

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“SINTONIZACIÓN DE METAHEURÍSTICAS MEDIANTE
APRENDIZAJE AUTOMÁTICO PARA PROBLEMAS DE
RUTEO DE VEHÍCULOS MULTI-PRODUCTO CON
MEZCLAS.”

CARLOS VEGA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Elizabeth Montero
Profesor Correferente: Nicolás Rojas-Morales

Julio - 20XX



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: Sintonización de metaheurísticas mediante aprendizaje automático para problemas de ruteo de vehículos multi-producto con mezclas

Nombre del candidato(a): Carlos Javier Alejandro Vega Muñoz

Carrera / Grado: Ingeniería civil informática

Campus: ; Santiago San Joaquín Departamento: Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Elizabeth Montero, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO** contiene información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 13/08/25

; Firma:

Estudiante o Candidato(a):

Fecha: 13/08/25

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Dedico esta memoria especialmente a mi madre Verónica, gracias por estar incondicionalmente y todo el amor que me has dado. A mi padre Carlos por su apoyo, guía, que a pesar de la distancia siempre hemos mantenido nuestro vínculo y amor.

A mi hermano Ronny por motivarme a ser ingeniero, despertar mi curiosidad de pequeño por los astros y la ciencia. A mi hermano Emerson por enseñarme lo verdaderamente importante en la vida. A mi novia Antonia por su corazón gentil, el apoyo incondicional durante estos años de universidad y el bello amor que tenemos. A mi hermano Carlos Michel por su gran corazón y cariño. A mi abuela Lucia por siempre estar por nosotros y ser un pilar en nuestra familia. A mi abuela Aida por su corazón dulce y creer en mí.

A Astrid por acogerme en su familia como uno más de sus hijos en un momento de mi vida.

A mi tía Marianela por estar siempre por su familia y entregarnos mucho amor, A mi tío Ronny por ayudar a sus hermanas y unir a la familia. A mis dos tíos Rubén por ser los únicos de sus madres, entregar amor y risas a sus familias.

A mis tíos Alexander y Michel por el cariño que siempre me han dado. A mi prima Andrea, Raymond por su cariño y permanente apoyo.

A mis primos grandes y chicos por ser buenas personas y continuar el legado de nuestra familia. A mi familia cubana por siempre recibirme como uno más de ellos. A mis amigos y conocidos por hacer el camino más ameno y divertido. A los que se fueron y vendrán.

AGRADECIMIENTOS

Agradezco a mis padres por todo su sacrificio, trabajo y amor. No lo podía haber logrado sin su apoyo. Agradezco a la profesora Elizabeth por ser una buena tutora, mantenerse positiva y transmitirme confianza a lo largo de todo el proceso.

Agradezco a mi mánager Miguel por siempre priorizar mi carrera, otorgarme la flexibilidad necesaria para cursar mis estudios y ser un excelente modelo a seguir.

Al profesor Camilo por darme tips acerca de la implementación de los modelos y enseñar con pasión. Al profesor Arroyuelo por despertar en mi interés por los algoritmos, a todos los profesores que me formaron durante mi vida.

Agradezco a todos los que creyeron en mí.

RESUMEN

Resumen— Esta memoria aborda la sintonización de parámetros para metaheurísticas que resuelven el Problema de Ruteo de Vehículos Multi-Producto con Mezcla Gradual (VRPMPWB), buscando validar si un modelo de aprendizaje automático puede predecir configuraciones efectivas para una metaheurística de Búsqueda Local Iterada (ILS). La metodología incluyó la generación de datos mediante el sintonizador irace y una ingeniería de atributos holística. Sobre estos datos, se entrenaron modelos de XGBoost y Redes Neuronales para las tareas de regresión y clasificación asociadas a la configuración del algoritmo.

Los resultados revelaron un rendimiento predictivo mixto: los modelos de regresión mostraron ser ineficaces, mientras que los de clasificación alcanzaron un desempeño moderado. A pesar de ello, al aplicar los parámetros predichos en instancias no vistas, se lograron mejoras marginales en la calidad de las soluciones para problemas con potencial de optimización, validando la hipótesis central. Este estudio demuestra que el aprendizaje automático es una alternativa viable para la sintonización de parámetros. Su relevancia aumenta en instancias de gran escala, donde los sintonizadores tradicionales se vuelven computacionalmente prohibitivos, ofreciendo un potencial significativo para la optimización logística en la industria.

