Electrodunas - Sistema De Detección De Anomalías
Desarrollo y Prueba de los Modelos
Grupo 3

Introducción

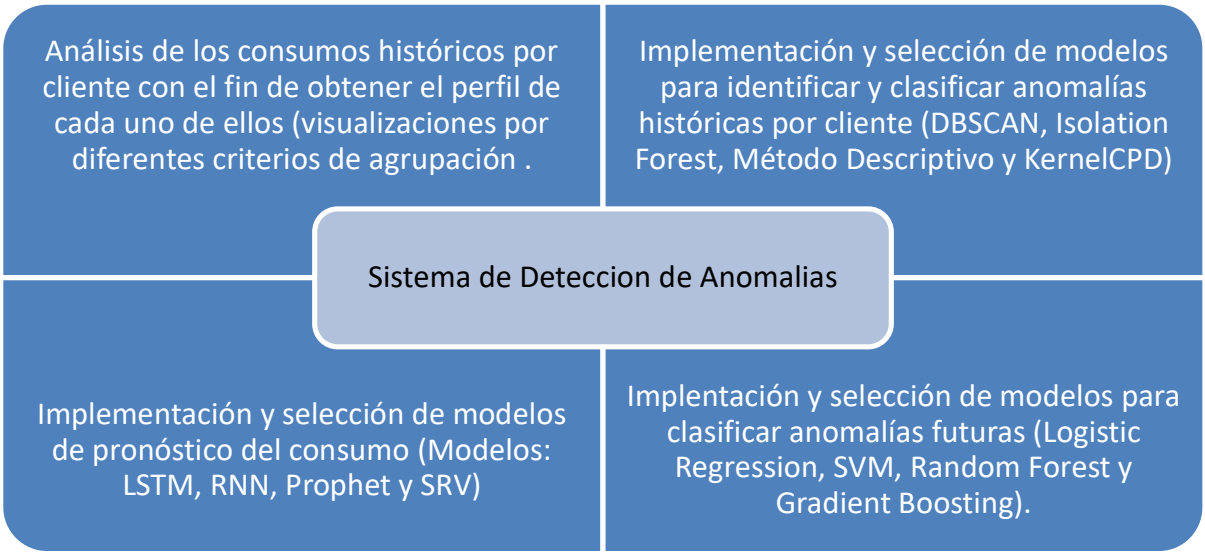
Electro Dunas, empresa peruana especializada en distribución de energía eléctrica, opera en las provincias de Ica, Huancavelica y Ayacucho, abarcando 5,402 km2 y atendiendo a 264,480. Desde su integración al Grupo Energía Bogotá en agosto de 2019, pertenece a un conglomerado líder en energía y gas natural en Colombia, Perú y Brasil. El GEB, comprometido con generar valor, bienestar comunitario y sostenibilidad ambiental, refleja la visión y enfoque de Electro Dunas.

La empresa se desenvuelve en dos segmentos de mercado: el regulado y el de competencia. Este último se compone de clientes libres, tanto propios como terceros. A los clientes libres propios se les facturan los precios de generación según sus acuerdos contractuales, además de los cargos regulados por transmisión. Por otro lado, los clientes terceros son facturados de acuerdo con los cargos regulados por transmisión y/o distribución, en función de la utilización que hagan del sistema eléctrico de Electro Dunas.

Con un crecimiento significativo de clientes no regulados, la empresa se propone utilizar analítica de datos para identificar posibles anomalías en el comportamiento de sus clientes no regulados. El proyecto se enfoca en desarrollar un Producto Mínimo Viable (PMV) que visualice datos históricos, resuma comportamientos, identifique anomalías y proporcione alertas, con el objetivo de ser adoptado como una herramienta eficaz en los flujos operativos de Electro Dunas.

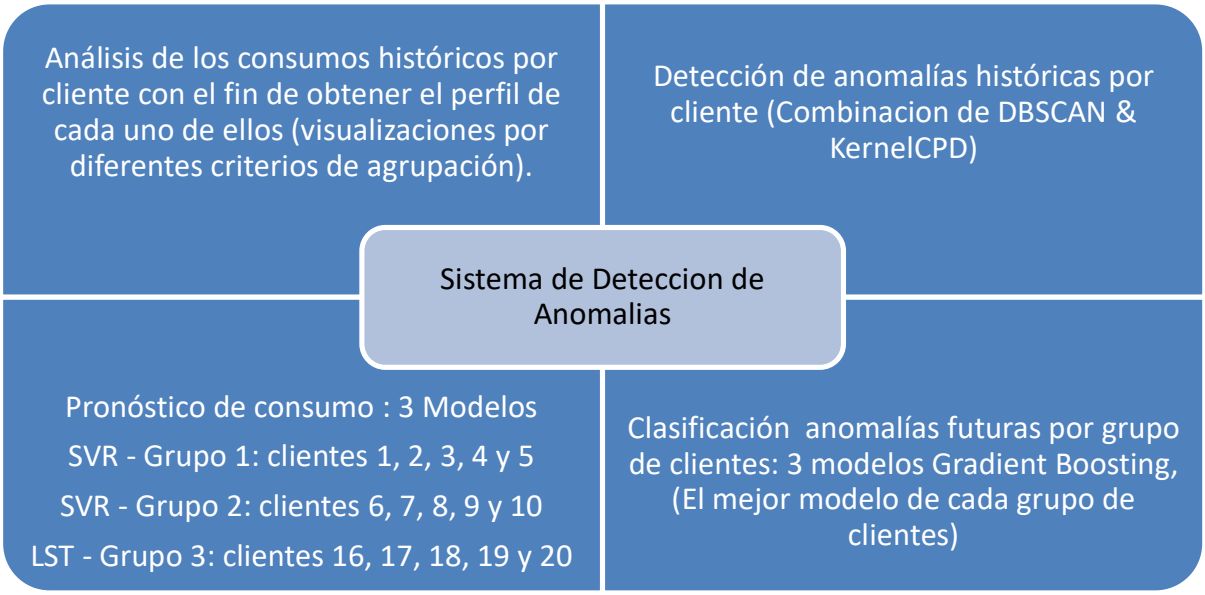
El conjunto de requerimientos propuestos en el módulo 1 (semana 1 & 2) han sido diseñados para ofrecer una variedad de herramientas analíticas que permiten comprender y gestionar el consumo de energía de los clientes no regulados de Electrodunas. Estos requerimientos abarcan desde la visualización de datos básicos, como la cantidad de clientes activos y la energía entregada por año, hasta la identificación de patrones de consumo por cliente y la detección de anomalías. Además, incluyen a proyección de consumos futuros y la detección de anomalías futuras.

El siguiente gráfico presenta las alternativas **evaluadas** para satisfacer los requerimientos propuestos.



Este documento incluye una sección para cada una de estas cuatro áreas, donde se plantean las alternativas de modelos o técnicas para satisfacer los requerimientos asociados con cada una de ellas. Se incluyen las ventajas y desventajas de cada alternativa, así como el tratamiento de los datos, las parametrizaciones, las métricas y, finalmente, un análisis del proceso de selección de las alternativas que mejor se ajustan para satisfacer los requerimientos de cada área.

El siguiente gráfico presenta las alternativas **seleccionadas** para satisfacer los requerimientos propuestos en cada área.



Procesamiento de Datos

La información de energía activa, energía reactiva, voltaje en la fase A y voltaje en la fase B proporcionada por el medidor de cada cliente se compila en archivos con formato CSV con un mapeo de campos estandarizado en los archivos. En total se cuentan con la información de 30 clientes y su clasificación en sectores económicos.

#	Columna	Descripción	Tipo	# Valores No Nulos
0	Fecha	Fecha y hora del consumo	Datetime64	463425
1	Active_energy	Cantidad de energía eléctrica entregada que realiza trabajo efectivo medida en kilovatios-hora (kWh)	float64	463425
2	Reactive_energy	Cantidad de energía intercambiada entre la fuente de energía y una carga sin realizar trabajo útil medida en kilovatios-ampere-reactivos-hora (kVarh)	float64	463425
3	Voltaje_FA	Voltaje en la fase A del sistema trifásico medido en voltios	float64	463425
4	Voltaje_FC	Voltaje en la fase C del sistema trifásico, medido en voltios	float64	463425

La información relativa a las fechas abarca un periodo de tres años, desde el 1 de enero de 2021 hasta el 1 de abril de 2023. En este intervalo, cada fecha se presenta el consumo por hora.

En la exploración de las series de tiempo para cada cliente, se observa que no todos comprenden el mismo periodo de tiempo. Sin embargo, al revisar cada serie de manera individual, se observa que no existen valores perdidos en ninguna de ellas. Dado que no se cuenta con información desde el punto de vista de los stakeholders sobre las razones por las cuales 15 clientes presentan niveles de completitud del 71%, 57% y 45% en la ventana temporal mencionada, se ha decidido enfocarse en los clientes que presentan una completitud superior al 99%. Esta precaución en el enfoque busca garantizar la integridad y la representatividad de los datos, evitando posibles distorsiones en el análisis y resultados posteriores.

A continuación, se listan los clientes que presentan una completitud mayor al 99% y que serán incluidos en el MPV:

```
['Cliente 01', 'Cliente 02', 'Cliente 03', 'Cliente 04', 'Cliente 05',
 'Cliente 06', 'Cliente 07', 'Cliente 08', 'Cliente 09', 'Cliente 10',
 'Cliente 16', 'Cliente 17',  'Cliente 18', 'Cliente 19', 'Cliente 20']
```

En el análisis de correlación entre variables, se destacó una correlación significativa de más del 97% entre Active_energy y Reactive_energy. No obstante, se optó por no descartar ninguna variable, ya que la naturaleza única de cada una contribuye de manera independiente al problema de negocio. Como resultado, no se efectuó una reducción de dimensionalidad en los datos.

Adicionalmente, se ha incorporado la marca del Sector Económico a los datos con el propósito de realizar un análisis por cliente. Esta inclusión busca identificar patrones de consumo según el sector económico y explorar posibles relaciones directas con el consumo de energía activa.

