Proyecto Final de Aprendizaje de Máquina

Estudio sobre registros de consumo de datos, voz y sms

Integrantes:

Claudia Hernández Pérez
Joel Aparicio Tamayo
Kevin Márquez Vega
Javier A. González Díaz
José Miguel Leyva Cruz
Luis E. Amat Cárdenas

Grupo: C-412

Índice

1. Estudio del estado del arte

La investigación sobre predicción y clasificación de tráfico en redes ha evolucionado de manera notable en los últimos quince años. En la primera etapa (2010–2013), los modelos estadísticos clásicos como ARMA, ARIMA y SARIMA fueron la base para el análisis de series temporales, mientras que las redes neuronales superficiales como MLP (Multi-Layer Perceptron) demostraron capacidad para capturar patrones complejos antes del auge del DL (Deep Learning).

A partir de 2014, el DL (Deep Learning) comenzó a aplicarse con redes profundas como DNN (Deep Neural Network), mostrando mejoras claras frente a los métodos tradicionales. Posteriormente, los modelos híbridos que combinan CNN (Convolutional Neural Network) y LSTM (Long Short-Term Memory) permitieron capturar tanto correlaciones espaciales como temporales, consolidando el enfoque de Tráfico espacio-temporal en la predicción de tráfico.

Entre 2017 y 2018 surgieron propuestas orientadas a la eficiencia y nuevas representaciones: RCLSTM redujo parámetros manteniendo rendimiento, mientras que la representación del tráfico como imágenes espacio-temporales con CNN 2D mejoró la precisión. En paralelo, se exploraron integraciones más complejas como DMVST-Net, y se mantuvo la relevancia de algoritmos clásicos como SVM (Support Vector Machine), K-NN (K-Nearest Neighbors) y árboles, en tareas de clasificación.

En el periodo 2019–2021 se diversificaron las técnicas, incorporando clustering y reducción de dimensionalidad por ejemplo: K-means, DBSCAN y PCA (Principal Component Analysis), además de modelos optimizados como SVR (Support Vector Regression) y XGBoost, que superaron a los estadísticos en escenarios no lineales. Las comparativas entre ML (Machine Learning) y DL (Deep Learning) confirmaron la superioridad de LSTM (Long Short-Term Memory), aunque MLP (Multi-Layer Perceptron) se mantuvo competitivo.

Los avances recientes (2022–2023) destacan el uso de K-NN (K-Nearest Neighbors) con selección de características en LTE (Long Term Evolution), la incorporación de grafos mediante GNN (Graph Neural Network) combinados con RNN (Recurrent Neural Network) para modelar dependencias espaciales y temporales, y la validación de modelos estadísticos como SARIMA y Holt-Winters en contextos específicos. También se resaltan alternativas rápidas y robustas como OS-ELM y Random Forest.

Finalmente, las tendencias emergentes (2024–2025) apuntan hacia arquitecturas modernas como Transformer y TCN (Temporal Convolutional Network), capaces de capturar dependencias globales y temporales con gran

precisión. Se exploran enfoques eficientes como HTM (Hierarchical Temporal Memory) y FL (Federated Learning) para preservar privacidad, y se proponen marcos híbridos adaptativos para entornos 5G / 6G. Las revisiones recientes concluyen que el aprendizaje profundo moderno con LSTM (Long Short-Term Memory), GNN (Graph Neural Network) y Transformer domina en problemas Tráfico espacio-temporal complejos, mientras que los métodos ML clásico siguen siendo útiles en escenarios simples o con restricciones de recursos.

2. Estudio sobre los datos

La Empresa de Telecomunicaciones de Cuba S.A. (ETECSA) proporcionó un conjunto de datos con el propósito de desarrollar estudios y modelos basados en técnicas de Aprendizaje de Máquina (Machine Learning). Dichos datos reflejan el uso de diversos servicios de telecomunicaciones por parte de los usuarios, tales como llamadas telefónicas, mensajes de texto (SMS), recargas de saldo y consumo de datos móviles.

El objetivo principal de este análisis es comprender la estructura, el contenido y las características de los datos, con vistas a su preparación y posterior aplicación en modelos predictivos o de análisis de comportamiento.

2.1. Descripción general del dataset

El conjunto de datos se encuentra en formato tabular y contiene aproximadamente 10 mil registros y 40 variables, distribuidas en columnas que describen los diferentes aspectos de cada transacción o evento de uso de servicios.

Cada fila representa un registro detallado de uso de servicio (CDR, por sus siglas en inglés: Call Detail Record), que documenta información relacionada con un evento generado por el cliente, como una llamada, el envío de un mensaje o una conexión a internet móvil.

