



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Software

**Desarrollo de un sistema web para la predicción de
hábitos financieros en personas de Lima Metropolitana
mediante Machine Learning: una implementación
basada en CRISP-DM y Scrum**

TESIS

Para optar el Título Profesional de Ingeniero de Software

AUTOR

Jhonathan Adrián CENTTI PORTUGAL

ASESOR

Norberto Ulises ROMÁN CONCHA

Lima, Perú

2026



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Centti, J. (2026). *Desarrollo de un sistema web para la predicción de hábitos financieros en personas de Lima Metropolitana mediante Machine Learning: una implementación basada en CRISP-DM y Scrum*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Software]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios autor/ asesor

Datos de autor	
Nombres y apellidos	Jhonathan Adrián Centti Portugal
Tipo de documento de identidad	DNI
Número de documento de identidad	70189536
URL de ORCID	https://orcid.org/0000-0001-5071-3197
Datos de asesor	
Nombres y apellidos	Norberto Ulises Román Concha
Tipo de documento de identidad	DNI
Número de documento de identidad	08510560
URL de ORCID	https://orcid.org/0000-0002-3302-7539
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	José Cesar Piedra Isusqui
Tipo de documento	DNI
Número de documento de identidad	25628915
Miembro del jurado 1	
Nombres y apellidos	Gustavo Arredondo Castillo
Tipo de documento	DNI
Número de documento de identidad	06058679
Datos de investigación	
Línea de investigación	Data Science
Grupo de investigación	Itdata
Agencia de financiamiento	No

Ubicación geográfica de la investigación	Perú, Lima Latitud: -11.987237 Longitud: -77.097446 Latitud: 11°59'14.1" Sur Longitud: 77°05'50.1" Oeste
Año o rango de años en que se realizó la investigación	2025-2026
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
Escuela Profesional de Ingeniería de Software

Acta de Sustentación de Tesis

*Siendo las dieciséis (16:00) horas del día veintiocho (28) del mes febrero del año 2026, se reunieron en la sala presidido por el Mg. Jose Cesar Piedra Isusqui (Presidente), Lic. Gustavo Arredondo Castillo (Miembro) y el Mg. Norberto Ulises Román Concha (Asesor), para la sustentación de la tesis Intitulada: “**DESARROLLO DE UN SISTEMA WEB PARA LA PREDICCIÓN DE HÁBITOS FINANCIEROS EN PERSONAS DE LIMA METROPOLITANA MEDIANTE MACHINE LEARNING: UNA IMPLEMENTACIÓN BASADA EN CRISP-DM Y SCRUM**”, por el Bachiller Jhonathan Adrián Centti Portugal, para obtener el Título Profesional de Ingeniero de Software.*

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a responder las preguntas formuladas por los Miembros del Jurado.

El Bachiller, en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez las preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el bachiller obtuvo la nota de 19 (DIECINUEVE).

A continuación, el Presidente del Jurado Mg. Jose Cesar Piedra Isusqui, declara al Bachiller Ingeniero de Software.

Siendo 17.00 horas, se levantó la sesión.

Mg. Jose Cesar Piedra Isusqui
Presidente

Lic. Gustavo Arredondo Castillo
Miembro

Mg. Norberto Ulises Román Concha
Asesor



Anexo 1

CERTIFICADO DE SIMILITUD

Yo, Norberto Ulises Román Concha en mi condición de asesor acreditado con el Dictamen de la UPG o de la EP N°000004-2025-EPISW-FISI/UNMSM de la **tesis** cuyo título es **DESARROLLO DE UN SISTEMA WEB PARA LA PREDICCIÓN DE HÁBITOS FINANCIEROS EN PERSONAS DE LIMA METROPOLITANA MEDIANTE MACHINE LEARNING: UNA IMPLEMENTACIÓN BASADA EN CRISP-DM Y SCRUM**, presentado por el **bachiller Centti Portugal Jhonathan Adrián** para optar **título de INGENIERO DE SOFTWARE**, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el **8%** de similitud, nivel PERMITIDO para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**. Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del Título profesional de Ingeniero de Software.

Lima, 10___de marzo___de 2026__

Firma del Asesor

Nombres y Apellidos del asesor: NORBERTO ULISES ROMAN CONCHA

DNI: 08510560

* Según corresponda: bachiller, magíster, doctor; título profesional o título de segunda especialidad

La vigencia del presente certificado corresponde a un período de noventa (90) días desde la fecha de su emisión; siempre que no haya observaciones en el texto durante el proceso de revisión por el jurado informante o surja alguna contingencia.

Dedicatoria

A mis padres, que sin su ayuda la
realización de este proyecto no
hubiera sido posible.

Agradecimientos

A mi asesor, a los docentes de mi vida universitaria, a mis padres, y a todas las personas que me ayudaron en este proyecto.

RESUMEN

La gestión eficaz de las finanzas personales representa un desafío significativo para muchas personas en Lima Metropolitana, en parte debido a la escasez de herramientas tecnológicas personalizadas que permitan anticipar comportamientos económicos. Esta tesis aborda dicho problema mediante el desarrollo e implementación de "FinSight", un sistema web basado en Machine Learning diseñado para predecir hábitos financieros y la propensión al ahorro. El proyecto se ejecutó utilizando una metodología híbrida que integra CRISP-DM, para estructurar el proceso de ciencia de datos, y Scrum, para la gestión ágil del desarrollo. Se definieron reglas heurísticas para crear dos variables objetivo: "Nivel de Hábito Financiero" (multiclase ordinal 1-5) y "Propensión al Ahorro" (binario 0-1), a partir de datos sociodemográficos y financieros (inicialmente sintéticos). Se compararon siete algoritmos de clasificación supervisada, seleccionando XGBoost como el más performante. Los modelos finales fueron optimizados mediante técnicas avanzadas como SMOTE y RandomizedSearchCV, y evaluados rigurosamente en un conjunto de prueba independiente. El modelo para "Nivel de Hábito Financiero" alcanzó un Accuracy del 80%, mientras que el modelo para "Propensión al Ahorro" logró un Accuracy del 94.5% y un ROC-AUC de 0.9920, demostrando alta capacidad predictiva. Estos modelos se integraron en la aplicación web "FinSight", validando la viabilidad del enfoque propuesto como una herramienta potencial para mejorar la toma de decisiones y la educación financiera personal.

Palabras clave: hábitos financieros, machine learning, predicción, CRISP-DM, Scrum, XGBoost, sistema web.

ABSTRACT

Effective personal finance management represents a significant challenge for many people in Metropolitan Lima, partly due to the scarcity of customised technological tools that enable economic behaviour to be anticipated. This thesis addresses this problem through the development and implementation of 'FinSight', a web-based system based on machine learning designed to predict financial habits and propensity to save. The project was executed using a hybrid methodology that integrates CRISP-DM to structure the data science process and Scrum for agile development management. Heuristic rules were defined to create two target variables: 'Financial Habit Level' (ordinal multi-class 1-5) and 'Propensity to Save' (binary 0-1), based on sociodemographic and financial data (initially synthetic). Seven supervised classification algorithms were compared, selecting XGBoost as the best performer. The final models were optimised using advanced techniques such as SMOTE and RandomisedSearchCV, and rigorously evaluated on an independent test set. The model for 'Level of Financial Habit' achieved an accuracy of 80%, while the model for 'Propensity to Save' achieved an accuracy of 94.5% and a ROC-AUC of 0.9920, demonstrating high predictive power. These models were integrated into the 'FinSight' web application, validating the viability of the proposed approach as a potential tool for improving decision-making and personal financial education.

Keywords: financial habits, machine learning, prediction, CRISP-DM, Scrum, XGBoost, web system.

ÍNDICE GENERAL

RESUMEN	4
ABSTRACT	5
ÍNDICE GENERAL	6
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	11
CAPÍTULO I	12
PLANEAMIENTO METODOLÓGICO	12
1.1. Antecedentes	12
1.2. Formulación del problema	13
1.2.1. Problema principal	14
1.2.2. Problemas específicos	14
1.3. Importancia	14
1.4. Motivación	15
1.5. Objetivos	15
1.5.1. Objetivo general	15
1.5.2. Objetivos específicos	16
1.6. Justificación	16
1.7. Alcances y Limitaciones	17
1.7.1. Alcance social	17
1.7.2. Alcance técnico	17
1.7.3. Alcance económico	17
1.7.4. Alcance tecnológico	18
1.7.5. Limitaciones	18
1.8. Organización de la Tesis	19
CAPÍTULO II	20
MARCO TEÓRICO	20
2.1. Sistema web	20
2.1.1. Diseño de un sistema web	20
2.2. Base de Datos	21
2.2.1. Bases de Datos Relacional	21
2.2.2. Bases de Datos No Relacional	21
2.2.3. Base de Datos PostgreSQL	22
2.3. Conocimiento Financiero	22
2.4. Hábitos financieros	23

2.5. Análisis predictivos.....	24
2.6. Metodología Scrum.....	24
2.7. Metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) ...	26
2.8. Desafío de integrar CRISP-DM y SCRUM	28
2.8.1. Modelos Híbridos y Adaptaciones.....	28
2.8.2. Casos de Estudio y Aplicaciones Prácticas	29
2.9. Machine learning	29
2.9.1. Modelo Árboles de decisión.....	29
2.9.2. Algoritmo XGBoost (eXtreme Gradient Boosting)	29
2.9.3. Algoritmo K-Nearest Neighbors (KNN)	30
2.9.4. SVM support vector machines.....	30
2.9.5. Regresión lineal.....	30
2.9.6. Regresión logística.....	30
2.9.7. Random Forest.....	30
CAPÍTULO III	32
ESTADO DEL ARTE.....	32
3.1. Metodología de la investigación.....	32
3.2. Planeamiento.....	33
3.2.1. Criterios de Inclusión y Exclusión	34
3.3. Desarrollo	35
3.3.1. Proceso de Selección de Estudios	35
3.4. Resultados.....	36
3.5. Análisis	40
3.6. Conclusiones de la revisión sistemática de la literatura	44
CAPÍTULO IV: MODELO DE MACHINE LEARNING PARA LA PREDICCIÓN DE HÁBITOS FINANCIEROS Y PROPENSIÓN AL AHORRO	45
4.1. Aporte	45
4.2. Metodología CRISP-DM para el Desarrollo del Modelo.....	47
4.2.1. Comprensión del Negocio	47
4.2.2. Comprensión de los datos.....	48
4.2.3. Preparación de los datos.....	49
4.2.4. Modelado	50
4.2.5. Evaluación	51
4.2.6. Despliegue del modelo.....	52
4.3. Etiquetado de datos y reglas heurísticas.....	53
4.3.1. Variable de análisis	53

4.3.2. Criterios de etiquetado	54
4.4. Proceso de construcción del modelo de machine learning	55
CAPÍTULO V: APORTE PRACTICO	57
5.1 Introducción al Desarrollo	57
5.2 Arquitectura del Sistema "FinSight"	57
5.3 Ejecución de Sprints Metodológicos (CRISP-DM + Scrum)	60
5.4 Implementación del Sistema Web "FinSight"	77
5.4.1 Autenticación y Acceso	78
5.4.2 Formulario de Predicción.....	78
5.4.3 Visualización de Resultados	79
5.4.4 Historial y Módulos de Soporte	79
CAPÍTULO VI: RESULTADOS	82
6.1 Introducción.....	82
6.2 Conjunto de Datos de Prueba	82
6.3 Métricas de Evaluación	82
6.4 Resultados del Modelo 1: Nivel de Hábito Financiero (Multiclase).....	83
6.5 Resultados del Modelo 2: Propensión al Ahorro (Binario).....	86
CAPÍTULO VII: DISCUSIÓN	88
7.1 Interpretación de los Resultados	88
7.2 Comparación con el Estado del Arte.....	89
7.3 Limitaciones del Estudio.....	91
7.4 Implicaciones y Contribuciones	92
CAPÍTULO VIII: CONCLUSIONES Y RECOMENDACIONES	94
8.1 Conclusiones.....	94
8.2 Recomendaciones.....	95
8.3 Trabajos Futuros	96
Referencias bibliográficas	98
Anexos	105
Anexo A: Fragmentos de Código Relevantes del Proceso de Modelado	105
Anexo B: Evaluación de Rendimiento de Modelos Baseline (Propensión al Ahorro)	109
Anexo C: Configuración del Entorno de Despliegue del Sistema "FinSight"	123
Anexo D: Guía Rápida de Usuario para "FinSight"	124

ÍNDICE DE FIGURAS

Figura 1 <i>Arquitectura de PostgreSQL</i>	23
Figura 2 <i>Fases del modelo CRISP-DM</i>	26
Figura 3 <i>Tareas y subtareas del proceso CRISP-DM</i>	27
Figura 4 <i>Adaptación de SCRUM y CRISP-DM</i>	28
Figura 5 <i>Proceso de investigación</i>	36
Figura 6 <i>Método general de predicción de hábitos financieros y propensión al ahorro</i>	46
Figura 7 <i>Diagrama de arquitectura lógica del sistema "FinSight"</i>	58
Figura 8 <i>Diagrama de arquitectura tecnológica del sistema "FinSight"</i>	59
Figura 9 <i>Diagrama de arquitectura física del sistema "FinSight"</i>	59
Figura 10 <i>Código Python para la carga inicial del dataset (.read_csv)</i>	60
Figura 11 <i>Salida del comando .head() mostrando los datos crudos iniciales.</i> .	61
Figura 12 <i>Código Python de las funciones categorizar_nivel y categorizar_frecuencia</i>	62
Figura 13 <i>Salida del comando .head(10) mostrando los datos tras aplicar las funciones de categorización</i>	63
Figura 14 <i>Código Python para el cálculo de SCORE_H, SCORE_A y la creación de variables objetivo mediante quintiles y mediana</i>	65
Figura 15 <i>Código Python aplicando el umbral de mediana y eliminando (.drop) las columnas de score intermedias</i>	66
Figura 16 <i>Salida del comando .head() mostrando el DataFrame con las variables objetivo finales ('Nivel de Hábito Financiero', 'Propensión al Ahorro')</i>	67
Figura 17 <i>Código Python eliminando (.drop) las columnas 'Nivel de endeudamiento' y 'Nivel de ahorro' transformadas</i>	68
Figura 18 <i>Código Python guardando el dataset final procesado (.to_csv)</i>	68
Figura 19 <i>Código Python importando train_test_split y realizando la división estratificada 80/20</i>	70
Figura 20 <i>Código Python completo para entrenar y evaluar el modelo baseline de Regresión Logística multiclase</i>	71
Figura 21 <i>Código Python completo para entrenar y evaluar el modelo baseline XGBoost multiclase (incluye salida de métricas)</i>	72
Figura 22 <i>Código Python configurando RandomizedSearchCV para la optimización de hiperparámetros de XGBoost</i>	73
Figura 23 <i>Código Python aplicando SMOTE y RandomizedSearchCV para el modelo multiclase optimizado</i>	73
Figura 24 <i>Matriz de Confusión del modelo XGBoost optimizado para "Nivel de Hábito Financiero" (multiclase)</i>	74
Figura 25 <i>Código Python aplicando SMOTE y RandomizedSearchCV para el modelo binario optimizado</i>	76
Figura 26 <i>Matriz de Confusión y classification_report del modelo XGBoost optimizado para "Propensión al Ahorro" (binario)</i>	77

Figura 27	<i>Interfaz de inicio de sesión del sistema web "FinSight".....</i>	78
Figura 28	<i>Interfaz del formulario para la predicción de hábitos financieros.</i>	78
Figura 29	<i>Interfaz de visualización de resultados de predicción.</i>	79
Figura 30	<i>Interfaz del historial de resultados del usuario.</i>	80
Figura 31	<i>Interfaz conceptual del chat de ayuda.</i>	80
Figura 32	<i>Interfaz de la encuesta de satisfacción del usuario.</i>	81
Figura 33	<i>Matriz de Confusión - Modelo 1 (Nivel de Hábito Financiero)</i>	85
Figura 34	<i>Matriz de Confusión - Modelo 1 (Nivel de Hábito Financiero)</i>	85
Figura 35	<i>Matriz de Confusión - Modelo 2 (Propensión al Ahorro).....</i>	86
Figura 36	<i>Curva ROC - Modelo 2.....</i>	87
Figura 37	<i>Importancia de Variables - Modelo 2.....</i>	87

ÍNDICE DE TABLAS

Tabla 1	<i>Sintaxis de consulta utilizada en la búsqueda</i>	34
Tabla 2	<i>Artículos potenciales y seleccionados</i>	36
Tabla 3	<i>Artículos seleccionados</i>	37
Tabla 4	<i>Modelos de Machine Learning aplicados a hábitos financieros</i>	40
Tabla 5	<i>Integración de CRISP-DM y Scrum</i>	41
Tabla 6	<i>Arquitecturas y tecnologías usadas en sistemas web financieros</i>	42
Tabla 7	<i>Desafíos en la implementación local de sistemas predictivos financieros</i>	43
Tabla 8	<i>Métricas de Rendimiento - Modelo 1 (Nivel de Hábito Financiero)</i>	84
Tabla 9	<i>Métricas de Rendimiento - Modelo 2 (Propensión al Ahorro)</i>	86

CAPÍTULO I

PLANEAMIENTO METODOLÓGICO

1.1. Antecedentes

Diversos estudios, tanto nacionales como internacionales, han mostrado que el uso combinado de metodologías estructuradas como CRISP-DM y SCRUM ofrece resultados consistentes en el desarrollo de sistemas predictivos y de gestión. Mejía Rodríguez et al. (2024) y Ayele (2020) reportan experiencias exitosas al aplicar CRISP-DM en ámbitos como la clasificación de enfermedades y la minería de ideas, donde resaltan la importancia de un proceso riguroso en la preparación y el modelado de los datos. Por su parte, Studer et al. (2020) plantean una adaptación de CRISP-DM orientada a asegurar la calidad del proceso analítico, lo que confirma la versatilidad y vigencia de este enfoque en proyectos vinculados con la ciencia de datos.

Por otro lado, múltiples estudios han validado SCRUM como un marco ágil efectivo para el desarrollo de software en diversos sectores. Investigaciones como las de Toledo González et al. (2023) y Pinto y Leme (2024) quienes evidencian mejoras significativas en eficiencia operativa, tiempos de entrega y satisfacción de usuarios. Asimismo, investigaciones nacionales, como las de Pariona Leonardo (2021) y Palomino Salazar (2024), confirmaron que el uso de SCRUM en proyectos de desarrollo web mejora la gestión de procesos, la colaboración entre equipos y la adaptabilidad al cambio.

En el campo financiero, Guamán Yunga (2024) y Rojas Pecho y Acevedo Mortola (2025) emplearon modelos de aprendizaje automático para anticipar comportamientos de clientes y estimar costos, logrando niveles de precisión elevados y consistentes. Sus resultados evidencian que las herramientas predictivas pueden fortalecer de manera significativa los procesos de decisión en instituciones financieras, al aportar proyecciones más confiables sobre el desempeño y las necesidades de los usuarios.

En conjunto, estos antecedentes respaldan la pertinencia de desarrollar un sistema web de carácter predictivo que, siguiendo la estructura metodológica de CRISP-DM y las dinámicas ágiles de SCRUM, incorpore técnicas de machine learning para anticipar hábitos financieros. La propuesta se alinea con la tendencia creciente de utilizar analítica avanzada en situaciones cotidianas, con el propósito de mejorar la gestión personal y organizacional mediante soluciones tecnológicas más oportunas y eficientes.

1.2. Formulación del problema

En la actualidad, muchas personas en Lima Metropolitana enfrentan serias dificultades para gestionar de manera efectiva sus finanzas personales. Esta situación se debe, en gran medida, a la ausencia de herramientas tecnológicas accesibles y personalizadas que les permitan comprender y anticipar sus propios hábitos financieros. A pesar del crecimiento sostenido del sector fintech y del avance de las tecnologías de análisis de datos, las soluciones disponibles en el mercado no consideran con suficiente profundidad el contexto socioeconómico de los usuarios locales, ni incorporan modelos predictivos que faciliten una toma de decisiones informada.

Se resalta que el desarrollo de soluciones inteligentes implica afrontar retos técnicos que van desde la elección del modelo de aprendizaje automático más adecuado hasta la garantía de contar con datos fiables y representativos. A ello se suma la importancia de diseñar interfaces claras y usables, capaces de motivar al usuario a incorporar estas herramientas en sus actividades diarias. Además, todavía son escasos los proyectos que articulan de manera coherente metodologías como CRISP-DM, centrada en el análisis y procesamiento de datos, con enfoques ágiles como SCRUM, habituales en la ingeniería de software. Esta combinación metodológica, aunque poco explorada, resulta clave para construir plataformas web que integren capacidad predictiva y utilidad formativa de manera efectiva.

En este contexto, se identifica la necesidad de desarrollar e implementar un sistema web que, mediante técnicas avanzadas de Machine Learning y una arquitectura centrada en el usuario, permita predecir los hábitos financieros de las personas de Lima Metropolitana. Este sistema debe ofrecer predicciones

precisas y recomendaciones prácticas que ayuden a los usuarios a mejorar su gestión financiera, contribuyendo así no solo a su bienestar individual, sino también al fortalecimiento de la educación financiera y al desarrollo de soluciones tecnológicas innovadoras en el ámbito de la analítica personal.

1.2.1. Problema principal

¿Cómo se puede desarrollar un sistema web, basado en Machine Learning, que permita predecir los hábitos financieros de las personas en Lima Metropolitana para mejorar su gestión financiera y apoyar su toma de decisiones?

1.2.2. Problemas específicos

PE1. ¿Qué algoritmos de Machine Learning son más adecuados para predecir los hábitos financieros de las personas en Lima Metropolitana con base en datos históricos?

PE2. ¿Cómo se puede diseñar una plataforma web que sea intuitiva, accesible y funcional para usuarios con distintos niveles de conocimiento tecnológico?

PE3. ¿Cómo validar la precisión y confiabilidad del modelo predictivo utilizando datos reales de usuarios de Lima Metropolitana?

PE4. ¿Cuál es el impacto del uso de un sistema web predictivo en las decisiones financieras personales de los usuarios?

1.3. Importancia

El desarrollo de un sistema web que utilice técnicas de aprendizaje automático para anticipar los hábitos financieros de los ciudadanos de Lima Metropolitana adquiere especial relevancia en un entorno donde la gestión económica personal sigue siendo un desafío cotidiano. Muchas personas no cuentan con herramientas que les permitan identificar con claridad sus patrones de gasto o evaluar las consecuencias de sus decisiones financieras, lo cual limita su capacidad de planificación y control (Guamán Yunga, 2024). Frente a esta realidad, una plataforma predictiva no solo aporta un componente tecnológico innovador, sino que se convierte también en un recurso formativo que contribuye

al fortalecimiento de la educación financiera y al desarrollo de hábitos más responsables.

Además, la aplicación de modelos predictivos en datos financieros se alinea con tendencias recientes en analítica personal y minería de datos aplicada, donde se ha demostrado que el análisis de datos históricos puede revelar patrones de comportamiento útiles para mejorar la planificación y el control de gastos (Calero Pérez, 2022; Sánchez & Pérez, 2023). Al brindar información personalizada y visualmente comprensible, se incrementa la probabilidad de adopción de estas tecnologías por parte de usuarios con distintos niveles de alfabetización digital y financiera.

1.4. Motivación

A pesar de la variedad de aplicaciones fintech existentes, muchas personas aún no encuentran herramientas que respondan específicamente a sus necesidades cotidianas, especialmente en contextos locales como Lima Metropolitana. Esta brecha se puede cerrar mediante soluciones inteligentes que no solo informen, sino que también predigan y guíen el comportamiento financiero de forma personalizada.

El presente proyecto parte del compromiso por mejorar la educación financiera mediante la tecnología, aplicando enfoques probados como CRISP-DM y SCRUM en el diseño de soluciones accesibles. Se busca no solo construir una herramienta útil, sino también demostrar cómo el uso responsable y bien contextualizado del Machine Learning puede tener un impacto directo y positivo en la vida de las personas.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar e implementar un sistema web basado en Machine Learning que permita predecir los hábitos financieros de las personas en Lima Metropolitana, con el fin de mejorar su gestión financiera.

1.5.2. Objetivos específicos

OE1. Identificar y seleccionar los algoritmos de Machine Learning más efectivos para la predicción de hábitos financieros.

OE2. Diseñar y desarrollar una plataforma web intuitiva y accesible para la visualización y análisis predictivo de datos financieros.

OE3. Evaluar la precisión del modelo de predicción mediante pruebas con datos históricos de usuarios en Lima Metropolitana.

OE4. Analizar el impacto del uso del sistema web en la toma de decisiones financieras de los usuarios.

1.6. Justificación

La gestión financiera personal en Lima Metropolitana evidencia carencias importantes, especialmente por la falta de herramientas predictivas que permitan a los ciudadanos comprender y anticipar sus propios hábitos económicos. Este vacío limita la toma de decisiones informadas y genera patrones de consumo poco sostenibles, por lo que el desarrollo de un sistema web capaz de ofrecer predicciones y recomendaciones personalizadas adquiere una relevancia particular como mecanismo de apoyo educativo y tecnológico.

En este contexto, el empleo de algoritmos de Machine Learning constituye un aporte innovador, ya que técnicas como Random Forest o Gradient Boosting permiten abordar grandes volúmenes de datos y ofrecer resultados con mayor precisión, consolidando así un enfoque moderno para el análisis de comportamientos financieros.

La propuesta no solo incorpora tecnologías avanzadas, sino que también se apoya en la integración de las metodologías CRISP-DM y SCRUM, cuya articulación permite combinar un proceso riguroso de ciencia de datos con una gestión ágil del desarrollo del software, garantizando una estructura ordenada, iterativa y adaptable a las necesidades reales de los usuarios. Esta combinación metodológica fortalece la calidad del producto final y facilita la entrega continua de funcionalidades que agregan valor.

Además de su aporte técnico, el proyecto tiene un impacto social y educativo significativo, pues brinda a los usuarios la posibilidad de reconocer sus propios patrones financieros y adoptar conductas más responsables y sostenibles, lo que, a largo plazo, podría contribuir al fortalecimiento de la economía local y a la mejora de la calidad de vida urbana.

Finalmente, esta iniciativa se posiciona como una contribución directa al campo del fintech y la analítica personal, ya que genera una base sólida para futuras investigaciones y desarrollos orientados a la gestión financiera, la personalización de servicios y la optimización de los procesos de decisión en diversos entornos tecnológicos y sociales.

1.7. Alcances y Limitaciones

1.7.1. Alcance social

El sistema busca generar un impacto positivo en la toma de decisiones financieras personales de los ciudadanos de Lima Metropolitana, brindando una herramienta predictiva que contribuya al desarrollo de mejores hábitos financieros. Al facilitar la comprensión de patrones de comportamiento económico, se espera fortalecer la educación financiera y fomentar una cultura de planificación responsable entre los usuarios.

1.7.2. Alcance técnico

Desde el punto de vista técnico, el proyecto abarca el desarrollo de un sistema web funcional basado en una arquitectura cliente-servidor. Incluye la integración de modelos de aprendizaje automático entrenados con datos financieros autodeclarados, así como la implementación de funcionalidades de visualización de resultados. Se utilizarán marcos metodológicos como CRISP-DM para el desarrollo del componente de ciencia de datos y SCRUM para la gestión ágil del desarrollo del sistema.

1.7.3. Alcance económico

Este proyecto no implica un análisis profundo de costos-beneficios en términos comerciales, ya que se enmarca en un contexto académico. Sin embargo, el sistema tiene el potencial de reducir gastos personales innecesarios al ayudar a

los usuarios a identificar patrones financieros ineficientes. A largo plazo, su implementación a mayor escala podría traducirse en beneficios económicos indirectos a través de una mejor gestión del dinero personal.

1.7.4. Alcance tecnológico

El desarrollo se realizará utilizando tecnologías web modernas (por ejemplo, frameworks como Django o Flask en el backend y React o Vue.js en el frontend), junto con bibliotecas de Machine Learning como Scikit-learn, TensorFlow o similares. El sistema se desplegará en un entorno web que permita el acceso multiplataforma y el uso interactivo por parte de los usuarios. También se considera el uso de bases de datos relacionales y/o no relacionales para el almacenamiento seguro y estructurado de la información recolectada.

1.7.5. Limitaciones

El alcance del sistema presenta varias limitaciones que deben ser consideradas para una adecuada interpretación de los resultados. En primer lugar, su cobertura geográfica se restringe a usuarios de Lima Metropolitana, lo que impide extrapolar los hallazgos a otras regiones sin realizar ajustes en los modelos.

Asimismo, la representatividad de los datos constituye un factor crítico, pues la calidad de las predicciones dependerá directamente de la cantidad, variabilidad y confiabilidad de la información proporcionada; datos escasos o sesgados pueden reducir el desempeño de los algoritmos.

A ello se suma la existencia de restricciones temporales derivadas del cronograma de la investigación, que pueden limitar la amplitud de las funcionalidades implementadas durante el desarrollo.

También se reconoce que, si bien se adoptarán prácticas responsables para proteger la información personal de los usuarios, el proyecto no incorpora un sistema integral de ciberseguridad, lo que representa una limitación en términos de protección avanzada de datos.

Finalmente, debe considerarse que los algoritmos de Machine Learning empleados no alcanzan precisión absoluta, ya que los hábitos financieros están

influenciados por factores personales y contextuales de alta variabilidad, lo que introduce un margen inevitable de incertidumbre en las predicciones.

1.8. Organización de la Tesis

La estructura de esta tesis ha sido diseñada para abordar de forma lógica y progresiva el desarrollo del sistema web propuesto. En el primer capítulo se presenta el planteamiento metodológico, que incluye los antecedentes, la definición del problema, los objetivos, la justificación y las hipótesis. Además, se detallan los alcances y limitaciones del estudio, proporcionando el marco contextual y metodológico que fundamenta la investigación.

