

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Predicción de Estados Financieros

Trabajo de grado en modalidad de práctica profesional para optar al título de Estadístico

Nombre del Estudiante:
Jose Alfonso Victoria Floréz
Estudiante de Estadística

Docente Orientador:
Duván Humberto Cataño Salazar
Instituto de Matemáticas
Facultad de Ciencias Exactas y Naturales
Universidad de Antioquia

Medellín, 2025

Agradecimientos

A todo ser aquel que de una u otra forma me apoyó en algún momento de mi proceso académico. Gracias totales.

Índice

1. Resumen **3**

2. Introducción **3**

3. Objetivos **4**

3.1. Objetivo General 4

3.2. Objetivos Específicos 4

4. Marco Teórico **4**

4.1. Redes neuronales 5

4.2. Modelos de series de tiempo SARIMA 8

4.3. Neural Prophet 9

4.4. XGBoost 11

4.5. Cross Validation (Expanding Window Validation) 13

5. Herramientas utilizadas **14**

6. Implementación **15**

6.1. Primera fase 15

6.2. Segunda fase 15

6.3. Tercera fase 16

7. Resultados y Conclusiones **17**

7.1. Resultados 17

7.2. Recomendaciones 18

8. Referencias **19**

1. Resumen

El objetivo de este trabajo de grado es desarrollar y comparar modelos de predicción de series de tiempo aplicados a los estados financieros de una empresa, específicamente el Flujo de Caja Libre (FCF), el Capital de Trabajo Neto Estricto (SNWC) y el Estado de Pérdidas y Ganancias (P&L). Para lograrlo, se propone implementar modelos de predicción como Redes Neuronales LSTM, SARIMA, NeuralProphet y XGBoost, seleccionados por su capacidad para manejar datos temporales con características complejas.

La metodología se estructura en tres fases: la primera fase consiste en un proceso ETL (Extracción, Transformación y Carga) para la selección y preparación de datos históricos correspondientes a un periodo de 36 meses, descartando cuentas con patrones inconsistentes. En la segunda fase, se construyen, entrenan y validan los modelos predictivos. Finalmente, en la tercera fase, se realiza la selección de las mejores predicciones, análisis comparativo de los resultados, identificando fortalezas y limitaciones de cada modelo evaluando su rendimiento mediante métricas como el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el Coeficiente de Determinación (R^2), para realizar el posterior tablero de reporte para comunicar los resultados obtenidos.

Este estudio contribuye a la gestión financiera al proporcionar herramientas predictivas robustas que permiten anticipar escenarios económicos futuros, optimizar la toma de decisiones estratégicas y garantizar un manejo eficiente de los recursos. Los hallazgos buscan demostrar la eficacia de los modelos aplicados en la predicción de estados financieros, lo que resulta clave para el crecimiento y sostenibilidad de la organización.

2. Introducción

Para la correcta administración de una empresa, es crucial contar con una buena gestión de sus Estados Financieros, ya que estos son la carta de presentación a nivel corporativo que refleja la salud financiera de la organización. Los Estados Financieros desempeñan un papel fundamental tanto en el ámbito interno, para facilitar la toma de decisiones estratégicas, la planificación y el crecimiento, como en el ámbito externo, al proporcionar información transparente y confiable a inversores, auditores y otras partes interesadas.

A través del análisis de los Estados Financieros, es posible evaluar el desempeño de la empresa en términos de rentabilidad, liquidez y eficiencia operativa. Permiten identificar áreas de mejora, prever posibles riesgos financieros y tomar decisiones informadas que contribuyan al desarrollo sostenible de la organización.

Un seguimiento adecuado y sistemático de los Estados Financieros es clave para detectar tendencias, medir resultados y garantizar un manejo eficiente de los recursos. En este contexto, contar con predicciones financieras resulta particularmente útil, ya que estas herramientas permiten anticipar escenarios futuros y preparar a la empresa para afrontar desafíos económicos, aprovechar oportunidades de inversión y optimizar sus estrategias de crecimiento.

