueda para ofrecer un buen servicio al cliente en el año 2010 decide migrar toda su infraestructura a la tecnología de red óptica pasiva gigabit (GPON) con el concepto de fibra hasta el hogar (FTTH).

GPON es una tecnología que salió a la luz a finales de los 90 (Millán, 2007), la industria de las telecomunicaciones la consideró como una buena opción por su característica multipunto. Esto significa ahorros en la instalación de la fibra óptica (FO) y la infraestructura necesaria para proporcionar el servicio a los clientes.

Las tecnologías PON han tenido una evolución en el tiempo que le ha permitido ser considerada practicante un estándar para el servicio de banda ancha para el hogar, a esto ayudo en gran media que tiene una enorme popularidad en Japón, Corea del Sur y Taiwán (Millán, 2007).

A continuación, en la Tabla 5 se muestra el detalla de características de las tecnologías pon como son: BPON, GPON y EPON, en las que podemos apreciar que GPON tiene mejoras características que las otras alcanzando tasa de bits mayores al triple de distancia de la otras dos, con una compartición mayor y con una eficacia sobre el 90%.

Características	BPON	GPON	EPON
Tasa de bits	Down: 1.224, 622, 155 Up: 622, 155	Down: 2.448, 1.244 Up: 2.488, 1.244, 622, 155	Down: 1.224, 622, 155 Up: 622, 155
Codificación de línea	NRZ (+scrambling)	NRZ (+scrambling)	8b/10b
Ratio de división máximo	1:32	1:28 (1:64 real)	1:32
Alcance máximo	20 km	60km	20km
Estándares	Serie ITU-T G.983.x	Serie ITU-T G.984.x	IEEE 802.3ah
Soporte TDM	TDM sobre ATM	TDM nativo, TDM sobre ATM, TDM sobre paquetes	TDM sobre paquetes
Soporte video RF	No	Si	No
Eficacia típica	83% downstream 80% upstream	93% downstream 94% upstream	61% downstream 73% upstream
OAM	PLOAM+OMCI	PLOAM+OMCI	Ethernet OAM (+ SNMP opcional)
Seguridad downstream	Churning o AES	AES	No definida

Tabla 5: Características Xpon (Millán, 2007)

Una red tradicional Ethernet en su estructura tiene tres niveles partiendo del núcleo en el que tenemos los equipos L3 (aquellos que se utilizan para interconectar y rutear entre host que

pueden estar en redes diferentes (Todo de Redes, s.f.)) que se encuentran interconectados proporcionando redundancia y que usan protocolos de enrutamiento dinámico, pasamos al nivel de distribución en el que tenemos equipos L3 y L2(proporcionan conectividad a dispositivos en una misma red (Worton, 2022)), finalmente tenemos el nivel de acceso donde van conectados los equipos terminales del cliente, a diferencia de esto una red GEPON consta únicamente de dos niveles, la terminal de línea óptica (OLT) que es parte de la infraestructura del proveedor y los terminales ópticos de interconexión que son los puntos finales en el cliente (López, 2022).

Como se muestra en la Figura 2 la red GPON desde el OLT se conectan por fibra óptica los splitters de primero y segundo nivel, quienes son los encargados de mutiplexar la señal óptica, de ese modo se conectan 64 clientes (ONTs) en un mismo puerto, tal como lo indica la Tabla 5 y pudiendo estar a distancias de 60km dando una enorme cobertura por cada uno de estos equipos.

Se puede observar los elementos intermedios entre la OLT y el ONT son pasivos (López, 2022), por lo que el costo este tipo de redes es menor si la comparamos con una Ethernet, además simplifica los procesos de manteamiento que se deban efectuar, reduciendo costos para el proveedor y otorgándole la posibilidad de entregar grandes anchos de banda que puedan soportar las aplicaciones y servicios de transmisión bajo demanda que actualmente se consumenen el mercado (Millán, 2007).

Ahora es el momento de entender a que nos referimos al mencionar la palabra mantenimiento, además de conocer cuáles son sus tipos, eso se lo describe en a continuaciónen la sección 2.1.3.

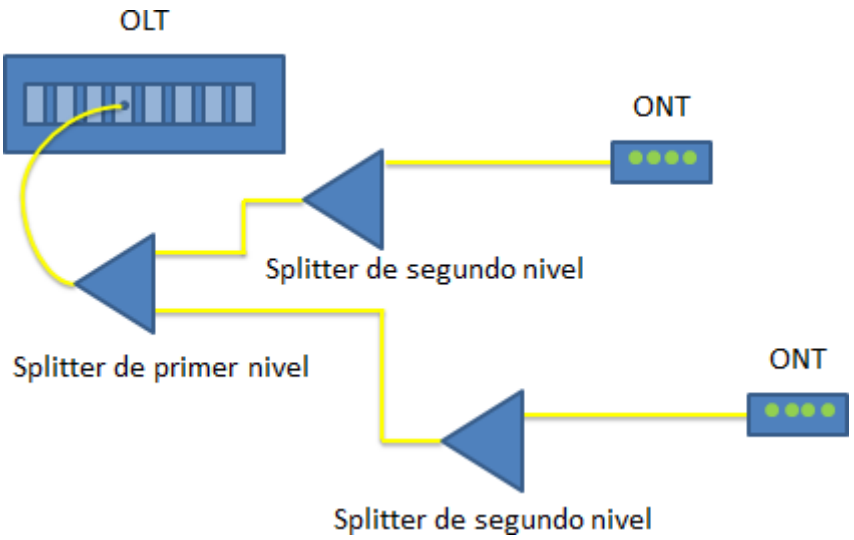


Figura 2: Esquema de funcionamiento de una red GPON

2.1.3. Mantenimiento de la red de telecomunicaciones

(Westreicher, s.f.) define al mantenimiento con un proceso formado por una serie de tareas que se ejecuta sobre algún elemento de la producción para que continúe con su funcionamiento según lo esperado.

Este tema es agnóstico a la industria, esta tarea debe ser realizada para evitar fallos en el proceso productivo que puedan generar gasto no considerados, por ello es necesario monitorear estos elementos en búsqueda de desperfectos que se presentan por el desgaste natural que se genera al paso del tiempo (Westreicher, s.f.).

El mantenimiento puede ser de una de las siguientes categorías principalmente (Dynamox, 2021):

Mantenimiento correctivo

Este es un tipo de mantenimiento reactivo, es decir se lo aplica una vez que equipo presenta la falla y trabaja de forma anómala o deja de funcionar totalmente (Dynamox, 2021).

Este mantenimiento es costoso porque es imprevisto y afecta directamente a la producción (Dynamox, 2021).

- **Ventajas:**
    - No se tiene costos asociados a la planificación y supervisión de su ejecución (Dynamox, 2021).
- **Desventajas:**
    - Tiempo de parada de la producción desconocido (Dynamox, 2021)
    - Averías subyacentes con diferentes niveles de impacto (leve, grave o catastrófica) (Dynamox, 2021).
    - Costos altos de ejecución (Dynamox, 2021)

Mantenimiento preventivo

Este mantenimiento parte de una planificación previa dentro de intervalos definidos en base a experiencia o recomendaciones de la industrio y/o proveedor, tiene como objetivo evitar una falla o la degradación del rendimiento, además de generar ahorros al extender la vida útil de los equipos (Dynamox, 2021).

El plan debe ser elaborado de tal modo que no afecte la producción y se cumpla con los tiempos establecidos.

- **Ventajas**
    - Equipos con vida útil extendida (Dynamox, 2021).
    - Reducción de fallas subyacentes (Dynamox, 2021).
    - Minimizar fallas graves o catastróficas (Dynamox, 2021).
    - Menor impacto en la producción (Dynamox, 2021).
    - Posibilidad de decir cuándo y cómo ejecutarlo (Dynamox, 2021).
    - Evita cantidades innecesarias de repuestos almacenados en la bodega(Dynamox, 2021).
- **Desventajas**
    - Excesivo mantenimiento que termina dañando el equipo (Dynamox, 2021).
    - Aumento de costo por incremento de personas que participan en la planificación, ejecución, validación y monitoreo de la ejecución (Dynamox, 2021).

Mantenimiento predictivo

Aquí se requiere del monitoreo constante del equipo en búsqueda de indicios que puedan alertar de posibles fallas y así poder efectuar el mantenimiento (Dynamox, 2021).

Con esta técnica (Dynamox, 2021) indica que se puede identificar alrededor del 80% de patrones de fallas, el monitoreo genera data que se almacena para su posterior análisis, aquí se puede utilizar inteligencia artificial que identifiquen componentes con estado crítico e indiquen cual puede ser el origen del defecto (Dynamox, 2021).

- **Ventajas**
    - Detecta las posibles fallas con anticipación (Dynamox, 2021).
    - Extiende la vida útil del equipo y sus componentes (Dynamox, 2021).
    - Suprime revisiones innecesarias (Dynamox, 2021).
    - Reduce perdidas por interrupción de la producción (Dynamox, 2021)
- **Desventajas**
    - Se requiere personal calificado para el análisis de los datos

Mantenimiento prescriptivo

Se trata de la evolución natural del mantenimiento predictivo en el marco de la industria 4.0, no solo se analizan las grandes cantidades de datos (big data) para determinar cuándo un equipo fallará, sino que además sugerirá qué medidas se puede ejecutar (Dynamox, 2021).

Aquí no existe una planificación previa, el análisis es ejecutado a distancia en base a parámetros configurados con anterioridad y es el mismo sistema que pose inteligencia artificial quien aprende de los mantenimientos anteriores (preventivos y correctivos) para identificar patrones por cada tipo de falla y decide si se continúa en funcionamiento o se pasa amantenimiento (Dynamox, 2021).

En estos casos el técnico simplemente acude y ejecuta los ajustes que se determinen como resultado del análisis, optimizando tiempos de ejecución (Dynamox, 2021).

- **Ventajas**
    - Extiende la vida útil del equipo y sus componentes (Dynamox, 2021).
    - Reducción de costes al ejecutar solo los mantenimientos que realmente se requieren (Dynamox, 2021).
    - Sugerencia de tareas específicas a ser ejecutadas por el personal técnico(Dynamox, 2021)
- **Desventajas**
    - Alta inversión en equipos y herramientas informáticas (Dynamox, 2021).
    - Personal con capacitado para operar esta tecnología (Dynamox, 2021).
    - Cambio de la cultura organizacional (Dynamox, 2021).

En el caso de la empresa “Xnet” el enfoque se enmarca en mantenimiento correctivo y preventivo en base a buenas prácticas o experiencia del encargado, aun no se ha dado el paso hacia el mantenimiento predictivo y esto es algo que se lo debe realizar en el corto plazo sacando provechó de todos los datos que se posee (big data).

Un concepto muy importante que se menciona es big data, dada su relevancia se le dedico toda una sección, que es la que tenemos en seguida.

2.1.4. Big data

(ORACLE, s.f.) define a big data como un conjunto de datos de gran tamaño que provienen de un sin número de fuentes y que tienen una alta complejidad por lo que es demasiado complicado procesarlos en herramientas tradicionales, big data ofrece la posibilidad de solucionar problemas empresariales que anteriormente no era posible hacerlo.

(Powerdatas.f.) complementa el concepto indicando que, aunque no está plenamente establecido el tamaño para considerar un conjunto de datos como big data, una gran cantidad de profesionales toman como valores referencias por lo menos 30 o 50 Terabytes

Cuando se piensa en biga data se lo relaciona con las tres "V” que son volumen, velocidad y

variedad, pero (Mailjet, s.f.) indica que actualmente existen más “V” en el concepto.

- **Volumen** se refiere a la cantidad de datos que se deberá procesar (Mailjet, s.f.).
- **Velocidad** hace referencia al ritmo con el que se recibe la data, se guarda en disco o se la procesa en memoria (Mailjet, s.f.).
- **Variedad** esto es respecto al tipo de dato, estructurado (bases de datos relacionales), no estructurado (sin un modelo predefinido, como por ejemplo un documento de Word o una conversación) o semiestructurados (sin estructura definida, pero poseen etiquetas que permiten agruparlos jerárquicamente, por ejemplo, un JSON (Aiofthings, s.f.)) (Mailjet, s.f.).
- **Veracidad:** se refiere a si los datos son exactos ya que la mala calidad de los mismos puede afectar el análisis (Mailjet, s.f.).
- **Valor** enfocado en la relevancia e importancia según el caso en que se utilice (Mailjet, s.f.).
- **Variabilidad** es utilizar el mismo conjunto de datos, pero para objetos diversos (Mailjet, s.f.).

El termino big data es reciente, pero los enormes sets de datos ya se los tenía la década de los 60's y 70's, esto es cuando recién salían a luz los primeros centros de datos (ORACLE, s.f.). Ya para el 2005 la industria se da cuenta que en las diferentes plataformas se generan todo tipos de datos y es en esta época se popularizan los No SQL, el desarrollo de Spark y Hadoop fue esenciales para el desarrollo del big data (ORACLE, s.f.).

La data es un activo muy valioso y útil para las empresas, ya que permite dar respuesta a preguntas que incluso la empresa no había considerado, en la Figura 3 vemos una representación de diferentes fuentes de información que entregan los codiciados datos para la toma de decisiones. Con volúmenes tan grandes que requieren de infraestructura especializada para su gestión, afortunadamente al ser esto una tendencia se han reducido costos en tema dealmacenamiento (Mailjet, s.f.).

Figura 3: Representación de la importancia de los datos para las empresas

¿Dónde se utiliza?

Se lo utiliza en mantenimiento preventivo al encontrar fallos ocultos dentro de la gran cantidad de registros que permitan identificar problemas potenciales (ORACLE, s.f.).

Para el desarrollo de productos las empresas construyen modelos predictivos que analizan las características de los mismos y cuál ha sido su impacto, para determinar el éxito de los nuevos productos (ORACLE, s.f.).

Big data es un aliado indispensable en la experiencia del cliente, al trabajar con la data de servicios web, redes sociales, llamadas, etc. De tal modo de personaliza las ofertas por cliente, ofreciendo lo que realmente el cliente busca (ORACLE, s.f.).

En seguridad informática big data apoya al reconocer patrones que indican fraude o amenazas a la seguridad de la información (ORACLE, s.f.).

Big data es uno de los responsables de que el aprendizaje automático este en auge, al entregar esos colosales volúmenes de datos para que las maquinas aprendan (ORACLE, s.f.).

En la salud, big data aporta generando diagnósticos y presentando opciones de tratamiento, en base a la identificación de patrones en datos de pacientes con una determinada afección, y así podríamos continuar con un sinfín de ejemplos sobre el uso del big data (Powerdatas.f.).

Retos de big data

Hasta aquí se ha visto su utilidad, pero big data se enfrenta a grandes desafíos, por ejemplo, se ha indicado que disminuyo el costo de almacenamiento en años recientes, pero también se requiere nueva tecnología para esto ya que cada 2 años se duplica el volumen de datos según lo indica (ORACLE, s.f.).

Otro factor importante es que los datos estén disponibles y sean utilizable, según (ORACLE, s.f.) un científico de datos ocupa entre el 50 y 80% de su tiempo en la preparación de los datos para ser utilizables.

Y como todo en el mundo actual, las tecnologías asociadas al big data evolucionan rápidamente, si en un inicio el estándar era utilizar Hadoop, al paso del tiempo fue utilizar Spark, hoy en día una combinación de ambos y mañana será algo nuevo que revolucione la industria (ORACLE, s.f.).

¿Cómo funciona?

Biga data necesita de tres acciones clave (ORACLE, s.f.).

Integración

Como se ha indicado se tiene diversas fuentes de datos a lo que no se puede aplicar un proceso clásico de extracción, transformación y carga (ORACLE, s.f.), aquí se requiere definir la estrategia y tecnología que permitan recibir los datos de manera adecuada. Tomando en cuenta que en algunos casos serán petabytes de información (Mailjet, s.f.).

Gestión

Toda esta data necesita un lugar en el que se la almacenará, pudiendo ser en la infraestructura local en la nube o en un esquema mixto. Además, se debe definir si los datos estarán disponibles o no en tiempo real. Hoy en día tenemos centros de datos disponibles a nivel mundial o local lo que hace más atractiva esta solución (ORACLE, s.f.).

Análisis

El siguiente paso es analizar la data obtenida, mostrando de forma gráfica los hallazgos para toma de decisiones. Cada organización busca sacar el mayor provecho a sus datos y

obtener ventajas competitivas, para devengar la inversión realizada en infraestructura para big data.

En la Figura 4, podemos observar como típicamente se trabaja con datos, partiendo de fuentes de diversas índoles como redes sociales, correos, aplicativos, sensores, etc., son adquiridos por software como Spark o Kinesis, para ser procesados y almacenados en un lago de datos o base de datos, que servirán de insumo para el análisis y aplicación de modelos, que finalmente se entregan a alguien para la toma de decisiones.

Figura 4: Proceso de recolección, procesamiento y visualización de datos

Si tenemos claro que es big data, su importancia y cuál ha sido su contribución almundo contemporáneo y el auge de los modelos de inteligencia artificial, necesitamos profundizar en este nuevo concepto en 2.1.5

2.1.5. Inteligencia artificial

(López, s.f.) dice que el objetivo de la inteligencia artificial consiste en que una maquina tenga inteligencia general comparable a la humana. Esto es realmente algo muy ambicioso e incluso lo compara, con querer explicar el origen de la vida o del universo.

A lo largo de la historia humana este anhelo de querer construir máquinas inteligentes ha conducido a modelos que emulen el cerebro humano, Descartes en el siglo XVII lanzaba la pregunta si con un sistema mecánico lleno de engranajes y poleas se podría emular pensamientos humanos (López, s.f.).

Hoy en día al pensar en inteligencia artificial se viene a la mente motores de recomendaciones que automáticamente muestra programas en base a los hábitos de la persona

(ORACLE México, s.f.). Aunque hay quienes relacionan el concepto con robos humanoides que quieren tomar el control del mundo, lo que realmente se busca es mejorar las capacidades y contribuciones de la humanidad lo que en el ámbito empresarial es muy valioso (ORACLE México, s.f.).

La inteligencia artificial de divide en fuerte y débil:

(López, s.f.) menciona que la inteligencia artificial fuerte es aquella en la que el computador no simula la mente es realmente una mente pudiendo tener una inteligencia superior a la humana. Aunque indica que esto es improbable que ocurra.

Mientas que para inteligencia artificial débil (López, s.f.) indica que se enmarca en construcción de algoritmos que ejecuten tareas específicas. Los computadores actuales han demostrado tener capacidades superiores a las humanas en ciertos dominios como cálculos matemáticos con cientos de variables o al jugar al ajedrez, aquí también se incluye la formulación de hipótesis por lo que se puede decir que todo avance actual en este campo es parte de la inteligencia artificial débil.

Los algoritmos de inteligencia artificial que fueron entrenada para detectar un fraude no son capaces de conducir un automóvil o jugar a GO o recomendarnos que película ver (SAS. S.f.), esto porque fueron entrenados para cumplir únicamente con esa tarea específica.

En la industria, la inteligencia artificial no es un producto que se pueda vender de manera individual, en la mayoría de los casos será un complemento que mejora lo que ya utilizamos enel día a día, un claro ejemplo de esto se dio al incorporar Siri a los equipos de Apple (SAS. S.f.).

La inteligencia artificial no llego para sustituir a los humanos, ella nos ayuda a ser mejores en lo que hacemos, tomando en cuenta que los algoritmos de inteligencia artificial aprenden diferente de como lo hacen los humanos y ven patrones que nosotros no vemos (SAS. S.f.).

¿Cómo funciona?

Lo hace combinando volúmenes inmensos de datos iterativos que se procesan rápidamente por algoritmos inteligentes, que permiten que la maquina aprenda patrones y características (SAS. S.f.).

En la Figura5 como subcampos principales podemos observar:

Aprendizaje basado en máquina (machine learnig) emplea redes neuronales, investigación de operaciones, física para encontrar criterios ocultos en los datos, sin haber sido programado de manera expresa para ello (SAS. S.f.).

Redes neuronales es aprendizaje basado en máquina que se basa en unidades interconectadas similar a las neuronas del cerebro para procesar información y encontrar significado a los datos (SAS. S.f.).

Aprendizaje profundo (deep learning) utiliza grandes redes neuronales que aprovechan el poder de cómputo para aprender patrones complejos dentó de big data (SAS. S.f.).

Computo cognitivo busca la interacción del tipo persona con la computadora en la que se simule el proceso humano de entender imágenes y el habla, para entregar una respuesta coherente (SAS. S.f.).

Visión por computadora basada en aprendizaje profundo y en el reconocimiento de patrones para reconocer lo que está dentro de una imagen o de un video, pudiendo ejecutarlo en tiempo real (SAS. S.f.).

Procesamiento de lenguaje natural esto es la capacidad para que un computador sea capaz de entender, analizar y generar lenguaje humano (SAS. S.f.).

A partir de aquí, nos adentramos en estos subconjuntos a ver algo más de detalle que los clarifique.

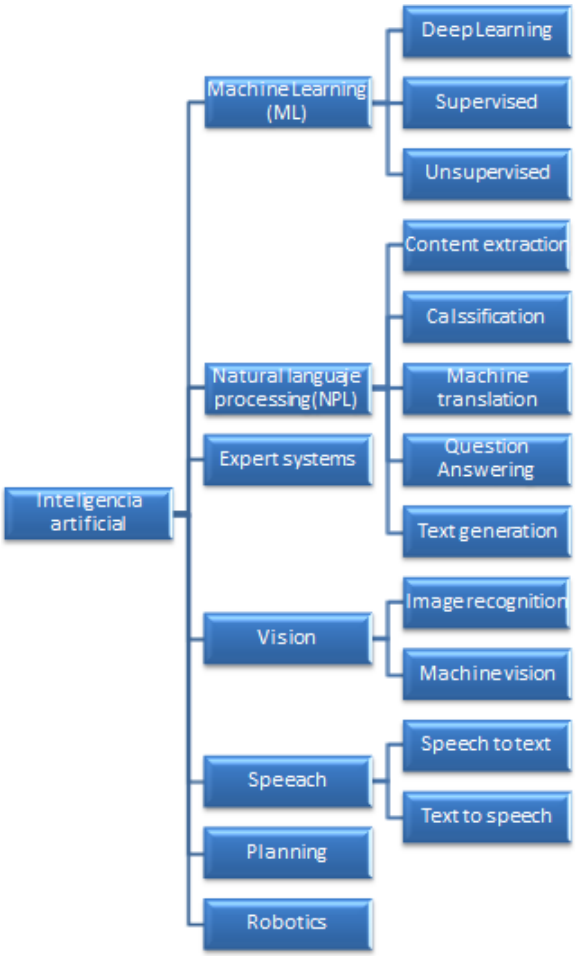


Figura 5: Subcampos de la inteligencia artificial

2.1.6. Machine learning

Machine learning o el aprendizaje automático como se ve en la Figura 6, es un subconjunto dentro del mundo de la inteligencia artificial, y trata de dar a las maquinas la capacidad de aprender en base a la experiencia (Janiesch, Zschech, & Heinrich, 2021).

Para crear este concepto se debió pasar por un sin número de investigaciones con las que se concluye que en lugar de programar manualmente sofisticados y complejos algoritmos que logren imitar la inteligencia humana, sería mejor entregar una gran cantidad de datos a la máquina para que encuentre patrones y de ese modo aprenda (Janiesch, Zschech, & Heinrich, 2021).

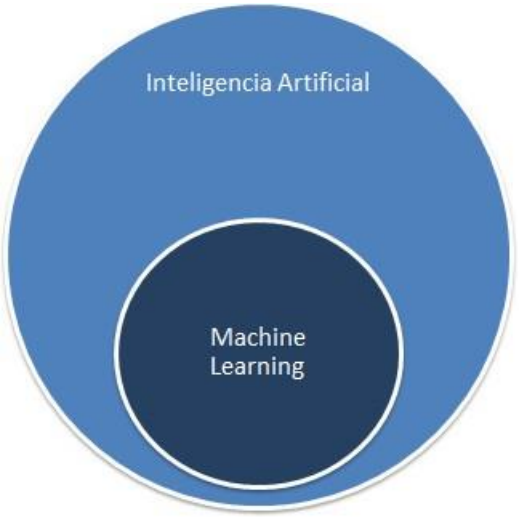


Figura 6: Representación de machine learning dentro de la inteligencia artificial

¿Pero cómo empezó todo? (Canal AprendeIA con Lidgi Gonzalez, 2018, 0m28s)

Reamente este no es un concepto nuevo, para iniciar debemos remontarnos a 1950 y la famosa prueba del matemático Alan Turing, en la que teóricamente se podría predecir si una máquina era capaz de pensar como un humano. Aquí un evaluador humano analizaba las respuestas dadas en lenguaje natural (texto) por un computador y otra persona, si el evaluador no es capaz de distinguir entre la máquina y el otro humano de forma correcta, la maquina habríapasado la prueba (Biografías y Vidas, s.f.).

Para 1952 el informático Arthur Samuel crea el primer programa capaz de aprender, se trataba de un juego de damas que aprendía de los errores que cometía en cada juego (Conferencia de Darthmouth 1956, s.f.).

Fran Rosenblatt en 1958 crea la red neurona (en hardware) llamada Perceptrón que posteriormente se convirtió en un algoritmo y la base para el deep learning y el reconocimiento de características (Fran, 2018).

Primer invierno de la inteligencia artificial, debido a las altas expectativas y bajos resultados, se dejan de recibir fondos para continuar con la investigación (CICE, s.f.)

Para 1979 estudiantes de la universidad de Stanford inventan robot móvil capaz de evitar los obstáculos que se encontraban en la habitación (Gonzalez, 2018).

En la década de los 80’s nacen los sistemas expertos basados en reglas, esto es adoptado por el mundo corporativo lo que da un nuevo empuje a machine learning, en 1981 Gerald De Jong crea el aprendizaje basado en explicaciones, aquí un computador analiza datos y crea reglas generales para descartar datos según su importancia. Dentro de esta misma época Terrence Sejnowski inventa NETtalk que es un sistema que aprende a pronunciar palabras comoun niño (Gonzalez, 2018).

A final de los 80´s tenemos el segundo invierno de la inteligencia artificial, el que se extendió hasta 1993, con el triunfo de Deep Blue de IMB sobre el campeón mundial de ajedrez Gary Kaspárov se retoma el impulso machine learning (Gonzalez, 2018).

En 2011 Watson de IBM vence a competidores humanos en un concurso de respuestas a preguntas en lenguaje natural (Gonzalez, 2018).

Con el avance de en la tecnología los pasos empiezan a ser más grandes, en 2012 se crea el proyecto GoogleBrain que es una red neuronal profunda que detecta patrones en imágenes y videos (Gonzalez, 2018).

En 2014 Facebook crea DeepFace que es capaz de reconocer personas, tal como lo haría un humano, en este mismo año DeepMind crea un algoritmo que en base a los pixeles de las imágenes de la pantalla es capaz de jugar juegos de Atari y ganar a humanos expertos en aquellos juegos (Gonzalez, 2018).

La organización OpenIA aparece en el 2015 para promover, desarrollar y asegurar que los desarrollos de inteligencia artificial tengan un impacto positivo para el ser humano (Gonzalez,2018).

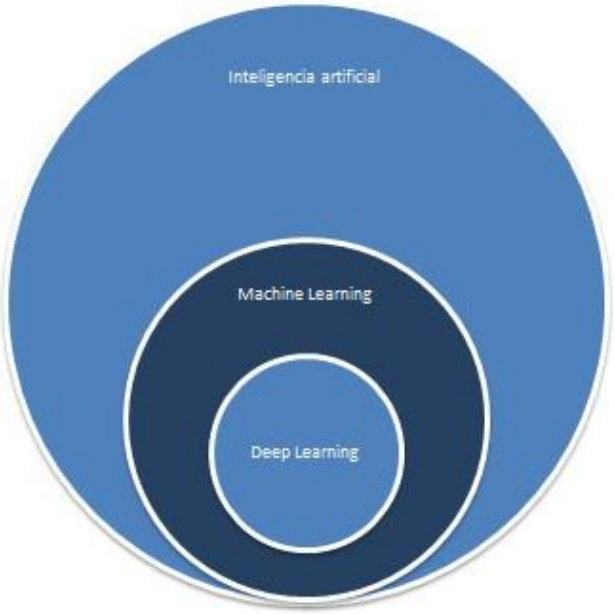
A partir de aquí estamos en una nueva ola de avances y popularización de machine learning, aunque existen quienes consideran que podemos llegar también a un tercer invierno para la inteligencia artificial (Gonzalez, 2018).

2.1.7. Deep learning

Deep learning es un nuevo horizonte de la inteligencia artificial y como se ve en la Figura 7, es un subconjunto de machine learning, imita la conectividad de las neuronas dentro del

cerebro humano, siendo capaz de encontrar correlaciones entre los datos, pero para ello necesita una gran cantidad de datos y entre más grande sea el número de datos sus predicciones serán más precisas (Gonzalez, 2021).

Deep learning es la que tiene mayor auge recientemente debido a que se tiene disponible enormes cantidades de datos y se posee el poder computacional que requieren estos algoritmos



(DataScientest, 2022).

Figura 7: Representación de deep learning dentro de la inteligencia artificial

¿Cómo funciona?

Como se ha mencionado anteriormente, se trata de un conjunto de neuronas artificiales entrelazadas, y que se distribuyen en tres capas principales como se muestra en la Figura 8 (DataScientest, 2022).

Capa de entrada: compuesta por las neuronas que reciben los datos de entrada, como imágenes, texto, audio, etc. (Gonzalez, 2021).

Capa oculta: son las n capas encargadas del procesamiento y extracción de características (Gonzalez, 2021).

Capa de salida: es aquella que entrega la conclusión en base a los cálculos ejecutados (Gonzalez, 2021).

Una vez que se entrega los datos de entrada y estos pasan a través de las diferentes capas para obtener una salida, se aplica Backpropagation que consiste en encontrar el error al comparar la salida obtenida con la salida deseada. El error es propagado desde la salida a cada neurona de las diferentes capas ocultas entregándoles la fracción del error que les corresponde, en base a ese valor del error se ajustan los pesos y se mejora la predicción.

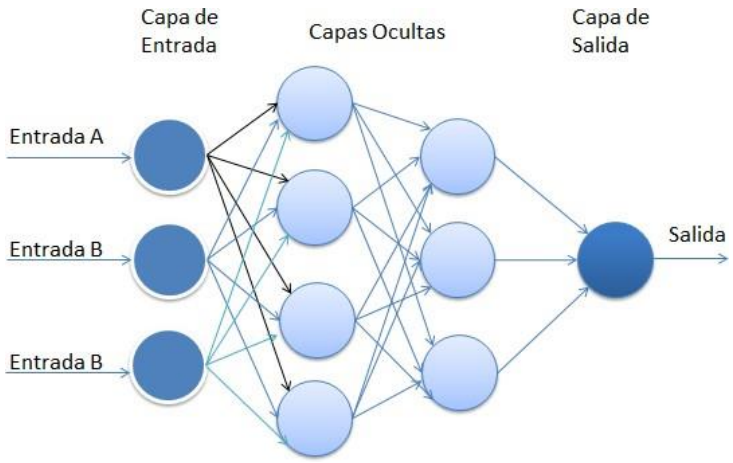


Figura 8: Red neuronal

¿Dónde se aplica?

Hoy en día está en todas partes, como usuarios no estamos conscientes de que en nuestras actividades del día a día se aplican este tipo de algoritmos, por ejemplo, cuando subimos fotografías en una red social, cuando en las aplicaciones de streaming nos sugiere títulos que pueden interesarnos o si utilizamos una de las asistentes virtuales para ejecutar cualquier acción como pedir comida por una aplicación (González, 2021).

Ahora si pasamos a pensar en la industria, la medicina, la educación, la ciencia, etc. Los ejemplos de uso son aún más interesantes, como en el sistema de navegación de un vehículo autónomo que es capaz de reconocer situaciones peligros y evitar accidentes al reaccionar a cambios bruscos de carril o frenazos sin aviso previo.

En la investigación médica el estudio del cáncer con algoritmos de deep learning es lo que está de moda y entregando resultados alentadores tanto así que los proveedores como NVIDIA han lanzado su propio framework Nvidia Clara (NVIDIA. DEVELOPER, s.f.) para este propósito, con el objetivo de

tener diagnósticos tempranos, precisos y a costos accesibles.

En tema servicios han proliferado los chatsbots que interactúan con el usuario para contratar nuevos servicios, pedir cambios al servicio actual, cancelar suscripciones, reportar problemas o simplemente solicitar información.

Otro campo con importantes avances es el reconocimiento facial, es mas en Ecuador los municipios en camino a convertirse en ciudades inteligentes han optado por esta tecnología, un claro ejemplo es la ciudad de Guayaquil con la ejecución de algoritmos de reconocimiento facial y reconocimiento de actividades violentas que se ejecutan en el sistema de cámara desplegadas a lo largo de la ciudad.

2.1.8. Aprendizaje supervisado

Es otro de los subconjuntos de machine learning en el que se requiere para su funcionamiento dos grupos de datos debidamente etiquetados, uno para entrenamiento y otro para pruebas, en las que se determinará si la eficacia del modelo es la deseada y se lo puede pasar a producción con datos reales que no han sido vistos en el entrenamiento ni en las pruebas(Bagnato, 2017).

Los algoritmos más utilizados son (Bagnato, 2017):

- k-vecinos cercanos
- Regresión lineal
- Regresión logística
- Máquina de vector de soporte
- Clasificador bayesiano
- Arboles de decisión y bosques aleatorios
- Redes neuronales
- Aprendizaje profundo

El aprendizaje supervisado se divide en dos grupos:

Aprendizaje supervisado por clasificación: es aquel en el que las entradas producen una salida discreta, es decir se trata de etiquetarlas dentro de una de las clases conocidas por

el algoritmo como por ejemplo en la Figura 9, se muestra un clasificador de imágenes al que se entrena con imágenes de vegetales y para la prueba le pasamos imágenes nunca antes vistas por el modelo y pueden ser clasificadas correctamente (TIBCO, s.f.).

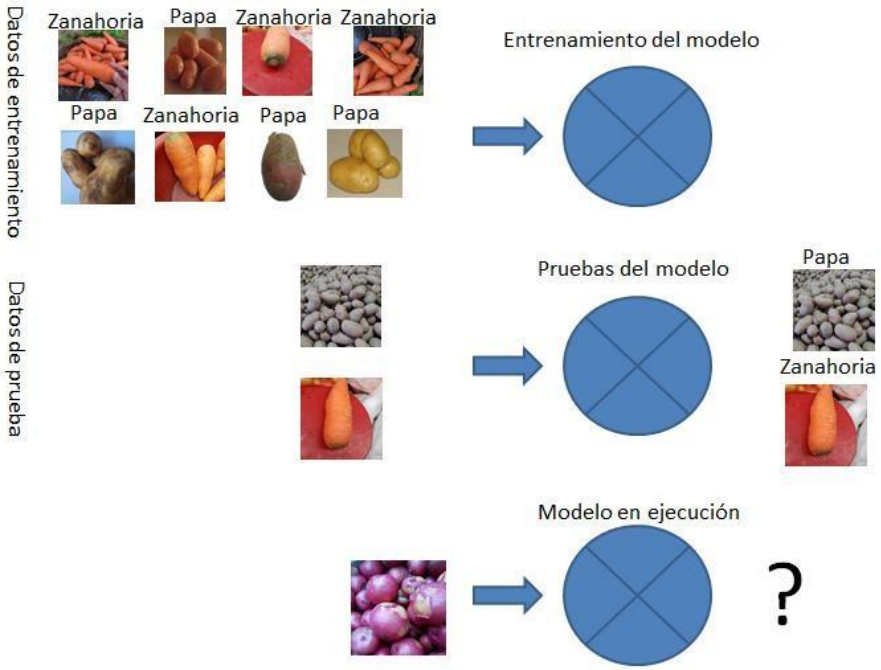


Figura 9: Aprendizaje supervisado por clasificación

Aprendizaje supervisado por regresión: Es aquel que su salida es un valor continuo y lo predice en base a la relación entre las variables independientes y la variable dependiente como lo podemos observar en la Figura 10 donde se representa por medio de una línea la relación de las variable X y Y, como ejemplos podemos mencionar, determinar el valor de una propiedad en base a su ubicación, antigüedad, número de habitaciones, cercanía a institucioneseducativas, seguridad, etc. (Blanco, 2019).

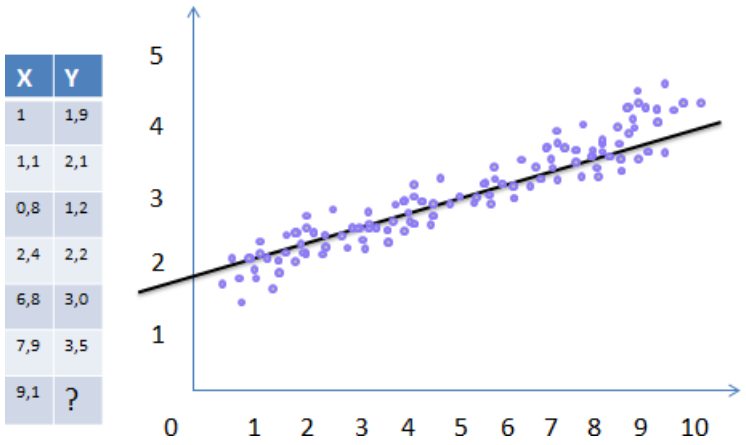


Figura 10: Relación entre variable dependiente y la variable independiente

2.1.9. Aprendizaje no supervisado

Al igual que el aprendizaje supervisado y Deep learning, este también es parte de machine learning. Este tipo de algoritmos se utiliza para resolver problemas de la vida real, al realizar procesamiento más complejo partiendo de datos no etiquetados, que son los que mayoritariamente se encuentran. Los algoritmos se encargan de identificar patrones, relaciones complejas, características, etc. Entre los diferentes grupos de datos.