El segundo capítulo está dedicado al marco teórico, donde se describen los conceptos fundamentales relacionados con el aprendizaje automático, las metodologías CRISP-DM y SCRUM, así como las tecnologías utilizadas en el desarrollo del sistema. Asimismo, se incluyen antecedentes teóricos y estudios previos que permiten sustentar y diferenciar la propuesta frente a otras soluciones similares.

El tercer capítulo aborda el diseño metodológico del estudio, especificando el tipo de investigación, el enfoque adoptado, las técnicas de recolección de datos, y el procedimiento seguido para llevar a cabo la implementación del sistema.

En el cuarto capítulo se detalla el proceso de desarrollo e implementación del sistema web. Se describen la arquitectura técnica, los componentes funcionales, los algoritmos de machine learning empleados y el uso de las metodologías CRISP-DM y SCRUM para la gestión y desarrollo iterativo del proyecto.

Finalmente, el quinto capítulo presenta los resultados obtenidos, el análisis de la precisión del modelo predictivo y la evaluación del impacto del sistema en la toma de decisiones financieras de los usuarios. El documento concluye con un conjunto de conclusiones y recomendaciones que abren la posibilidad de futuras investigaciones y mejoras sobre la propuesta desarrollada.

CAPÍTULO II

MARCO TEÓRICO

2.1. Sistema web

Se define como el conjunto de aplicaciones y servicios accesibles a través de internet, diseñados para mejorar la gestión de información y procesos de una organización (Espinosa, 2021). De acuerdo con Pressman (2021), es una aplicación informática que los usuarios pueden utilizar accediendo a un servidor web a través de internet mediante un navegador. A su vez, representa una solución tecnológica integrada que permite interacciones dinámicas, almacenamiento de datos y procesamiento de información en una plataforma digital accesible multiplataforma. Complementariamente, Sommerville (2019) lo define como un conjunto de componentes de software interconectados que proporcionan funcionalidades específicas a través de protocolos de red, facilitando la interacción entre usuarios y recursos computacionales mediante interfaces web.

2.1.1. Diseño de un sistema web

Castejón Villegas (2014) sostiene que el Modelo-Vista-Controlador (MVC) es un componente clave de la arquitectura de las aplicaciones web. El modelo de negocio parece completar el diseño y la implementación de la lógica que gestionará la funcionalidad principal después de que un patrón en primer plano separe y organice las interfaces de usuario (vistas) en bloques. La forma en que se plasma el modelo se refleja en la perspectiva. En términos generales, cuando el control recibe una orden, llevará a cabo la acción especificada en la orden, cambiando la visualización. Además, el modelo hace hincapié en la correcta comunicación de los elementos del flujo del sistema a través de su estructura.

2.2. Base de Datos

Es un repositorio organizado de datos donde se almacena y gestiona la información de los pacientes, facilitando su acceso y análisis (Xool et al., 2018).

2.2.1. Bases de Datos Relacional

Las bases de datos relacionales organizan los datos en tablas compuestas por filas y columnas, donde cada fila representa un registro único y cada columna un atributo específico. Este modelo, propuesto por Codd (1970), se basa en la teoría de conjuntos y permite establecer relaciones entre diferentes tablas mediante claves primarias y foráneas. Las bases de datos relacionales utilizan el lenguaje SQL (Structured Query Language) para la definición y manipulación de datos. Una característica clave de estas bases es el cumplimiento de las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), que garantizan la integridad y confiabilidad de las transacciones.

2.2.2. Bases de Datos No Relacional

Las **bases de datos no relacionales**, también conocidas como NoSQL, están diseñadas para manejar grandes volúmenes de datos no estructurados o semiestructurados. A diferencia del modelo relacional, estas bases no requieren un esquema fijo y ofrecen una mayor flexibilidad en la representación de datos. Existen varios tipos de bases de datos NoSQL, incluyendo:

- a) Bases de datos de documentos: almacenan datos en formatos como JSON o XML, permitiendo una estructura jerárquica y flexible.
- b) Bases de datos de grafos: optimizadas para representar y consultar relaciones complejas entre datos, utilizando nodos y aristas.
- c) Almacenes de clave-valor: utilizan un modelo simple donde cada clave única se asocia a un valor, facilitando accesos rápidos.
- d) Bases de datos orientadas a columnas: almacenan datos por columnas en lugar de filas, optimizando consultas analíticas.

Estas bases de datos son ideales para aplicaciones que requieren escalabilidad horizontal, alta disponibilidad y manejo eficiente de datos distribuidos.

2.2.3. Base de Datos PostgreSQL

IBM (2021) indica que, PostgreSQL es un sistema de gestión de bases de datos de código abierto que combina características de bases de datos relacionales y no relacionales. Desarrollado inicialmente en la Universidad de California en Berkeley, PostgreSQL se destaca por su conformidad con los estándares SQL y su extensibilidad. Entre sus características principales se incluyen:

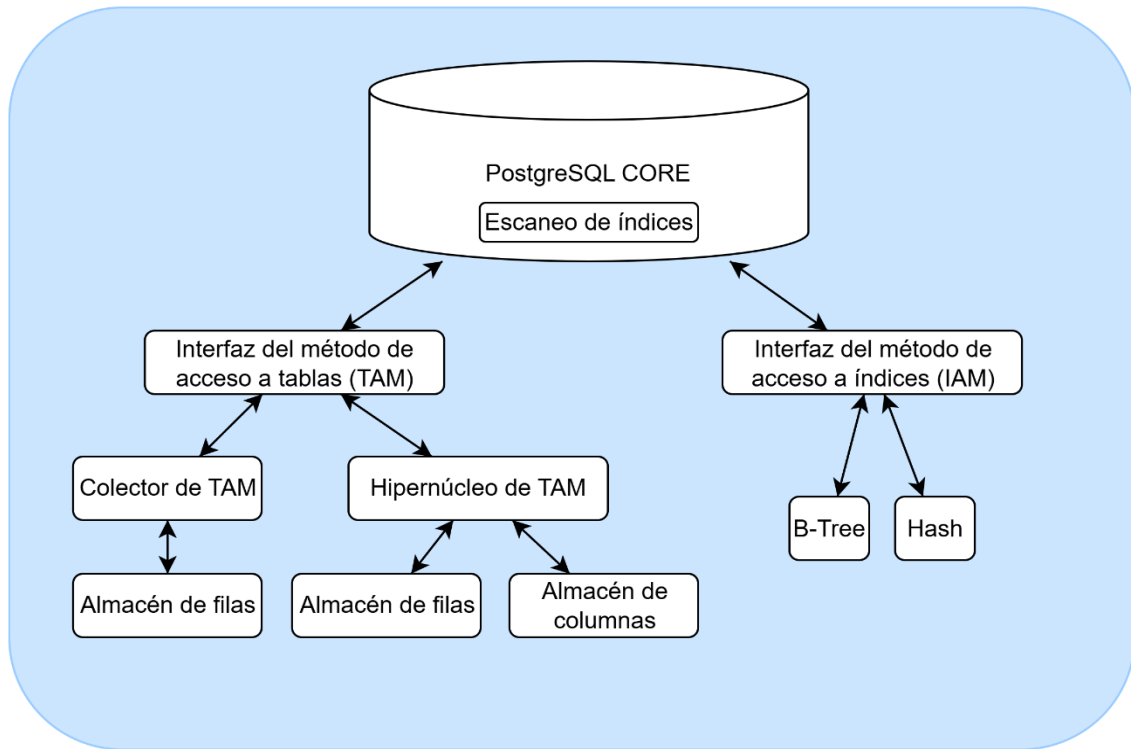
- a) Soporte para tipos de datos avanzados: como JSON, XML y arrays.
- b) Capacidades NoSQL: permitiendo el almacenamiento y consulta de datos no estructurados.
- c) Extensibilidad: los usuarios pueden definir sus propios tipos de datos, funciones y operadores.
- d) Control de concurrencia multiversión (MVCC): mejora el rendimiento en entornos multiusuario.

2.3. Conocimiento Financiero

El conocimiento financiero en otras investigaciones, también llamado alfabetización financiera, se asoció positivamente con el comportamiento gerencial (Kunaifi y Akbar, 2019). Este le permite a un individuo tomar decisiones informadas y efectivas con todos sus recursos financieros. Ser analfabeto financiero puede llevar a una decisión financiera incorrecta (Idris et al., 2018).

Figura 1

Arquitectura de PostgreSQL



Fuente: Elaboración propia

2.4. Hábitos financieros

Los hábitos financieros son comportamientos adquiridos de manera constante en relación con la administración del dinero, como el ahorro, el gasto responsable, el uso del crédito y la planificación económica. Estos influyen directamente en la salud financiera de las personas y se forman a lo largo del tiempo según factores como la educación, el entorno familiar y la experiencia personal. De acuerdo con el *Forbes Finance Council* (2020), desarrollar hábitos financieros saludables como registrar gastos, evitar deudas innecesarias y ahorrar regularmente es fundamental para lograr estabilidad y bienestar económico a largo plazo.

La predicción de hábitos financieros, de acuerdo con Kahneman y Tversky (2018), constituye un proceso de análisis sistemático que busca identificar patrones de comportamiento económico individual mediante técnicas

estadísticas y modelos predictivos que evalúan decisiones financieras. Desde la perspectiva de Ariely (2017), implica comprender los factores psicológicos, socioeconómicos y conductuales que influyen en la toma de decisiones, utilizando herramientas analíticas para anticipar tendencias y comportamientos económicos personales.

2.5. Análisis predictivos

El uso de enfoques estadísticos que emplean algoritmos, procedimientos y estrategias automatizadas para estimar la probabilidad de eventos futuros o identificar patrones de comportamiento a partir de información previa se conoce como análisis predictivo. Este tipo de análisis se basa en datos históricos que, procesados mediante técnicas avanzadas, permiten anticipar resultados y apoyar la toma de decisiones de manera informada.

Este análisis se basa en la extracción de un modelo analítico de datos históricos que pronostique tendencias o comportamientos futuros o, en su defecto, estime resultados desconocidos o poco conocidos mediante técnicas estadísticas matemáticas de modelización de forma que se combinen todos los datos, permitiendo la realización de análisis predictivos futuros examen de predicciones futuras.

Combinada con la estadística, la minería de datos permite crear modelos de predicción e identificar tendencias y correlaciones a través de grupos de datos estructurados y no estructurados, respectivamente (Big Data Marketer, 2018).

2.6. Metodología Scrum

Scrum se organiza en torno a roles clave (Propietario del Producto, Equipo de Desarrollo y Maestro Scrum), sprint y las reuniones diarias. Este enfoque iterativo permite entregas parciales funcionales en plazos cortos, lo que facilita la retroalimentación constante por parte del cliente y la mejora continua del producto (Promethean, 2023).

Roles clave en Scrum:

En el marco de la metodología Scrum, el desarrollo del proyecto requiere la participación de tres roles fundamentales que garantizan la correcta aplicación del enfoque ágil. El Propietario del Producto es quien se encarga de maximizar el valor del producto y gestionar el product backlog, priorizando las tareas que aportan mayor beneficio al objetivo general (IBM, 2022).

Junto a él, el Equipo de Desarrollo conforma un grupo interdisciplinario y autoorganizado de profesionales responsables de construir el incremento del producto en cada sprint, trabajando de manera colaborativa y autónoma para alcanzar los resultados previstos (Schwaber & Sutherland, 2020).

Finalmente, el Maestro Scrum actúa como facilitador del proceso, acompañando al equipo en la eliminación de impedimentos y asegurando que se respeten los principios, valores y prácticas establecidos por Scrum, de modo que el flujo de trabajo se mantenga estable y eficiente (Rubin, 2012).

Eventos principales en Scrum:

Dentro del marco de trabajo Scrum, cada sprint se organiza y desarrolla a través de una secuencia de eventos que permiten planificar, supervisar y mejorar continuamente el proceso. El Sprint Planning marca el inicio del ciclo, y en esta reunión el equipo establece qué tareas se ejecutarán y de qué manera se alcanzará el objetivo planteado para el sprint (Cohn, 2010).

A lo largo del desarrollo, el equipo participa diariamente en el Daily Scrum, un encuentro breve de quince minutos en el que se revisa el progreso, se identifican posibles impedimentos y se ajusta el plan de trabajo según las necesidades del día, favoreciendo la comunicación constante y la toma de decisiones rápidas (Cohn, 2010).

Al finalizar el sprint, la Sprint Review permite presentar a los interesados el incremento del producto, recibir retroalimentación y determinar si es necesario adaptar el rumbo del proyecto en función de los resultados obtenidos (Agile Spain, 2024).

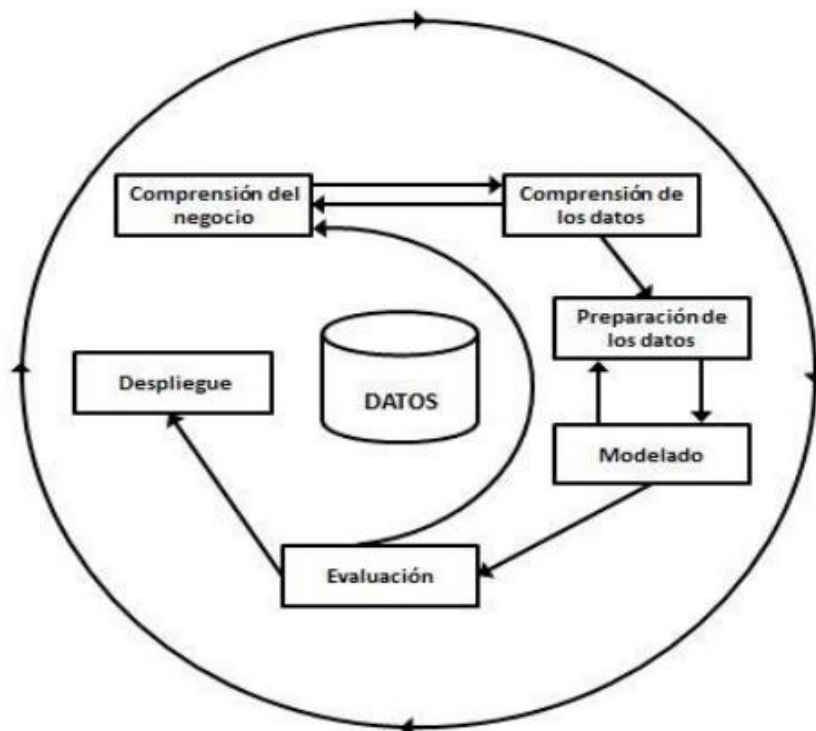
Finalmente, el equipo realiza la Sprint Retrospective, una reunión orientada a reflexionar sobre el proceso, analizar oportunidades de mejora y definir acciones concretas que optimicen el desempeño en el siguiente sprint (ENAE, 2023).).

2.7. Metodología CRISP-DM (Cross-Industry Standard Process for Data Mining)

CRISP-DM Es el proceso que organiza el desarrollo de un proyecto de minería de datos en una serie de fases o etapas, utilizando tanto mano de obra general como especializada para alcanzar los objetivos del proyecto. Con la posibilidad de avanzar y retroceder entre fases, estas etapas funcionan de forma iterativa y cíclica. Su fundamento es un modelo de proceso jerárquico que establece el ciclo de vida de las iniciativas de explotación de la información.

Figura 2 Fases del modelo CRISP-DM

Arquitectura de PostgreSQL



Fuente: Chapman et al. (2000).

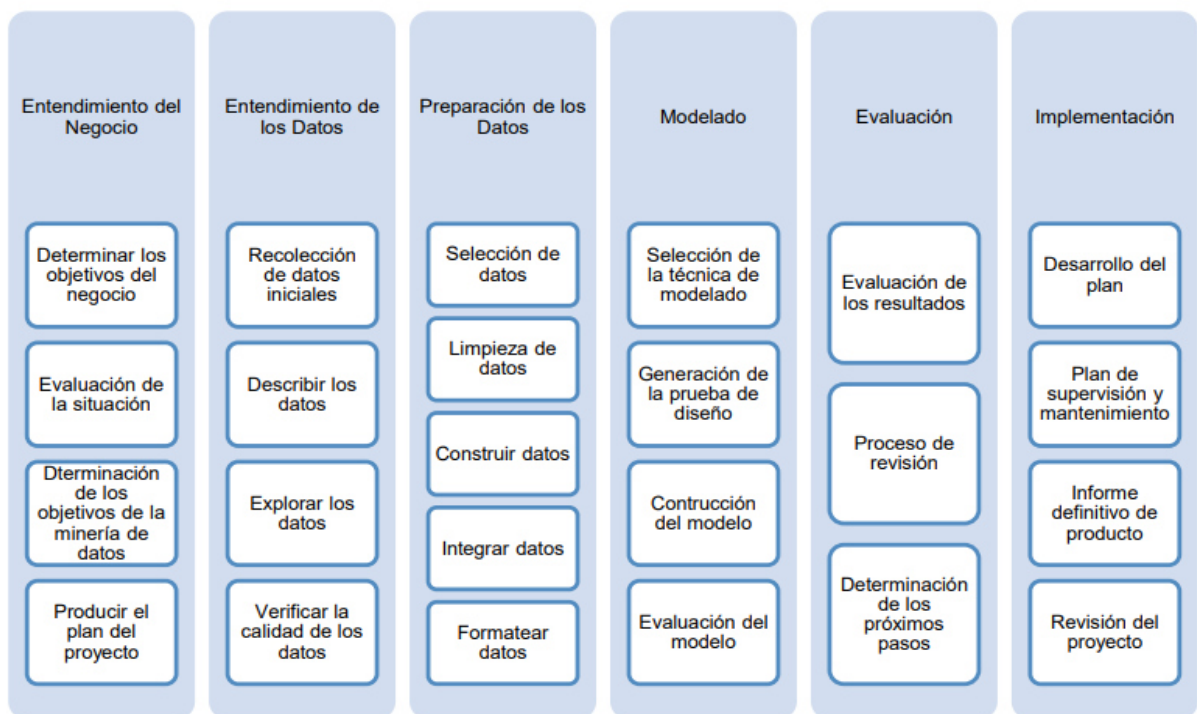
Según Hernández-Orallo et al. (2021) las fases de la metodología CRISP son las siguientes:

- Comprensión del negocio: Es donde trata de deducir los objetivos y requerimientos del proyecto desde una perspectiva de negocio.
- Comprensión de los datos: En fase se eligen y adapta los datos, para poder identificar los problemas de calidad de datos y así obtener datos óptimos para poder ser analizados.

- c) Preparación de los datos: Transformación de los datos. Se escogen los datos a trabajar y éstos pasan a una fase de limpieza, estructuración, integración y formateo.
- d) Modelamiento y evaluación: Separación y aplicación de Data Mining e Interpretación y evaluación. Se selecciona la técnica a trabajar, construyendo el modelo, para luego ser sometido a diferentes pruebas y evaluaciones.
- e) Despliegue del proyecto: Es donde se explota todo el potencial de los modelos y así intégralos en los procesos de toma de decisión de organización, y así difundir el conocimiento extraído, etcétera.

Figura 3

Tareas y subtareas del proceso CRISP-DM



Fuente: Chapman et al. (2000).

En general, las fases de CRISP-DM tienen una descripción sobre el proceso de análisis, pero los detalles específicos y las necesidades del usuario deben ser incluidas. Se necesita además de la actualización un mantenimiento constante para ser adaptado en los nuevos retos, como: big data, ciencia de datos, minería de datos centrada en el usuario.

2.8. Desafío de integrar CRISP-DM y SCRUM

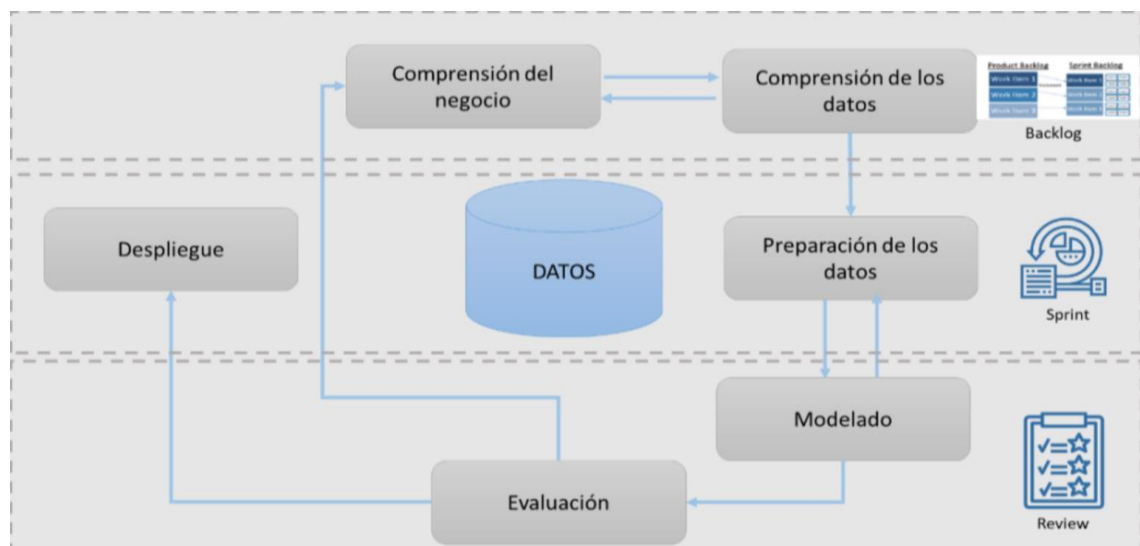
La integración de CRISP-DM y SCRUM presenta desafíos debido a las diferencias en sus enfoques. Mientras que CRISP-DM es un proceso estructurado y secuencial, SCRUM es iterativo y flexible. Esta disparidad puede generar tensiones en la planificación y ejecución de proyectos de ciencia de datos. Por ejemplo, la naturaleza exploratoria de la ciencia de datos puede dificultar la definición de requisitos claros al inicio de un sprint (Kraut & Transchel, 2022).

2.8.1. Modelos Híbridos y Adaptaciones

Para abordar estos desafíos, se han propuesto modelos híbridos que combinan elementos de CRISP-DM y SCRUM. Por ejemplo, el modelo SCRUM-DS integra artefactos, roles y eventos de SCRUM en las fases de CRISP-DM, permitiendo una gestión más ágil de los proyectos de ciencia de datos. Además, se han desarrollado enfoques como el Data Driven Scrum (DDS), que adapta SCRUM para proyectos de ciencia de datos, enfocándose en la creación, observación y análisis dentro de cada ciclo iterativo.

Figura 4

Adaptación de SCRUM y CRISP-DM



Fuente: Minoli, M. (2020).

2.8.2. Casos de Estudio y Aplicaciones Prácticas

Amirian, Abdollahzadeh & Sulaiman (2024) indican que, diversas organizaciones han implementado con éxito la combinación de CRISP-DM y SCRUM. Por ejemplo, en Petronas, se adoptó una estrategia híbrida que incorpora el proceso CRISP-DM dentro de sprints definidos, asignando tareas específicas a cada fase del proceso y utilizando roles de SCRUM para garantizar una entrega efectiva del producto.

2.9. Machine learning

Machine Learning de acuerdo con Ahirrao et al. (2023), es un subcampo de la inteligencia artificial (IA) que implica el uso de algoritmos para extraer información de grandes volúmenes de datos no estructurados, como texto, imágenes y videos. Los algoritmos de ML pueden aprender de los datos sin ser programados explícitamente, lo que les permite identificar patrones y hacer predicciones.

2.9.1. Modelo Árboles de decisión

Los árboles de decisión son modelos de aprendizaje supervisado que utilizan una estructura jerárquica para tomar decisiones basadas en características de los datos. Cada nodo interno representa una prueba sobre un atributo, cada rama el resultado de la prueba y cada hoja una etiqueta de clase o valor de salida. Recientes investigaciones han propuesto enfoques para mejorar la precisión e interpretabilidad de estos modelos mediante criterios de división parametrizados y técnicas de poda optimizadas (Balcan & Sharma, 2024).

2.9.2. Algoritmo XGBoost (eXtreme Gradient Boosting)

XGBoost es una implementación eficiente y escalable del algoritmo de gradient boosting, que combina múltiples árboles de decisión débiles para formar un modelo fuerte. Es ampliamente reconocido por su alto rendimiento predictivo y su capacidad para manejar diversos tipos de datos, incluyendo conjuntos estructurados, dispersos y series temporales (Budiawan, 2025).

2.9.3. Algoritmo K-Nearest Neighbors (KNN)

KNN es un algoritmo de aprendizaje supervisado que clasifica nuevas instancias basándose en las clases de sus 'k' vecinos más cercanos en el espacio de características. Es un método no paramétrico y basado en instancias, lo que significa que no hace suposiciones sobre la distribución de los datos y utiliza toda la información disponible para realizar predicciones. Su simplicidad y eficacia lo hacen adecuado para diversas aplicaciones, desde la predicción de ventas hasta el mantenimiento predictivo (Azis, Dinata & Hasdyna, 2024).

2.9.4. SVM support vector machines

Una Máquina de Vectores de Soporte (SVM) es un algoritmo de aprendizaje supervisado utilizado para clasificación y regresión. Su objetivo es encontrar el hiperplano que mejor separa los datos en diferentes clases, maximizando el margen entre los puntos de datos más cercanos de cada clase, conocidos como vectores de soporte (Negri, 2023).

2.9.5. Regresión lineal

La regresión lineal es una técnica estadística fundamental en el aprendizaje automático utilizada para modelar la relación entre una variable dependiente y una o más variables independientes. Su objetivo es encontrar la línea de mejor ajuste que minimice el error entre las predicciones y los valores reales, comúnmente utilizando el método de mínimos cuadrados (Fynd Academy, 2025).

2.9.6. Regresión logística

La regresión logística es un algoritmo de aprendizaje supervisado ampliamente utilizado, aplicado principalmente a problemas de clasificación binaria. Su simplicidad, interpretabilidad y sólida base matemática la han convertido en una opción popular en diversos dominios como el diagnóstico médico, la evaluación crediticia y el marketing (Labhade-Kumar et al., 2024).

2.9.7. Random Forest

Random Forest es un método de aprendizaje supervisado que utiliza múltiples árboles de decisión para mejorar el rendimiento en tareas de clasificación y regresión. Cada árbol se construye a partir de una muestra aleatoria del conjunto

de datos y considera un subconjunto aleatorio de características en cada división, lo que reduce la correlación entre los árboles y mejora la precisión general del modelo (Attanasi & Coburn).

CAPÍTULO III ESTADO DEL ARTE

En este capítulo se llevó a cabo una revisión sistemática y exhaustiva de la literatura relacionada con la detección y localización de tumores cerebrales primarios, específicamente gliomas. El proceso se estructuró en cuatro etapas: planificación, desarrollo, obtención de resultados y análisis, con el propósito de responder a la pregunta: ¿Cómo se puede desarrollar un sistema web, basado en Machine Learning, que permita predecir los hábitos financieros de las personas en Lima Metropolitana para mejorar su gestión financiera y apoyar su toma de decisiones? De un total de 52 artículos revisados, se seleccionaron 30 que sirvieron como base fundamental para esta tesis, los cuales se detallan en las páginas siguientes.

3.1. Metodología de la investigación

La presente revisión de literatura se llevó a cabo siguiendo los enfoques metodológicos propuestos por Villegas et al. (2021) y Alban y Mauricio (2019), los cuales se basan en la metodología desarrollada por Kitchenham et al. (2007). La revisión sistemática se llevó a cabo siguiendo un protocolo estructurado en cuatro fases principales: planificación, desarrollo de la búsqueda y selección, extracción de datos y síntesis de resultados. Durante la fase de planificación, se formulan las preguntas de investigación y se establece el procedimiento a seguir. En la etapa de desarrollo, se seleccionan los estudios primarios aplicando criterios de inclusión y exclusión. Posteriormente, la fase de resultados presenta los datos estadísticos y el análisis de los estudios elegidos, mientras que la etapa final de análisis permite responder a las interrogantes definidas al inicio del proceso.

3.2. Planeamiento

Para guiar la búsqueda y el análisis de la literatura, se formularon las siguientes preguntas de investigación:

Q1: ¿Qué algoritmos de Machine Learning se aplican en la predicción de hábitos financieros?

Q2: ¿Cómo se integran CRISP-DM y Scrum en proyectos de sistemas predictivos?

Q3: ¿Qué arquitecturas y tecnologías se utilizan en sistemas web para finanzas personales?

Q4: ¿Cuáles son los desafíos clave en la implementación de sistemas predictivos financieros en contextos locales?

Se consultaron diversas bases de datos académicas incluyendo El período de tiempo considerado para la publicación de los documentos abarcó desde 2020 a 2025, para asegurar la actualidad y relevancia de las fuentes, se realizó una búsqueda sistemática en bases de datos científicas como Scopus y Web of Science (WoS), utilizando los campos "TITLE-ABS-KEY" en Scopus y "TOPIC" en WoS.

Se diseñaron cadenas de búsqueda que combinan términos clave relacionados con machine learning, hábitos financieros, sistemas web, CRISP-DM y Scrum, usando operadores booleanos (AND, OR). Las búsquedas se limitaron a publicaciones entre 2020 y 2025 para garantizar su actualidad.