Los registros reportan fechas entre las 4 de la madrugada y 12 del mediodía en UTC (lo que serían las 12 de la madrugada y 8 de la mañana hora de Cuba 'UTC-4') del día 1 de octubre de 2025.

A continuación, se presenta un resumen de los tipos de variables más relevantes (ver Cuadro 1):

Tipo de variable	Ejemplo de campos	Descripción general
Identificadores	CDR_ID, OBJ_ID, OWNER_CUST_ID	Identifican de manera única cada registro, objeto o cliente asociado.
Temporales	START_DATE, END_DATE	Indican la fecha y hora de inicio y fin del servicio utilizado.
Categóricas	SERVICE_CATEGORY, FLOW_TYPE, USAGE_SERVICE_TYPE	Especifican el tipo de servicio, su categoría (voz, datos, SMS, recarga) y dirección del tráfico (entrante/saliente).
Numéricas	ACTUAL_USAGE, ACTUAL_CHARGE, TOTAL_TAX_AMOUNT	Miden el volumen de uso (minutos, megabytes, mensajes) y los cargos monetarios asociados.
Listas o estructuras anidadas	CHARGE_LIST, CHARGE_SERVICE_INFO, BALANCE_CHG_LIST	Detallan cargos, impuestos y modificaciones de saldo que se producen en cada evento.

Cuadro 1: Descripciones de los datos.

Estos campos se complementan con información auxiliar relacionada con unidades de medida, identificadores de cuenta, ciclos de facturación y valores reservados para futuras ampliaciones del sistema.

2.2. Distribución de los datos

Con el fin de visualizar el comportamiento de los datos se muestran sus distribuciones (ver Figura 1).

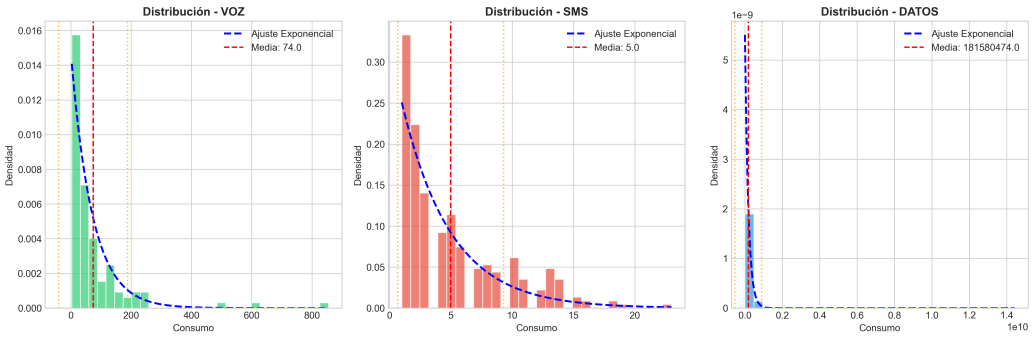

Figura 1: Distribución de los datos

Las gráficas muestran ciertas irregularidades que resultan de interés. La

intención es realizar un estudio más profundo de estas anomalías por independiente, dado que no existe ninguna relación entre ellos (ver Figura 2).

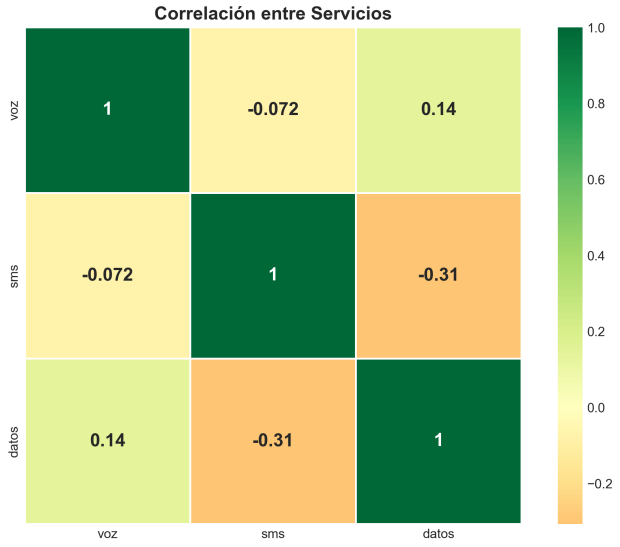

Figura 2: Correlación entre servicios.

3. Detección de patrones

Como mostraron las gráficas en cada uno de los servicios estudiados se observan "picos" que pueden describir eventos en los datos (ver Figura 3).