Al no contar con estas etiquetas no es posible aplicar directamente sobre problemas de clasificación o regresión al no tener clara cuál es la salida (APRENDEIA, s.f.).

Generalmente son utilizados para exploración de la data agrupándola según patrones que el algoritmo identifica y que eran desconocidos en inicio (APRENDEIA, s.f.).

Pudiendo ser utilizado para detección de anomalías encontrando datos que se comportan de manera inusual, puede identificar la aceptación de un nuevo producto en un mercado desconocido, en los equipos inteligentes es el responsable de la forma en la que se agrupan las imágenes en las galerías (APRENDEIA, s.f.).

Los algoritmos más utilizados son (Bagnato, 2017):

- Agrupamiento k-medias
- Análisis de componentes principales
- Detección de anomalías

El aprendizaje no supervisado se divide en dos grupos (Bagnato, 2017):

Agrupación: Como se ve en la Figura 11 los algoritmos buscan generar cúmulos de datos en base a características similares, pudiendo indicarle la cantidad de grupos que se desea tener o determinar cuál es el número óptimo de ellos. Dentro de esto podemos tener:

- Grupos exclusivos donde un dato solo puede pertenecer a un grupo.
- Grupos solapados cuando un dato puede pertenecer a varios grupos
- Grupos aglomerados cuando existe una relación jerárquica.

Grupos probabilísticos aquellos que su distribución es por medio de la probabilidad de que dato pertenece a uno u otro grupo.

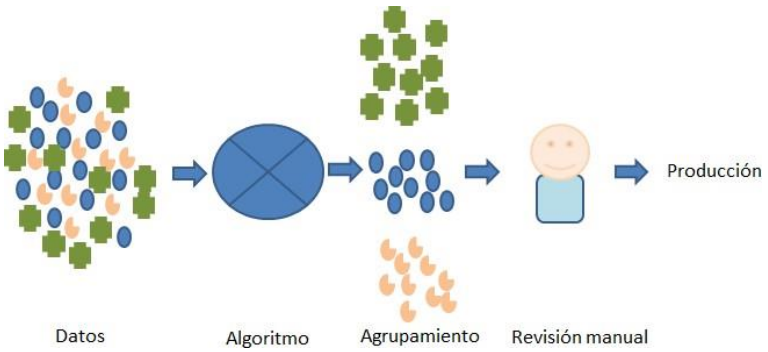


Figura 11: Aprendizaje no supervisado por agrupación

Reducción de dimensiones: los algoritmos están enfocados en entregar el mejor resultado utilizando las características más relevantes de los datos, pero eso no siempre se inicia de este modo, podemos partir con una enrome cantidad de datos con exagerada cantidad de características que, en lugar de aportar a la eficiencia del modelo, empezaran a degradar los resultados por un sobreajuste, además dificultan que la personas pueda visualizar los datos conlos que este trabajado (APRENDEIA, s.f.).

Es aquí donde se utiliza la reducción de dimensiones y podemos aplicar eliminar características en base a la relevancia que poseen o generar nuevas variables a partir de las existentes (APRENDEIA, s.f.).

No existe una única técnica para ello, como se observa en la Figura 12, tenemos una variedad de técnicas divididas en selección de características (ratio de valores perdidos, filtros

de baja varianza, filtro de alta correlación, etc.) y reducción de dimensionalidad (análisis defactores, análisis de componentes, escalar multidimensional, etc.) (APRENDEIA, s.f.).

Figura 12: Técnicas para reducir la dimensionalidad

2.1.10. Aprendizaje semisupervisado.

Técnica utilizada cuando se tiene una pequeña cantidad de datos etiquetados y una gran cantidad de datos sin etiquetar porque resulta demasiado costoso hacerlo, esto es lo que generalmente se tiene en la vida real. Es aquí que se requiere combinar el aprendizaje supervisado con el no supervisado para obtener mejores resultados (Ibáñez. 2019), esto lo podemos ver en la Figura 13 donde los datos sin etiqueta se muestran en gris, en un conjunto más grande que los etiquetados, pero de todos modos el modelo logra realizar la clasificación.

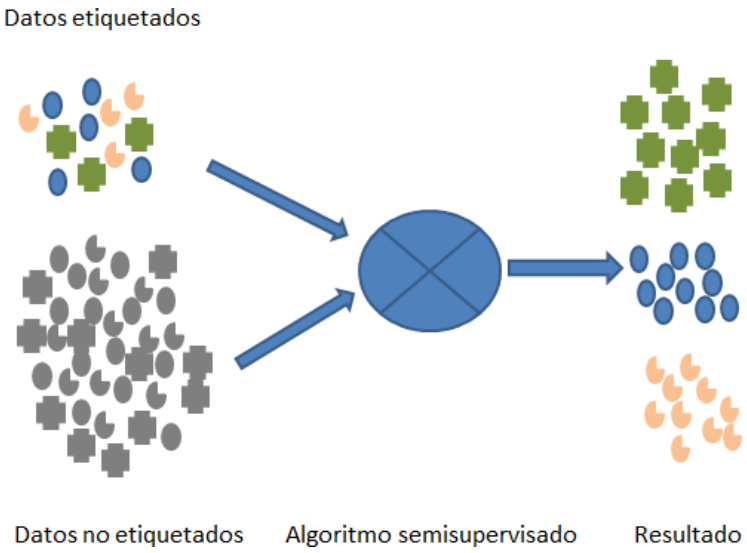


Figura 13: Aprendizaje semisupervisado

Un claro ejemplo de esto es trabajar con grabaciones telefónicas en las que su etiquetado resulta demasiado costoso y requiere de mucho tiempo por parte de las personas que tiene la labor de etiquetar, (Ibáñez, 2019).

El método comúnmente utilizado para estos casos es auto entrenamiento, como se muestra en la Figura 14 consiste en entrenar un clasificador con la data etiquetada, luego etiquetamos lo data no etiquetada con este clasificador, como paso siguiente se agrega en los datos originales etiquetados a aquellos que el clasificador le asignó una etiqueta y tiene un alto índice de confianza, este proceso se repite hasta que no sea posible añadir nuevos datos etiquetado al conjunto original (Morales, s.f.).

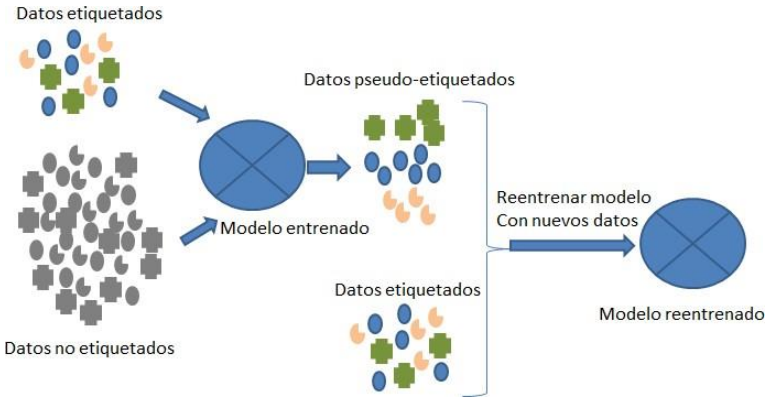


Figura 14: Auto entrenamiento

2.1.11. Detección de anomalías

(TIBCO, s.f.) define que una anomalía es todo comportamiento que difiere de lo esperado para el conjunto de datos con el que se está trabajando y generalmente se utilizan como alertas de ellos, como se lo puede apreciar en la Figura 15 como aparecen puntos que no pueden ser colocados como parte de la clase 1 o 2 y en la Figura 16 se tiene un comportamiento lejano a lo esperado lo cual también es considerado una anomalía, además de ser parte del aprendizaje nosupervisado según se indica en 2.1.9.

Algo que debe tener muy claro, es que las anomalías no se deben considerar como algo malo o bueno, simplemente pasan, pero deben ser visibles para que el momento que se presente la institución analice si es necesario tomad alguna acción (TIBCO, s.f.).

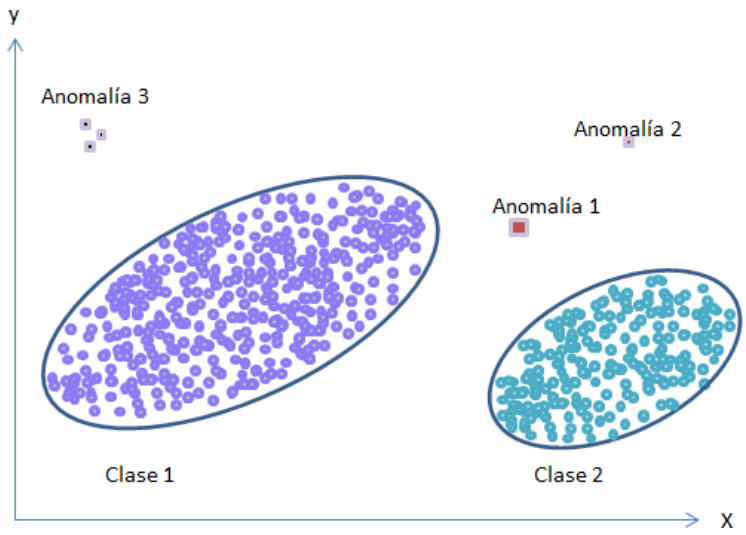


Figura 15: Ejemplo de como se ve una anomalía entre clases

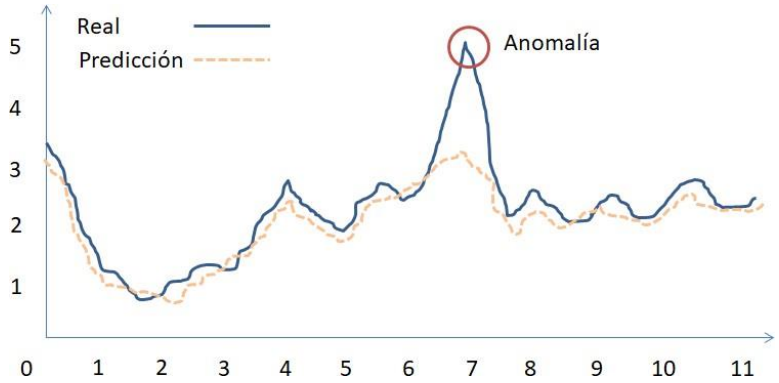


Figura 16: Ejemplo de como se ve una anomalía en una predicción

Dado que al día de hoy lo común es que las empresas generen grandes cantidades de datos, el uso correcto de ellos le otorga una ventaja competitiva sobre el resto de la industria, tomando decisiones rápidamente al encontrar estas anomalías puede evitar que problemas pequeños crezcan y se conviertan en dolores de cabeza innecesarios (TIBCO, s.f.).

¿Cómo se detectan estas anomalías?

Se tiene un abanico de opciones para este propósito (identificarlas o predecirlas) (TIBCO, s.f.).

Descubrimiento visual

Tener visualizaciones que nos muestren el comportamiento de los datos ayuda a las empresas a verificar si todo sigue los patrones esperados, pero se debe considerar que estas visualizaciones requieren de profesionales de ciencia de datos con conocimiento de otras ramascomo el comercial para presentar algo que realmente comunique lo que está pasando de unamanera entendible para quien lo interprete, cosa que no es tan sencilla de ejecutar (TIBCO, s.f.).Se puede utilizar técnicas de **aprendizaje supervisado o no supervisado**, en 2.1.8 y 2.1.9 se habla del detalle de ello.

Por medio de series de tiempo

Estos modelos son capaces de capturar estacionalidad, niveles o tendencias en un grupode datos, por todo aquello que incumpla esto se puede considerar como una anomalía, siempre y cuando no sea una falla del propio modelo (TIBCO, s.f.).

Por agrupamiento

De modo similar al anterior, los datos pueden ser agrupados en base a sus característicasen alguno de los grupos existentes y aquellos datos que no calcen en alguno de ellos, pueden ser considerados anomalías (TIBCO, s.f.).

¿En qué campo se utiliza la detención de anomalías?

Pues esta práctica se ha extendido en prácticamente toda industria, pero los casos máscomunes tienen que ver con detectar delitos financieros, fraudes en seguros, defectos en el proceso de fabricación e identificación de daños por medio de los sensores en equipos,

ciberseguridad, protección contra piratería, en imágenes médicas para encontrar tumores o

detectar cáncer, y así podríamos continuar nombrado un sin número de ejemplos más (TIBCO, s.f.).

Se menciona principalmente estos, debido a que por ejemplo en el sector financiero la afectación puede ser por miles de millones de dólares y es necesaria una identificación de estas anomalías en tiempo real, por ello estas empresas son las que lideran el campo de la detección de anomalías (TIBCO, s.f.).

Otro método bastante utilizado para identificar fraudes es por medio de la detección de datos aberrantes, esto lo exploraremos seguidamente.

2.1.12. Detección de valores aberrantes

(TIBCO, s.f.) dice que los valores aberrantes o atípicos son aquellos que están en los extremos alejados del promedio y fuera de lo q se esperaba, sobre estos valores aberrantes se debe decir si se los elimina o se los tratarlos de algún modo para evitar sesgo y tener datos útiles para el análisis que se requiera.

Por ejemplo, si se tiene una curva de campana normal todo lo que este por fuera a la izquierda o derecha, se considera valores atípicos que pueden indicar fraudes o el comportamiento que se esté tratando de encontrar, al ser datos que difieren radicalmente de lo estimado como se muestra en la Figura 17, por cada mes se resalta en rojo como valores aberrantes aquellos que esta por fuera de los bigotes del diagrama de cajas (TIBCO, s.f.).

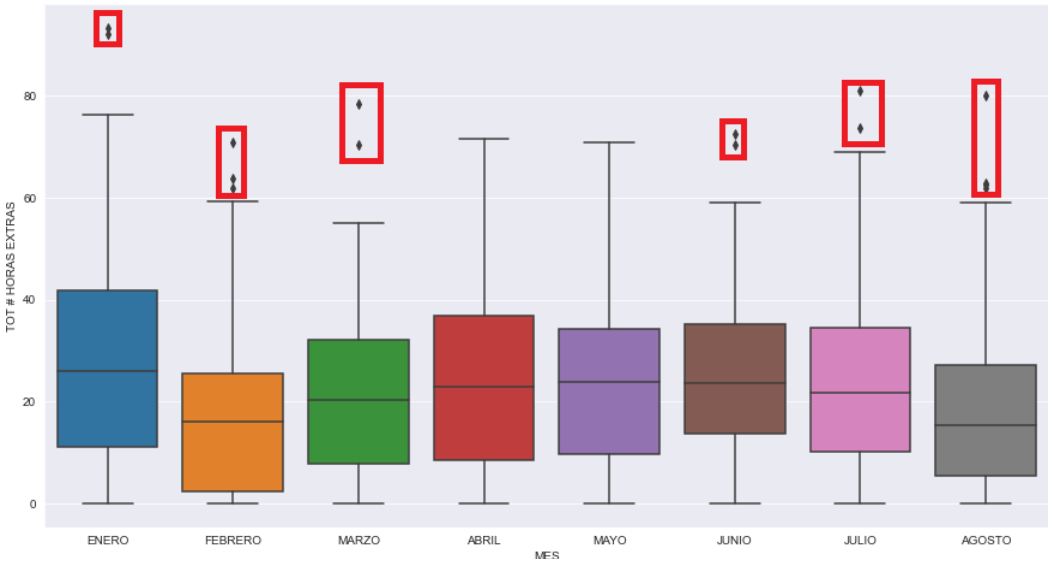


Figura 17: Representación de valores aberrantes

Se tienen valores atípicos univariados cuando se trata de un dato con valores extremos y, los atípicos multivariados se tratan de una combinación de puntos de datos (por lo menos dos) con valores aberrantes (TIBCO, s.f.).

¿Ahora bien, pero que produce estos valores aberrantes?

(TIBCO, s.f.) nos indica que existen ocho causas principales para la aparición de ellos:

1. Como el más común se tiene el error humano en el ingreso de los datos (TIBCO, s.f.).
2. Emplear códigos en reemplazo de valores (TIBCO, s.f.).
3. Extracción incorrecta de datos (fuentes erróneas o contaminados con otros datos)(TIBCO, s.f.).
4. Variables distribuidas de una manera no esperada (TIBCO, s.f.).
5. Errores producidos por la propia aplicación (TIBCO, s.f.).
6. Valores generados a propósito con la intención de probar el modelo (TIBCO, s.f.).
7. Errores producto de la extracción de datos (TIBCO, s.f.).
8. Variante natural de los datos que representa el escenario que se está tratando de predecir(TIBCO, s.f.).

Todos estos puntos son los que debe analizar el científico de datos en la fase de limpieza y preparación de los datos para el uso del modelo (TIBCO, s.f.).

Hoy en día la cantidad de datos que se tiene a disposición es enorme y sigue creciendo, lo mismo ocurre con valores aberrantes que pueden interferir o confundir al modelo y generar información errónea o impedir que se tome la decisión adecuada en el momento justo (TIBCO, s.f.).

La identificación de datos aberrantes se puede utilizar para detectar fraudes electorales, en ensayos clínicos para verificar variaciones en las opciones de tratamiento, entre otros (TIBCO,s.f.).

Las técnicas que generalmente se utilizan para su detección son:

Valores numéricos anómalos

Es la técnica más sencilla, utilizando el diagrama de cajas los límites son los bigotes y todo lo que esté por fuera de ellos son considerados valores aberrantes y se los elimina (TIBCO,s.f.).

Z-score

Técnica poderosa que se utiliza para pequeños conjuntos de datos paramétricos, aquí se supone una distribución normal en base a la desviación estándar, por heurística se define los puntos de corte que se consideran normales y tolo que este por fuera se los remueve por ser valores atípicos (TIBCO, s.f.).

DBSCAN

Utilizado para cuando se tiene más de dos dimensiones, se debe escalar los datos para utilizarlo, este divide al cúmulo de datos en puntos centrales, puntos de borde y puntos que no son parte de ningún grupo (valores aberrantes), la mayor dificulta que presenta DBSCAN es la selección de los parámetros óptimos para el conjunto de datos, además de que para nuevos datos se requiere calibrar nuevamente el modelo (TIBCO, s.f.).

Bosques de aislamiento

Tienen pocos parámetros por lo que son fáciles de usar, no requieren escala los datos y no es necesario saber la distribución de los mismos, se basa en árboles de decisión binarios, construidos a partir de la selección aleatoria de características y del valor de división, para formar

un bosque, los valores se promedian, los que son cero representan a los valores normales y el1 a los atípicos (TIBCO, s.f.).

El problema de este método es la complejidad de la representación visual y el costo computacional (TIBCO, s.f.).

2.2. Soluciones de analítica y aprendizaje relacionadas al problema

Según (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022) el tema de churn en las telecomunicaciones tiene un auge importante por la nueva normalidad adoptada por el menudo entero, además de que la tasa de abandono en este sector es bastante alta entre un 20 y 40% anual.

Para tratar esta problemática se han propuesto aplicar diversidad de modelos a lo largo del tempo como:

- Máquinas de vectores de soporte (SVM)
- SVM con kernel gaussiano
- SVM con kernel de base radial
- SVM con kernel polinomial
- Metodologías híbridas
- XGBoost
- Bosques aleatorios
- Regresión logística
- LightGBM

Como podemos observar en la Figura 18 para predicción del churn se suelen utilizar modelos como SVM, Xgboost, bosques aleatorios, regresión logística, lightGBM etc., todos ellos con sus propias particularidades (set de datos y atributos distintos) que obtuvieron precisiones que están en el rango de 65% al 99%.

References	Dataset	The best classifier	Accuracy
Zhao et al., 2005 [3]	100000 instances, 171 attributes	Gaussian kernel Support Vector Machines	87.15%
Xia et al., 2008 [4]	3333 instances, 21 features	Radial basis kernel Support Vector Machines	90.88%
Shaaban et al., 2012 [5]	5000 instances, 23 attributes	Support Vector Machines	87.3%
Brandusoiu et al., 2013 [6]	3333 instances, 21 attributes	Polynomial kernel Support Vector Machines	85.63%
Rodan et al., 2014 [7]	5000 instances, 11 features	Radial basis kernel Support Vector Machines	94.3%
Keramati et al., 2014 [8]	3150 instances, 12 features	Hybrid methodology	95%
Zhou et al., 2019 [9]	3333 instances, 21 features	Polynomial kernel Support Vector Machines	91.67%
Ebrah et al., 2019 [10]	Dataset 1: 7033 instances, 21 features Dataset 2: 71,047 observations and 57 attributes	Support Vector Machines	Dataset 1: 87% Dataset 2: 99%
Kriti, 2019 [11]	7055 instances, 20 features	XGBoost	85%
Ullah et al., 2019 [12]	Dataset 1: 64,107 instances, 29 features Dataset 2: 3333 instances, 16 features	Random forest	Data set 1: 88.63% Dataset 2: 89.59%
Xin Hu et al., 2020 [13]	2681 instances	Hybrid methodology	98.87%
Panjasuchat et al., 2020 [14]	100000 instances, 99 attributes	Deep Q-Network	65.26%
Jain et al., 2020 [15]	3333 instances, 20 attributes	Logistic regression	85.24%
Mallika et al., (2020) [16]	7043 instances, 21 attributes	LightGBM	78.9%

https://doi.org/10.1371/journal.pone.0267935.t001

Figura 18: Accuracy por modelos para predecir el churn

https://doi.org/10.1371/journal.pone.0267935.t001

La metodología definida como clásica por (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022), de forma general es muy similar a lo que se ha decidió utilizar para el presente trabajo:

- Obtención y entendimiento del set de datos
- Escalar los datos
- Aplicar métodos de aumento de datos para mejorar el desequilibrio de clases
- Seleccionar las características relevantes (por ejemplo: selección secuencial haciaadelante (SFS) y selección secuencial hacia atrás (SBS))
- Desarrollar y entrenar el modelo
- Evaluar los resultados del modelo

Un detalle importante a tomar en cuenta es el mencionado por (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022), es el desbalanceo de los datos, generalmente la cantidad de clientes que cancelan sus servicios será menos que los clientes que se mantienen activos, y en este caso nosotros no somos la excepción. Aquí nos ofrecen dos recomendaciones importantes:

1. Con datos desbalanceados no enfocarse en la precisión y tomar como una mejor medidael F1.
2. Utilizar técnicas que nos ayuden a balancear la data, eso impactará directamente en los resultados de F1 y precisión.

En la Figura 19, se ve la composición del set de datos con el que trabajaron (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022) publicado en Kaggle de la empresa Orange SA Telecom Company, que tiene el detalle histórico de 3333 clientes, de los 51 estados de EE. UU, tiene 19 características y una etiqueta que indica si el cliente cancelo o no el servicio.

En nuestro caso el set de datos es propio con 46 características y alrededor de 300000 clientes.

No.	Variables	Variable description	Variable types
1	State	51 states of the US	Categorical
2	Account length	Active duration of accounts	Integer
3	Area code	Code of areas	Categorical
4	International plan	1 = use service, 2 = not use service	Boolean
5	Voice mail plan	1 = use service, 2 = not use service	Boolean
6	Number vmail messages	Amount of voice mail messages	Integer
7	Total day minutes	Total day call minutes clients have used	Continuous
8	Total day calls	Total amount of day calls	Integer
9	Total day charge	Total day fee	Continuous
10	Total eve minutes	Total call minutes clients have used in the evening	Continuous
11	Total eve calls	Total amount of evening calls	Integer
12	Total eve charge	Total evening fee	Continuous
13	Total night minutes	Total night call minutes clients have used	Continuous
14	Total night calls	Total amount of night calls	Integer
15	Total night charge	Total night fee	Continuous
16	Total intl minutes	Total call minutes clients have used for international calls	Continuous
17	Total intl calls	Total amount of international calls	Integer
18	Total intl charge	Total International fee	Continuous
19	Customer service calls	Amount of customer service calls made	Integer
20	Churn	1 = churner, 0 = non-churner	Boolean

https://doi.org/10.1371/journal.pone.0267935.t002

Figura 19: Set de datos utilizando por (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022)

https://doi.org/10.1371/journal.pone.0267935.t002

Como resultado del estudio utilizando SVM con kernel de base radial, ajustando sus hiperparámetros y utilizando métodos SFS o SBS para seleccionar las características principales, consiguieron una precisión del 99.01% y un F1 de 98.88%.

(Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) parten de una premisa diferente que es enfocarse en la optimización de los hiperparámetros por medio de algoritmos genéticos (GA), búsqueda aleatoria (RS) y búsqueda en cuadricula (GS), considerando que una buena definición de hiperparámetros repercute directamente en el resultado que arrojara el modelo. Tema muy importante para ser tomado en cuenta para nuestro modelo.

Como modelos (Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) utilizan al igual que (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022) bosques aleatorios y SVM, lo nuevo aquí es el uso de k-vecinos más cercanos (KNN).

Al igual que (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022) consideran que se debe trabajar solo con las características principales del set de datos por lo que utilizan un clasificador de información mutua para seleccionarlo, diferente a SFS o SBS propuesto en el trabajo de (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022).

Otra consideración en la que coinciden (Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) y (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022) es en buscar como equilibrar los datos con lo que se trabaja, aquí (Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) utilizan un submuestreo de proporción controlada.

En la Figura 20 podemos ver el esquema de los pasos que siguieron (Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022), procesando los datos y extrayendo las características principales como paso uno, luego la definición del submuestreo para mejorar el rendimiento del algoritmo como paso dos, el paso tres es la optimización de los modelos RF, DVM y KNN utilizando como optimizadores de hiperparámetros GA, RS, GS, y como paso final la evaluacióndel rendimiento de los modelos.

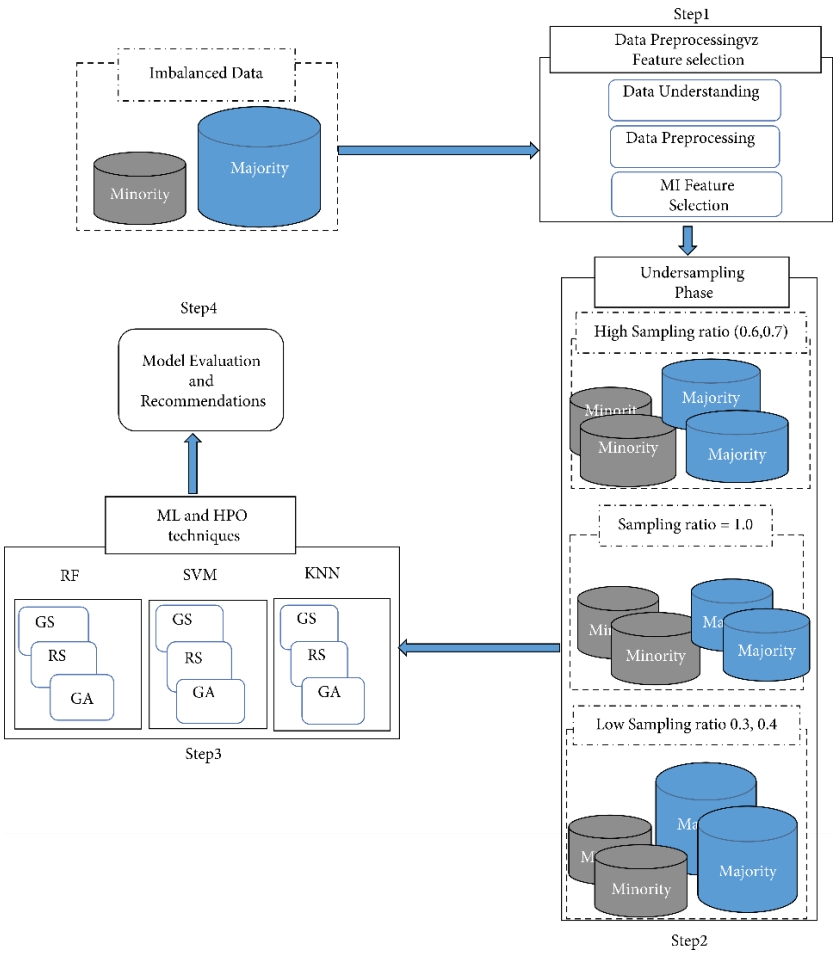


Figura 20: Esquema de (Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) para predecir el churn

https://www.hindawi.com/journals/mpe/2022/8534739/fig1/

(Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) determina que el mejor rendimiento se logra con un bosque aleatorio con optimización de hiperparámetros de búsqueda en cuadricula y un submuestreo de baja proporción (0.3) con lo que loga obtener como resultados 99% de precisión, un AUC del 99%, un F1 del 97% y un MAE 0.014

2.3. Librerías y software a utilizar

Python

Python lenguaje de programación cuyo origen fue a finales de los 80's e inicios de los 90's y se enmarcaba en ser fácil de usar y aprender, pero de gran capacidad, la limitante de la época fue la capacidad computacional que se requeriría para su ejecución (ESIC, 2020).

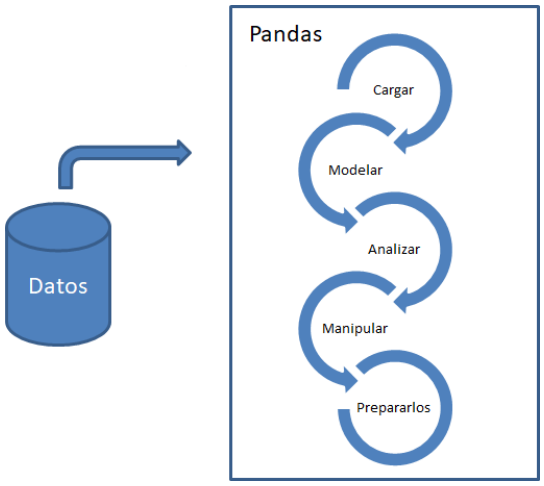
Big data, la ciencia de datos, la inteligencia artificial ha dado un repunte a este lenguaje ya que muchas de las herramientas desarrolladas para este tema han sido hechas en Python (ESIC, 2020).

El lenguaje tiene una estructura sencilla, por lo que cualquier persona con nociones básicas de programación lo puede entender, es gratuito y no requiere de licencia, por esta misma característica tienen una enorme comunidad que sigue desarrollado librerías, es multiplataforma(ESIC, 2020).

Como desventajas tenemos que es lento porque no se compila sino sé que se interpreta, tiene un alto consumo de memoria y fue pensado para aplicaciones de escritorio por lo que para el ambiente móvil no es muy recomendable.

Pandas

(Chacón J, 2021) nos dice que se trata de una librería de código abierto bastante utilizada en el mundo de la ciencia de datos, al ser flexible y potente, que tiene las funciones que un científico de datos requiere trabajar con cualquier grupo de datos, en la Figura 21 se muestra las etapas que se ejecuta con



pandas: carga, modelar, analizar, manipular y preparar los datos.

Figura 21: Funcionamiento de pandas

El propósito de Pandas es poder manejar cualquier tipo de dato y por ello permite cargarlo desde diferentes formatos de fuentes como por ejemplo json, xls, csv, etc. (Chacón J, 2021).

En su estructura se maneja principalmente dos tipos de datos que son series y dataframe, en la Figura 22 se puede ver que:

Serie: es un array que posee una única dimensión en la cual se puede almacenar cualquier dato (Chacón J, 2021).

Dataframe: es una estructura de dos dimensiones, con N columnas y cada columna es una serie (Chacón J, 2021)

Serie 1

Nombre
Juan
Dina
Pedro
Carol

Serie 2

Apellido
Castro
García
Romo
Escalante

Serie 3

Edad
15
21
12
45

Dataframe

Nombre	Apellido	Edad
Juan	Castro	15
Dina	García	21
Pedro	Romo	12
Carol	Escalante	45

Figura 22: Series y dataframe

Numpy

Es una librería de código abierto que maneja matrices con varias dimensiones y con una variedad de herramientas para manejarlo en el terreno del cálculo científico. Esta librería es tan importante que hay librerías basadas en ella, por ejemplo, Pandas (Cardellino F, 2021).

Maneja herramientas para estadística, algebra, etc., y permite conectarse con una gran variedad de base de datos (Cardellino F, 2021).

Matplotlib

(DataScientest, s.f.) nos dice que esta librería de código abierto fue desarrollada para tener las funcionalidades de MATLAB en Python, pero al ver su potencial a lo largo del tiempo de la ha ido mejorando, pudiendo dibujar histogramas, barras o cualquier otra clase de gráfico con una mínima cantidad de código.

La principal dificultad que presenta es la cantidad de opciones que posee, al ser una librería de 70 000 líneas de código que permiten generar gráficos simples en dos dimensiones o complejos en tres dimensiones, un problema adicional que se debe tener en mente es que a pesar de que esta librería evoluciona constantemente su documentación no lo hace al mismo ritmo por lo que los ejemplos que se encuentre pueden ser obsoletos (DataScientest, s.f.).

Una vez realizado todo el procesamiento de los datos se los debe mostrar al usuario, aquí entra en acción esta librería generando visualizaciones que permiten al humano interpretar

rápidamente los resultados, en la Figura 23 podemos ver ejemplos de su galería como: plot,satter, bar, steam, step, stackplot, etc.(DataScientest, s.f.)

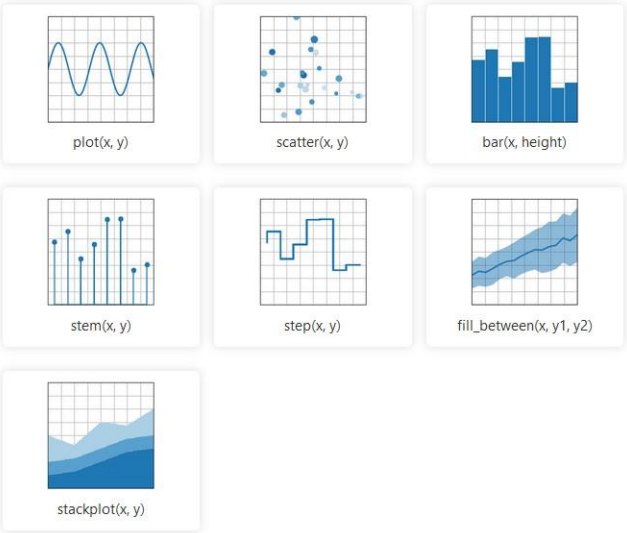


Figura 23: Ejemplo de galería de matplotlib

Seaborn

(Carmona P, 2020) nos dice que esta librería que permite realizar gráficos estilizados, esta librería está basada en Matplotlib, altamente relacionada con Pandas y utilizada para representación estadística.

Como una de las ventajas se tiene que esta librería es trabajar directamente con los dataframe, con una galería de opciones para todos los gustos y se puede generar de forma fácil gráficos que en matplotlib resultarían bastante trabajoso, en la Figura 24 se ven ejemplos de gráficos generados con esta librería, en la que resalta la presentación más estilizada (CarmonaP, 2020).

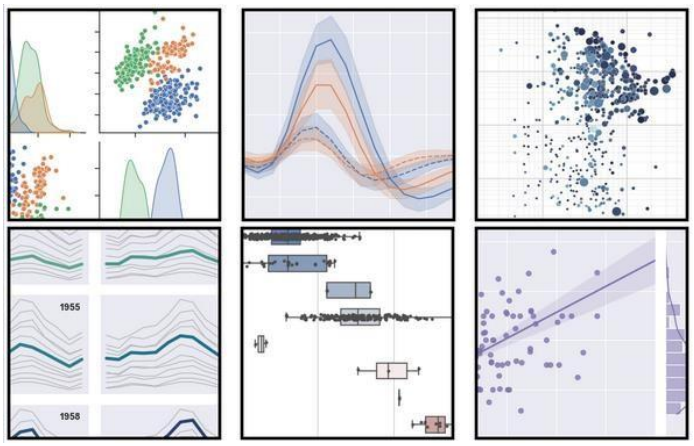


Figura 24: Ejemplo de gráficos generado en seaborn

Tensorflow

Es un desarrollo de código abierto de Google que actualmente es la más importante que se utiliza para deep learning (Puentes Digitales, 2018).

Esta biblioteca se utiliza para construir y entrenar redes neuronales, sus cálculos pueden ser ejecutados sobre CPUs o GPUs, su desarrollo fue para ser utilizada en la investigación de redes neuronales profundas, por un grupo de ingenieros de Google que trabajaban en el proyecto Google Brain Team (Puentes Digitales, 2018).

Su funcionamiento se basa en gráficos del flujo de los datos, en ellos los nodos son las operaciones matemáticas y los bordes son los tensores o matrices de datos de N dimensiones (Puentes Digitales, 2018).

Actualmente se lo utiliza en el campo de la fotografía para mejorar imágenes, en el análisis de radiografías y para interactuar con el internet de las cosas (IoT) (Puentes Digitales, 2018).