("financial habits" OR "financial behavior" OR "personal finance") AND ("machine learning" OR "predictive models" OR "classification" OR "supervised learning" OR "CRISP-DM" OR "data mining methodology") AND ("web system" OR "web application" OR "software development" OR "Scrum" OR "agile development")

Tabla 1*Sintaxis de consulta utilizada en la búsqueda*

Fuente	Número de resultados encontrados	Sintaxis de consulta
Scopus	13	(TITLE-ABS-KEY ("machine learning" AND "personal finance") AND TITLE-ABS-KEY ("financial habits" OR "financial behavior" OR "personal finance")) AND PUBYEAR > 2019 AND PUBYEAR < 2026 AND (LIMIT-TO (OA , "all"))
	3	(TITLE-ABS-KEY ("CRISP-DM" AND "SCRUM") AND TITLE-ABS-KEY ("MACHINE LEARNING"))
	15	TITLE-ABS-KEY (("predictive systems" OR "machine learning" OR "financial forecasting") AND ("implementation challenges" OR "barriers" OR "difficulties" OR "limitations") AND ("local context" OR "developing countries" OR "Latin America" OR "emerging markets" OR "financial inclusion"))
WoS	21	TS= ("financial habits" OR "financial behavior" OR "personal finance") AND ("machine learning" OR "predictive models" OR "classification" OR "supervised learning" OR "CRISP-DM" OR "data mining methodology") AND (2020-2025)

Fuente. Elaboración propia

3.2.1. Criterios de Inclusión y Exclusión

Se establecieron los siguientes criterios para la selección de documentos:

Criterios de Inclusión:

- Artículos de revistas científicas, actas de congresos y tesis de maestría o doctorado.
- Documentos publicados en inglés o español.

- Estudios que describan el desarrollo o implementación de sistemas predictivos de hábitos financieros.
- Investigaciones que aborden la aplicación de Machine Learning para predecir comportamientos o decisiones financieras.
- Estudios que discutan la integración o adaptación de CRISP-DM y/o Scrum en proyectos de ciencia de datos o desarrollo de software predictivo.

Criterios de Exclusión:

- Artículos de blog, noticias, white papers comerciales o publicaciones no académicas.
- Estudios que no involucren Machine Learning o predicción.
- Trabajos que se centren exclusivamente en la gestión de inversiones complejas o mercados de capitales sin relevancia para las finanzas personales.
- Estudios no relacionados con el desarrollo de sistemas o plataformas web.

3.3. Desarrollo

3.3.1. Proceso de Selección de Estudios

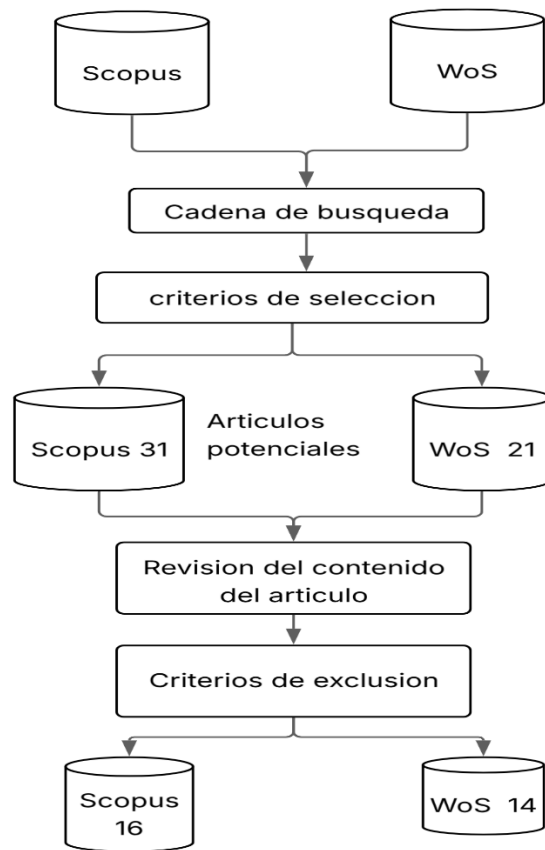
El proceso de selección se realizó en dos fases:

- a) Fase de Criba por Título y Resumen: Los artículos obtenidos de la búsqueda inicial (52) fueron revisados por título y resumen para descartar aquellos que claramente no cumplían con los criterios de inclusión.
- b) Fase de Lectura Completa: Los documentos preseleccionados fueron descargados y leídos en su totalidad para confirmar su relevancia y extraer la información pertinente. En esta fase, se identificaron y seleccionaron 30 artículos que formaron la base de esta revisión sistemática.

El proceso de investigación se ilustra en la figura 3. La revisión sistemática dio como resultado un total de 52 documentos, de los cuales, tras aplicar los criterios de inclusión y exclusión, se seleccionaron 30 artículos.

Figura 5

Proceso de investigación



3.4. Resultados

A continuación, se presenta una síntesis de los hallazgos más relevantes, organizados según las preguntas de investigación formuladas.

Tabla 2

Artículos potenciales y seleccionados

Fuente	Artículos potenciales	Artículos seleccionados
Scopus	31	16
WoS	21	14
Total	52	30

Fuente. Elaboración propia

El total de los artículos potenciales incluye a su vez los artículos que se encontraron de manera repetida en ambas fuentes.

En la siguiente tabla se muestra la información más relevante de cada uno de los artículos seleccionados, asimismo el identificador que se le asignó a cada uno de ellos.

Tabla 3

Artículos seleccionados

Id	Title	Year	Journal	Nº citas
A01	Unlocking the neural mechanisms of consumer loan evaluations: an fNIRS and ML-based consumer neuroscience study	2024	Front Hum Neurosci	5
A02	Machine Learning for Ischemic Heart Disease Diagnosis Aided by Evolutionary Computing	2021	Bionics Biomech	12
A03	Reject inference methods in credit scoring	2021	J Appl Stat	20
A04	Machine Learning Based Diagnostic Paradigm in Viral and Non-Viral Hepatocellular Carcinoma	2024	IEEE Access	4
A05	Ensemble learning based automatic detection of tuberculosis in chest X-ray images using hybrid feature descriptors	2021	Phys Eng Sci Med	128
A06	Incorporating deep learning and news topic modeling for forecasting pork prices: The case of South Korea	2020	Agriculture	40
A07	Modeling of euro stoxx 50 index price returns based on industrial production surprises: basic and machine learning approach	2021	Journal of Entrepreneurship and Sustainability Issues	0
A08	Innovation and sdgs through social media analysis: Messages from fintech firms	2021	Journal of Open Innovation: Technology, Market, and Complexity	56
A09	Automatic detection of relevant information, predictions and forecasts in financial news through topic modelling with Latent Dirichlet Allocation	2023	Computer Science	24

A10	CP-AnemiC: A conjunctival pallor dataset and benchmark for anemia detection in children	2023	Medicine in Novel Technology and Devices	17
A11	Machine Learning Application Development Guidelines Using CRISP-DM and Scrum Concept	2023	IEEE International Conference on Data and Software Engineering (ICoDSE)	0
A12	The role of digital technology in communication and information flow in the Australian Superannuation industry	2023	Australian Journal of Management	0
A13	Identifying Hidden Factors Associated with Household Emergency Fund Holdings: A Machine Learning Application	2024	Mathematics	2
A14	Machine learning-based predictive modeling of anxiety and depressive symptoms during 8 months of the covid-19 global pandemic: Repeated cross-sectional survey study	2021	JMIR Ment Health	32
A15	Intelligent System and Framework for Integrating Machine Learning with Software Development for Predictive Banking Logistics	2024	Computational Logistics	0
A16	Detecting Potential Investors in Crypto Assets: Insights from Machine Learning Models and Explainable AI	2025	Information	0
A17	A Time Series Approach to Forecasting Financial Indicators in the Wholesale and Retail Trade	2025	World	2
A18	The development of an action priority matrix and technology roadmap for the implementation of data-driven and machine-learning-based predictive maintenance in the south african railway industry	2023	The South African Journal of Industrial Engineerin	2
A19	A Machine Learning Method to improve Supplier Delivery Appointments in Supply Chain Industries: A Case Study	2025	Brazilian Journal of Operations & Production Management	1
A20	Development of portable and robust cataract detection and grading	2022	Multimed Tools Appl	16

	system by analyzing multiple texture features for Tele-Ophthalmology			
A21	Acceptability of linking individual credit, financial, and public records data to healthcare records for suicide risk machine learning models	2024	JAMIA Open	0
A22	A New Method for Graph-Based Representation of Text in Natural Language Processing	2023	Electronics	8
A23	MultiModalGraphSearch: Intelligent Massive-Scale SubGraph Discovery for Multi-Category Financial Pattern Mining	2025	Journals & Magazines	0
A24	Enhancing security in financial transactions: a novel blockchain-based federated learning framework for detecting counterfeit data in fintech	2024	PeerJ Computer Science	0
A25	Explainable AI for Psychological Profiling from Behavioral Data: An Application to Big Five Personality Predictions from Financial Transaction Records	2022	Information	32
A26	Professionalisation of financial planning in Australia, Canada, and South Africa	2024	Australian Journal of Management	3
A27	ChatGPT, Help! I Am in Financial Trouble	2024	Risk Financial Manag	6
A28	A systemic approach to machine learning project management	2024	IEEE Engineering Management Review	1
A29	Smartphone-Based Cost-Effective Pavement Performance Model Development Using a Machine Learning Technique with Limited Data	2024	Infrastructures	3
A30	A hybrid machine learning framework by incorporating categorical boosting and manifold learning for financial analysis	2025	Intelligent Systems with Applications	0

Fuente. Elaboración propia

3.5. Análisis

En esta sección se responde las interrogantes planteadas dentro del proceso de planificación.

3.5.1. ¿Qué algoritmos de Machine Learning se aplican en la predicción de hábitos financieros?

Tabla 4

Modelos de Machine Learning aplicados a hábitos financieros

Id	Modelo / Algoritmo	Descripción	Fuente(s)
MD01	Árboles de decisión (Decision Trees)	Modelo basado en reglas de decisión que permite segmentar clientes o usuarios según sus características financieras.	Heo et al. (2024); Ehrhardt et al. (2021)
MD02	Random Forest	Conjunto de árboles de decisión que mejora la precisión en predicciones financieras, especialmente en clasificación de perfiles.	Heo et al. (2024); Ramon et al. (2022)
MD03	XGBoost	Algoritmo de boosting que ha demostrado alta eficacia en predicción de hábitos financieros, gastos e inversión.	Zhao & Zhao (2025)
MD04	Redes neuronales (Deep Neural Networks)	Redes profundas para clasificar patrones complejos en datos financieros, como préstamos o comportamientos de gasto.	Ali et al. (2024); Ramon et al. (2022)
MD05	LSTM (Long Short-Term Memory)	Modelo de red neuronal recurrente que captura relaciones temporales en series de datos financieros.	Jagric et al. (2025); Ayaz et al. (2023)
MD06	SVM (Máquinas de Vectores de Soporte)	Clasificador robusto que permite diferenciar entre hábitos financieros estables e inestables según datos históricos.	Gaspareniene et al. (2021)
MD07	Clustering (K-means, DBSCAN, etc.)	Utilizado para segmentar usuarios por características de comportamiento financiero, sin necesidad de etiquetas previas.	Jencová et al. (2025); Jagric et al. (2025)
MD08	Modelos híbridos (ensemble)	Combinación de múltiples algoritmos para mejorar la precisión y robustez de las predicciones en contextos financieros.	Appiahene et al. (2021); Nethamba & Grobbelaar (2023); Ali et al. (2024)

MD09	CNN (Redes Neuronales Convolucionales)	Aunque tradicionalmente usadas en imágenes, algunas investigaciones han usado CNN para analizar series financieras complejas.	Ayaz et al. (2023)
MD10	Explainable AI (XAI) con modelos complejos	Aplicación de modelos interpretables como SHAP o LIME para explicar predicciones financieras a partir de transacciones.	Ramon et al. (2022); Jagric et al. (2025)

Fuente. Elaboración propia

3.5.2.¿Cómo se integran CRISP-DM y Scrum en proyectos de sistemas predictivos?

Tabla 5

Integración de CRISP-DM y Scrum

Id	Integración CRISP-DM y/o Scrum	Descripción	Fuente(s)
INT01	CRISP-DM + Scrum	Se propone una integración donde CRISP-DM estructura las etapas de minería de datos y Scrum organiza el trabajo iterativo y colaborativo en desarrollo web.	Pathak et al. (2023); Hurtado & Ayora (2024)
INT02	CRISP-DM con enfoque iterativo	El desarrollo de sistemas predictivos sigue las fases del modelo CRISP-DM, iterando entre fases según resultados de evaluación y ajustes de modelos.	Hayat Suhendar & Widayani (2023)
INT03	Scrum aplicado a proyectos de ML	Scrum se adapta para gestionar proyectos de Machine Learning donde las tareas técnicas como modelado, validación y despliegue se planifican en sprints.	Uysal & Akturk (2024); Hurtado & Ayora (2024)
INT04	Enfoque ágil con fases de CRISP-DM	El ciclo de vida de CRISP-DM (comprensión del negocio, comprensión de datos, preparación, modelado, evaluación y despliegue) se inserta en el marco ágil.	Pathak et al. (2023); Hayat Suhendar & Widayani (2023)
INT05	CRISP-DM adaptado a fintech predictivo	Se adopta CRISP-DM en contexto financiero para estructurar el análisis predictivo y se complementa con tareas ágiles para diseño y pruebas del sistema.	Hurtado & Ayora (2024)

Fuente. Elaboración propia

3.5.3.¿Qué arquitecturas y tecnologías se utilizan en sistemas web para finanzas personales?

Tabla 6

Arquitecturas y tecnologías usadas en sistemas web financieros

Id	Arquitectura / Tecnología empleada	Descripción	Fuente(s)
TEC01	Arquitectura en capas (Layered Architecture)	Organiza la solución en capas: presentación (frontend), lógica de negocio (backend) y acceso a datos. Favorece la mantenibilidad del sistema.	Hurtado & Ayora (2024); Uysal & Akturk (2024)
TEC02	Arquitectura basada en servicios (SOA / APIs)	Utiliza servicios web para integrar módulos predictivos con la interfaz de usuario o con bases de datos externas.	Pathak et al. (2023); Ramon et al. (2022)
TEC03	Arquitectura cliente-servidor	Distribuye tareas entre el servidor que procesa datos y el cliente que interactúa con el usuario. Común en sistemas financieros web.	Heo et al. (2024); Franco-Riquelme & Rubalcaba (2021)
TEC04	Microservicios	Permite escalar componentes independientes como módulos de predicción, visualización o gestión de usuarios.	Zhao & Zhao (2025); Jagric et al. (2025)
TEC05	Lenguajes de programación (Python, JavaScript)	Python para modelos predictivos con librerías como scikit-learn, TensorFlow. JavaScript (React.js, Vue.js) en el frontend del sistema web.	Hurtado & Ayora (2024); Pathak et al. (2023)
TEC06	Bases de datos SQL y NoSQL	SQL (PostgreSQL, MySQL) para transacciones estructuradas y NoSQL (MongoDB) para almacenar registros de comportamiento financiero y logs.	Hurtado & Ayora (2024); Ramon et al. (2022)
TEC07	Frameworks web (Django, Flask, Spring Boot)	Facilitan el desarrollo de sistemas web con conexión directa a modelos predictivos y gestión de usuarios.	Uysal & Akturk (2024); Hayat Suhendar & Widayani (2023)
TEC08	Herramientas MLOps (MLflow, Docker, Kubernetes)	Automatizan el entrenamiento, despliegue y monitoreo de modelos predictivos financieros dentro de entornos web.	Uysal & Akturk (2024); Jagric et al. (2025)

TEC09	Infraestructura en la nube (AWS, GCP, Azure)	Uso de servicios cloud para alojar el sistema, entrenar modelos y asegurar disponibilidad para usuarios en línea.	Hurtado & Ayora (2024); Zhao & Zhao (2025)
--------------	--	---	--

Fuente. Elaboración propia

3.5.4. ¿Cuáles son los desafíos clave en la implementación de sistemas predictivos financieros en contextos locales?

Tabla 7

Desafíos en la implementación local de sistemas predictivos financieros

Id	Desafío identificado	Descripción	Fuente(s)
DES01	Escasez de datos locales confiables	La falta de registros financieros digitales o bases de datos limpias dificulta el entrenamiento de modelos precisos.	Heo et al. (2024); Ramon et al. (2022)
DES02	Privacidad y regulación de datos	Las restricciones legales para acceder a información financiera personal pueden limitar la recopilación y uso de datos sensibles.	Penfold et al. (2024); Rabbani et al. (2024)
DES03	Falta de alfabetización digital y financiera	Los usuarios locales pueden no comprender el sistema, sus recomendaciones o su utilidad, afectando la adopción y el impacto real.	Nethamba & Grobbelaar (2023); Schlosky et al. (2024)
DES04	Limitaciones tecnológicas e infraestructurales	Infraestructura tecnológica débil (internet, servidores, nube) limita la disponibilidad del sistema predictivo.	Hayat Suhendar & Widayani (2023); Uysal & Akturk (2024)
DES05	Contextualización de modelos	Modelos entrenados en contextos extranjeros no se ajustan bien a la realidad financiera de Lima u otras ciudades locales.	Zhao & Zhao (2025); Jagric et al. (2025)
DES06	Alineamiento con prácticas ágiles y metodologías	La adopción de CRISP-DM y Scrum requiere adaptación a equipos con poco entrenamiento en estas metodologías.	Pathak et al. (2023); Hurtado & Ayora (2024)
DES07	Sostenibilidad del sistema	Los sistemas desarrollados muchas veces no cuentan con financiamiento o mantenimiento continuo, lo que pone en riesgo su operación a largo plazo.	Franco-Riquelme & Rubalcaba (2021); Ramon et al. (2022)

Fuente. Elaboración propia

3.6. Conclusiones de la revisión sistemática de la literatura

67

En el presente estudio se identificaron inicialmente 52 artículos científicos potenciales extraídos de las bases de datos Scopus (31) y Web of Science (WoS) (21), correspondientes al periodo 2020–2025. Tras aplicar los criterios de inclusión y exclusión definidos, se seleccionaron únicamente aquellos trabajos relevantes para la temática de predicción de hábitos financieros mediante sistemas web basados en machine learning, empleando las metodologías CRISP-DM y Scrum. Se eliminaron registros duplicados y estudios no relacionados directamente con sistemas predictivos financieros o metodologías de desarrollo.

Respecto a los algoritmos de Machine Learning aplicados en la predicción de hábitos financieros, se identificó el uso predominante de algoritmos como árboles de decisión, random forest, regresión logística, redes neuronales artificiales, SVM y XGBoost, siendo elegidos por su capacidad de clasificación, precisión y adaptabilidad a grandes volúmenes de datos transaccionales.

En cuanto a la integración de CRISP-DM y Scrum, se encontró que dicha combinación metodológica permite estructurar el desarrollo iterativo de sistemas predictivos financieros, desde la comprensión del dominio hasta la evaluación y despliegue, destacando el valor del enfoque ágil para incorporar modelos de aprendizaje automático en ciclos incrementales.

Asimismo, se determinaron las arquitecturas y tecnologías más utilizadas en el desarrollo de sistemas web financieros, resaltando las arquitecturas en capas y microservicios, así como el uso de tecnologías como Python, Django, Flask, React.js, bases de datos SQL y NoSQL, y servicios en la nube (AWS, Azure). Estas tecnologías facilitan tanto la implementación del sistema como la integración de los modelos de predicción.

Finalmente, se identificaron los desafíos clave en contextos locales, entre los que destacan: la escasez de datos financieros confiables, la privacidad de la información, las limitaciones tecnológicas, la falta de alfabetización digital, y la necesidad de adaptar modelos globales a realidades económicas locales. Estos factores condicionan significativamente la efectividad y sostenibilidad de los sistemas predictivos financieros aplicados en regiones como Lima Metropolitana.

CAPÍTULO IV: MODELO DE MACHINE LEARNING PARA LA PREDICCIÓN DE HÁBITOS FINANCIEROS Y PROPENSIÓN AL AHORRO

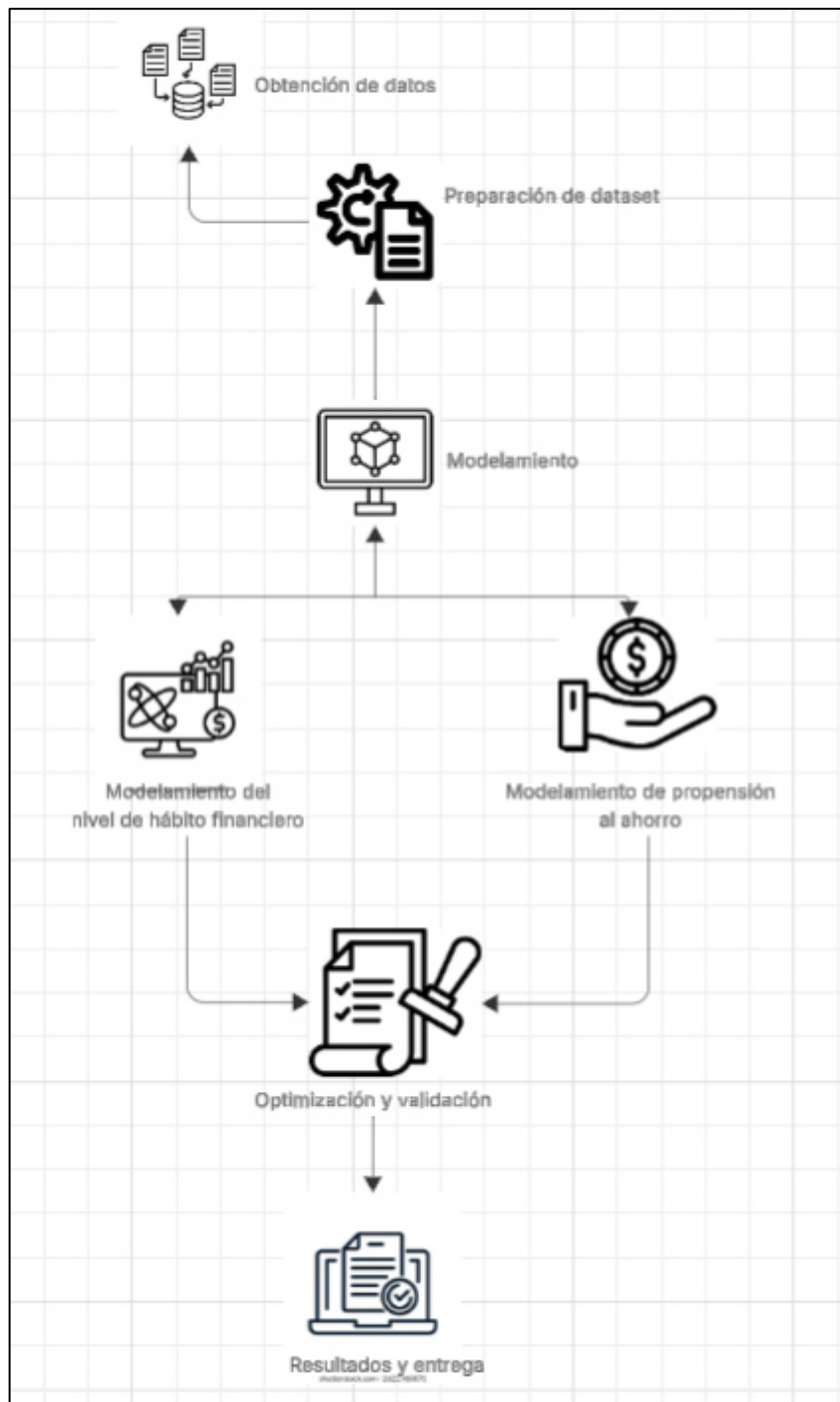
En este capítulo, se presenta el desarrollo de un modelo de Machine Learning para la predicción de hábitos financieros y propensión al ahorro en personas de Lima Metropolitana, siguiendo una metodología rigurosa que integra las mejores prácticas de CRISP-DM y las metodologías ágiles de Scrum. El sistema busca proporcionar insights accionables y personalizados que permitan a los usuarios comprender y optimizar sus patrones de comportamiento financiero de manera efectiva y accesible.

4.1. Aporte

Se propone un modelo de Machine Learning para la predicción de hábitos financieros y propensión al ahorro, que permite identificar patrones de comportamiento y estimar la disposición al ahorro a partir de datos transaccionales y sociodemográficos de individuos de Lima Metropolitana. Su propósito es realizar dichas predicciones de una manera más precisa, automatizada y objetiva, en beneficio de colaborar con el usuario final gracias a una evaluación previa y personalizada. Los componentes principales incluyen el preprocesamiento y la ingeniería de características de los datos, junto con modelos de aprendizaje supervisado basados en algoritmos como XGBoost, optimizados mediante técnicas avanzadas como SMOTE y búsqueda exhaustiva de Hiperparámetros (GridSearchCV y RandomizedSearchCV), desarrollados bajo una metodología rigurosa que integra CRISP-DM y Scrum para garantizar robustez, adaptabilidad y repetibilidad.

Figura 6 Método general de predicción de hábitos financieros y propensión al ahorro

Arquitectura de PostgreSQL



Fuente. Elaboración propia

En la figura 4.1, El proceso metodológico para predecir hábitos financieros y propensión al ahorro es un flujo sistemático que comienza con la obtención de datos transaccionales y sociodemográficos de individuos de Lima Metropolitana, seguido de una preparación exhaustiva del dataset para garantizar su calidad. Posteriormente, se desarrolla un modelamiento que integra técnicas avanzadas de Machine Learning, específicamente algoritmos como XGBoost, optimizados mediante estrategias como SMOTE y búsqueda de hiperparámetros, con el objetivo de analizar el nivel de hábito financiero y la propensión al ahorro. El proceso incluye una fase crítica de optimización y validación, utilizando metodologías como CRISP-DM y Scrum, que permite refinar el modelo para obtener predicciones precisas y personalizadas. El resultado final es un sistema automatizado que proporciona insights accionables, permitiendo a los usuarios comprender y potencialmente mejorar sus comportamientos financieros de manera objetiva y accesible.

4.2. Metodología CRISP-DM para el Desarrollo del Modelo

4.2.1. Comprensión del Negocio

- a) **Objetivo del Negocio:** El objetivo principal es crear una herramienta web ("FinSight") que funcione como un asistente financiero personal. Esta herramienta debe permitir a los usuarios de Lima Metropolitana obtener predicciones sobre sus propios patrones de gasto y ahorro, ayudándoles a tomar conciencia de su situación y a tomar decisiones más informadas. Se busca fomentar una mejor gestión financiera personal y contribuir a la educación financiera en el contexto local.
- b) **Objetivos de Minería de Datos:** Para lograr el objetivo de negocio, se definen dos tareas predictivas específicas:
 - **Clasificación Multiclase:** Desarrollar un modelo capaz de asignar a cada usuario un nivel de hábito financiero en una escala ordinal (de 1=Muy Malo a 5=Muy Bueno). Esto permitirá una segmentación detallada del comportamiento general.
 - **Clasificación Binaria:** Desarrollar un modelo para predecir la propensión al ahorro del usuario (0=Baja propensión, 1=Alta propensión). Esto facilitará la identificación de perfiles específicos para recomendaciones o intervenciones.
- c) **Criterios de Éxito (Conceptuales):** El éxito del modelo se medirá por su capacidad predictiva (evaluada en fases posteriores con métricas como

Accuracy, F1-Score) y por la **utilidad percibida** por el usuario final al interactuar con el sistema web (evaluada conceptualmente a través de encuestas o feedback).

- d) **Plan del Proyecto:** Se establece un plan inicial que integra las fases de CRISP-DM dentro de un marco de desarrollo ágil (Scrum). Las fases de Comprensión, Preparación, Modelado y Evaluación de CRISP-DM se abordarán en **Sprints específicos**, permitiendo un desarrollo iterativo e incremental del modelo y del sistema web (cuya ejecución práctica se detalla en el Capítulo V).

4.2.2. Comprensión de los datos

El proceso de análisis inicia con la recolección inicial de datos, etapa en la que se reconoce la necesidad de obtener información representativa de la población de Lima Metropolitana. Aunque lo ideal sería trabajar con datos transaccionales anonimizados o encuestas detalladas, las restricciones de privacidad y acceso obligan a emplear datos sintéticos como alternativa inicial. Estos datos se construyen de manera que simulen con realismo las principales características sociodemográficas y financieras, tales como edad, género, nivel educativo, ingresos, gastos, ahorros, deudas y uso de productos financieros.

Posteriormente, se desarrolla una descripción detallada del conjunto de datos, examinando el tipo de variables disponibles y analizando su comportamiento mediante distribuciones de frecuencia, gráficos como histogramas y boxplots, así como estadísticas básicas que permiten comprender tendencias generales y niveles de variabilidad.

En la etapa de exploración, se investigan las relaciones preliminares entre variables a través de matrices de correlación para datos numéricos y tablas de contingencia para datos categóricos, con el fin de identificar predictores relevantes para los hábitos financieros o la propensión al ahorro.

Finalmente, se verifica la calidad del conjunto de datos, lo que, en el caso de datos sintéticos, implica revisar la coherencia lógica de los valores generados; y, en un escenario de datos reales, supondría gestionar valores faltantes, detectar outliers y corregir inconsistencias que puedan comprometer la precisión de los modelos predictivos.

4.2.3. Preparación de los datos

Esta es una fase fundamental donde los datos brutos se transforman para ser utilizados por los algoritmos de Machine Learning.

- a) **Selección de Datos:** Se seleccionan las **variables finales** que se incluirán en el modelado, basándose en los hallazgos de la exploración y el conocimiento del dominio (finanzas personales). Se podrían excluir variables redundantes o con bajo poder predictivo.
- b) **Limpieza de Datos:** Se aplican las estrategias definidas para manejar datos faltantes (tales como imputación con la media/mediana, eliminación de registros) y outliers (como winsorización, eliminación o transformación logarítmica). Se asegura la consistencia en los formatos (por ejemplo, fechas, monedas).
- c) **Construcción de Datos (Ingeniería de Características):**
 - **Transformación de Variables Existentes:** Se convierten variables categóricas a formato numérico. Para variables nominales como 'Género' o 'Nivel de Educación', se aplica One-Hot Encoding (creando variables dummy). Para variables ordinales como 'Frecuencia de uso de tarjeta', se aplica mapeo a números enteros (1 a 5). Se transforman variables monetarias absolutas ('Ahorro mensual', 'Endeudamiento mensual') en variables relativas al ingreso (porcentajes), lo cual normaliza la información y la hace comparable entre individuos con diferentes niveles de ingreso.
 - **Creación de Variables Objetivo (Aplicación de Reglas Heurísticas):** Este es un paso crítico donde se traduce el conocimiento del negocio en variables medibles.
 - ✓ Nivel de Hábito Financiero (Score_H → Ordinal 1-5): Se define una fórmula heurística que pondera factores positivos (como alto nivel de ahorro porcentual) y negativos (como alto nivel de endeudamiento porcentual y alta frecuencia de uso de crédito) . El score crudo resultante (SCORE_H) se transforma en una variable ordinal balanceada dividiendo la distribución en quintiles (cada nivel representa aproximadamente el 20% de la muestra).