Palabras Clave— Metaheuristica; Aprendizaje Automatico; Sintonizacion; Ingenieria de atributos; Optimizacion

ABSTRACT

Abstract— This thesis addresses parameter tuning for metaheuristics that solve the Vehicle Routing Problem with Multi-Product Blending (VRPMPWB), seeking to validate if a machine learning model can predict effective configurations for an Iterated Local Search (ILS) metaheuristic. The methodology included data generation via the irace tuner and comprehensive feature engineering. Using this data, XGBoost and Neural Network models were trained for the regression and classification tasks associated with the algorithm's configuration.

The results revealed mixed predictive performance: the regression models proved ineffective, while the classification models achieved moderate performance. Despite this, applying the predicted parameters to unseen instances led to marginal improvements in solution

quality for problems with optimization potential, validating the core hypothesis. This study demonstrates that machine learning is a viable alternative for parameter tuning. Its relevance grows for large-scale instances, where traditional tuners become computationally prohibitive, offering significant potential for industrial logistic optimization.

Keywords— Metaheuristic; Machine Learning; Sintonization; Feature Engineering; Optimization

GLOSARIO

Aquí se deben colocar las siglas mencionadas en el trabajo y su explicación, por orden alfabético. Por ejemplo:

ACO: Ant Colony Optimization
Adam: Adaptive Moment Estimation
ALNS: Adaptive Large Neighborhood Search
API: Application Programming Interface
CV: Coeficiente de Variacion
DI: Departamento de Informática.
FFNN: Feed Forward Neural Network
GDA: Gaussian discriminant analysis.
GRASP: Greedy randomized adaptive search procedure
IDSEQ: Identificador de la secuencia
ILS: Iterated Local Search.
ML: Machine Learning
MS-ILS: Multi Start Iterated Local Search
NP: Non Polynomial
OPTICS: Ordering Points To Identify the Clustering Structure
PERT: Numero de Perturbaciones
RCL: Restricted Candidate List
ReLU: Rectified Linear Unit
SSE: Sum Squared Error.
UTFSM: Universidad Técnica Federico Santa María.
VNS: Variable Neighborhood Search
VRP: Vehicle Routing Problem
VRPMPWB: Vehicle Routing Problem Multi Product With Blending
VRPWGMB: Vehicle Routing Problem With Gradual Milk Blending
XGBoost: Extreme Gradient Boosting

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	VI
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	X
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 El problema de ruteo de vehículos multiproducto con mezcla gradual	3
1.2 Metaheurísticas para el VRPMPWB	5
1.3 Sintonización de parámetros	6
1.3.1 Capa de Optimización del VRP	6
1.3.2 Capa de Predicción de Parámetros	6
1.4 Predicción de parámetros mediante aprendizaje automático	7
1.5 Árbol del Problema	8
CAPÍTULO 2: ESTADO DEL ARTE	9
2.1 El problema de ruteo de vehículos con mezclas (VRPMPWB)	9
2.2 Metaheurísticas para el VRPMPWB	10
2.3 Técnicas de Sintonización de Parámetros para Problemas de Ruteo de Vehículos	12
2.4 Predicción de Parámetros mediante Aprendizaje Automático Supervisado . .	14
2.4.1 XGBoost: Marco de Potenciación de Gradiente	15
2.4.2 Redes Neuronales Artificiales	17
2.5 Sintonización de Hiperparámetros en Modelos de Aprendizaje Automático . .	19
2.5.1 Métodos Tradicionales de Optimización	19
2.5.2 Optimización Bayesiana	19
2.5.3 Técnicas Metaheurísticas	19
2.5.4 Multi-dimensional Bagging	20
2.5.5 Consideraciones Técnicas	20
2.6 Resumen	20
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	22
3.1 Introducción	22
3.2 Selección de Metaheurísticas	22
3.3 Operadores de Búsqueda Local	23
3.4 Generación de Datos	24
3.5 Ingeniería de Atributos	26
3.5.1 Características de la instancia	26