Keras

Keras es una biblioteca escrita en Python, es de código abierto y desarrollada por ingeniero de Google, cuyo objetivo era dotara una herramienta que permita a los principiantes adentrarse en el mundo de las redes neuronales, Keras por si se integra a los demás marcos de trabajo que actualmente existen para aprendizaje automático y los utiliza como su motor pudiendo ser TensorFlow (con el cual está íntimamente relacionado), Theano entre otros (IONOS. 2020).

Keras tiene interfaces que presenta al usuario por lo que es más sencillo su uso, además les da limitadas opciones de manipulación para evitar errores y en caso de que sucedan muestra mensajes claros y amigables con el usuario, es utilizado por grandes empresas y es compatible con una alta cantidad de plataformas GPUs (IONOS. 2020).

Scikit-learn

(Fernández Jáuregui, s.f.) nos indica que esta librería con mayor utilidad en el mundo de machine learning, es de código abierto y se utiliza en el ámbito comercial y educativo, al contar con distintas funciones que permiten enfrentar problemas del aprendizaje autónomo.

Scikit-learn está construida sobre SciPy y NumPy, algunas de las funciones que pose estas funciones son: aprendizaje supervisado y no supervisado, validación cruzada, extracción de características además de contar para práctica, cuenta con el respaldo de una comunidad de desarrolladores que la mejoran y mantienen (Fernández Jáuregui, s.f.).

Plotly

Se trata de una librería de código abierto que se emplea para generar gráficos interactivos, como ejemplos de gráficos que se pueden generar tenemos histogramas, diagramas de barras, gráficos de dispersión, etc., tal cual las librerías graficas mencionadas en los párrafos previos (Plotly Open Source Graphing Library for Python, 2023).

Óptima

Es una librería para optimizar de manera automática los hiperparámetros de modelos de machine learning (Optuna: A hyperparameter optimization framework, 2023).

2.4. Fuentes de datos relacionadas al problema

Tanto (Nhu, N.Y., Van Ly, T., Truong Son, D.V., 2022) como (Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) hicieron análisis partiendo de información que se encuentra publicada en Kaggle, y (Edwine, N., Wang, W., Song, W., Ssebuggwawo, D., 2022) completo su análisis con información de una empresa que lograron conseguir, la información es bastante sensible por lo que es poco probable tener acceso a la misma a menos que uno pertenezca a la organización y cuento con la autorización indicada.

En nuestro caso si se tiene acceso a la información que se posee en los sistemas internos de la empresa, para enriquecer nuestros datos se podría utilizar información de fuentes externas, pero dependiendo de la sensibilidad puede que sea poco probable, como por ejemplo nivel de educativo, capacidad económica, tomando en cuenta que el segmento de clientes de la empresaes el residencial.

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN

3.1. Exploración y validación de datos y fuentes

3.1.1. Fase de Recolección

Como se ha indicado previamente y se muestra en la Figura 25, se tiene cuatro fuentes de datos, de las cuales tres son esquemas del ERP en base de datos Oracle y el último es una base de dato Mongo, cada una de ellas con sus particularidades.

La información no es accesible directamente, por lo que se tiene archivos en formato xls que son descargados de la herramienta de inteligencia de negocio que posee la empresa.

Los datos de estas fuentes están repartidos del siguiente modo:

Módulo de servicios: seis archivos que corresponden a información del periodo 2018 al 2020.

Módulo de soporte: sesenta archivos, uno por cada mes en el periodo 2018 la 2022.

Nivel de potencia un solo archivo.

Motivo de cancelación un archivo.

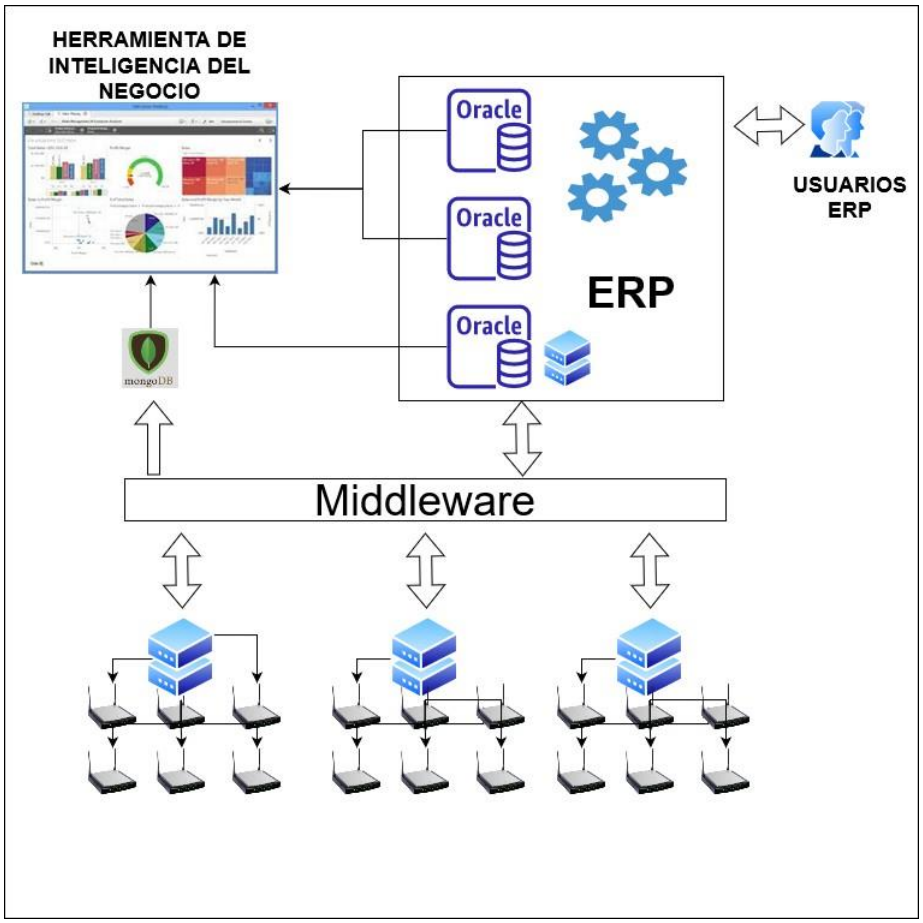


Figura 25: Disposición de la data de la empresa Xnet

3.1.2. Fase de Preprocesamiento

El tratamiento de los datos se realizará utilizando Python, e iniciamos una vez que tenemos los archivos de los diferentes módulos.

Módulo servicios

Como paso inicial partimos con la información del módulo de servicios, aquí cargamos la cada uno de los seis archivos que tenemos en un dataframe diferente.

Ahora procedemos a consolidar la información en un solo dataframe, podríamos tener datos repetidos, por ello eliminamos todos aquellos que sean iguales.

Revisamos las columnas que tenemos e identificamos, el tipo de dato paro determinar si se puede ejecutar una transformación que permita reducir el tamaño y por ende disminuir el uso de la memoria como por ejemplo datos numéricos almacenados como Int64 que por su longitud pueden ser convertidos a Int16, todo este detalle se lo ha esquematizado en forma de un flujo general de trabajo

que se muestra en la Figura 26.

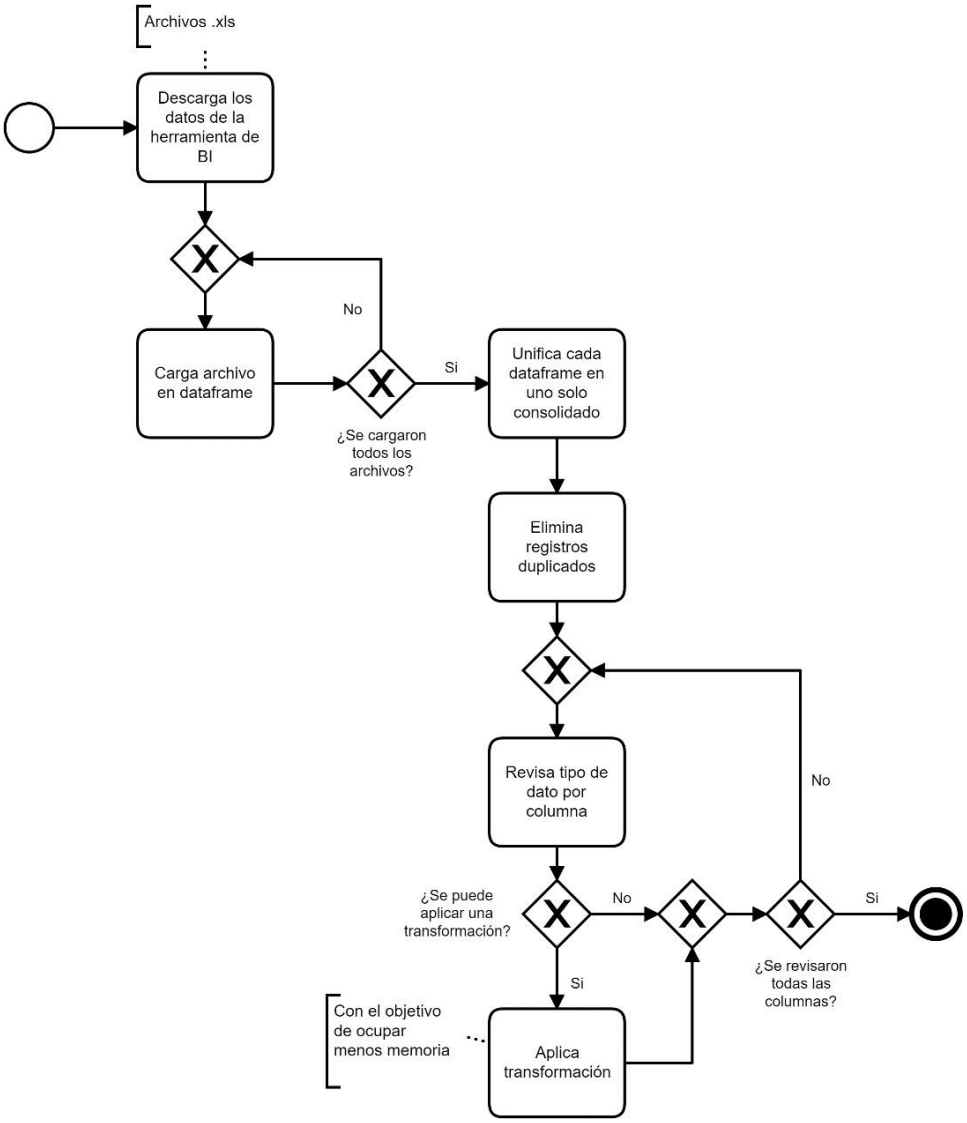


Figura 26: Flujo de preprocesamiento módulo de servicios

Módulo soporte

Ahora pasamos a trabajar con los datos del módulo de soporte, vamos a repetir el proceso realizado para el módulo de servicios utilizando el mismo flujo de trabajo mostrado en la Figura 26, la diferencia que tenemos es la cantidad de archivos, ahora son sesenta.

Ahora bien, con este dataframe final de soporte procedemos a generar más información que nos resultará valiosa, iniciamos agrupando la cantidad de tickets que tiene cada cliente.

Agregamos más información, calculamos el tiempo promedio que tomó la solucionar los problemas técnicos reportados por el cliente, este cálculo se aplica también al tiempo que le corresponde al cliente, todo esto se efectuará considerando tickets en estado cerrado (es decir solucionados).

Nivel de potencia

Con respecto a los datos del nivel de potencia, únicamente se posee un archivo por lo que el proceso es más corto que el descrito anteriormente.

Descargamos los datos desde la herramienta de inteligencia del negocio y lo cargamos en un dataframe, revisamos el tipo de dato de cada columna para verificar si se puede aplicar alguna transformación que permita disminuir la memoria utilizada, todo este detalle se lo ha esquematizado en forma de un flujo general de trabajo que se muestra en la Figura 27.

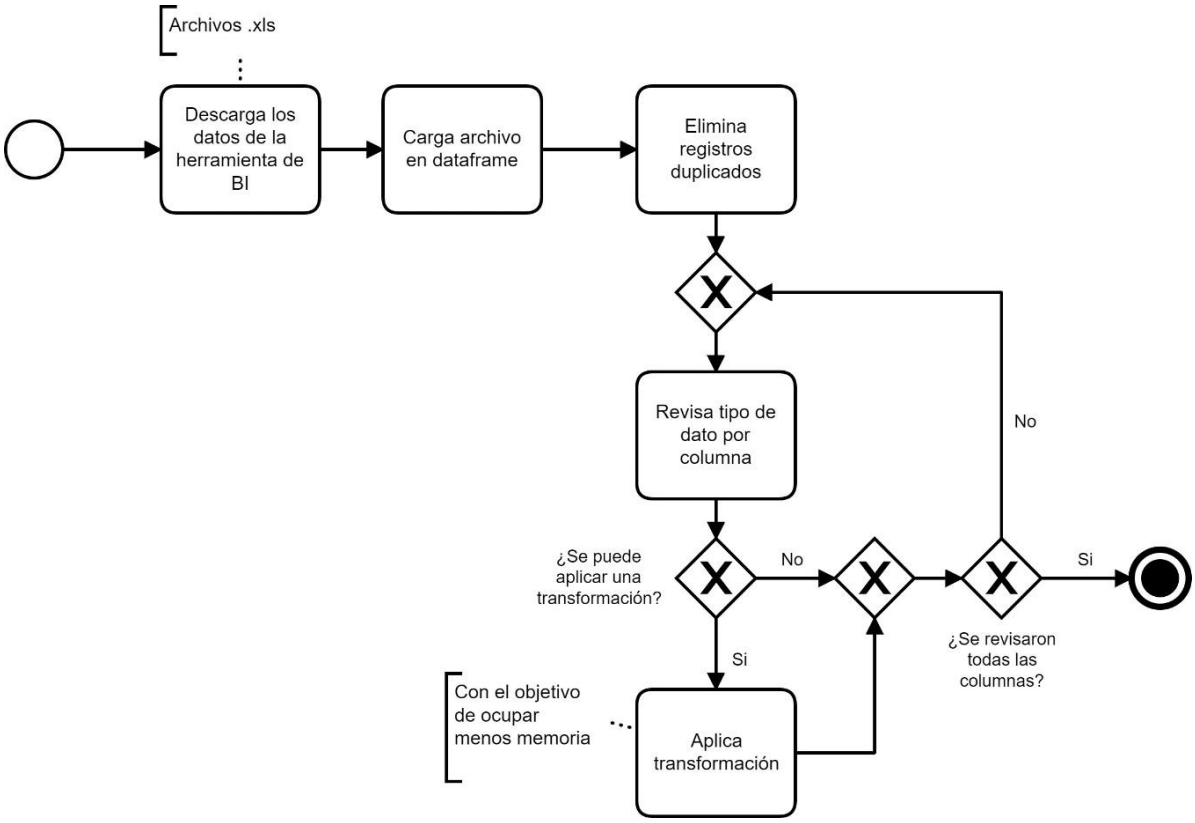


Figura 27: Flujo de preprocesamiento nivel de potencia

Motivo de cancelación

Es momento de trabajamos con los datos del motivo de cancelación, para esto utilizaremos el mismo flujo de trabajo mostrado en la Figura 27, al igual que en niveles de potencia, al tener un solo documento.

Para finalizar esta fase unificamos en un solo dataframe utilizando como campo común el login del cliente, el dataframe final de servicios, el dataframe final de soporte, el dataframe de nivel de

potencia y el dataframe de motivo de cancelaciones, el resultado es la base con la que

continuara el proceso, todo este detalle se lo ha esquematizado en forma de un flujo general detrabajo que se muestra en la Figura 28.

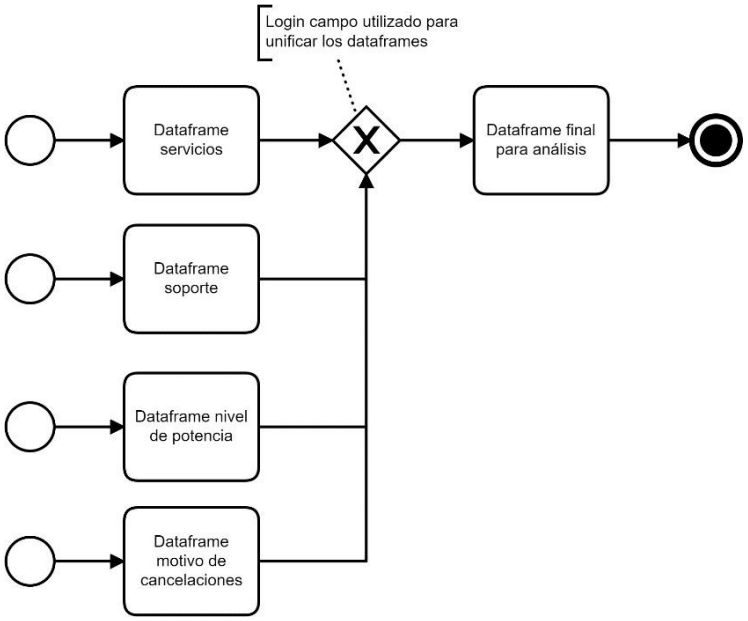


Figura 28: Flujo de preprocesamiento modulo cancelaciones

3.1.3. Fase de Limpieza de datos

Ahora que tenemos una sola base sobre la cual trabajaremos, iniciamos el proceso de limpieza, esto es fundamental para evitar tener ruido que afecte el funcionamiento del modelo generado, para ello ejecutaremos lo siguiente:

1. Estandarizamos los nombres de las columnas reemplazando espacios en blanco, puntos, paréntesis por guion bajo.
2. Eliminamos los datos duplicados que se pudieron generar en la fase de preprocesamientoal combinar los diferentes dataframes
3. Verificamos valores únicos por columna para evitar errores tipográficos que puedan generar clases erróneas

Debemos revisar en cada columna si existen valores faltantes y definir qué hacer con ellos. En nuestro caso un valor faltante que podría ser importante es la latitud y longitud en la que se encuentra el cliente, afortunadamente tenemos como solventarlo al existir el campo zona que es la agrupación más pequeña dentro de la ciudad definida por la empresa, por lo que podemos prescindir de ellos y

eliminarlos del dataframe de ser necesario.

Otro campo importante que tiene espacios vacíos es la fecha de activación (año, mes y/o día), con estos campos se puede calcular la antigüedad del cliente, en este caso no tenemos otra opción que eliminar a aquellos que poseen el campo vacío. Se elimina también los registros que no poseen detalles de su forma de pago, estado_servicio, estado_ticket y tiempo de solución, debido a que no es posible generar datos que los completen.

Para el caso de dbm, de define que si el campo está vacío se calcula el promedio de aquellos que se encuentran en el mismo sector, todo este detalle se lo ha esquematizado en forma de un flujo general de trabajo que se muestra en la Figura 29.

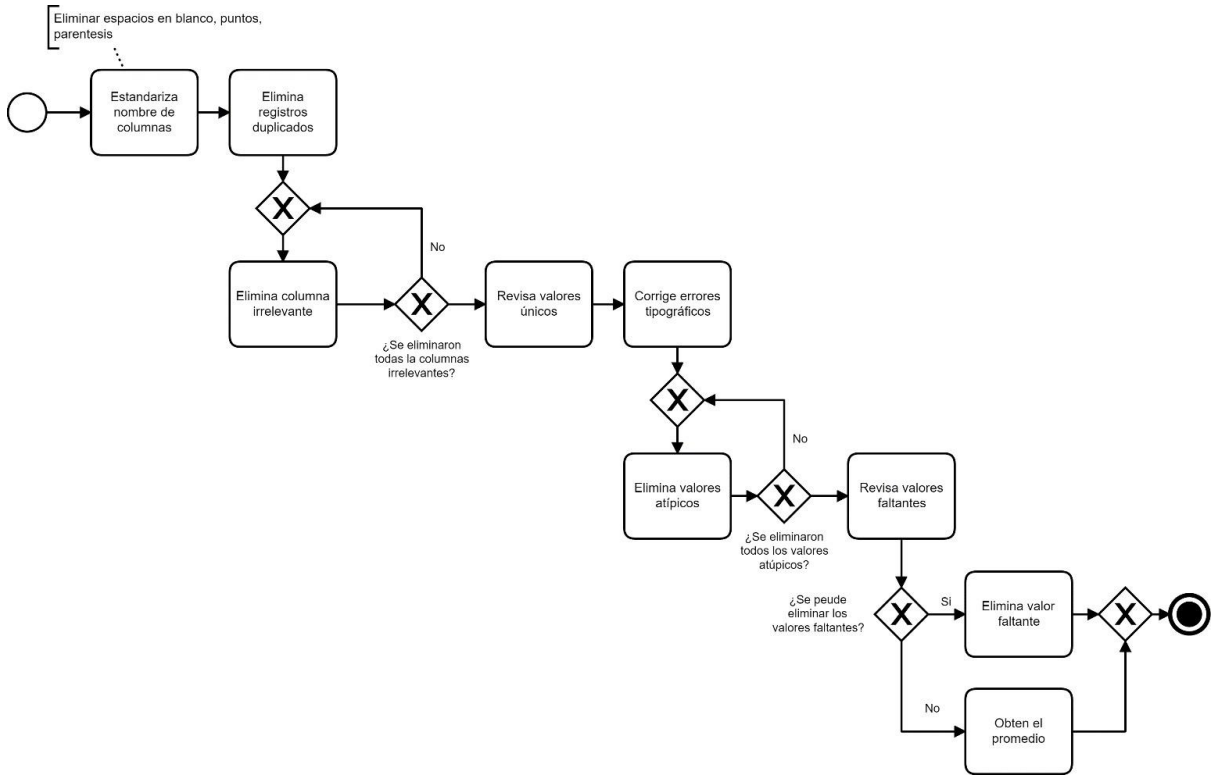


Figura 29: Flujo de limpieza de datos

3.1.4. Fase de Análisis e interpretación de los datos

Con el dataframe limpio, procedemos a explorar los datos para entenderlos.Iniciamos revisando la preferencia de pago de los clientes, aquí se identifica que el débito bancario es la forma preferida para el pago de servicios contratados con un 92% del total, esto tiene sentido ya que la empresa constantemente lanza promociones para captar clientes con esta forma de pago, este detalle lo podemos ver en la Figura 30.

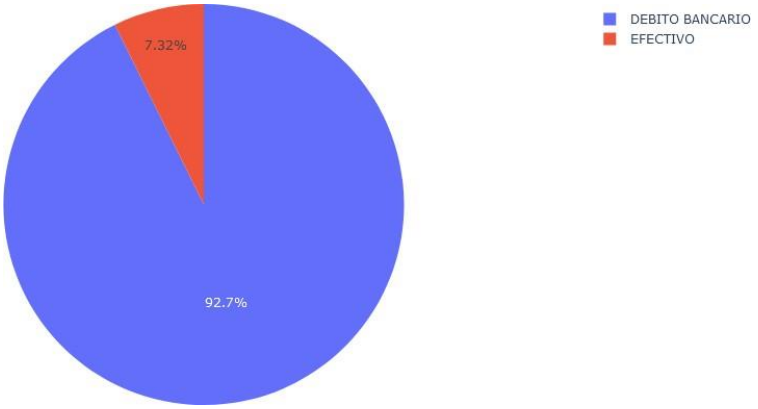


Figura 30: Resumen forma de pago

Ahora si exploramos la forma de pago débito, podemos identificar que se tienen setenta y siete instrucciones financieras, esto se observa en la Figura 31. En la Figura 32 podemos ver un top diez de los bancos, aquí se verifica que la mayoría de las cuentas corresponde a banco Pichincha y como segundo más grande a banco Pacífico y como tercero el subgrupo llamado tarjetas, estos datos son valiosos para que el modelo determine qué tan relacionada está esta característica con la posibilidad de cancelación de servicios.

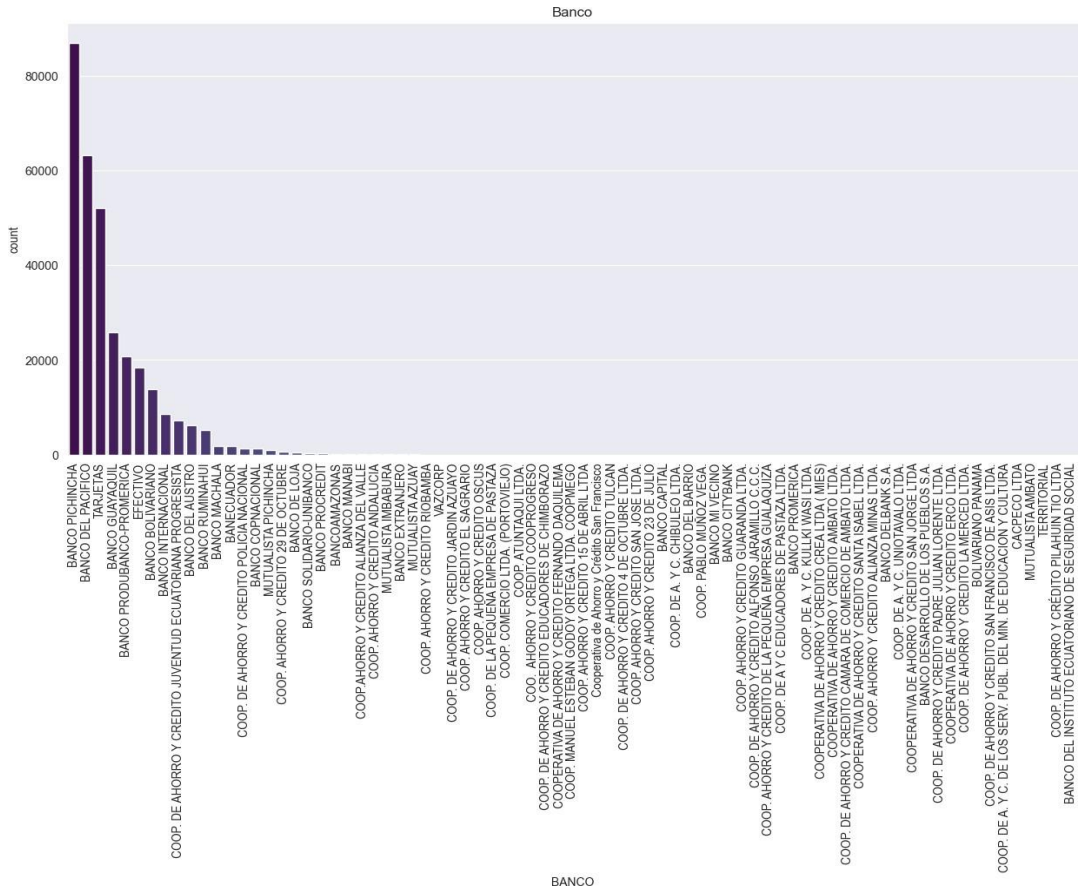


Figura 31: Detalle bancos desde el que pagan los clientes

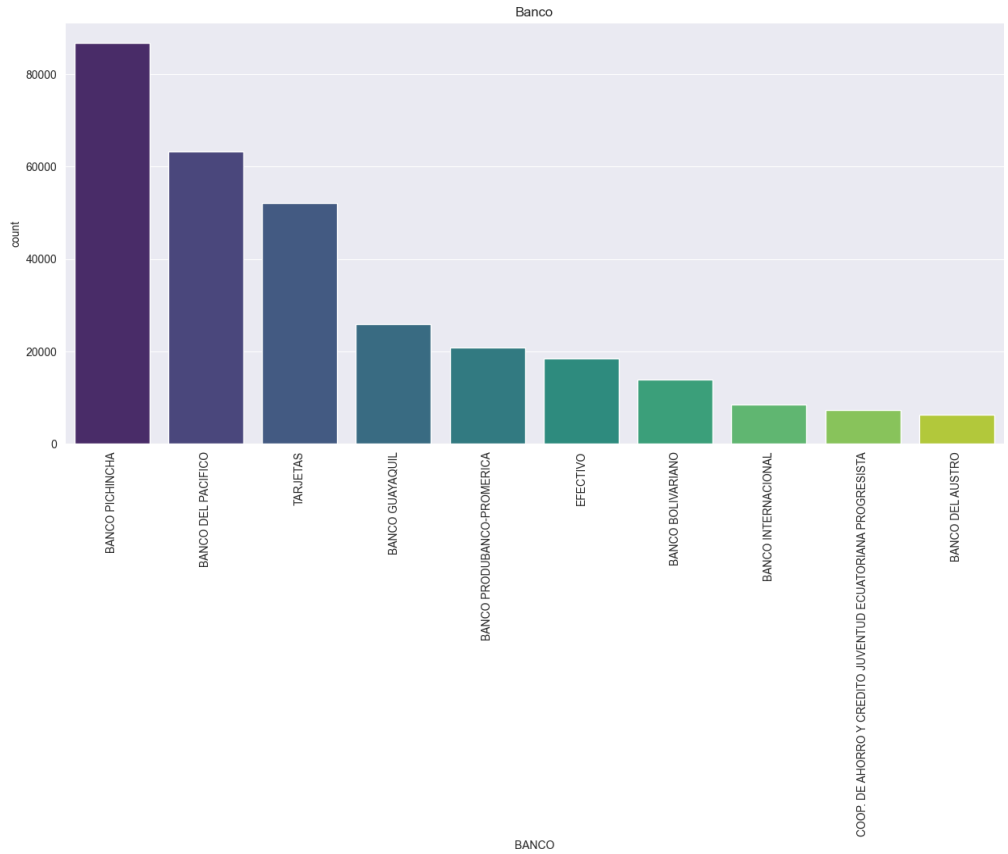


Figura 32: Top 10 de bancos más utilizados por los clientes para pagar sus servicios

Si descendemos un nivel más podemos ver que el 58.7% de los débitos corresponde a las tarjetas Diners, seguidos por American con 21.4 y Discover en tercer lugar con 10.1%, esto lo podemos observar en la Figura 33

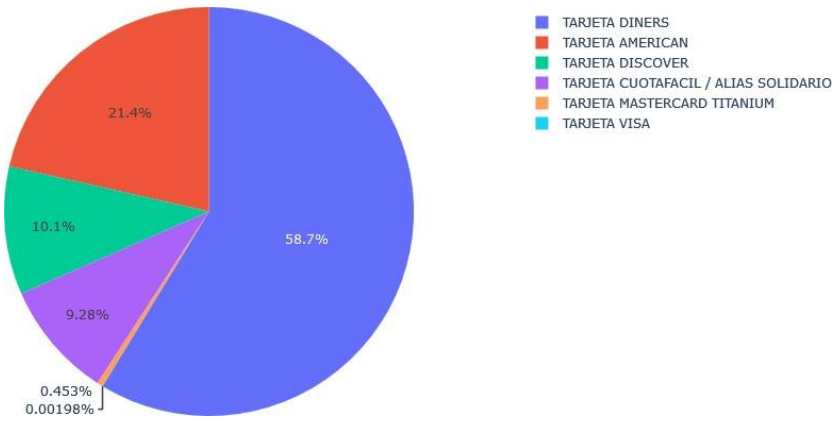
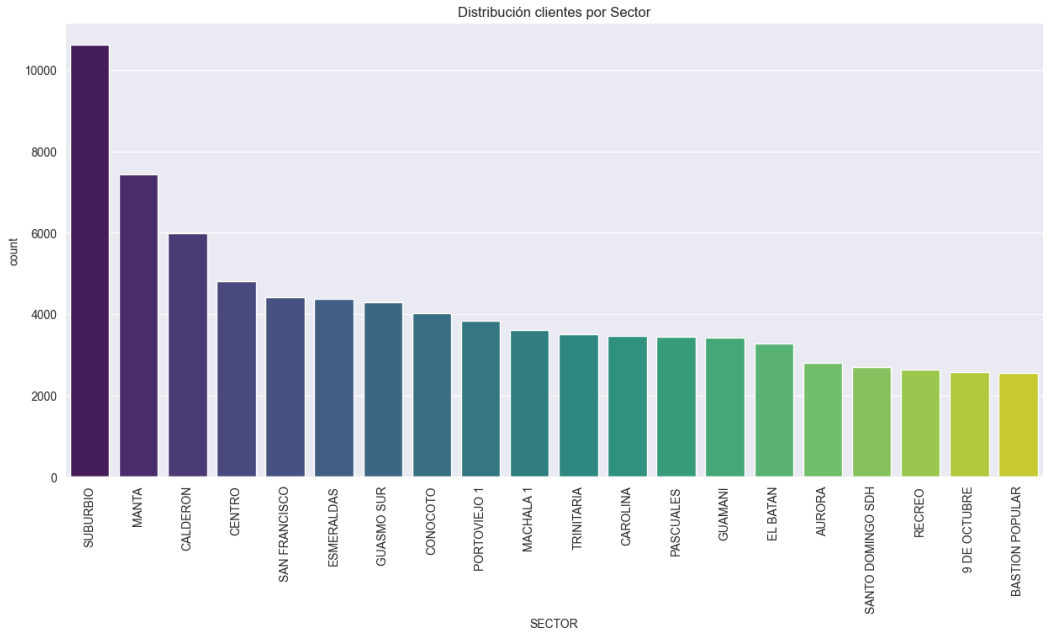


Figura 33: Distribución de tarjetas de crédito con las que pagan los clientes

Si revisamos la distribución de los clientes en los diferentes sectores se puede observar que su concentración principal se encuentra en aquellas que corresponde a Guayas y luego seguidas por las que pertenecen a Manta y Pichincha, este detalle se muestra en la Figura 34,

bajo este contexto se entendería que en estos dos lugares se pueden concentrar los esfuerzos debido a que su impacto sería sobre un gran grupo de clientes, esto se lo validaría con la ejecución del modelo.

Figura 34: Concentración de clientes por sector



Analizando el nivel de potencia podemos observar en la Figura 35 que la gran mayoría de clientes tienen un nivel de potencia adecuado (entre -20 a -24 dbms), hay un grupo prominente que presenta el valor 1234 que hacer referencia a que el equipo estaba apagado en el momento de la toma de potencia.

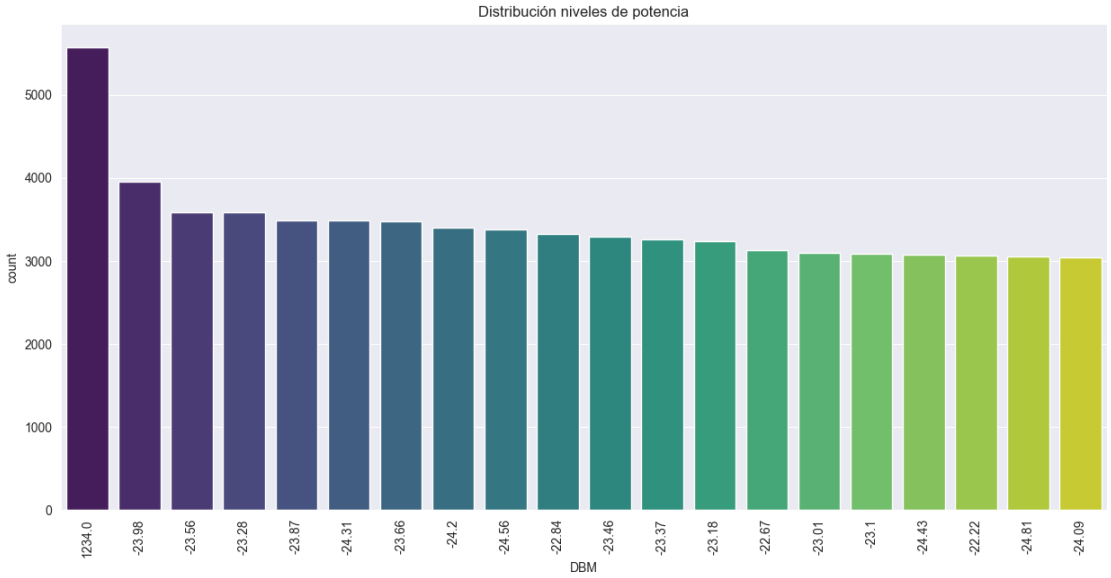
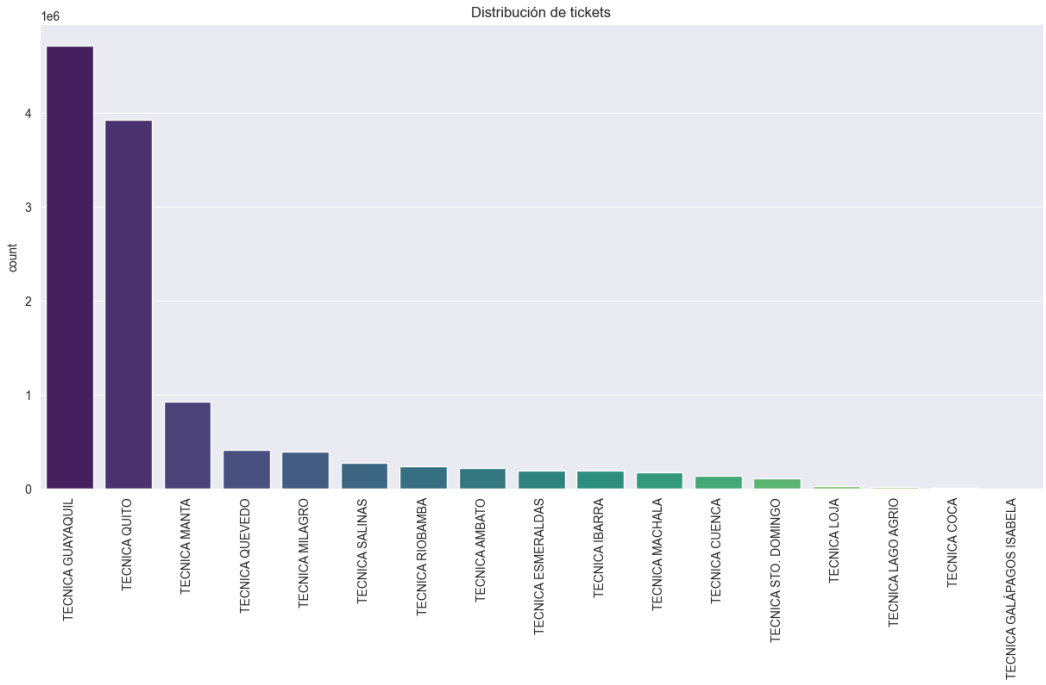


Figura 35: Agrupación de clientes por nivel de potencia

La distribución de los tickets que muestra la Figura 36, se ve que la mayor concentración se tiene en el cantón Quito, seguido por Guayaquil y Portoviejo, esto apoya en parte a lo que se mencionó con



respecto a la Figura 34 sobre la posible priorización de estas ubicaciones.