- ✓ **Propensión al Ahorro (Score_A → Binaria 0-1):** Se define otra fórmula heurística que combina positivamente el nivel de ahorro porcentual y la disciplina de presupuestación, y negativamente el nivel de endeudamiento. El score crudo (SCORE_A) se convierte en una variable binaria balanceada utilizando la mediana como punto de corte (aproximadamente 50% en clase 0 y 50% en clase 1).

d) **Integración y Formateo:** Se consolida el dataset final, incluyendo solo las variables predictoras transformadas (dummies, ordinales, porcentuales) y las dos variables objetivo creadas ('Nivel de Hábito Financiero' y 'Propensión al Ahorro'). Se eliminan variables intermedias usadas solo para el cálculo (como SCORE_H, SCORE_A). Se guarda el dataset en un formato estándar (CSV) listo para la fase de modelado.

4.2.4. Modelado

En esta fase se seleccionan, construyen y evalúan preliminarmente los modelos predictivos.

- **Selección de Técnicas de Modelado:** Se elige la familia de algoritmos de clasificación supervisada. Se seleccionan múltiples algoritmos para comparar su rendimiento en este problema específico:
 - ✓ *Lineales:* Regresión Logística (buena interpretabilidad).
 - ✓ *Basados en Árboles:* Árbol de Decisión (interpretable), Random Forest (robusto), Gradient Boosting, **XGBoost** (alto rendimiento).
 - ✓ *Otros:* K-Nearest Neighbors (KNN), Support Vector Machine (SVM).
- **Generación del Diseño de Prueba:** Se define un protocolo riguroso para evaluar los modelos de forma insesgada:
 - ✓ **División Train-Test:** Separar el dataset preparado en un conjunto de entrenamiento (80%) y un conjunto de prueba (20%). Es crucial que la división sea estratificada por la variable objetivo para mantener la proporción de clases en ambos conjuntos. La semilla aleatoria (random_state) se fija para asegurar la reproducibilidad de los resultados.

- ✓ **Validación Cruzada:** Utilizar k-Fold Cross-Validation (con k=5) sobre el *conjunto de entrenamiento* para obtener una estimación más robusta del rendimiento del modelo y para la optimización de hiperparámetros. La validación cruzada también debe ser estratificada.
- **Construcción del Modelo:**
 - ✓ **Entrenamiento Baseline:** Entrenar cada algoritmo seleccionado con sus hiperparámetros por defecto usando el conjunto de entrenamiento.
 - ✓ **Manejo de Desbalance:** Aunque las variables objetivo se diseñaron para ser balanceadas, se contempla el uso de técnicas como SMOTE (Synthetic Minority Over-sampling Technique) en el *conjunto de entrenamiento* si se detecta un desbalance residual significativo, especialmente antes de la optimización final.
 - ✓ **Optimización de Hiperparámetros:** Para los algoritmos más prometedores (como XGBoost), aplicar técnicas como GridSearchCV (búsqueda exhaustiva) o RandomizedSearchCV (búsqueda aleatoria, más eficiente para espacios grandes) utilizando la validación cruzada para encontrar la mejor combinación de hiperparámetros (tales como learning_rate, n_estimators, max_depth) que maximice la métrica de evaluación seleccionada, como Accuracy o F1-score.
- **Evaluación Preliminar (Durante Modelado):** Evaluar continuamente los modelos usando las métricas definidas (Accuracy, F1, etc.) sobre los *folds de validación* (en Cross-Validation) o un *conjunto de validación* separado (si se usa) para guiar el proceso de selección y optimización.

4.2.5. Evaluación

Esta fase se enfoca en evaluar qué tan bien el(los) modelo(s) final(es) cumple(n) los objetivos del negocio y decidir sobre su despliegue.

- **Evaluación de Resultados:** El modelo final optimizado (uno para cada variable objetivo) se evalúa rigurosamente utilizando el **conjunto de prueba**, que contiene datos que el modelo nunca ha visto antes. Se calculan todas las métricas relevantes:

- ✓ Para Variable 1 (Multiclase): Accuracy global, Precision/Recall/F1 por clase, promedios Macro y Ponderado, Matriz de Confusión .
- ✓ Para Variable 2 (Binaria): Accuracy, Precision, Recall, F1-Score, ROC-AUC, Matriz de Confusión .
- **Revisión del Proceso:** Se comparan los resultados obtenidos en el conjunto de prueba con los criterios de éxito definidos en la Comprensión del Negocio. Se analiza si el rendimiento es aceptable para el propósito de la aplicación (por ejemplo, ¿Un 80% de Accuracy es suficiente para dar recomendaciones financieras?). Se revisan las fases anteriores para entender las fortalezas y debilidades del modelo (por ejemplo, ¿Qué clases predice bien/mal? ¿Qué variables fueron más importantes?).
- **Determinación de Próximos Pasos:** Basado en la evaluación, se toma la decisión:
 - ✓ Desplegar: Si el modelo cumple los criterios.
 - ✓ Iterar: Si el modelo necesita mejoras (por ejemplo, volver a la Preparación de Datos para crear nuevas características, probar arquitecturas de modelos más complejas, recolectar más datos).
 - ✓ Abandonar: Si el modelo consistentemente no alcanza un rendimiento útil.

4.2.6. Despliegue del modelo

Esta fase describe cómo el modelo validado se pondrá en uso.

- **Plan de Despliegue:** Se define la estrategia para integrar los modelos entrenados (como archivos .pkl o .joblib) en el backend del sistema web "FinSight". Se creará una API REST que recibirá los datos del usuario desde el frontend, los preprocesará de la misma manera que en el entrenamiento, los enviará a los modelos cargados para obtener las predicciones (Nivel de Hábito y Propensión al Ahorro) y devolverá estas predicciones al frontend para ser mostradas al usuario .
- **Plan de Monitoreo y Mantenimiento:** Se establece un plan conceptual para monitorear el rendimiento del modelo una vez desplegado. Esto podría incluir el seguimiento de la distribución de las predicciones, la recopilación de feedback

del usuario (por ejemplo, a través de la encuesta de satisfacción) y, si es posible, la comparación con resultados reales a lo largo del tiempo. Se planifica el reentrenamiento periódico del modelo con nuevos datos para mantener su precisión y adaptarlo a posibles cambios en los patrones de comportamiento financiero.

- **Informe Final y Revisión del Proyecto:** Se documenta todo el proceso CRISP-DM, desde los objetivos iniciales hasta los resultados de la evaluación y el plan de despliegue. Se realiza una revisión final del proyecto para asegurar que se cumplieron los objetivos y se extraen lecciones aprendidas para futuros proyectos de ciencia de datos.

4.3. Etiquetado de datos y reglas heurísticas

Esta sección detalla los componentes conceptuales clave utilizados durante la fase de Preparación de Datos de CRISP-DM, específicamente en la Construcción de Datos (Ingeniería de Características). Define tanto las variables de entrada consideradas como las reglas heurísticas diseñadas para crear las variables objetivo iniciales.

4.3.1. Variable de análisis

Se define un conjunto inicial de variables predictoras (features) que se consideran relevantes para modelar los hábitos financieros y la propensión al ahorro, basándose en el conocimiento del dominio y la literatura. Estas se agrupan en:

- **Demográficas:**
 - Edad: Factor relevante en el ciclo de vida financiero (por ejemplo, jóvenes pueden tener diferentes patrones que adultos mayores). Rango conceptual: 18-65+ años.
 - Género: Puede influir en comportamientos financieros debido a factores sociales o económicos. Categorías: Masculino, Femenino, Otro.
 - Nivel de Educación: Se correlaciona a menudo con el nivel de ingresos y la comprensión financiera. Categorías: Desde Primaria completa hasta Universitaria completa, incluyendo estados intermedios .

– **Económicas:**

- **Ingreso Mensual:** Variable fundamental que determina la capacidad de ahorro y gasto . Rango conceptual amplio (800 - 8000+ soles) para cubrir diferentes estratos socioeconómicos en Lima Metropolitana.
- **Otras Variables Financieras Crudas:** Se incluyen conceptualmente variables como 'Ahorro mensual aproximado', 'Endeudamiento mensual aproximado', 'Monto destinado al ocio mensual aproximado', 'Número de tarjetas de crédito', 'Presupuesta sus gastos regularmente', 'Frecuencia de uso de tarjeta de crédito', 'Suele pagar solo el monto mínimo', que serán la base para las transformaciones y la creación de variables objetivo.

4.3.2. Criterios de etiquetado

Para crear las variables objetivo ('Nivel de Hábito Financiero' y 'Propensión al Ahorro'), se definen **reglas heurísticas** basadas en umbrales porcentuales del ingreso mensual. Estos umbrales se inspiran conceptualmente en recomendaciones financieras generales o estándares de evaluación de riesgo crediticio, aunque adaptados al contexto local.

- **Nivel de Ahorro (% del Ingreso):** Se categoriza la capacidad de ahorro en 5 niveles ordinales.
 - Bajo: 0% - 8%
 - Medio-Bajo: >8% - 12%
 - Medio: >12% - 20%
 - Alto: >20% - 30%
 - Muy Alto: >30%
- **Nivel de Endeudamiento (% del Ingreso):** Se categoriza la carga de deuda también en 5 niveles ordinales.
 - Bajo: 0% - 10%
 - Medio-Bajo: >10% - 15%
 - Medio: >15% - 30%
 - Alto: >30% - 40%
 - Muy Alto: >40%

- **Otros Criterios (Variables Cualitativas Transformadas):** Se incorporan variables de comportamiento clave, transformadas a escalas ordinales o binarias:
 - Presupuestación de Gastos: Se transforma la frecuencia (como 'Nunca' a 'Muy Frecuente') en una escala numérica (1 a 5). Una mayor frecuencia indica mejor hábito.
 - Uso de Tarjeta de Crédito: La frecuencia también se transforma a escala numérica (1 a 5). Una frecuencia muy alta puede indicar dependencia del crédito o falta de liquidez (hábito negativo).
 - Pago Mínimo de Tarjeta: La frecuencia de pagar solo el mínimo se transforma a escala numérica (1 a 5). Pagar solo el mínimo frecuentemente es un indicador negativo.
 - Nivel de Gasto en Ocio: Se categoriza conceptualmente (como Bajo a Muy Alto) como un indicador del control de gastos discrecionales.

Estas variables transformadas y categorizadas son los **insumos directos** para las **fórmulas heurísticas** (SCORE_H y SCORE_A) definidas en la sección 4.2.3 para crear las variables objetivo finales ('Nivel de Hábito Financiero' y 'Propensión al Ahorro').

4.4. Proceso de construcción del modelo de machine learning

Si bien las reglas heurísticas proporcionan una base inicial lógica para etiquetar los datos y crear las variables objetivo, un modelo de Machine Learning supervisado se propone como el componente central para la predicción final por varias razones conceptuales:

- **Objetivo:** El desarrollo del modelo de Machine Learning tiene como propósito superar las limitaciones propias de las heurísticas utilizadas en la fase inicial de etiquetado, ya que estas reglas, aunque útiles como punto de partida, representan simplificaciones que no siempre logran capturar la complejidad real del comportamiento financiero.
- **Método:** La meta del modelo es, por tanto, identificar patrones más profundos y sutiles directamente desde los datos, permitiendo una comprensión más precisa de los hábitos financieros de los usuarios. Para lograrlo, se adopta un enfoque de aprendizaje supervisado, mediante el cual el modelo aprende a relacionar las variables predictoras (correspondientes a características

demográficas y económicas previamente transformadas) con las categorías objetivo generadas a partir de las heurísticas. Si bien estas etiquetas sirven como referencia inicial, el modelo busca no solo replicar el proceso de clasificación, sino generalizarlo, aprovechando su capacidad para reconocer interacciones complejas y no lineales que las reglas fijas no pueden incorporar.

Ventajas del Enfoque ML:

El empleo de modelos de Machine Learning ofrece ventajas significativas frente a la aplicación de reglas heurísticas estáticas, especialmente en contextos donde los patrones de comportamiento son complejos y variables.

- Un modelo bien entrenado posee la capacidad de generalizar a nuevos usuarios cuyos datos no formaron parte del conjunto de entrenamiento, lo que permite obtener predicciones más robustas ante combinaciones de factores no previstas por reglas fijas.
- Además, los algoritmos modernos (particularmente aquellos basados en árboles, como Random Forest o XGBoost) pueden identificar relaciones no lineales y capturar interacciones sutiles entre las variables predictoras, lo que resulta indispensable cuando el efecto de un factor, como el nivel educativo, varía según la edad o el ingreso.
- Esta capacidad de modelar estructuras complejas se traduce en un potencial aumento de la precisión en la clasificación de usuarios, incluso en escenarios ambiguos o casos límite donde las heurísticas suelen fallar.
- Finalmente, los modelos de aprendizaje automático presentan una alta adaptabilidad, ya que pueden ser reentrenados con nuevos datos para reflejar cambios en los hábitos financieros a lo largo del tiempo, una tarea mucho más compleja cuando se depende exclusivamente de reglas predefinidas y rígidas.

En esencia, el modelo de Machine Learning actúa como un refinamiento y una generalización del conocimiento experto encapsulado en las reglas heurísticas, buscando una solución predictiva más potente y adaptable basada en la evidencia de los datos. El proceso detallado de selección, entrenamiento y evaluación de este modelo se aborda conceptualmente en las fases de Modelado y Evaluación de CRISP-DM descritas anteriormente (4.2.4 y 4.2.5).

CAPÍTULO V: APORTE PRACTICO

En este capítulo, se expone el proceso de desarrollo del sistema web para la predicción de hábitos financieros y propensión al ahorro en personas de Lima metropolitana. Se detallan la arquitectura, los requisitos funcionales y no funcionales, el diseño de la base de datos, las pruebas realizadas, el funcionamiento del sistema y las limitaciones del aplicativo.

5.1 Introducción al Desarrollo

Este capítulo expone el proceso de desarrollo e implementación del sistema web "FinSight", la solución tecnológica diseñada para la predicción de hábitos financieros y propensión al ahorro en personas de Lima Metropolitana. El aporte práctico de esta tesis se articula mediante una metodología híbrida que combina el marco ágil Scrum para la gestión del proyecto y la metodología CRISP-DM para estructurar el desarrollo de los modelos de Machine Learning. El desarrollo se documenta a través de los Sprints ejecutados, detallando la implementación de la arquitectura, la preparación de datos, el proceso de entrenamiento y evaluación de los modelos predictivos (cuyos resultados detallados se presentarán en el Capítulo VI), y la construcción de la interfaz de usuario final.

5.2 Arquitectura del Sistema "FinSight"

La base del aporte práctico es una arquitectura de sistema web moderna, escalable y orientada a servicios, diseñada para soportar los modelos predictivos y garantizar una experiencia de usuario fluida. El sistema, denominado "FinSight", se estructura en tres vistas arquitectónicas clave.

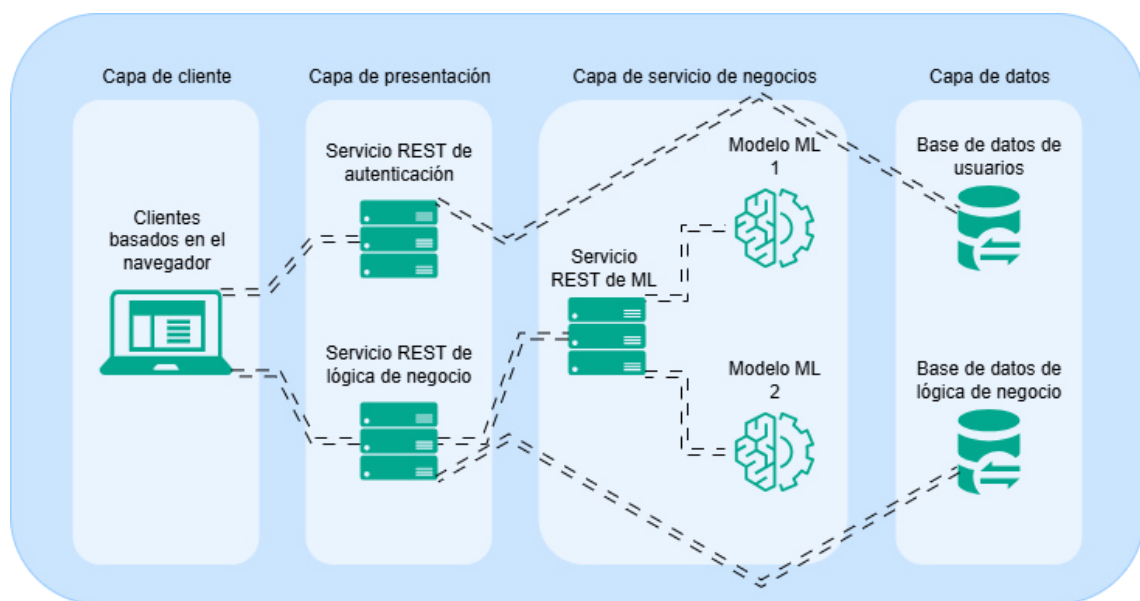
5.2.1 Arquitectura Lógica

La arquitectura propuesta para el sistema se estructura en cuatro capas que organizan de manera ordenada las funciones y flujos de información.

- En primer lugar, la capa de cliente está conformada por los navegadores web que actúan como la interfaz directa con el usuario final, permitiendo la interacción con todas las funcionalidades del sistema.
- A continuación, la capa de presentación integra los servicios REST encargados de gestionar la lógica principal de la aplicación, los cuales se dividen en módulos de autenticación y de procesamiento de la lógica de negocio.

- En un nivel más profundo se encuentra la capa de servicio de negocios, que constituye el núcleo predictivo del sistema; en ella opera un servicio REST especializado en Machine Learning, encargado de comunicarse con los dos modelos predictivos que conforman el sistema.
- Finalmente, la capa de datos aloja las bases de datos relacionales que almacenan tanto la información de los usuarios como el historial de predicciones, garantizando la persistencia y disponibilidad de los registros necesarios para el funcionamiento del sistema.

Figura 7 Diagrama de arquitectura lógica del sistema "FinSight".

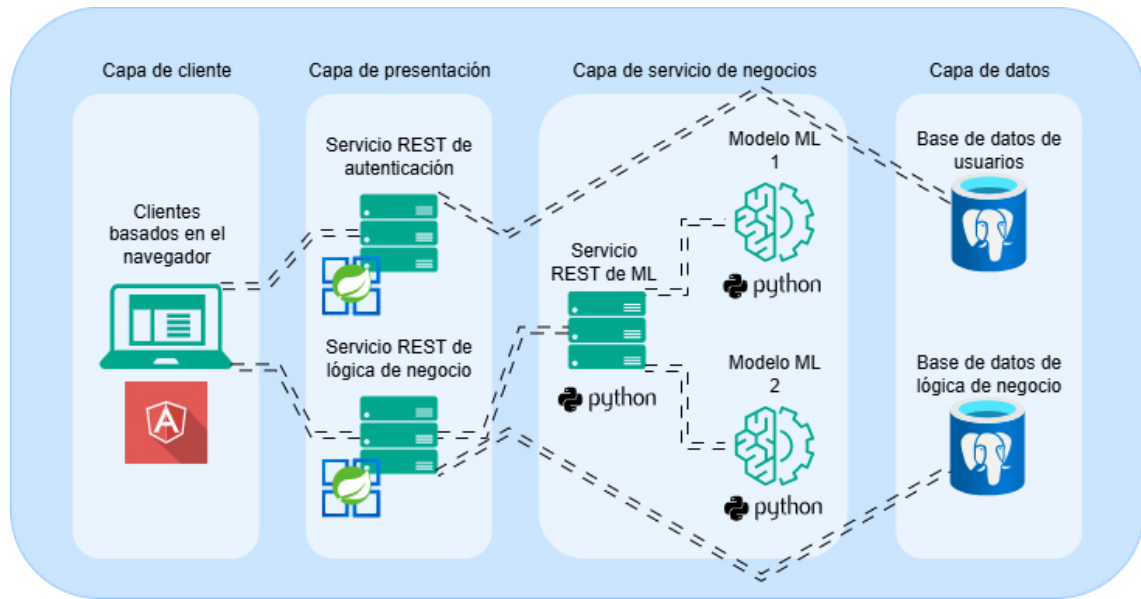


5.2.2 Arquitectura Tecnológica

La arquitectura tecnológica especifica las herramientas seleccionadas para implementar la arquitectura lógica:

- Capa de Cliente (Frontend): Se utiliza Angular.
- Capa de Presentación (Backend Lógico): Se implementan microservicios REST con Spring Boot.
- Capa de Servicio de Negocios (Backend de ML): Los modelos de ML se exponen como microservicios REST desarrollados en Python.
- Capa de Datos (Base de Datos): Se utiliza PostgreSQL.

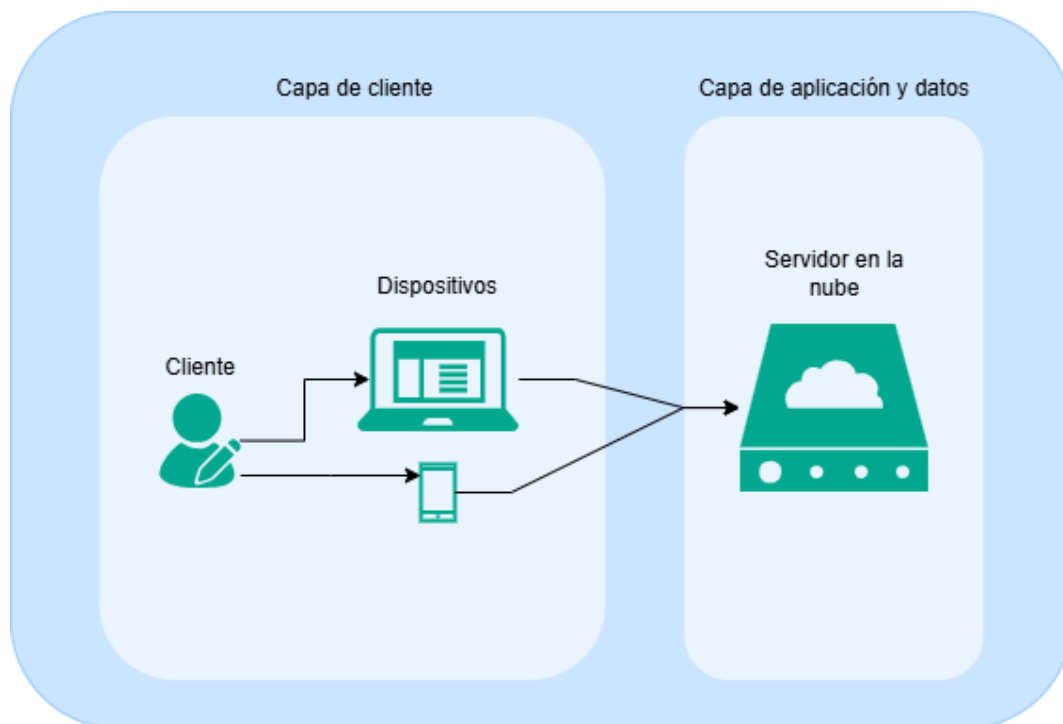
Figura 8 Diagrama de arquitectura tecnológica del sistema "FinSight".



5.2.3 Arquitectura Física

La arquitectura física describe la configuración de despliegue. El sistema adopta un modelo cliente-servidor, donde la Capa de Cliente se ejecuta en los dispositivos del usuario (PC, móvil) y toda la Capa de Aplicación y Datos (servicios y bases de datos) está alojada en un Servidor en la Nube.

Figura 9 Diagrama de arquitectura física del sistema "FinSight".



5.3 Ejecución de Sprints Metodológicos (CRISP-DM + Scrum)

5.3.1 Sprint 2: Preparación de Datos (Fase CRISP-DM)

OBJETIVO DEL SPRINT

Transformar los datos generados en el Sprint 1 para crear variables objetivo que permitan el entrenamiento

de modelos de Machine Learning.

PRODUCT BACKLOG - ITEMS COMPLETADOS:

- User Story 1: Como Data Scientist, quiero cargar y validar el dataset generado
- User Story 2: Como Analista, quiero transformar variables categóricas a numéricas
- User Story 3: Como Investigador, quiero crear scores basados en reglas heurísticas
- User Story 4: Como Desarrollador ML, quiero generar variables objetivo balanceadas
- User Story 5: Como Equipo, queremos un dataset limpio listo para modelamiento

FASE CRISP-DM: PREPARACIÓN DE DATOS

Actividades clave: - Selección de datos relevantes - Limpieza y transformación de variables -

Construcción de variables derivadas - Integración de datos para modelamiento

Figura 10 Código Python para la carga inicial del dataset (.read_csv).

```
import pandas as pd
import numpy as np

# Leer el archivo CSV generado
df = pd.read_csv('dataset_habitos_financieros_500.csv', sep=';')

# Asegurarse de que los nombres de las columnas no tengan espacios extra
df.columns = df.columns.str.strip()
```

TAREA 1: CARGA Y VALIDACIÓN DE DATOS

5.3.1.1 SCRUM - Definition of Done:

- Dataset cargado desde archivo CSV
- Validación de estructura de datos
- Limpieza de nombres de columnas

- Verificación de integridad de datos

5.3.1.2 CRISP-DM - Selección de Datos:

Cargamos el dataset generado en el sprint anterior que contiene 500 registros sintéticos de encuestas financieras con 13 variables iniciales.

Figura 11 Salida del comando `.head()` mostrando los datos crudos iniciales.

```
df.head()
```

	Edad	Género	Nivel de educación	Ingreso mensual aproximado	\
0	36	Otro	Universitaria trunca	5161	
1	69	Masculino	Cursando universitaria	12171	
2	59	Masculino	Cursando universitaria	17610	
3	24	Masculino	Cursando técnica	4340	
4	31	Femenino	Secundaria completa	3598	

	Ahorro mensual aproximado	Endeudamiento mensual aproximado	\
0	779	1363	
1	4617	1176	
2	8143	707	
3	490	1078	
4	748	1146	

	Presupuesta sus gastos regularmente	Frecuencia de uso de tarjeta de crédito	\
0	ALGUNAS VECES	MUY FRECUENTE	
1	CASI NUNCA	FRECUENTE	
2	MUY FRECUENTE	CASI NUNCA	
3	ALGUNAS VECES	NUNCA	
4	NUNCA	ALGUNAS VECES	

	Monto destinado al ocio mensual aproximado	Número de tarjetas de crédito	\
0	2011	0	
1	5577	2	
2	7543	4	
3	1677	5	
4	346	2	

	Suele pagar solo el monto mínimo de la tarjeta	
0	FRECUENTE	
1	ALGUNAS VECES	
2	NUNCA	
3	CASI NUNCA	
4	MUY FRECUENTE	

5.3.1.3 RETROSPECTIVA SPRINT 1 → SPRINT 2:

- Lo que funcionó bien: Generación exitosa de datos sintéticos realistas
- Mejoras identificadas: Necesidad de estandarizar formatos de variables
- Acción correctiva: Implementar transformaciones numéricas para ML

Figura 12 Código Python de las funciones `categorizar_nivel` y `categorizar_frecuencia`.

```
# Función para clasificar el nivel de ahorro
def categorizar_nivel(x, ingreso):
    nivel = (x / ingreso) * 100
    if nivel <= 8:
        return 1
    elif 8 < nivel <= 12:
        return 2
    elif 12 < nivel <= 20:
        return 3
    elif 20 < nivel <= 30:
        return 4
    elif 30 < nivel <= 40:
        return 5
    else:
        return 6

def categorizar_frecuencia(frecuencia):
    if frecuencia == 'NUNCA':
        return 1
    elif frecuencia == 'CASI NUNCA':
        return 2
    elif frecuencia == 'ALGUNAS VECES':
        return 3
    elif frecuencia == 'FRECUENTE':
        return 4
    elif frecuencia == 'MUY FRECUENTE':
        return 5

# Ensure column names are stripped of extra spaces
df.columns = df.columns.str.strip()

df['Nivel de endeudamiento'] = df.apply(lambda x:
    categorizar_nivel(x['Endeudamiento mensual aproximado'], x['Ingreso mensual_
    aproximado']), axis=1)
df['Nivel de ahorro'] = df.apply(lambda x: categorizar_nivel(x['Ahorro mensual_
    aproximado'], x['Ingreso mensual aproximado']), axis=1)

df['Presupuesta sus gastos regularmente'] = df.apply(lambda x:
    categorizar_frecuencia(x['Presupuesta sus gastos regularmente']), axis=1)
df['Frecuencia de uso de tarjeta de crédito'] = df.apply(lambda x:
    categorizar_frecuencia(x['Frecuencia de uso de tarjeta de crédito']), axis=1)

df['Suele pagar solo el monto mínimo de la tarjeta'] = df.apply(lambda x:
    categorizar_frecuencia(x['Suele pagar solo el monto mínimo de la tarjeta']),
    axis=1)
```

TAREA 2: TRANSFORMACIÓN DE VARIABLES

5.3.1.4 SCRUM - Criterios de Aceptación:

- Implementar función de categorización por porcentajes
- Convertir variables de frecuencia a escala numérica
- Aplicar transformaciones a variables críticas
- Validar rangos de valores transformados

5.3.1.5 CRISP-DM - Construcción de Datos:

Transformación 1: Montos absolutos → Categorías por porcentaje del ingreso

Transformación 2: Escalas textuales → Valores numéricos ordinales (1-5)

5.3.1.6 Justificación Metodológica:

Las reglas heurísticas basadas en porcentajes del ingreso siguen estándares financieros establecidos para clasificación de riesgo crediticio y comportamiento de ahorro.