Las proyecciones financieras, respaldadas por un análisis riguroso, facilitan la identificación de patrones y comportamientos económicos, lo cual es esencial para implementar acciones preventivas y correctivas que aseguren la sostenibilidad y el éxito a largo plazo de la empresa. De este modo, la gestión financiera no solo se limita a registrar y evaluar el pasado, sino que también se convierte en una guía para proyectar el futuro con mayor certeza y eficiencia.

3. Objetivos

3.1 Objetivo General

El objetivo general es proponer, desarrollar y comparar diferentes modelos de predicción de series de tiempo, incluyendo modelos de Redes Neuronales LSTM, SARIMA, Neural Prophet y XGBoost para predecir las cuentas de los estados financieros de la empresa, para el seccional de Costa Rica; tales estados financieros son: Flujo de Caja Libre (FCF), Capital de Trabajo Neto Estricto (SNWC) y el Estado de Pérdidas y Ganancias (PL), por sus siglas en inglés. Este modelo se tomará como piloto para el desarrollo en otros 7 países de Latinoamérica.

3.2 Objetivos Específicos

- Comprender la necesidad del departamento de finanzas, recolección y preparación de datos históricos, modelado y despliegue. La necesidad corresponde a predecir estos estados financieros claves para la toma de decisiones dentro de la empresa. La recolección y preparación de datos históricos corresponde a adquirir y preparar las cuentas que conforman los estados financieros, abarcando un periodo de 36 meses comprendido entre febrero de 2022 y septiembre de 2024. Posteriormente, se procederá al modelado y despliegue de los modelos para su análisis, selección del mejor modelo y generación de pronósticos.
- Diseñar y configurar (relaizar la parametrización) de los distintos modelos predictivos, ajustando sus parámetros para optimizar su rendimiento según las características de los datos. Asimismo, se realizará el entrenamiento y la validación de cada modelo utilizando métricas estadísticas adecuadas, como el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el Coeficiente de Determinación (R^2). Finalmente, se efectuará un análisis comparativo de los resultados obtenidos para determinar la eficacia de cada modelo de predicción de los estados financieros seleccionados, identificando las fortalezas y limitaciones de cada modelo.

4. Marco Teórico

A continuación abordaremos los conceptos y herramientas fundamentales que sustentan el desarrollo de la metodología para las predicciones financieras. Se analizarán técnicas

de modelado de series de tiempo, el uso de algoritmos avanzados como LSTM, SARIMA, NeuralProphet y XGBoost, así como métricas de evaluación y procesos de optimización. Este marco proporciona la base teórica indispensable para comprender y aplicar las metodologías utilizadas en el proyecto.

4.1 Redes neuronales

Las redes neuronales son sistemas computacionales inspirados en la estructura y funcionalidad del cerebro humano. Constan de capas interconectadas de nodos (neuronas) que procesan la información, asignando pesos y aplicando funciones de activación. Esto les permite modelar relaciones complejas y no lineales, convirtiéndolas en potentes herramientas para la resolución de problemas en diversos ámbitos.

Cada neurona de una red neuronal realiza cálculos al computar una suma ponderada de sus entradas, añadir un término de sesgo y aplicar una función de activación como la función tangente hiperbólica o la función sigmoide. Este proceso se expresa como:

$$z = \sum_{i=1}^{n} (w_i x_i) + b, \quad a = f(z),$$

donde:

- w_i: Peso asociado a la entrada x_i.
- x_i: Valor de la i-ésima entrada.
- b: Término de sesgo.
- $f(z)$: Función de activación.

Existen diversos tipos de redes neuronales, entre los que destacan las Redes Neuronales Feedforward (FNN), Redes Neuronales Recurrentes (RNN) y Redes de Memoria a Corto y Largo Plazo (LSTM); siendo esta última la seleccionada para el proyecto.

4.1.1 Redes Neuronales Feedforward (FNN)

Las Redes Neuronales Feedforward son el tipo más básico de arquitectura de red neuronal. Los datos fluyen en una sola dirección: desde la capa de entrada, pasando por una o más capas ocultas, hasta la capa de salida. Este diseño permite realizar tareas como clasificación y regresión, pero su incapacidad para manejar datos secuenciales o temporales limita sus aplicaciones en contextos como series de tiempo.