Figura 36: Distribución de tickets por oficina técnica

En la figura 37 se puede observar la distribución de tickets por jurisdicción y año, aquí resalta el comportamiento distinto entre jurisdicciones mientras unas tienen una tendencia creciente como Quito, Guayaquil o Ibarra, otras aumentan y disminuyen a lo largo del tiempo como es el caso de Manta, Quevedo o Riobamba.

La mayor concentración de tickets esta en Quito y Guayaquil, en el caso de Manta se ve un pico en el 2020 y luego de lo cual empezó una disminución.

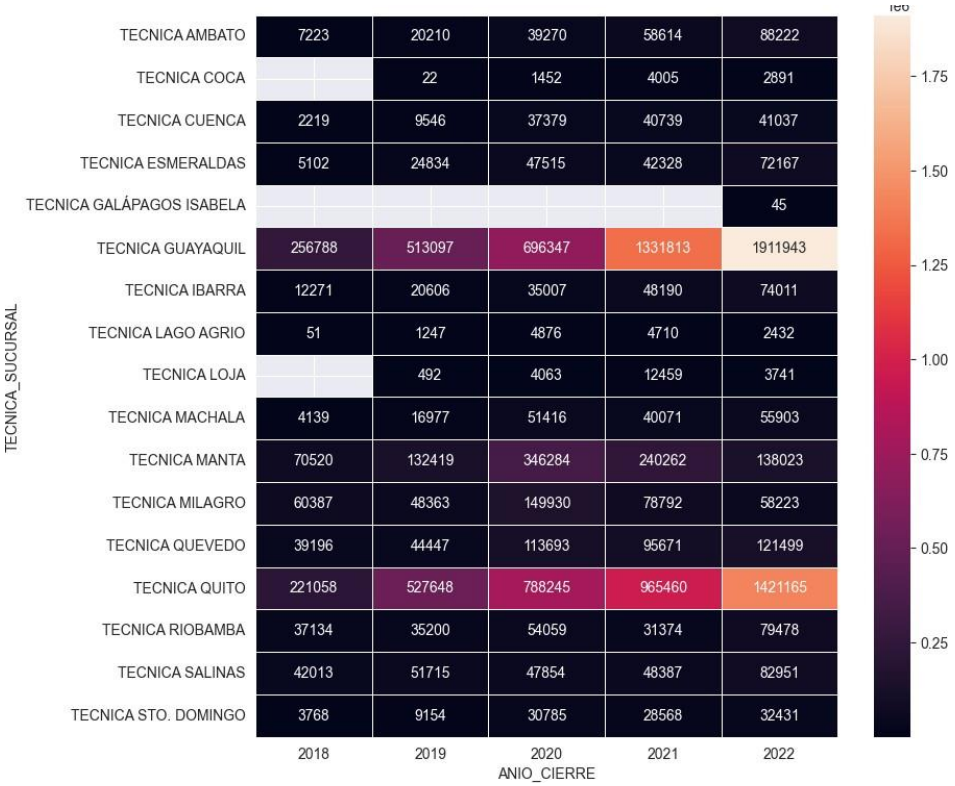


Figura 37: Distribución de tickets por jurisdicción y por año

3.2. Prototipos de algoritmos, modelos, y módulos del sistema

Prototipos de algoritmos y modelos

Para poder identificar a los clientes que tienen la intención de cancelar sus servicios se ha definido que los modelos que se utilizan y se comparan sus resultados son codificador automático, máquinas de vectores de soporte, bosques de asilamiento y xgboost, a continuación, explicaremos cómo funcionan estos algoritmos.

Codificador automático

Es un algoritmo de aprendizaje automático que se utilizan generalmente para compresiónde datos y representación, además de encontrar variables latentes (aquellas que a simplementeno se las identifica, pero el modelo las encuentra a partir de otras variables), (Significado, s.f.). **Funcionamiento**

En la Figura 38 podemos observar que se trata de una arquitectura de red neuronal artificial, que consta de un codificador encargado de transformar los datos de entrada en una representación comprimida (espacio latente), que tiene la información esencial de los datos

ingresados, con esto el decodificador toma esta información e intenta reconstruir los datos originales, entre más precisa es la reconstrucción podemos decir que el codificador automático mejor captura las características de los datos de entrada (Significado, s.f.).

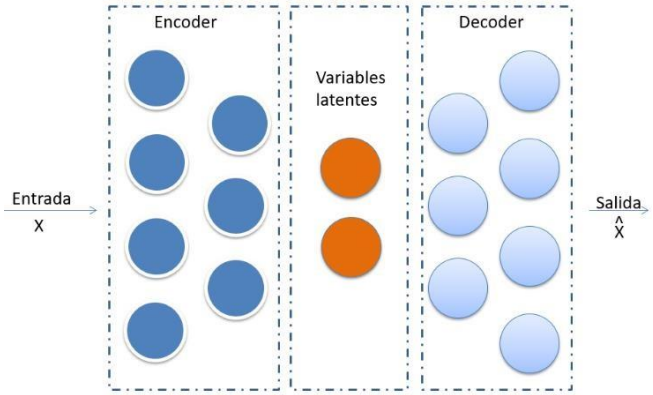


Figura 38: Codificador automático

Configuraciones

Para este tipo de algoritmos generalmente se configuran cuatro parámetros, iniciamos como el número de niveles que está directamente relacionados a la complejidad de los datos, como segundo parámetro tenemos el número de nodos por cada nivel tomando en cuenta que el codificador/decodificador tiene el mismo tamaño, el tercer parámetro las colas (número de nodos intermedios) que son los responsables del nivel de compresión y finalmente la función de perdida (Innovaciondigital360, 2023).

Ventajas

Dentro de las principales ventajas que ofrece el codificador automático, tenemos la extracción de características relevantes que permiten reducir dimensiones y eliminar el ruido (Innovaciondigital360, 2023).

Desventajas

Los codificadores automáticos también presentan desventajas, entre las cuales tenemos por ejemplo el sobreajuste al no ser configurado adecuadamente, generando salidas deficientes, otro problema que se suele presentar es la dificultad para capturar características de datos complejos lo que también afecta el funcionamiento, si el algoritmo no logra modelar los datos la reconstrucción de la data será limitada o ineficiente (Amat Joaquín, 2021).

Máquina de vectores de soporte de una sola clase

Funcionamiento

Este es un tipo de algoritmo que intenta separar datos normales de aquellos atípicos utilizando hiperplanos que máxime la distancia entre cada clase (IBM, 2021), en la Figura 39, podemos observar la representación de este trabajo en el que tenemos dos clases y se genera el límite máximo que separa las clases, para este ejemplo es una línea recta pero dependiendo de la complejidad podría ser un polinomio de grado N, pero entre mayor en N puede que caigamos en sobreajustes, es decir que el algoritmo solo funcione con las datos actuales pero no funcione de manera adecuada para nuevos datos.

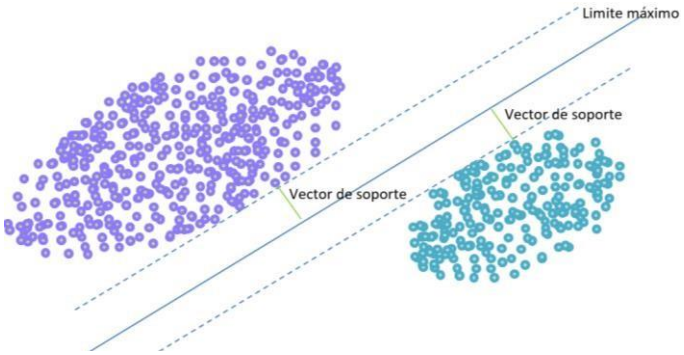


Figura 39: Maquina de vectores de soporte

Configuraciones

Lo principal para este tipo de algoritmos es la selección del kernel encargado de encontrar la mejor forma de separar los datos, además de los parámetros de tolerancia y coeficientes de regulación que serán los responsables de que el modelo tenga equilibrio al separar las clases, contaminación nu (Configuración del modelo SVM, 2023).

Los kernels son funciones que por medio de funciones complejas transforman un espacio con pocas dimensiones en un espacio con más dimensiones (Martínez C, 2018), los kernels comúnmente más utiliza son:

Kernel lineal verifica la similitud utilizando la dependencia lineal entre dos valores, obteniendo un clasificador (Martínez C, 2018).

Kernel polinómico con un grado mayor a 1 nos da un límite de decisión más flexible (Martínez C, 2018).

Kernel radial mayor flexibilidad que el polinómico y en el que las observaciones más cercanas influyen en la clasificación, aquí se tiene el parámetro gamma que determina como calcular la importancia de las fronteras, si por el número de características (scale) o por el número de observaciones (auto) (Martínez C, 2018)

Ventajas

Funciona bien con grandes dimensiones, optimizando la memoria requerida al trabajar con subconjunto de puntos de entrenamiento, además ese puede utilizar kernels genéricos o personalizados (Support vector machines, s.f.).

Desventajas

El desempeño está directamente relacionado con el kernel seleccionado, además de ser lento cuando se tiene enormes cantidades de datos, y requerir que los datos estén escalados para que su desempeño sea optimo (Ventajas y desventajas de SVM, s.f.).

Bosques de aislamiento

Funcionamiento

Inspirado en algoritmos de clasificación y regresión, en la Figura 40 podemos observar que está formado por múltiples árboles, en las que cada observación que se considera una anomalía quedará aislada rápidamente de un nodo terminal, basándose en el hecho de que las anomalías son raras y toman menos divisiones para ser identificadas (Amat Joaquín, 2020).

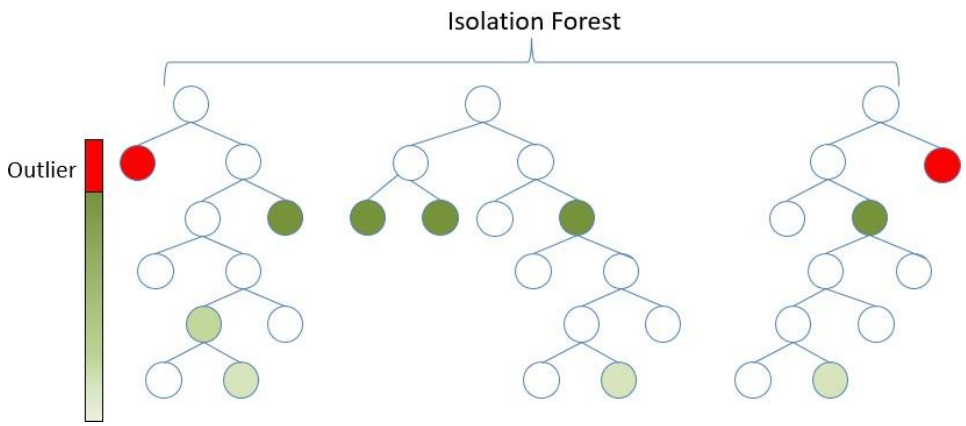


Figura 40: Bosque de aislamiento

Configuraciones

Dentro de los parámetros que se deben configurar tenemos la cantidad de árboles que tendrá el bosque (n_estimators), cantidad de muestras para cada árbol (max_samples), contaminación que tiene el ser de datos (anomalías) y por cantidad de características que se utilizaran (max_features) (Isolation Forest en Python, s.f.).

Ventajas

Adecuado para set de datos de gran cantidad de dimensiones, al utilizar sub-sampling trabaja muy rápido, efectivo con set de datos enormes y se desempeña bien a pesar de que los datos anómalos y no anómalos estén muy cercanos (Isolation Forest en Python, s.f.).

Desventajas

El ruido en el set de datos lo puede afectar si no se aplica un correcto tuning, al ser basado en árboles puede ser difícil de interpretar (Isolation Forest en Python, s.f.).

Xgboost

Funcionamiento

Es uno de los algoritmos más utilizados hoy en día, es fácilmente implementable, da buenos resultados y funciona de manera paralelizada lo que le da rapidez de entrenamiento (Cómo funciona el algoritmo Xgboost en Python, s.f.).

Se basa en bosques aleatorios y en el refuerzo del gradiente, donde se crean múltiples árboles uno a continuación del otro para corregir los errores del árbol previo (Cómo funciona el algoritmo XGBoost, s.f.), en la Figura 41 tenemos la representación del funcionamiento, en el que se tiene data de entrada de la que se generan las submuestras y se las pasa a los diferentes modelos generados y se van calculado los diferentes puntajes hasta conseguir el que entregue el mejor resultado.

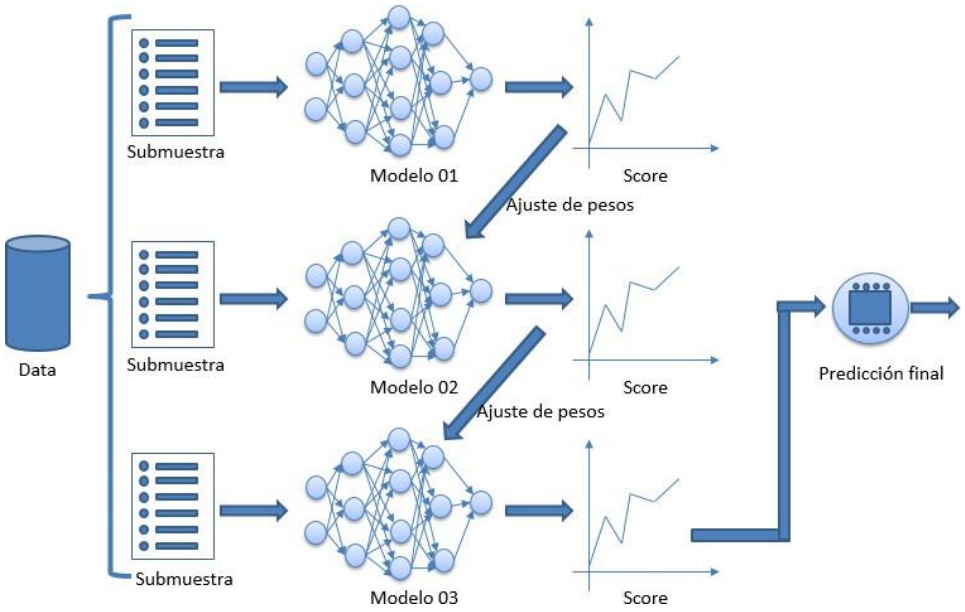


Figura 41: XGBoost

Configuraciones

Dentro de las configuraciones generales que se aplican al algoritmo son la tasa de aprendizaje (laerning_rate) que nos indica que tanto se ajusta el modelo, la cantidad de árboles (n_stimators),la profundidad máxima de los árboles (max_depth), si se utiliza o no GPUs, el peso mínimo por nodo para ser dividido (min_child_weight), reducción de la ganancia que determina las particiones (gamma), el subconjunto para entrenar cada árbol (subsample), la cantidad de características por árbol (colsample_bytree), método de construcción del árbol (tree_method) y la regularización L1 (reg_alpha) y L2 (reg_lambda). (XGBoost Parameters, s.f.), dentro del método para construir los árboles el propio algoritmo lo elije en base a las características (auto), se puede tomar las posibles divisiones (exact), en base a histogramas (hist) o busca un equilibrio entre que tan preciso es vs su rendimiento (approx) (Hiperparámetros de XGboost, s.f.).

Ventajas

Como ventajas del algoritmo podemos mencionar que puede trabajar con enormes bases de datos con gran cantidad de características, esta implementado en diferentes leguajes como R o Python, además funciona bien con valores perdidos sin imputaciones, incorpora el paso de la regularización para evitar sobreajustes (Rodríguez Yeifer, 2018).

Desventajas

Se debe poner mucho énfasis en la correcta configuración de los parámetros, utiliza grandes cantidades de memoria por lo que es común tener que seleccionar primero las características relevantes antes de empezar con el modelo y de requerir trabajar con todas las características se necesitará equipos realmente robustos (Rodríguez Yeifer, 2018).

Ahora que hemos dado un vistazo rápido para conocer estos algoritmos, es hora de pasar a la práctica aplicando los diferentes modelos descritos anteriormente.

3.2.1. Prototipos de algoritmos y modelos

Iniciamos escalamos los valores del set final de datos, esto porque los modelos trabajan de mejor manera cunado las escalas en la que se encuentran el conjunto de datos es similares y reduce el riesgo de tener valores atípicos, esto se ve en la Figura 42 un ejemplo de cómo se ven los datos de las diferentes columnas como son: ESTADO_SERVICIO, PLAN, ES_VENTA, COMPORTAMIENTO_PAGO, etc.

ESTADO_SERVICIO	PLAN	ES_VENTA	PRECIO_VENTA	PORCENTAJE_DESCUENTO	VALOR_DESCUENTO	FRECUENCIA_PRODUCTO	...	COMPORTAMIENTO_PAGO	
-0.733111	0.885984	0.030057	-0.429592	-0.002506	-0.227980	-0.003544	...	-0.683350	(
-0.733111	0.729674	0.030057	-0.429592	-0.002506	-0.227980	-0.003544	...	-0.683350	(
1.364050	0.346005	0.030057	0.196751	-0.002506	-0.227980	-0.003544	...	-0.683350	(
-0.733111	0.729674	0.030057	-0.429592	-0.002506	-0.227980	-0.003544	...	-0.683350	(
-0.733111	0.346005	0.030057	0.196751	-0.002506	-0.227980	-0.003544	...	-0.683350	(
...	...	...	...	...	...	...	...	...	
-0.733111	0.836249	0.030057	0.286228	-0.002506	-0.227980	-0.003544	...	2.474266	(
-0.733111	0.729674	0.030057	-0.429592	-0.002506	2.351465	-0.003544	...	2.474266	(
-0.733111	0.729674	0.030057	-0.429592	-0.002506	-0.227980	-0.003544	...	2.474266	(
-0.733111	0.637310	0.030057	-1.055934	-0.002506	-0.227980	-0.003544	...	2.474266	(
-0.733111	0.459685	0.030057	-0.429592	-0.002506	-0.227980	-0.003544	...	2.474266	(

Figura 42: Dataset codificado y normalizado

Verificando la matriz de correlaciones de las diferentes columnas Figuras 43 y 44, lo que se ve es que temas de ubicación, región y sucursal que son geográficos presentan una alta correlación, se ve también relación entre los tickets y la tecnología en la que se ha desplegado el cliente, otro dato que resalta es que el estado del servicio tiene una alta correlación con la antigüedad del cliente, resultando más probable que clientes antiguos tengan propensión a cancelar sus servicios.

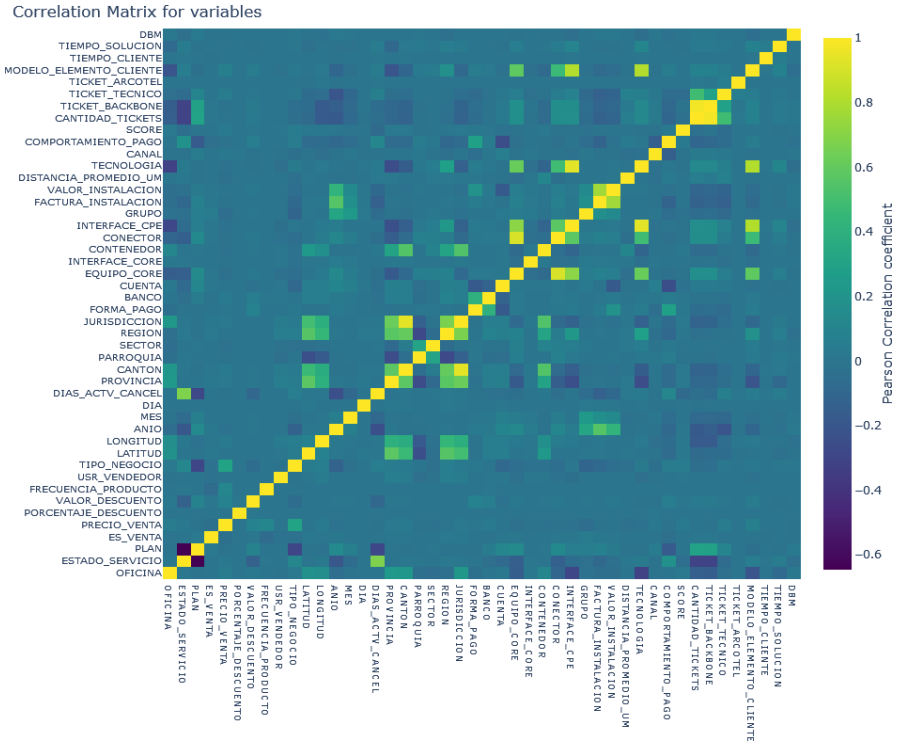


Figura 43: Matiz de correlación de variables

	LATITUD	LONGITUD	ANIO	MES	DIA	DIAS_ACTV_CANCEL	PROVINCIA	CANTON	PARROQUIA	SECTOR	REGION	JURISDICCION	FORMA_PA
	0.184293	0.161674	-0.057936	-0.022364	0.001290	-0.000644	0.231537	0.218926	-0.038977	-0.056130	-0.043881	0.222672	-0.037
	0.004208	0.014373	-0.189510	-0.068182	-0.004910	0.677583	-0.007996	0.006593	0.008435	-0.003547	0.023379	0.008879	-0.003
	-0.013295	-0.025135	0.139545	0.064335	0.004115	-0.288378	-0.001404	-0.010766	-0.008760	0.009724	-0.016864	-0.012032	-0.015
	0.012565	0.010077	0.014133	0.003811	-0.001195	0.004270	0.013602	0.010412	-0.001880	0.002951	0.020311	0.015125	-0.003
	-0.001511	0.032377	-0.025039	0.001455	0.001840	-0.011520	0.022109	0.029123	-0.009726	-0.024453	-0.006707	0.021480	0.006
	0.002157	0.001834	-0.002107	0.000985	0.003712	0.002067	0.002511	0.002361	0.000099	-0.001875	0.001830	0.002413	-0.000
	0.004511	0.004561	0.006719	0.008349	0.003546	-0.104605	0.005688	0.006548	-0.004833	-0.001378	0.003784	0.002936	0.082
	0.001498	0.001514	-0.000807	-0.001329	-0.001621	-0.000267	0.001966	0.003712	0.000538	0.001608	0.000727	0.003610	0.002
	0.008468	0.036706	-0.003066	-0.005515	-0.001298	0.000795	0.058329	0.040036	-0.039120	-0.022683	0.052241	0.042232	-0.003
	0.037648	0.052201	-0.124985	-0.033798	0.001626	0.050584	0.034177	0.030202	-0.004471	-0.031997	0.048516	0.032302	0.013
	1.000000	0.086921	0.011182	0.002175	0.012832	0.005330	0.572729	0.467633	-0.238599	-0.032205	0.557288	0.497203	-0.016
	0.086921	1.000000	0.014267	0.005325	0.003944	0.005259	0.391622	0.367630	-0.172722	-0.028381	0.447101	0.383533	-0.019
	0.011182	0.014267	1.000000	0.063624	-0.006925	-0.243143	0.008339	-0.014013	-0.009326	-0.002145	0.011452	-0.008866	0.030
	0.002175	0.005325	0.063624	1.000000	-0.003853	-0.092095	0.000244	-0.010016	-0.006299	-0.004686	-0.006122	-0.008619	0.007
	0.012832	0.003944	-0.006925	-0.003853	1.000000	-0.003949	0.015117	0.011078	-0.000272	-0.001352	0.022180	0.011817	-0.003
	0.005330	0.005259	-0.243143	-0.092095	-0.003949	1.000000	-0.007480	0.011332	0.004127	0.004901	0.024622	0.014055	-0.017
	0.572729	0.391622	0.008339	0.000244	0.015117	-0.007480	1.000000	0.574395	-0.256474	-0.082742	0.568112	0.617060	-0.023
	0.467633	0.367630	-0.014013	-0.010016	0.011078	0.011332	0.574395	1.000000	-0.053271	0.033578	0.601984	0.936226	-0.008
	-0.238599	-0.172722	-0.009326	-0.006299	-0.000272	0.004127	-0.256474	-0.053271	1.000000	0.308396	-0.279549	-0.136020	0.010
	-0.032205	-0.028381	-0.002145	-0.004686	-0.001352	0.004901	-0.082742	0.033578	0.308396	1.000000	-0.058920	-0.009658	0.009
	0.557288	0.447101	0.011452	-0.006122	0.022180	0.024622	0.568112	0.601984	-0.279549	-0.058920	1.000000	0.666058	-0.003
	0.497203	0.383533	-0.008866	-0.008619	0.011817	0.014055	0.617060	0.936226	-0.136020	-0.009658	0.666058	1.000000	-0.009
	-0.016350	-0.019219	0.030833	0.007340	-0.003707	-0.017419	-0.023557	-0.008400	0.010514	0.009819	-0.003649	-0.009329	1.000
	0.047647	0.053196	0.026783	0.016297	0.004092	-0.029858	0.051934	0.059084	-0.037992	-0.012403	0.082258	0.060281	0.399
	0.049772	0.051128	0.102193	0.063173	0.005580	-0.033723	0.062612	0.051230	-0.042440	-0.010381	0.066991	0.051020	-0.178
	-0.005613	0.029866	0.122833	0.038644	0.001758	-0.038145	-0.178762	-0.084027	0.061058	0.064012	0.078623	-0.085583	0.020
	0.011149	0.011355	0.046752	0.022144	0.001177	-0.007139	0.013044	0.018043	0.000206	-0.001434	-0.006045	0.018092	-0.001
	0.231803	0.171998	-0.010065	-0.014678	0.003710	0.013883	0.300470	0.548467	0.003115	0.094417	0.309146	0.541142	0.003
	-0.003601	0.036918	0.117508	0.038873	0.000292	-0.045964	-0.155248	-0.067806	0.074169	0.065194	0.041252	-0.078936	0.014
	-0.003743	0.030184	0.040179	0.000134	0.004405	-0.000663	-0.242922	-0.149094	0.022431	0.047368	0.225835	-0.129295	0.031

Figura 44: Matriz de correlación de variables con detalle numérico

Se ha decidido tomar el set de datos y verificar la importancia de las diferentescaracterísticas, este proceso se repite con percentiles 10, 20 y 30, sobre los sets de datos

resultantes, se filtra los archivos por la columna comportamiento de pago debido a que, a pesar de no tener registros vacíos se tiene como dato “Sin categoría” como resultado final se obtiene 8 sets de datos para entrenar los modelos como se observa en la Figura 45, con las siguientes características:

- Datos sin filtrar (318440 registros)
    - dfFinalModelosBinarios con 45 columnas
    - dfFinalModelosBinarios_10 con 40 columnas
    - dfFinalModelosBinarios_20 con 36 columnas
    - dfFinalModelosBinarios_30 con 31 columnas
- Datos filtrados (50756 registros).
    - dfFinalModelosFiltradoBinarios con 45 columnas
    - dfFinalModelosFiltradoBinarios_10 con 40 columnas
    - dfFinalModelosFiltradoBinarios_20 con 36 columnas
    - dfFinalModelosFiltradoBinarios_30 con 31 columnas

```
dfFinal = pd.read_excel('dfFinalModelosBinarios.xlsx')
dfFinal_10 = pd.read_excel('dfFinalModelosBinarios_10.xlsx')
dfFinal_20 = pd.read_excel('dfFinalModelosBinarios_20.xlsx')
dfFinal_30 = pd.read_excel('dfFinalModelosBinarios_30.xlsx')

dfFinal_ftr = pd.read_excel('dfFinalModelosFiltradoBinarios.xlsx')
dfFinal_ftr_10 = pd.read_excel('dfFinalModelosFiltradoBinarios_10.xlsx')
dfFinal_ftr_20 = pd.read_excel('dfFinalModelosFiltradoBinarios_20.xlsx')
dfFinal_ftr_30 = pd.read_excel('dfFinalModelosFiltradoBinarios_30.xlsx')
```

Figura 45: Sets de datos para la predicción

3.2.1.1. Codificar automático (autoencoder)

Como paso inicial se define el uso de optuna como el optimizador de hiperparámetros, en la Figura 46 observamos como se ve la función utilizada para la optimización, se destaca temas como la cantidad de capas, la cantidad de neuronas por capa, cantidad de neuronas que se apagan y las épocas con las que se realizará el proceso de optimización.

```python
def objective(trial):
    dim_entrada = X_train.shape[1]
    capa_entrada = Input(shape=(dim_entrada,))

    encoder = capa_entrada
    decoder = capa_entrada

    # Definir las capas del encoder
    num_layers = trial.suggest_int('num_layers', 2, 5)
    units_list = []
    dropout_list = []
    for i in range(num_layers):
        units = trial.suggest_int('units_encoder_{}'.format(i), 10, 50)
        dropout = trial.suggest_float('dropout_encoder_{}'.format(i), 0.1, 0.5)
        encoder = Dense(units, activation='tanh')(encoder)
        encoder = Dropout(dropout)(encoder)
        units_list.append(units)
        dropout_list.append(dropout)

    # Definir las capas del decoder
    for i in range(num_layers):
        decoder = Dense(units_list[num_layers - i - 1], activation='tanh')(decoder)
        decoder = Dropout(dropout_list[num_layers - i - 1])(decoder)
    decoder = Dense(dim_entrada, activation='sigmoid')(decoder)

    autoencoder = Model(inputs=capa_entrada, outputs=decoder, name="autoencoder")
    autoencoder.compile(optimizer='adam', loss='mae', metrics=['accuracy'])

    epochs = trial.suggest_int('epochs', 50, 200)
    batch_size = trial.suggest_categorical('batch_size', [32, 64,128,256,512])

    early_stopping = EarlyStopping(monitor='val_accuracy', patience=10, mode='max')

    historia = autoencoder.fit(X_train, X_train, epochs=epochs, batch_size=batch_size, shuffle=True,
                               validation_data=(X_test, X_test), verbose=0, callbacks=[early_stopping])

    return -max(historia.history['val_accuracy'])
```

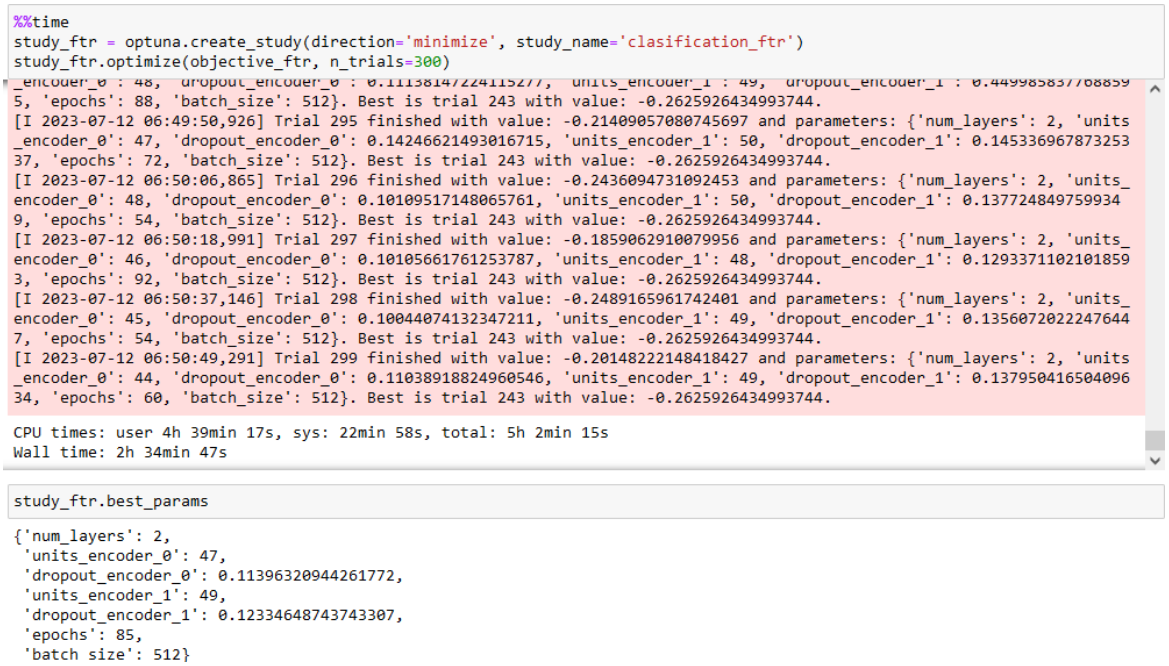
Figura 46: Función de optimización para autoencoder

En la Figura 47 se tiene:

- Ejecución de optuna con 300 iteraciones
- Una línea por iteración en la que se muestra los parámetros utilizados y el resultado obtenido
- Muestra un detalle del tiempo de uso del procesamiento del server aquí tenemos dos temas importantes el tiempo de CPU y el wall time (tiempo transcurrido), el primero hace referencia a la ejecución tomando en cuenta todos los eventos que pudieron interrumpir o ralentizar la ejecución y el segundo hace referencia al tiempo real que toma la ejecución de las líneas de código, siendo los valores respectivamente 5 horas 2 minutos 15 segundos y 2 horas 34 minutos y 47 segundos.

Finalmente se muestra el contenido de la variable best_params que contiene el resultado de la optimización, por cada parámetro definido en la función de optimización nos da el valor que debemos utilizar para obtener el mejor resultado, en este caso tenemos que se debe utilizar 2

capas, una con 47 neuronas y otra con 49, los valores de dropout son de 0.11396320944261772 y



```
%%time
study_ftr = optuna.create_study(direction='minimize', study_name='clasification_ftr')
study_ftr.optimize(objective_ftr, n_trials=300)
```

```
_encoder_0 : 48, 'dropout_encoder_0 : 0.11138147224115277, 'units_encoder_1 : 49, 'dropout_encoder_1 : 0.44998583768859
5, 'epochs': 88, 'batch_size': 512}. Best is trial 243 with value: -0.2625926434993744.
[I 2023-07-12 06:49:50,926] Trial 295 finished with value: -0.21409057080745697 and parameters: {'num_layers': 2, 'units
_encoder_0': 47, 'dropout_encoder_0': 0.14246621493016715, 'units_encoder_1': 50, 'dropout_encoder_1': 0.145336967873253
37, 'epochs': 72, 'batch_size': 512}. Best is trial 243 with value: -0.2625926434993744.
[I 2023-07-12 06:50:06,865] Trial 296 finished with value: -0.2436094731092453 and parameters: {'num_layers': 2, 'units_
encoder_0': 48, 'dropout_encoder_0': 0.10109517148065761, 'units_encoder_1': 50, 'dropout_encoder_1': 0.137724849759934
9, 'epochs': 54, 'batch_size': 512}. Best is trial 243 with value: -0.2625926434993744.
[I 2023-07-12 06:50:18,991] Trial 297 finished with value: -0.1859062910079956 and parameters: {'num_layers': 2, 'units_
encoder_0': 46, 'dropout_encoder_0': 0.10105661761253787, 'units_encoder_1': 48, 'dropout_encoder_1': 0.1293371102101859
3, 'epochs': 92, 'batch_size': 512}. Best is trial 243 with value: -0.2625926434993744.
[I 2023-07-12 06:50:37,146] Trial 298 finished with value: -0.2489165961742401 and parameters: {'num_layers': 2, 'units_
encoder_0': 45, 'dropout_encoder_0': 0.10044074132347211, 'units_encoder_1': 49, 'dropout_encoder_1': 0.1356072022247644
7, 'epochs': 54, 'batch_size': 512}. Best is trial 243 with value: -0.2625926434993744.
[I 2023-07-12 06:50:49,291] Trial 299 finished with value: -0.20148222148418427 and parameters: {'num_layers': 2, 'units
_encoder_0': 44, 'dropout_encoder_0': 0.11038918824960546, 'units_encoder_1': 49, 'dropout_encoder_1': 0.137950416504096
34, 'epochs': 60, 'batch_size': 512}. Best is trial 243 with value: -0.2625926434993744.

CPU times: user 4h 39min 17s, sys: 22min 58s, total: 5h 2min 15s
Wall time: 2h 34min 47s
```

```
study_ftr.best_params
```

```
{'num_layers': 2,
 'units_encoder_0': 47,
 'dropout_encoder_0': 0.11396320944261772,
 'units_encoder_1': 49,
 'dropout_encoder_1': 0.12334648743743307,
 'epochs': 85,
 'batch size': 512}
```

0.12334648743743307 respectivamente, 85 épocas y un batch_size de 512.