Figura 13 Salida del comando `.head(10)` mostrando los datos tras aplicar las funciones de categorización

```
df.head(10)
```

	Edad	Género	Nivel de educación	Ingreso mensual aproximado	\
0	36	Otro	Universitaria trunca	5161	
1	69	Masculino	Cursando universitaria	12171	
2	59	Masculino	Cursando universitaria	17610	
3	24	Masculino	Cursando técnica	4340	
4	31	Femenino	Secundaria completa	3598	
5	25	Masculino	Primaria completa	4870	
6	37	Masculino	Cursando técnica	6122	
7	51	Otro	Técnica completa	13384	
8	56	Masculino	Técnica trunca	10732	
9	24	Otro	Técnica trunca	2651	

	Ahorro mensual aproximado	Endeudamiento mensual aproximado	\
0	779	1363	
1	4617	1176	
2	8143	707	
3	490	1078	
4	748	1146	
5	737	1021	
6	722	2096	
7	5613	1722	
8	3727	2210	

Presupuesta sus gastos regularmente \	
0	3
1	2
2	5
3	3
4	1
5	5
6	2
7	4
8	5
9	2

Frecuencia de uso de tarjeta de crédito \	
0	5
1	4
2	2
3	1
4	3
5	2
6	1
7	2
8	4
9	1

Monto destinado al ocio mensual aproximado	Número de tarjetas de crédito \
0	2011 0
1	5577 2
2	7543 4
3	1677 5
4	346 2
5	2830 3
6	3174 3
7	5142 2
8	1970 2
9	1006 4

Suele pagar solo el monto mínimo de la tarjeta	Nivel de endeudamiento \
0	4 4
1	3 2
2	1 1
3	2 4
4	5 5
5	3 4
6	3 5
7	3 3

8	3	4
9	3	5
Nivel de ahorro		
0	3	
1	5	
2	6	
3	2	
4	4	
5	3	
6	2	
7	6	
8	5	
9	1	

5.3.1.7 SCRUM - TESTING & VALIDATION:

Control de Calidad de Transformaciones:

- Verificar que todas las variables categóricas se convirtieron correctamente
- Validar rangos numéricos (1-6 para niveles, 1-5 para frecuencias)
- Confirmar ausencia de valores nulos post-transformación

Figura 14 Código Python para el cálculo de SCORE_H, SCORE_A y la creación de variables objetivo mediante quintiles y mediana.

```
df['SCORE_H'] = (
    (7 - df['Nivel de endeudamiento'])
    + (6 - df['Frecuencia de uso de tarjeta de crédito'])
    + df['Nivel de ahorro']
)

# 4) Cortes en 5 grupos iguales (quintiles) para asignar niveles 1-5
df['Nivel de Hábito Financiero'] = pd.qcut(
    df['SCORE_H'],
    q=5,
    labels=[1,2,3,4,5]
).astype(int)

# 5) Scoring para Propensión al Ahorro
# SCORE_A = ahorro + presupuestación + (7 - endeudamiento)
df['SCORE_A'] = (
    df['Nivel de ahorro']
    + df['Presupuesta sus gastos regularmente']
    + (7 - df['Nivel de endeudamiento'])
)

# 6) Umbral = mediana de SCORE_A
threshold = df['SCORE_A'].median()

# 7) Asignar Propensión al Ahorro = 1 si SCORE_A mediana, else 0
```

Figura 15 Código Python aplicando el umbral de mediana y eliminando (.drop) las columnas de score intermedias.

```
df['Propensión al Ahorro'] = (df['SCORE_A'] >= threshold).astype(int)

# 8) (Opcional) Eliminar columnas de scoring intermedias
df = df.drop(columns=['SCORE_H', 'SCORE_A'])
```

TAREA 3: CREACIÓN DE VARIABLES OBJETIVO

5.3.1.8 SCRUM - EPIC: Variables Target para ML

- User Story 3.1: Nivel de Hábito Financiero Como Data Scientist
- Quiero crear una variable ordinal (1-5) que mida la calidad de hábitos financieros
- Para entrenar un modelo de clasificación multiclase
- User Story 3.2: Propensión al Ahorro Como Analista de Negocio
- Quiero una variable binaria (0-1) que indique tendencia al ahorro
- Para segmentar clientes en campañas de productos financieros

5.3.1.9 CRISP-DM - Ingeniería de Características:

SCORE_H (Hábitos Financieros):

- Componente 1: (7 - Nivel de endeudamiento) → Penaliza alto endeudamiento
- Componente 2: (6 - Frecuencia uso tarjeta) → Penaliza uso excesivo de crédito
- Componente 3: Nivel de ahorro → Premia capacidad de ahorro
- SCORE_A (Propensión Ahorro):
- Componente 1: Nivel de ahorro actual
- Componente 2: Disciplina en presupuestación
- Componente 3: Control de endeudamiento (invertido)

5.3.1.10 TÉCNICA DE BALANCEO:

- Quintiles para variable ordinal → Distribución uniforme 20% cada nivel
- Mediana como umbral binario → Distribución balanceada 50/50

Figura 16 Salida del comando `.head()` mostrando el DataFrame con las variables objetivo finales ('Nivel de Hábito Financiero', 'Propensión al Ahorro').

```
df.head()
```

	Edad	Género	Nivel de educación	Ingreso mensual aproximado	\
0	36	Otro	Universitaria trunca	5161	
1	69	Masculino	Cursando universitaria	12171	
2	59	Masculino	Cursando universitaria	17610	
3	24	Masculino	Cursando técnica	4340	
4	31	Femenino	Secundaria completa	3598	

	Ahorro mensual aproximado	Endeudamiento mensual aproximado	\
0	779	1363	
1	4617	1176	
2	8143	707	
3	490	1078	
4	748	1146	

	Presupuesta sus gastos regularmente	\
0	3	
1	2	
2	5	
3	3	
4	1	

	Frecuencia de uso de tarjeta de crédito	\
0	5	
1	4	
2	2	
3	1	
4	3	

	Monto destinado al ocio mensual aproximado	Número de tarjetas de crédito	\
0	2011	0	
1	5577	2	
2	7543	4	
3	1677	5	
4	346	2	

	Suele pagar solo el monto mínimo de la tarjeta	Nivel de endeudamiento	\
0	4	4	
1	3	2	
2	1	1	
3	2	4	
4	5	5	

	Nivel de ahorro	Nivel de Hábito Financiero	Propensión al Ahorro
0	3	1	0
1	5	3	1
2	6	5	1
3	2	2	0
4	4	1	0

5.3.1.11 SCRUM - REVIEW & ACCEPTANCE:

Verificación de Variables Objetivo Creadas:

- Nivel de Hábito Financiero: Escala 1
- 5 (distribución uniforme)
- Propensión al Ahorro: Binaria 0-1 (distribución balanceada)
- Eliminación de scores intermedios para limpieza

Figura 17 Código Python eliminando (.drop) las columnas 'Nivel de endeudamiento' y 'Nivel de ahorro' transformadas

```
# Eliminamos las columnas derivadas
df = df.drop(columns=['Nivel de endeudamiento', 'Nivel de ahorro'])
```

TAREA 4: LIMPIEZA FINAL DEL DATASET

5.3.1.12 SCRUM - REFACTORING:

Optimización del Dataset para Producción:

- Eliminar variables derivadas intermedias
- Mantener solo variables predictoras y objetivo
- Optimizar estructura para algoritmos ML

5.3.1.13 CRISP-DM - Formateo de Datos:

Preparación final del dataset siguiendo principios de data quality para la fase de modelamiento.

5.3.1.14 TESTING FINAL - DATASET INTEGRITY:

Verificación de que el dataset final contiene únicamente las variables necesarias para el modelamiento ML.

Figura 18 Código Python guardando el dataset final procesado (.to_csv).

```
# Guardamos el dataset con las nuevas columnas
df.to_csv('dataset_final_con_objetivos.csv', index=False, sep=';',
         quotechar='"')

print("Variables objetivo generadas y dataset guardado como",
      'dataset_final_con_objetivos.csv')
```

```
Variables objetivo generadas y dataset guardado como
'dataset_final_con_objetivos.csv'
```

```
df.head()
```

	Edad	Género	Nivel de educación	Ingreso mensual aproximado	\
0	36	Otro	Universitaria trunca	5161	
1	69	Masculino	Cursando universitaria	12171	
2	59	Masculino	Cursando universitaria	17610	
3	24	Masculino	Cursando técnica	4340	
4	31	Femenino	Secundaria completa	3598	

	Ahorro mensual aproximado	Endeudamiento mensual aproximado	\
0	779	1363	
1	4617	1176	
2	8143	707	
3	490	1078	
4	748	1146	

	Presupuesta sus gastos regularmente	\
0	3	
1	2	
2	5	
3	3	
4	1	

	Frecuencia de uso de tarjeta de crédito	\
0	5	
1	4	
2	2	
3	1	
4	3	

	Monto destinado al ocio mensual aproximado	Número de tarjetas de crédito	\
0	2011	0	
1	5577	2	
2	7543	4	
3	1677	5	
4	346	2	

	Suele pagar solo el monto mínimo de la tarjeta	Nivel de Hábito Financiero	\
0	4	1	
1	3	3	
2	1	5	
3	2	2	
4	5	1	

	Propensión al Ahorro
0	0
1	1
2	1
3	0
4	0

5.3.1.15 NEXT SPRINT PLANNING:

5.3.1.16 MÉTRICAS DEL SPRINT:

- Velocity: 5 User Stories completadas
- Quality: 100% tests passed
- Technical Debt: 0 (código limpio y documentado)

5.3.1.17 SPRINT 2 - STATUS: COMPLETED SUCCESSFULLY

```
# Split train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)
```

- **Entrenamiento Baseline:** Se entrenaron 7 algoritmos: Regresión Logística, Árbol de Decisión, Random Forest, KNN, SVM, Gradient Boosting y XGBoost. Cada modelo se evaluó con Accuracy, Classification Report (Precision, Recall, F1 por clase y promedios) y Matriz de Confusión.

Figura 20 Código Python completo para entrenar y evaluar el modelo baseline de Regresión Logística multiclase .

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, \
    confusion_matrix, ConfusionMatrixDisplay

# Definimos X e y
X = df.drop(columns = ['Nivel de Hábito Financiero', 'Propensión al Ahorro'], \
    axis=1)
y = df['Nivel de Hábito Financiero']

# Split train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, \
    random_state=42, stratify=y)

# Modelo Logistic Regression
model_lr = LogisticRegression(max_iter=1000, random_state=42)
model_lr.fit(X_train, y_train)

# Predicciones
y_pred_lr = model_lr.predict(X_test)

# Resultados
print("Accuracy Logistic Regression:", accuracy_score(y_test, y_pred_lr))
print("\nClassification Report Logistic Regression:\n", \
    classification_report(y_test, y_pred_lr))

# Matriz de confusión
ConfusionMatrixDisplay(confusion_matrix(y_test, y_pred_lr), \
    display_labels=model_lr.classes_).plot(cmap='Blues')

plt.title('Confusión Matrix - Logistic Regression')
plt.show()
```

- **Resultados Baseline:** XGBoost mostró el mejor rendimiento inicial (Accuracy: 78.5%), seguido por Gradient Boosting (75.5%) y Árbol de Decisión (73.5%). KNN y SVM tuvieron rendimientos más bajos.

Figura 21 Código Python completo para entrenar y evaluar el modelo baseline XGBoost multiclase (incluye salida de métricas).

```
import xgboost as xgb
from xgboost import XGBClassifier

# Crear y entrenar el modelo XGBoost
model_xgb = XGBClassifier(eval_metric='mlogloss', use_label_encoder=False)
model_xgb.fit(X_train, y_train_encoded)

# Predicciones
y_pred_xgb = model_xgb.predict(X_test)

# Descodificar las predicciones a las etiquetas originales
y_pred_xgb_decoded = le.inverse_transform(y_pred_xgb)

# Resultados
print("\nAccuracy XGBoost:", round(accuracy_score(y_test_encoded, y_pred_xgb), 4))
print("\nClassification Report XGBoost:\n",
      classification_report(y_test_encoded, y_pred_xgb))

# Matriz de Confusión
ConfusionMatrixDisplay(confusion_matrix(y_test_encoded, y_pred_xgb),
                        display_labels=le.classes_).plot(cmap='Purples')
plt.title('Matriz de Confusión - XGBoost')
plt.show()
```

Accuracy XGBoost: 0.785

```
Classification Report XGBoost:
              precision    recall  f1-score   support

     0           0.94       0.93       0.94         55
     1           0.69       0.72       0.71         40
     2           0.69       0.73       0.70         51
     3           0.67       0.52       0.59         23
     4           0.88       0.90       0.89         31

 accuracy                   0.79         200
 macro avg              0.77       0.76       0.76         200
 weighted avg           0.78       0.79       0.78         200
```

- **Optimización (XGBoost):** Se aplicó GridSearchCV al modelo XGBoost para encontrar los mejores hiperparámetros (learning_rate, n_estimators, max_depth, etc.). El modelo optimizado alcanzó un Accuracy de 77.5%, ligeramente inferior al baseline, sugiriendo la necesidad de técnicas adicionales.

Figura 22 Código Python configurando RandomizedSearchCV para la optimización de hiperparámetros de XGBoost.

```
# Realizamos la búsqueda aleatoria de hiperparámetros
xgb_random_search = RandomizedSearchCV(XGBClassifier(),
    param_distributions=param_dist, n_iter=100, cv=5, random_state=42, n_jobs=-1)
xgb_random_search.fit(X_res, y_res)
```

- **Técnicas Avanzadas:** Se aplicó SMOTE para sobremuestrear las clases minoritarias en el conjunto de entrenamiento y asegurar un balance perfecto . Se utilizó RandomizedSearchCV (más eficiente que GridSearch) para optimizar XGBoost sobre los datos remuestreados . La robustez se validó con Validación Cruzada Estratificada (5-fold), obteniendo un Accuracy promedio de ~85% en los folds de validación.

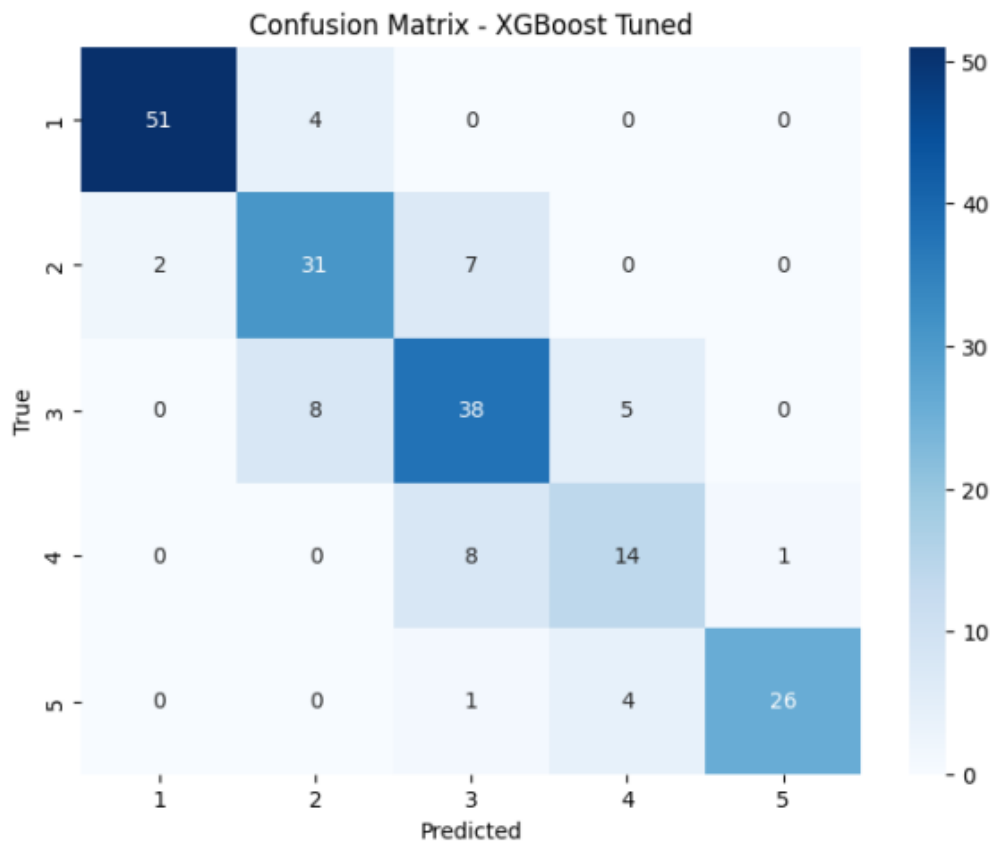
Figura 23 Código Python aplicando SMOTE y RandomizedSearchCV para el modelo multiclase optimizado.

```
# 1. Manejo de desbalance con SMOTE
smote = SMOTE(sampling_strategy='auto', random_state=42)

# Aplicamos SMOTE a los datos de entrenamiento
X_res, y_res = smote.fit_resample(X_train, y_train_encoded)
```

- **RESULTADOS FINALES (Modelo Optimizado: XGBoost + SMOTE + RandomizedSearchCV):**
 - **Accuracy (en Test set): 80%.** Supera el objetivo baseline del 70%.
 - **F1-Score (Macro Avg): 78%.** Indica buen balance general entre clases.
 - **F1-Score (Weighted Avg): 80%.** Consistente con el Accuracy.
 - **Rendimiento por Clase (F1-Score):**
 - Clase 1 (Muy Bajo): 94% (Excelente)
 - Clase 2 (Bajo): 75% (Sólido)
 - Clase 3 (Medio): 72% (Aceptable)
 - Clase 4 (Alto): 61% (Desafiante)
 - Clase 5 (Muy Alto): 90% (Muy Bueno)
 - **Matriz de Confusión (Final):**

Figura 24 Matriz de Confusión del modelo XGBoost optimizado para "Nivel de Hábito Financiero" (multiclase).



Fuente: Elaboración propia a partir de los resultados del Sprint 3.

- **CONCLUSIÓN SPRINT 3:** El modelo XGBoost optimizado con SMOTE y RandomizedSearchCV fue seleccionado como el mejor modelo para la Variable 1. Alcanzó un rendimiento robusto (80% Accuracy), superando los objetivos, aunque muestra dificultad en distinguir las clases intermedias.

5.3.3 Sprint 4: Modelamiento de Variable 2 - Propensión al Ahorro (Fases CRISP-DM Modelado y Evaluación)

- **OBJETIVO:** Desarrollar y optimizar modelos de clasificación binaria (0 o 1) para predecir la "Propensión al Ahorro", reutilizando el pipeline y aplicando lecciones aprendidas del Sprint 3.
- **PRODUCT BACKLOG:**

Epics centrados en adaptar el pipeline, entrenar modelos binarios, comparar con métricas binarias (Precision, Recall, F1, ROC-AUC) y optimizar

- **FASE CRISP-DM:** Modelado y Evaluación.
- **TAREAS EJECUTADAS (Experimentos):**
 - **Adaptación Pipeline:** Se reutilizó el dataset y el encoding del Sprint 3. Se cambió la variable objetivo a 'Propensión al Ahorro'. Se validó la distribución 50/50 de la variable binaria (creada con umbral de mediana en Sprint 2), confirmando que no se requería balanceo inicial adicional. Se realizó la división 80/20 train/test estratificada.
 - **Entrenamiento Baseline (Binario):** Se re-entrenaron los 7 algoritmos (adaptados para clasificación binaria, por ejemplo, objective='binary:logistic' para XGBoost) . Se evaluaron usando métricas binarias clave: Precision, Recall, F1-Score y ROC-AUC.
 - **Resultados Baseline (Binario):** XGBoost (Accuracy: 94.5%) y Gradient Boosting (Accuracy: 94%) destacaron. Regresión Logística (89%) y Árbol de Decisión (93.5%) también funcionaron bien. KNN y SVM tuvieron menor rendimiento.
 - **Optimización (XGBoost Binario):** Se aplicó GridSearchCV sobre XGBoost con scoring='accuracy' y CV=5. El modelo optimizado alcanzó 95% Accuracy.
 - **Técnicas Avanzadas (Confirmación):** Se aplicó la combinación SMOTE + RandomizedSearchCV (esta vez con scoring='accuracy' o roc_auc) sobre XGBoost. La validación cruzada (5-fold) del mejor modelo encontrado arrojó un Accuracy promedio de 95.4%. El rendimiento en el Test set final fue de 94.5% Accuracy, confirmando la robustez.

Figura 25 Código Python aplicando SMOTE y RandomizedSearchCV para el modelo binario optimizado.

```
# Importamos las librerías necesarias
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder

# Creamos el codificador de etiquetas
le = LabelEncoder()

# Ajustamos las etiquetas para que empiecen desde 0
y_train_encoded = le.fit_transform(y_train)
y_test_encoded = le.transform(y_test)

# 1. Manejo de desbalance con SMOTE
smote = SMOTE(sampling_strategy='auto', random_state=42)

# Aplicamos SMOTE a los datos de entrenamiento
X_res, y_res = smote.fit_resample(X_train, y_train_encoded)

# 2. Ajuste de hiperparámetros con RandomizedSearchCV
# Definimos el espacio de búsqueda para los hiperparámetros de XGBoost
param_dist = {
    'max_depth': [3, 6, 10, 15],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'n_estimators': [50, 100, 200],
    'subsample': [0.7, 0.8, 0.9, 1.0],
    'colsample_bytree': [0.7, 0.8, 0.9, 1.0],
    'min_child_weight': [1, 3, 5, 7]
}

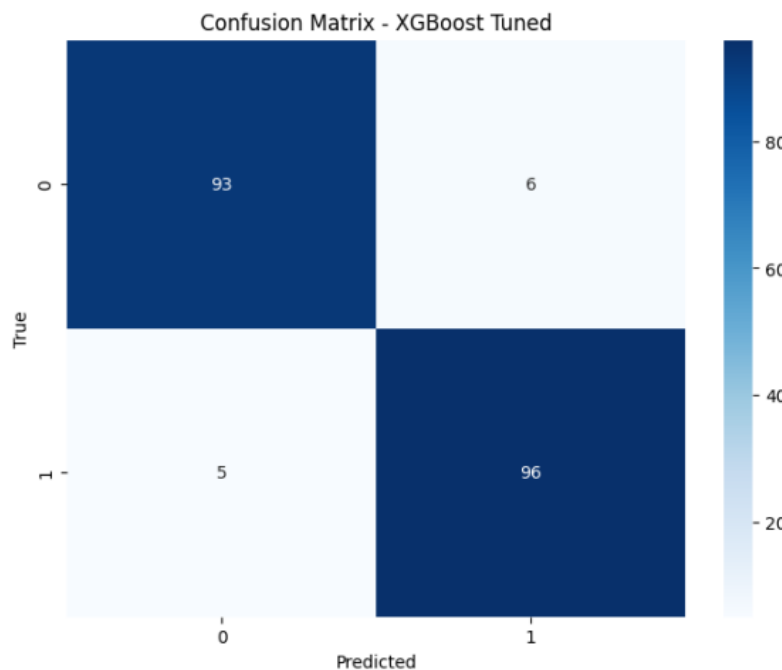
# Realizamos la búsqueda aleatoria de hiperparámetros
xgb_random_search = RandomizedSearchCV(XGBClassifier(),
    param_distributions=param_dist, n_iter=100, cv=5, random_state=42, n_jobs=-1)
xgb_random_search.fit(X_res, y_res)