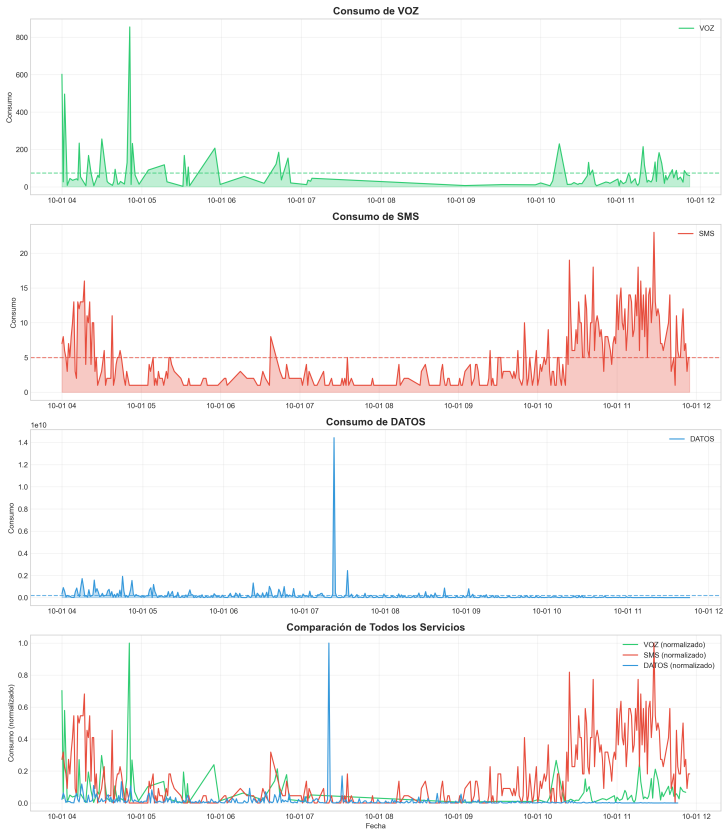

Figura 3: Comparación de consumo de servicios.

Se analizaron los comportamientos más relevantes de cada servicio utilizando DBSCAN y los patrones de horario por servicio usando series de tiempo.

3.1. Datos

En las horas registradas en el horario entre las 4 y 6 de la madrugada (ver Figura 4), se refleja un notable consumo, que pudiera estar relacionado con las promociones de recargas con datos ilimitados en el horario de la madrugada.

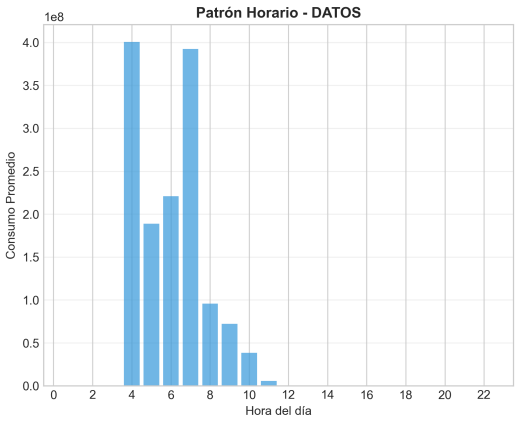

Figura 4: Patrones de horario del servicio de datos.

Tras un análisis realizado sobre el uso del servicio en un intervalo de tiempo se observaron anomalías extremas, habían registros de muy bajo consumo pero también de muy alto. Esto se refleja en una notable diferencia entre la media de consumo de datos en estas anomalías y en los consumidores "normales" (ver Cuadro 2).

Variable	**Valor (Byte)**
Media Normal	7 853 641.44
Media Outlier	249 439 482.75
Min Outlier	64.0
Max Outlier	14 114 331 638.0

Cuadro 2: Resumen de valores de la variable ACTUAL_USAGE

3.2. Voz

En las horas registradas se muestra un consumo significativo en el horario entre las 12 y 2 de la madrugada (ver Figura 5).

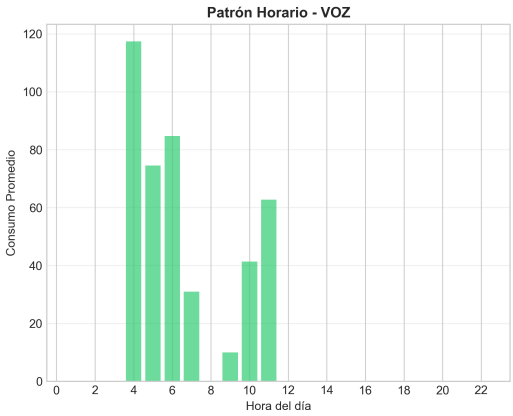

Figura 5: Patrones de horario del servicio de voz.