Figura 47: Resultado de la ejecución de optuna para el autoencoder

Se definió y entreno un autoencoder en base a los resultados de la optimización de optuna, el tiempo real de ejecución de esto fue de 10 minutos 48 segundos como se puede ver en la Figura 48, paso a repetirse por cada set de datos.

```
%%time
dim_entrada = X_train.shape[1]
capa_entrada = Input(shape=(dim_entrada,))

encoder = Dense(49, activation='tanh')(capa_entrada)
encoder = Dropout(0.131162985642098)(encoder)
encoder = Dense(48, activation='tanh')(encoder)
encoder = Dropout(0.11438066802952078)(encoder)

decoder = Dense(48, activation='tanh')(encoder)
decoder = Dropout(0.11438066802952078)(decoder)
decoder = Dense(49, activation='tanh')(decoder)
decoder = Dropout(0.131162985642098)(decoder)

decoder = Dense(dim_entrada, activation='sigmoid')(decoder)

autoencoder = Model(inputs=capa_entrada, outputs=decoder, name="autoencoder")
autoencoder.compile(optimizer='adam', loss='mae', metrics=['accuracy'])

historia = autoencoder.fit(X_train, X_train, epochs=160, batch_size=64, shuffle=True, validation_data=(X_test,X_test), verbo
```

```
3185/3185 [==============================] - 4s 1ms/step - loss: 0.4796 - accuracy: 0.1353 - val_loss: 0.4701 - val_accu
racy: 0.1476
Epoch 156/160
3185/3185 [==============================] - 4s 1ms/step - loss: 0.4798 - accuracy: 0.1348 - val_loss: 0.4700 - val_accu
racy: 0.1453
Epoch 157/160
3185/3185 [==============================] - 4s 1ms/step - loss: 0.4797 - accuracy: 0.1340 - val_loss: 0.4699 - val_accu
racy: 0.1451
Epoch 158/160
3185/3185 [==============================] - 4s 1ms/step - loss: 0.4796 - accuracy: 0.1347 - val_loss: 0.4699 - val_accu
racy: 0.1474
Epoch 159/160
3185/3185 [==============================] - 4s 1ms/step - loss: 0.4796 - accuracy: 0.1340 - val_loss: 0.4701 - val_accu
racy: 0.1469
Epoch 160/160
3185/3185 [==============================] - 4s 1ms/step - loss: 0.4796 - accuracy: 0.1346 - val_loss: 0.4701 - val_accu
racy: 0.1490
CPU times: user 18min 36s, sys: 55.6 s, total: 19min 32s
Wall time: 10min 48s
```

Figura 48: Ejecución del autoencoder con los mejores parámetros entregados por optuna

En la figura 49, podemos ver las pérdidas del set de entrenamiento y de prueba, lo primero que se observa es que las curvas generadas son basten estables.

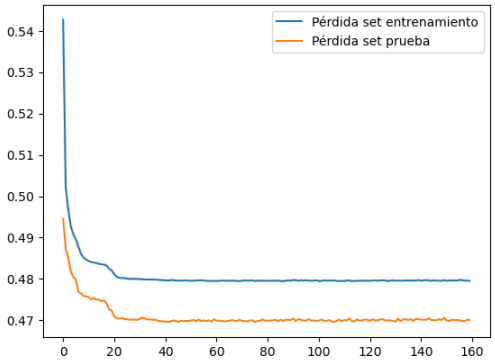


Figura 49: Perdidas del set de entrenamiento y pruebas

Ahora utilizando el set de test en la Figura 50 se ve que al ejecutar la predicción tenemos como accuracy un valor de 0.57 es decir el resultado es aleatorio y en las Figura 51 se muestre gráficamente este resultado por medio de una matriz de confusión.

Con otro set de datos que no se sido visto por el modelo realizamos una nueva predicción Figura

52, aquí se ve que el resultado se mantiene al entregar un accuracy de 0.57 y en las

Figura 51 se muestre gráficamente este resultado por medio de una matriz de confusión, estospasos se repiten para los ocho sets de datos.

```
%%time
# Predecir el churn de clientes en el conjunto de test
X_pred = autoencoder.predict(X_test)
ecm = np.mean(np.power(X_test-X_pred,2), axis=1)

losses = historia.history['loss']
val_losses = historia.history['val_loss']

# Mapear las etiquetas  a (0, 1)
umbral = np.mean(losses) + np.std(losses)
y_pred = np.where(ecm > umbral, 1, 0)

# Evaluar el rendimiento del modelo en el conjunto de test
print("Performance on Validation Set:")
print(classification_report(y_test, y_pred))
```

```
1991/1991 [==============================] - 1s 507us/step
Performance on Validation Set:
              precision    recall  f1-score   support

           0       0.73      0.54      0.62     41497
           1       0.42      0.62      0.50     22191

    accuracy                           0.57     63688
   macro avg       0.57      0.58      0.56     63688
weighted avg       0.62      0.57      0.58     63688

CPU times: user 2.09 s, sys: 263 ms, total: 2.36 s
Wall time: 1.67 s
```

Figura 50: Resultados de la predicción con el set de pruebas para autoencoder

```
# Matriz de confusión
cm = confusion_matrix(y_test, y_pred)
f = sns.heatmap(cm, annot=True, fmt='d')
```

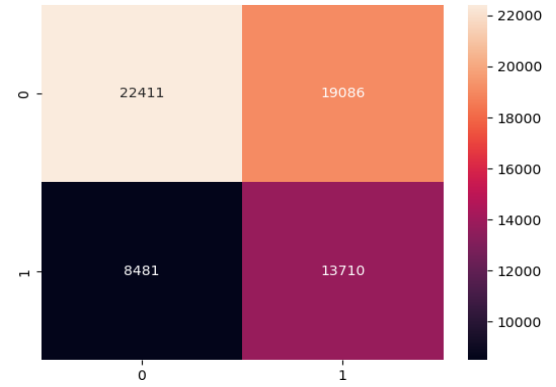


Figura 51: Matriz de confusión del set de datos de prueba para autoencoder

```
%%time
# Predecir el churn de clientes en el conjunto de validación
X_pred_val = autoencoder.predict(X_val)
ecm_val = np.mean(np.power(X_val-X_pred_val,2), axis=1)

# Mapear las etiquetas  a (0, 1)
y_pred_val = np.where(ecm_val > umbral, 1, 0)

# Evaluar el rendimiento del modelo en el conjunto de validación
print("Performance on Validation Set:")
print(classification_report(y_val, y_pred_val))
```

```
1593/1593 [==============================] - 1s 513us/step
Performance on Validation Set:
              precision    recall  f1-score   support

           0       0.72      0.54      0.62     32901
           1       0.43      0.62      0.51     18050

    accuracy                           0.57     50951
   macro avg       0.57      0.58      0.56     50951
weighted avg       0.62      0.57      0.58     50951

CPU times: user 1.74 s, sys: 153 ms, total: 1.89 s
Wall time: 1.33 s
```

Figura 52: Resultados de la predicción con el set de validación para autoencoder

```
# Matriz de confusión
cm = confusion_matrix(y_val, y_pred_val)
f = sns.heatmap(cm, annot=True, fmt='d')
```

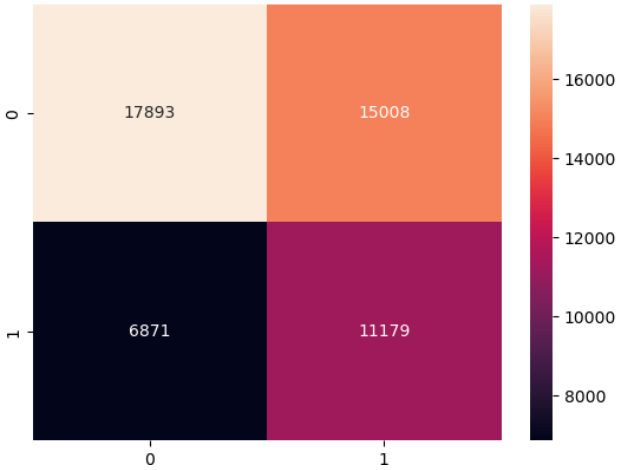


Figura 53: Matriz de confusión del set de datos de validación para autoencoder

En la Figura 54 se muestra la comparación entre los resultados de los ocho modelos generados, lo que salta a la vista es el hecho de que los mejores resultados se obtienen con el set de datos filtrado, su accuracy está en el rango de 0.58 a 0.67 mientras que para los datos sin filtrar su rango esta entre 0.57 a 0.62.

45 Colum. y toda la data
Accuracy: 0.57

	0	1
Precision	0.72	0.43
Recall	0.54	0.62
F1-Score	0.62	0.51
Support	32901	18050

40 Colum. y toda la data
Accuracy: 0.58

	0	1
Precision	0.71	0.43
Recall	0.6	0.56
F1-Score	0.65	0.49
Support	32901	18050

36 Colum. y toda la data
Accuracy: 0.59

	0	1
Precision	0.73	0.44
Recall	0.58	0.5
F1-Score	0.64	0.51
Support	32901	18050

31 Colum. y toda la data
Accuracy: 0.62

	0	1
Precision	0.75	0.47
Recall	0.61	0.62
F1-Score	0.67	0.53
Support	32901	18050

45 Colum. y data filtrada
Accuracy: 0.58

	0	1
Precision	0.72	0.44
Recall	0.58	0.6
F1-Score	0.64	0.5
Support	32901	18050

40 Colum. y data filtrada
Accuracy: 0.59

	0	1
Precision	0.77	0.45
Recall	0.53	0.7
F1-Score	0.63	0.55
Support	32901	18050

36 Colum. y data filtrada
Accuracy: 0.67

	0	1
Precision	0.68	0.58
Recall	0.9	0.24
F1-Score	0.78	0.34
Support	32901	18050

36 Colum. y data filtrada
Accuracy: 0.59

	0	1
Precision	0.73	0.44
Recall	0.58	0.61
F1-Score	0.65	0.51
Support	32901	18050

Figura 54: Resultado de la ejecución del modelo optimizado con cada set de datos para autoencoder

En la Figura 55, se muestra el mejor modelo posible en base a los resultados de cada uno, en este caso es el modelo que tiene 36 columnas y la data filtrada, con un accuracy de 0.67.

```python
# Función para obtener la accuracy de un modelo
def get_accuracy(model):
    return model['accuracy']

# Obtener el modelo con mejor desempeño según la accuracy
best_model = max(results, key=get_accuracy)

# Mostrar el nombre del modelo con mejor desempeño
print("Mejor modelo según la accuracy:", best_model['model_name'])
print("Accuracy del mejor modelo:", best_model['accuracy'])
```

```
Mejor modelo según la accuracy: 36 Colum. y data filtrada
Accuracy del mejor modelo: 0.67
```

Figura 55: Mejor modelo de autoencoder

3.2.1.2. Máquinas de vectores de soporte (SVM)

Partimos de los mismos sets de datos que se usó para el autoencoder Figura 45. Se utiliza optuna para la optimización de hiperparámetros ahora lo que cambia es la función que se maneja para la optimización Figura 56, dentro de los parámetros definidos para la optimización se tiene:

- Como tipos de kernel lineal, polinómico o radial
- Para contaminación rango entre 10 y 100%
- Importancia de las fronteras se define que pruebe scale y gamma.

Finalmente, para el grado del polinomio se da como opciones segundo, tercero y cuarto grado.

```python
def objective(trial):
    param = {
        'kernel': trial.suggest_categorical('kernel', ['linear', 'rbf', 'poly']),
        'nu': trial.suggest_uniform('nu', 0.1, 1.0),
        'gamma': trial.suggest_categorical('gamma',  ['scale', 'auto']),
        'degree': trial.suggest_int('degree', 2,3,4)
    }

    model = OneClassSVM(**param)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_pred_mapped = [1 if pred == -1 else 0 for pred in y_pred]

    return 1 - accuracy_score(y_test, y_pred_mapped)
```

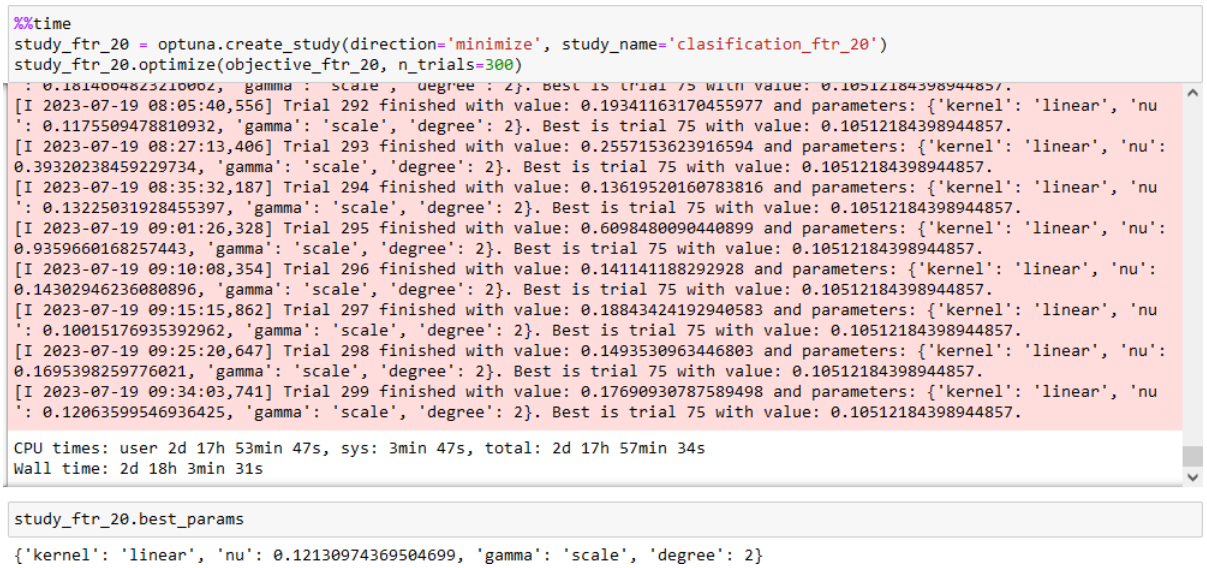
Figura 56: Función de optimización para SVM

En la figura 57 se tiene:

- Ejecución de optuna con 300 iteraciones
- Una línea por iteración en la que se muestra los parámetros utilizados y el resultado obtenido
- Muestra un detalle del tiempo de uso del procesamiento del server aquí tenemos dos temas importantes el tiempo de CPU y el wall time (tiempo transcurrido), el primero hace referencia a la ejecución tomando en cuenta todos los eventos que pudieron interrumpir o ralentizar la ejecución y el segundo hace referencia al tiempo real que toma la ejecución

de las líneas de código, siendo los valores respectivamente 2 días 17 horas 57 minutos 34 segundos y 2 días 18 horas 3 minutos y 31 segundos.

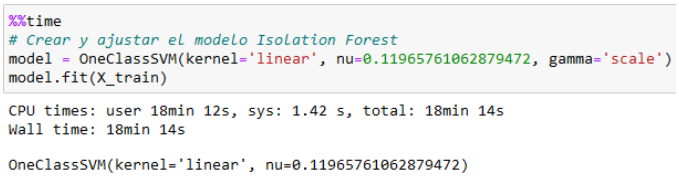
Finalmente se muestra el contenido de la variable best_params que contiene el resultado de la optimización, por cada parámetro definido en la función de optimización nos da el valor que debemos utilizar para obtener el mejor resultado, en este caso tenemos que se debe utilizar un kernel lineal, con



nu de 0.12130974369504699, gamma scale y degree 2.

Figura 57: Resultado de la ejecución de optuna para SVM

Se definió y entrenó el SVM en base a los resultados de la optimización de optuna, el tiempo real de ejecución de esto fue de 18 minutos 14 segundos como se puede ver en la Figura 58, paso a



repetirse por cada set de datos.

Figura 58: Ejecución del SVM con los mejores parámetros entregados por optuna

Ahora utilizando el set de test en la Figura 59 se ve que al ejecutar la predicción tenemos como accuracy un valor de 0.48 es decir el resultado es aleatorio y en las Figura 60 se muestre gráficamente este resultado por medio de una matriz de confusión.

Con otro set de datos que no se sido visto por el modelo realizamos una nueva predicción Figura 61, aquí se ve que el resultado mejora considerablemente al entregar un accuracy de 0.82

y en las Figura 52 se muestre gráficamente este resultado por medio de una matriz de confusión,estos pasos se repiten para los ocho sets de datos.

```
%%time
# Predecir el churn de clientes en el conjunto de prueba
# Validación cruzada y predicciones
y_pred_ftr_30 = cross_val_predict(model_ftr_30, X_test_ftr_30, y_test_ftr_30, cv=50)

# Mapear las etiquetas (-1, 1) a (0, 1)
y_pred_mapped_ftr_30 = [1 if pred == -1 else 0 for pred in y_pred_ftr_30]
```

```
CPU times: user 10min 10s, sys: 580 ms, total: 10min 11s
Wall time: 10min 11s
```

```
# Evaluar el rendimiento del modelo en el conjunto de prueba
print("Performance on Test Set:")
print(classification_report(y_test_ftr_30, y_pred_mapped_ftr_30))
```

```
Performance on Test Set:
              precision    recall  f1-score   support

           0       0.67      0.40      0.50     41497
           1       0.36      0.64      0.46     22191

    accuracy                           0.48     63688
   macro avg       0.52      0.52      0.48     63688
weighted avg       0.56      0.48      0.48     63688
```

Figura 59: Resultados de la predicción con el set de pruebas para SVM

```
# Matriz de confusión
cm_ftr_30 = confusion_matrix(y_test_ftr_30, y_pred_mapped_ftr_30)
f = sns.heatmap(cm_ftr_30, annot=True, fmt='d')
```

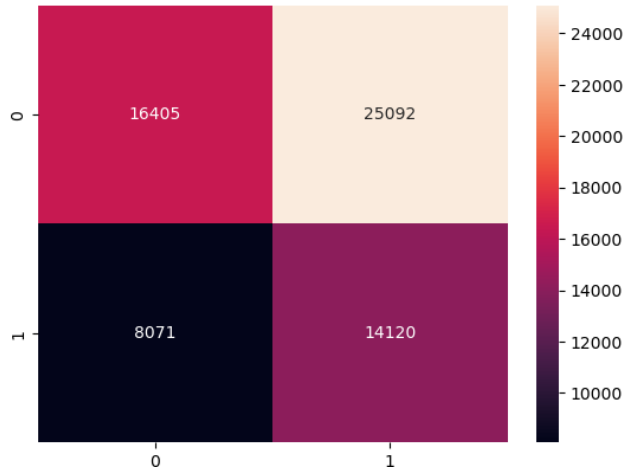


Figura 60: Matriz de confusión del set de datos de prueba para SVM

```
# Predecir el churn de clientes en el conjunto de validación
y_pred_val_ftr_30 = model_ftr_30.predict(X_val_ftr_30)

# Mapear las etiquetas (-1, 1) a (0, 1)
y_pred_val_mapped_ftr_30 = [1 if pred == -1 else 0 for pred in y_pred_val_ftr_30]
```

```
# Evaluar el rendimiento del modelo en el conjunto de validación
print("Performance on Validation Set:")
print(classification_report(y_val_ftr_30, y_pred_val_mapped_ftr_30))
```

```
Performance on Validation Set:
              precision    recall  f1-score   support

           0       0.93      0.78      0.85     32901
           1       0.69      0.89      0.78     18050

    accuracy                           0.82     50951
   macro avg       0.81      0.83      0.81     50951
weighted avg       0.84      0.82      0.82     50951
```

Figura 61: Resultados de la predicción con el set de validación para SVM

```
# Matriz de confusión
cm_val_ftr_30 = confusion_matrix(y_val_ftr_30, y_pred_val_mapped_ftr_30)
f = sns.heatmap(cm_val_ftr_30, annot=True, fmt='d')
```

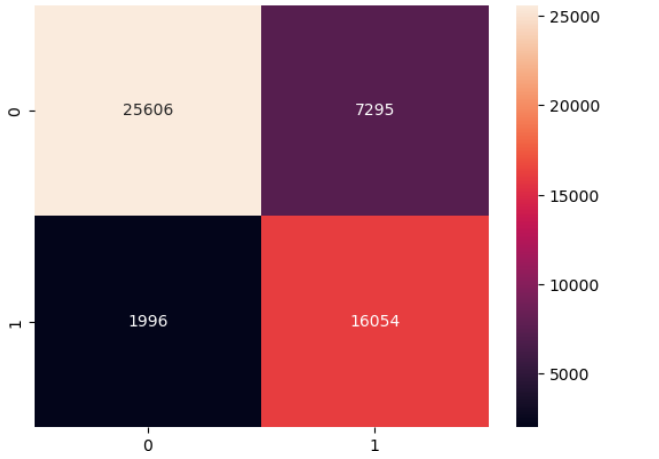


Figura 62: Matriz de confusión del set de datos de validación para SVM

En la Figura 63 se muestra la comparación entre los resultados los ocho modelos generados, lo que salta a la vista es el hecho de que los mejores resultados se obtienen con el set de datos sin filtrar, su accuracy está en el rango de 0.86 a 0.81 mientras que para los datos filtrados su rango esta entre 0.70 a 0.82.

45 Colum. y toda la data
Accuracy: 0.86

	0	1
Precision	0.91	0.78
Recall	0.87	0.85
F1-Score	0.89	0.81
Support	32901	18050

40 Colum. y toda la data
Accuracy: 0.87

	0	1
Precision	0.92	0.8
Recall	0.88	0.86
F1-Score	0.9	0.82
Support	32901	18050

36 Colum. y toda la data
Accuracy: 0.91

	0	1
Precision	0.93	0.87
Recall	0.93	0.87
F1-Score	0.93	0.87
Support	32901	18050

31 Colum. y toda la data
Accuracy: 0.89

	0	1
Precision	0.93	0.84
Recall	0.91	0.87
F1-Score	0.92	0.85
Support	32901	18050

45 Colum. y data filtrada
Accuracy: 0.70

	0	1
Precision	0.89	0.55
Recall	0.61	0.87
F1-Score	0.73	0.67
Support	32901	18050

40 Colum. y data filtrada
Accuracy: 0.82

	0	1
Precision	0.92	0.7
Recall	0.79	0.87
F1-Score	0.85	0.77
Support	32901	18050

36 Colum. y data filtrada
Accuracy: 0.89

	0	1
Precision	0.94	0.82
Recall	0.89	0.9
F1-Score	0.92	0.86
Support	32901	18050

36 Colum. y data filtrada
Accuracy: 0.82

	0	1
Precision	0.93	0.69
Recall	0.78	0.89
F1-Score	0.85	0.78
Support	32901	18050

Figura 63: Resultado de la ejecución del modelo optimizado con cada set de datos para SVM

En la Figura 64, se muestra el mejor modelo posible en base a los resultados de cada uno, en este caso es el modelo que tiene 36 columnas y la toda la data, con un accuracy de 0.91.

```python
# Función para obtener la accuracy de un modelo
def get_accuracy(model):
    return model['accuracy']

# Obtener el modelo con mejor desempeño según la accuracy
best_model = max(results, key=get_accuracy)

# Mostrar el nombre del modelo con mejor desempeño
print("Mejor modelo según la accuracy:", best_model['model_name'])
print("Accuracy del mejor modelo:", best_model['accuracy'])
```

```
Mejor modelo según la accuracy: 36 Colum. y toda la data
Accuracy del mejor modelo: 0.91
```

Figura 64: Mejor modelo de SVM

3.2.1.3. Isolation forest

Se continúa trabajando con los mismos sets de datos que se muestran en la Figura 45.Se utiliza optuna para la optimización de hiperparámetros ahora lo que cambia es la función que se utiliza para la optimización Figura 65, dentro de los parámetros definidos para la optimización setiene:

- La cantidad de árboles entre 100 y 700
- La cantidad de muestras para construir el árbol entre 10 y 100%
- El nivel de contaminación entre 1 y 50%.
- La cantidad de características a utilizar entre el 10 y 100%.

Adicional se coloca random_state para que el resultado sea reproducible

```python
def objective(trial):
    param = {
        'n_estimators': trial.suggest_int('n_estimators', 100, 700, step=100),
        'max_samples': trial.suggest_uniform('max_samples', 0.1, 1.0),
        'contamination': trial.suggest_uniform('contamination',  0.01, 0.5),
        'random_state': trial.suggest_int('random_state', 1, 2000),
        'max_features': trial.suggest_uniform('max_features', 0.1, 1.0)
        #'tree_method': 'gpu_hist'
    }

    model = IsolationForest(**param)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_pred_mapped = [1 if pred == -1 else 0 for pred in y_pred]

    return 1 - accuracy_score(y_test, y_pred_mapped)
```

Figura 65: Función de optimización para Isolation forest

En la Figura 66 se tiene:

- Ejecución de optuna con 300 iteraciones
- Una línea por iteración en la que se muestra los parámetros utilizados y el resultado obtenido.

- Muestra un detalle del tiempo de uso del procesamiento del server aquí tenemos dos temas importantes el tiempo de CPU y el wall time (tiempo transcurrido), el primero hace referencia a la ejecución tomando en cuenta todos los eventos que pudieron interrumpir o ralentizar la ejecución y el segundo hace referencia al tiempo real que toma la ejecución de las líneas de código, siendo los valores respectivamente 5 horas 32 minutos 55 segundos y 5 horas 32 minutos y 57 segundos.

Finalmente se muestra el contenido de la variable best_params que contiene el resultado de la optimización, por cada parámetro definido en la función de optimización nos da el valor que debemos utilizar para obtener el mejor resultado, en este caso tenemos que se debe utilizar 400 árboles, 0.748350046617172 muestras para construir los árboles, el nivel de contaminación es de 0.12717485409388574, 0.1196081307906796039 características y un random_state de1124.

Figura 66: Resultado de la ejecución de optuna para Isolation forest

```
%%time
study_ftr = optuna.create_study(direction='minimize', study_name='clasification_ftr')
study_ftr.optimize(objective_ftr, n_trials=300)
```

```
x_samples' : 0.10095508574983935, 'contamination' : 0.09969014148200S, 'random_state' : 816, 'max_features' : 0.11515852978948252}. Best is trial 213 with value: 0.3219915839718628.
[I 2023-07-10 07:55:28,515] Trial 295 finished with value: 0.3302662982037432 and parameters: {'n_estimators': 400, 'max_samples': 0.23147784448960962, 'contamination': 0.12571435368763248, 'random_state': 855, 'max_features': 0.1341979370497215}. Best is trial 213 with value: 0.3219915839718628.
[I 2023-07-10 07:56:32,343] Trial 296 finished with value: 0.3529864338650923 and parameters: {'n_estimators': 500, 'max_samples': 0.43335168992395884, 'contamination': 0.2322159302638304, 'random_state': 1151, 'max_features': 0.46422972259626266}. Best is trial 213 with value: 0.3219915839718628.
[I 2023-07-10 07:58:10,709] Trial 297 finished with value: 0.33843110162039947 and parameters: {'n_estimators': 700, 'max_samples': 0.6826811057621776, 'contamination': 0.11217104927974704, 'random_state': 1076, 'max_features': 0.3601886962346137}. Best is trial 213 with value: 0.3219915839718628.
[I 2023-07-10 07:58:57,183] Trial 298 finished with value: 0.3380699660846628 and parameters: {'n_estimators': 400, 'max_samples': 0.602720392430739, 'contamination': 0.1480425465288676, 'random_state': 1174, 'max_features': 0.11419554463055642}. Best is trial 213 with value: 0.3219915839718628.
[I 2023-07-10 08:00:02,587] Trial 299 finished with value: 0.34482163044843617 and parameters: {'n_estimators': 400, 'max_samples': 0.75829260202S245, 'contamination': 0.13423860672657442, 'random_state': 757, 'max_features': 0.7041855672418597}. Best is trial 213 with value: 0.3219915839718628.
CPU times: user 5h 30min 55s, sys: 1min 59s, total: 5h 32min 55s
Wall time: 5h 32min 57s
```

```
study_ftr.best_params
```

```
{'n_estimators': 400,
 'max_samples': 0.7483050046617172,
 'contamination': 0.12717485409388574,
 'random_state': 1124,
 'max_features': 0.11960813806796039}
```

Se definió y entreno el modelo en base al resultado de la optimización de optuna, el tiempo real de ejecución de esto fue de 23.2 segundos (tiempo extremadamente bajo en comparación del resto de modelos ejecutados) como se puede ver en la Figura 67, paso a repetirse por cada set de datos.

```
%%time
# Crear y ajustar el modelo Isolation Forest
model = IsolationForest(n_estimators=200,contamination=0.12049673194604617,max_samples=0.9488697173582842, max_features= 0.1
model.fit(X_train)
```

```
CPU times: user 23.1 s, sys: 124 ms, total: 23.2 s
Wall time: 23.2 s

IsolationForest(contamination=0.12049673194604617,
                max_features=0.1174583797437434, max_samples=0.9488697173582842,
                n_estimators=200, random_state=1343)
```

Figura 67: Ejecución del Isolation forest con los mejores parámetros entregados por optuna

Ahora utilizando el set de test en la Figura 68 se ve que al ejecutar la predicción tenemos como accuracy un valor de 0.68 este resultado con el set de pruebas es mejor comparado con los modelos previos y en las Figura 69 se muestre gráficamente este resultado por medio de unamatriz de confusión.

Con otro set de datos que no se sido visto por el modelo realizamos una nueva predicción Figura 70, aquí se ve que el resultado se mantiene al entregar un accuracy de 0.68 y en las Figura 71 se muestre gráficamente este resultado por medio de una matriz de confusión, estos pasos se repiten para los ocho sets de datos

```
%%time
# Predecir el churn de clientes en el conjunto de prueba
# Validación cruzada y predicciones
y_pred = cross_val_predict(model, X_test, y_test, cv=50)

# Mapear las etiquetas (-1, 1) a (0, 1)
y_pred_mapped = [1 if pred == -1 else 0 for pred in y_pred]

# Evaluar el rendimiento del modelo en el conjunto de prueba
print("Performance on Test Set:")
print(classification_report(y_test, y_pred_mapped))
```

```
Performance on Test Set:
              precision    recall  f1-score   support

           0       0.69      0.92      0.79     41497
           1       0.61      0.22      0.32     22191

    accuracy                           0.68     63688
   macro avg       0.65      0.57      0.56     63688
weighted avg       0.66      0.68      0.63     63688

CPU times: user 5min 24s, sys: 248 ms, total: 5min 24s
Wall time: 5min 24s
```

Figura 68: Resultados de la predicción con el set de pruebas para Isolation forest

```python
# Matriz de confusión
cm = confusion_matrix(y_test, y_pred_mapped)
f = sns.heatmap(cm, annot=True, fmt='d')
```

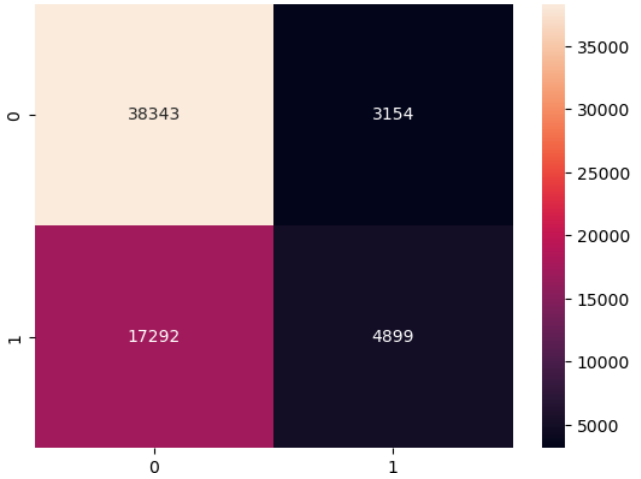


Figura 69: Matriz de confusión del set de datos de prueba para Isolation forest

```python
# Predecir el churn de clientes en el conjunto de validación
y_pred_val = model.predict(X_val)

# Mapear las etiquetas (-1, 1) a (0, 1)
y_pred_val_mapped = [1 if pred == -1 else 0 for pred in y_pred_val]

# Evaluar el rendimiento del modelo en el conjunto de validación
print("Performance on Validation Set:")
print(classification_report(y_val, y_pred_val_mapped))
```

```
Performance on Validation Set:
              precision    recall  f1-score   support

           0       0.68      0.93      0.79     32901
           1       0.63      0.22      0.33     18050

    accuracy                           0.68     50951
   macro avg       0.66      0.57      0.56     50951
weighted avg       0.66      0.68      0.62     50951
```

Figura 70: Resultados de la predicción con el set de validación para Isolation forest

```python
# Matriz de confusión
cm = confusion_matrix(y_val, y_pred_val_mapped)
f = sns.heatmap(cm, annot=True, fmt='d')
```

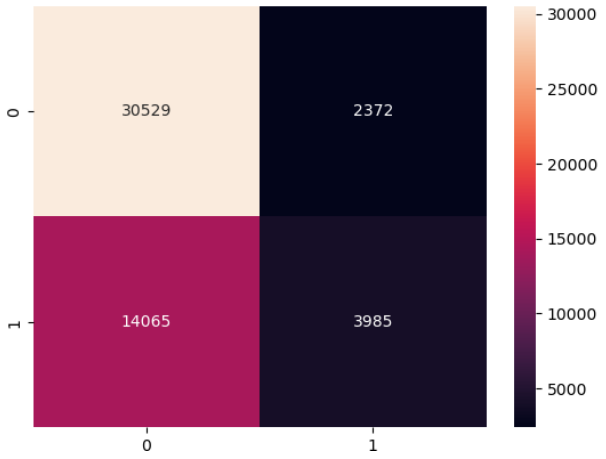


Figura 71: Matriz de confusión del set de datos de validación para Isolation forest

En la Figura 72 se muestra la comparación entre los resultados los ocho modelos generados, lo que salta a la vista es el hecho de que los resultados son similares para la data sin filtrar como para la filtrada, teniendo como rangos de accuracy de 0.68 a 0.76 y de 0.67 a 0.75 respectivamente.

45 Colum. y toda la data
Accuracy: 0.68

	0	1
Precision	0.68	0.63
Recall	0.93	0.22
F1-Score	0.79	0.33
Support	32901	18050

40 Colum. y toda la data
Accuracy: 0.67

	0	1
Precision	0.68	0.62
Recall	0.94	0.19
F1-Score	0.79	0.29
Support	32901	18050

36 Colum. y toda la data
Accuracy: 0.68

	0	1
Precision	0.68	0.68
Recall	0.95	0.19
F1-Score	0.79	0.3
Support	32901	18050

31 Colum. y toda la data
Accuracy: 0.76

	0	1
Precision	0.81	0.65
Recall	0.81	0.66
F1-Score	0.81	0.66
Support	32901	18050

45 Colum. y data filtrada
Accuracy: 0.67

	0	1
Precision	0.68	0.61
Recall	0.92	0.22
F1-Score	0.78	0.33
Support	32901	18050

40 Colum. y data filtrada
Accuracy: 0.68

	0	1
Precision	0.69	0.6
Recall	0.91	0.25
F1-Score	0.78	0.35
Support	32901	18050

36 Colum. y data filtrada
Accuracy: 0.67

	0	1
Precision	0.68	0.58
Recall	0.9	0.24
F1-Score	0.78	0.34
Support	32901	18050

36 Colum. y data filtrada
Accuracy: 0.75

	0	1
Precision	0.79	0.65
Recall	0.82	0.61
F1-Score	0.81	0.63
Support	32901	18050

Figura 72: Resultado de la ejecución del modelo optimizado con cada set de datos para Isolation forest

En la Figura 73, se muestra el mejor modelo posible en base a los resultados de cada uno, en este caso es el modelo que tiene 31 columnas y la toda la data, con un accuracy de 0.76

```
# Función para obtener la accuracy de un modelo
def get_accuracy(model):
    return model['accuracy']

# Obtener el modelo con mejor desempeño según la accuracy
best_model = max(results, key=get_accuracy)

# Mostrar el nombre del modelo con mejor desempeño
print("Mejor modelo según la accuracy:", best_model['model_name'])
print("Accuracy del mejor modelo:", best_model['accuracy'])
```

```
Mejor modelo según la accuracy: 31 Colum. y toda la data
Accuracy del mejor modelo: 0.76
```

Figura 73: Mejor modelo de Isolation forest

3.2.1.4. Xgboost (predicción de churn)

Para este caso se utiliza únicamente el set de datos completo Figura 74.

```
dfFinal = pd.read_excel('dfFinalModelosBinarios.xlsx')
```

Figura 74: Set de datos para XGBoost

Se utiliza optuna de hiperparámetros ahora lo que cambia es la función que se utiliza para la optimización Figura 56, dentro de los parámetros definidos para la optimización se tiene:

- Cantidad de árboles entre 1 y 6000

- Profundidad de los árboles entre 1 y 200
- Tasa de aprendizaje entre 0.00001 y 1
- Peso mínimo por nodo entre 1 y 100
- Reducción de ganancia 0.00001 y 1
- Subconjunto de entrenamiento entre 0.00001 y 1
- Porcentaje de características entre 0.00001 y 1
- L1 entre 0.00001 y 1
- L2 entre 0.00001 y 1
- Método de construcción histograma con uso de GPU
- Parámetro random_state para que el resultado sea reproducible

```
def objective(trial):
    param = {
        'max_depth': trial.suggest_int('max_depth', 1, 200),
        #'num_class': trial.suggest_int('num_class', 2, 2),
        'learning_rate': trial.suggest_float('learning_rate', 0.00001, 1.0),
        'n_estimators': trial.suggest_int('n_estimators', 1, 6000),
        'min_child_weight': trial.suggest_int('min_child_weight', 1, 100),
        'gamma': trial.suggest_float('gamma', 0.00001, 1.0),
        'subsample': trial.suggest_float('subsample', 0.00001, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.00001, 1.0),
        'reg_alpha': trial.suggest_float('reg_alpha', 0.00001, 1.0),
        'reg_lambda': trial.suggest_float('reg_lambda', 0.00001, 1.0),
        'random_state': trial.suggest_int('random_state', 1, 6000),
        'tree_method': 'gpu_hist'
    }

    model = xgb.XGBClassifier(**param)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    return 1 - accuracy_score(y_test, y_pred)
```

Adicional se coloca random_state para que el resultado sea reproducible

Figura 75: Función de optimización para XGBoost

En la Figura 76 se tiene:

- Ejecución de optuna con 2000 iteraciones
- Una línea por iteración en la que se muestra los parámetros utilizados y el resultado obtenido
- Muestra un detalle del tiempo de uso del procesamiento del server aquí tenemos dos temas importantes el tiempo de CPU y el wall time (tiempo transcurrido), el primero hace referencia a la ejecución tomando en cuenta todos los eventos que pudieron interrumpir o ralentizar la ejecución y el segundo hace referencia al tiempo real que toma la ejecución

de las líneas de código, siendo los valores respectivamente 1 días 9 horas 25 minutos 46 segundos y 2 horas 59 minutos y 4 segundos.