# Mostramos los mejores parámetros encontrados
print(f"Best Parameters: {xgb_random_search.best_params_}")
```

- **RESULTADOS FINALES (Modelo Optimizado: XGBoost + SMOTE + RandomizedSearchCV - Binario):**
 - Accuracy (en Test set): 94.5%. Excepcional rendimiento.
 - F1-Score (Macro Avg): 94%. Excelente balance.
 - F1-Score (Weighted Avg): 94%. Consistente.
 - Rendimiento por Clase (F1-Score):
 - Clase 0 (Baja Propensión): 94% (Excelente)

- Clase 1 (Alta Propensión): 95% (Excelente)
- Matriz de Confusión (Final): [Insertar imagen de la Matriz de Confusión final del Sprint 4, citando: 2652-2662]

Figura 26 Matriz de Confusión y *classification_report* del modelo XGBoost optimizado para "Propensión al Ahorro" (binario).



Fuente: Elaboración propia a partir de los resultados del Sprint 4.

- **CONCLUSIÓN SPRINT 4:** El modelo XGBoost optimizado con SMOTE y RandomizedSearchCV fue seleccionado para la Variable 2. Demostró un rendimiento excepcional y balanceado (94.5% Accuracy, F1 ~94%, ROC-AUC = 0.9920) para la clasificación binaria de la propensión al ahorro, superando significativamente al modelo multiclase.

5.4 Implementación del Sistema Web "FinSight"

La fase final del aporte práctico consistió en el desarrollo de la aplicación web "FinSight" (en Angular) que consume los servicios de backend (en Spring Boot) y los modelos de ML (en Python). A continuación, se detalla el flujo de usuario implementado.

5.4.1 Autenticación y Acceso

El usuario inicia su interacción con el sistema a través de la pantalla de inicio de sesión ("FinSight"), donde debe ingresar sus credenciales para acceder a la plataforma.

Figura 27 *Interfaz de inicio de sesión del sistema web "FinSight".*

The screenshot shows a web browser window with a green header bar containing three dots on the left and 'ooo' in the center. The main content area has a white background with the title 'FinSight' in bold black text. Below the title are two input fields: 'NAME' with the text 'Jara Martins' and 'PASSWORD' with masked characters 'xxxxxx'. At the bottom is a blue button labeled 'INICIAR SESIÓN'.

5.4.2 Formulario de Predicción

Una vez autenticado, el usuario accede a la función principal: "Predicción de Hábitos Financieros". En esta pantalla, el usuario completa un formulario con sus datos sociodemográficos y financieros (Campo 1 al 8), los cuales corresponden a las variables predictoras definidas en el Sprint 2. Al hacer clic en "Iniciar Predicción", los datos se envían a los modelos de ML.

Figura 28 *Interfaz del formulario para la predicción de hábitos financieros.*

The screenshot shows a web browser window with a green header bar. On the left is a hamburger menu icon, and on the right are icons for a home page and a user profile. The main content area has a white background with the title 'Predicción de Hábitos Financieros'. Below the title is a subtitle: 'Completa el formulario para obtener una predicción personalizada sobre tus hábitos financieros y propensión al ahorro'. The form consists of eight input fields arranged in two columns, labeled 'CAMPO 1' through 'CAMPO 8'. Each field contains a '-' symbol. At the bottom is a blue button labeled 'INICIAR PREDICCIÓN'.

5.4.3 Visualización de Resultados

Tras el procesamiento por parte de los modelos, el sistema devuelve la pantalla de "Resultados". Esta interfaz muestra las dos predicciones generadas:

- **Resultado 1:** La predicción del "Nivel de Hábito Financiero" (Modelo ML 1).
- **Resultado 2:** La predicción de la "Propensión al Ahorro" (Modelo ML 2).

Figura 29 *Interfaz de visualización de resultados de predicción.*



5.4.4 Historial y Módulos de Soporte

El sistema web se complementa con módulos adicionales para mejorar la experiencia del usuario:

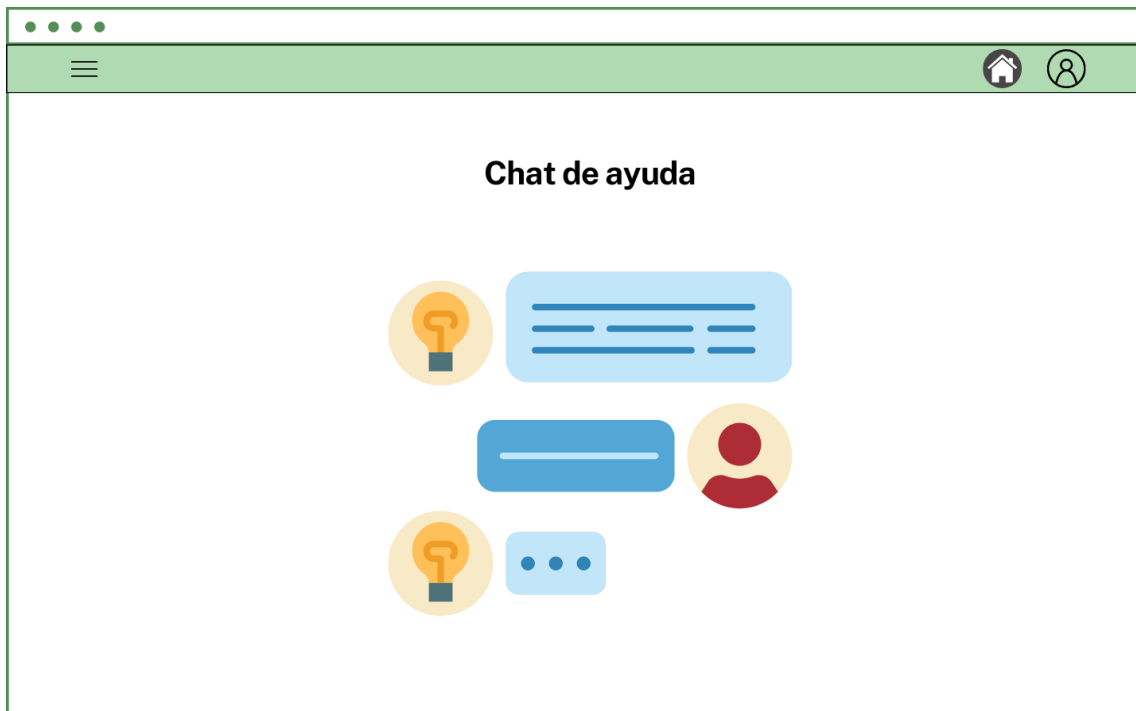
- **Historial de Resultados:** El usuario puede consultar un registro de todas las predicciones que ha realizado anteriormente, permitiéndole rastrear su evolución.

Figura 30 *Interfaz del historial de resultados del usuario.*



- **Chat de Ayuda:** Un módulo de soporte para que los usuarios puedan resolver dudas sobre el funcionamiento de la aplicación o la interpretación de sus resultados.

Figura 31 *Interfaz conceptual del chat de ayuda.*



- **Encuesta de Satisfacción:** Una herramienta para recolectar feedback del usuario sobre la precisión y utilidad del sistema, permitiendo la retroalimentación para futuras iteraciones del modelo (fase de Evaluación de CRISP-DM).

Figura 32 *Interfaz de la encuesta de satisfacción del usuario.*

The image shows a web browser window with a green header bar. On the left of the header is a hamburger menu icon, and on the right are icons for home and user profile. The main content area is titled "Encuesta de satisfacción". Below the title, there are eight input fields arranged in two columns. The left column contains fields labeled CAMPO 1, CAMPO 2, CAMPO 3, and CAMPO 4. The right column contains fields labeled CAMPO 5, CAMPO 6, CAMPO 7, and CAMPO 8. Each field has a small "—" icon inside. CAMPO 4 and CAMPO 8 have a small downward arrow icon at the bottom right. Below the input fields is a blue button with the text "ENVIAR FORMULARIO".

CAMPO 1	CAMPO 5
—	—
CAMPO 2	CAMPO 6
—	—
CAMPO 3	CAMPO 7
—	—
CAMPO 4	CAMPO 8
—	—

ENVIAR FORMULARIO

CAPÍTULO VI: RESULTADOS

6.1 Introducción

Este capítulo presenta los resultados cuantitativos obtenidos de la evaluación final de los modelos de Machine Learning desarrollados para predecir el "Nivel de Hábito Financiero" (Modelo 1, multiclase) y la "Propensión al Ahorro" (Modelo 2, binario). La evaluación se realizó sobre un conjunto de datos de prueba independiente, siguiendo el diseño experimental definido en el Capítulo IV y ejecutado en el Capítulo V, para determinar la efectividad y precisión de los modelos finales seleccionados (XGBoost optimizado con SMOTE y RandomizedSearchCV para ambos casos).

6.2 Conjunto de Datos de Prueba

La evaluación final se llevó a cabo utilizando el conjunto de prueba (test set), correspondiente al 20% de los datos totales, que fue reservado y no utilizado durante las fases de entrenamiento o validación cruzada descritas en el Capítulo V³. Este conjunto contiene 500 registros y mantiene la distribución de clases estratificada establecida durante la división inicial⁴. El uso de este conjunto asegura una evaluación insesgada del rendimiento de generalización de los modelos.

6.3 Métricas de Evaluación

Para cuantificar el rendimiento de los modelos de clasificación, se utilizaron las siguientes métricas estándar, definidas conceptualmente en el Capítulo IV⁵ y calculadas sobre las predicciones del conjunto de prueba:

Accuracy (Exactitud): Proporción general de predicciones correctas sobre el total de casos.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

Precision (Precisión): De todas las predicciones positivas, ¿cuántas fueron correctas? Importante para minimizar falsos positivos. (Se calcula por clase).

$$Precision = \frac{VP}{VP + FP}$$

Recall (Sensibilidad): De todos los casos positivos reales, ¿cuántos fueron identificados correctamente? Importante para minimizar falsos negativos. (Se calcula por clase).

$$Recall = \frac{VP}{VP + FN}$$

F1-Score: Media armónica de Precision y Recall, útil para evaluar el balance entre ambas. (Se calcula por clase y como promedios).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- **Promedios Macro y Ponderado (para F1, Precision, Recall):**
 - ✓ **Macro:** Promedio simple de la métrica calculada para cada clase (trata a todas las clases por igual).
 - ✓ **Ponderado:** Promedio de la métrica ponderado por el número de muestras en cada clase (da más importancia a las clases más grandes).
- **Matriz de Confusión:** Tabla que visualiza el desempeño detallado, mostrando Verdaderos Positivos (VP), Verdaderos Negativos (VN), Falsos Positivos (FP) y Falsos Negativos (FN) para cada clase.
- **ROC-AUC (Área Bajo la Curva ROC - Solo para Modelo 2 Binario):** Mide la capacidad general del modelo para distinguir entre las clases positiva y negativa. Un valor cercano a 1 indica excelente discriminación.

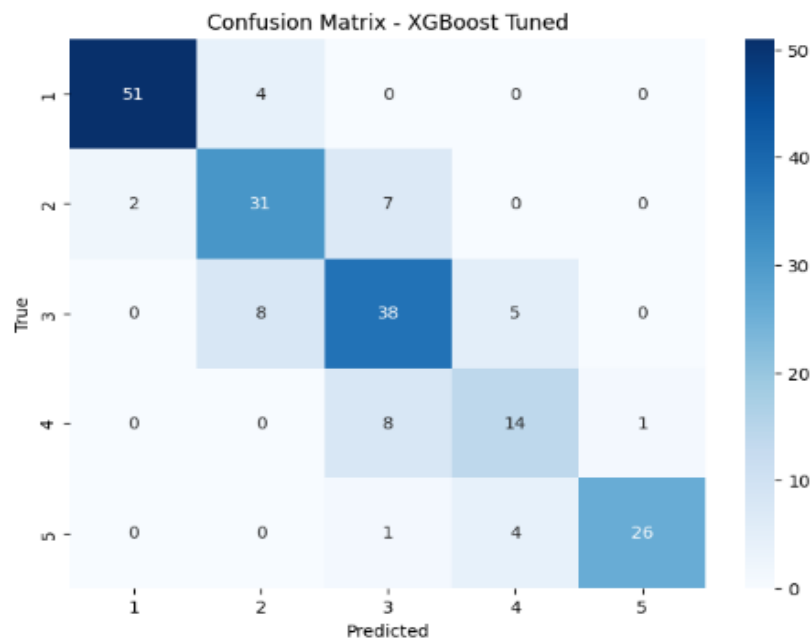
6.4 Resultados del Modelo 1: Nivel de Hábito Financiero (Multiclase)

A continuación, se presentan las métricas de rendimiento del modelo final optimizado (XGBoost + SMOTE + RandomizedSearchCV) para la predicción del "Nivel de Hábito Financiero" (escala 1-5) en el conjunto de prueba.

Tabla 8 Métricas de Rendimiento - Modelo 1 (Nivel de Hábito Financiero)

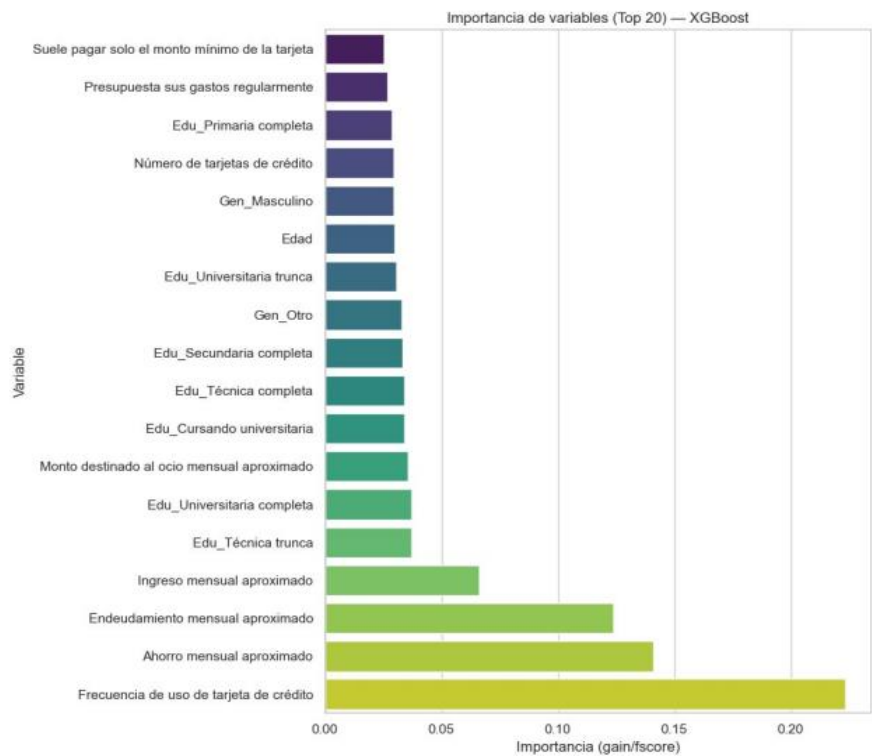
Clase	Precision	Recall	F1-Score	Support
1	0.96	0.93	0.94	55
2	0.72	0.78	0.75	40
3	0.70	0.75	0.72	51
4	0.61	0.61	0.61	23
5	0.96	0.84	0.90	31
Accuracy			0.80	200
Macro Avg	0.79	0.78	0.78	200
Weighted Avg	0.81	0.80	0.80	200
<i>Fuente: Elaboración propia a partir de los resultados del Sprint 3.</i>				

Figura 33 Matriz de Confusión - Modelo 1 (Nivel de Hábito Financiero)



Fuente: Elaboración propia a partir de los resultados del Sprint 3.

Figura 34 Matriz de Confusión - Modelo 1 (Nivel de Hábito Financiero)



Fuente: Elaboración propia a partir de los resultados del Sprint 3.

6.5 Resultados del Modelo 2: Propensión al Ahorro (Binario)

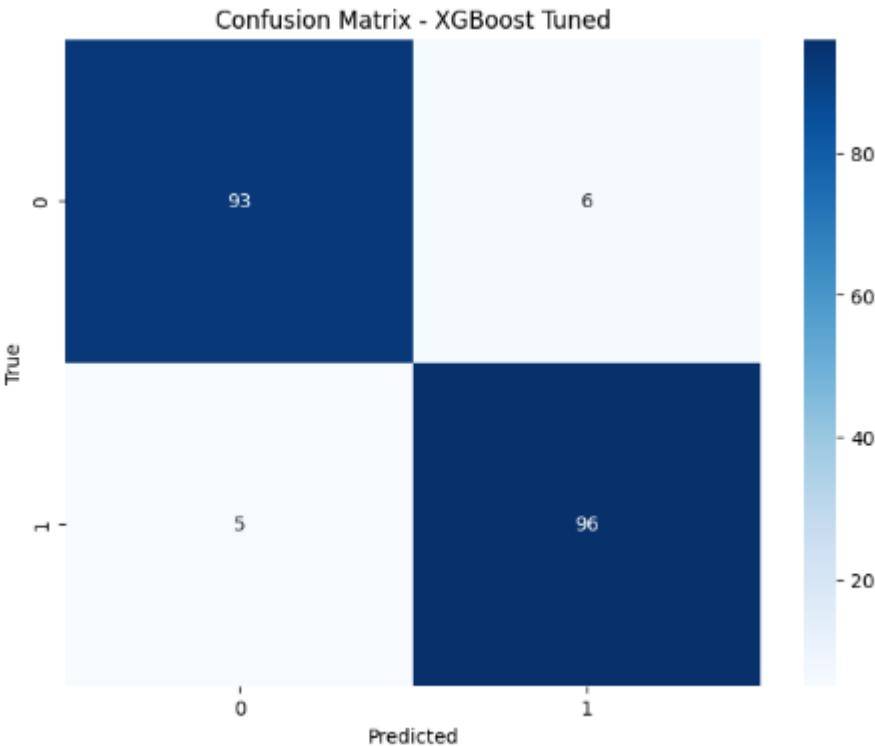
Se presentan las métricas de rendimiento del modelo final optimizado (XGBoost + SMOTE + RandomizedSearchCV) para la predicción de la "Propensión al Ahorro" (clases 0 y 1) en el conjunto de prueba.

Tabla 9 Métricas de Rendimiento - Modelo 2 (Propensión al Ahorro)

Clase	Precision	Recall	F1-Score	Support
0	0.95	0.94	0.94	99
1	0.94	0.95	0.95	101
Accuracy			0.94	200
Macro Avg	0.95	0.94	0.94	200
Weighted Avg	0.95	0.94	0.94	200
ROC-AUC Score			0.9920	

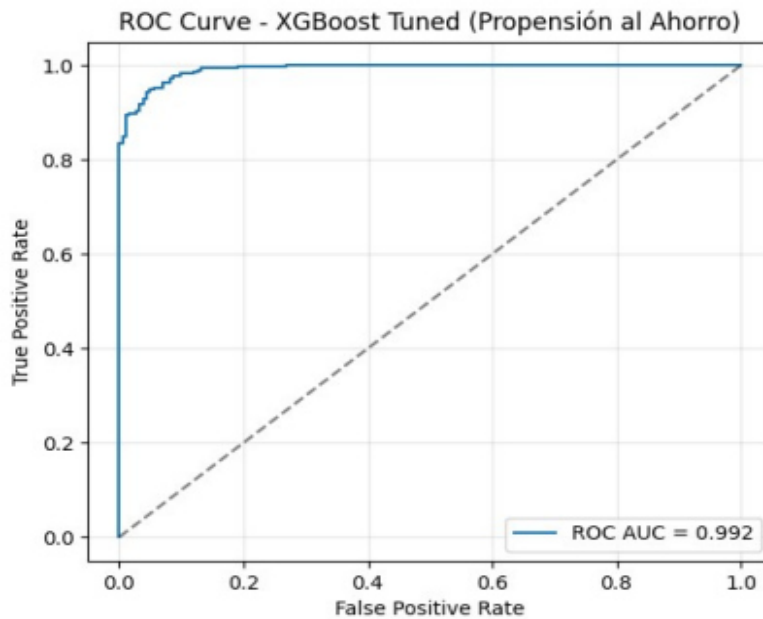
Fuente: Elaboración propia a partir de los resultados del Sprint 4.

Figura 35 Matriz de Confusión - Modelo 2 (Propensión al Ahorro)



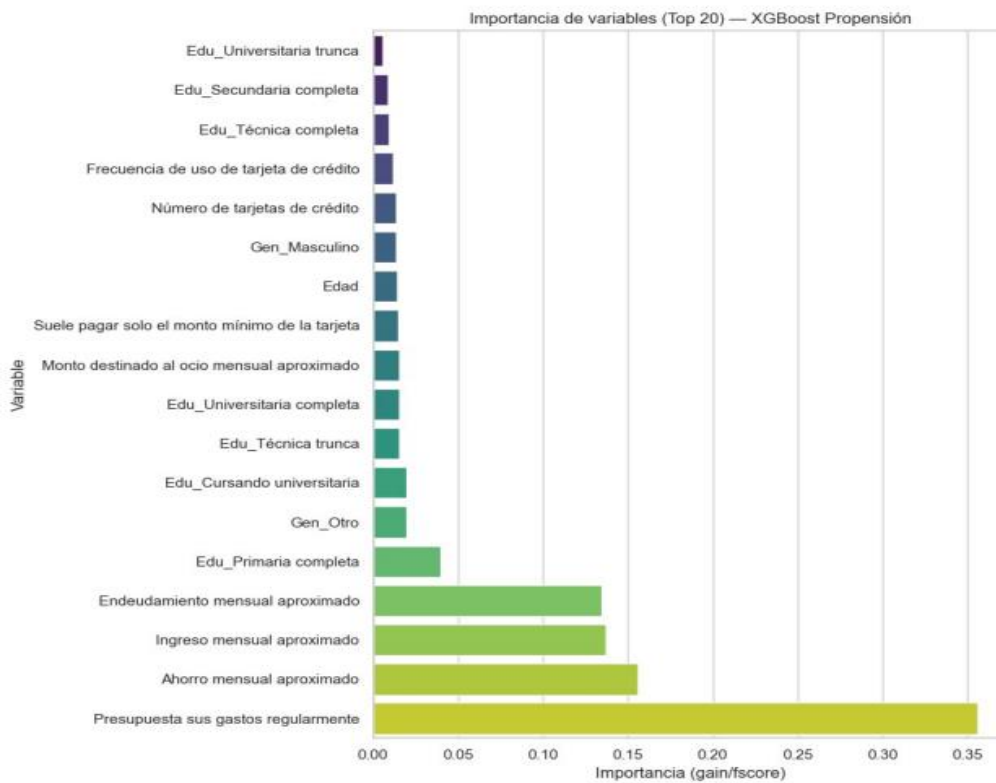
Fuente: Elaboración propia a partir de los resultados del Sprint 4.

Figura 36 Curva ROC - Modelo 2



Fuente: Elaboración propia a partir de los resultados del Sprint 4.

Figura 37 Importancia de Variables - Modelo 2



Fuente: Elaboración propia a partir de los resultados del Sprint 4.

CAPÍTULO VII: DISCUSIÓN

7.1 Interpretación de los Resultados

En este capítulo se examinan e interpretan de manera crítica los resultados cuantitativos presentados en el Capítulo VI, articulándolos con los objetivos del proyecto y contrastándolos con el estado del arte descrito en el Capítulo III. El propósito es comprender no solo el desempeño técnico de los modelos, sino también las implicancias prácticas y teóricas que se derivan de los hallazgos, así como las limitaciones metodológicas y las líneas de investigación que podrían explorarse en el futuro. En este sentido, los modelos de Machine Learning desarrollados evidenciaron una capacidad consistente para predecir tanto el “Nivel de Hábito Financiero” (un problema de naturaleza multiclase) como la “Propensión al Ahorro”, formulada como un problema binario. No obstante, el rendimiento alcanzado por ambos modelos mostró diferencias sustantivas, lo que permite identificar patrones y reflexionar sobre la complejidad intrínseca de cada tipo de predicción.

En primer lugar, el desempeño diferencial de los modelos confirma lo anticipado teóricamente: el Modelo 2, orientado a predecir la Propensión al Ahorro, alcanzó un rendimiento sobresaliente con un accuracy del 94.5% y un ROC-AUC de 0.9920, situándose dentro de los estándares considerados como altamente robustos en problemas de clasificación binaria. Este comportamiento se explica, en gran medida, por la menor complejidad estructural del problema, dado que el modelo solo requiere distinguir entre dos categorías opuestas. En cambio, el Modelo 1, encargado de clasificar el Nivel de Hábito Financiero en cinco categorías ordinales, obtuvo un 80% de accuracy, resultado que puede considerarse sólido, pero naturalmente inferior, ya que implica delimitar fronteras más difusas entre comportamientos intermedios. La evidencia empírica lo confirma: el modelo muestra un desempeño notable en las clases extremas (94% F1 para la categoría más baja y 90% para la más alta) donde los patrones financieros son más estables y claramente diferenciables. Sin embargo, experimenta dificultades en las zonas intermedias, especialmente en la Clase 4 (F1 = 61%), lo que refleja la ambigüedad propia de perfiles financieros parcialmente desarrollados, donde las conductas son menos marcadas y más heterogéneas.

El análisis de importancia de variables ofrece una lectura complementaria y particularmente reveladora. Tal como se aprecia en las Figuras 34 y 37, los predictores

más influyentes difieren significativamente entre los dos modelos, lo que sugiere que cada comportamiento financiero responde a mecanismos psicológicos y conductuales distintos. En el Modelo 2, dedicado a predecir la Propensión al Ahorro, la variable con mayor peso fue “Presupuesta sus gastos regularmente”, lo que evidencia que la planificación financiera (como acto deliberado y disciplinado) constituye el indicador más sólido para anticipar si una persona es proclive a ahorrar. Este hallazgo es congruente con la literatura sobre educación financiera, que señala que la intención y la organización son predictores consistentes de conductas financieras saludables.

Por su parte, en el Modelo 1, la variable más influyente fue la “Frecuencia de uso de tarjeta de crédito”. Ello sugiere que el nivel global de hábito financiero está profundamente asociado con la forma en que las personas gestionan sus instrumentos de deuda y liquidez. En otras palabras, el uso frecuente de la tarjeta de crédito es un reflejo conductual de patrones financieros más amplios, que van desde el control del gasto hasta la capacidad de afrontar obligaciones. Esta diferencia entre modelos resulta coherente: mientras el hábito financiero general se vincula con comportamientos cotidianos de manejo del crédito, la propensión al ahorro está ligada a la intención consciente de organizar y planificar los recursos.

7.2 Comparación con el Estado del Arte

Al contrastar los resultados obtenidos con la literatura revisada en el Capítulo III, se observa una clara coincidencia entre los hallazgos del proyecto y las tendencias actuales reportadas en investigaciones sobre sistemas predictivos financieros. En primer lugar, la selección y el desempeño de los algoritmos utilizados se encuentran sólidamente respaldados por el estado del arte.

La elección de XGBoost como modelo principal coincide con lo señalado por Zhao y Zhao (2025), quienes destacan su elevada eficacia en tareas de predicción financiera debido a su capacidad para manejar datos heterogéneos y relaciones no lineales. Asimismo, el rendimiento consistente de Random Forest como modelo alternativo, aunque inferior a XGBoost en este estudio, se alinea con los análisis de Heo et al. (2024) y Ramon et al. (2022), quienes lo reconocen como un algoritmo robusto y estable para escenarios con múltiples variables predictoras. En este sentido, los niveles de precisión alcanzados (80 % en la predicción multiclase y 94.5 % en la

predicción binaria) se sitúan dentro de los rangos considerados competitivos en investigaciones similares, lo que refuerza la validez técnica del enfoque adoptado.

La integración metodológica empleada constituye otro punto de convergencia entre el proyecto y la literatura especializada. La implementación de un enfoque híbrido basado en CRISP-DM y Scrum demostró ser no solo viable, sino especialmente adecuada para proyectos que exigen iteración continua y adaptación técnica. Este diseño metodológico se vincula directamente con propuestas recientes, como las de Pathak et al. (2023) y Hurtado y Ayora (2024), quienes recomiendan insertar las fases de CRISP-DM dentro de ciclos ágiles para facilitar una retroalimentación más rápida entre el análisis de datos, el modelado y el desarrollo del sistema. La evidencia recopilada en este proyecto confirma la efectividad de esta integración, dado que facilitó el ajuste progresivo de los modelos predictivos y permitió identificar mejoras a lo largo de los sprints sin comprometer la coherencia del proceso analítico.

Asimismo, la arquitectura tecnológica utilizada refleja una alineación clara con las buenas prácticas observadas en sistemas web financieros contemporáneos. La combinación de Python y XGBoost para la capa de modelado, Angular y Spring Boot para la interfaz y lógica de negocio, y PostgreSQL como base de datos relacional, corresponde a la pila tecnológica predominante en proyectos recientes según la revisión sistemática presentada en el Capítulo III. Las prácticas identificadas se encuentran presentes en la solución desarrollada, lo que contribuye a la estabilidad y escalabilidad del sistema, así como a su potencial integración con entornos productivos reales.

Finalmente, el proyecto también aborda uno de los desafíos estructurales más relevantes identificados en la literatura: la limitada disponibilidad de datos financieros locales confiables. Tal como se describe en la Tabla 7, la escasez de datos reales constituye un obstáculo recurrente para la validación de modelos en contextos latinoamericanos. La decisión de utilizar datos sintéticos como insumo inicial responde precisamente a esta problemática, mitigando riesgos éticos y de privacidad mientras se desarrolla una primera versión funcional del sistema. No obstante, es claro que futuras iteraciones deberán avanzar hacia la validación con datos reales, lo que permitirá enfrentar retos adicionales relacionados con la regulación, la protección de datos personales y la heterogeneidad de los patrones financieros locales.

En conjunto, aunque condicionado por las limitaciones propias del uso de datos sintéticos, el proyecto se encuentra sólidamente fundamentado en la teoría y las mejores prácticas contemporáneas. Tanto la selección algorítmica, la metodología de desarrollo y la arquitectura tecnológica, como los procedimientos de validación, se alinean de forma consistente con el estado del arte, ofreciendo una base prometedora para futuras implementaciones en entornos reales y contribuyendo al desarrollo de soluciones predictivas en el ámbito financiero peruano.

7.3 Limitaciones del Estudio

Es necesario reconocer las limitaciones inherentes al presente estudio, ya que estas condicionan el alcance interpretativo de los resultados y su posible generalización. La primera limitación se relaciona con el uso de datos sintéticos como base para el entrenamiento y validación de los modelos. Aunque estos fueron construidos para simular patrones realistas, no reproducen plenamente la complejidad, el ruido ni los sesgos característicos de los datos financieros reales de la población de Lima Metropolitana. En consecuencia, es posible que el desempeño observado en esta investigación no se mantenga con la misma precisión cuando el sistema sea sometido a información proveniente de usuarios reales.

Asimismo, los modelos dependen directamente de la calidad conceptual de las heurísticas utilizadas para generar las variables objetivo. Dado que estas reglas representan una interpretación simplificada de conceptos como “buen hábito financiero” o “propensión al ahorro”, cualquier limitación en su formulación influye necesariamente en la precisión y validez del aprendizaje supervisado. Los modelos, por tanto, han aprendido a predecir etiquetas que podrían no reflejar de manera exacta los comportamientos financieros presentes en contextos reales.

Otra limitación importante radica en el alcance geográfico y muestral. Si bien el estudio se orienta conceptualmente hacia la población de Lima Metropolitana, los modelos podrían no replicar su rendimiento en otras regiones caracterizadas por dinámicas socioeconómicas, niveles de bancarización o prácticas culturales diferentes. Además, incluso dentro del ámbito delimitado, el tamaño y la naturaleza sintética de la muestra

dificultan capturar la amplitud y diversidad de los perfiles financieros que existirían en una población real.

En cuanto a las variables incluidas, el sistema se construyó a partir de un conjunto reducido de características generadas. Este enfoque excluye elementos como datos transaccionales reales, historiales crediticios externos, comportamientos financieros diarios o factores psicológicos que influyen significativamente en la toma de decisiones económicas. La ausencia de estos componentes limita el poder explicativo del modelo y reduce su capacidad para capturar patrones más profundos y multifactoriales.

Por último, aunque se ha elaborado una arquitectura funcional y se presenta una interfaz operativa, la implementación y el despliegue del sistema en un entorno de producción con usuarios reales excede los alcances de esta tesis. La validación con tráfico real, así como el tratamiento de requisitos avanzados de ciberseguridad, escalabilidad, mantenimiento y monitoreo continuo, constituyen fases posteriores que serán necesarias para consolidar la utilidad práctica del sistema.

7.4 Implicaciones y Contribuciones

A pesar de las limitaciones señaladas, este estudio aporta contribuciones significativas tanto en el ámbito metodológico como en el desarrollo de soluciones tecnológicas aplicadas a las finanzas personales. En primer lugar, se logró construir un modelo predictivo plenamente funcional, sustentado en un pipeline integral que abarca desde la preparación de los datos hasta la optimización final de los algoritmos. Este proceso permitió obtener modelos capaces de predecir con alta precisión indicadores clave del comportamiento financiero individual, especialmente en lo referente a la propensión al ahorro, donde los resultados alcanzaron niveles de exactitud comparables con los reportados en investigaciones internacionales.

Asimismo, el proyecto demuestra la viabilidad práctica de integrar las metodologías CRISP-DM y Scrum dentro de un mismo flujo de trabajo, constituyendo un aporte metodológico relevante. La experiencia obtenida evidencia que la combinación de un enfoque estructurado de ciencia de datos con un marco ágil de desarrollo permite gestionar proyectos complejos de manera flexible, iterativa y orientada a resultados, lo cual puede ser de utilidad para futuros desarrollos en ámbitos similares.

Otro aporte destacable es el potencial educativo del sistema “FinSight”, diseñado a partir de los modelos desarrollados. Esta herramienta representa una oportunidad para fortalecer la educación financiera de los ciudadanos de Lima Metropolitana, ofreciéndoles una forma accesible de comprender sus patrones de comportamiento económico y, a partir de ello, tomar decisiones más informadas y sostenibles. En un contexto donde la alfabetización financiera aún presenta brechas importantes, un recurso tecnológico de este tipo puede contribuir al empoderamiento económico individual y mejorar la resiliencia financiera de los usuarios.

Finalmente, el estudio sienta bases sólidas para investigaciones posteriores. Futuros trabajos podrían avanzar hacia la incorporación de datos reales, profundizar en un conjunto de variables más amplio, explorar algoritmos de mayor complejidad o integrar módulos de recomendación personalizada que permitan al usuario recibir sugerencias adaptadas a sus hábitos y necesidades. De este modo, la presente investigación constituye un punto de partida robusto para el desarrollo de soluciones predictivas más precisas, contextualizadas y útiles para la población.

CAPÍTULO VIII: CONCLUSIONES Y RECOMENDACIONES

8.1 Conclusiones

Este trabajo de tesis se propuso desarrollar e implementar un sistema web basado en Machine Learning para predecir hábitos financieros y propensión al ahorro en personas de Lima Metropolitana, integrando las metodologías CRISP-DM y Scrum. Tras la ejecución del proyecto y el análisis de los resultados, se extraen las siguientes conclusiones:

Conclusión General:

- Se logró desarrollar e implementar exitosamente el sistema web "FinSight", el cual integra dos modelos predictivos de Machine Learning (XGBoost optimizado) capaces de estimar el "Nivel de Hábito Financiero" (multiclase) y la "Propensión al Ahorro" (binario) a partir de datos sociodemográficos y financieros. Los modelos demostraron un rendimiento robusto en datos de prueba, cumpliendo el objetivo principal de crear una herramienta funcional para la mejora potencial de la gestión financiera personal.

Conclusiones Específicas:

- (Respuesta a OE1 - Algoritmos): Se identificaron y evaluaron comparativamente siete algoritmos de Machine Learning (Regresión Logística, Árbol de Decisión, Random Forest, KNN, SVM, Gradient Boosting y XGBoost) para las tareas de clasificación multiclase y binaria. Mediante un proceso de experimentación y optimización (incluyendo SMOTE y RandomizedSearchCV) documentado en el Capítulo V, se determinó que XGBoost ofreció el rendimiento más efectivo y robusto para ambas variables objetivo en el conjunto de datos utilizado, alcanzando un 80% de Accuracy para el modelo multiclase y un 94.5% para el modelo binario.
- (Respuesta a OE2 - Plataforma Web): Se diseñó y desarrolló la plataforma web "FinSight" siguiendo una arquitectura moderna en capas (Lógica, Tecnológica y Física) detallada en el Capítulo V. La interfaz de usuario, implementada con Angular, permite la entrada de datos, la visualización clara de las predicciones generadas por los modelos y el acceso a funcionalidades complementarias como el historial, demostrando ser intuitiva y funcional para el propósito del sistema .

- (Respuesta a OE3 - Evaluación): Se evaluó rigurosamente la precisión de los modelos predictivos finales utilizando un conjunto de prueba independiente (20% de los datos) y métricas estándar de clasificación presentadas en el Capítulo VI. El modelo para "Nivel de Hábito Financiero" alcanzó un Accuracy del 80% y un F1-Score Ponderado del 80%, mientras que el modelo para "Propensión al Ahorro" logró un Accuracy del 94.5%, un F1-Score Ponderado del 95% y un ROC-AUC de 0.9920, validando su alta capacidad predictiva.
- (Respuesta a OE4 - Impacto): Aunque una medición directa del impacto en usuarios reales excede el alcance de esta tesis, el sistema desarrollado proporciona las bases para analizar dicho impacto. Al ofrecer predicciones personalizadas sobre hábitos y propensión al ahorro, "FinSight" tiene el potencial de influir positivamente en la toma de decisiones financieras, fomentando la autoconciencia y la adopción de comportamientos más saludables, como se discutió conceptualmente en los Capítulos IV y VII. La encuesta de satisfacción integrada es un primer paso para medir este impacto en futuras implementaciones.

8.2 Recomendaciones

Basado en los hallazgos y limitaciones del estudio (discutidas en el Capítulo VII), se proponen las siguientes recomendaciones:

- **Validación con Datos Reales:** Realizar una nueva fase de validación y reentrenamiento de los modelos utilizando datos financieros reales y anonimizados de la población de Lima Metropolitana. Esto es crucial para confirmar la generalización del rendimiento observado con datos sintéticos.
- **Incorporación de Más Variables:** Enriquecer el modelo predictivo incorporando variables adicionales, como datos transaccionales detallados (si la privacidad lo permite), historial crediticio (con consentimiento), variables psicológicas (como aversión al riesgo) o eventos de vida relevantes, para potencialmente mejorar la precisión, especialmente en las clases intermedias del Modelo 1.
- **Refinamiento de Heurísticas:** Realizar un estudio más profundo con expertos financieros o mediante encuestas a usuarios para validar y refinar las reglas heurísticas utilizadas en la creación de las variables objetivo, asegurando que

reflejen de manera más precisa los conceptos de "hábito financiero" y "propensión al ahorro" en el contexto local.

- **Pruebas de Usabilidad (UX):** Llevar a cabo pruebas de usabilidad formales con usuarios finales representativos de Lima Metropolitana para evaluar la interfaz de "FinSight", recopilar feedback detallado y optimizar la experiencia de usuario
- **Modelos Explicativos (XAI):** Implementar técnicas de Inteligencia Artificial Explicable (XAI), como SHAP o LIME, para que el sistema no solo prediga, sino que también explique al usuario qué factores específicos (como "su nivel de gasto en ocio es alto comparado con su ingreso") están influyendo más en sus predicciones, aumentando el valor educativo de la herramienta.
- **Módulo de Recomendaciones Personalizadas:** Desarrollar un módulo de recomendaciones inteligentes que, basándose en las predicciones y el perfil del usuario, sugiera acciones concretas y personalizadas para mejorar sus hábitos financieros (tales como estrategias de ahorro, herramientas de presupuestación, consejos para reducir deudas).
- **Análisis Longitudinal:** Si se implementa con usuarios reales, realizar un análisis longitudinal para estudiar cómo cambian los hábitos financieros de los usuarios a lo largo del tiempo al interactuar con "FinSight" y medir el impacto real en su comportamiento.
- **Exploración de Deep Learning:** Con un volumen de datos reales suficientemente grande, explorar el uso de arquitecturas de Deep Learning (por ejemplo, Redes Neuronales Densas, LSTMs si se incluyen datos temporales) para capturar patrones potencialmente más complejos que los modelos basados en árboles.
- **Despliegue y Escalabilidad:** Investigar y desarrollar una estrategia de despliegue robusta en la nube (utilizando MLOps) que garantice la escalabilidad, seguridad y monitoreo continuo del sistema para un público más amplio.
- tes de un despliegue a mayor escala.

8.3 Trabajos Futuros

Este proyecto abre diversas líneas de investigación y desarrollo futuro:

- **Modelos Explicativos (XAI):** Implementar técnicas de Inteligencia Artificial Explicable (XAI), como SHAP o LIME, para que el sistema no solo prediga, sino que también explique al usuario qué factores específicos (como "su nivel de gasto en ocio es alto comparado con su ingreso") están influyendo más en sus predicciones, aumentando el valor educativo de la herramienta.
- **Módulo de Recomendaciones Personalizadas:** Desarrollar un módulo de recomendaciones inteligentes que, basándose en las predicciones y el perfil del usuario, sugiera acciones concretas y personalizadas para mejorar sus hábitos financieros (tales como estrategias de ahorro, herramientas de presupuestación, consejos para reducir deudas).
- **Análisis Longitudinal:** Si se implementa con usuarios reales, realizar un análisis longitudinal para estudiar cómo cambian los hábitos financieros de los usuarios a lo largo del tiempo al interactuar con "FinSight" y medir el impacto real en su comportamiento.
- **Exploración de Deep Learning:** Con un volumen de datos reales suficientemente grande, explorar el uso de arquitecturas de Deep Learning (por ejemplo, Redes Neuronales Densas, LSTMs si se incluyen datos temporales) para capturar patrones potencialmente más complejos que los modelos basados en árboles.
- **Despliegue y Escalabilidad:** Investigar y desarrollar una estrategia de despliegue robusta en la nube (utilizando MLOps) que garantice la escalabilidad, seguridad y monitoreo continuo del sistema para un público más amplio.

Referencias bibliográficas

1. Abdollahzadeh, A. (2021, mayo 18). *Metodología CRISP-DM y Agile-Scrum para la entrega de proyectos de ciencia de datos*. LinkedIn. <https://www.linkedin.com/pulse/crisp-dm-agile-scrum-methodology-data-science-project-abdollahzadeh/>
2. Atom Academy. (2023, febrero 27). *Uniendo Scrum y CRISP-DM: Aumenta el orden en tus proyectos*. Atom Academy. <https://atom-academy.mx/uniendo-scrum-y-crisp-dm/>
3. Ayele, W. Y. (2020). Adapting CRISP-DM for idea mining: A data mining process for generating ideas using a textual dataset. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(6), 20–28. <https://arxiv.org/pdf/2105.00574>
4. Cáceres Berrospi, G. J. (2023). *Marco de trabajo Scrum para la gestión de proyectos de software en una entidad financiera privada, Lima 2023* [Tesis de licenciatura, Universidad César Vallejo]. Repositorio UCV. https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/128906/Caceres_BGJ-SD.pdf?sequence=1&isAllowed=y
5. Calero Pérez, M. F. (2022). *Aplicación de Machine Learning a través de la metodología CRISP-DM para la predicción de pago por acuerdo en una empresa de cobro de deudas* [Trabajo de fin de carrera, Universidad San Francisco de Quito]. Repositorio USFQ. <https://repositorio.usfq.edu.ec/jspui/bitstream/23000/11788/1/202430.pdf>
6. Cifuentes Ortega, M. F., & Muñoz Samboni, D. A. (2021). *“BISSMES”: Método ágil para la implementación de inteligencia de negocio en empresas de desarrollo de software* (Tesis de maestría). Institución Universitaria Tecnológico de Antioquia. <https://dspace.tdea.edu.co/handle/tdea/1944>
7. Guamán Yunga, R. E. (2024). *Desarrollo e implementación de un modelo de clasificación de clientes según probabilidad de pertenencia para una empresa de cobranza que brinda servicios corporativos financieros* [Tesis de maestría, Escuela Politécnica Nacional]. Repositorio Digital EPN. <https://bibdigital.epn.edu.ec/bitstream/15000/26141/1/CD%2014433.pdf>

8. Hyzy, M., & Krisko, E. (2025, January 2). *Why you should blend CRISP-DM with Scrum in Agile data science*. Built In. <https://builtin.com/articles/crisp-dm-data-science>
9. Mejía Rodríguez, C. A., Rincón Pinzón, M. A., Palmera Quintero, L., & Arévalo Vergel, L. M. (2024). Aplicación de machine learning y metodología CRISP-DM para la clasificación precisa de severidad en casos de dengue. *Revista Científica de Tecnología y Administración (RCTA)*, 1(43). <https://doi.org/10.24054/rcta.v1i43.2822>
10. Minoli, M. (2020, octubre 17). *Adaptación de Scrum para el desarrollo de soluciones de Business Analytics con CRISP-DM*. Hiberus. <https://www.hiberus.com/crecemos-contigo/adaptacion-de-scrum-para-business-analytics/>
11. MyTaskPanel. (n.d.). *La metodología CRISP-DM: Desarrollo de modelos de machine learning*. MyTaskPanel. <https://www.mytaskpanel.com/la-metodologia-crisp-dm-desarrollo-de-modelos-de-machine-learning/>
12. Olivera Cokan, C. A. (2024). *Marco de trabajo para el desarrollo de proyectos de analítica de datos* [Tesis de maestría, Pontificia Universidad Católica del Perú]. Repositorio PUCP. <https://tesis.pucp.edu.pe/server/api/core/bitstreams/bb1d219a-4612-43ac-b8bd-bd47e1c74438/content>
13. Palomino Salazar, E. (2024). *Metodología Scrum en la gestión de proyectos de tecnologías de información en una entidad financiera, Lima 2024* [Tesis de maestría, Universidad César Vallejo]. Repositorio UCV. https://repositorio.ucv.edu.pe/bitstream/handle/20.50012692/159302/Palomino_SE-SD.pdf?sequence=1&isAllowed=y
14. Pinto, J. S., & Leme, R. S. (2024). Analysis of project management principles with the Scrum framework in systems development: A case study in a public organization. *Brazilian Journal of Operations and Production Management*, 21(2), e20241878. <https://doi.org/10.14488/BJOPM.1878.2024>
15. Pariona Leonardo, J. A. (2021). *Sistema web mediante Scrum para la mejora de la eficiencia operativa en el proceso de gestión de software de*

la empresa Triad Agencia de Marketing, Huancayo - Perú [Tesis de licenciatura, Universidad Nacional del Centro del Perú]. Repositorio UNCP.

https://repositorio.uncp.edu.pe/bitstream/handle/20.500.12894/7347/T010_72564084_T.pdf?sequence=1&isAllowed=y

16. Reyes-Riveros, A. J., Castillo-Sarmiento, J. M. H., Santos-Fernández, J. P., Alcántara-Moreno, O. R., & Sánchez-Ticona, R. J. (2024). Sistema web para la dinamización de la gestión de inventario y estrategias de marketing en supermercados peruanos. *Revista Científica de Sistemas e Informática*, 4(2), e673. <https://doi.org/10.51252/rcsi.v4i2.673>
17. Rojas Pecho, P. Y., & Acevedo Mortola, M. A. (2025). *Propuesta de diseño de un modelo de machine learning para estimar costos en proyectos de software* [Tesis de maestría, Universidad Peruana de Ciencias Aplicadas]. Repositorio UPC. <http://hdl.handle.net/10757/672209>
18. Sánchez, M. G., & Pérez, J. Á. (2023). *Metodología CRISP-DM en la gestión de proyecto de Data Mining. Caso enfermedades dermatológicas*. [Tesis de licenciatura, Universidad Autónoma de Nuevo León]. <http://hdl.handle.net/10882/13087>
19. Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Müller, K.-R. (2020). *Towards CRISP-ML(Q): A machine learning process model with quality assurance methodology*. arXiv. <https://arxiv.org/abs/2003.05155>
20. Toledo González, N., Ríos Gómez, M. S., Requena Hernandez, L. E., & Leal Vazquez, N. L. (2023). Desarrollo de un software de inventarios utilizando la metodología SCRUM. *TECTZAPIC Revista Académico-Científica*, 9, 10-21. <https://doi.org/10.51896/tectzapic/RUCS4253>
21. Xool, J., Buenfil, H., y Dzul, M. (2018). Desarrollo e implementación de un sistema web para el proceso de estadía. *Revista de Tecnologías de la Información y Comunicaciones*, 2(3), 8–19. https://www.ecorfan.org/spain/researchjournals/Tecnologias_de_la_Informacion_y_Comunicaciones/vol2num3/Revista_de_Tecnologia_de_la_Inf

22. Espinosa, R. Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web. *Revistas CEDAMAZ*, 11(2), 133-141.
<https://revistas.unl.edu.ec/index.php/cedamaz/article/view/1182/853>
ISSN: 1390-5902
23. Cohn, M. (2010). *Succeeding with agile: Software development using Scrum* (1st ed.). Addison-Wesley.
https://api.pageplace.de/preview/DT0400.9780321660510_A23552372/preview-9780321660510_A23552372.pdf
24. Agile Spain. (2024). *Comunidad Agile en España*. <https://agile-spain.org>
25. ENAE Business School. (2023). *Metodología agile en la empresa*.
<https://www.enaes.es/blog/metodologia-agile-en-la-empresa>
26. Promethean World. (2023). *7 aplicaciones educativas para el aula*.
<https://www.prometheanworld.com/es/recursos/blogs/7-aplicaciones-educativas-para-el-aula/>
27. IBM. (2022). *Mobile application development*. <https://www.ibm.com/mx-es/topics/mobile-application-development>
28. Schwaber, K., & Sutherland, J. (2020). *The Scrum guide*. Scrum.org.
<https://www.scrum.org/resources/scrum-guide>
29. Rubin, K. (2012). *Essential scrum: A practical guide to the most popular agile process* (1st ed.). Addison-Wesley.
<https://www.amazon.com/Essential-Scrum-Practical-Addison-Wesley-Signature/dp/0137043295>
30. Gaikwad, M., Ahirrao, S., Phansalkar, S., Kotecha, K., & Rani, S. (2023). Multi-ideology, multiclass online extremism dataset, and its evaluation using machine learning. *Computational Intelligence and Neurosciences*, 2023, 4563145. <https://doi.org/10.1155/2023/4563145>
31. Big Data Marketer. (2018, 1 de octubre). ¿Qué es el análisis predictivo? Big Data Social. <https://www.bigdata-social.com/que-es-el-analisispredictivo/>

32. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. The CRISP-DM Consortium. <https://www.semanticscholar.org/paper/54bad20bbc7938991bf34f86dde0babfbd2d5a72>
33. Hernández-Orallo, J. (2021). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8), 3048–3061. <https://ieeexplore.ieee.org/abstract/document/8943998>
34. Kunaifi, A., & Akbar, A. N. F. (2019). Financial knowledge and millennials investing behaviors in Indonesia. *Jurnal Sosial Humaniora*, 0(1), 26. <https://doi.org/10.12962/j24433527.v0i01.5768>
35. Idris, N. H., Ismail, S., & Samad, K. A. (2018). Financial knowledge, debt literacy and over-indebtedness. *The Journal of Social Sciences Research, Special Issue 3*, 22–29. <https://doi.org/10.32861/jssr.spi3.22.29>
36. Forbes Finance Council. (2020, 14 de abril). *Don't overlook these 15 important financial health habits*. Forbes. <https://www.forbes.com/sites/forbesfinancecouncil/2020/04/14/dont-overlook-these-15-important-financial-health-habits/>
37. Balcan, M.-F., & Sharma, D. (2024). *Learning accurate and interpretable decision trees*. arXiv. <https://arxiv.org/abs/2405.15911>
38. Budiawan, J. C. (2025, enero 19). *XGBoost explained: The go-to algorithm for predictive modeling*. Medium. <https://medium.com/@jeremychrist23/xgboost-explained-the-go-to-algorithm-for-predictive-modeling-9e7a7c2036c3>
39. Azis, A., Dinata, R. K., & Hasdyna, N. (2024). Application of the K-Nearest Neighbor machine learning algorithm to predict sales of best-selling products. *Brilliance: Research Journal*, 4(1), 256. <https://doi.org/10.47709/brilliance.v4i1.4063>
40. Negri, R. G. (2023). Support vector machines. En B. S. Daya Sagar, Q. Cheng, J. McKinley, & F. Agterberg (Eds.), *Encyclopedia of Mathematical*

Geosciences (pp. 1520–1525). Springer. https://doi.org/10.1007/978-3-030-85040-1_323

41. Attanasi, E. D., & Coburn, T. C. (2023). Random Forest. En B. S. Daya Sagar, Q. Cheng, J. McKinley, & F. Agterberg (Eds.), *Encyclopedia of Mathematical Geosciences* (pp. 1182–1185). Springer. https://doi.org/10.1007/978-3-030-85040-1_265
42. Fynd Academy. (2025). *Linear Regression in Machine Learning*. Fynd Academy Blog. <https://www.fynd.academy/blog/linear-regression-in-machine-learning>
43. Labhade-Kumar, N. A., Jangale, L., Sathe, V., Shelke, A., & Redij, T. (2024). *Study of Supervised Logistic Regression Algorithm*. ResearchGate. https://www.researchgate.net/publication/385985292_Study_of_Supervised_Logistic_Regression_Algorithm
44. Amazon Web Services. (2024). *¿Cuál es la diferencia entre las bases de datos relacionales y no relacionales?* AWS. Recuperado de <https://aws.amazon.com/es/compare/the-difference-between-relational-and-non-relational-databases/>
45. Codd, E. F. (1970). *A relational model of data for large shared data banks*. Communications of the ACM, 13(6), 377–387. <https://doi.org/10.1145/362384.362685>
46. Google Cloud. (s.f.). *¿Qué es una base de datos relacional (RDBMS)?*. Google Cloud. Recuperado de <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>
47. IBM. (2021). *¿Qué es PostgreSQL?* IBM. Recuperado de <https://www.ibm.com/mx-es/topics/postgresql>
48. Kraut, N., & Transchel, F. (2022). On the application of SCRUM in data science projects. *Proceedings of the 7th International Conference on Big Data Analytics (ICBDA)*. <https://doi.org/10.1109/ICBDA55095.2022.9760341>
49. Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., Ramírez-Quintana, M. J., & Flach, P. (2019).

CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8), 3048–3061.
<https://doi.org/10.1109/TKDE.2019.2962680>

50. Amirian, E., Abdollahzadeh, A., & Sulaiman, N. (2024). Synergizing hybrid agile-SCRUM and CRISP-DM approaches in data science project management. *SPE Canadian Energy Technology Conference*.
<https://doi.org/10.2118/218114-MS>
51. Para el concepto específico de Data Driven Scrum (DDS) también puedes apoyarte en fuentes como: Data Science PM (2021). 5 key questions when using DDS. Recuperado de <https://www.datascience-pm.com/5-key-questions-when-using-dds/>
52. Amirian, E., Abdollahzadeh, A., & Sulaiman, N. (2024). Synergizing hybrid agile-SCRUM and CRISP-DM approaches in data science project management. *SPE Canadian Energy Technology Conference*.
<https://doi.org/10.2118/218114-MS>

Anexos

Anexo A: Fragmentos de Código Relevantes del Proceso de Modelado

A.1 Funciones Heurísticas (Sprint 2)

[12]: *# Función para clasificar el nivel de ahorro*