3.5.2	Características espaciales	27
3.5.3	Características basadas en centroides	30
3.5.4	Características basadas en agrupamiento	33
3.5.5	Características de Mezclado	35
3.5.6	Características Eliminadas	38
3.6	Diseño Propuesto de Modelos de Aprendizaje Automático	40
3.6.1	Arquitectura General del Sistema	40
3.6.2	Diseño e implementación de modelos <i>XGBoost</i>	41
3.6.3	Diseño e implementación de la rama de redes neuronales	42
3.6.4	Consideraciones metodológicas: modelos multiclase binarios	43
3.7	Resumen	44
CAPÍTULO 4: EXPERIMENTOS Y RESULTADOS		45
4.1	Metaheurísticas objetivo	45
4.2	Configuración Experimental y Arquitecturas de Modelos	45
4.3	<i>XGBoost</i>	46
4.3.1	Análisis de hiperparámetros de <i>XGBoost</i>	46
4.3.2	Configuraciones de <i>XGBoost</i>	48
4.3.3	Importancia de Características en <i>XGBoost</i>	48
4.3.4	Modelo de Regresión para RCL	51
4.3.5	Modelo de Clasificación para PERT	52
4.3.6	Modelo de Características Binarias para la Predicción de Secuencia de Operadores	53
4.4	Redes Neuronales	56
4.4.1	Análisis de Hiperparámetros de Redes Neuronales	56
4.4.2	Configuraciones de Redes Neuronales (NN)	60
4.4.3	Modelo de Regresión para RCL	60
4.4.4	Modelo de Clasificación para PERT	62
4.4.5	Modelo de Características Binarias para la Predicción de Secuencia de Operadores	63
4.5	Resultados y análisis comparativo en instancias no vistas	64
4.6	Resumen	68
CAPÍTULO 5: CONCLUSIONES		69
5.1	Experimentos y Resultados	70
5.2	Limitaciones y Direcciones Futuras	72
ANEXOS		74
REFERENCIAS BIBLIOGRÁFICAS		75

ÍNDICE DE FIGURAS

1	Árbol del Problema: Sintonización de metaheurísticas con aprendizaje automático para el ruteo de vehículos multiproducto con mezcla.	8
2	Diseño inicial de la <i>pipeline</i> de sintonización de parámetros	24
3	Diseño de alto nivel de la validación de soluciones	25
4	Diseño de alto nivel de la ingeniería de atributos	26
5	Plano cartesiano de las 10 primeras instancias que conforman el conjunto de datos de entrenamiento con área de operación entrelineas.	32
6	Matriz de correlación perteneciente al conjunto de datos analizado.	39
7	Diseño de alto nivel del entrenamiento y optimización de los modelos de aprendizaje.	40
8	Diseño de alto nivel de la arquitectura particular de los modelos de aprendizaje automático.	43
9	Impacto de la tasa de aprendizaje (<code>learning_rate</code>) en las métricas de rendimiento de los modelos XGBoost.	46
10	Impacto del número de estimadores (<code>n_estimators</code>) en las métricas de rendimiento de los modelos XGBoost.	47
11	Impacto de la Profundidad Máxima del Árbol (<code>max_depth</code>) en las métricas de rendimiento de los modelos XGBoost.	47
12	Importancia de las características para el modelo XGBoost optimizado de predicción del parámetro RCL.	49
13	Importancia de las características para el modelo XGBoost optimizado de predicción del parámetro PERT.	50
14	Importancia de las características para cada una de las seis salidas del modelo XGBoost de características binarias.	51
15	Matriz de Confusión para el modelo XGBoost de predicción de PERT (conjunto de datos reducido).	54
16	Matrices de Confusión Agregadas para cada una de las seis Características Binarias predichas por el modelo XGBoost.	55
17	Impacto del número de capas ocultas en las métricas de rendimiento de las redes neuronales.	56
18	Impacto del número de neuronas en las capas ocultas respecto a las métricas de rendimiento de las redes neuronales.	57
19	Impacto de la función de activación en capas ocultas (ReLU, Tanh, Sigmoid) en las métricas de rendimiento de las redes neuronales.	58
20	Impacto de la tasa de aprendizaje en las métricas de rendimiento de las redes neuronales (RCL: MAE, PERT: Accuracy, Binarias: Hamming Loss). Arquitectura fija: [128, 64], Activación ReLU, 50 épocas, lote 64.	58
21	Impacto del número de épocas en las métricas de rendimiento de las redes neuronales (RCL: MAE, PERT: Accuracy, Binarias: Hamming Loss). Arquitectura fija: [128, 64], Activación ReLU, LR 0.001, lote 64.	59