Se identificaron anomalías significativas en las llamadas. Se consideraron aquellas llamadas cuya duración superaba ampliamente la media, utilizando la columna `ACTUAL_USAGE` (ver Cuadro 3). Además, se examinó la variable `ACTUAL_CHARGE` (ver Cuadro 4), la cual mostró una irregularidad marcada: la mayoría de los registros presentan valores en cero, mientras que algunos alcanzan cifras superiores a 2 000. Finalmente, se analizó la cantidad de llamadas recibidas por los usuarios (ver Cuadro 5)

Variable	**Valor (min)**
Media Normal	36.52
Media Outlier	310.5
Min Outlier	97.0
Max Outlier	831.0

Cuadro 3: Resumen de valores de la variable ACTUAL_USAGE.

Variable	**Valor (centavos)**
Media Normal	0.0
Media Outlier	18 888.89
Min Outlier	2 500.0
Max Outlier	80 500.0

Cuadro 4: Resumen de valores de la variable ACTUAL_CHARGE.

Variable	Valor
Media Normal	1.20
Media Outlier	5.0
Min Outlier	4
Max Outlier	6

Cuadro 5: Resumen de valores de la variable llamadas recibidas.

3.3. SMS

En las horas registradas se observa un pequeño aumento del consumo en el horario entre las 6 y 8 de la mañana (ver Figura 6). No obstante no se distinguen diferencias significativas.

Figura 6: Patrones de horario del servicio de sms.

En el caso de los mensajes de texto se realizó un análisis específico considerando únicamente la cantidad de SMS recibidos, utilizando la columna `OTHER_NUMBER`. Se identificaron outliers que reflejan un comportamiento anómalo en comparación con los usuarios normales. Mientras que la mayoría de los registros presentan un número reducido de mensajes recibidos, algunos casos muestran valores significativamente superiores (ver Cuadro 6).

Variable	Valor
Media Normal	2.54
Media Outlier	25.96
Min Outlier	7
Max Outlier	168

Cuadro 6: Resumen de valores de la variable mensajes recibidos.

4. Reducción de dimensionalidad y ruido

Si bien se tienen pocas variables que no deberían suponer un problema al ejecutar los algoritmos clásicos, existe una alta correlación (ver Cuadro 7) entre algunas de ellas, de las que se pueden prescindir.

Variable 1	**Variable 2**	**Correlación**
RATING_USAGE	ACTUAL_USAGE	1.000000
DEFAULT_ACCT_ID	OBJ_ID	1.000000
DEFAULT_ACCT_ID	OWNER_CUST_ID	1.000000
OWNER_CUST_ID	OBJ_ID	1.000000
OBJ_ID	CDR_ID	0.913608
DEFAULT_ACCT_ID	CDR_ID	0.913608
OWNER_CUST_ID	CDR_ID	0.913607

Cuadro 7: Correlaciones entre variables seleccionadas

Para abordar este problema se proponen dos enfoques con supuestos conceptuales distintos:

- **Análisis de Componentes Principales (PCA)**: considera que la información está contenida en la varianza de los datos.
- **Análisis de Componentes Independientes (ICA)**: asume que la información se encuentra en la independencia estadística de las variables.

Los resultados muestran que las dos primeras componentes extraídas por PCA e ICA corresponden principalmente a los identificadores de usuario y al consumo, junto con las columnas altamente correlacionadas con ellos.

En contraste, las variables `RESERVE[0-10]` no aportan información relevante y no justifican un análisis más profundo. De este modo, se concluye que los aspectos más significativos para el estudio son los relacionados con los usuarios y su nivel de consumo.

Finalmente, se observa que tanto PCA como ICA omitieron las variables temporales. Esto resulta coherente con el conjunto de datos analizado, ya que la información disponible se restringe al intervalo comprendido entre las 4:00 a.m. y las 12:00 p.m. del día 1 de octubre de 2025.

5. Framework de AutoML

Tradicionalmente, el ciclo de vida de un proyecto de Machine Learning (ML) involucra fases iterativas como:

- **Preprocesamiento y Feature Engineering:** transformación y creación de variables para capturar patrones relevantes.
- **Selección del Modelo:** elección del algoritmo o familia de algoritmos más apropiada.
- **Entrenamiento y Validación:** ajuste de los parámetros del modelo y evaluación preliminar.
- **Afinamiento de Hiperparámetros:** optimización de los parámetros que gobiernan el proceso de aprendizaje.
- **Evaluación Final y Despliegue:** validación en conjuntos de prueba e implementación.

La falta de estandarización, derivada de la naturaleza diversa de los problemas abordados por el ML (Machine Learning), junto con el alto costo en *horas de trabajo especializado* y recursos de cómputo, ha generado históricamente ineficiencias significativas. Estas se manifiestan en procesos manuales propensos a errores humanos, una experimentación limitada por las restricciones de tiempo y soluciones con baja reproducibilidad que dificultan su auditoría.

5.1. El Paradigma AutoML

El avance de las técnicas englobadas bajo el término **AutoML** (Automated Machine Learning) ha permitido automatizar gran parte de estas tareas, que antes dependían capitalemente del criterio y la experiencia del especialista (*"ojo del experto"*).