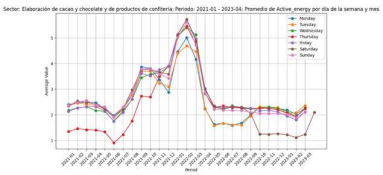
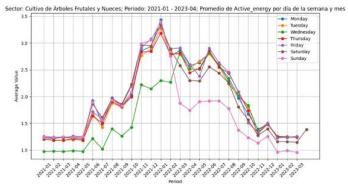
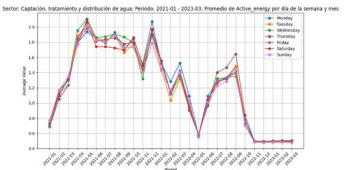
Los clientes muestran completitud en sus datos iniciales en cuanto a energía activa, reactiva y voltajes. Sin embargo, al analizar las series temporales por cliente, se observa disparidad en los registros. Algunos llegan hasta el 24 de marzo de 2023, mientras que otros se extienden hasta el 1 de abril de 2023. En lugar de interpolar para igualar las fechas máximas, lo cual podría introducir ruido al modelo de pronóstico, hemos decidido limitar las series más extensas (hasta el 1 de abril) hasta el 24 de marzo. Esto asegura que todos los clientes tengan datos desde el 1 de enero de 2021 hasta el 24 de marzo de 2023, evitando así inconsistencias en el rango temporal.

Perfilamiento de Clientes

Se comenzó por analizar el comportamiento de los sectores de los clientes seleccionados. En esta tabla se presenta el consumo promedio de energía activa por sector por día de la semana y mes.

En el eje horizontal del gráfico se encuentran los meses de cada año. En el eje vertical se muestra el valor promedio de consumo de energía activa. Este valor promedio se calcula a partir de los datos disponibles para cada día de la semana dentro de ese período. El gráfico consiste en varias líneas, una para cada día de la semana (lunes a domingo). Cada línea muestra cómo varía el valor promedio de la variable a lo largo del tiempo para ese día de la semana en particular.

Al observar este gráfico, se pueden identificar patrones de comportamiento o tendencias en el valor promedio de la variable a lo largo de los días de la semana y los meses para el sector económico específico.

A continuación, mencionamos algunas observaciones relacionadas con las gráficas anteriores:

Para el sector de elaboración de cacao y chocolate y de productos de confitería, se observan consumos superiores a 4 entre los meses de enero y abril de 2022 para cada día de la semana. Por otro lado, durante el periodo comprendido entre septiembre de 2022 y marzo de 2023, se observa un consumo alrededor de 2.1 para todos los días de la semana, excepto los sábados, donde el consumo está alrededor de 1.2.

En cuanto al sector de cultivo de árboles frutales y nueces, se observan consumos superiores a 3 entre los meses de noviembre de 2021 y febrero de 2022 para cada día de la semana, excepto los miércoles. Por otro lado, durante el periodo comprendido entre enero y marzo de 2023, se observa un consumo alrededor de 2.1 para todos los días hábiles de la semana.

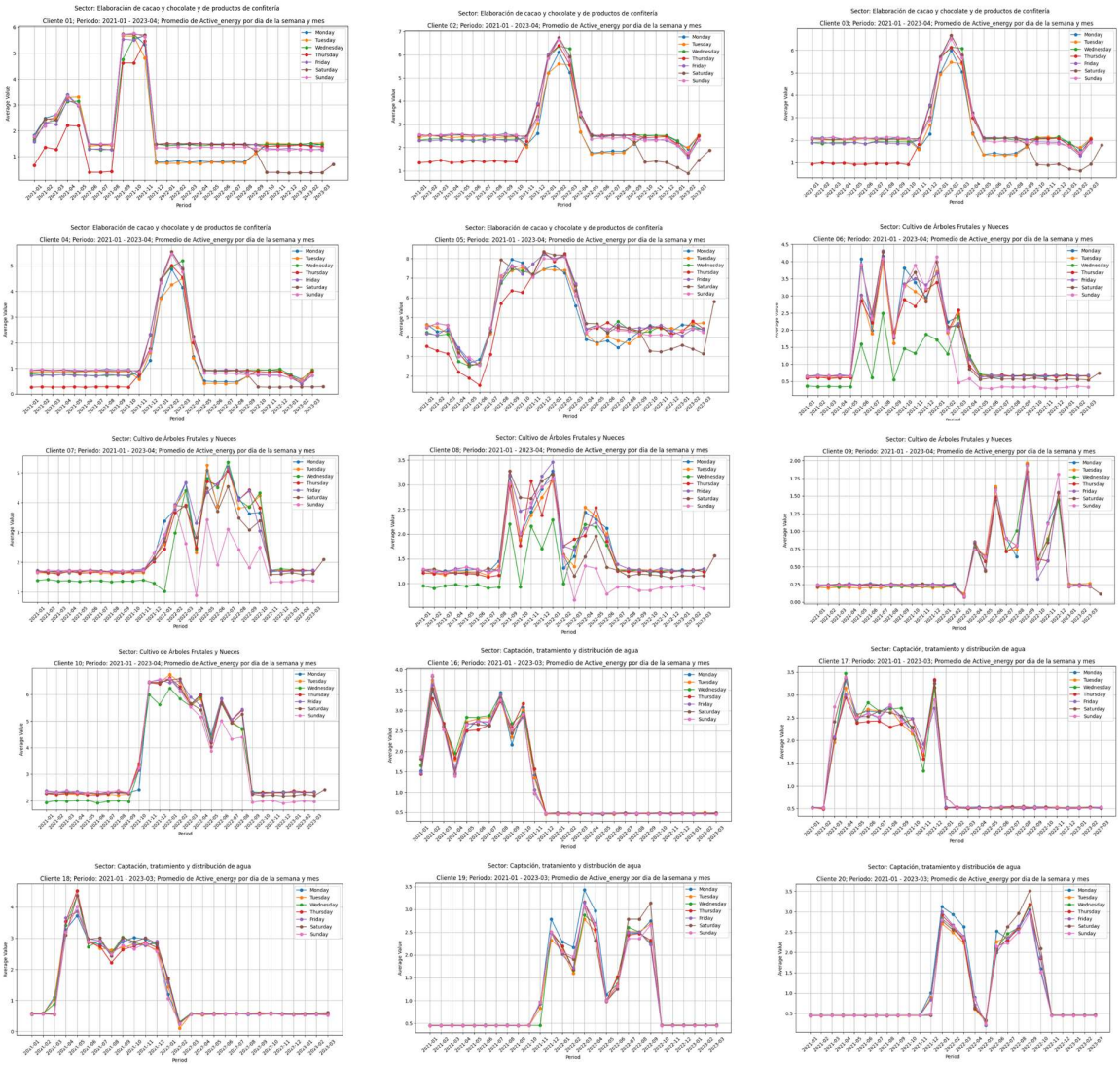
En el sector de captación, tratamiento y distribución de agua, se observan comportamientos superiores a 2 los domingos, martes, miércoles y viernes. Por otro lado, durante el periodo comprendido entre diciembre de 2022 y marzo de 2023, se observa un consumo entre 0.48 y 0.5 para todos los días de la semana.

En esta sección se presentan dos ejemplos que permiten analizar los comportamientos de consumo agrupados por diferentes criterios.

Consumo promedio de energía activa por cliente por día de la semana y mes

Esta tabla muestra el consumo promedio de energía activa de los clientes por día de la semana y mes. Los clientes están agrupados por el sector al que pertenecen y luego se presentan ordenados por su identificador. En la categoría de Elaboración de cacao, chocolate y productos de confitería, encontramos a los Clientes 1, 2, 3, 4 y 5. En la categoría de Cultivo de Árboles Frutales y Nueces, se incluyen los Clientes 6, 7, 8, 9 y 10. Por último, en la categoría de Captación, tratamiento y distribución de agua, están identificados los Clientes 16, 17, 18, 19 y 20.

A modo de ejemplo, podemos observar que los Clientes 2, 3 y 4 del sector de elaboración de cacao, chocolate y productos de confitería presentan un comportamiento similar, mientras que el comportamiento del Cliente 5 difiere sustancialmente del de los otros clientes en el sector.



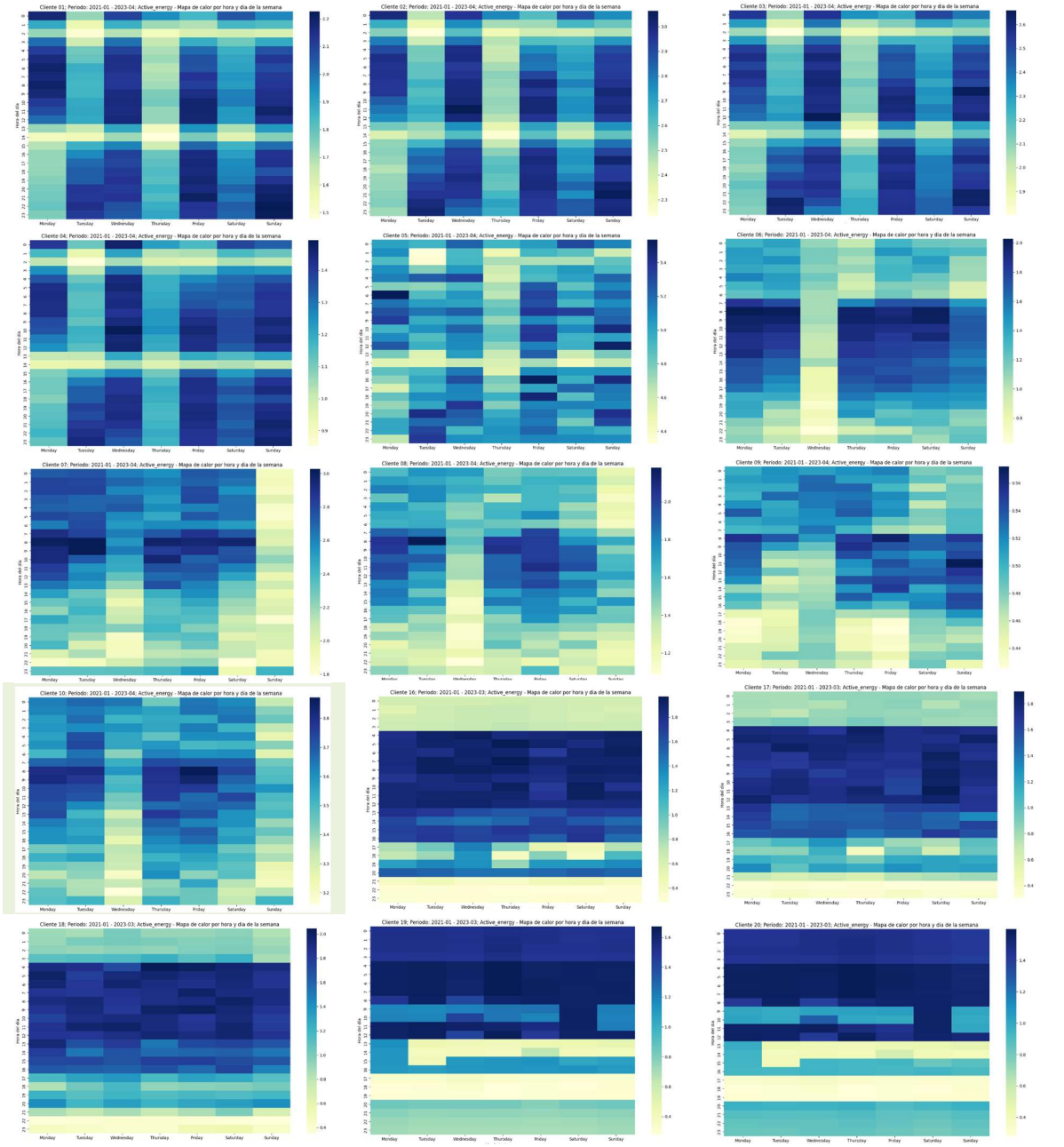
Mapa de calor por hora y día de la semana

