

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Modelo de Predicción de Ventas

Julio García, 22076
Sofía García, 22210
Diego García, 22404

Javier Fong
Deep Learning y Sistemas Inteligentes
Guatemala, 2025

Introducción

La predicción fiable de ventas es un elemento central para la planificación de inventario, la asignación de recursos y la toma de decisiones estratégicas en empresas de comercio y servicios. Sin embargo, la demanda suele estar afectada por estacionalidad, promociones, días festivos y eventos externos. Esto vuelve difícil anticipar con precisión el comportamiento del mercado solo con criterios intuitivos o reglas simples.

En los últimos años, se han creado varios modelos de aprendizaje automático para abordar este problema. No obstante, en la práctica empresarial existen dos brechas importantes. Por un lado, muchos modelos de alto desempeño sacrifican interpretabilidad. Por otro, las soluciones experimentales rara vez se integran en herramientas accesibles para usuarios no técnicos, lo que limita su uso.

En el presente trabajo se describe el diseño e implementación de un sistema orientado a la predicción de ventas a distintos horizontes temporales mediante una arquitectura híbrida. El sistema combina un modelo estructural explicable, un modelo de ensamble basado en gradiente boosting y una red LSTM cuantil, integrados en un servicio web. El objetivo es ofrecer una solución que equilibre precisión, robustez e interpretabilidad, así como un uso sencillo y universal.

Metodología

La metodología seguida se estructura en varias etapas que abarcan desde la preparación de los datos hasta la exposición de las predicciones en una aplicación web.

En primer lugar, la arquitectura general se organiza en dos capas, un backend y un frontend. La primera, implementada en Python, concentra toda la lógica de datos y modelado, por lo que se encarga de la lectura y limpieza de los datos, ingeniería de características, entrenamiento de los modelos, evaluación y servicio de predicciones mediante una API basada en FastAPI. La segunda está desarrollada con Next.js y React. Esta ofrece una interfaz web donde los usuarios pueden cargar sus datos históricos, configurar los meses de predicción que desean y visualizar resultados y métricas sin necesidad de interactuar directamente con el código.

En la etapa de preparación de datos, el sistema acepta diferentes formatos de archivos de ventas. Un módulo de ingestión identifica de manera flexible las columnas de fecha y monto de venta, filtra únicamente las operaciones que ya fueron pagadas y normaliza los nombres de columnas a un esquema interno simple. Posteriormente, se agregan las transacciones a nivel diario y se convierten los datos en una serie de tiempo continua. Sobre esta serie se aplican técnicas de preprocesamiento como interpolación de huecos, imputación basada en medianas estacionales y recorte de valores extremos. Opcionalmente se trabaja en el espacio logarítmico de las ventas para estabilizar la varianza y reducir el impacto de picos aislados.

Después de eso se realiza la ingeniería de características. A partir del índice temporal se generan variables de calendario (día de la semana, fin de semana, mes y trimestre), términos de Fourier para representar de forma suave las estacionalidades semanal y anual, lags de la serie y medidas rodantes de tendencia y volatilidad. Además, se incorporan variables binarias asociadas a períodos relevantes del negocio, cómo inicio y fin de mes o proximidad a días festivos y eventos comerciales. Este conjunto de características resume tanto el comportamiento histórico como el contexto de cada observación y sirve de base para los modelos de aprendizaje supervisado.

Sobre estos datos enriquecidos se emplean tres tipos de modelos de pronóstico. En primer lugar se utiliza un modelo Prophet como componente estructural. Prophet permite modelar de forma explicable la tendencia a largo plazo, las estacionalidades de distintas frecuencias y el efecto de eventos especiales como Black Friday o Cyber Monday. Sus hiperparámetros de cambio de tendencia y fuerza de estacionalidad se ajustan mediante validación cruzada temporal, de modo que el modelo de base capture la mayor parte de la estructura sistemática de la serie.