4.1.2 Redes Neuronales Recurrentes (RNN)

Las Redes Neuronales Recurrentes están diseñadas para procesar datos secuenciales, como texto o series temporales. Introducen conexiones recurrentes que permiten a la red recordar información previa al calcular la salida actual. Esto se expresa matemáticamente como:

$$h_t = f(W_x x_t + W_h h_{t-1} + b),$$

donde:

- h_t: Estado oculto en el tiempo t.
- W_x: Matriz de pesos para las entradas x_t.
- W_h: Matriz de pesos recurrentes aplicada al estado oculto anterior h_{t-1}.
- b: Vector de sesgo.
- f: Función de activación (e.g., tanh o ReLU).

Sin embargo, las RNN tradicionales presentan problemas al manejar dependencias a largo plazo debido al fenómeno del desvanecimiento o explosión de gradientes durante el entrenamiento.

4.1.3 Redes de Memoria a Corto y Largo Plazo (LSTM)

Las Redes de Memoria a Corto y Largo Plazo (LSTM) superan las limitaciones de las RNN tradicionales al introducir un mecanismo de compuertas que controla de manera selectiva qué información se retiene, actualiza o descarta. Estas compuertas son:

- **Compuerta de entrada:** Controla qué nueva información se almacena en la célula.
- **Compuerta de olvido:** Decide qué información previa debe eliminarse.
- **Compuerta de salida:** Determina qué información de la célula se utiliza para calcular la salida.

La célula de memoria de una LSTM se define matemáticamente como:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f),$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C),$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t,$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o),$$

$$h_t = o_t \odot \tanh(C_t),$$

donde:

- f_t: Compuerta de olvido.
- i_t: Compuerta de entrada.
- $\tilde{C}_t$: Candidata para el nuevo estado celular.
- C_t: Estado celular actualizado.
- o_t: Compuerta de salida.
- h_t: Estado oculto actualizado.
- W_f, W_i, W_C, W_o: Matrices de pesos asociadas a las compuertas.
- b_f, b_i, b_C, b_o: Vectores de sesgo.
- σ: Función sigmoide.
- tanh: Función tangente hiperbólica.
- $\odot$: Producto elemento a elemento.

Las LSTM manejan tanto la memoria a corto como a largo plazo al procesar secuencias temporales, lo que las hace ideales para tareas como predicción de series de tiempo, traducción automática y análisis de sentimientos.

4.2 Modelos de series de tiempo SARIMA

El modelo Autorregresivo Integrado de Media Móvil Estacional(SARIMA, por sus siglas en inglés) es una extensión del modelo ARIMA que incorpora componentes estacionales, denotada por:

$$SARIMA(p, d, q) \times (P, D, Q)_s$$

donde:

- p: Orden de la parte autorregresiva no estacional (AR).
- d: Grado de diferenciación no estacional.
- q: Orden de la parte de media móvil no estacional (MA).
- P: Orden de la parte autorregresiva estacional (SAR).
- D: Grado de diferenciación estacional.
- Q: Orden de la parte de media móvil estacional (SMA).
- s: Período estacional (cantidad de pasos de tiempo que representan la estacionalidad).

El modelo SARIMA se expresa matemáticamente como:

$$\Phi_P(B^s)\phi_p(B)\nabla^d\nabla_s^D X_t = \Theta_Q(B^s)\theta_q(B)\epsilon_t$$

donde:

- X_t: Serie temporal observada en el tiempo t.
- ∇^d: Operador de diferenciación no estacional, definido como $\nabla^d = (1 - B)^d$, donde B es el operador de rezago ($BX_t = X_{t-1}$).
- ∇_s^D: Operador de diferenciación estacional, definido como $\nabla_s^D = (1 - B^s)^D$.
- $\phi_p(B)$: Polinomio autorregresivo no estacional, definido como:

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p.$$

- $\theta_q(B)$: Polinomio de media móvil no estacional, definido como:

$$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q.$$

- $\Phi_P(B^s)$: Polinomio autorregresivo estacional, definido como:

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps}.$$

- $\Theta_Q(B^s)$: Polinomio de media móvil estacional, definido como:

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs}.$$

- ϵ_t: Término de error de ruido blanco en el tiempo t, con media 0 y varianza σ^2.