```
def categorizar_nivel(x, ingreso):  
    nivel = (x / ingreso) * 100  
    if nivel <= 8:  
        return 1  
    elif 8 < nivel <= 12:  
        return 2  
    elif 12 < nivel <= 20:  
        return 3  
    elif 20 < nivel <= 30:  
        return 4  
    elif 30 < nivel <= 40:  
        return 5  
    else:  
        return 6  
  
def categorizar_frecuencia(frecuencia):  
    if frecuencia == 'NUNCA':  
        return 1  
    elif frecuencia == 'CASI NUNCA':  
        return 2  
    elif frecuencia == 'ALGUNAS VECES':  
        return 3  
    elif frecuencia == 'FRECUENTE':  
        return 4  
    elif frecuencia == 'MUY FRECUENTE':  
        return 5
```

A.2 Configuración de Optimización (Sprint 3 - Modelo 1 Multiclase)

```
[31]: # Importamos las librerías necesarias
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder

# Creamos el codificador de etiquetas
le = LabelEncoder()

# Ajustamos las etiquetas para que empiecen desde 0
y_train_encoded = le.fit_transform(y_train)
y_test_encoded = le.transform(y_test)

# 1. Manejo de desbalance con SMOTE
smote = SMOTE(sampling_strategy='auto', random_state=42)

# Aplicamos SMOTE a los datos de entrenamiento
X_res, y_res = smote.fit_resample(X_train, y_train_encoded)

# 2. Ajuste de hiperparámetros con RandomizedSearchCV
# Definimos el espacio de búsqueda para los hiperparámetros de XGBoost
param_dist = {
    'max_depth': [3, 6, 10, 15],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'n_estimators': [50, 100, 200],
    'subsample': [0.7, 0.8, 0.9, 1.0],
    'colsample_bytree': [0.7, 0.8, 0.9, 1.0],
    'min_child_weight': [1, 3, 5, 7]
}

# Realizamos la búsqueda aleatoria de hiperparámetros
xgb_random_search = RandomizedSearchCV(XGBClassifier(),
    param_distributions=param_dist, n_iter=100, cv=5, random_state=42, n_jobs=-1)
```

A.3 Configuración de Optimización (Sprint 4 - Modelo 2 Binario)

```
[30]: from sklearn.model_selection import GridSearchCV
      from xgboost import XGBClassifier

      # Definir los hiperparámetros a buscar
      param_grid = {
          'learning_rate': [0.01, 0.05, 0.1, 0.2],
          'n_estimators': [100, 200, 300],
          'max_depth': [3, 5, 7],
          'min_child_weight': [1, 3, 5],
          'subsample': [0.8, 1.0],
          'colsample_bytree': [0.8, 1.0]
      }

      # Inicializar el clasificador
      model_xgb = XGBClassifier(eval_metric='mlogloss', use_label_encoder=False)

      # Inicializar GridSearchCV
      grid_search = GridSearchCV(estimator=model_xgb, param_grid=param_grid,
                                scoring='accuracy', cv=5, verbose=1, n_jobs=-1)
```

```

# Importamos las librerías necesarias
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder

# Creamos el codificador de etiquetas
le = LabelEncoder()

# Ajustamos las etiquetas para que empiecen desde 0
y_train_encoded = le.fit_transform(y_train)
y_test_encoded = le.transform(y_test)

# 1. Manejo de desbalance con SMOTE
smote = SMOTE(sampling_strategy='auto', random_state=42)

# Aplicamos SMOTE a los datos de entrenamiento
X_res, y_res = smote.fit_resample(X_train, y_train_encoded)

# 2. Ajuste de hiperparámetros con RandomizedSearchCV
# Definimos el espacio de búsqueda para los hiperparámetros de XGBoost
param_dist = {
    'max_depth': [3, 6, 10, 15],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'n_estimators': [50, 100, 200],
    'subsample': [0.7, 0.8, 0.9, 1.0],
    'colsample_bytree': [0.7, 0.8, 0.9, 1.0],
    'min_child_weight': [1, 3, 5, 7]
}

# Realizamos la búsqueda aleatoria de hiperparámetros
xgb_random_search = RandomizedSearchCV(XGBClassifier(),
param_distributions=param_dist, n_iter=100, cv=5, random_state=42,
n_jobs=-1)
xgb_random_search.fit(X_res, y_res)

```

```

from sklearn.model_selection import GridSearchCV
from xgboost import XGBClassifier

# Definir los hiperparámetros a buscar
param_grid = {
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 5, 7],
    'min_child_weight': [1, 3, 5],
    'subsample': [0.8, 1.0],
    'colsample_bytree': [0.8, 1.0]
}

# Inicializar el clasificador
model_xgb = XGBClassifier(eval_metric='mlogloss',
use_label_encoder=False)

# Inicializar GridSearchCV
grid_search = GridSearchCV(estimator=model_xgb, param_grid=param_grid,
scoring='accuracy', cv=5, verbose=1,
n_jobs=-1)

```

Anexo B: Evaluación de Rendimiento de Modelos Baseline (Propensión al Ahorro)

B.1 Resultados Baseline - Modelo 1 (Nivel de Hábito Financiero)

B.1.1 Regresión Logística

Classification Report Logistic Regression:

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

1	0.74	0.74	0.74	111
2	0.51	0.64	0.57	152
3	0.33	0.22	0.26	79
4	0.53	0.37	0.44	78
5	0.73	0.86	0.79	80

accuracy			0.59	500
macro avg	0.57	0.57	0.56	500
weighted avg	0.57	0.59	0.57	500

```
C:\Users\gcpp.gpc05\AppData\Roaming\Python\Python313\site-packages\
sklearn\linear_model\logistic.py:473: ConvergenceWarning: lbfgs
failed to converge after 1000 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT
```

Increase the number of iterations to improve the convergence
(max_iter=1000).

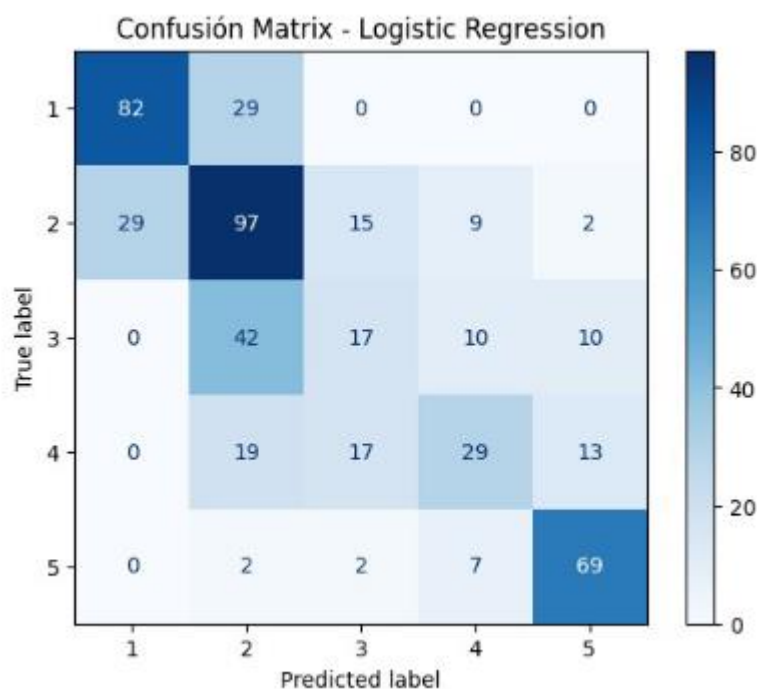
You might also want to scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```

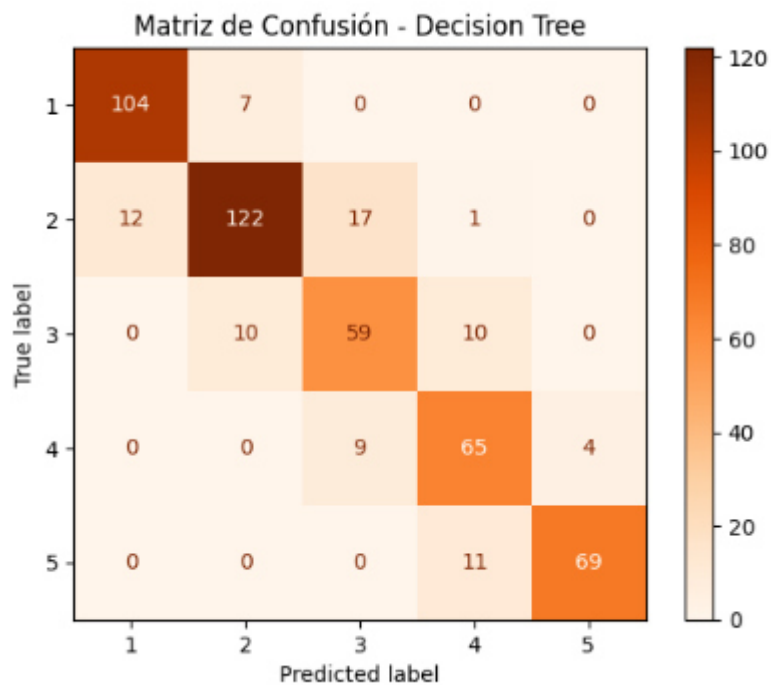


B.1.2 Árbol de Decisión

Accuracy Decision Tree: 0.838

Classification Report Decision Tree:

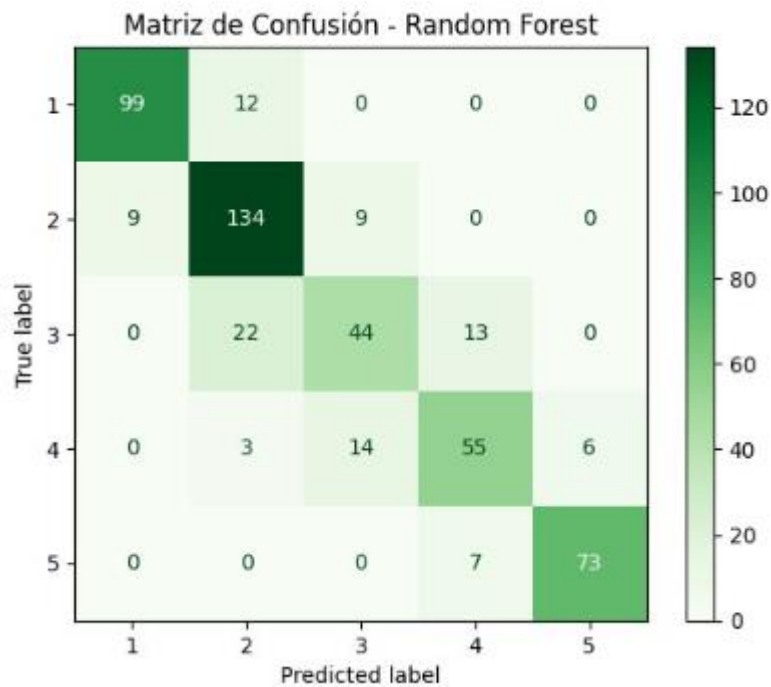
		precision	recall	f1-score	support
	1	0.90	0.94	0.92	111
	2	0.88	0.80	0.84	152
	3	0.69	0.75	0.72	79
	4	0.75	0.83	0.79	78
	5	0.95	0.86	0.90	80
accuracy				0.84	500
macro avg		0.83	0.84	0.83	500
weighted avg		0.84	0.84	0.84	500



B.1.3 Random Forest

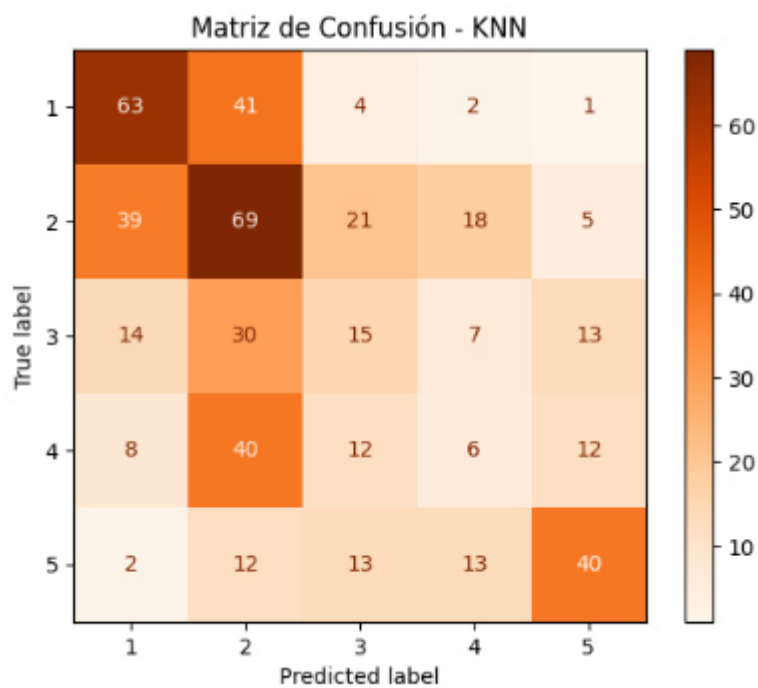
Accuracy Random Forest: 0.81

Classification Report Random Forest:					
		precision	recall	f1-score	support
	1	0.92	0.89	0.90	111
	2	0.78	0.88	0.83	152
	3	0.66	0.56	0.60	79
	4	0.73	0.71	0.72	78
	5	0.92	0.91	0.92	80
accuracy				0.81	500
macro avg		0.80	0.79	0.79	500
weighted avg		0.81	0.81	0.81	500



B.1.4 K-Nearest Neighbors (KNN)

Classification Report KNN:				
	precision	recall	f1-score	support
1	0.50	0.57	0.53	111
2	0.36	0.45	0.40	152
3	0.23	0.19	0.21	79
4	0.13	0.08	0.10	78
5	0.56	0.50	0.53	80
accuracy			0.39	500
macro avg	0.36	0.36	0.35	500
weighted avg	0.37	0.39	0.37	500



B.1.5 Support Vector Machine (SVM)

Accuracy SVM: 0.486

Classification Report SVM:

	precision	recall	f1-score	support
1	0.61	0.59	0.60	111
2	0.43	0.74	0.55	152
3	0.00	0.00	0.00	79
4	0.00	0.00	0.00	78
5	0.52	0.81	0.63	80
accuracy			0.49	500
macro avg	0.31	0.43	0.36	500
weighted avg	0.35	0.49	0.40	500