En segundo lugar, se entrena un modelo de gradiente boosting, XGBoost, que no predice directamente las ventas, sino el residuo que deja Prophet al comparar sus pronósticos con los datos históricos. Este modelo recibe como entradas tanto los componentes del propio Prophet como lags y estadísticas rodantes adicionales. Su objetivo es aprender relaciones no lineales más sutiles y corregir sesgos sistemáticos del modelo estructural. La combinación de ambos se interpreta como un ensamble en dos capas: Prophet proporciona el esqueleto explicable del pronóstico y XGBoost actúa como una corrección flexible sobre dicho esqueleto.

En tercer lugar, se entrena una red neuronal LSTM cuantil orientada a capturar patrones secuenciales complejos. El modelo recibe ventanas deslizantes de 60 días, cada una enriquecida con las mismas variables calendario, eventos y estadísticas rodantes, y aprende a predecir el logaritmo de las ventas del día siguiente. En lugar de minimizar el error cuadrático medio, se utiliza una función de pérdida de cuantiles, lo que hace al modelo más robusto frente a outliers y facilita la extensión a intervalos de predicción en el futuro. El entrenamiento se realiza con división temporal entreno-validación y detención temprana para evitar sobreajuste.

Una vez entrenados los modelos, sus artefactos se almacenan en disco (pesos de Prophet, XGBoost, LSTM y el escalador utilizado para normalizar las características). La evaluación se lleva a cabo sobre una ventana de validación situada al final de la serie, empleando métricas estandar en pronóstico de demanda: MAE, RMASE, MAPE, sMAPE y coeficiente de determinación R². Esto permite comparar de forma justa el desempeño del modelo estructural, de la red LSTM y del ensamble híbrido.

Para su uso operativo, se expone un endpoint de API denominado /predict-horizon. A partir de un archivo CSV proporcionado por el usuario, el backend reproduce el mismo pipeline de limpieza y generación de características. Seguido de eso, genera pronósticos multi-paso hacia adelante. Para cada día del horizonte solicitado, la red LSTM produce una predicción a partir de las últimas observaciones disponibles, mientras que Prophet aporta su estimación estructural y XGBoost

calcula una corrección del residuo, creando el ensamble. Se actualiza el historial interno con las predicciones para simular la llegada de nuevos datos. El resultado se devuelve como una lista de fechas futuras con dos valores por día: la predicción del LSTM cuantil y la del ensamble.

Finalmente, la interfaz web organiza este flujo en tres vistas principales. La página de carga permite seleccionar el tiempo, en meses, de la predicción y subir el archivo de ventas, mostrando instrucciones y un ejemplo del formato. La página de resultados presenta las predicciones en una gráfica de líneas que compara el comportamiento del LSTM cuantil y del ensamble a lo largo del horizonte, además, de una tabla numérica detallada. La página de métricas resume el desempeño histórico de Prophet, del LSTM y del ensamble mediante tablas, gráficos de barras e interpretaciones textuales que traducen las métricas a un lenguaje accesible para perfiles de negocio.

Resultados

Los experimentos se llevaron a cabo utilizando los datos históricos de ventas obtenidos de la base de datos de una organización anónima y se buscó que estos consideraran distintos horizontes de pronósticos. En todos los casos se utilizó la misma ventana de validación temporal situada al final de la serie para garantizar la comparabilidad entre modelos.

El modelo Prophet se evaluó, al principio, de forma individual. Luego el modelo LSTM cuantil, fue entrenado sobre el mismo horizonte que Prophet. Por último, se evaluó el modelo de ensamble, el cuál demostró ser el más capaz y potente.

Métrica	Prophet	Quantile LSTM	Ensemble
MAE	899.18	0.13	245.78
RMSE	1198.29	0.24	344.22
MAPE	126.40%	17.60%	48.52%
sMAPE	46.60%	13.48%	20.43%
R ²	0.15	0.342	0.93

Imagen 1: Tabla de comparación de métricas entre modelo Prophet, LSTM cuantil y Ensamble

Finalmente, el modelo final fue puesto a prueba con un conjunto de datos completamente aparte en horizontes de 1, 6, 12 y 24 meses.