Si bien actualmente no existe un framework universal que, de manera rápida y para el caso general, genere un prototipo de solución óptima para proyectos reales complejos (y teoremas como el *"No Free Lunch"* demuestran que tal herramienta omnipotente no puede existir), la implementación de soluciones AutoML *ad-hoc* persigue objetivos clave:

1. **Reducción del sesgo humano:** minimizar la subjetividad en la selección y configuración de modelos.
2. **Aumento de la productividad:** liberar al experto de tareas repetitivas para que se centre en el diseño del problema y la interpretación de resultados.

3. **Mejora de la exhaustividad:** explorar de forma sistemática y automática espacios de búsqueda (modelos, hiperparámetros) mucho más amplios.

En aras de estos objetivos y para abordar las necesidades específicas de este proyecto (en particular, el trabajo con **series de tiempo**), se optó por diseñar e implementar un *mini-framework* de AutoML especializado (ver Figura 7).

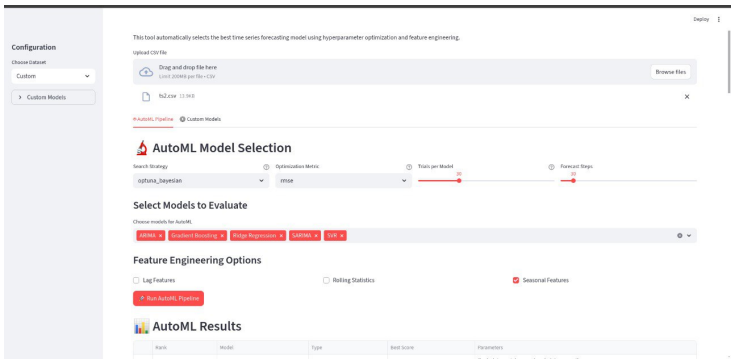

Figura 7: Framework para AutoML.

5.2. Arquitectura del Mini-Framework

El sistema propuesto se estructura en los siguientes componentes modulares:

- **Interfaz Visual de Configuración:** permite al usuario definir los parámetros de alto nivel del *pipeline* AutoML (no los hiperparámetros internos de cada modelo). Desde aquí se seleccionan:
  - Conjuntos de características (*features*) a evaluar.
  - Métricas de evaluación primarias y secundarias (e.g., RMSE, MAE, MAPE).
  - Restricciones de tiempo y recursos computacionales.
- **Adaptadores de Modelo (*Model Adapters*):** capa de abstracción que unifica la interfaz de entrada/salida para diversos tipos de modelos (estadísticos como ARIMA, basados en árboles, redes neuronales, etc.), facilitando su integración en el *pipeline*.

- **Motor de Búsqueda y Optimización:** componente central que implementa el algoritmo de búsqueda. Opera bajo la hipótesis simplificadora de que el *pipeline* óptimo está compuesto por un **único modelo** (es decir, no explora automáticamente *ensambles* o *stacks* de modelos). Realiza una **búsqueda exhaustiva** (o semi-exhaustiva, según espacios muy grandes) sobre el espacio definido de modelos e hiperparámetros, guiada por la métrica objetivo especificada por el usuario.

- **Módulo de Visualización y Comparación:** genera reportes y visualizaciones unificadas (pronóstico vs. real, comparación de métricas entre modelos) que facilitan el análisis comparativo y la toma de decisiones final por parte del experto.

Esta arquitectura permite **eliminar de forma automática variantes de modelo poco efectivas** en las etapas tempranas, filtrando solo las configuraciones más prometedoras para una evaluación detallada. Esto resulta en un **uso más eficiente del tiempo del experto** y de los recursos computacionales.

5.3. Resultados y Análisis

La aplicación del framework al dominio específico de las series de tiempo del proyecto arrojó resultados consistentes con la intuición teórica:

- Los modelos más exitosos fueron, en su mayoría, **modelos estadísticos clásicos** que imponen un sesgo (*inductive bias*) fuerte y apropiado sobre la estructura temporal de los datos (estacionalidad, tendencia, autocorrelación). Estos modelos, al estar bien especificados para el problema, requieren relativamente pocos datos para un ajuste robusto.

- El proceso AutoML permitió descartar rápidamente enfoques más complejos (como ciertas redes neuronales) que, sin un ajuste muy especializado y mayor volumen de datos, no superaban la robustez y simplicidad de los modelos estadísticos.

Además se hizo un pequeño estudio extra para validar que el *feature engineering* no cause ruido en predicciones de largas distancias. Se utilizó otro conjunto de datos para este análisis y se obtuvieron buenos resultados (ver Figura 8). Ningún algoritmo tuvo un comportamiento anómalo con respecto a los datos esperados.