Cada mapa de calor representa la distribución del consumo promedio de energía activa a lo largo del día y de la semana para un cliente específico.

En el eje horizontal del gráfico, se muestran los días de la semana, organizados de lunes a domingo. En el eje vertical, se representan las horas del día, desde la medianoche hasta las 23:00 horas.

Cada celda del mapa de calor está coloreada de acuerdo con el nivel de consumo promedio de energía activa en esa hora específica y día de la semana. Los colores más claros indican un menor consumo de energía, mientras que los colores más oscuros indican un mayor consumo.

Al observar este gráfico, se pueden identificar fácilmente los patrones de consumo de energía a lo largo de la semana y durante diferentes horas del día para el cliente seleccionado.



Detección de Anomalías

Para abordar la detección de anomalías se implementaron cuatro enfoques: Isolation Forest, DBSCAN, un método descriptivo con umbrales definidos mediante la desviación estándar y KernelCPD una versión optimizada de la técnica Pruned Exact Linear Time (PELT) a la cual nos referiremos también como análisis de series de tiempo. A continuación de describe detalladamente el proceso desarrollado con cada uno de estos enfoques.

Es importante resaltar que se empleó la técnica de muestreo por Bootstrap para generar muestras, extrayendo el 10% del tamaño total de la información disponible. Posteriormente, se generan ocho muestras, cada una seleccionando observaciones aleatorias del conjunto de datos original con reemplazo.

Isolation Forest

Este modelo empieza por construir árboles de decisión aleatorios utilizando un subconjunto aleatorio de características y datos. En cada nodo del árbol, se elige aleatoriamente una característica y un umbral para dividir los datos. Estos puntos de datos se separan en dos subconjuntos basados en si están por encima o por debajo del umbral seleccionado. Este proceso se repite recursivamente hasta que las anomalías se aíslen. Una vez que se construyen los árboles de aislamiento, se evalúa la anomalía de un punto de datos calculando la longitud promedio de la ruta desde la raíz hasta ese punto a lo largo de todos los árboles. Los puntos de datos que tienen rutas más cortas se consideran anomalías.

La calificación de una observación es una medida relativa en comparación con las demás observaciones. Se identifican como posibles anomalías aquellas observaciones cuya distancia predicha se sitúa por debajo de un cuantil específico. Por ejemplo, si se define un 5% de las observaciones como anomalías, el límite de decisión se establece en el cuantil 0.05 de todas las distancias calculadas.

Al entrenar el modelo, es necesario especificar el porcentaje de anomalías esperadas en los datos de entrenamiento (contamination). Este valor permite al modelo aprender el umbral a partir del cual una observación se considera una anomalía. Por ejemplo, si se estima que hay un 5% de anomalías, se utiliza el cuantil 0.05 de todas las distancias calculadas como límite de decisión

Ventajas

Eficiente con grandes volúmenes de información ya que construye árboles de decisión aleatorios y utiliza submuestras de características y datos. No asume ninguna distribución particular de los datos y puede funcionar bien incluso en conjuntos de datos con muchas características

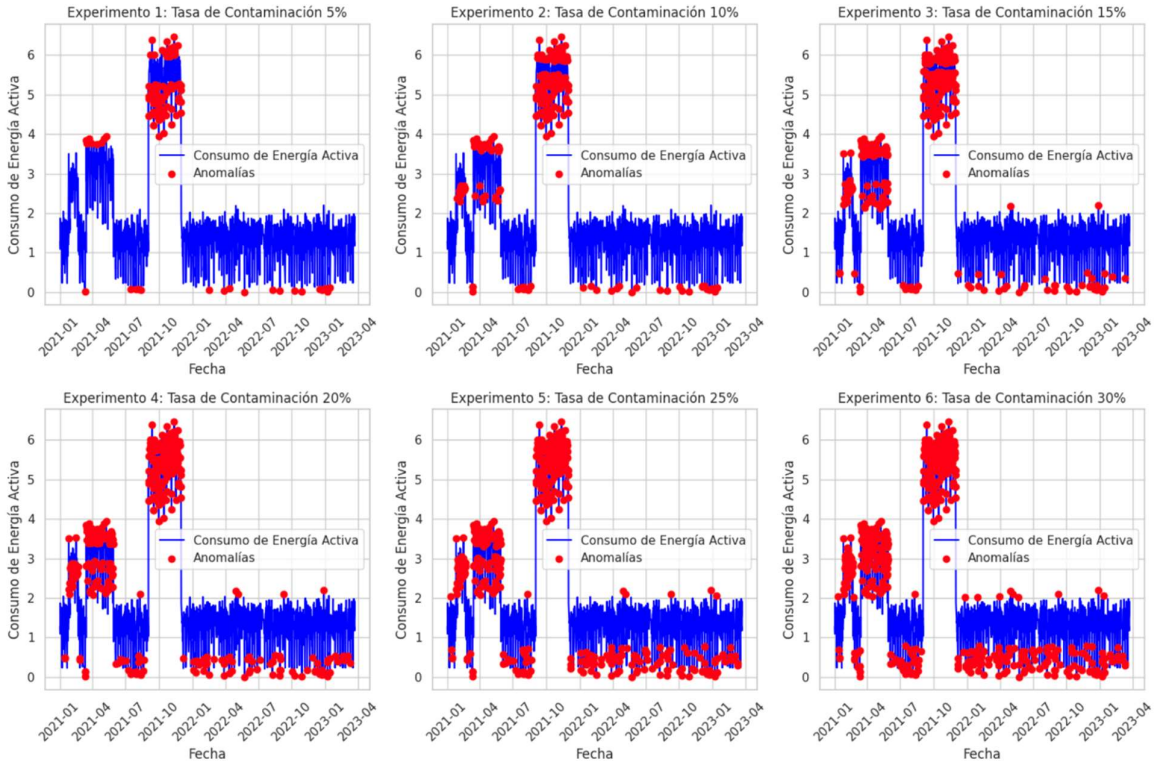
Desventajas

Puede mostrar un sesgo hacia valores extremos es decir que puede tener dificultades para detectar anomalías más sutiles que no se encuentran en regiones extremas del espacio de características.
Adicionalmente, no es tan efectivo en datos muy densos donde la mayoría de los puntos de datos son normales y solo unas pocas instancias son anómalas.

Parámetros y experimentos

Los experimentos se llevaron a cabo individualmente para cada cliente, variando el hiperparámetro de tasa de contaminación con el propósito de determinar el más adecuado para cada caso. Se utilizó una métrica conocida como "precisión subjetiva", que oscila entre 0 y 1, para evaluar la eficacia en la detección de anomalías. Este enfoque involucró un análisis visual de las gráficas de cada experimento, así como una revisión minuciosa de los resultados en las muestras de datos, con el fin de que todo el equipo pudiera decidir si la detección de anomalías fue precisa o no.
A continuación, se presentan los resultados de los experimentos llevados a cabo para el Cliente 1, acompañados de las respectivas gráficas de análisis:

Experimento	Tasa de Contaminación	Anomalías Detectadas	% Anomalías	Precisión subjetiva
1	5%	101	5,03%	0,4
2	10%	201	10,01%	0,6
3	15%	302	15,04%	0,7
4	20%	402	20,02%	0,8
5	25%	501	24,95%	0,75
6	30%	601	39,5%	0,5



Aquí se observa que para el Cliente 1, el parámetro de contaminación que arrojó los mejores resultados fue del 20%, ya que en este punto la detección de anomalías no es tan agresiva en comparación con la muestra. También se encontró que un valor demasiado bajo resulta en una detección poco rigurosa de anomalías, mientras que un valor demasiado alto conduce a una detección excesivamente agresiva.
Además, se nota que el modelo detecta anomalías en los extremos de cada tramo de tiempo, sin llegar a establecer un umbral fijo, sino más bien en función de cómo se presenta el patrón de comportamiento de la energía.

A continuación, se presentan los resultados para los otros clientes. Los valores de tasa de contaminación fueron los siguientes:

Tasa de contaminación	Cliente
15%	6, 7, 10, 18
20%	1,2,3,4, 9, 17, 19, 20
25%	5, 8, 16

DBSCAN

(Density-Based Spatial Clustering of Applications with Noise) es un algoritmo de clustering que identifica regiones densas de puntos en el espacio de características, separadas por regiones de baja densidad, lo que lo hace especialmente adecuado para detectar anomalías en conjuntos de datos con estructuras complejas. El algoritmo comienza identificando puntos centrales en el conjunto de datos, que son aquellos que tienen al menos un número mínimo de vecinos dentro de un radio especificado. Luego, expande los grupos de puntos centrales conectados a través de sus vecinos, formando así los grupos densos. Los puntos que quedan sin asignar a ningún grupo se consideran ruido o anomalías.