Imagen 2: Gráfica de la predicción de ventas del dataset de prueba para 1 mes

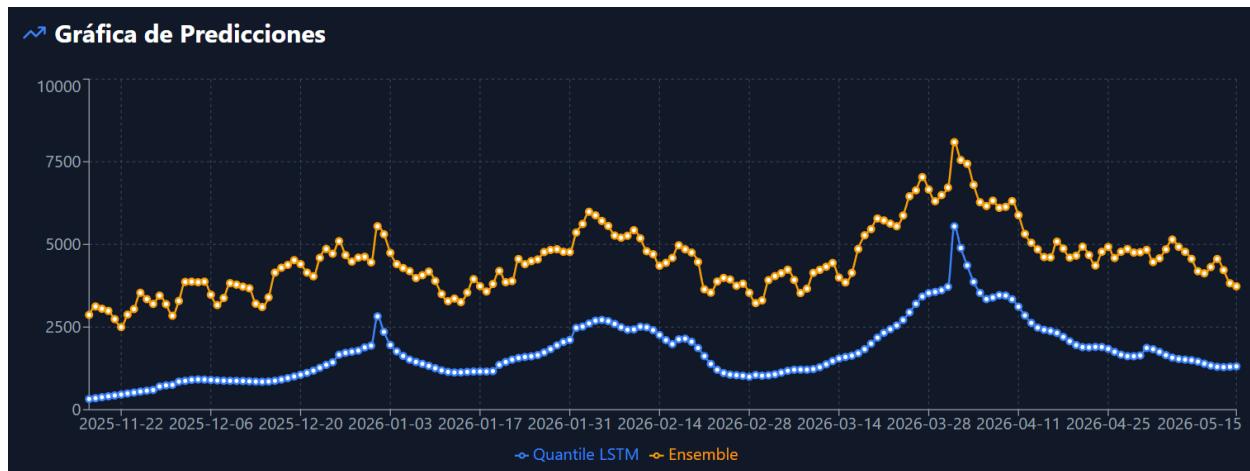


Imagen 3: Gráfica de la predicción de ventas del dataset de prueba para 6 meses



Imagen 4: Gráfica de la predicción de ventas del dataset de prueba para 12 meses

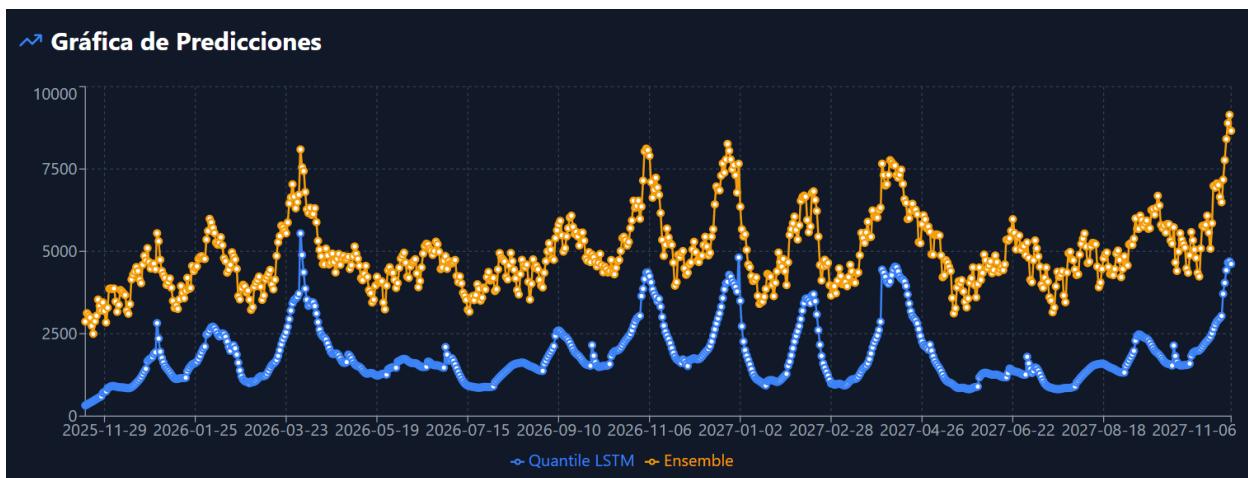