El modelo SARIMA combina tanto los componentes estacionales como los no estacionales. Las partes autorregresivas y de media móvil estacionales operan con un rezago igual al período estacional s, mientras que las operaciones de diferenciación aseguran que la serie sea estacionaria en escalas de tiempo tanto estacionales como no estacionales.

4.3 Neural Prophet

El Neural Prophet es un modelo de regresión modular y descomponible diseñado para pronosticar series temporales con una combinación de patrones estacionales, cambios en la tendencia y efectos de eventos especiales. El modelo es particularmente robusto para series temporales de negocios que presentan múltiples estacionalidades, tendencias no lineales y efectos de días festivos.

La serie temporal observada $y(t)$ se expresa como la suma de tres componentes principales:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{1}$$

- $g(t)$: **Componente de Tendencia**, modela la progresión a largo plazo y no periódica utilizando un modelo lineal por partes o de crecimiento logístico saturado.
- $s(t)$: **Componente de Estacionalidad**, captura las fluctuaciones periódicas usando series de Fourier.
- $h(t)$: **Componente de Festividades**, tiene en cuenta los efectos de festivos y eventos en fechas específicas.
- ϵ_t: **Término Residual**, asumido como ruido normalmente distribuido: $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$.

Componente de Tendencia $g(t)$: El componente de tendencia modela los cambios no periódicos en la serie temporal. Se implementan dos opciones:

1. **Tendencia Lineal por Partes:**

$$g(t) = (k + a(t)^T\delta)t + (m + a(t)^T\gamma), \tag{2}$$

donde k es la tasa de crecimiento base, δ son ajustes de tasa en los puntos de cambio, y γ asegura la continuidad.

2. **Crecimiento Logístico Saturado:**

$$g(t) = \frac{C(t)}{1 + \exp(-(k + a(t)^T\delta)(t - (m + a(t)^T\gamma)))}, \tag{3}$$

donde $C(t)$ representa la capacidad de carga, k es la tasa de crecimiento base, y los puntos de cambio se ajustan de manera similar.

Los puntos de cambio s_j, se seleccionan automáticamente o se especifican manualmente, donde ocurren ajustes en las tasas de crecimiento.

4.3.2. Componente de Estacionalidad $s(t)$: El componente de estacionalidad se modela como una serie de Fourier para aproximar funciones suaves y periódicas. Dado un periodo estacional P, la estacionalidad se expresa como:

$$s(t) = \sum_{n=1}^{N} \left[a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right], \tag{4}$$

donde N es el número de términos de Fourier, y a_n, b_n son coeficientes estimados durante el ajuste del modelo.

4.3.3. Componente de Festividades $h(t)$: Los efectos de los festivos se modelan como variables indicadoras para fechas de festivos especificadas D_i:

$$h(t) = \sum_{i=1}^{L} \kappa_i \mathbb{I}(t \in D_i), \tag{5}$$

donde κ_i es el tamaño del efecto del festivo i, y $\mathbb{I}(\cdot)$ es la función indicadora.

4.3.4. Componente Residual ϵ_t: Los errores residuales ϵ_t se asumen como ruido independiente y normalmente distribuido:

$$\epsilon_t \sim \mathcal{N}(0, \sigma^2). \tag{6}$$

4.4 XGBoost

XGBoost, por sus siglas en inglés para Extreme Gradient Boosting, es un potente algoritmo de aprendizaje automático que destaca en diversas tareas de modelización predictiva, incluida la previsión de series de tiempo. Se trata de un método de aprendizaje por conjuntos que combina las predicciones de varios modelos débiles (árboles de decisión) para crear un modelo predictivo sólido. El XGBoost es conocido por su escalabilidad, velocidad y capacidad para manejar relaciones complejas en los datos. El funcionamiento de XGBoost se basa en optimizar una función objetivo compuesta por dos elementos principales: una función de pérdida que mide la calidad del modelo y un término de regularización que controla la complejidad del modelo; este se expresa como:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega(f_k),$$

donde:

- $\ell(y_i, \hat{y}_i)$: Función de pérdida, que mide la diferencia entre el valor real (y_i) y la predicción ($\hat{y}_i$).
- $\Omega(f_k)$: Término de regularización que controla la complejidad del árbol f_k.
- n: Número de observaciones.
- K: Número de árboles en el modelo.