22	Impacto del tamaño del lote (<i>batch size</i>) en las métricas de rendimiento de las redes neuronales (RCL: MAE, PERT: Accuracy, Binarias: Hamming Loss). Arquitectura fija: [128, 64], Activación ReLU, LR 0.001, 50 épocas. El eje X se presenta en escala logarítmica.	59
----	--	----

ÍNDICE DE TABLAS

1	<i>Características de modelado de problemas VRP para recolección de leche.</i> Adaptada de [López-Castro <i>et al.</i> , 2023]. MF: Multifábricas, HF: Flota heterogénea, MC: Multicompartimento, TT: Camión y tráiler, MP: Multiperiodo, TW: Ventanas de tiempo, PD: Recogida y entrega, GB: Mezcla global.	12
2	Características de la instancia derivadas de los datos brutos	27
3	Características espaciales	29
4	Características derivadas del agrupamiento	34
5	Características estáticas clave relacionadas con el mezclado	35
6	Métricas de rendimiento de los modelos XGBoost finales optimizados en el conjunto de prueba.	52
7	Métricas de rendimiento por característica individual para el Modelo XGBoost de Características Binarias Optimizado.	53
8	Métricas de rendimiento de los modelos de redes neuronales finales en el conjunto de prueba, usando hiperparámetros personalizados.	61
9	Métricas de rendimiento por clase para el modelo PERT, con hiperparámetros personalizados.	62
10	Métricas de rendimiento por característica para el modelo <i>Binary Features</i> , con hiperparámetros personalizados.	63
11	Predicciones de modelos XGB y Feed Forward	64
12	ILS XGB (10 ejecuciones).	65
13	ILS Feed Forward (10 ejecuciones).	66
14	ILS Merge (10 ejecuciones).	66
15	Resultados para la instancia completa, ILS y B&C con la ganancia y tiempo total de los 25 clústeres añadidos. Extraída de [Villagrán <i>et al.</i> , 2020]	67

INTRODUCCIÓN

El problema de ruteo de vehículos multiproducto con mezcla, VRPMPWB por sus siglas en inglés, es una variante compleja del clásico problema de ruteo de vehículos VRP, que incorpora la posibilidad de mezclar productos durante el transporte. Este problema tiene aplicaciones importantes en áreas como la distribución de combustibles [Avella *et al.*, 2004b], la recolección de leche [Paredes-Belmar *et al.*, 2016] y la entrega de productos químicos [Popović *et al.*, 2012].

Debido a su naturaleza NP-hard [Lenstra y Kan, 1981], la resolución de diferentes variantes del *Vehicle Routing Problem* para instancias de tamaño real requiere el uso de técnicas metaheurísticas para obtener soluciones de tamaño aceptable en tiempos reducidos. Sin embargo, el rendimiento de estas metaheurísticas depende de la correcta configuración de sus parámetros, un proceso conocido como sintonización de parámetros [Hutter *et al.*, 2009].

La sintonización de parámetros es un proceso computacionalmente costoso, especialmente cuando se aplica a cada nueva instancia del problema. En este contexto, el uso de técnicas de aprendizaje automático para predecir configuraciones de parámetros óptimas basadas en las características de las instancias del problema surge como una alternativa prometedora [Hutter *et al.*, 2014].

Este trabajo de título se centra en el desarrollo y evaluación de un enfoque basado en aprendizaje automático para la predicción de parámetros metaheurísticos en el contexto del VRP. La variante específica que será investigada durante este trabajo es el *Multi-product Vehicle Routing Problem With Gradual Milk Blending* (GMB) introducido por [Montero *et al.*, 2022] en el contexto particular de la recolección de leche.

La hipótesis es que un modelo de aprendizaje automático entrenado adecuadamente puede lograr un rendimiento similar a los algoritmos de sintonización tradicionales, reduciendo significativamente el tiempo de configuración sin comprometer la calidad de las soluciones. Además, se espera que este enfoque proporcione perspectivas valiosas sobre la relación entre las características de las instancias y las configuraciones de parámetros óptimos. Esta investigación contribuirá al campo de la optimización combinatoria y el aprendizaje automático, elucidando este enfoque para abordar el desafío de la sintonización de parámetros en metaheurísticas para problemas de ruteo de vehículos complejos. Los resultados obtenidos podrían tener implicaciones significativas para la resolución eficiente de problemas de optimización en la industria logística y de transporte.