Imagen 5: Gráfica de la predicción de ventas del dataset de prueba para 24 meses

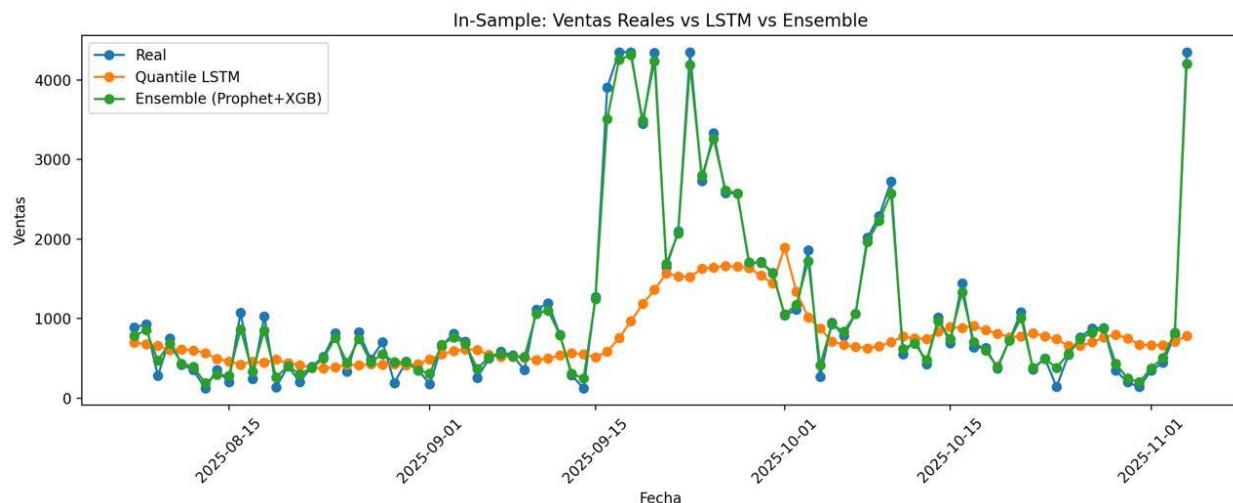


Imagen 6: Gráfica de predicción de real vs Quantile LSTM y Ensemble

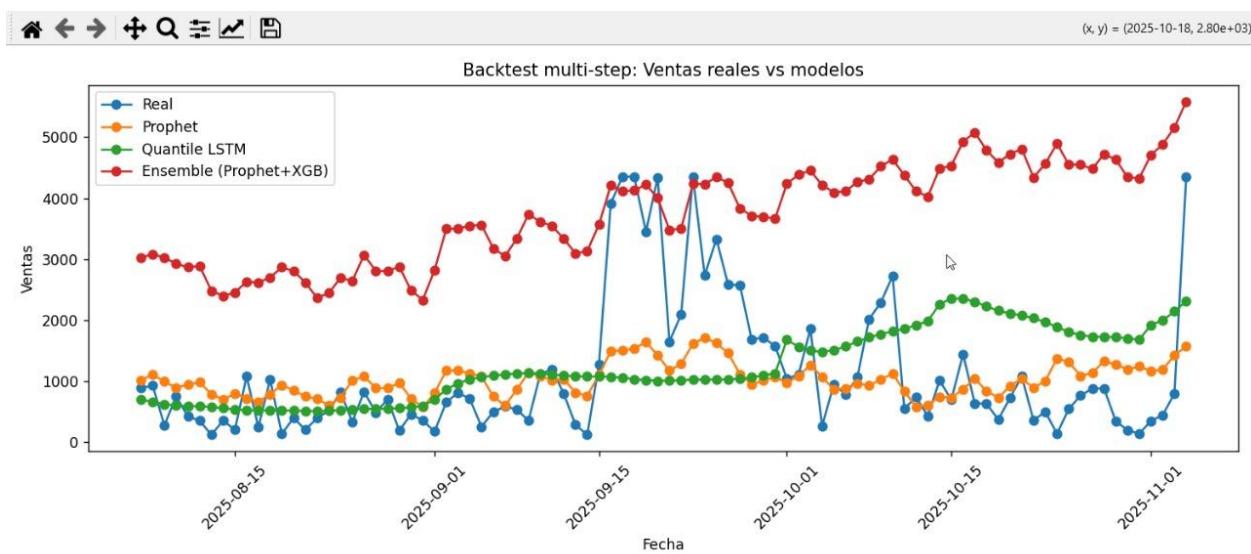


Imagen 7: Gráfica de los resultados de test de los modelos contra los datos reales