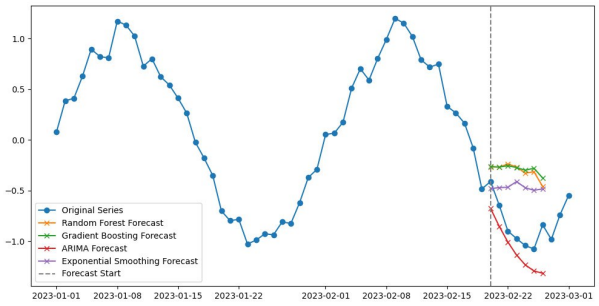

Figura 8: Predicción usando feature engineering.

6. Elección de modelos para predicción

Utilizando los módulos del *framework* se hicieron predicciones de cada uno de los servicios del estudio. Se tuvieron en cuenta los algoritmos: Random Forest, Gradient Boosting, Regresión Lineal y Ridge Regression. La métrica utilizada fue la raíz del error cuadrático medio (mientras más pequeña mejor). Para todos los servicios se obtuvo que el mejor algoritmo fue Random Forest, aunque hubo pequeñas diferencias en el orden de eficiencia según el servicio.

6.1. Datos

En el conjunto de prueba se obtuvo 500 194 342.53 de RMSE que puede parecer elevado, pero en este caso, el modelo logra capturar adecuadamente la tendencia general del consumo, como se evidencia en la superposición visual entre la serie real y la predicha.

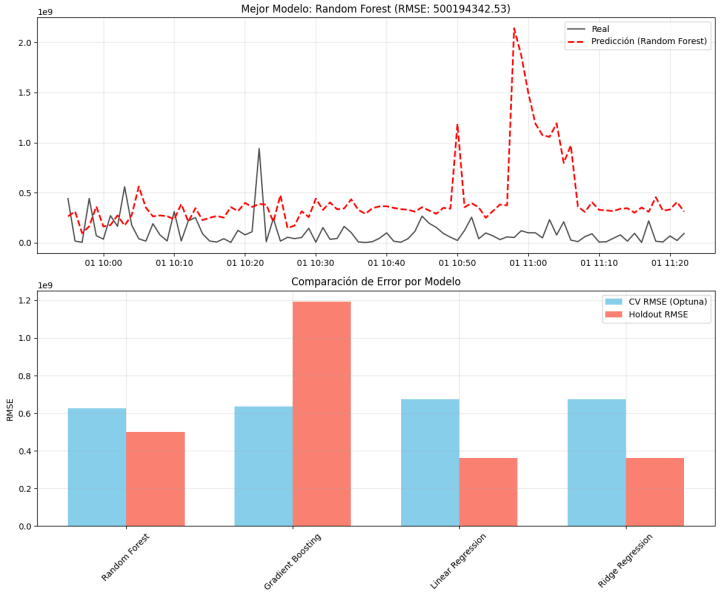

Figura 9: Proyección de los datos de prueba sobre consumo de datos.

Sin embargo, también se observan momentos de subestimación y sobre-estimación, lo que sugiere que aún existe margen de mejora en la precisión del modelo, especialmente en los picos de consumo (ver Figura 9).

6.2. Voz

En la Figura 10 se observa que se obtuvo 42.08 de RMSE lo cual indica que el modelo logra una predicción bastante precisa.

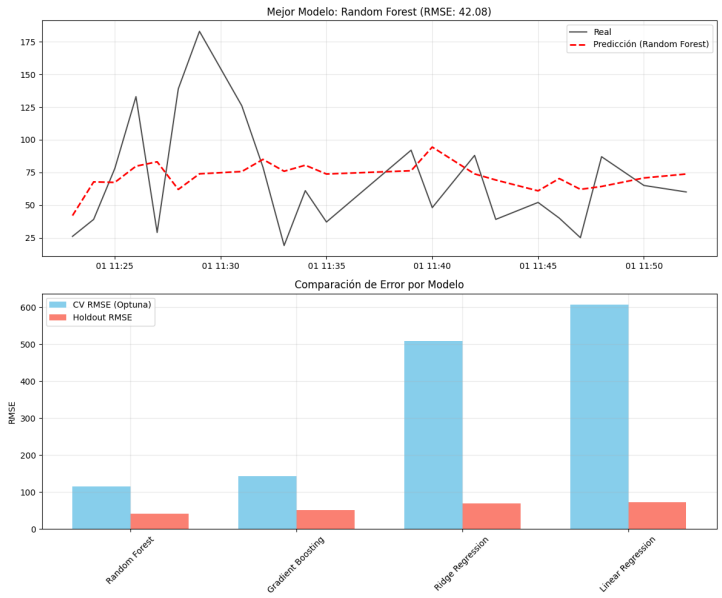

Figura 10: Proyección de los datos de prueba sobre consumo de voz.