Los objetivos principales de esta investigación son:

1. Validar si un modelo de aprendizaje automático es capaz de predecir los parámetros para un algoritmo del estado del arte, y comparar dichas predicciones en términos de eficacia y eficiencia con los resultados de sintonizadores tradicionales.

2. Desarrollar modelos de aprendizaje automático capaz de predecir configuraciones de parámetros efectivas para una metaheurística que resuelva el *Gradual Milk Blending*.
3. Identificar las características de las instancias que más influyen en la selección de parámetros óptimos.
4. Analizar la capacidad de generalización del modelo a través de instancias de pruebas del *Gradual Milk Blending* procedentes de diferentes fuentes.

Este documento se estructura de la siguiente manera, en el capítulo primero se plasma el problema particular a resolver y las particularidades de la versión de problema de ruteo de vehículos con mezcla, se presentan las metaheurísticas vistas en el diseño de los algoritmos que lo resuelven y se explican las distintas capas que se deben validar/optimizar. Además, se introducen algunos modelos de aprendizaje automático utilizados para la predicción de parámetros. El capítulo segundo está orientado a proveer los fundamentos teóricos y la validación de las componentes del experimento, se presentan técnicas y herramientas del estado del arte para sintonizar problemas de optimización combinatoria. Además, se contextualiza en los fundamentos y variantes de los modelos de aprendizaje automático.

El capítulo tercero presenta el diseño del *framework* propuesto, explica la interacción entre componentes, se selecciona la metaheurística para resolver el *VRPWGMB*, se presenta el ambiente y herramientas a utilizar, se propone una fase de ingeniería de atributos para reducir dimensionalidad y darle predictores nutritivos a los modelos, así como el diseño de los modelos de aprendizaje automático. El capítulo cuarto describe el ambiente de experimentación, se presentan y analizan los resultados de los modelos tanto en conjuntos de validación como instancias no vistas. El último capítulo se revelan las conclusiones generales del trabajo desarrollado y se presentan recomendaciones para investigaciones futuras derivados del estudio.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

En este capítulo se presenta el problema de predicción de parámetros para metaheurísticas que resuelven el problema de ruteo de vehículos multiproducto con mezcla gradual (VRPMPWB). Se describirá dicho problema de ruteo y se presentarán los enfoques del estado del arte para su optimización, incluyendo la sintonización y predicción de parámetros en algoritmos metaheurísticos mediante aprendizaje automático, todo lo cual se resume en el árbol del problema.

1.1. El problema de ruteo de vehículos multiproducto con mezcla gradual

El problema de ruteo de vehículos multiproducto con mezcla gradual es una extensión del clásico problema de ruteo de vehículos, diseñado específicamente para la logística de recolección y mezcla de leche. Introducido por [Montero *et al.*, 2022], este problema incorpora la mezcla gradual de diferentes calidades de leche durante el transporte, lo cual afecta la calidad final del producto.

Además, considera categorías de leche basadas en la concentración de células somáticas, la demanda de la planta procesadora, la oferta de las granjas, disposición espacial de las granjas y diversas restricciones como la capacidad de los vehículos. El objetivo del problema es optimizar rutas que minimicen costos de transporte, a su vez, maximizando las ganancias y cumpliendo con las restricciones establecidas. El *Vehicle Routing Problem Multi-Product with blending*, VRPMPWB por sus siglas en inglés es una extensión del clásico problema de ruteo de vehículos (VRP), que incorpora características específicas relacionadas con la logística de recolección y mezcla de leche en la industria láctea. Este problema fue introducido por [Montero *et al.*, 2022] con el objetivo de abordar los desafíos únicos en esta área. Las principales características adicionales del VRPMPWB incluyen:

1. **Mezcla gradual de productos:** A lo largo del transporte, se permite la mezcla de diferentes calidades de leche. Esta mezcla afecta gradualmente la calidad y el valor del producto final, lo cual refleja con mayor precisión las condiciones reales del proceso logístico en la industria láctea [Paredes-Belmar *et al.*, 2017].
2. **Categorías de leche:** La calidad de la leche está definida por la concentración de células somáticas (SC) por mililitro, acorde a la legislación chilena [Ministerio de Agricultura de Chile, 1979], y se clasifica en:
 - Grado A: Menos de 500.000 SC por mililitro de leche.
 - Grado B: Entre 500.000 y 1.000.000 SC por mililitro de leche.

- Grado C: Más de 1.000.000 SC por mililitro de leche.