La regularización $\Omega(f_k)$ está diseñada para penalizar modelos complejos y puede expresarse como:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2,$$

donde:

- T: Número de hojas en el árbol.
- w: Vector de pesos de las hojas.
- γ: Penalización por cada hoja adicional.
- λ: Parámetro de regularización para evitar valores extremos en los pesos.

El algoritmo XGBoost utiliza el método de boosting por gradiente para minimizar la función objetivo. En cada iteración, agrega un nuevo árbol que corrige los errores de predicción del modelo actual. La actualización se basa en el gradiente de la función de pérdida respecto a las predicciones. El modelo actualizado se expresa como:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i),$$

donde:

- $\hat{y}_i^{(t)}$: Predicción en la t-ésima iteración.
- $\hat{y}_i^{(t-1)}$: Predicción en la iteración anterior.
- η: Tasa de aprendizaje, que controla cuánto contribuye el nuevo árbol.
- $f_t(x_i)$: Nuevo árbol construido en la iteración t.

  Dentro de las ventajas de XGBoost para la predicción de series de tiempo se encuentran:

  - Manejo de relaciones no lineales: Puede capturar relaciones no lineales complejas entre las características de entrada y la variable objetivo, lo que lo hace adecuado para datos de series temporales con patrones intrincados.
  - Proporciona información sobre la importancia de las distintas características, lo que permite a los analistas identificar los factores más influyentes en los datos de series temporales.
  - Regularización: Incorpora técnicas de regularización para evitar el sobreajuste y garantizar que el modelo se generalice bien a datos desconocidos.
  - Gestión de valores perdidos y valores atípicos: Puede gestionar valores perdidos y valores atípicos en los datos, reduciendo la necesidad de un preprocesamiento exhaustivo de los datos.

4.5 Cross Validation (Expanding Window Validation)

La validación cruzada con ventana expansiva(expanding window validation) es una técnica especial- mente diseñada para trabajar con datos de series de tiempo, donde el orden cronológico de las observaciones es fundamental. A diferencia de la validación cruzada tradicional, aquí no se permite mezclar los datos, ya que esto además de ir en contra del orden cronológico de los eventos que se tenga interés en predecir, podría causar una fuga de información del futuro al pasado, lo que llevaría a resultados poco realistas y sesgados. La idea principal es entrenar un modelo utilizando un conjunto inicial de datos y, progresivamente, añadir datos más recientes al conjunto de entrenamiento en cada iteración. Posteriormente, el modelo se evalúa en un intervalo de tiempo posterior, que actúa como el conjunto de prueba. Este proceso se repite varias veces, expandiendo la ventana de entrenamiento con cada iteración.