La superposición entre la serie real y la predicha muestra una cierta concordancia, lo que no logra capturar las fluctuaciones notables en el transcurso del tiempo.

6.3. SMS

Visualmente, la línea roja discontinua (predicción) sigue de cerca la trayectoria de la línea negra (real), lo que sugiere que el modelo logra capturar con precisión la dinámica del fenómeno observado. No se aprecian desviaciones sistemáticas ni errores persistentes, lo que refuerza la interpretación de que el modelo tiene buen ajuste en este intervalo temporal (ver Figura 11).

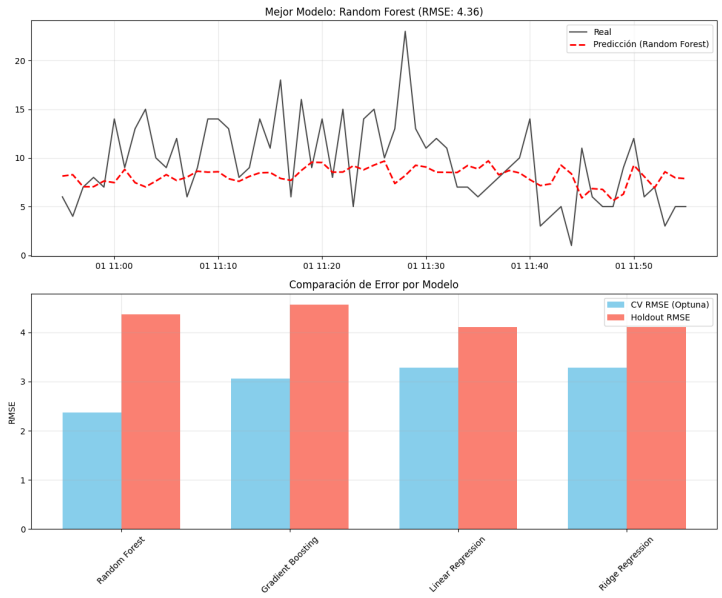

Figura 11: Proyección de los datos de prueba sobre consumo de sms.

Mientras que el RMSE de 4.36 es bajo en términos absolutos, lo que implica que el modelo tiene un margen de error reducido y puede ser considerado confiable para tareas de predicción en este contexto.

6.4. Predicción de consumo

Como resultado de todo el análisis realizado se pudieron generar predicciones de todos los servicios en un espacio de tiempo de 30 minutos a partir de que finaliza el tiempo registrado en el conjunto de datos inicial (ver Figura 12, Figura 13 y Figura 14 respectivamemte).

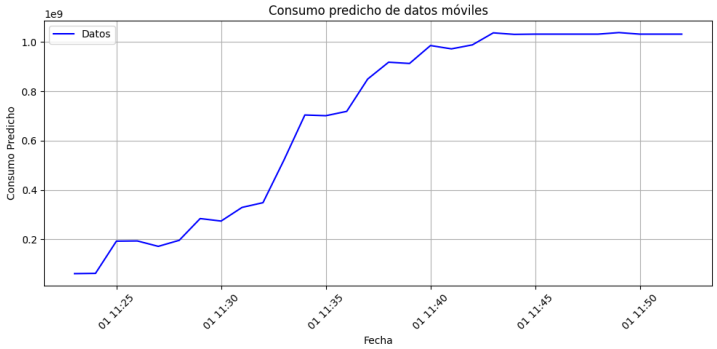

Figura 12: Predicción sobre consumo de datos. Se estima que se comporte creciente en el tiempo.

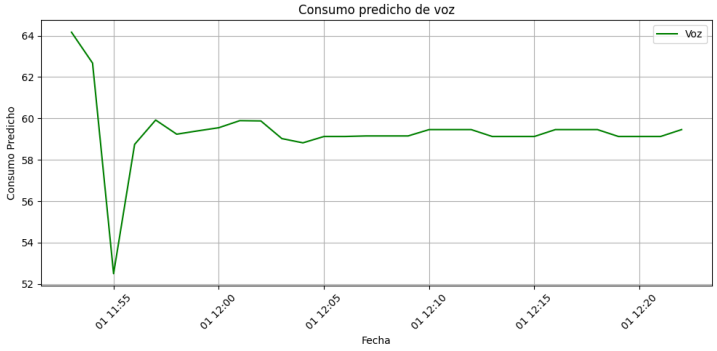

Figura 13: Predicción sobre consumo de voz. Se espera un brusco descenso pero luego se mantiene constante en el tiempo.