Discusión

Los resultados obtenidos ponen de manifiesto varias ideas relevantes. En primer lugar, confirman que ningún modelo aislado es suficiente para capturar toda la complejidad del problema de predicción de ventas. Prophet aporta una representación clara y explicable de tendencia, estacionalidad y eventos especiales, pero tiende a subestimar ciertas no linealidades y cambios de régimen. La red LSTM cuantil, por su parte, se adapta mejor a patrones locales y a interacciones complejas entre variables de calendario, pero resulta más difícil de interpretar y puede mostrar mayor variabilidad fuera del rango de datos vistos en el entrenamiento.

La arquitectura de ensamble aprovecha las fortalezas de ambos enfoques. Al utilizar Prophet como columna vertebral y delegar en XGBoost la tarea de aprender la estructura residual, se obtiene un modelo que mantiene interpretabilidad a nivel de componentes y, al mismo tiempo, corrige de manera flexible los errores sistemáticos. El hecho de que este ensamble alcance mejoras significativas en todas las métricas sugiere que una parte importante de la información relevante para el pronóstico reside, precisamente, en esos residuos, y que un modelo de árboles de decisión es adecuado para explotarla.

Desde el punto de vista de la aplicación, otro aspecto destacable, es la importancia del pipeline de datos y de la interfaz de usuario. La capacidad de aceptar distintos formatos de CSV, detectar automáticamente las columnas relevantes y limpiar la serie de manera robusta reduce de forma notable la probabilidad de cometer errores en la fase de carga. Asimismo, la separación entre backend y frontend permite que usuarios no técnicos trabajen con la herramienta mediante una interfaz guiada, en la que el proceso se entiende como una secuencia simple de pasos: subir datos, revisar gráficas, analizar métricas y tomar decisiones.

No obstante, el sistema presenta limitaciones. Los pronósticos multi-paso se basan en la realimentación de predicciones pasadas, lo que puede acumular error en horizontes muy largos, especialmente en contextos de cambios estructurales fuertes. Además, el modelo se ha evaluado sobre una única serie agregada de ventas; para entornos con cientos de productos o tiendas sería conveniente estudiar extensiones jerárquicas o modelos globales que exploten mejor la información compartida. Finalmente, aunque la evaluación actual utiliza métricas puntuales, resultaría valioso incorporar medidas probabilísticas que reflejen la incertidumbre de las predicciones.

En la imagen 6 se observa la comparación in-sample entre las ventas reales, el modelo Quantile LSTM y el ensemble Prophet+XGBoost. La curva naranja del LSTM sigue bastante bien la forma general de la serie (niveles bajos al inicio, meseta alrededor de mediados de septiembre y un ligero repunte en octubre), pero se ve mucho más suave y tiende a “aplastar” los picos: cuando las ventas reales se disparan a valores cercanos a 4000, el LSTM se queda alrededor de 1500-1900. Esto es consistente con un modelo entrenado con pérdida cuantílica o tipo MAE, que aprende algo cercano a un cuantil/mediana condicional y penaliza menos los errores grandes, por

lo que prefiere soluciones más conservadoras y no replica del todo eventos extremos o muy esporádicos.

En contraste, la curva verde del ensemble Prophet+XGBoost casi se superpone a la serie real, capturando tanto los cambios de nivel como los picos abruptos y los valles (por ejemplo, el “bloque” de valores muy altos a mediados de septiembre y el gran pico de inicios de noviembre). Esto sugiere que Prophet está modelando bien la tendencia y la estacionalidad, mientras que XGBoost se ajusta con mucha flexibilidad a los residuales, permitiéndole aprender patrones específicos día a día; el resultado es un fit casi perfecto en el periodo de entrenamiento, aunque también implica que el ensemble corre más riesgo de estar capturando ruido o efectos muy particulares de ese histórico que podrían no generalizar tan bien fuera de muestra.