- Finalmente se muestra el contenido de la variable best_params que contiene el resultado de la optimización, por cada parámetro definido en la función de optimización nos da el valor que debemos utilizar para obtener el mejor resultado, en este caso tenemos que se debe utilizar 2678 árboles, con una profundidad de 97, una tasa de aprendizaje de 0.2192310230606184, peso por nodo 1, reducción de ganancia 0.4377813335517329, subconjunto de 0.7708059738162309, porcentaje de características 0.8358255973585463, L1 de 0.24279719709906583, L2 de 0.948475715116304 y un random_state 5119.

Figura 76: Resultado de la ejecución de optuna para XGBoost

```
%%time
study = optuna.create_study(direction='minimize', study_name='clasification1')
study.optimize(objective, n_trials=2000)
```

```
1151242, 'random_state': 5101}. Best is trial 1004 with value: 0.001617259138299243.
[I 2023-07-06 04:01:46,804] Trial 1996 finished with value: 0.002292425574676571 and parameters: {'max_depth': 71, 'learning_rate': 0.19361799417780973, 'n_estimators': 3061, 'min_child_weight': 6, 'gamma': 0.24876405382169575, 'subsample': 0.8073370733671793, 'colsample_bytree': 0.8192988549990656, 'reg_alpha': 0.17463812876819804, 'reg_lambda': 0.7921927619039781, 'random_state': 4921}. Best is trial 1004 with value: 0.001617259138299243.
[I 2023-07-06 04:01:50,890] Trial 1997 finished with value: 0.001758573043587508 and parameters: {'max_depth': 78, 'learning_rate': 0.06041919797309815, 'n_estimators': 1408, 'min_child_weight': 1, 'gamma': 0.1581964412803348, 'subsample': 0.7834040026869863, 'colsample_bytree': 0.7438365186325483, 'reg_alpha': 0.42511404860627716, 'reg_lambda': 0.6352852743317017, 'random_state': 5425}. Best is trial 1004 with value: 0.001617259138299243.
[I 2023-07-06 04:01:54,396] Trial 1998 finished with value: 0.002166813214420249 and parameters: {'max_depth': 84, 'learning_rate': 0.508211355853615, 'n_estimators': 3235, 'min_child_weight': 3, 'gamma': 0.6565632448577958, 'subsample': 0.8346725134825591, 'colsample_bytree': 0.7943497522321105, 'reg_alpha': 0.653264883420694, 'reg_lambda': 0.5534874850867932, 'random_state': 5884}. Best is trial 1004 with value: 0.001617259138299243.
[I 2023-07-06 04:02:13,825] Trial 1999 finished with value: 0.002433739479964836 and parameters: {'max_depth': 97, 'learning_rate': 0.02038013767685829, 'n_estimators': 2773, 'min_child_weight': 4, 'gamma': 0.01937067483352893, 'subsample': 0.6855033972019108, 'colsample_bytree': 0.39523111140322875, 'reg_alpha': 0.3244891774487898, 'reg_lambda': 0.14435431454701128, 'random_state': 5223}. Best is trial 1004 with value: 0.001617259138299243.

CPU times: user 1d 9h 20min 46s, sys: 5min, total: 1d 9h 25min 46s
Wall time: 2h 59min 4s
```

```
study.best_params
```

```
{'max_depth': 97,
 'learning_rate': 0.2192310230606184,
 'n_estimators': 2678,
 'min_child_weight': 1,
 'gamma': 0.4377813335517329,
 'subsample': 0.7708059738162609,
 'colsample_bytree': 0.8358255973585463,
 'reg_alpha': 0.24279719709906583,
 'reg_lambda': 0.940475715116304,
 'random_state': 5119}
```

Se definió el modelo en base a los resultados de la optimización, el tiempo real de ejecución de esto fue de 52.5 microsegundos como se puede ver en la Figura 77.

```
%%time
## ---------- XGBoost model v1 ----------
## base run of model with default hyperparameters

xgb_clf = xgb.XGBClassifier(objective='multi:softmax',
                            num_class=2,
                            max_depth= 97,
                            learning_rate=0.2192310230606184,
                            min_child_weight= 1,
                            gamma= 0.4377813335517329,
                            subsample= 0.7708059738162609,
                            colsample_bytree= 0.8358255973585463,
                            reg_alpha= 0.24279719709906583,
                            reg_lambda= 0.940475715116304,
                            eval_metric=['merror','mlogloss'],
                            n_estimators=2678,
                            random_state= 5119
                           #  seed=42
                            )
```

```
CPU times: user 0 ns, sys: 42 µs, total: 42 µs
Wall time: 52.5 µs
```

Figura 77: Configuración de XGBoost con los mejores parámetros entregados por optuna

En la figura 78 vemos el entrenamiento del modelo, mismo que tomo un tiempo de 2 horas 29 minutos 10segundos y en la Figura 79 podemos ver que almacenamos el modelo almacenado.

```
%%time
xgb_clf.fit(X_train,
            y_train,
            verbose=1,
            eval_set=[(X_train, y_train), (X_test, y_test)]
           )
```

```
CPU times: user 23d 13h 35min 20s, sys: 34min 44s, total: 23d 14h 10min 4s
Wall time: 2h 29min 10s

XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1,
              colsample_bytree=0.8358255973585463, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=['merror', 'mlogloss'],
              feature_types=None, gamma=0.4377813335517329, gpu_id=-1,
              grow_policy='depthwise', importance_type=None,
              interaction_constraints='', learning_rate=0.2192310230606184,
              max_bin=256, max_cat_threshold=64, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=97, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints='()', n_estimators=2678,
              n_jobs=0, num_class=2, num_parallel_tree=1,
              objective='multi:softmax', ...)
```

Figura 78: Ejecución del XGBoost con los mejores parámetros entregados por optuna

```
joblib.dump(xgb_clf, "xgb_clf.pkl")
```

```
['xgb_clf.pkl']
```

Figura 79: Almacenamiento del modelo XGBoost

Ahora utilizando el set de test en la Figura 80 y 81 se ve que al ejecutar la predicción tenemos como accuracy un valor de casi el 100% es decir el resultado es el mejor obtenido hasta el momento y en las Figura 82 se muestre gráficamente este resultado por medio de una matrizde confusión.

```
%%time
# preparing evaluation metric plots
results = xgb_clf.evals_result()
epochs = len(results['validation_0']['mlogloss'])
x_axis = range(0, epochs)

# xgboost 'mlogloss' plot
fig, ax = plt.subplots(figsize=(9,5))
ax.plot(x_axis, results['validation_0']['mlogloss'], label='Train')
ax.plot(x_axis, results['validation_1']['mlogloss'], label='Test')
ax.legend()
plt.ylabel('mlogloss')
plt.title('GridSearchCV XGBoost mlogloss')
plt.show()

# xgboost 'merror' plot
fig, ax = plt.subplots(figsize=(9,5))
ax.plot(x_axis, results['validation_0']['merror'], label='Train')
ax.plot(x_axis, results['validation_1']['merror'], label='Test')
ax.legend()
plt.ylabel('merror')
plt.title('GridSearchCV XGBoost merror')
plt.show()
```

Figura 80: Generación de métricas de XGBoost

```
--------------- Classification Report ---------------

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     41497
           1       1.00      1.00      1.00     22191

    accuracy                           1.00     63688
   macro avg       1.00      1.00      1.00     63688
weighted avg       1.00      1.00      1.00     63688

---------------------- XGBoost ----------------------
CPU times: user 7.96 s, sys: 23.9 s, total: 31.8 s
Wall time: 771 ms
```

Figura 81: Métricas de XGBoost

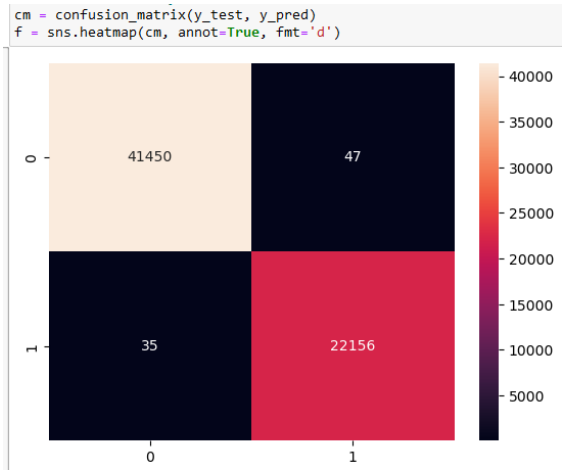


Figura 82: Matriz de confusión de XGBoost

Con otro set de datos que no se sido visto por el modelo realizamos la validación cruzada Figura 83, aquí se ve que el resultado se mantiene al entregar un accuracy de 99% y en las Figura 84 se muestre gráficamente este resultado por medio de una matriz de confusión.

```
%%time
# Realizar la validación cruzada
cv_scores = cross_val_score(xgb_clf, X_val, y_val, cv=5, scoring='accuracy')
```

```
CPU times: user 67d 9h 15min 54s, sys: 1h 30min 7s, total: 67d 10h 46min 2s
Wall time: 7h 7min 22s
```

```
cv_scores
```

```
array([0.99754686, 0.9983317 , 0.99764475, 0.99842983, 0.99842983])
```

Figura 83: Validación cruzada con datos de validación para XGBoost

```
cm = confusion_matrix(y_val, y_pred_val)
f = sns.heatmap(cm, annot=True, fmt='d')
```

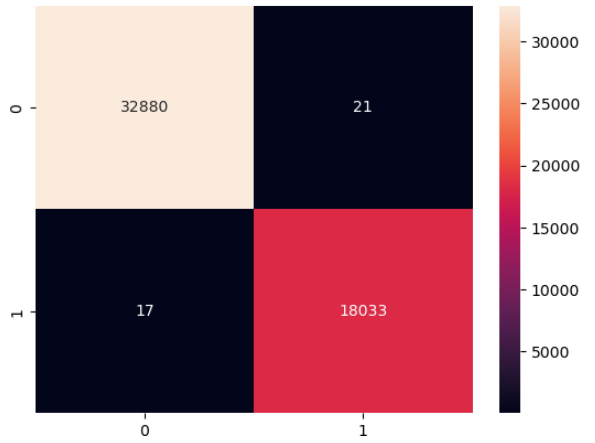


Figura 84: Matiz de confusión con datos de validación para XGBoost

3.2.1.5. Xgboost (clasificación de motivos de churn)

Dado el buen resultado obtenido previamente, se define el uso de Xgboost comoclasificador de los motivos de cancelación de los clientes.

Se utiliza optuna de hiperparámetros ahora lo que cambia es la función que se utiliza parala optimización Figura 85, dentro de los parámetros definidos para la optimización se tiene:

- Cantidad de árboles entre 1 y 6000
- Profundidad de los árboles entre 1 y 200
- Tasa de aprendizaje entre 0.00001 y 1
- Peso mínimo por nodo entre 1 y 100
- Reducción de ganancia 0.00001 y 1
- Subconjunto de entrenamiento entre 0.00001 y 1
- Porcentaje de características entre 0.00001 y 1
- L1 entre 0.00001 y 1

- L2 entre 0.00001 y 1
- Método de construcción histograma con uso de GPU
- Parámetro random_state para que el resultado sea reproducible

```
def objective(trial):
    param = {
        'max_depth': trial.suggest_int('max_depth', 1, 200),
        #'num_class': trial.suggest_int('num_class', 2, 2),
        'learning_rate': trial.suggest_float('learning_rate', 0.00001, 1.0),
        'n_estimators': trial.suggest_int('n_estimators', 1, 6000),
        'min_child_weight': trial.suggest_int('min_child_weight', 1, 100),
        'gamma': trial.suggest_float('gamma', 0.00001, 1.0),
        'subsample': trial.suggest_float('subsample', 0.00001, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.00001, 1.0),
        'reg_alpha': trial.suggest_float('reg_alpha', 0.00001, 1.0),
        'reg_lambda': trial.suggest_float('reg_lambda', 0.00001, 1.0),
        'random_state': trial.suggest_int('random_state', 1, 6000),
        'tree_method': 'gpu_hist'
    }

    model = xgb.XGBClassifier(**param)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    return 1 - accuracy_score(y_test, y_pred)
```

Figura 85: Función de optimización para XGBoost de clasificación de motivos de cancelación

En la Figura 86 se tiene:

- Ejecución de optuna con 2000 iteraciones
- Una línea por iteración en la que se muestra los parámetros utilizados y el resultado obtenido
- Muestra un detalle del tiempo de uso del procesamiento del server aquí tenemos dos temas importantes el tiempo de CPU y el wall time (tiempo transcurrido), el primero hace referencia a la ejecución tomando en cuenta todos los eventos que pudieron interrumpir o ralentizar la ejecución y el segundo hace referencia al tiempo real que toma la ejecución de las líneas de código, siendo los valores respectivamente 2 días 28 minutos 12 segundos y 13 horas 24 minutos y 1 segundos.

Finalmente se muestra el contenido de la variable best_params que contiene el resultado de la optimización, por cada parámetro definido en la función de optimización nos da el valor que debemos utilizar para obtener el mejor resultado, en este caso tenemos que se debe utilizar 5549 árboles, con una profundidad de 11, una tasa de aprendizaje de 0.01579308697551782, peso por nodo 6, reducción de ganancia 0.4940764587500138, subconjunto de 0.9149492688195568, porcentaje de características 0.4999309598337212, L1 de 0.7236105995620841, L2 de 0.0005024100023978129 y un random_state 3908.

```
%%time
study = optuna.create_study(direction='minimize', study_name='clasification')
study.optimize(objective, n_trials=2000)
```

```
79582, 'random_state': 4637}. Best is trial 1016 with value: 0.17768871075484305.
[I 2023-07-20 04:02:38,056] Trial 1996 finished with value: 0.18269873079492316 and parameters: {'max_depth': 13, 'learn
ing_rate': 0.0014120390236250693, 'n_estimators': 5700, 'min_child_weight': 5, 'gamma': 0.095123369774541, 'subsample':
0.8324019852234328, 'colsample_bytree': 0.5802304512553482, 'reg_alpha': 0.5506922057533363, 'reg_lambda': 0.04182935634
7903836, 'random_state': 4528}. Best is trial 1016 with value: 0.17768871075484305.
[I 2023-07-20 04:03:08,323] Trial 1997 finished with value: 0.1834621624200783 and parameters: {'max_depth': 64, 'learni
ng_rate': 0.0678957792230224, 'n_estimators': 5735, 'min_child_weight': 1, 'gamma': 0.5900309081703633, 'subsample': 0.8
147123877047671, 'colsample_bytree': 0.6392757272630935, 'reg_alpha': 0.4786642293368174, 'reg_lambda': 0.35357377003902
435, 'random_state': 4228}. Best is trial 1016 with value: 0.17768871075484305.
[I 2023-07-20 04:03:32,152] Trial 1998 finished with value: 0.17978814772401952 and parameters: {'max_depth': 7, 'learni
ng_rate': 0.03595859340423472, 'n_estimators': 5880, 'min_child_weight': 3, 'gamma': 0.5137863610639609, 'subsample': 0.
7437686053345326, 'colsample_bytree': 0.5997219636761183, 'reg_alpha': 0.6811442332369881, 'reg_lambda': 0.3687603412104
131, 'random_state': 4132}. Best is trial 1016 with value: 0.17768871075484305.
[I 2023-07-20 04:03:50,114] Trial 1999 finished with value: 0.20417024525240957 and parameters: {'max_depth': 16, 'learn
ing_rate': 0.3553260902316494, 'n_estimators': 5889, 'min_child_weight': 5, 'gamma': 0.6065226871060276, 'subsample': 0.
31388362547192256, 'colsample_bytree': 0.88241748791613, 'reg_alpha': 0.5208778188889173, 'reg_lambda': 0.12045571436898
946, 'random_state': 4271}. Best is trial 1016 with value: 0.17768871075484305.

CPU times: user 2d 14min 48s, sys: 13min 24s, total: 2d 28min 12s
Wall time: 13h 24min 1s
```

```
study.best_params
```

```
{'max_depth': 11,
 'learning_rate': 0.01579308697551782,
 'n_estimators': 5549,
 'min_child_weight': 6,
 'gamma': 0.4940764587500138,
 'subsample': 0.9149492688195568,
 'colsample_bytree': 0.4999309598337212,
 'reg_alpha': 0.7236105995620841,
 'reg_lambda': 0.0005024100023978129,
 'random_state': 3908}
```

Figura 86: Resultado de la ejecución de optuna para XGBoost de clasificación de motivos de cancelación

Se definió el modelo en base a los resultados de la optimización, el tiempo real de ejecución de esto fue de 63.2 microsegundos como se puede ver en la Figura 87.

```
%%time
## ---------- XGBoost model v1 ----------
## base run of model with default hyperparameters

xgb_clf = xgb.XGBClassifier(objective='multi:softmax',
                            num_class=3,
                            max_depth= 11,
                            learning_rate=0.01579308697551782,
                            min_child_weight= 6,
                            gamma= 0.4940764587500138,
                            subsample= 0.9149492688195568,
                            colsample_bytree= 0.4999309598337212,
                            reg_alpha= 0.7236105995620841,
                            reg_lambda= 0.0005024100023978129,
                            eval_metric=['merror','mlogloss'],
                            n_estimators=5549,
                            random_state= 3908
                            )
```

```
CPU times: user 0 ns, sys: 52 µs, total: 52 µs
Wall time: 63.2 µs
```

Figura 87: Configuración de XGBoost con los mejores parámetros entregados por optuna para la clasificación de motivos decancelación

En la figura 88 vemos el entrenamiento del modelo, mismo que tomo un tiempo de 54minutos 59 segundos y en la Figura 89 podemos ver que almacenamos el modelo almacenado.

```
%%time
xgb_clf.fit(X_train,
            y_train,
            verbose=1, # set to 1 to see xgb training round intermediate results
            eval_set=[(X_train, y_train), (X_test, y_test)]
           )
```

```
gloss:0.49497
[5541]  validation_0-merror:0.00010     validation_0-mlogloss:0.06869   validation_1-merror:0.17573     validation_1-mlo
gloss:0.49497
[5542]  validation_0-merror:0.00010     validation_0-mlogloss:0.06868   validation_1-merror:0.17564     validation_1-mlo
gloss:0.49496
[5543]  validation_0-merror:0.00010     validation_0-mlogloss:0.06868   validation_1-merror:0.17564     validation_1-mlo
gloss:0.49497
[5544]  validation_0-merror:0.00010     validation_0-mlogloss:0.06867   validation_1-merror:0.17559     validation_1-mlo
gloss:0.49497
[5545]  validation_0-merror:0.00010     validation_0-mlogloss:0.06867   validation_1-merror:0.17559     validation_1-mlo
gloss:0.49497
[5546]  validation_0-merror:0.00010     validation_0-mlogloss:0.06867   validation_1-merror:0.17564     validation_1-mlo
gloss:0.49498
[5547]  validation_0-merror:0.00010     validation_0-mlogloss:0.06866   validation_1-merror:0.17559     validation_1-mlo
gloss:0.49498
[5548]  validation_0-merror:0.00010     validation_0-mlogloss:0.06866   validation_1-merror:0.17564     validation_1-mlo
gloss:0.49498
CPU times: user 54min 50s, sys: 9.74 s, total: 55min
Wall time: 54min 59s
```

Figura 88: Ejecución del XGBoost con los mejores parámetros entregados por optuna

```
joblib.dump(xgb_clf, "xgb.pkl")
```

Figura 89: Almacenamiento del modelo XGBoost de clasificación de motivos de cancelación

Ahora utilizando el set de test en la Figura 90 y 91 se ve que al ejecutar la predicción tenemos como accuracy un valor de 82% bastante aceptable y en las Figura 92 se muestre gráficamente este resultado por medio de una matriz de confusión.

```
%%time
# preparing evaluation metric plots
results = xgb_clf.evals_result()
epochs = len(results['validation_0']['mlogloss'])
x_axis = range(0, epochs)

# xgboost 'mlogloss' plot
fig, ax = plt.subplots(figsize=(9,5))
ax.plot(x_axis, results['validation_0']['mlogloss'], label='Train')
ax.plot(x_axis, results['validation_1']['mlogloss'], label='Test')
ax.legend()
plt.ylabel('mlogloss')
plt.title('GridSearchCV XGBoost mlogloss')
plt.show()

# xgboost 'merror' plot
fig, ax = plt.subplots(figsize=(9,5))
ax.plot(x_axis, results['validation_0']['merror'], label='Train')
ax.plot(x_axis, results['validation_1']['merror'], label='Test')
ax.legend()
plt.ylabel('merror')
plt.title('GridSearchCV XGBoost merror')
plt.show()

## ---------- Model Classification Report ----------
## get predictions and create model quality report

y_pred = xgb_clf.predict(X_test)
```

Figura 90: Generación de métricas de XGBoost de clasificación de motivos de cancelación

```
--------------- Classification Report ---------------

              precision    recall  f1-score   support

           0       0.78      0.72      0.75      6081
           1       0.46      0.01      0.03       819
           2       0.84      0.92      0.88     14058

    accuracy                           0.82     20958
   macro avg       0.69      0.55      0.55     20958
weighted avg       0.81      0.82      0.81     20958

---------------------- XGBoost ----------------------
CPU times: user 38.9 s, sys: 214 ms, total: 39.1 s
Wall time: 732 ms
```

Figura 91: Métricas de XGBoost de clasificación de motivos de cancelación

```
cm = confusion_matrix(y_test, y_pred)
f = sns.heatmap(cm, annot=True, fmt='d')
```

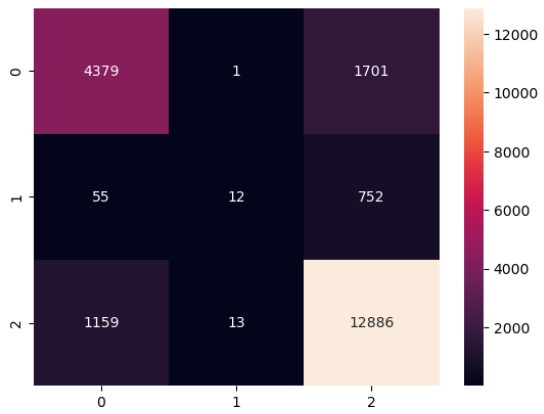


Figura 92: Matriz de confusión de XGBoost de clasificación de motivos de cancelación

Con otro set de datos que no se sido visto por el modelo realizamos la validación cruzada Figura 93, aquí se ve que el resultado del accuracy está en un rango del 80 al 81% y en las Figura 94 se muestre gráficamente este resultado por medio de una matriz de confusión.

```
%%time
# Realizar la validación cruzada
cv_scores = cross_val_score(xgb_clf, X_val, y_val, cv=5, scoring='accuracy')
```

```
CPU times: user 50min 44s, sys: 12.4 s, total: 50min 57s
Wall time: 50min 57s
```

```
cv_scores
```

```
array([0.81037567, 0.81156828, 0.81658217, 0.80316135, 0.81181032])
```

Figura 93: Validación cruzada con datos de validación para XGBoost de clasificación de motivos de cancelación

```
cm = confusion_matrix(y_val, y_pred_val)
f = sns.heatmap(cm, annot=True, fmt='d')
```

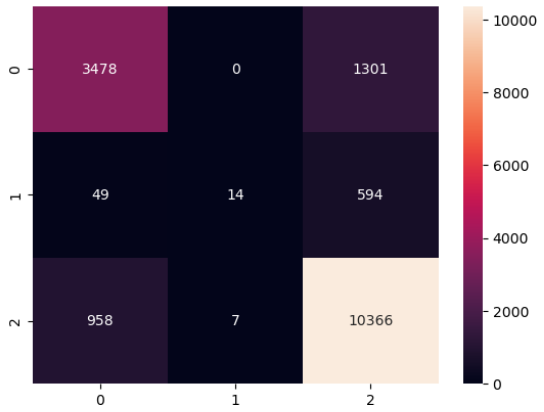


Figura 94: Matiz de confusión con datos de validación para XGBoost de clasificación de motivos de cancelación

3.2.2. Módulos del sistema

La herramienta web es un cuadro de mando que en base a segmentación y clasificación ejecutada por el modelo se determinará la posibilidad de que un cliente abandone la empresa y se enfoca en aquellos cuya motivación son los problemas técnicos.

La herramienta cuenta con cinco módulos que le permiten mostrar al usuario el resultado de la predicción y realizar una exploración de los datos, en la Figura 96 se muestra los diferentes módulos y opciones que se tiene.

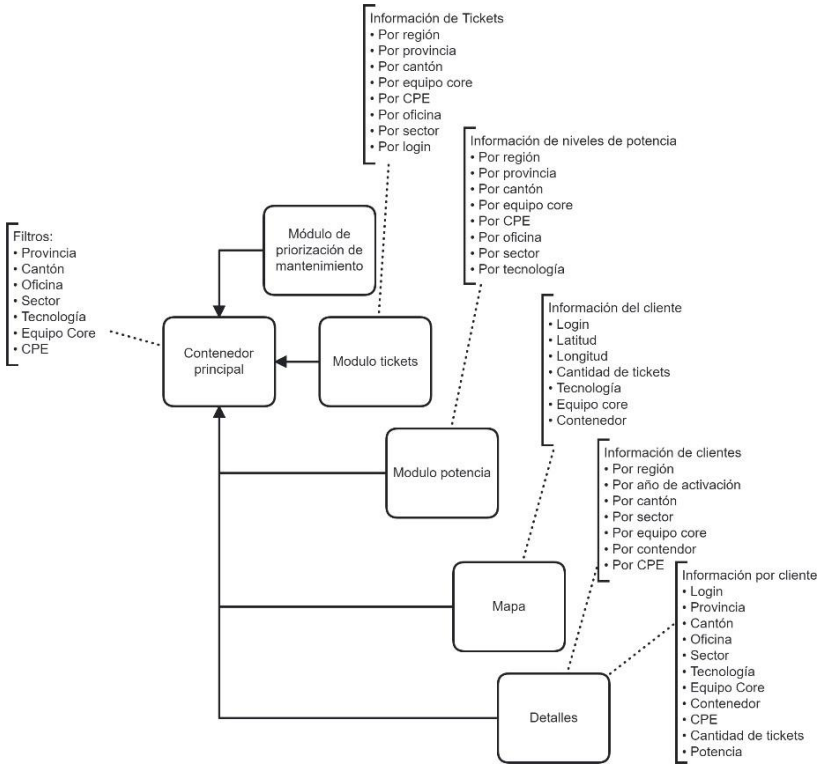


Figura 95: Esquema de los módulos del sistema

3.2.2.1. Módulo de priorización de mantenimiento

En base a la predicción del modelo se determina a aquellos clientes que tienen una probabilidad alta de abandono, de ellos se identifica la infraestructura de la cual dependen, esta será la que se presente al usuario y está dividida en elementos activos y pasivos, para que sean considerados dentro de la planificación, al ser una sugerencia será un apoyo para el trabajo del personal técnico, y la persona podrá acoger o no la recomendación.

En la Figura 96 podemos visualizar como se ve esta pantalla, la que consta de 4 partes que son: oficina, tecnología, equipo core y contenedor.

SECTOR	EQUIPO_CORE	CLIENTES
SECTOR 012	EQP-CORE-328	43
SECTOR 141	EQP-CORE-219	41
SECTOR 049	EQP-CORE-061	39
SECTOR 004	EQP-CORE-384	34
SECTOR 091	EQP-CORE-393	32
SECTOR 023	EQP-CORE-026	31
SECTOR 045	EQP-CORE-119	31
SECTOR 082	EQP-CORE-106	26
SECTOR 012	EQP-CORE-044	22
SECTOR 041	EQP-CORE-051	22
SECTOR 127	EQP-CORE-245	22
SECTOR 001	EQP-CORE-098	21
SECTOR 044	EQP-CORE-055	20
SECTOR 122	EQP-CORE-169	20
SECTOR 012	EQP-CORE-046	19
SECTOR 047	EQP-CORE-127	19
SECTOR 030	EQP-CORE-034	18
SECTOR 075	EQP-CORE-097	18
SECTOR 040	EQP-CORE-050	17
SECTOR 073	EQP-CORE-103	17
SECTOR 102	EQP-CORE-138	17
SECTOR 050	EQP-CORE-340	17
SECTOR 074	EQP-CORE-095	16

SECTOR	CONTENEDOR	CLIENTES
SECTOR 083	CONT-CORE-CA-1247	5
SECTOR 061	CONT-CORE-CA-0076	4
SECTOR 067	CONT-CORE-CA-0280	4
SECTOR 048	CONT-CORE-CA-0780	4
SECTOR 269	CONT-CORE-CA-1503	4
SECTOR 083	CONT-CORE-CA-3166	4
SECTOR 003	CONT-CORE-CA-0003	3
SECTOR 038	CONT-CORE-CA-0043	3
SECTOR 074	CONT-CORE-CA-0102	3
SECTOR 062	CONT-CORE-CA-0142	3
SECTOR 009	CONT-CORE-CA-0227	3
SECTOR 117	CONT-CORE-CA-0289	3
SECTOR 062	CONT-CORE-CA-0398	3
SECTOR 050	CONT-CORE-CA-0479	3
SECTOR 036	CONT-CORE-CA-0745	3
SECTOR 307	CONT-CORE-CA-1014	3
SECTOR 012	CONT-CORE-CA-1373	3
SECTOR 050	CONT-CORE-CA-1457	3
SECTOR 016	CONT-CORE-CA-1944	3
SECTOR 144	CONT-CORE-CA-2485	3
SECTOR 001	CONT-CORE-CA-0001	2
SECTOR 002	CONT-CORE-CA-0002	2
SECTOR 023	CONT-CORE-CA-0019	2

Figura 96: Pantalla del módulo de priorización de mantenimiento

En la Figura 97, se tiene por cada oficina la cantidad de clientes que tienen la intención de cancelar los servicios, esto le indica gráficamente al personal operativo donde aplicar sus esfuerzos, en este caso la oficina Quito con 1659 clientes.

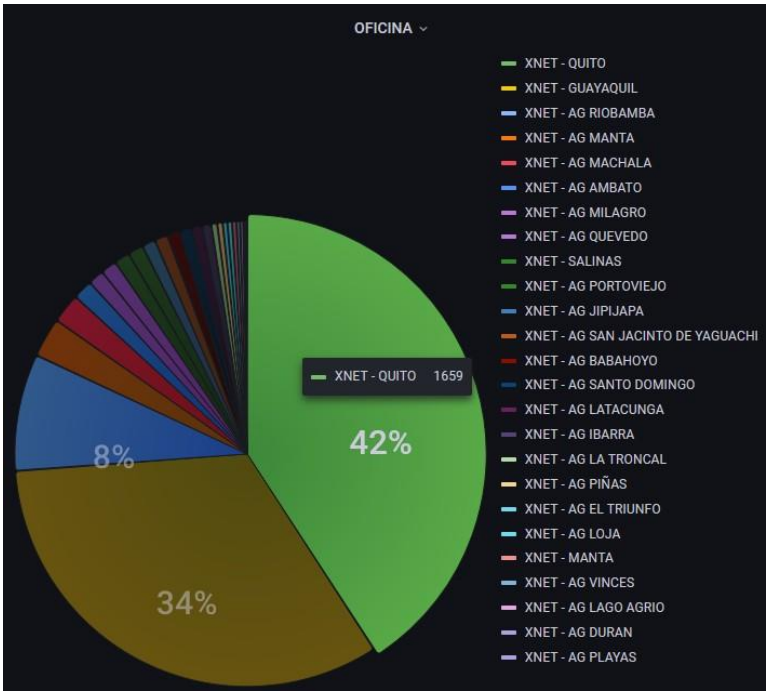


Figura 97: Posible cancelación de clientes por oficina

En la Figura 98, se muestra clientes por tipo de tecnología, aquí se puede observar que hay que la mayor afectación corresponde a TECK 01 con 2642 clientes.

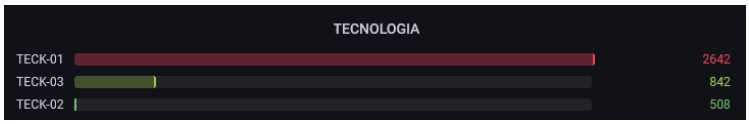


Figura 98: Posible cancelación de clientes por tecnología

En la Figura 99, se muestra dos tablas en las que por sector se muestre el equipo core y el contenedor con mayor afectación, la data se encuentra ordenada de mayor a menor, mostrando que en el SECTOR 012 se tiene el equipo (EQP-CORE-328) con 43 clientes que probablemente cancelen el servicio, también se ve que en el SECTOR 083 el contenedor (CONT- CORE-CA-1247) posee 5 clientes con tendencia a cancelar, toda esta data es la guía para que el personal de operaciones pueda planificar sus mantenimientos.

EQUIPO CORE

SECTOR	EQUIPO_CORE	CLIENTES
SECTOR 012	EQP-CORE-328	43
SECTOR 141	EQP-CORE-219	41
SECTOR 049	EQP-CORE-061	39
SECTOR 004	EQP-CORE-384	34
SECTOR 091	EQP-CORE-393	32
SECTOR 023	EQP-CORE-026	31
SECTOR 045	EQP-CORE-119	31
SECTOR 082	EQP-CORE-106	26
SECTOR 012	EQP-CORE-044	22
SECTOR 041	EQP-CORE-051	22
SECTOR 127	EQP-CORE-245	22
SECTOR 001	EQP-CORE-098	21
SECTOR 044	EQP-CORE-055	20
SECTOR 122	EQP-CORE-169	20
SECTOR 012	EQP-CORE-046	19
SECTOR 047	EQP-CORE-127	19
SECTOR 030	EQP-CORE-034	18
SECTOR 075	EQP-CORE-097	18
SECTOR 040	EQP-CORE-050	17
SECTOR 073	EQP-CORE-103	17
SECTOR 102	EQP-CORE-138	17
SECTOR 050	EQP-CORE-340	17
SECTOR 074	EQP-CORE-095	16

CONTENEDOR

SECTOR	CONTENEDOR	CLIENTES
SECTOR 083	CONT-CORE-CA-1247	5
SECTOR 061	CONT-CORE-CA-0076	4
SECTOR 067	CONT-CORE-CA-0280	4
SECTOR 048	CONT-CORE-CA-0780	4
SECTOR 269	CONT-CORE-CA-1503	4
SECTOR 083	CONT-CORE-CA-3166	4
SECTOR 003	CONT-CORE-CA-0003	3
SECTOR 038	CONT-CORE-CA-0043	3
SECTOR 074	CONT-CORE-CA-0102	3
SECTOR 062	CONT-CORE-CA-0142	3
SECTOR 009	CONT-CORE-CA-0227	3
SECTOR 117	CONT-CORE-CA-0289	3
SECTOR 062	CONT-CORE-CA-0398	3
SECTOR 050	CONT-CORE-CA-0479	3
SECTOR 036	CONT-CORE-CA-0745	3
SECTOR 307	CONT-CORE-CA-1014	3
SECTOR 012	CONT-CORE-CA-1373	3
SECTOR 050	CONT-CORE-CA-1457	3
SECTOR 016	CONT-CORE-CA-1944	3
SECTOR 144	CONT-CORE-CA-2485	3
SECTOR 001	CONT-CORE-CA-0001	2
SECTOR 002	CONT-CORE-CA-0002	2
SECTOR 023	CONT-CORE-CA-0019	2

Figura 99: Posible cancelación de clientes por equipo core y contenedor

3.2.2.2. Módulo tickets

Aquí el usuario tendrá una visión general del soporte, representado por los tickets que se han gestionado como se muestra en la Figura 100, este dashboard está compuesto por seis pates que son: tickets por región, provincia, cantón, equipo core, CPE, oficina, sector y login, además posee como filtros provincia, cantón, oficina, sector, equipo core y CPE, que permiten interactúa con la data que se

presenta.