La mejor leche es de grado A ya que una menor concentración de células somáticas en la leche es indicio de una mejor calidad y, por lo tanto, un mayor precio por esta.

3. **Dinámica de la mezcla:** La calidad resultante de mezclar leches de distintas calidades se calcula de forma proporcional al volumen y calidad de cada componente, lo que añade complejidad a la optimización de rutas y cargas [Montero *et al.*, 2022].
4. **Demanda y oferta:** El problema considera tanto la demanda de la planta procesadora como la oferta proveniente de las granjas, debiendo equilibrar ambas para llegar a una solución adecuada [Paredes-Belmar *et al.*, 2016].
5. **Restricciones de capacidad:** Los vehículos tienen una capacidad limitada y pueden contar con múltiples compartimentos, cada uno con su propia capacidad [Montero *et al.*, 2022].
6. **Restricciones de compatibilidad (Opcional):** Algunas combinaciones de mezclas pueden estar prohibidas debido a regulaciones o requerimientos de calidad [Paredes-Belmar *et al.*, 2017].
7. **Ventanas de tiempo (Opcional):** La recolección de la leche debe realizarse dentro de ciertos intervalos de tiempo para preservar la frescura y calidad del producto [Montero *et al.*, 2022].

El objetivo principal del VRPMPWB es optimizar la operación de recolección de leche, buscando:

- Minimizar el costo total del transporte.
- Maximizar la ganancia o *profit*.
- Maximizar la calidad de la leche recolectada.
- Satisfacer la demanda de la planta procesadora y la oferta de las granjas.

La formulación matemática detallada del VRPMPWB se encuentra en [Montero *et al.*, 2022]. Esta formulación incluye la función objetivo, variables de decisión relacionadas con las rutas de los vehículos, las cantidades de leche recolectadas, mezcladas, y las calidades resultantes, además de restricciones que modelan las características únicas del problema.

Debido a la complejidad inherente del VRPMPWB, su resolución para instancias de tamaño realista es computacionalmente intratable. Esto justifica el uso de metaheurísticas, cuya eficacia depende en gran medida de la configuración adecuada de sus parámetros, lo que motiva el enfoque de investigación en la predicción de parámetros a través de modelos de aprendizaje automático.

1.2. Metaheurísticas para el VRPMPWB

Las metaheurísticas han demostrado ser particularmente efectivas para resolver el VRPMPWB, especialmente debido a la complejidad añadida por la mezcla gradual de productos y las restricciones de calidad. Las principales metaheurísticas que han sido aplicadas exitosamente a este problema específico incluyen:

- **Búsqueda Local Iterada (ILS):** Esta metaheurística ha demostrado ser especialmente efectiva para el VRPMPWB debido a su capacidad para manejar las restricciones de mezcla gradual y calidad [Montero *et al.*, 2022]. Los parámetros clave incluyen:
 - Número de iteraciones sin mejora
 - Intensidad de perturbación
 - Criterios de aceptación de nuevas soluciones
- **Algoritmos Evolutivos:** Adaptados específicamente para manejar la complejidad de la mezcla de productos y las restricciones de calidad en el VRPMPWB [Paredes-Belmar *et al.*, 2016]. Los parámetros críticos incluyen:
 - Tamaño de la población
 - Operadores de cruce específicos para preservar rutas factibles
 - Tasas de mutación adaptadas a las restricciones de calidad
- **Búsqueda de Vecindario Variable (VNS):** Particularmente efectiva para explorar diferentes estructuras de vecindario que consideren tanto las rutas como las decisiones de mezcla [Polat *et al.*, 2022]. Los parámetros relevantes son:
 - Número de estructuras de vecindario
 - Criterios de cambio entre vecindarios
 - Profundidad de la búsqueda local

Estas metaheurísticas han sido modificadas específicamente para abordar las características únicas del VRPMPWB, como la mezcla gradual de productos y las restricciones de calidad. La eficacia de cada método depende en gran medida de la correcta configuración de sus parámetros, los cuales deben ajustarse considerando una mirada holística a los aspectos de las instancias como aspectos logísticos, de mezcla y *clustering*. Esto motiva el desarrollo de la investigación para la predicción de parámetros metaheurísticos mediante modelos de aprendizaje automático.