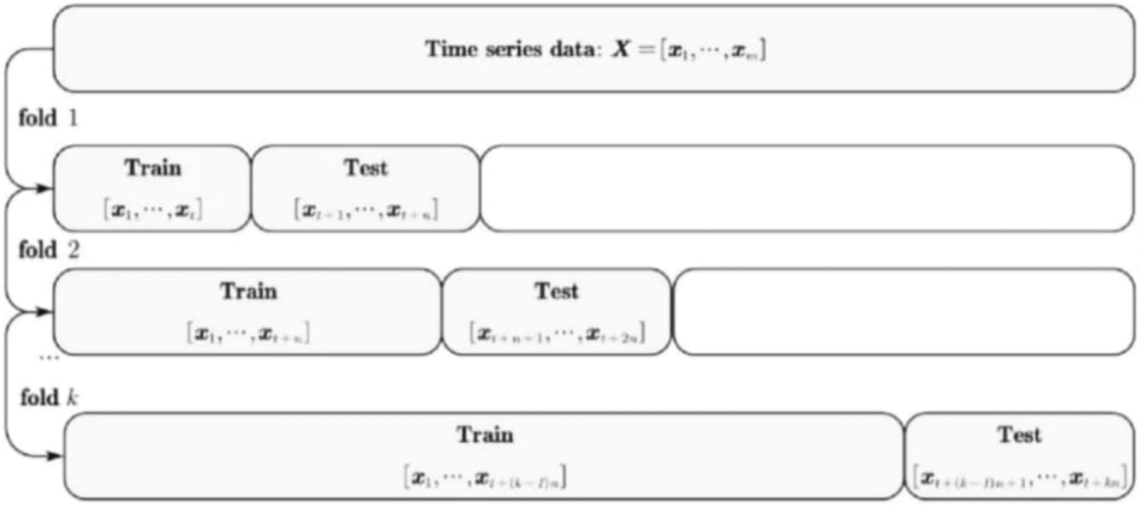

Figura 1: Diagrama ilustrativo de la metodología Cross Validation Expanding Window.

5. Herramientas utilizadas

- **Python** es un lenguaje de programación ampliamente utilizado en análisis de datos y aprendizaje automático debido a su simplicidad y poder. Ofrece una amplia gama de bibliotecas y frameworks que facilitan el procesamiento de texto, la manipulación de datos y la implementación de algoritmos de machine learning.
- **SAP** es un sistema de gestión de bases de datos relacional (RDBMS) desarrollado por SAP AG. Este sistema ofrece soporte para una amplia variedad de tipos de datos y funcionalidades avanzadas, lo que lo convierte en una solución eficiente para gestionar grandes volúmenes de información. SAP es utilizado principalmente en productos donde cumple un papel fundamental en el almacenamiento y procesamiento de datos, enfocados a la inteligencia de negocio.

  Cabe destacar que la base de datos de SAP es una tecnología propietaria, lo que significa que no está disponible para su uso en productos o sistemas externos a la suite de SAP y no utiliza el lenguaje de consulta estructurado (SQL) de manera estándar. Por lo tanto, SAP en sí mismo no es una base de datos SQL.
- **Qlik** es una plataforma poderosa de inteligencia de negocio (BI) y visualización de datos que permite a las organizaciones transformar datos en bruto en conocimientos accionables mediante reportes y análisis intuitivos. Sus principales productos están diseñados para facilitar la toma de decisiones basada en datos, proporcionando a los usuarios paneles interactivos.

  El motor asociativo de Qlik permite explorar datos provenientes de múltiples fuentes de manera integrada, descubriendo patrones y relaciones ocultas que las herramientas tradicionales de BI basadas en consultas podrían pasar por alto. La plataforma ofrece funcionalidades robustas de reporte, incluyendo visualizaciones personalizables, alertas automatizadas y actualizaciones de datos en tiempo real, lo que la hace especialmente adecuada para entornos que requieren reportes dinámicos y escalables.

6. Implementación

La implementación fue dividida en tres fases:

6.1 Primera fase

La implementación realizada consistió en un proceso ETL (Extracción, Transformación y Carga) aplicado sobre los datos en crudo, siguiendo parámetros de rendimiento previamente definidos en conjunto con el área financiera de la organización. Como resultado de este proceso, se seleccionaron 43 cuentas del Estado de Pérdidas y Ganancias (P&L), considerando un periodo de análisis de 36 meses comprendidos entre febrero de 2022 y septiembre de 2024. Las demás cuentas fueron descartadas debido a que presentaban numerosos valores faltantes sin un patrón claro, lo que dificultaba la aplicación efectiva de métodos de imputación.

6.2 Segunda fase

En la segunda fase de la implementación, se plantearon varios modelos para realizar las predicciones requeridas, resultando en la selección de los cuatro modelos previamente mencionados en el desarrollo del marco teórico. Estos modelos fueron elegidos teniendo en cuenta la naturaleza de los datos. Cada modelo fue seleccionado con base en su capacidad para manejar las características particulares de los datos financieros:

- LSTM: Ideal para manejar dependencias a largo plazo en series temporales.
- SARIMA: Eficaz para capturar estacionalidad y tendencias en datos estacionarios.
- NeuralProphet: Combina flexibilidad y facilidad de uso, lo que es útil para datos con tendencias complejas.
- XGBoost: Excelente para datos estructurados y predicciones precisas con características derivadas.