Figura 100: Pantalla del módulo del módulo tickets

En la Figura 101, se muestra los tickets por región, siendo muy evidente de Quito es laregión más afectada en este sentido con 38138 tickets vs 11098 que tiene Guayaquil.

Figura 101: Tickets por región

En la Figura 102, se muestra los tickets a nivel de provincia siendo Pichincha la que poseela mayor cantidad con 13707 tickets.

Figura 102: Tickets por provincia

En la Figura 103, se muestra los tickets a nivel de cantón siendo Quito la que posee lamayor cantidad con 12625 tickets.

Figura 103: Tickets por cantón

En la Figura 104, se muestra los tickets a nivel de equipo core siendo EQP-CORE-219 elque posee la mayor cantidad con 540 tickets.

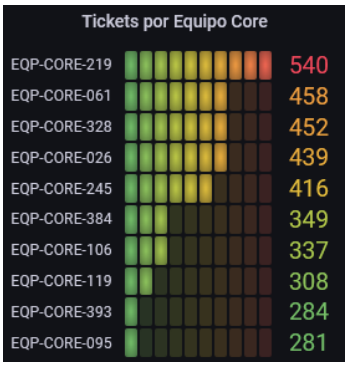


Figura 104: Tickets por cantón por equipo core

En la Figura 105, se muestra los tickets a nivel de CPE siendo EG8M8145V5G06 el queposee la mayor cantidad con 17855 tickets.

Figura 105: Tickets por CPE

En la Figura 106, se muestra los tickets a nivel de oficina siendo XNET-GUAYAQUIL laque posee la mayor cantidad con 19.9k tickets

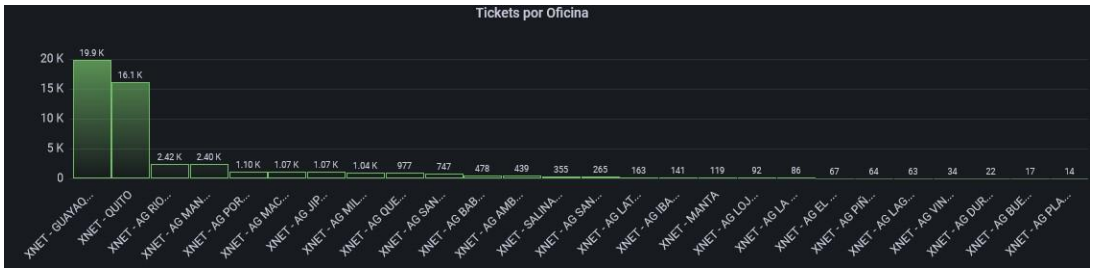


Figura 106: Tickets por oficina

En la Figura 107, se muestra los tickets a nivel de sector siendo SECTOR 012 el que posee la mayor cantidad con 2.72k tickets.

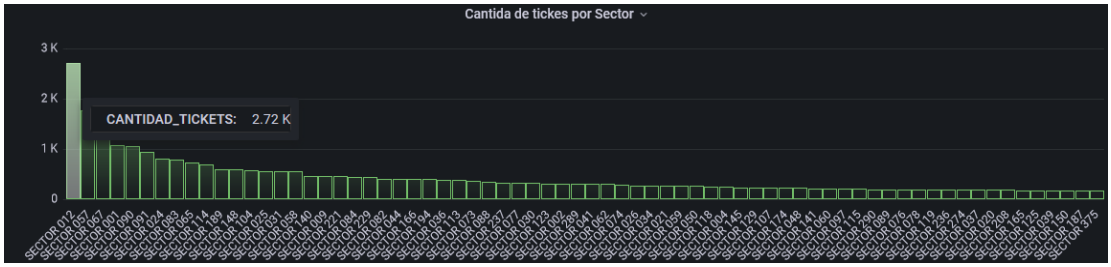


Figura 107: Tickets por sector

En la Figura 108, se muestra los tickets a nivel de login siendo Clie-xnet-0371 el que posee la mayor cantidad con 44 tickets.

Figura 108: Tickets por login

3.2.2.3. Módulo potencia

Muestra rangos de potencia, valor de potencia promedio por ubicación geográfica Figura 109, el dashboard está compuesto por nueve partes que son: potencia promedio nacional, por provincia, cantón, equipo core, CPE, oficina, sector y tecnología, además posee como filtros provincia, cantón, oficina, sector, equipo core y CPE, que permiten interactúa con la data que sepresenta.

Figura 109: Pantalla del módulo del módulo potencia

En la Figura 110, se muestra la potencia promedio a nivel nacional que actualmente está en -22.8 valor bastante bueno a nivel técnico.

Figura 110: Promedio potencia nacional

En la Figura 111, se muestra la potencia máxima, promedio y mínima por región, siendo para Quito –31.5, -22.6, -15.9 y para Guayaquil –33, -23.6, -17.2, en primera instancia se observa que los valores para Quito son más bajos que los de Guayaquil, a pesar de que en el valor promedio Guayaquil tiene un valor más alto, pero aun así dentro del rango adecuado que a interno se ha considerado que es entre –20 a –24.

Figura 111: Potencia promedio por región

En la Figura 112, se muestra la potencia promedio a nivel de provincia siendo CAÑAR la que posee la potencia más baja con un –24.1, valor fuera del rango aceptable.

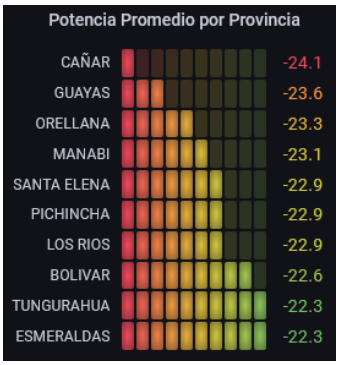


Figura 112: Potencia promedio por provincia

En la Figura 113, se muestra la potencia promedio a nivel de cantón siendo NARANJITO la que posee la potencia más baja con un –25.2, valor fuera del rango aceptable.

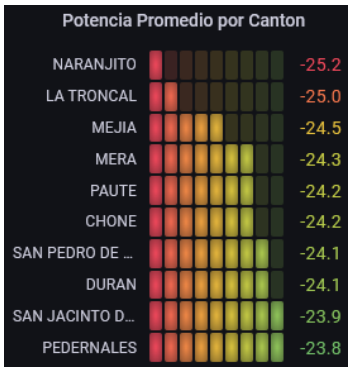


Figura 113: Potencia promedio por cantón

En la Figura 114, se muestra la potencia promedio a nivel de equipo core siendo EQP- CORE-736 la que posee la potencia más baja con un –33, valor no solo fuera del rango aceptable, sino que tiene un valor crítico que afecta al cliente.

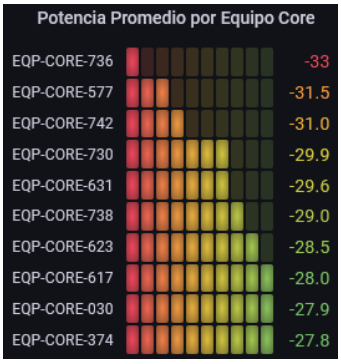


Figura 114: Potencia promedio por equipo core

En la Figura 115, se muestra la potencia promedio a nivel de CPE siendo EG8M8245H5G05 la que posee la potencia más baja con un –23.6, valor dentro del rango aceptable.

Figura 115: Potencia promedio por CPE

En la Figura 116, se muestra la potencia promedio a nivel de oficina siendo XNET-MANTA la que posee la potencia más baja con un –25.2, valor bajo del rango aceptable.

Figura 116: Potencia promedio por oficina

En la Figura 117, se muestra la potencia promedio a nivel de sector siendo SECTOR 544 la que posee la potencia más baja con un –31, valor no solo fuera del rango aceptable, sino que tiene un valor crítico que afecta al cliente.

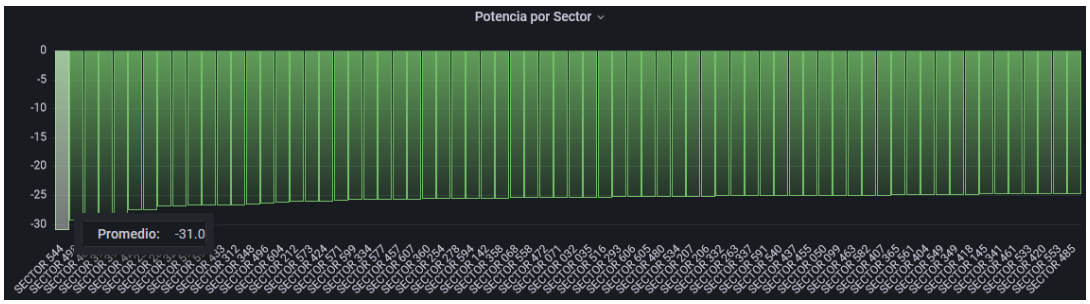


Figura 117: Potencia promedio por sector

En la Figura 118, se muestra la potencia máxima, promedio y mínima por tecnología, siendo para TECK-01 -33, -23, -15.9, TECK-02 -30.5, -21.9, -16.1 y para TECK-03 -30.9, -22.7,

-17.2, en primera instancia se observa que los valores entre las diferentes tecnologías son similares, el valor promedio para los tres está dentro del rango adecuado que a interno se ha considerado que es entre –20 a –24.

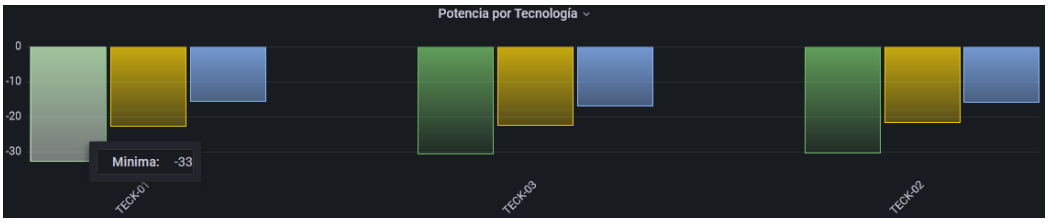


Figura 118: Potencia promedio por tecnología

3.2.2.4. Módulo geográfico (mapa)

Aquí tenemos una visión de cómo están repartidos los clientes con alta probabilidad de abandono, a lo largo de la del territorio nacional, Figura 119.

El mapa permite que usuario interaccione con el pudiendo acercarse, alejarse o moverlo según requiera y por cada cliente, en la Figura 119 se puede observar la información que se muestra al seleccionar un cliente, en este caso login, latitud, longitud, cantidad de tickets, tecnología, equipo_core, contendor del que depende y la potencia.

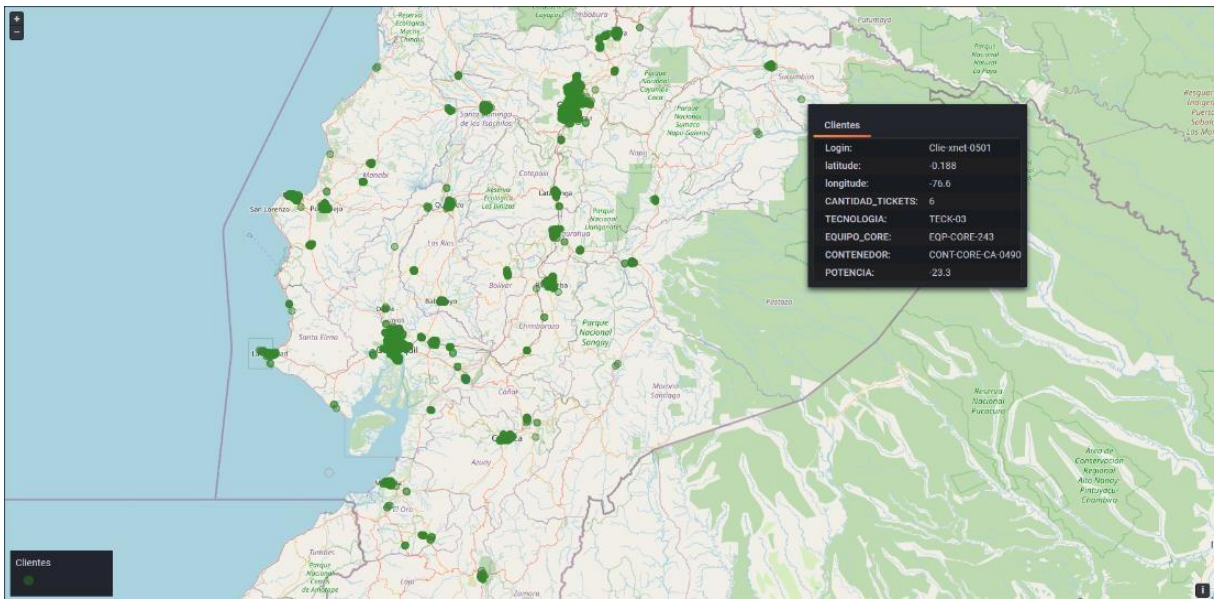


Figura 119: Pantalla del módulo del módulo geográfico

3.2.2.5. Módulo de detalles

En este módulo se tiene una base con el detalle de los clientes, esta dashboard tiene nueve partes que son: clientes por región, año, provincia, cantón, sector, equipo ocre, contenedor, CPE y tabla resumen de clientes como se ve en la Figura 120.

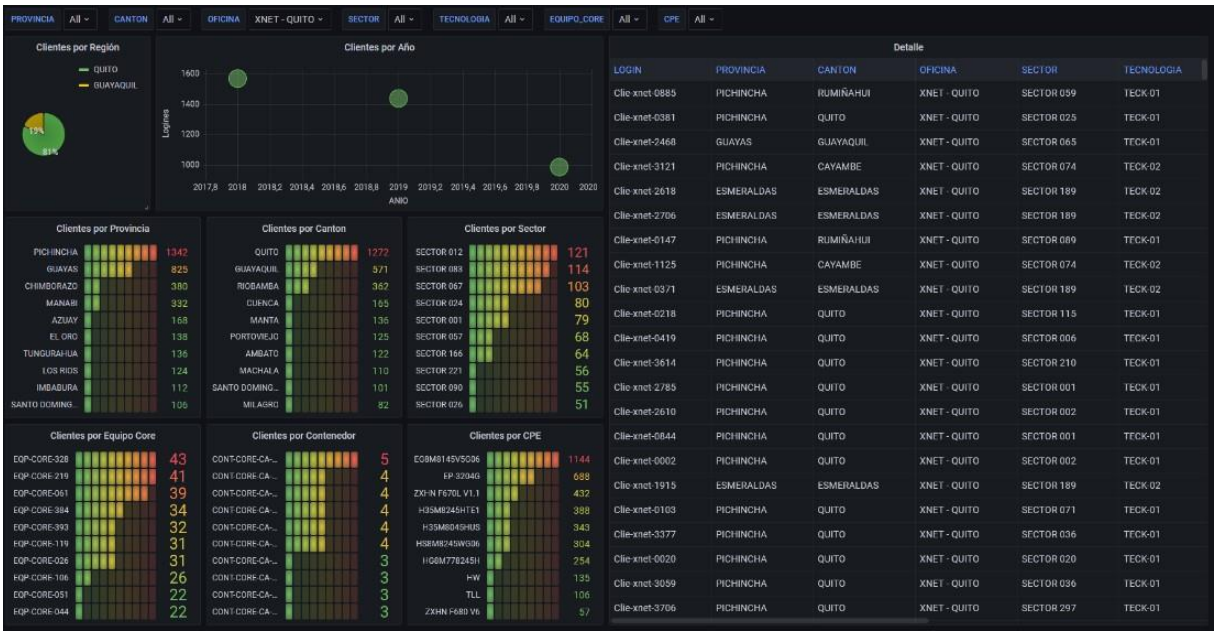


Figura 120: Pantalla del módulo del módulo detalles

En la Figura 121, se muestra la cantidad de clientes por región teniendo en Quito 3.25k correspondiente al 81% del total y en Guayaquil 146 que corresponde al 19% del total.

Figura 121: Cantidad de clientes por región

En la Figura 122, se muestra la cantidad de clientes por año en el rango de 2018 al 2020, con los valores de 1568, 1437 y 987 respectivamente lo que indica que la cantidad de clientes que han cancelado está a la baja.

Figura 122: Cantidad de clientes que cancelan por año

En la Figura 123, se muestra la cantidad de clientes por provincia siendo Pichincha la que posee la

mayor cantidad de clientes con 1342.

Figura 123: Cantidad de clientes por provincia

En la Figura 124, se muestra la cantidad de clientes por cantón siendo Quito la que poseela mayor cantidad de clientes con 1272.

Figura 124: Cantidad de clientes por provincia

En la Figura 125, se muestra la cantidad de clientes por sector siendo SECTOR 012 elque posee la mayor cantidad de clientes con 121.

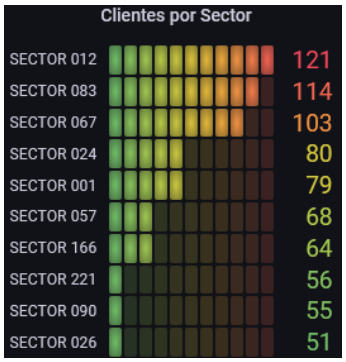


Figura 125: Cantidad de clientes por provincia

En la Figura 126, se muestra la cantidad de clientes por equipo core siendo EQP-CORE-238 el que posee la mayor cantidad de clientes con 43.

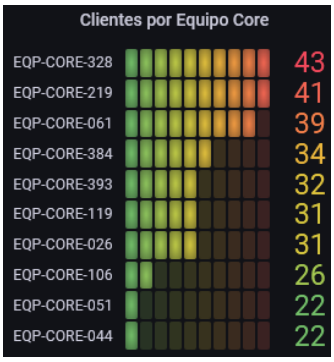


Figura 126: Cantidad de clientes por equipo core

En la Figura 127, se muestra la cantidad de clientes por contenedor siendo CONT-CORE-CA-1247 el que posee la mayor cantidad de clientes con 5.

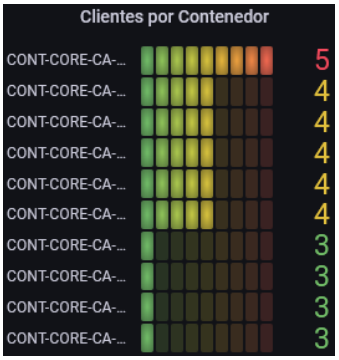


Figura 127: Cantidad de clientes por contenedor

En la Figura 128, se muestra la cantidad de clientes por CPE siendo Pichincha la que posee la mayor cantidad de clientes con 1144.

Figura 128: Cantidad de clientes por CPE

En la Figura 129 y 130, se muestra el detalle de los clientes, aquí se tiene una tabla que consta de las columnas login, provincia, cantón, oficina, sector, tecnología, equipo core, contenedor, CPE, cantidad_tickets y potencia, por ejemplo, de se tiene Clie-xnet-0885,

Pichincha, Rumiñahui, XNET-QUITO, SECTOR 059, TECK-01, EQP-CORE464, CONT-CORE-CA-0858, EG8M8145V5GC06, 57, -25.4.

Detalle

LOGIN	PROVINCIA	CANTON	OFICINA	SECTOR	TECNOLOGIA
Clie-xnet-0885	PICHINCHA	RUMIÑAHUI	XNET - QUITO	SECTOR 059	TECK-01
Clie-xnet-0381	PICHINCHA	QUITO	XNET - QUITO	SECTOR 025	TECK-01
Clie-xnet-2468	GUAYAS	GUAYAQUIL	XNET - QUITO	SECTOR 065	TECK-01
Clie-xnet-3121	PICHINCHA	CAYAMBE	XNET - QUITO	SECTOR 074	TECK-02
Clie-xnet-2618	ESMERALDAS	ESMERALDAS	XNET - QUITO	SECTOR 189	TECK-02
Clie-xnet-2706	ESMERALDAS	ESMERALDAS	XNET - QUITO	SECTOR 189	TECK-02
Clie-xnet-0147	PICHINCHA	RUMIÑAHUI	XNET - QUITO	SECTOR 089	TECK-01
Clie-xnet-1125	PICHINCHA	CAYAMBE	XNET - QUITO	SECTOR 074	TECK-02
Clie-xnet-0371	ESMERALDAS	ESMERALDAS	XNET - QUITO	SECTOR 189	TECK-02
Clie-xnet-0218	PICHINCHA	QUITO	XNET - QUITO	SECTOR 115	TECK-01
Clie-xnet-0419	PICHINCHA	QUITO	XNET - QUITO	SECTOR 006	TECK-01
Clie-xnet-3614	PICHINCHA	QUITO	XNET - QUITO	SECTOR 210	TECK-01
Clie-xnet-2785	PICHINCHA	QUITO	XNET - QUITO	SECTOR 001	TECK-01
Clie-xnet-2610	PICHINCHA	QUITO	XNET - QUITO	SECTOR 002	TECK-01
Clie-xnet-0844	PICHINCHA	QUITO	XNET - QUITO	SECTOR 001	TECK-01
Clie-xnet-0002	PICHINCHA	QUITO	XNET - QUITO	SECTOR 002	TECK-01
Clie-xnet-1915	ESMERALDAS	ESMERALDAS	XNET - QUITO	SECTOR 189	TECK-02
Clie-xnet-0103	PICHINCHA	QUITO	XNET - QUITO	SECTOR 071	TECK-01
Clie-xnet-3377	PICHINCHA	QUITO	XNET - QUITO	SECTOR 036	TECK-01
Clie-xnet-0020	PICHINCHA	QUITO	XNET - QUITO	SECTOR 020	TECK-01
Clie-xnet-3059	PICHINCHA	QUITO	XNET - QUITO	SECTOR 036	TECK-01
Clie-xnet-3706	PICHINCHA	QUITO	XNET - QUITO	SECTOR 297	TECK-01

Figura 129: Tabla detalle por cliente parte 1

Detalle

CNOLOGIA	EQUIPO_CORE	CONTENEDOR	CPE	CANTIDAD_TICKETS	POTENCIA
CK-01	EQP-CORE-464	CONT-CORE-CA-0858	EG8M8145V5G06	57	-25.4
CK-01	EQP-CORE-268	CONT-CORE-CA-0373	HS8M8245WG06	57	-24.8
CK-01	EQP-CORE-185	CONT-CORE-CA-2333	EG8M8145V5G06	52	-30
CK-02	EQP-CORE-290	CONT-CORE-CA-2448	ZXHN F670L V1.1	48	-21.3
CK-02	EQP-CORE-021	CONT-CORE-CA-2464	ZXHN F670L V1.1	46	-22.4
CK-02	EQP-CORE-583	CONT-CORE-CA-2546	ZXHN F670L V1.1	45	-28.9
CK-01	EQP-CORE-124	CONT-CORE-CA-0140	EG8M8145V5G06	45	-22.4
CK-02	EQP-CORE-071	CONT-CORE-CA-1085	ZXHN F670L V1.1	44	-21.6
CK-02	EQP-CORE-264	CONT-CORE-CA-0363	ZXHN F670L V1.1	44	101
CK-01	EQP-CORE-042	CONT-CORE-CA-0211	HS8M8245WG06	44	-19.6
CK-01	EQP-CORE-280	CONT-CORE-CA-0373	EG8M8145V5G06	43	-26.2
CK-01	EQP-CORE-001	CONT-CORE-CA-3364	EG8M8145V5G06	42	-23.6
CK-01	EQP-CORE-079	CONT-CORE-CA-2617	H35M8245HTE1	42	-25.7
CK-01	EQP-CORE-219	CONT-CORE-CA-2458	EG8M8145V5G06	42	-24.1
CK-01	EQP-CORE-447	CONT-CORE-CA-0819	HS8M8245WG06	42	-22.5
CK-01	EQP-CORE-003	CONT-CORE-CA-0002	HS8M8245WG06	42	-21.9
CK-02	EQP-CORE-511	CONT-CORE-CA-1823	ZXHN F670L V1.1	39	-22.3
CK-01	EQP-CORE-092	CONT-CORE-CA-0096	EG8M8145V5G06	39	-25.5
CK-01	EQP-CORE-522	CONT-CORE-CA-2861	HG8M778245H	38	-21.6
CK-01	EQP-CORE-021	CONT-CORE-CA-0016	EG8M8145V5G06	38	-25.8
CK-01	EQP-CORE-205	CONT-CORE-CA-2861	EG8M8145V5G06	37	-23.2
CK-01	EQP-CORE-188	CONT-CORE-CA-3452	HG8M778245H	36	-20.9

Figura 130: Tabla detalle por cliente parte 2

3.3. Infraestructura para procesamiento y almacenamiento

En la Figura 131 se puede observar la infraestructura requerida para la ingesta, entrenamiento, despliegue y visualización, enfocada en el hoy como primera interacción generando una primera mejora al proceso actual.

Aquí se puede ver que tenemos cuatro fuentes de datos, tres de ellas son esquemas Oracle del ERP y la cuarta es un Mongo que guarda el dato de nivel de potencia por cada equipo terminal del cliente, este último dato es el que puede carecer de información, esto porque el cliente puede apagar el equipo en el momento de la recolección de datos.

Python se utiliza para el preprocesamiento, limpieza de datos, generación del modelo y entrenamiento del modelo, en el despliegue el resultado de la predicción es la base que utiliza la herramienta de visualizaciones para mostrar el cuadro de mando con los resultados que utilizará el personal de operaciones.

Con esto en producción se mide la eficacia del modelo y en caso de ser necesario, se realiza un ajuste al modelo.

Ahora si pensamos en a futuro se debe considerar:

- Generación de APIs para acceso a la data de los diferentes esquemas de bases de datos.
- Utilizar para almacenamiento una base de datos Cassandra o similar para un rendimientoóptimo
- Posibilidad de incorporar a la herramienta de inteligencia de negocio de la empresa este aplicativo.
- Notificaciones bajo condiciones específicas

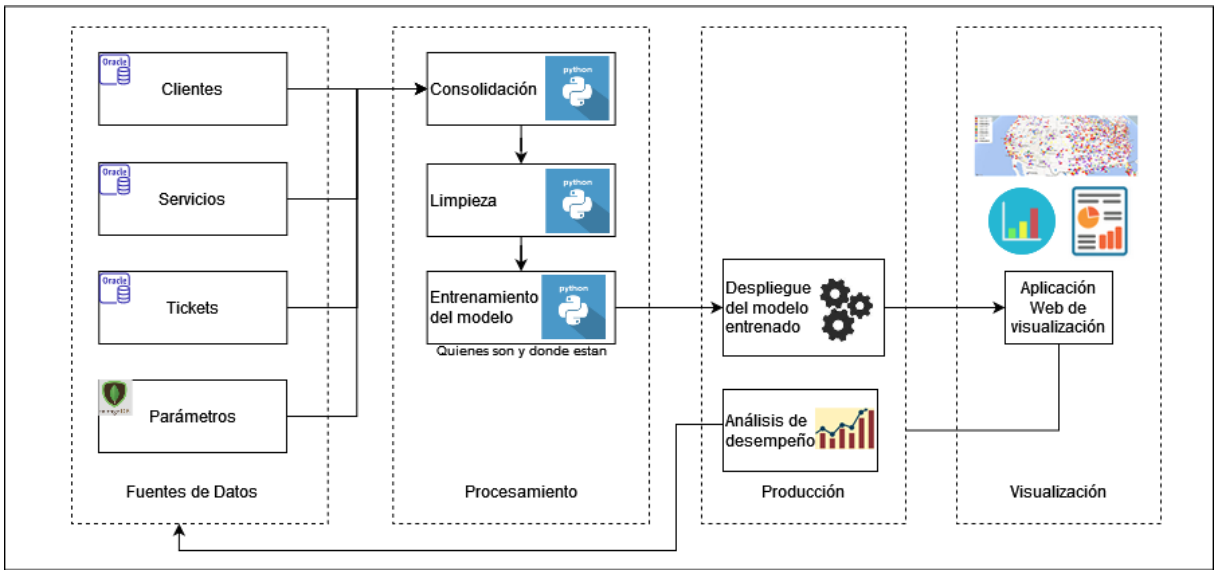


Figura 131: Esquema de la solución

3.4. Plataformas y prototipos de visualización

Se inició con una implementación local, en la que se contaba con una laptop con un procesador Core i7 de 8 núcleos, 64 Gb de memoria RAM y un disco duro de 2TB, aquí se tiene instalado Jupyter Notebook, Mysql y Grafana.

En este caso los datos se encuentran en el disco duro de la computadora y se los carga utilizando Jupyter Notebook, en esta misma herramienta se entrena el modelo y se genera como resultado un archivo CSV que contiene el resultado de la predicción y el detalle de la infraestructura que debe ser considerada para el mantenimiento.

Ahora bien, para tenerlo en producción se utiliza como entorno un ambiente de un DGX A-100 segmentado a 4GPUs y 1 TB en RAM, y para la visualización se tiene Grafana ejecutando sobre infraestructura virtual todo esto dentro de los centros de datos de la empresa.

El aplicativo es alcanzado únicamente desde la red privada de la empresa o por medio de una VPN que cuente con los permisos adecuados, esta restricción es propia de la empresa y parte de las políticas de seguridad de la información que se tienen certificadas bajo norma ISO27001.

Como se indica en el punto 3.3 al estar dentro del ecosistema de la empresa, el desarrolló y uso de APIs se debe enfocar en definir qué información se requiere de cada fuente disponible y contar con la autorización del dueño de ese activo.

3.5. Métricas y comunicación de resultados

Para la validación de los modelos indicados en el apartado 3.2 se utiliza datos que no han sido vistos por el modelo y se aplican las siguientes métricas:

Matriz de confusión: es una representación en forma de matriz de los resultados predichos vs los resultados reales (Singh Chauhan, 2020), en la Figura 132 se muestra como está conformada esta matriz.

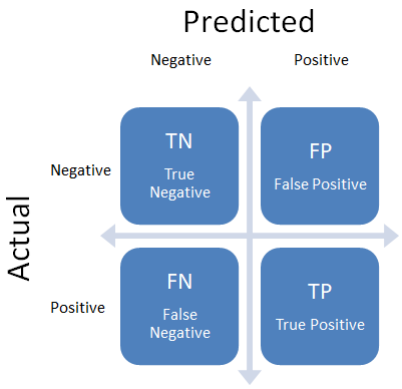


Figura 132: Detalle de la matriz de confusión

TP: valores verdaderos predichos como verdaderos (Singh Chauhan, 2020).

TN: valores falsos predichos como falsos (Singh Chauhan, 2020).

FP: valores falsos predichos como verdaderos (Singh Chauhan, 2020).

FN: valores verdaderos predichos como falsos (Singh Chauhan, 2020).

Accuracy: Cantidad de elementos clasificados de manera correcta (verdaderos y falsos)(Singh Chauhan, 2020).

Recall: Frecuencia en la que un valor verdadero es catalogado como verdadero (SinghChauhan, 2020).

F1: combinación de accuracy y recall, en uno solo valor utilizando la media armónica enel cual el

mejor valor está en 1

y el peor en 0 (Singh Chauhan, 2020).

En este punto se muestra por cada modelo los resultados con data que no ha sido vista por el modelo, utilizando los ocho sets de datos.

3.5.1. Codificar automático (autoencoder)

En las figuras 133 y 134, se puede ver el accuracy y la matriz de confusión de la ejecución con cada set de datos, teniendo como resultados que accuracy para 45 columnas y toda la data es 0.57, 40 columnas y toda la data es 0.58, 36 columnas y toda la data es 0.59, 31 columnas y toda la data es 0.62, 45 columnas y data filtrada es 0.58, 40 columnas y data filtrada es 0.59, 36 columnas y data filtrada es 0.67, 31 columnas y data filtrada es 0.59, por lo que el mejor resultado corresponde a 36 columnas y data filtrada con un 0.67.

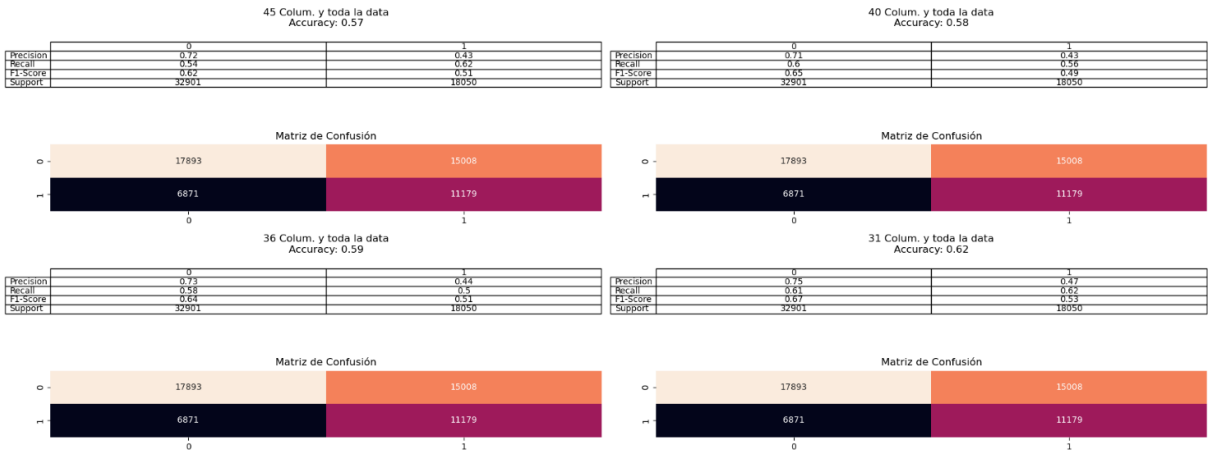


Figura 133: Resultados del autoencoder con data nueva parte 1

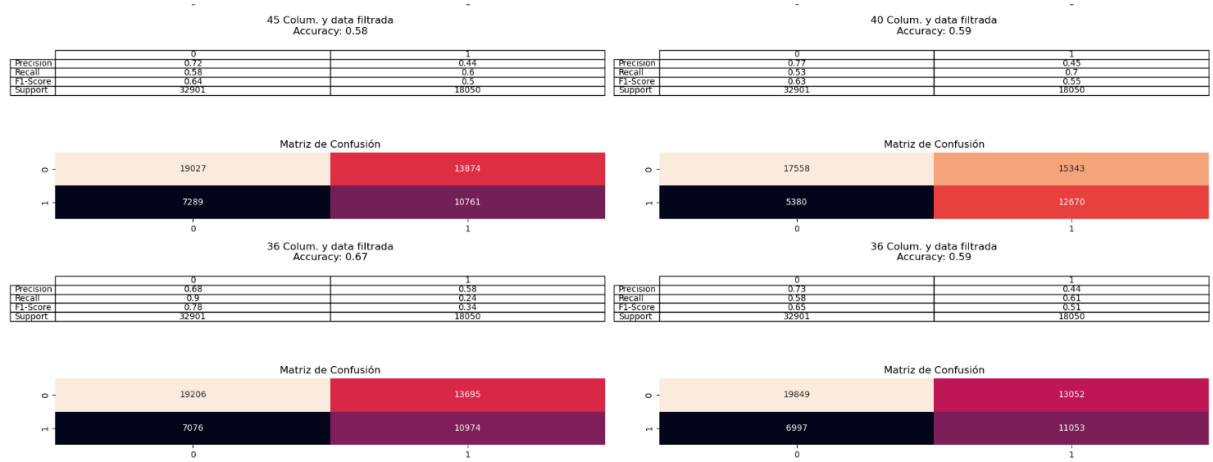


Figura 134: Resultados del autoencoder con data nueva parte 2

3.5.2. Máquinas de vectores de soporte (SVM)

En las figuras 135 y 136, se puede ver el accuracy y la matriz de confusión de la ejecución con cada set de datos, teniendo como resultados que accuracy para 45 columnas y toda la data es 0.86, 40 columnas y toda la data es 0.87, 36 columnas y toda la data es 0.91, 31 columnas y toda la data es 0.89, 45 columnas y data filtrada es 0.70, 40 columnas y data filtrada es 0.82, 36 columnas y data filtrada es 0.89, 31 columnas y data filtrada es 0.82, por lo que el mejor resultado corresponde a 36 columnas y toda la data con un 0.91.