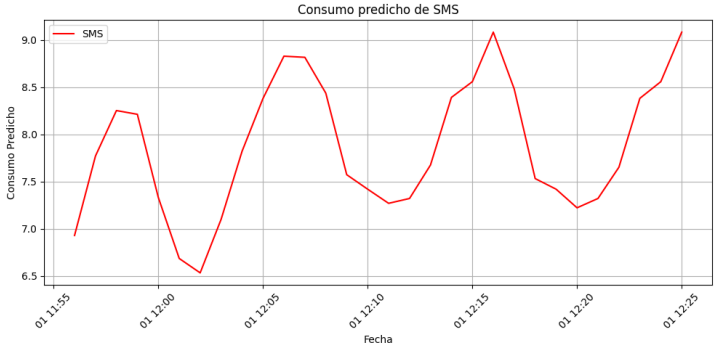

Figura 14: Predicción sobre consumo de sms. Se proyectan variaciones sostenidas a lo largo del tiempo

7. Conclusiones

La implementación de un mini-framework de AutoML especializado demostró ser una herramienta valiosa dentro del proyecto. No solo **agilizó el ciclo de experimentación**, sino que también **proporcionó una base objetiva y reproducible** para la selección del modelo final.

La experiencia reafirma que, si bien el AutoML no reemplaza al experto, lo potencia al permitirle centrarse en la formulación del problema, el diseño de características de alto nivel y la interpretación de resultados, mientras automatiza la búsqueda extensiva y sistemática en el espacio de soluciones técnicas.

Glosario

5G / 6G Quinta y futura sexta generación de redes móviles. 3

ARIMA Extensión de ARMA que incorpora diferenciación para manejar series no estacionarias. 2

ARMA Modelo estadístico para series temporales que combina componentes autorregresivos y de media móvil. 2

CNN (Convolutional Neural Network) Red neuronal especializada en la extracción de patrones espaciales. 2

CNN 2D CNN aplicada a representaciones matriciales espacio-temporales del tráfico. 2

DBSCAN Algoritmo de clustering basado en densidad capaz de detectar ruido y anomalías. 2

DL (Deep Learning) Subcampo de Machine Learning basado en redes neuronales profundas con múltiples capas ocultas. 2

DNN (Deep Neural Network) Red neuronal profunda con varias capas ocultas. 2

FL (Federated Learning) Paradigma de aprendizaje distribuido que preserva la privacidad de los datos. 2

GNN (Graph Neural Network) Red neuronal diseñada para datos estructurados como grafos. 2, 3

Holt-Winters Método de suavizado exponencial que modela nivel, tendencia y estacionalidad. 2

HTM (Hierarchical Temporal Memory) Modelo inspirado en la neocorteza para procesamiento temporal, no basado en deep learning. 2

K-means Algoritmo de clustering no supervisado basado en centroides. 2

K-NN (K-Nearest Neighbors) Algoritmo de aprendizaje basado en instancias que utiliza vecinos más cercanos. 2

LSTM (Long Short-Term Memory) Tipo de RNN capaz de capturar dependencias de largo plazo en series temporales. 2, 3

LTE (Long Term Evolution) Estándar de comunicaciones móviles de cuarta generación (4G). 2

ML (Machine Learning) Conjunto de métodos que permiten a un sistema aprender patrones a partir de datos sin programación explícita. 2

ML clásico Algoritmos de Machine Learning no profundos, como k-NN, SVM, Árboles de decisión y modelos estadísticos. 3

MLP (Multi-Layer Perceptron) Red neuronal feed-forward clásica compuesta por múltiples capas totalmente conectadas. 2

OS-ELM Variante de ELM para aprendizaje secuencial u online. 2

PCA (Principal Component Analysis) Técnica de reducción de dimensionalidad basada en varianza. 2

Random Forest Método de ensamble basado en múltiples árboles de decisión. 2

RCLSTM LSTM de complejidad reducida o dispersa que mantiene rendimiento con menos parámetros. 2

RNN (Recurrent Neural Network) Red neuronal diseñada para procesar datos secuenciales. 2

SARIMA Modelo ARIMA que incluye explícitamente componentes estacionales. 2

SVM (Support Vector Machine) Algoritmo de clasificación y regresión basado en maximización del margen. 2

SVR (Support Vector Regression) Versión de SVM para tareas de regresión. 2

TCN (Temporal Convolutional Network) Red convolucional causal diseñada para modelado temporal. 2

Transformer Arquitectura basada en mecanismos de atención que captura dependencias globales de largo alcance. 2, 3

Tráfico espacio-temporal Modelado del tráfico considerando dimensiones espaciales y temporales. 2, 3

XGBoost Algoritmo de gradient boosting eficiente y altamente competitivo. 2

Referencias

[1] Autor, Título del libro/artículo, Editorial, Año.

[2] Autor, Título del paper, Revista, Volumen, Año.