6.3 Tercera fase

Para la tercera fase, aplicamos la metodología de *cross-validation* para realizar una distribución eficiente de los datos con el objetivo de evaluar el desempeño de los diferentes modelos. Este enfoque nos permitió seleccionar las mejores predicciones de cada uno de los estados financieros, basándonos en métricas clave como el **MAPE** (Mean Absolute Percentage Error) y el **RMSE** (Root Mean Square Error). La metodología de *cross-validation* fue utilizada consistentemente a lo largo de este proceso para garantizar resultados robustos y comparables entre los modelos evaluados.

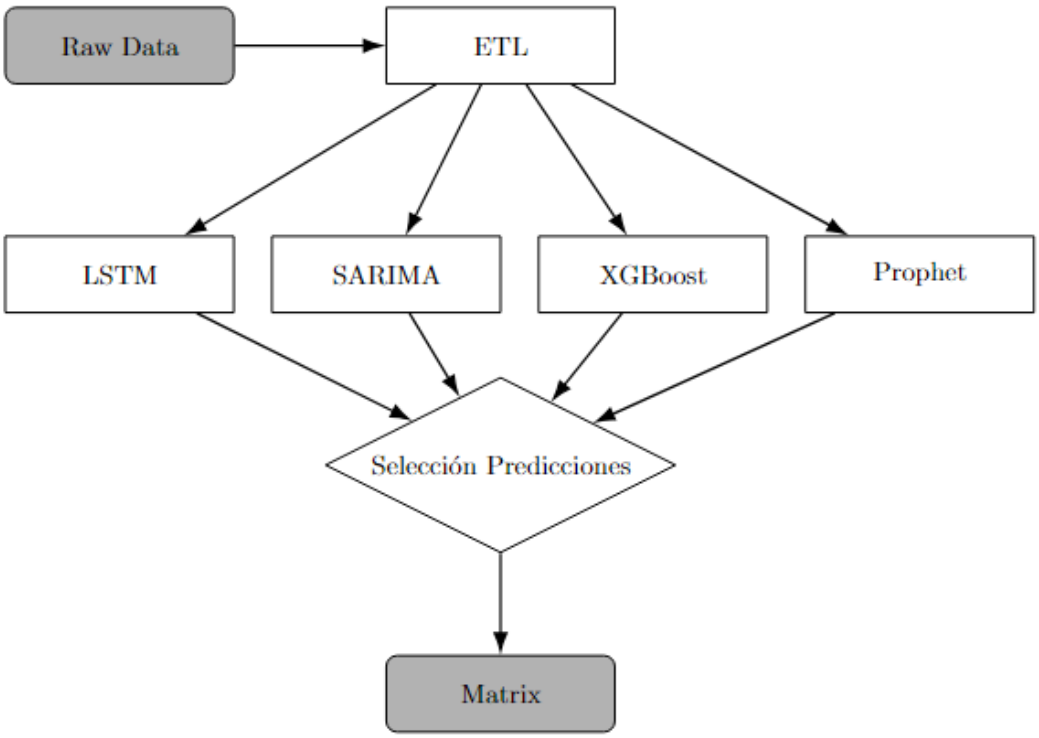

Figura 2: Diagrama de flujo que ilustra el desarrollo del proyecto, destacando las tres fases principales y sus componentes clave.

7. Resultados y Conclusiones

7.1 Resultados

En la siguiente tabla se muestran los porcentajes en los que cada modelo obtuvo el mejor desempeño relativo, así como la proporción de datos que se usó en la validación cruzada con ventana expandida. Dicho esquema inicia con un cierto porcentaje de datos para entrenamiento y en cada nuevo pliegue se amplía la ventana de entrenamiento conservando el orden temporal:

Modelo	**% de Mejores Desempeños**	**Proporción en CV (Ventana Expandida)**
SARIMA	26 %	Entrenamiento inicial: 70 %
XGBoost	24 %	Último pliegue: 85 %
NeuralProphet	18 %	(5 pliegues en total)
LSTM	32 %	

Cuadro 1: Ejemplo de porcentajes de éxitos para cada modelo y proporciones de entrenamiento utilizadas en validación cruzada con ventanas crecientes (5 pliegues).