```
C:\Users\gcpp.gpc05\AppData\Roaming\Python\Python313\site-packages\
sklearn\metrics\_classification.py:1731: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this
behavior.
```

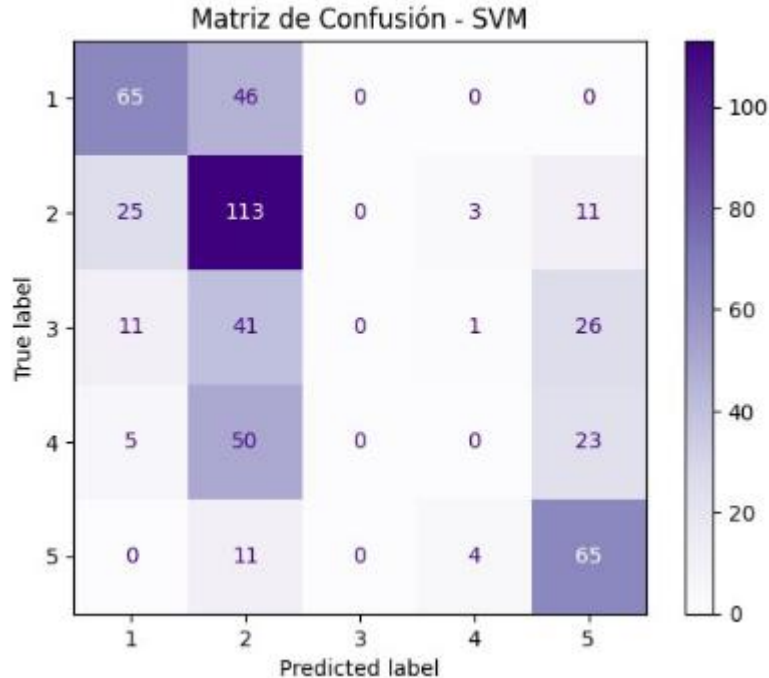
```
warn_prf(average, modifier, f"{metric.capitalize()} is",
result.shape[0])
```

```
C:\Users\gcpp.gpc05\AppData\Roaming\Python\Python313\site-packages\
sklearn\metrics\_classification.py:1731: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this
behavior.
```

```
warn_prf(average, modifier, f"{metric.capitalize()} is",
result.shape[0])
```

```
C:\Users\gcpp.gpc05\AppData\Roaming\Python\Python313\site-packages\
sklearn\metrics\_classification.py:1731: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this
behavior.
```

```
warn_prf(average, modifier, f"{metric.capitalize()} is",
result.shape[0])
```

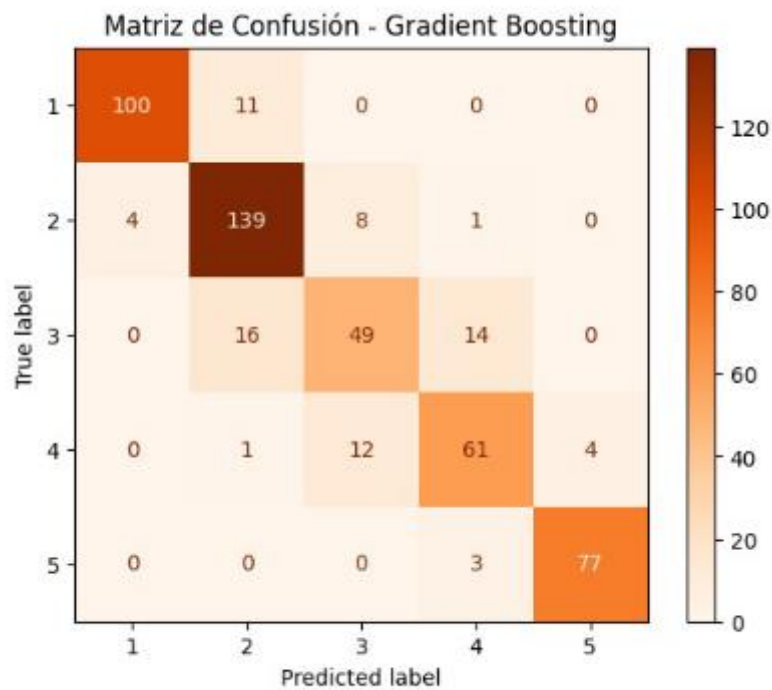


B.1.6 Gradient Boosting

Accuracy Gradient Boosting: 0.852

Classification Report Gradient Boosting:

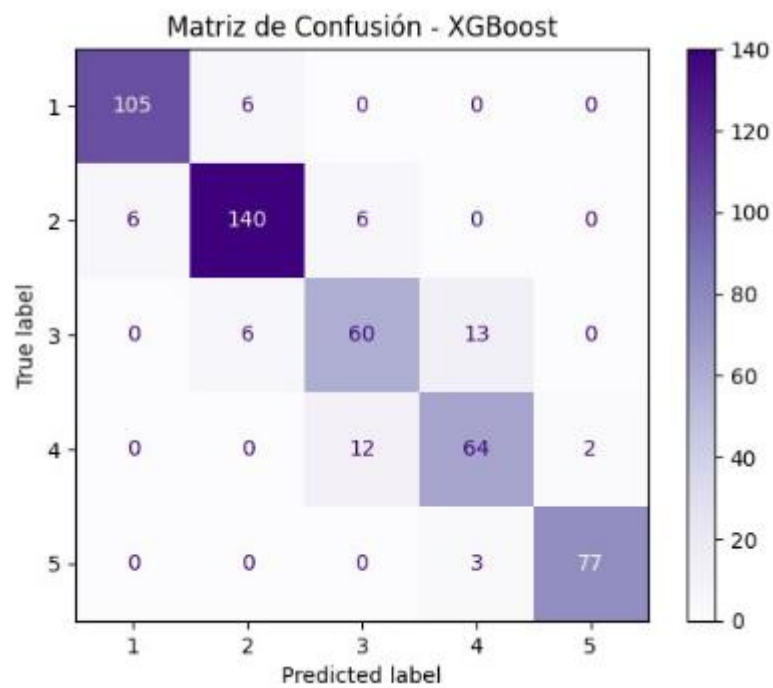
	precision	recall	f1-score	support
1	0.96	0.90	0.93	111
2	0.83	0.91	0.87	152
3	0.71	0.62	0.66	79
4	0.77	0.78	0.78	78
5	0.95	0.96	0.96	80
accuracy			0.85	500
macro avg	0.85	0.84	0.84	500
weighted avg	0.85	0.85	0.85	500



XGBoost Classifier

B.1.7 XGBoost

Classification Report XGBoost:					
		precision	recall	f1-score	support
	0	0.95	0.95	0.95	111
	1	0.92	0.92	0.92	152
	2	0.77	0.76	0.76	79
	3	0.80	0.82	0.81	78
	4	0.97	0.96	0.97	80
	accuracy			0.89	500
	macro avg	0.88	0.88	0.88	500
	weighted avg	0.89	0.89	0.89	500



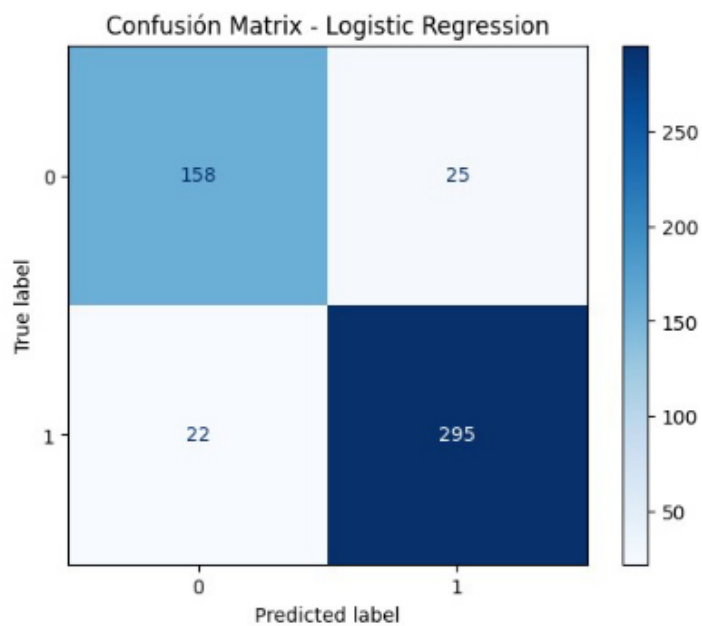
B.2 Resultados Baseline - Modelo 2 (Propensión al Ahorro)

B.2.1 Regresión Logística

Accuracy Logistic Regression: 0.906				
Classification Report Logistic Regression:	precision	recall	f1-score	support
0	0.88	0.86	0.87	183
1	0.92	0.93	0.93	317
accuracy			0.91	500
macro avg	0.90	0.90	0.90	500
weighted avg	0.91	0.91	0.91	500

C:\Users\gcpp.gpc05\AppData\Roaming\Python\Python313\site-packages\sklearn\linear_model\logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 1000 iteration(s) (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

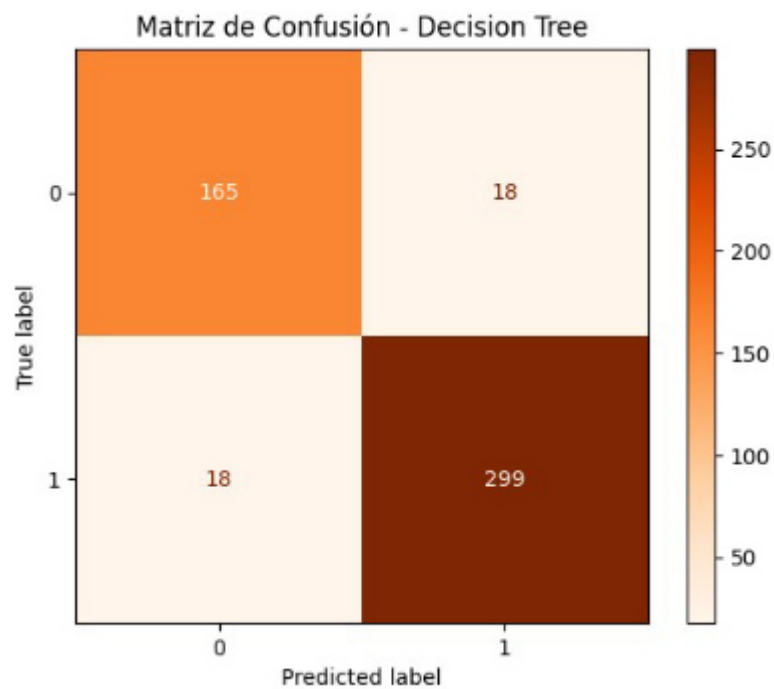
Increase the number of iterations to improve the convergence (max iter=1000).
You might also want to scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(



B.2.2 Árbol de Decisión

Accuracy Decision Tree: 0.928

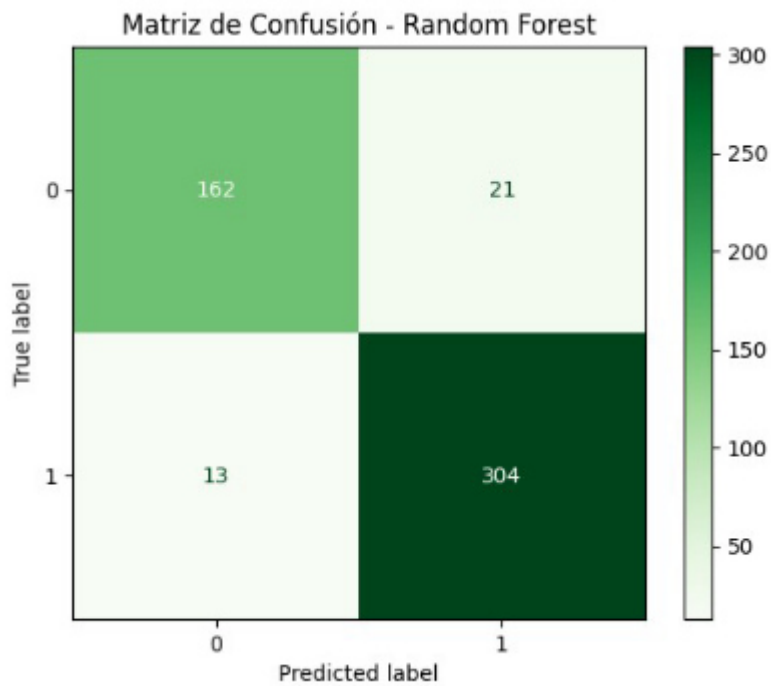
Classification Report Decision Tree:					
	precision	recall	f1-score	support	
0	0.90	0.90	0.90	183	
1	0.94	0.94	0.94	317	
accuracy			0.93	500	
macro avg	0.92	0.92	0.92	500	
weighted avg	0.93	0.93	0.93	500	



B.2.3 Random Forest

Accuracy Random Forest: 0.932

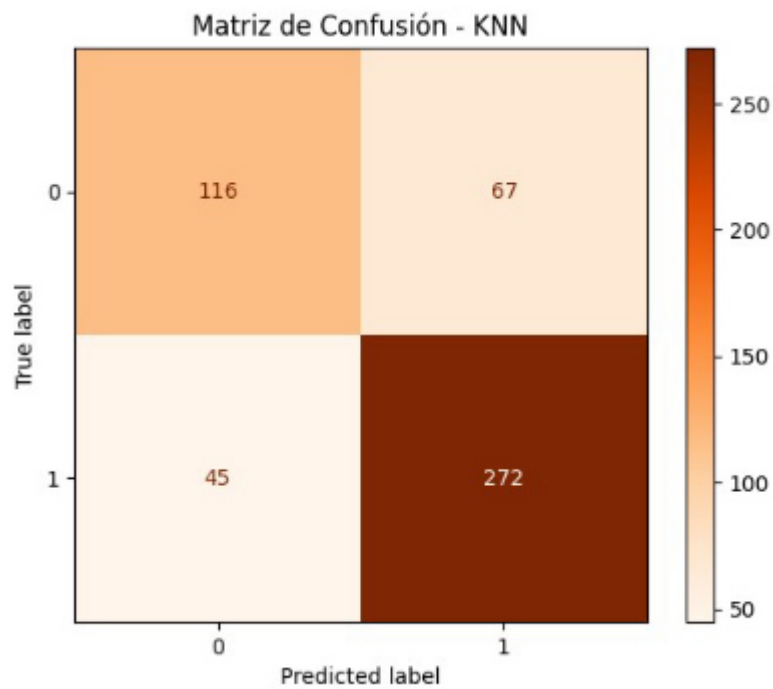
Classification Report Random Forest:					
	precision	recall	f1-score	support	
0	0.93	0.89	0.91	183	
1	0.94	0.96	0.95	317	
accuracy			0.93	500	
macro avg	0.93	0.92	0.93	500	
weighted avg	0.93	0.93	0.93	500	



B.2.4 K-Nearest Neighbors (KNN)

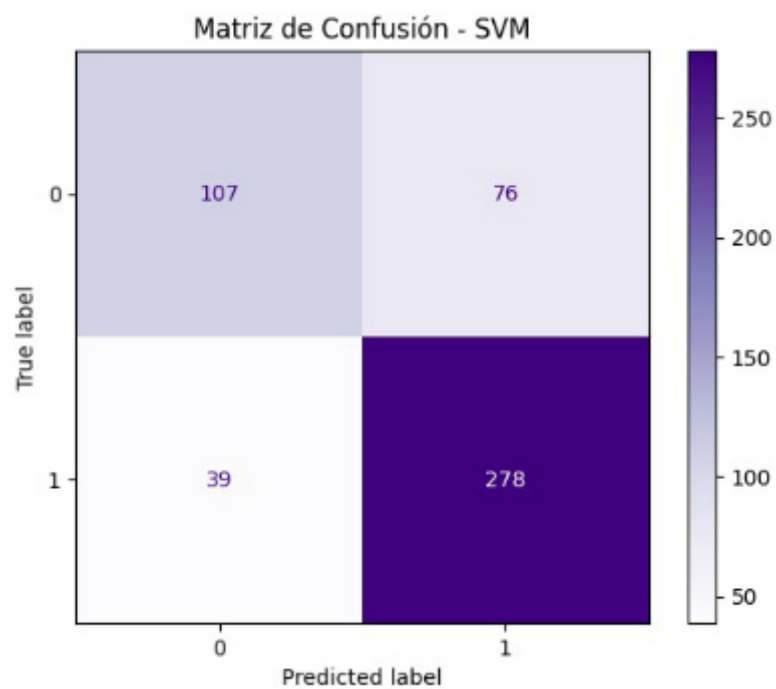
Accuracy KNN: 0.776

Classification Report KNN:					
		precision	recall	f1-score	support
	0	0.72	0.63	0.67	183
	1	0.80	0.86	0.83	317
accuracy				0.78	500
macro avg		0.76	0.75	0.75	500
weighted avg		0.77	0.78	0.77	500



B.2.5 Support Vector Machine (SVM)

Accuracy SVM: 0.77				
Classification Report SVM:				
	precision	recall	f1-score	support
0	0.73	0.58	0.65	183
1	0.79	0.88	0.83	317
accuracy			0.77	500
macro avg	0.76	0.73	0.74	500
weighted avg	0.77	0.77	0.76	500

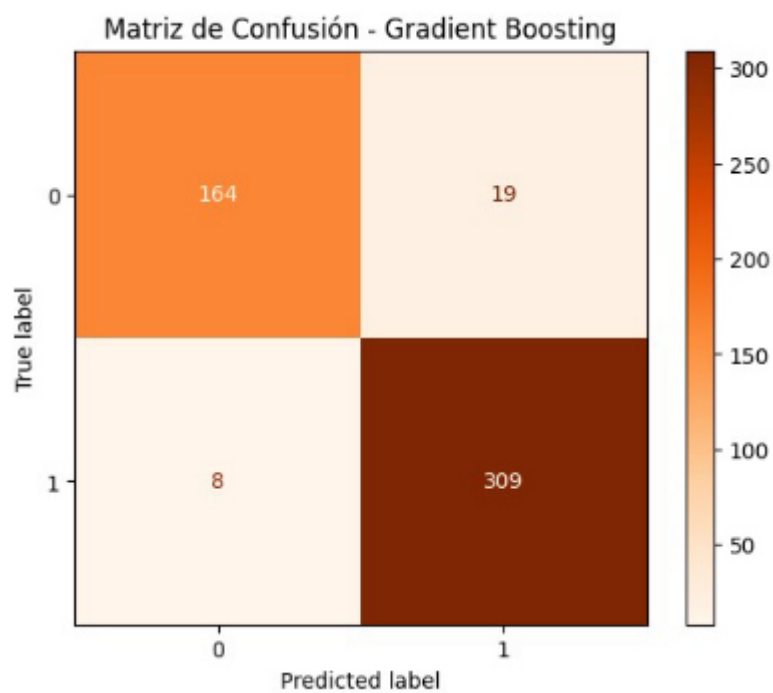


B.2.6 Gradient Boosting

Accuracy Gradient Boosting: 0.946

Classification Report Gradient Boosting:

	precision	recall	f1-score	support
0	0.95	0.90	0.92	183
1	0.94	0.97	0.96	317
accuracy			0.95	500
macro avg	0.95	0.94	0.94	500
weighted avg	0.95	0.95	0.95	500



B.2.7 XGBoost

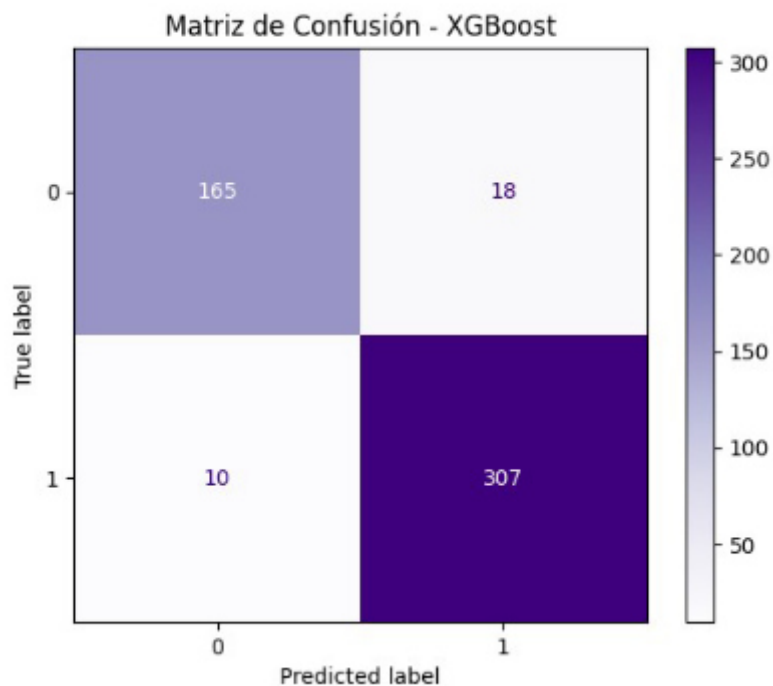
Accuracy XGBoost: 0.944

Classification Report XGBoost:

	precision	recall	f1-score	support
0	0.94	0.90	0.92	183
1	0.94	0.97	0.96	317
accuracy			0.94	500
macro avg	0.94	0.94	0.94	500
weighted avg	0.94	0.94	0.94	500

```
C:\Users\gcpp.gpc05\AppData\Roaming\Python\Python313\site-packages\
xgboost\training.py:183: UserWarning: [14:18:02] WARNING: C:\actions-
runner\_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "use_label_encoder" } are not used.
```

```
bst.update(dtrain, iteration=i, fobj=obj)
```



Anexo C: Configuración del Entorno de Despliegue del Sistema "FinSight"

El sistema se encuentra desplegado y funcional en la siguiente dirección:

* URL: <https://quantumtechpe.com>

Detalles del entorno de producción:

* Proveedor de Hosting:

* Servidor:

* Pila Tecnológica:

Anexo D: Guía Rápida de Usuario para "FinSight"

Este anexo describe el flujo de uso principal del sistema web "FinSight", desplegado en <https://quantumtechpe.com>.

D.1 Acceso e Inicio de Sesión

El usuario accede a la plataforma a través de la URL pública. La primera pantalla es el formulario de autenticación. El usuario debe ingresar su nombre de usuario y contraseña (previamente registrados) y hacer clic en "Iniciar Sesión".



Figura D.1: Interfaz de inicio de sesión del sistema "FinSight". Fuente: Elaboración propia.

D.2 Ingreso de Datos para Predicción

Una vez autenticado, el usuario es dirigido a la página principal de "Predicción de Hábitos Financieros". En esta interfaz, el usuario debe completar el formulario con sus datos sociodemográficos y financieros, los cuales corresponden a las variables de entrada del modelo.

Información Personal

Edad *
Ingresar tu edad

Género *
Selecciona tu género

Nivel de Educación *
Selecciona tu nivel de educación

Información Financiera

Ingreso Mensual (\$/)*
Ingreso entre \$/ 500 - \$/ 50,000

Ahorro Mensual (\$/)*
Ingresar tu ahorro mensual

Endeudamiento Mensual (\$/)*
Ingresar tu endeudamiento mensual

Gasto en Ocio (\$/)*
Ingresar tu gasto en ocio

Tarjetas de Crédito y Hábitos

Número de Tarjetas de Crédito *
Número de tarjetas (0-10)

Frecuencia de Uso de Tarjeta *
¿Con qué frecuencia usas tarjeta?

¿Presupuestas tus gastos regularmente? *
¿Con qué frecuencia presupuestas?

¿Sueles pagar solo el monto mínimo de la tarjeta? *
¿Con qué frecuencia pagas el mínimo?

Limpiar Predecir Hábitos Financieros

© Jonathan Adrian Centi Portugal - 2025 - Versión: 1.0.0.05092025

Figura D.2: Formulario de ingreso de datos para predicción. *Fuente: Elaboración propia.*

D.3 Visualización de Resultados

Tras completar el formulario y hacer clic en "Iniciar Predicción", el sistema procesa los datos a través de los modelos de Machine Learning (alojados en el backend) y devuelve los resultados. La interfaz muestra las dos predicciones generadas:

1. **Resultado 1:** El "Nivel de Hábito Financiero" (clasificación multiclase, ej. Nivel 1 a 5).
2. **Resultado 2:** La "Propensión al Ahorro" (clasificación binaria, ej. Alta/Baja).



Figura D.3: Visualización de resultados de la predicción. *Fuente: Elaboración propia.*

D.4 Consulta de Historial y Módulos Adicionales

El sistema también permite al usuario navegar a otras secciones a través del menú de navegación, tales como:

- **Historial de Resultados:** Una vista donde el usuario puede consultar todas las predicciones que ha realizado anteriormente.
- **Chat de Ayuda / Encuesta:** Módulos complementarios para soporte y recolección de feedback.

Historial de Predicciones

Revisa todas tus predicciones financieras anteriores y su evolución en el tiempo

Actualizar Total: 2 predicciones Buscar...

Fecha	Datos Personales	Datos Financieros	Hábitos Financieros	Propensión al Ahorro	Confianza
10/09/2025 05:38	Edad: 25 años Género: M Educación: Técnica trunca	Ingreso: S/ 1,500.00 Ahorro: S/ 250.00 Deuda: S/ 250.00 Tarjetas: 2	Nivel 1 - Muy Bajo	Baja Propensión	Hábitos: 95.0% Propensión: 99.5%
09/09/2025 21:25	Edad: 25 años Género: M Educación: Primaria completa	Ingreso: S/ 1,500.00 Ahorro: S/ 25.00 Deuda: S/ 25.00 Tarjetas: 1	Nivel 1 - Muy Bajo	Baja Propensión	Hábitos: 97.4% Propensión: 92.3%

© Jonathan Adrian Centi Portugal - 2025 - Versión: 1.0.0.05092025

Figura D.4: Interfaz del historial de predicciones del usuario. *Fuente: Elaboración propia.*