Para utilizar DBSCAN en la detección de anomalías en Electrodunas, primero normalizamos los datos y luego determinamos el valor óptimo de epsilon, que es el radio máximo de vecindad alrededor de un punto central. Utilizamos el método del codo para encontrar este valor óptimo, que representa el punto donde hay un cambio significativo en la densidad de los puntos. Una vez que se determina epsilon, se ajusta el modelo DBSCAN y se asigna una etiqueta a cada punto, siendo -1 aquellos puntos considerados anomalías.

Ventajas

Es robusto frente a conjuntos de datos con estructuras complejas y presencia de ruido. Esto se debe a que DBSCAN no asume ninguna distribución particular de los datos, lo que le permite identificar regiones densas y separarlas de las áreas de baja densidad de manera efectiva. Además, su capacidad para trabajar con datos grandes lo hace eficiente en tiempo y memoria, lo que es beneficioso en aplicaciones prácticas donde se manejan grandes volúmenes de información.

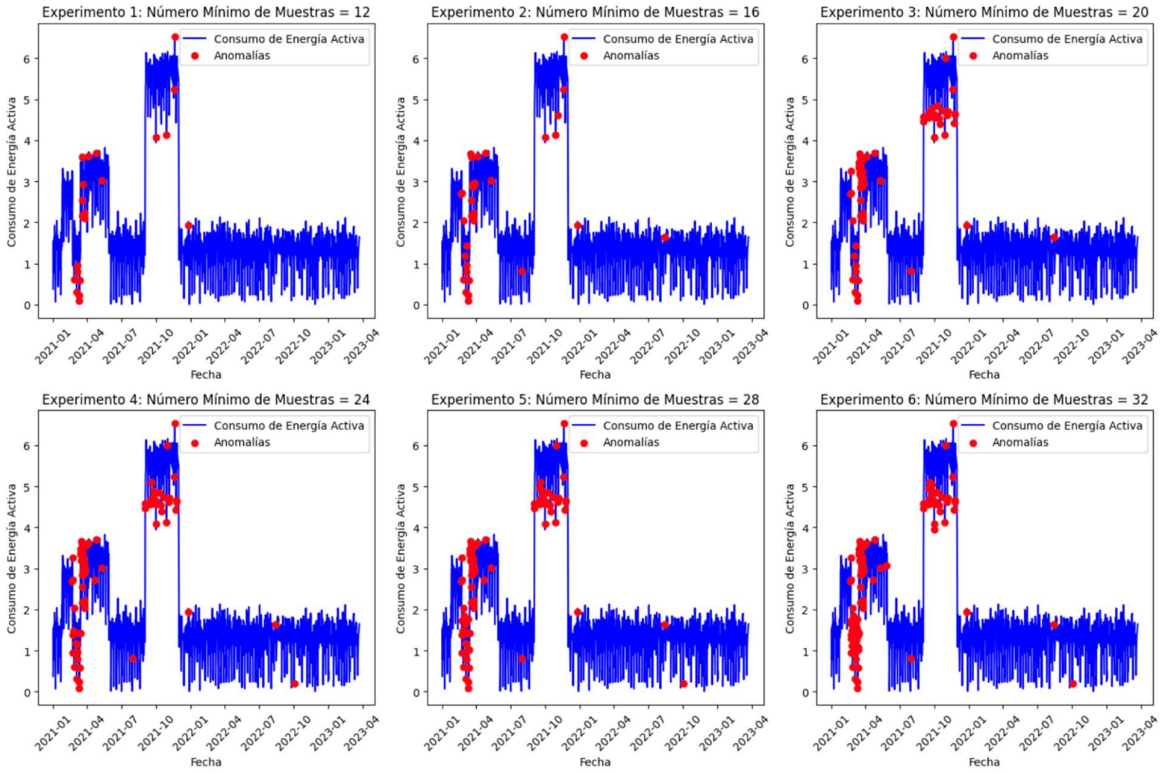
Desventajas

Es sensible a la elección de los parámetros epsilon y min_samples. La selección incorrecta de estos parámetros puede afectar significativamente la calidad de los resultados y la capacidad del algoritmo para detectar anomalías de manera efectiva. Además, DBSCAN puede tener dificultades para manejar conjuntos de datos de alta dimensionalidad debido al problema de la maldición de la dimensionalidad, lo que puede afectar su rendimiento en tales casos.

Parámetros y experimentos

Se realizan seis experimentos, variando el número mínimo de muestras utilizado por el algoritmo. Para cada experimento, se incrementa el número mínimo de muestras en 4 unidades, comenzando desde 12 y llegando a 32. Se elige esta secuencia para explorar cómo el número mínimo de muestras afecta la capacidad del algoritmo para detectar anomalías en los datos.

Experimento	Número mínimo de muestras	Anomalías Detectadas	% Anomalías	Precisión subjetiva
1	12	21	1,06%	0,10
2	16	35	1,76%	0,20
3	20	79	3,98%	0,40
4	24	88	4,43%	0,65
5	28	100	5,04%	0,60
6	32	135	6,80%	0,55



Observando los resultados, se nota que este modelo detecta anomalías de manera diferente en comparación con el método anterior. Las anomalías son identificadas en función del total de la serie, donde el umbral supera la distancia épsilon asignada. Cuando el parámetro es de 24 minutos de muestras, se observa que el modelo detecta anomalías específicas basadas en valores que están fuera de lo común en las observaciones.

A continuación, se presenta el parámetro para los demás clientes:

Número mínimo de Muestras	Cliente
20	4, 5, 6, 18, 19
24	1, 7, 8, 9, 20
28	2, 3, 16
30	10, 17

Método Descriptivo

Este método comienza calculando el primer y tercer cuartil, así como el rango intercuartil para la serie de tiempo. Luego, emplea un multiplicador de la desviación estándar, predeterminado en 1.5, para establecer un umbral basado en esta medida estadística. Las anomalías se identifican como aquellas observaciones que caen por debajo del primer cuartil menos el umbral multiplicado por el rango intercuartil (IQR), o por encima del tercer cuartil más el umbral multiplicado por el IQR

Ventajas

Es fácil de entender e implementar, ya que utiliza estadísticas simples como los cuartiles y la desviación estándar. Además, es relativamente resistente a valores extremos en los datos debido a la naturaleza robusta de los cuartiles y el rango intercuartil. Las anomalías se identifican basándose en umbrales

definidos de manera intuitiva, lo que facilita la interpretación de los resultados.

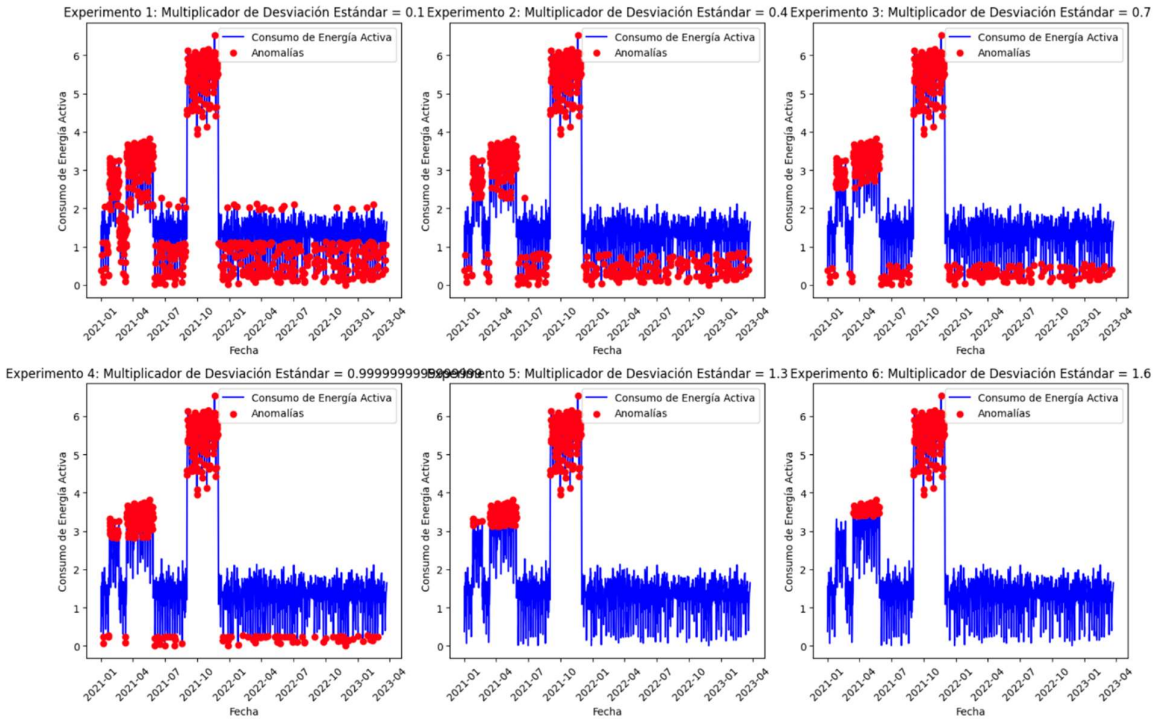
Desventajas

La detección de anomalías depende del valor del multiplicador de la desviación estándar: un valor bajo puede omitir anomalías importantes, mientras que uno alto puede generar falsas alarmas. Además, este método puede ser menos efectivo para detectar anomalías complejas que no sigan la distribución de los datos o estén influenciadas por factores externos no capturados por medidas estadísticas simples, y funciona mejor con datos que se ajusten a una distribución normal

Parámetros y experimentos

En cada experimento, se variará el multiplicador de desviación estándar, que determina el umbral para considerar un punto como anómalo. Esto permitirá explorar cómo diferentes configuraciones del parámetro afectan la capacidad del modelo para detectar anomalías en la serie temporal del consumo de energía activa. Se realizarán seis experimentos, incrementando el multiplicador de desviación estándar en 0.1 unidades en cada iteración, comenzando desde un valor base de 0.3. Los resultados son los siguientes:

Experimento	Multiplicador de Desviación Estándar	Anomalías Detectadas	% Anomalías	Precisión subjetiva
1	0,1	909	45,77%	0,0
2	0,4	861	34,29%	0,1
3	0,7	596	30,01%	0,2
4	1,0	474	23,87%	0,4
5	1,3	327	16,47%	0,35
6	1,6	274	13,80%	0,3



A diferencia de otros métodos, aquí el umbral asignado se aplica tanto en la parte superior como en la inferior de los datos. Se podría visualizar como una línea horizontal que busca identificar registros que

superen ese umbral, ya sea por encima o por debajo. Sin embargo, observamos que este enfoque no tiene en cuenta el comportamiento del consumo, lo que lo hace menos preciso.