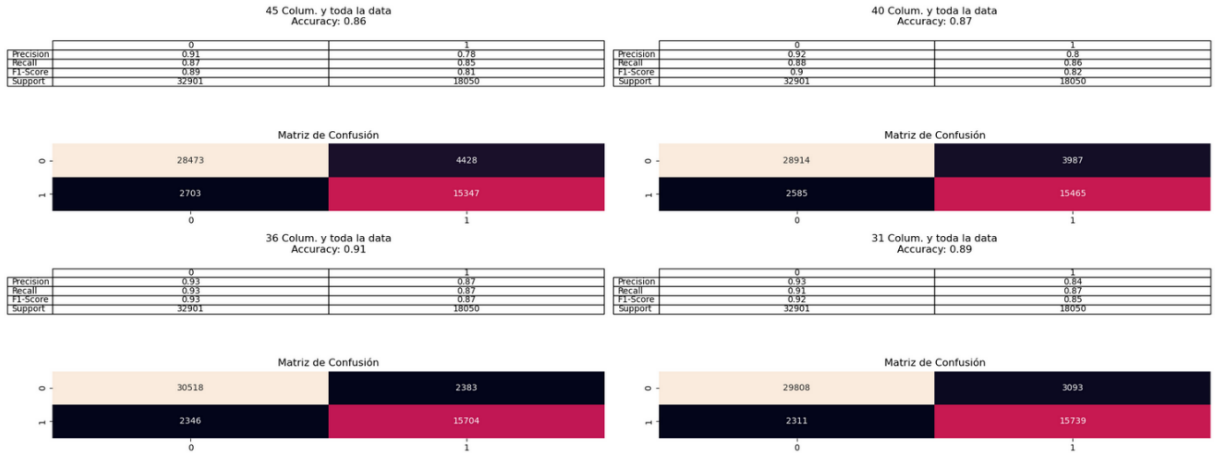


Figura 135: Resultados del SVM con data nueva parte 1

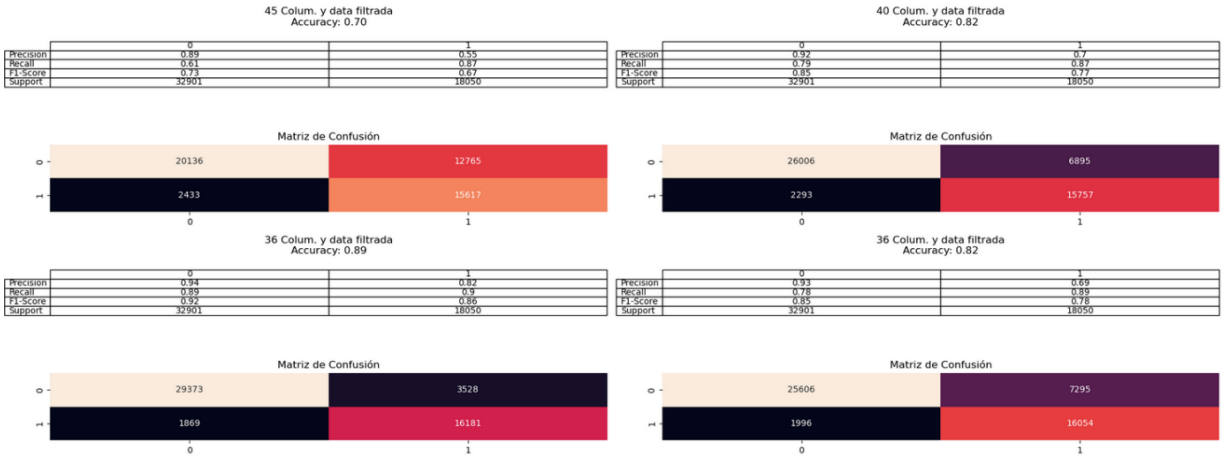


Figura 136: Resultados del SVM con data nueva parte 2

3.5.3. Isolation forest

En las figuras 137 y 138, se puede ver el accuracy y la matriz de confusión de la ejecución con cada set de datos, teniendo como resultados que accuracy para 45 columnas y toda la data es 0.68, 40 columnas y toda la data es 0.67, 36 columnas y toda la data es 0.68, 31 columnas y toda la data es 0.76, 45 columnas y data filtrada es 0.67, 40 columnas y data filtrada es 0.68, 36

columnas y data filtrada es 0.67, 31 columnas y data filtrada es 0.75, por lo que el mejor resultado corresponde a 31 columnas y toda la data con un 0.76.

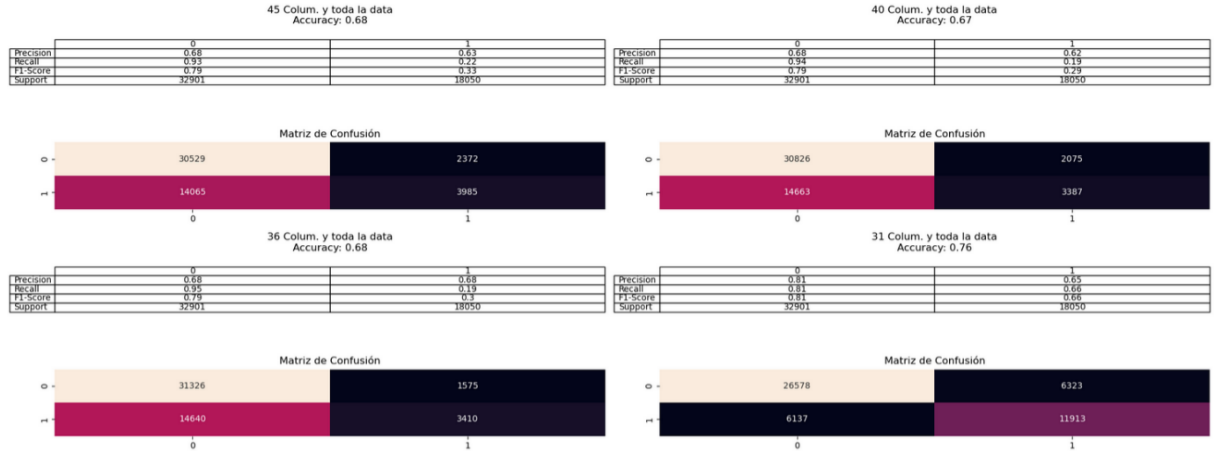


Figura 137: Resultados del Isolation forest con data nueva parte 1

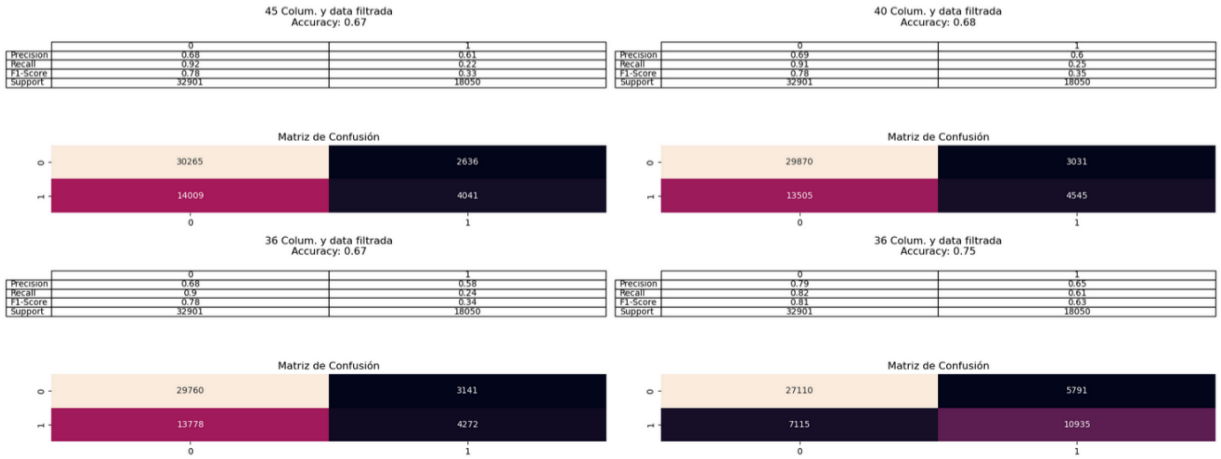


Figura 138: Resultados del Isolation forest con data nueva parte 1

3.5.4. Xgboost (predicción de churn)

En las figuras 139 y 140, se puede ver el accuracy y la matriz de confusión de la ejecucióncon el set de datos completo en el que se logró tener un accuracy del 99%.

```
--------------- Classification Report ---------------

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     41497
           1       1.00      1.00      1.00     22191

    accuracy                           1.00     63688
   macro avg       1.00      1.00      1.00     63688
weighted avg       1.00      1.00      1.00     63688

---------------------- XGBoost ----------------------
CPU times: user 7.96 s, sys: 23.9 s, total: 31.8 s
Wall time: 771 ms
```

Figura 139: Resultados del XGBoost con data nueva

```
cm = confusion_matrix(y_val, y_pred_val)
f = sns.heatmap(cm, annot=True, fmt='d')
```

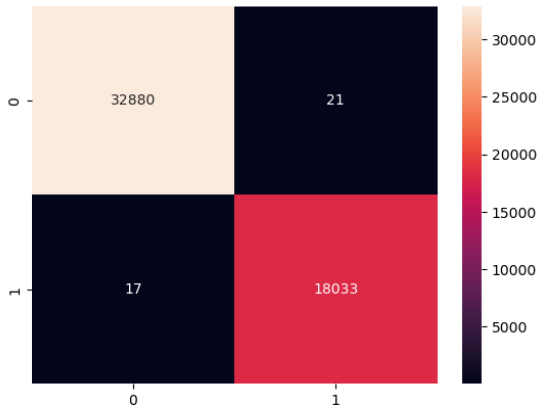


Figura 140: Matriz de confusión del XGBoost con data nueva

3.5.5. Xgboost (clasificación de motivos de churn)

Se utiliza nuevamente XGboost como algoritmo, en las figuras 141 y 142, se puede ver el accuracy y la matriz de confusión de la ejecución con el set de datos de los clientes que cancelaran su servicio y se los clasifica por el motivo, se obtienen un accuracy de 82%.

```
%%time
# Realizar la validación cruzada
cv_scores = cross_val_score(xgb_clf, X_val, y_val, cv=5, scoring='accuracy')
```

```
CPU times: user 50min 44s, sys: 12.4 s, total: 50min 57s
Wall time: 50min 57s
```

```
cv_scores
```

```
array([0.81037567, 0.81156828, 0.81658217, 0.80316135, 0.81181032])
```

Figura 141: Accuracy de XGBoost de clasificación con data nueva

```
cm = confusion_matrix(y_val, y_pred_val)
f = sns.heatmap(cm, annot=True, fmt='d')
```

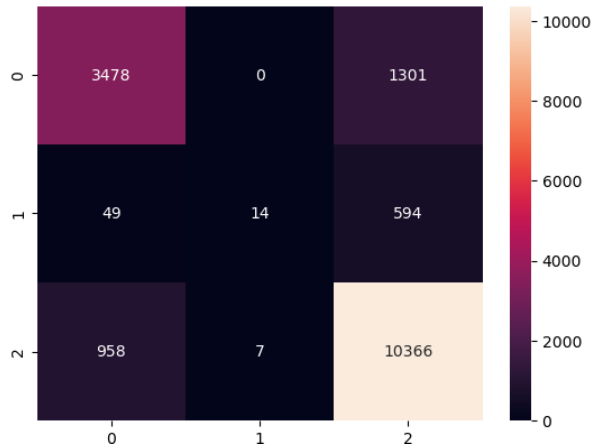


Figura 142: Matriz de confusión del XGBoost de clasificación con data nueva

A continuación, se muestra la tabla comparativa de los mejores resultados de los diferentes modelos, en los que destaca Xgboost (modelo seleccionado para la predicción) con un accuracy del 99% utilizando toda la data, Figura 143.

Modelo	Set de Datos	Accuracy
Xgboost	Todas las columnas y toda la data	0,99
SVM OneClass	36 columnas y toda la data	0,91
Isolation Forest	31 columnas y toda la data	0,76
Autoencoder	36 columnas y data filtrada	0,67

Figura 143: Resumen de los mejores resultados por cada modelo

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

4.1. Recolección de datos y estrategias para validación del proyecto

Como punto fundamental podemos indicar que la recolección de datos presento algo de problemas por cambios en permisos y alcance de las diferentes plataformas internas de la empresa, al final se contó con los accesos necesarios y las herramientas que permita la descarga de todos esos datos en los rangos de tiempo requeridos.

Se puede decir que la mayor restricción que se debió sortear es la política configurada en la herramienta de inteligencia de negocio que se utiliza al interno de la empresa, aquí se limita el tamaño del archivo y la cantidad de filas que posee, por tal motivo en lugar de tener un uno XLS con los tickets de los clientes, se debió descargar 60 archivos que en conjunto representa esta información, esto también nos aconteció con la información de los servicios cliente que debió ser dividida por periodo de alta para convertirse en seis archivos.

El propósito de esta restricción es evitar la sobrecarga del sistema y que un usuario al descargar demasiados datos bloque el acceso al aplicativo y afecte al resto de consumidores.

Para la validación del proyecto el primer punto es comprobar que se hayan cumplido con los objetivos planteados.

Actualmente se tiene una enorme cantidad de datos por ello para validar que los resultados del modelo sean los adecuados utilizaremos los datos históricos pudiendo ser prepandemia, pandemia o pos pandemia, según se considere necesario, pero esto nos proporciona únicamente la visión de que tan bueno es el modelo y si logra o no alcanzar el objetivo de tener una precisión del 90%, actualmente hemos superado el objetivo teniendo un modelo con un 99% de precisión.

El otro indicativo es el feedback del usuario con respecto a la información mostrada, si tiene lógica o les aporta dentro de la hoja de ruta de los mantenimientos planificados, al tener pantallas en el sistema (Figuras 97 a 131) que les muestre detalles como tickets, potencia,

clientes, etc., esto ha servido para que el encargado no lo tome como una caja negra y le devisión del porque se priorizan esos clientes/infraestructura

4.2. Puesta en marcha y funcionamiento

Como primer paso se levantó todo el ambiente en local de tal forma que el desarrollo, cambios, actualización y depuración se ejecuten fácil y rápidamente.

Una vez obtenido el modelo definitivo se procedió a trabajar con los ambientes NVIDIA, para el escenario actual se generó una instancia con 4 GPUs, 1Tb en Ram y atado a un NFS 5 TB para el almacenamiento. La preparación del ambiente consistió en la instalación y actualización de las librerías necesarias.

El cuadro de mando es una herramienta web que gracias al uso de Grafana herramienta muy utilizada a interno otorgó la interactividad necesaria para que el usuario pueda utilizarlo de forma sencilla como lo podemos observar en las Figuras 97 a 131.

Este cuadro de mando le da la interpretación gráfica de los datos que modelo analizó y generó la predicción de los clientes con alta posibilidad de churn y de ellos se segmento a aquello que su motivación son problemas técnicos para generar la sugerencia de la infraestructura en la que se debe ejecutar el mantenimiento.

El desarrollo de la herramienta se da debido a que cultura empresarial se orienta al uso de software libre por ello el uso de herramientas de inteligencia de negocio como Power BI o Tableau no es una opción, la única herramienta de este tipo que se maneja es la excepción a la regla y por ser heredada de un proyecto cuyo alcance la incluía.

El paso a producción se lo realizó dentro de la propia infraestructura de la empresa, ya que se posee los recursos necesarios para este propósito, a nivel de servidores físicos o máquinas virtuales.

Lo que si se debe remarcar y recordar es que por política de seguridad de la información únicamente puede ser accesible desde la red interna de la empresa o por medio del uso de una VPN autorizada.

4.3. Pruebas de funcionalidad

Lo primero que se debe indicar es que modelo desarrollado tienen una presión del 99% en la identificación de clientes que tiene alta probabilidad de abandono de la empresa y sobre el resultado del modelo se ha identificado la infraestructura de la que dependen y eso se muestra en el cuadro de mando tal como se lo planteo en los objetivos descritos en los apartados 1.4.1 y1.4.2.

Con respecto a la aceptación de la recomendación se ha logrado obtener feedback positivo lo cual nos indica que la herramienta si es de apoyo para el personal de operaciones encargado de la planificación de los mantenimientos en la infraestructura de telecomunicaciones.

4.4. Análisis costo/beneficio

Ahora bien, al tener la infraestructura levantada precisamente para este tipo de iniciativas, no se incurre en gastos adicionales por este rubro, simplemente se debe indicar por qué y paraqué del proyecto.

Analizando los beneficios partiremos de que se ha reducido la tasa de churn en un X%, lo que representa M clientes, y según lo indicado por (García, D.L., Nebot, À. & Vellido, A., 2017) que atraer un nuevo cliente costará entre 5 a 10 veces más que conservar a los actúales clientes (Z1) y teniendo en mente que sabemos el valor que pagan mensualmente estos clientes, podemos calcular el valor que dejamos de percibir (Z2).

A eso le podemos calcular lo que representa actualmente la planificación de los mantenimientos de forma manual Z3 versus la nueva forma propuesta Z4.

$$Z1 = Clientes \cdot tasa\ churn\% \cdot 5 \cdot valor\ cliente\ nuevo$$

$$Z2 = Clientes \cdot tasa\ churn\% \cdot valor\ del\ plan$$

$$Z3 = Planificadores \cdot 5\ horas \cdot suledo\ hora \cdot 4\ semanas$$

$$Z4 = Planificadores \cdot 1\ horas \cdot sueldo\ hora \cdot 4\ semanas$$

$$Total\ ahorro = Z1 + Z2 + (Z3 - Z4)$$

El total de ahorro que se tiene se calcula en base a:

- **Z1:** Costo de conseguir un nuevo cliente que reemplace al que cancelo el servicio
- **Z2:** Valor que se deja de recibir porque los clientes abandonaron la empresa
- **Z3:** Costo que representa la planificación en el proceso actual
- **Z4:** Costo que representa la planificación con la mejora

A Z3 le restamos Z4 porque no es que esta tarea desaparece por completo, pero si disminuye y ese valor es el que consideramos para el cálculo.

Tomando en cuenta estos números se puede observar que el proyecto realmente aporta valor a la empresa y genera benéficos evitando un costo y una perdida.

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Ahora bien, en esta sección se ha decidido mostrar los tres aspectos principales del problema planteado: componente técnico, las personas (recurso humano que ejecutaba el trabajo manual) y el negocio en sí.

Iniciando por el aspecto técnico, digamos que entre paréntesis lo menos complejo de los tres mencionados, aquí lo preponderante son los datos, por lo que la elección de características importantes que conforma cada set de datos tiene un impacto directo en los resultados, para nuestro caso no utilizar data filtrada otorgo los mejores resultados.

El contar con un ambiente que proporciona el procesamiento y GPUs necesarios para correr los modelos también ayudo enormemente, al permitir ejecutar pruebas pensadas que en otras condiciones hubiera tomado semanas o simplemente no se podrían realizar, como resultado de todo esto se tiene que para el codificador automático e isolation forest realmente fue desafiante el trabajar con los datos eso es evidente por los bajos resultados obtenidos, mientras que SVM y Xgboost logaron ser sobresalientes, aquí se debe mencionar que al interior de la empresa es común el uso de Xgboost por su robustez.

El tema del recurso humano por su parte realmente fue complejo, dada la carga de trabajo del

personal encargado de la planificación de los mantenimientos se pasó de cuatro a diez

personas, quienes fueron involucrados en el proyecto como expertos y asesores técnicos para garantizar el éxito de la iniciativa, pero, de entrada y dada la naturaleza humana son quienes se resistían al cambio, y veían como una amenaza a su puesto de trabajo o estatus quo la automatización propuesta, el discurso común se enfocó en: “eso ya lo hacemos”, “lo que hacemos funciona, no es necesario cambiarlo”, “acaso hay dudas sobre nuestro criterio para planificar el trabajo”, y temas de este estilo.

Aquí lo importante fue involucrar a las personas desde el primer día como parte de una nueva metodología de trabajo implementada en la empresa, no solo indicarles los beneficios que se tendrá en su trabajo sino el gran impacto en la organización y cuál será su aporte, se pasó del “te cuento lo que estamos haciendo” a probemos y mira estos son los resultados.

En la etapa de recolección de data fue complicado el conseguir el apoyo y guía de los expertos, pero a medida que se avanzó la gente se fue involucrando, el punto decisivo se centró en el análisis exploratorio aquí con la visualización de la data empezaron a verle utilidad (ya para su trabajo actual), plantear hipótesis y darse cuenta del camino que se plantea seguir, lo que si se debe mencionar es que la resistencia disminuyo notablemente pero aun no desaparece por completo, aunque los entusiasta de este grupo humano van ganando terreno, planteando nuevas posibilidades de análisis y mejoras al trabajo realizado, hoy la planificación toma menos de 30 minutos por cada encargado y les ha permitido mejora el seguimiento y apoyo a sus cuadrillas.

El siguiente componente es el negoció, aquí lo primero que se debe mencionar, es que la empresa decidió hacer una división en base al ingreso promedio por usuario (ARPU) y manejarlo en 2 marcas diferentes, aquellos que tienen un ARPU menor a X estarán bajo Xnet_1 y aquellos sobre este valor en Xnet, en este punto el trabajo realizado cubre y cumple para Xnet_1, pero no puede ser utilizando para Xnet.

Entre las principales diferencias que nos impacta es la proporción de clientes activos y cancelados, mientras que para Xnet_1 está en 65/35, para Xnet es 95/5, dando como resultado que los algoritmos no son capaces de reconocer los cancelados, por lo que para Xnet será necesario un nuevo análisis de opciones a implementar.

Un aspecto importante que apareció con el uso del modelo, es la mala clasificación del motivo de cancelación del cliente cuando se utiliza la versión indicada por el cliente cuando decide terminar su contrato con la empresa, mientras que, en llamadas o encuestas posteriores, la persona indica la causa real de su decisión cambiando la primera información en más del 50% de los casos, ahora sí, siendo consistente para ser interpretada por el modelo, este detalle fue valioso para marketing tener certeza de que data es la que debe utilizar.

El trabajo realizado es solo una parte de la estrategia de retención de clientes, pero si tiene un impacto positivo al tener esta información disponible no solo para técnico, el área comercial se ha servido del resultado del modelo y se inició con campañas de retención generación de nuevas opciones de servicio y llamadas proactivas a estos clientes.

Hoy dada la necesidad de ser muy precisos en detectar a potenciales cancelados, se decidió realizar una gran inversión al adquirir nueva infraestructura que permita tener más variables no solo para este escenario, sino para para predecir fallas y anticiparnos a soportes, para ello hoy se tiene levantado un sistema que recolecta todos estos logs, teniendo más detalle de comportamiento y uso, por ejemplo, nos indica por cada cliente que tanto de su canal de utiliza (saturación), en que frecuencia trabajan sus dispositivos inalámbricos, cuantos dispositivos tiene conectado, la existencia de interferencia o ruido en la señal, sobre el equipo terminal del cliente se puede saber si existe sobrecalentamiento, uso excesivo de CPU, etc.

Como se indicó al inicio no por tener más variables se tendrá mejores resultados, pero esto es el punto de partida, y sobre esta primera versión del modelo ya se plantearon nuevos retos e hipótesis con el uso de estas nuevas variables.

5.2. Recomendaciones

En Xnet partimos del análisis de churn con el porcentaje desbalanceado de clientes entre activos y cancelados, pero es necesario continuar el proceso a perfilación de clientes para ofreces nuevos productos, predicción de un potencial cliente con cancelación temprana de su servicio (menos de 3

meses) o clientes que no será buena paga, análisis de sentimientos en

llamadas para indicar al operario como responder y cuál debe ser el siguiente paso y unainfinidad de casos de uso que se identifiquen o surjan en la operación.

Hoy el mundo de la inteligencia artificial, es parte del giro de negocio de la empresa por lo que se ha impulsado el uso y aquí el siguiente paso recomendado es orquestar todas estas iniciativas dispersas a un objetivo común, bajo estándares de trabajo que aporte el mayor valor posible para la empresa, lo que se debe remarcar es el hecho de que se debe conversar con la gente, con la persona que ejecuta que serán los guías que permitirán tener buenos resultados en el corto plazo.

Finalmente es necesario levantar y disponibilizar el ambiente de producción para despliegue automático, evaluación y ajustes de modelos con acceso a las diferentes APIs, para lo cual se recomienda trabajar con los expertos de la marca para que la implementación sea sostenible y escalable en el tiempo, esto puede convertirse en una oportunidad de negocio con un producto de IA que sea incorporado al catálogo de la empresa Xnet.

6. Referencias bibliográficas

Aiofthings. (S.f.).Datos semi-estructurados.
https://aiofthings.telefonicatech.com/recursos/datapedia/datos-semi-estructurados

Amat Joaquín. (mayo 2020). Detección de anomalías: Isolation Forest.
https://cienciadedatos.net/documentos/66_deteccion_anomalias_isolationforest Amat
Joaquín. (abril 2021). Detección de anomalías con autoencoders y pythom.
https://cienciadedatos.net/documentos/py32-deteccion-anomalias-autoencoder-python

Amin, A., Al-Obeidat, F., Shah, B., Adnan, A., Loo, J., & Anwar, S. (2019). Customer churn prediction in telecommunication industry using data certainty. Journal of Business Research, 94,290-301.

APRENDEIA. (s.f.). Aprendizaje no supervisado. https://aprendeia.com/aprendizaje-no- supervisado-machine-learning/

APRENDEIA. (s.f.). Reducción de la dimensionalidad. https://aprendeia.com/reduccion-de-la-dimensionalidad-machine-learning/

Bagnato, J. (05 septiembre 2017). Aplicación de Machine Learning. https://www.aprendemachinelearning.com/aplicaciones-del-machine-learning/

Biografías y Vidas. (s.f.). Alan Turing. https://www.biografiasyvidas.com/biografia/t/turing.htm Blanco, D. (23 marzo 2019). Aprendizaje supervisado. https://www.diegocalvo.es/aprendizaje-supervisado/

Blanco, E. (17 octubre 2019). ¿Cómo funciona el algoritmo de Backpropagation en una red Neuronal? https://empresas.blogthinkbig.com/como-funciona-el-algoritmo-backpropagation-en-una-red-neuronal/

Canal AprendeIA con Lidgi Gonzalez. (6 de Abril 2018) HISTORIA DE MACINE LEARNING | # 2 Curso de introducción a Machine Learning [Archivo de Vídeo]. HISTORIA DE MACHINE LEARNING | #2 Curso de Introducción a Machine Learning

Gonzalez, V. (18 enero 2018). Una Breve Historia del Machine Learning. https://empresas.blogthinkbig.com/una-breve-historia-del-machine-learning/

Cardellino F, (20 marzo 2021). A guía definitiva del paquete NumPy para computación científica en Python. https://www.freecodecamp.org/espanol/news/la-guia-definitiva-del-paquete-numpy- para-computacion-cientifica-en-python/

Carmona P. (20 abril 2020). Data Visualization con pandas y seaborn. https://medium.com/ironhack/data-visualization-con-pandas-y-seaborn-1044906af34f

Chacón J. (22 marzo 2021). Introducción a Pandas, la librería de Python para trabajar con datos. https://profile.es/blog/pandas-python/

Cómo funciona el algoritmo Xgboost en Python. (s.f.).The machine Learners https://www.themachinelearners.com/xgboost-python/

Cómo funciona el algoritmo XGBoost. (s.f.). ArcGIS Pro https://pro.arcgis.com/es/pro-app/latest/tool-reference/geoai/how-xgboost-works.htm Conferencia de Darthmouth 1956. (s.f.). Arthur Samuel.

https://darthmouthconference.wordpress.com/participantes-en-la-conferencia/arthur-samuel/

Configuración del modelo SVM. (23 febreo 2023). 18.4.0
https://www.ibm.com/docs/es/spss-modeler/18.4.0?topic=nugget-svm-model-settings CICE.
(s.f.).Historia y evolución de la inteligencia artificial.
https://www.cice.es/blog/articulos/historia-evolucion-la-inteligencia-artificial/
Datapeaker. (s.f.). Guía para principiantes sobre clasificación de texto con PyCaret.
https://datapeaker.com/big-data/guia-para-principiantes-sobre-clasificacion-de-texto-con- pycaret/
DataScientest. (s.f.). Matplotlib: todo lo que tienes que saber sobre la librería Python de Dataviz.
https://datascientest.com/es/todo-sobre-matplotlib
Dynamox. (20 agosto 2021). Conozca sobre los 4 tipos de mantenimiento industrial.
https://dynamox.net/es/blog/cuatro-tipos-mantenimento-industrial
Edwine, N., Wang, W., Song, W., Ssebuggwawo, D. (18 julio 2022). Detecting the Risk of Churn in Telecom Sector: A Comparative Study. https://www.hindawi.com/journals/mpe/2022/8534739/ Edix. (26-07-2022). Framework. https://www.edix.com/es/instituto/framework/
ESIC. (octubre 2020). ¿Para qué sirve Python? Razones para utilizar este lenguaje de programación. https://www.esic.edu/rethink/tecnologia/para-que-sirve-python
Fernández Jáuregui A. (s.f.).Tutorial de Sklearn Python.
https://anderfernandez.com/blog/tutorial-sklearn-machine-learning-python/
Frank, R.(20 de Julio del 2018). Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957. Rastreator. https://empresas.blogthinkbig.com/historia-de-la-ia-frank-rosenblatt-y-e/
Funcionamiento de SVM. (17 agosto 2021). SAAS. https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-how-svm-works
García, D.L., Nebot, À. & Vellido, A. Intelligent data analysis approaches to churn as a business problem: a survey. Knowl Inf Syst 51, 719–774 (2017). https://doi.org/10.1007/s10115-016-0995-z
Gonzalez, L. (02 marzo 2021).¿Que es Deep Leerning?. https://aprendeia.com/que-es-deep-learning/

Hiperparámetros de XGboost. (s.f.). AWS
https://docs.aws.amazon.com/es_es/sagemaker/latest/dg/xgboost_hyperparameters.html Ibáñez, A.(16 abril 2019). Semi-Supervised Learning...el gran desconocido. https://empresas.blogthinkbig.com/semi-supervised-learningel-gran-desconocido/
Innovaciondigital360. (03 julio 2023). Qué son los Autoencoders y cómo funcionan. https://www.innovaciondigital360.com/i-a/que-son-los-autoencoders-y-como-funcionan/
IONOS. (08 octubre 2020). Keras: biblioteca de código abierto para crear redes neuronales. https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que- es-keras/
Isolation Forest en Python. (s.f.). En anderfernandez. https://anderfernandez.com/blog/isolation-forest-en-python/
Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. ElectronicMarkets, 31(3), 685-695. https://link.springer.com/article/10.1007/s12525-021-00475-2
John. (07 julio 2021). Definición y tipos del splitter fibra óptica. https://community.fs.com/es/blog/what-is-a-fiber-optic-splitter-2.html
Kotler, P., & Armstrong, G. (1994). Marketing management, analysis, planning, implementation,and control, Philip Kotler. London: Prentice-Hall International.
López-Avila, Leyanis, Acosta-Mendoza, Niusvel, & gago-Alonso, Andrés. (2019). Detección deanomalías basada en aprendizaje profundo: Revisión. Revista Cubana de Ciencias Informáticas,13(3), 107-123. Recuperado en 23 de septiembre de 2022, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227- 18992019000300107&lng=es&tlng=es.
López, A. (06 junio 2022). Qué es y cómo funciona la tecnología GPON: secretos técnicos. https://www.redeszone.net/tutoriales/redes-cable/tecnologia-ftth-gpon-que-es-funcionamiento/
López, R. (s.f.). El futuro de la IA: hacia inteligencias artificiales realmente inteligentes. https://www.bbvaopenmind.com/articulos/el-futuro-de-la-ia-hacia-inteligencias-artificiales-

realmente-inteligentes/

Mailjet. (s.f.). ¿Qué es el Big Data y cómo funciona?.

https://www.mailjet.com/es/blog/marketing/big-data/

MARKETING DIGITAL. (s.f.). Chatbot: ¿Qué es, para qué sirve y cómo funciona?Martínez

Cristina. (junio 2018). Máquina de vectores de soporte.

https://rpubs.com/Cristina_Gil/SVM#:~:text=Los%20kernels%20son%20funciones%20que,en%

20un%20nuevo%20espacio%20dimensional

MINTEL. (2020) Plan nacional del Gobierno Electrónico. Ministerio de telecomunicaciones y dela

Sociedad de la Información.

https://www.telecomunicaciones.gob.ec/wp-content/uploads/2020/06/Plan-Nacional-de-

Gobierno-Electr%C3%B3nico-2018-2021-firmado-1.pdf

Millán, P. (2007).Qué es … GPON (Gigabit Passive Optical Network).

https://www.ramonmillan.com/tutoriales/gpon.php#introduccion

Mirjalili, V., & Raschka, S. (2020). Python machine learning. Marcombo

Morales, E. (s.f.). Aprendizaje Semisupervisado.

https://ccc.inaoep.mx/~emorales/Cursos/Aprendizaje2/Acetatos/semisupervisado.pdf

Nhu, N.Y., Van Ly, T., Truong Son, D.V. (24 de mayo 2022).Churn prediction in telecommunication industry using kernel Support Vector

Machines.https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0267935

NVIDIA. DEVELOPER (s.f.). NVIDIA CLARA. https://developer.nvidia.com/claraOptuna:

A hyperparameter optimization framework. (2023). Optuna

https://optuna.readthedocs.io/en/stable/

ORACLE. (s.f.). ¿Qué es big data?. https://www.oracle.com/es/big-data/what-is-big-data/ ORACLE

México. (s.f.).¿Que es la inteligencia artificial? Obtenga más información sobreinteligencia artificial.

https://www.oracle.com/mx/artificial-intelligence/what-is-ai/

Peiró, R. (s.f.).Churn. https://economipedia.com/definiciones/churn.html

Plotly Open Source Graphing Library for Python. (2023). Plotly

https://plotly.com/python/

Powerdata. (s.f.).Big Data: ¿En qué consiste? Su importancia, desafíos y gobernabilidad. https://www.powerdata.es/big-data

ProkectPro. (12 septiembre 2022). How to do Anomaly Detection using Machine Learning in Python. https://www.projectpro.io/article/anomaly-detection-using-machine-learning-in-python- with-example/555

Puentes Digitales. (14 febrero 2018). Todo lo que necesitas saber sobre TensorFlow, la plataforma para Inteligencia Artificial de Google. https://puentesdigitales.com/2018/02/14/todo- lo-que-necesitas-saber-sobre-tensorflow-la-plataforma-para-inteligencia-artificial-de-google/

¿Qué es Ethernet (IEEE 802.3)?. (08 diciembre 2022). https://www.ionos.es/digitalguide/servidores/know-how/ethernet-ieee-8023/#:~:text=Ethernet%20designa%20a%20una%20tecnolog%C3%ADa,se%20crea%20mediante%20conexiones%20Ethernet.

¿Qué es un terabyte?.(13 agosto 2021). https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-un-terabyte/ Rodríguez Yeifer. (31 octubre 20218).XGBoost https://www.diegocalvo.es/xgboost/

SAS. (s.f.). Inteligencia Artificial Qué es IA y Por Qué Importa. https://www.sas.com/es_cl/insights/analytics/what-is-artificial-intelligence.html

DataScientest. (19 abril 2022). Deep Learning o Aprendizaje profundo: ¿qué es?. https://datascientest.com/es/deep-learning-definicion

Significado. (s.f). Significado de autoencoder. Autoencoder https://significado.com/autoencoder/

Singh Chauhan N. (02 septiembre 2020). Métricas De Evaluación De Modelos En El Aprendizaje Automático. https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de- modelos-en-el-aprendizaje-automatico

Support vector machines (SVM). (s.f.). Unipython https://unipython.com/support-vector-machines-svm/#google_vignette

TIBCO. (s.f.). ¿Qué es la detección de anomalías?. https://www.tibco.com/es/reference-center/what-is-anomaly-detection

TIBCO. (s.f.). ¿Qué es la detección de valores anómalos?. https://www.tibco.com/es/reference-center/what-is-outlier-detection

TIBCO. (s.f.). ¿Qué es el aprendizaje supervisado? https://www.tibco.com/es/reference-center/what-is-supervised-learning

Todo de Redes. s.f. Capa 3: NIVEL DE RED. https://tododeredes.com/modelo-osi/capa-3/

Ventajas y desventajas de SVM. (s.f.). InteractuveChaos https://interactivechaos.com/es/manual/tutorial-de-machine-learning/ventajas-y-desventajas-de- svm

Worton. (14 febrero 2022). ¿Cuál es la diferencia entre Switch de Capa 2 y el Switch de Capa 3?. https://community.fs.com/es/blog/layer-2-switch-vs-layer-3-switch-what-is-the-difference.html

Westreicher, G. (s.f.). Mantenimiento. https://economipedia.com/definiciones/mantenimiento.html

XGBoost Parameters. (s.f.).dmlc XGBoost https://xgboost.readthedocs.io/en/stable/parameter.html

Xu, T., Ma, Y., Kim, K. (21 mayo 2021). Telecom Churn Prediction System basado en Ensemble Learning usando Feature Grouping. Ciencias Aplicadas.https://www.mdpi.com/2076- 3417/11/11/4742/htm

7. Glosario

ERP: Sistema de planificación de recursos empresariales.

Churn: Tasa de cancelación de clientes.

Framework: Estructura previamente definida, para ejecutar su trabajo y lograr los objetivos planteados (Edix, 2022).

Nvidia Clara: Framework diseñado para utilizar la potencialidad de los equipos Nvidia optimizado para ciencias de la vida y aplicaciones médicas (NVIDIA. DEVELOPER, s.f.).

Chatbot: tecnología que permite interactuar con un asistente virtual por medio de lenguaje natural, generalmente texto y que está disponible 24 horas al día los 7 días de la semana (MARKETING DIGITAL, s.f.).

Ethernet: Tecnología utilizada para comunicación entre diferentes terminales para enviar yrecibir información (¿Qué es Ethernet (IEEE 802.3)?, 2023)

Equipo L3: Son aquellos que se utilizan para interconectar y rutear entre host que pueden estaren redes diferentes (Todo de Redes, s.f.)

Equipo L2: Se refiere a equipos que proporcionan conectividad a dispositivos en una mismared (Worton, 2022)

Splitters: Dispositivo encargado de tomar un haz de luz entrante y dividirlo en dos o más haces salientes según sea necesario (John, 2021).

Terabytes: unidad de almacenamiento que representa 1 099 511 627 776 bytes

Hiperplano: subespacio que no tiene que pasar por el origen para separar las clases (MartínezC, 2018).

8. ANEXOS