- **El Porcentaje de Éxitos** indica la fracción de escenarios en que cada modelo obtuvo el menor error de pronóstico frente a los demás.
- **La Proporción en CV (Ventana Expandida)** describe cómo se llevó a cabo la validación: se inició con un 70 % de los datos en la ventana de entrenamiento y se fue incrementando hasta alrededor de un 85 % en el último pliegue, asegurando que los datos de test siempre provinieran de observaciones posteriores en el tiempo.

7.2 Recomendaciones

- Al momento de analizar las series de tiempo derivadas de estados financieros, especialmente aquellas que muestran patrones de estacionalidad, cierta tendencia y fluctuaciones de corto plazo, no es inusual observar que los modelos LSTM y SARIMA tengan un mejor rendimiento a los de XGBoost (con pocas variables rezagadas) y NeuralProphet (sin componentes autorregresivas). Las ventajas observadas para SARIMA y LSTM en datos financieros suelen explicarse por el uso de componentes estacionales explícitos en el modelo SARIMA y la capacidad de aprendizaje no lineal y de memorización a corto plazo en la red neuronal LSTM.

  En cuanto a , los modelos XGBoost y NeuralProphet pueden requerir configuraciones adicionales y una mayor afinación (más rezagos, ajuste de hiperparámetros, funciones de estacionalidad más amplias, etc.) para alcanzar o superar el desempeño de los otros modelos. En definitiva, la elección del método de pronóstico depende tanto de la naturaleza de los datos (tendencias, estacionalidades, ruidos) como de la capacidad de cada modelo para capturar dichas características.

- Para una empresa contar con una buena calidad en las predicciones de sus estados financieros tal como lo es el P&L junto con un buen conocimiento del negocio es un factor clave para mejorar la planificación financiera, anticipando ingresos y gastos esto hace más fácil la creación de presupuestos precisos y ajustados a la realidad de la organización. Además, contribuyen a la gestión de riesgos al identificar variaciones recurrentes que permiten prevenir déficits y aprovechar de manera efectiva los períodos donde se tiene mayor rentabilidad. Al ofrecer proyecciones confiables, estas apoyan decisiones estratégicas en áreas como inversiones y auditorias, fortaleciendo la capacidad del área financiera para optimizar recursos, garantizar la sostenibilidad y mejorar su desempeño.

8. Referencias

1. Hochreiter, S., Schmidhuber, J. (1997). "Long Short-Term Memory". Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
2. Hyndman, R. J., Athanasopoulos, G. (2018). "Forecasting: Principles and Practice". OTexts. Available at: https://otexts.com/fpp3/
3. Taylor, S., Letham, B. (2018). "Forecasting at Scale". The American Statistician, 72(1), 37-45. https://doi.org/10.1080/00031305.2017.1380080
4. Chen, T., Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System". Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794. https://doi.org/10.1145/2939672.2939785
5. Seabold, S., Perktold, J. (2010). "Statsmodels: Econometric and Statistical Modeling with Python". Proceedings of the 9th Python in Science Conference, 57-61.
6. Goodfellow, I., Bengio, Y., Courville, A. (2016). "Deep Learning". MIT Press.
7. Hamilton, J. D. (1994). "Time Series Analysis". Princeton University Press.
8. Box, G. E. P., Jenkins, G. M., Reinsel, G. C. (2015). "Time Series Analysis: Forecasting and Control". Wiley.
9. SAP SE. (2023). "SAP Analytics Cloud for Predictive Scenarios". SAP Documentation. Retrieved from: https://help.sap.com
10. QlikTech International AB. (2023). "Qlik Sense Documentation". Available at: https://help.qlik.com
11. Laptev, N., Yosinski, J., Li, E., Smyl, S. (2017). "Time-series Extreme Event Forecasting with Neural Networks at Uber". Proceedings of the 31st International Conference on Neural Information Processing Systems, 5370-5380.
12. Durbin, J., Koopman, S. J. (2012). "Time Series Analysis by State Space Methods". Oxford University Press.