A continuación, se presenta el parámetro para los demás clientes:

Multiplicador de Desviación Estándar	Cliente
0,7	4, 5, 7, 9, 16
1	1, 2, 3, 10, 17, 18, 19, 20
1,3	6, 8

KernelCPD

La librería ruptures de Python ofrece varias técnicas para facilitar el proceso determinar la presencia de puntos de cambio en una serie. Cada técnica combina tres operaciones: una función de costo, que se busca minimizar; un método de búsqueda, que ayuda a determinar si realmente se ha alcanzado un punto de cambio; y una función de penalización, que agrega costos en función del número de puntos de cambio detectados. KernelCPD es una versión optimizada de la técnica Pruned Exact Linear Time (PELT) para detectar puntos de cambio en una serie.

Al aplicar este enfoque se comienza por ordenar la serie de tiempo por fecha. Luego, se realizan una serie de pruebas estadísticas en la serie de tiempo para detectar anomalías registrando el resultado de las pruebas realizadas, el número de iteraciones y otros diagnósticos. Después, se determinan las anomalías en función de los resultados de las pruebas, el puntaje de sensibilidad y la fracción máxima de anomalías.

En las pruebas estadísticas se inicializan diccionarios para registrar las pruebas y sus resultados, así como para almacenar información adicional sobre la ejecución de las pruebas. Luego, se convierte la serie de tiempo a un arreglo, se preparan los kernels y los valores de penalización a probar durante las pruebas. Los kernels incluyen "linear", "rbf" y "cosine", mientras que los valores de penalización varían desde 0.001 hasta 1000 en una escala logarítmica. Después se realizan las pruebas con cada combinación de kernel y penalización utilizando KernelCPD y registrando los puntajes de anomalía detectados.

Para identificar las anomalías con base en los puntajes previamente calculados se consideran varios parámetros, incluyendo los resultados de las pruebas realizadas, el número total de iteraciones, el puntaje de sensibilidad y la fracción máxima de anomalías permitidas. El umbral de sensibilidad se calcula ajustando el número total de iteraciones para tener en cuenta la sensibilidad deseada. Luego, se calcula el puntaje máximo de anomalía permitido utilizando la fracción máxima de anomalías especificada. Si el puntaje máximo de anomalía es mayor que el umbral de sensibilidad ajustado, lo que indica que hay más valores atípicos de lo esperado, se actualiza el umbral de sensibilidad para igualar el puntaje máximo de anomalía. Finalmente, se asigna la etiqueta de anomalía indicando si cada observación es o no un valor atípico en función del umbral de sensibilidad calculado.

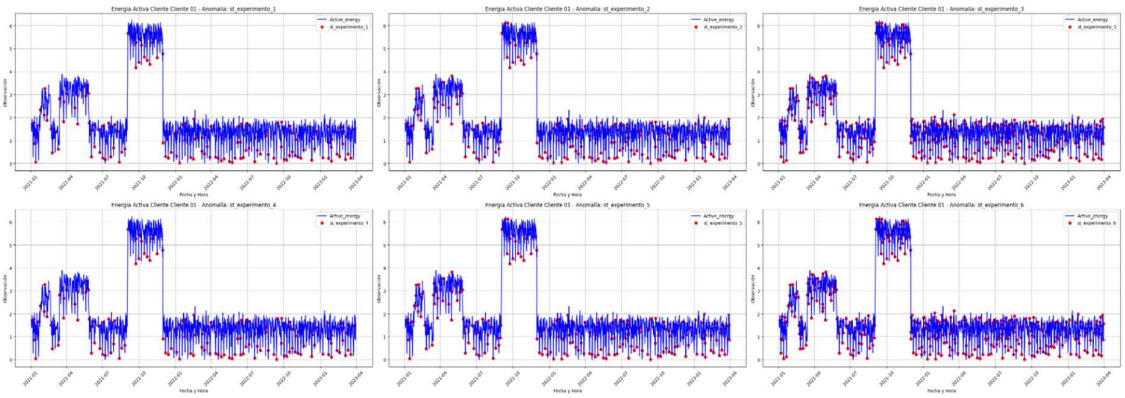
Ventajas

Capacidad para identificar cambios sutiles en los datos, flexibilidad para manejar diferentes tipos de distribuciones de datos. Eficiente en términos de tiempo de ejecución, especialmente cuando se trabaja con grandes volúmenes de datos.

Desventajas

Rendimiento puede verse afectado por la elección de parámetros, como el ancho de banda del kernel. Además, debido a su enfoque basado en kernels puede ser más sensible a ciertos tipos de ruido en los datos.

Experimento	Score de Sensibilidad	Máxima Fracción de Anomalías	Anomalías Detectadas	% Anomalías	Precisión subjetiva
1	80%	10%	168	8,55%	0,65
2	80%	20%	232	11,81%	0,5
3	80%	25%	457	23,27%	0,3
4	85%	10%	168	8,55%	0,6
5	85%	20%	232	11,81%	0,45
6	85%	25%	427	21,74%	0,35



Observando los resultados, se identifica que las anomalías son detectadas en función de las observaciones cercanas. El algoritmo captura tanto anomalías extremas como sutiles dentro de la serie de tiempo.

A continuación, se presenta el parámetro para los demás clientes:

Score de Sensibilidad	Máxima Fracción de Anomalías	Cliente
80%	10%	2,3,4, 6, 7, 9, 10, 17, 18, 19 y 20
80%	20%	5, 8, 16

Selección de la alternativa para detección de anomalías

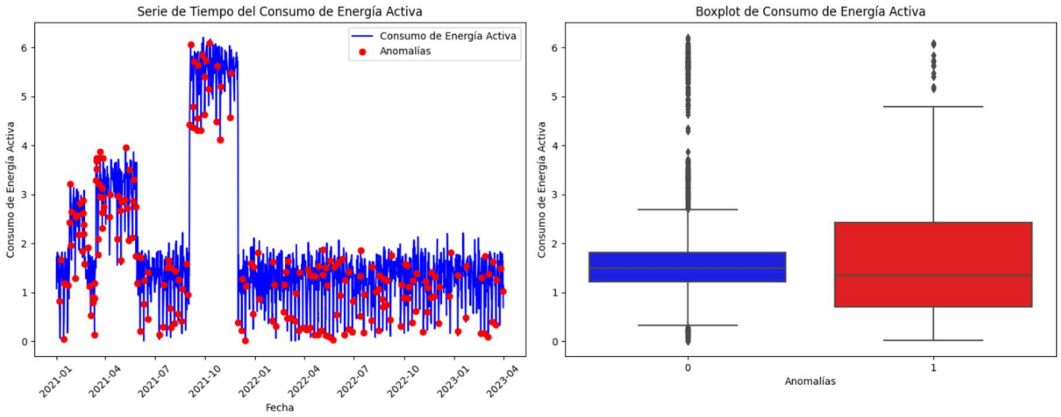
A continuación, se presenta el resumen de los métodos estudiados, donde se incluyen calificaciones para la interpretación de los modelos, su coherencia interna (la capacidad de los modelos para ofrecer resultados consistentes), robustez (la capacidad de los modelos para producir resultados sólidos con diferentes configuraciones) y tiempo de ejecución. Las calificaciones se clasificaron como baja, media y alta en comparación con los resultados generales de los modelos.

Método	% de Anomalías	Interpretación	Coherencia interna	Robustez	Tiempo de ejecución
Isolation Forest	5,0%	Modelo eficiente con grandes volúmenes de información y sin suposiciones sobre la distribución de los datos. Puede tener dificultades con anomalías sutiles o en regiones densas de datos.	Media	Baja	Media

DBSCAN	7,24%	Robusto frente a conjuntos de datos con estructuras complejas y ruido. Sensible a la elección de parámetros, especialmente epsilon y min_samples.	Alta	Media	Media
Descriptive Method	12,48%	Fácil de entender e implementar, utiliza estadísticas simples como los cuartiles y la desviación estándar. Sensible al valor del multiplicador de la desviación estándar.	Alta	Media	Baja
Kernel CPD	8,17%	Capacidad para identificar cambios sutiles en los datos, eficiente en términos de tiempo de ejecución. Sensible a la elección de parámetros, especialmente el ancho de banda del kernel.	Alta	Media	Media
Ensamble (DBSCAN + Kernel CPD)	16,7%	Combina las fortalezas de DBSCAN y Kernel CPD, equilibrando la identificación de anomalías extremas y sutiles.	Alta	Alta	Media

Después de analizar los resultados de cada alternativa, así como sus ventajas y desventajas, se decide combinar el método DBSCAN y el método KernelCPD, ya que permite obtener un equilibrio entre la identificación de anomalías extremas y sutiles manteniéndose eficiente en términos de precisión y tiempo de ejecución.

A continuación, se muestra gráficamente el resultado del ensamble de los dos modelos, en donde analizando los patrones de comportamiento da un resultado más acertado respecto a sus modelos pares de manera independiente.