En la imagen 7 se observa el backtest multi-test. En este se ve una serie real altamente volátil. Durante la primera parte del periodo, las ventas reales, en color azul, se mantienen bajas y con oscilaciones frecuentes; sin embargo, alrededor de mediado de septiembre hay un cambio abrupto en las ventas, subiendo bastante, pero seguido de caídas y repuntes irregulares. Prophet, de color naranja, se mantiene en un rango relativamente bajo y suave, capturando cierta tendencia general, pero sin reflejar el salto de nivel. En la zona del quiebre su respuesta es lenta y se queda subestimando casi todo el tramo que le sigue. Esto sugiere la limitación esperada de un modelo estructural, describiendo bien patrones promedio, pero perdiendo precisión cuando hay cambios no lineales.

Por su parte, el LSTM cuantil, de color verde, muestra un comportamiento más adaptivo que el Prophet, elevándose gradualmente su nivel conforme avanza el periodo y reaccionando parcialmente al cambio. Sin embargo, lo hace de forma muy conservadora y amortiguada. No alcanza los picos reales del salto y permanece por debajo de los máximos, lo cual es coherente con el hecho de que el LSTM aprende patrones secuenciales locales y evita extrapolación agresivamente fuera del rango visto y que al usar pérdida cuantil, la red tiende a aproximar una predicción central que reduce la sensibilidad a outliers. Por lo tanto, este modelo ofrece estabilidad, pero sacrifica capacidad para capturar shocks extremos.

Por último el ensemble, de color rojo, es el único que internaliza el cambio de régimen, elevando su nivel tras el quiebre y manteniendo una trayectoria coherente con un nuevo estado de demanda. Sin embargo, este enfrenta una sobre estimación sistemática, quedando por encima de los valores reales en todo momento. Además, este modelo falla en capturar las caídas posteriores con la misma amplitud, lo que sugiere que la capa de corrección por residuos está aprendiendo una señal positiva muy fuerte que se va acumulando a lo largo de los pasos. Esto indica que con el pasar del tiempo, este modelo va acumulando sesgo.

Con esto, se puede decir que el Ensemble es el modelo con un mejor desempeño práctico, en comparación al Prophet solo y al LSTM, pues es el único que logra adaptarse al cambio abrupto en la tendencia. Sin embargo, este presenta bastante sobre estimación en casi todas las ventas, lo que puede llevar a una toma de decisiones incorrecta dentro de la empresa. Por lo tanto, es un

modelo que aún puede llegar a tener ciertos ajustes que reduzcan el sesgo que acumula a lo largo del tiempo.

Conclusiones

El trabajo presentado describe un sistema integral para la predicción de ventas que combina un pipeline de datos robusto, una arquitectura híbrida de modelos y una interfaz web orientada a usuarios de negocio. A partir de archivos de ventas en formato de CSV, el sistema automatiza la limpieza, la ingeniería de características y la generación de pronósticos, ofreciendo como resultado curvas de demanda futuras y métricas que facilitan su interpretación.

La evidencia empírica muestra que el modelo de ensamble basado en un componente estructural Prophet y una corrección XGBoost sobre los residuos supera de forma consistente tanto al Prophet aislado, como a la red LSTM cuantil, alcanzando reducciones importantes en el error medio y un coeficiente de determinación cercano a 0.93 en la ventana de validación. Sin embargo, este también acumula mucho sesgo a lo largo del tiempo que crea sobre estimación de las ventas, lo que puede llegar a tomar decisiones erróneas.

Como líneas futuras se plantea extender la arquitectura para manejar pronósticos jerárquicos por producto y tienda, incorporar de manera explícita intervalos de confianza mediante cuantiles múltiples y evaluar el sistema en contextos adicionales, como series de demanda altamente esporádicas. Asimismo, sería de interés integrar mecanismos de actualización periódica y monitorización automática del desempeño, de modo que la herramienta pueda adaptarse de forma continua a cambios en el comportamiento de la demanda.