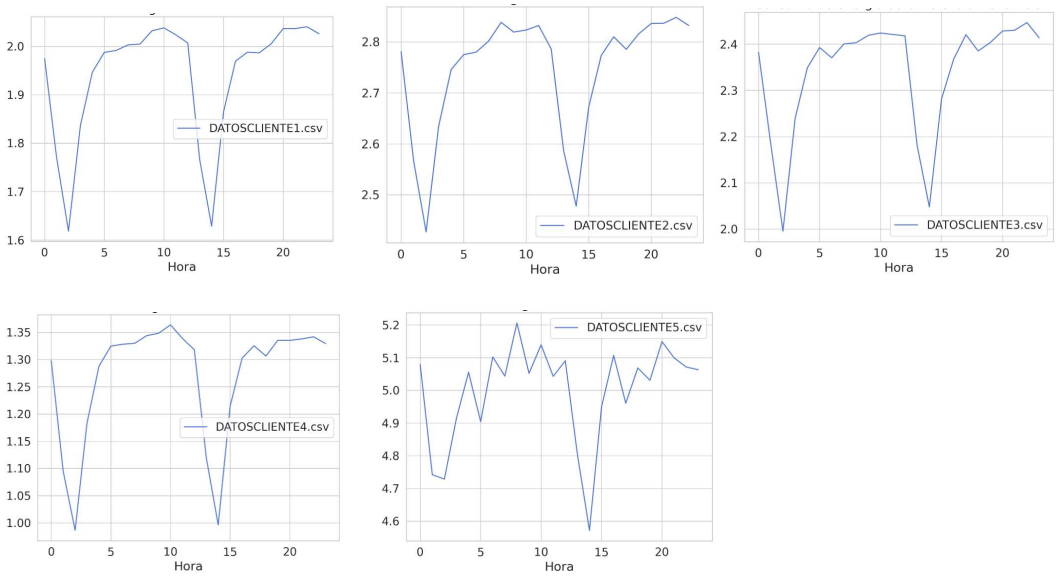


Pronóstico de Consumo

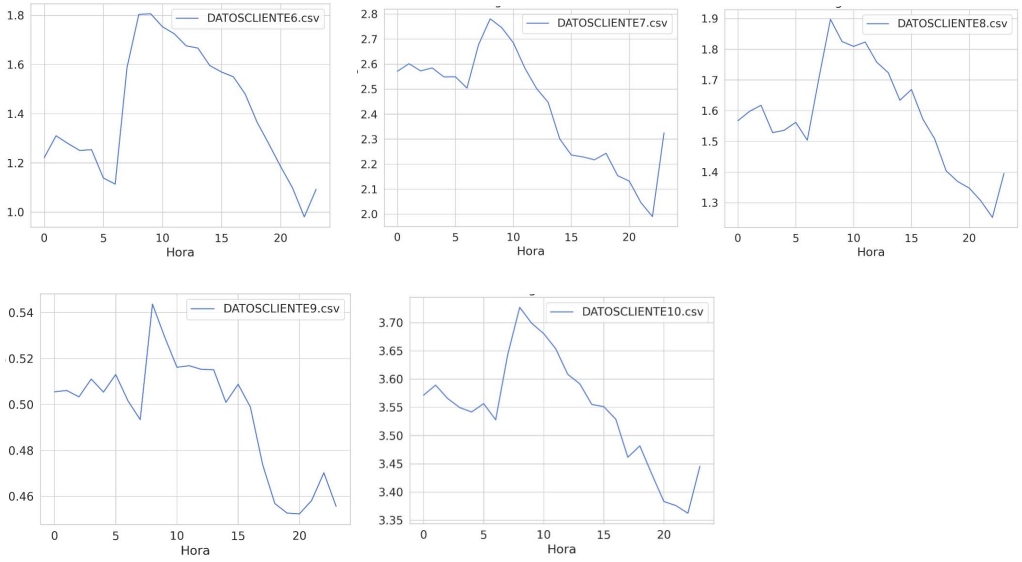
Para abordar de manera más optima, en cuanto a la exploración de modelos, selección de parámetros, ejecución y métricas de evaluación para los 15 clientes seleccionados. El enfoque es el comportamiento de consumo promedio por hora en un día de cada uno de los clientes, con ello se identificarán comportamientos similares y se agruparán los clientes por dichos comportamientos, así se tendrán 3 grupos y se realizara todos los experimentos para seleccionar el mejor modelo para cada uno de ellos.

Los grupos se encuentran conformados de la siguiente manera:

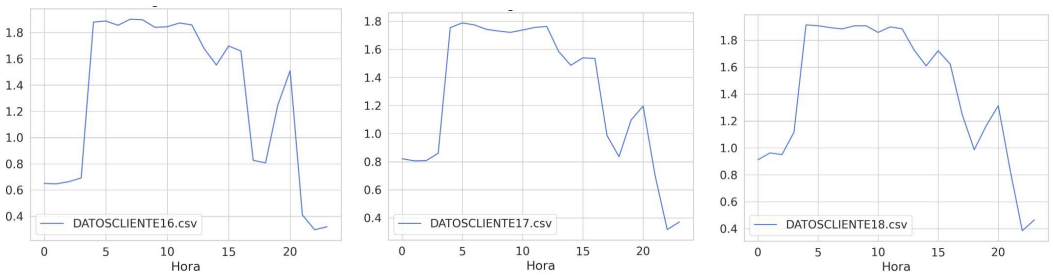
- Grupo 1: clientes 1, 2, 3, 4 y 5
  Su comportamiento de consumo promedio de energía activa es la siguiente:

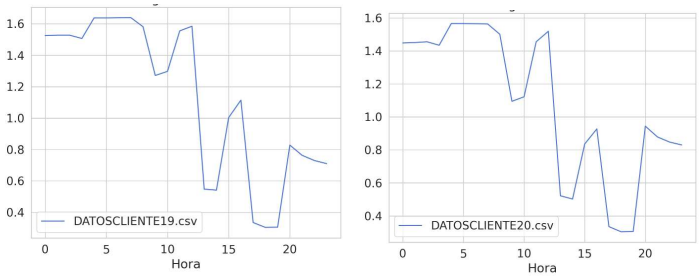


- Grupo 2: clientes 6, 7, 8, 9 y 10
  Su comportamiento de consumo promedio de energía activa es la siguiente:



- Grupo 3: clientes 16, 17, 18, 19 y 20
  Su comportamiento de consumo promedio de energía activa es la siguiente:





Cuando se observa los clientes en cada grupo existe la coincidencia que pertenecen al mismo sector economico, sin embargo, dado que el sector economico comprende subsectores que pueden dar más detalle de la labor o etapa de producción que realizan, se decide manterner el analisis por medio del comportamiento de consumo de energía activa. Cabe destacar que a su vez según la etapa de producción de cada cliente su historia puede tener comportamientos que den más información tanto de estacionalidad como de tendencia. Por ello el intervalo de tiempo disponible no permite tener un analisis más profundo sobre los componentes de la serie de tiempo.

Los modelos que se emplearan para realizar el pronóstico de la serie de tiempo para cada grupo son los siguientes:

- LSTM (Long Short-Term Memory):
    - Ventajas: Este modelo captura dependencias a largo plazo en los datos en modelos de series de tiempo, lo cual, permite manera relaciones compleja y no lineales en los datos. Cabe mencionar que este modelo puede manejar datos faltantes y valores atípicos, algo que todas las series de tiempo de los 15 clientes contiene, por ello suele ser una buena opción para realizar el pronostico
    - Desventajas: Se requiere una gran cantidad de datos para entrenar correctamente el modelo y evitar el sobreajuste de este. Sin embargo, los recursos computacionales y el tiempo que se debe emplear para su entrenamiento son alto en comparación con modelos más simples. Cabe mencionar que la búsqueda de los hiperparámetros y sus ajustes es de mucha atención y cuidado para obtener un buen rendimiento del modelo resultante.

- RNN (Redes Neuronales Recurrentes):
    - Ventajas: Este modelo puede modelar las dependencias temporales de los datos como tiempo, similar en el formato y estructura que deben transformase los datos para este tipo de RNA y LSTM. Adicional, este modelo tiene la flexibilidad de manejar diferentes longitudes y patrones secuenciales y dinámicos en los datos.
    - Desventajas: propenso al problema de desvanecimiento o explosión del gradiente, especialmente en secuencias largas. Adicional, es difícil de entrenar en comparación con otros modelos debido a la retroalimentación recurrente y como sucede en LSTM también requiere un alto uso de recursos computacionales y tiempo tanto para el entrenamiento como para la búsqueda de los hiperparámetros.

- Prophet:
    - Ventajas: es un modelo fácil de usar y de implementar, para usuarios que no son expertos en series de tiempo. Adicional, la capacidad de manejar de forma automática los diferentes tipos de patrones estacionales (diario, semana y anual) y tendencias en los datos. Esto resulta beneficioso dado que los clientes en su comportamiento diario y semanal tienen distintas tendencias o patrones de consumo.
    - Desventajas: Es menos flexible que modelos como LSTM y RNN y no es adecuado su implementación cuando los datos tiene patrones complejos o no lineales y se requiere de

una cantidad suficiente de datos históricos para identificar correctamente los patrones estacionales y de tendencia.

- SRV (Support Vector Regression):
  - Ventajas: Este modelo es robusto cuando se tiene valores atípicos en los datos, algo común en las series de tiempo de los distintos clientes y su regresión se puede emplear en conjuntos de datos pequeños o con pocos ejemplos de entrenamiento.
  - Desventajas: Como sucede en las RNA se debe tener bastante cuidado y atención a la selección y ajuste de los hiperparámetros (kernel y regularización).

Cabe mencionar que la selección inicial de estos modelos no tiene como propósito la interpretabilidad, ya que se desconoce la información de negocio de los clientes y el tiempo disponible puede ser una muestra del intervalo real de cada uno. Por ello, el objetivo es lograr un buen pronóstico con el menor error posible para cada grupo de clientes. En este contexto, las métricas de evaluación utilizadas serán el MSE (Error Cuadrático Medio), RMSE (Raíz del Error Cuadrático Medio), MAE (Error Absoluto Medio) y MAPE (Error Porcentual Absoluto Medio). Sin embargo, para la selección del mejor modelo de pronóstico de la serie temporal, las métricas clave serán el MAE y el MSE. El MAE nos proporcionará información sobre el rendimiento general del modelo, mientras que el MSE nos indicará la calidad del ajuste del modelo a los datos.

Parámetros y experimentos

LSTM: Cabe mencionar que, dado los recursos computacionales y de tiempo, el total de experimentos esperados no fueron realizados, por ende, se fijó valores en ciertos parámetros de la RNA

Parámetros	Valores usados en experimentos
Epochs	100, 200, 500
Batch size	24
Optimizer	adam
Learning rate	0.001
Capas	2
Unidades	40, 64, 128
Drop out	0.1, 0.15, 0.2
Activación	relu, tanh
Función de perdida	MSE

métricas del mejor modelo para cada grupo de clientes:

Métrica	Grupo 1	Grupo 2	Grupo 3
MSE	0,0580	0,0431	0,0134
MAE	0,1901	0,1665	0,0926

Modelo seleccionado:

Parámetros	Valores
Epochs	200
Batch size	24
Optimizer	adam
Learning rate	0.001

Capas	2
Unidades	128 (capa 1) y 64 (capa 2)
Drop out	0.2
Activación	Tanh (capa 1) y relu (capa 2)
Función de perdida	MSE

RNN: de la misma forma como se realizó con LSTM se fijaron valores para ciertos parámetros

Parámetros	Valores usados en experimentos
Epochs	100, 200, 500
Batch size	24
Optimizer	adam
Learning rate	0.001
Capas	2
Unidades	40, 100, 120
Drop out	0.1, 0.15, 0.2
Activación	relu, tanh
Función de perdida	MSE

métricas del mejor modelo para cada grupo de clientes:

Métrica	Grupo 1	Grupo 2	Grupo 3
MSE	0,0686	0,0476	0,0182
MAE	0,2069	0,1754	0,1067

Modelo seleccionado:

Parámetros	Valores
Epochs	200
Batch size	24
Optimizer	adam
Learning rate	0.001
Capas	3
Unidades	40
Drop out	0.2
Activación	tanh
Función de perdida	MSE

Prophet: En este modelo se realizó experimentos en default, agregando estacionalidad horaria, diaria y semanal y se agregaron dichas estacionalidades como regresores. Cabe mencionar que, dado que este modelo funciona con estacionalidad y tendencia, se pueden crear funciones que capturen de forma más personalizada dichos componentes. Con ello se construye para cada grupo un parámetro que indique los valores más altos y bajos del comportamiento promedio de la energía a nivel hora del día y a nivel de los días de la semana. Con ello se tiene 4 funciones con los valores más altos de cada hora y día de semana y los valores más bajos en cada hora del día y día de la semana.

Los experimentos iniciales dan que el mejor modelo es cuando se agrega estas funciones como

estacionalidad, con ello se realiza la búsqueda del resto de los parámetros.

Parámetros	Valores usados en experimentos
changepoint_prior_scale	0.001, 0.01, 0.1
seasonality_prior_scale	0.01, 1.0
daily_seasonality	True, False
weekly_seasonality	True, False
seasonality_mode	additive, multiplicative
Rolling_window	5
cut_offs	2022-04-01 a 2022-08-01
Horizon	10 days
Fourier_order	10

métricas del mejor modelo para cada grupo de clientes:

Métrica	Grupo 1	Grupo 2	Grupo 3
MSE	0,2400	0,1160	0,1220
MAE	0,3840	0,2800	0,2820

El modelo para cada grupo es el siguiente:

Parámetros	Grupo 1	Grupo 2	Grupo 3
changepoint_prior_scale	0.01	0.1	0.01
seasonality_prior_scale	0.01	1.0	1.0
daily_seasonality	False	True	False
weekly_seasonality	True	True	False
seasonality_mode	additive	multiplicative	multiplicative

SVR (Support Vector Regression):

Parámetros	Valores usados en experimentos
Kernel	RBF
Gamma	0.1, 0.3, 0.5
C	1, 10, 100
Epsilon	0.01, 0.05, 0.1
Timesteps	5, 12, 18, 24

Métricas del mejor modelo para cada grupo de clientes:

Métrica	Grupo 1	Grupo 2	Grupo 3
MSE	0,0800	0,0190	0,0254
MAE	0,2230	0,1094	0,1276

El modelo para cada grupo es el siguiente:

Parámetros	Grupo 1	Grupo 2	Grupo 3
Kernel	RBF	RBF	RBF
Gamma	0.5	0.3	0.1
C	10	10	1
Epsilon	0.05	0.05	0.1
Timesteps	18	5	24

Cabe mencionar que el desarrollo de cada modelo consta antes de un procesamiento y validación de la información, se utilizó en los 4 modelos escalamiento estándar (StandarScaler) y la partición del conjunto de datos fue un 90% para el entrenamiento y 10% para prueba.

Dado que existe una correlación de 0.84 entre la energía activa y energía reactiva, en esta fase del PMV se tomará estos modelos para realizar su pronóstico de energía reactiva. En próximas fases al ser aprobado el PMV se construirá un modelo particular para cada cliente tanto para la energía activa como energía reactiva.

Modelo	Métrica	Grupo 1	Grupo 2	Grupo 3
LSTM	MSE	0,058	0,0431	**0,0134**
	MAE	0,1901	0,1665	**0,0926**
RNN	MSE	0,0686	0,0476	0,0182
	MAE	0,2069	0,1754	0,1067
Prophet	MSE	0,24	0,116	0,122
	MAE	0,384	0,28	0,282
SVR	MSE	**0,0154**	**0,0190**	0,0254
	MAE	**0,0981**	**0,1094**	0,1276

El mejor modelo para el Grupo 1 y 2 es el SVR mientras para el Grupo 3 es el LSTM. De igual manera los otros modelos pueden ser refinados y se podría realizar una búsqueda más cuidadosa de los hiperparámetros en una siguiente fase del proyecto.

Clasificación de Anomalías del Pronóstico de Consumo

Para clasificar las anomalías futuras con base en la identificación de anomalías históricas y en los grupos de clientes con comportamiento similares se implementaron los siguientes modelos de clasificación Logistic Regression, SVM, Random Forest y Gradient Boosting. El área bajo la curva ROC (AUC) se utilizó como métrica de evaluación. Cada modelo se ajustó usando GridSearchCV para encontrar los mejores hiperparámetros.

Variables de entrada: 'Active_energy', 'Reactive_energy', 'Voltaje_FA' y 'Voltaje_FC'
Variable objetivo: 'Anomalies_final'.

Se dividen los datos en conjuntos de entrenamiento y prueba asignando el 80% de los datos para entrenamiento y el 20% para prueba.

Se itera sobre cada modelo construyendo un pipeline que incluye un escalador estándar y el clasificador del modelo en que se está iterando. Luego, se utiliza GridSearchCV para buscar los mejores hiperparámetros para el modelo de la interacción utilizando validación cruzada con 5 particiones.

Aquí tienes una lista de ventajas y desventajas comunes para los modelos supervisados de clasificación:

Regresión Logística

Es un método de clasificación que utiliza la función logística para modelar la relación entre las variables independientes y la variable dependiente. Se utiliza para predecir la probabilidad de que ocurra un evento binario (por ejemplo, sí es anomalía o no es anomalía).

Ventajas

Fácil de entender e interpretar, es útil para entender la relación entre variables independientes y la variable objetivo, puede proporcionar probabilidades de pertenencia a una clase específica y es menos propenso al sobreajuste en comparación con otros modelos más complejos.

Desventajas

No es tan flexible como otros modelos más avanzados, no maneja bien las relaciones no lineales entre las variables predictoras y la variable objetivo. Sensible a características irrelevantes o ruidosas y no es adecuado para problemas con múltiples clases si no se utiliza una extensión multinomial.

Máquinas de Soporte Vectorial (SVM)

Utilizado para problemas de clasificación y regresión. Su objetivo es encontrar el hiperplano óptimo que mejor separa las clases en un espacio de características de alta dimensión

Ventajas

Eficaz en espacios de alta dimensión, incluso cuando el número de dimensiones es mayor que el número de muestras. Versátiles, ya que diferentes funciones de kernel pueden ser especificadas para el hiperplano de decisión. Pueden manejar eficazmente conjuntos de datos con una separación no lineal entre clases utilizando. Robusto ante el problema de la maldición de la dimensionalidad.

Desventajas

No es adecuado para conjuntos de datos muy grandes debido a su alta complejidad computacional. Es sensible a la elección del parámetro de regularización y la función de kernel. No proporciona directamente probabilidades de pertenencia a las clases, lo que requiere una técnica adicional como la calibración de probabilidades. Puede ser difícil de interpretar, especialmente en espacios de alta dimensión con funciones de kernel complejas.

Random Forest (Bosques Aleatorios)

Utilizado en problemas de clasificación y regresión. Combina múltiples árboles de decisión durante el entrenamiento, formando así un "bosque aleatorio". Cada árbol se entrena independientemente con un conjunto de datos aleatorio, seleccionando características de manera aleatoria en cada división de nodo. Durante la predicción, cada árbol emite su propia predicción y la clase final se determina mediante votación en el caso de la clasificación o promediando en la regresión.

Ventajas

Reduce el sobreajuste al promediar múltiples árboles (ensamble). Proporciona una medida de la importancia de las características para la clasificación. Capaz de manejar un gran número de características y datos faltantes. Menos propenso al sobreajuste en comparación con árboles de decisión individuales.

Desventajas

No es tan fácil de interpretar como un árbol de decisión individual. Puede ser computacionalmente costoso, especialmente con un gran número de árboles y características. Menos eficaz en conjuntos de

datos muy dispersos o dispersos. Puede no funcionar bien en problemas con datos desequilibrados si no se ajusta correctamente.

Gradient Boosting

Se utiliza en u problemas de clasificación y regresión. Su funcionamiento implica la construcción secuencial de múltiples modelos de predicción débiles, como árboles de decisión simples. En cada iteración, el algoritmo ajusta un nuevo modelo para corregir los errores del modelo anterior, minimizando una función de pérdida que indica la discrepancia entre las predicciones actuales y los valores reales del conjunto de datos. Este proceso se lleva a cabo mediante el gradiente descendente, ajustando gradualmente el modelo para reducir el error y mejorar la precisión de las predicciones.

Ventajas

Proporciona predicciones precisas, ya que ajusta secuencialmente los modelos para corregir errores. Puede trabajar con diferentes tipos de datos y modelos débiles. El proceso de ajuste secuencial ayuda a reducir el sobreajuste, ya que cada nuevo modelo se enfoca en corregir los errores del modelo anterior. Maneja naturalmente valores faltantes en los datos sin necesidad de preprocesamiento adicional.

Desventajas

Sensibles a los hiperparámetros, los cuales deben ajustarse correctamente para obtener el mejor rendimiento, lo que puede requerir tiempo y esfuerzo. Debido a su naturaleza secuencial, el entrenamiento de estos modelos puede llevar más tiempo en comparación con otros modelos, especialmente en conjuntos de datos grandes. Si los hiperparámetros no se ajustan correctamente o si se utilizan demasiados modelos débiles, existe el riesgo de sobreajuste en los datos de entrenamiento, lo que puede afectar la capacidad del modelo para generalizar a nuevos datos

Parámetros y experimentos

Agrupación de clientes:

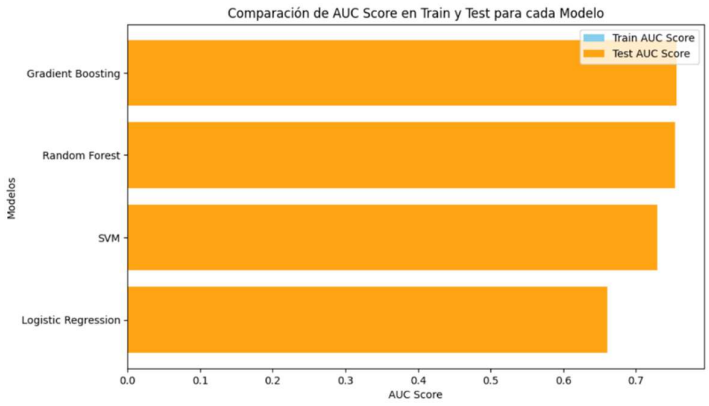
Grupo 1 = ['Cliente 01', 'Cliente 02', 'Cliente 03', 'Cliente 04', 'Cliente 05']
Grupo 2 = ['Cliente 06', 'Cliente 07', 'Cliente 08', 'Cliente 09', 'Cliente 10']
Grupo 3 = ['Cliente 16', 'Cliente 17', 'Cliente 18', 'Cliente 19', 'Cliente 20']

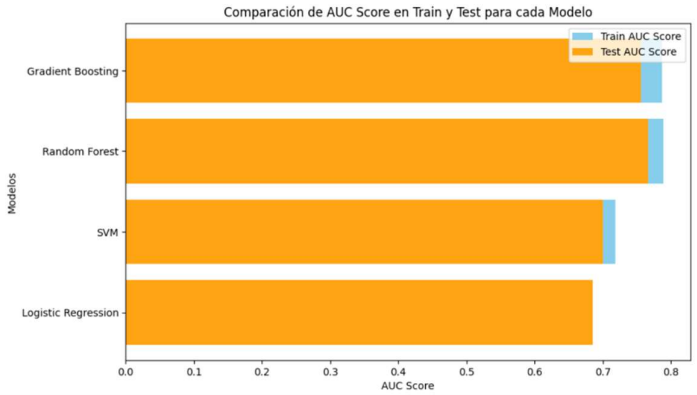
Esta tabla presenta una comparación detallada de los modelos de clasificación utilizados para cada grupo de clientes, junto con sus respectivos hiperparámetros parámetros de entrenamiento, así como los hiperparámetros identificados para el mejor modelo en cada experimento. Además, se resaltan los valores de AUC tanto en entrenamiento como en prueba para el mejor modelo seleccionado dentro de cada grupo de las alternativas evaluadas. Los resultados proporcionan una visión comparativa de cómo cada modelo se desempeña en cada grupo en términos de capacidad de generalización y ajuste al conjunto de datos de entrenamiento.

Grupo de Clientes	Modelo	Parámetros y Experimentos	Parámetros para el mejor modelo	Mejor AUC (Train	Mejor AUC (Test)
Grupo 1	Logistic Regression	{'classifier__C': [0.1, 1, 10, 100]}	'classifier__C':100	0,644472	0,660147
	SVM	{classifier__C': [0.1, 1, 10, 100], 'classifier__gamma': [0.1, 1, 10]}	'classifier__C': 1, 'classifier__gamma': 1	0,720263	0,729121
	Random Forest	{'classifier__n_estimators': [50, 100, 200], 'classifier__max_depth': [3, 5, None]},	'classifier__max_depth' : None, 'classifier__n_estimators': 100	0,749440	0,754253

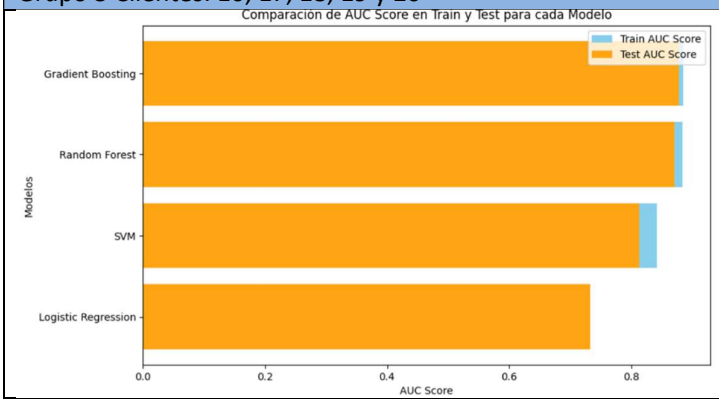
	Gradient Boosting	{classifier__n_estimators': [50, 100, 200], 'classifier__learning_rate': [0.01, 0.1, 1]	'classifier__learning_rate': 0.1, 'classifier__n_estimators': 200	0,741492	0,756270
Grupo 2	Logistic Regression	{'classifier__C': [0.1, 1, 10, 100]}	'classifier__C': 1	0,685351	0,684945
	SVM	{classifier__C': [0.1, 1, 10, 100], 'classifier__gamma': [0.1, 1, 10]}	'classifier__C': 0.1, 'classifier__gamma': 10	0,718294	0,700313
	Random Forest	{'classifier__n_estimators': [50, 100, 200], 'classifier__max_depth': [3, 5, None]},	'classifier__max_depth': 5, 'classifier__n_estimators': 200	0,789471	0,766102
	Gradient Boosting	{classifier__n_estimators': [50, 100, 200], 'classifier__learning_rate': [0.01, 0.1, 1]	'classifier__learning_rate': 0.1, 'classifier__n_estimators': 200	0,786646	0,756199
Grupo 3	Logistic Regression	{'classifier__C': [0.1, 1, 10, 100]}	'classifier__C': 0.1	0,730471	0,733102
	SVM	{classifier__C': [0.1, 1, 10, 100], 'classifier__gamma': [0.1, 1, 10]}	'classifier__C': 0.1, 'classifier__gamma': 10	0,842562	0,813278
	Random Forest	{'classifier__n_estimators': [50, 100, 200], 'classifier__max_depth': [3, 5, None]},	'classifier__max_depth': None, 'classifier__n_estimators': 200	0,883772	0,871223
	Gradient Boosting	{classifier__n_estimators': [50, 100, 200], 'classifier__learning_rate': [0.01, 0.1, 1]	'classifier__learning_rate': 0.1, 'classifier__n_estimators': 200	0,885888	0,878240

A continuación, se presenta la comparación de la métrica AUC en entrenamiento y test para los mejores modelos en cada alternativa evaluada:

Grupo 1 Clientes: 1, 2, 4, y 5	
	Los modelos Random Forest y Gradient Boosting muestran los mejores resultados en entrenamiento y pruebas. Random Forest: AUC Train = 0,749440 y AUC Test = 0,754253 Gradient Boosting: AUC Train = 0,741492 y AUC Test = 0,756270
Grupo 2 Clientes: 6, 7, 8, 9 y 10	

<table>
<tr><td> 
</td><td>Los modelos Random Forest y Gradient Boosting muestran los mejores resultados en entrenamiento y pruebas.

Random Forest: AUC Train = 0,789471 y AUC Test = 0,766102

Gradient Boosting: AUC Train = 0,786646 y AUC Test = 0,756199</td></tr>
<tr><td colspan="2">Grupo 3 Clientes: 16, 17, 18, 19 y 20</td></tr>
<tr><td> 
</td><td>Los modelos Random Forest y Gradient Boosting muestran los mejores resultados en entrenamiento y pruebas.

Random Forest: AUC Train = 0,883772 y AUC Test = 0,871223

Gradient Boosting: AUC Train = 0,885888 y AUC Test = 0,878240</td></tr>
</table>

Dado que ambos modelos, Random Forest y Gradient Boosting, muestran resultados bastante similares tanto en el conjunto de entrenamiento como en el de pruebas, con puntajes de AUC muy cercanos, la decisión entre ellos podría depender de otros factores como la complejidad del modelo, el tiempo de entrenamiento y la interpretabilidad. Si la interpretabilidad es una preocupación y se prefiere un modelo más fácil de entender, podría optarse por Random Forest. Por otro lado, si la precisión es la principal consideración y el tiempo de entrenamiento no es un problema, Gradient Boosting podría ser la elección preferida debido a su capacidad para ajustarse mejor a los datos.

Dado que no contamos con las preferencias del usuario, para la implantación del MVP asumiremos que la precisión es la principal consideración y que el tiempo de entrenamiento no es una restricción. Por lo tanto, seleccionaremos el mejor modelo de Gradient Boosting para cada grupo de clientes.

Componentes, Características o Requerimientos Pendientes

Id.	Actividad	Fecha
1	Ajustar prototipo fachada y tabla de requerimientos con base en la retroalimentación de la profesora Natalia	8-May-24
2	Completar el ensamblaje del prototipo: desarrollo de la interfaz del usuario e integración con los modelos seleccionados en Flask	6-May-24 15-May-24
3	Validación del prototipo con base en los requerimientos, los componentes del artefacto, y las pruebas y métricas propuestas. Diligenciar la rúbrica.	16-May-24 17-May-24
4	Desarrollar el manual de usuario	17-May-24 18-May-24
5	Script para la presentación	19-May-24

6	Realizar el video de la presentación ejecutiva	20-May-24
7	Presentar el video	23-May-24
8	Entregar el prototipo (los archivos ejecutables, código fuente, manual de usuario, pruebas del prototipo fachada y la rúbrica diligenciada)	23-May-24