1.3. Sintonización de parámetros

La sintonización de parámetros presenta un desafío multinivel en el contexto de esta investigación, donde se requiere optimizar tanto los parámetros de los algoritmos que resuelven el problema de ruteo como los hiperparámetros de los modelos de aprendizaje automático que predicen dichos parámetros. Así como el proceso mismo a nivel metodológico y técnico de el *framework* de la investigación. Este problema de sintonización multinivel se puede descomponer en las siguientes capas:

1.3.1. Capa de Optimización del VRP

En el nivel más básico, se encuentra la sintonización de los parámetros de las metaheurísticas que resuelven el VRPMPWB. Este proceso es crítico porque:

- Los parámetros tienen un impacto directo en la calidad de las rutas generadas y la gestión de la mezcla de productos.
- La configuración óptima varía según las características de cada instancia, como el número de clientes, la distribución geográfica y los tipos de productos.
- La interacción entre parámetros es compleja y no lineal, haciendo que la búsqueda manual sea ineficiente.

1.3.2. Capa de Predicción de Parámetros

El siguiente nivel implica la optimización de los modelos de aprendizaje automático que predicen los parámetros óptimos. Este proceso presenta sus propios desafíos:

- La selección y optimización de hiperparámetros de los modelos predictivos afecta directamente su capacidad para generalizar patrones entre diferentes instancias.
- Se requiere un balance entre la complejidad del modelo y su capacidad de generalización para evitar el *overfitting*.
- La calidad de las predicciones depende de la representatividad de los datos de entrenamiento, la selección adecuada de características, entre otras variables.

Este problema se vuelve aún más complejo debido a las siguientes consideraciones:

1. **Compromiso computacional:** Existe un balance necesario entre la precisión deseada en cada nivel, cantidad de datos y recursos computacionales disponibles. Una sintonización exhaustiva en todos los niveles puede ser computacionalmente prohibitiva.

2. **Variabilidad contextual:** Las configuraciones óptimas pueden variar según el contexto operativo, como las restricciones de tiempo, los requisitos de calidad de la solución y los recursos disponibles.
3. **Escalabilidad:** El sistema debe poder manejar eficientemente tanto instancias pequeñas como grandes, adaptando las estrategias de sintonización según sea necesario.

1.4. Predicción de parámetros mediante aprendizaje automático

Los mecanismos de predicción en el aprendizaje automático para la optimización de parámetros se basan en el desarrollo de modelos que pueden aprender y generalizar patrones a partir de datos históricos. Estos mecanismos buscan establecer relaciones entre las características de las instancias del problema y los parámetros óptimos que maximizan el rendimiento del algoritmo. Los principales mecanismos de predicción utilizados incluyen:

- **Modelos de regresión:** Establecen relaciones funcionales entre las características de entrada y los valores de los parámetros. Pueden ser lineales para capturar relaciones simples o no lineales para patrones más complejos. Los métodos como *Support Vector Regression* (SVR) han mostrado buenos resultados en la predicción de parámetros para problemas de optimización [Hutter *et al.*, 2014].
- **Modelos basados en árboles:** Como *Random Forests* y *Gradient Boosting*, son efectivos para capturar interacciones no lineales entre características y manejar tanto variables numéricas como categóricas. Tienen la ventaja de poder identificar automáticamente las características más relevantes para la predicción [Muñoz y Smith-Miles, 2017].
- **Redes neuronales:** Pueden aprender representaciones complejas y no lineales entre las características del problema y los parámetros óptimos. Son particularmente útiles cuando existen relaciones intrincadas entre las características de las instancias y los parámetros que maximizan el rendimiento.
- **Modelos de ensemble:** Combinan las predicciones de múltiples modelos base para obtener predicciones más robustas y precisas. Métodos como *stacking* y *bagging* han demostrado ser efectivos para la predicción de parámetros en problemas de optimización combinatoria.

La efectividad de estos mecanismos depende fundamentalmente de:

1. La calidad y representatividad de los datos de entrenamiento.
2. La selección adecuada de las características que describen las instancias del problema.

3. El volumen de datos disponible.
4. La capacidad del modelo para generalizar a nuevas instancias no vistas durante el entrenamiento.
5. La robustez frente a ruido e incertidumbre en los datos.

En el contexto del VRPMPWB, estos mecanismos de predicción permiten adaptar dinámicamente los parámetros de las metaheurísticas según las características específicas de cada instancia, mejorando tanto la calidad de las soluciones como la eficiencia computacional del proceso de optimización.

1.5. Árbol del Problema

A continuación se presenta el árbol del problema asociado.

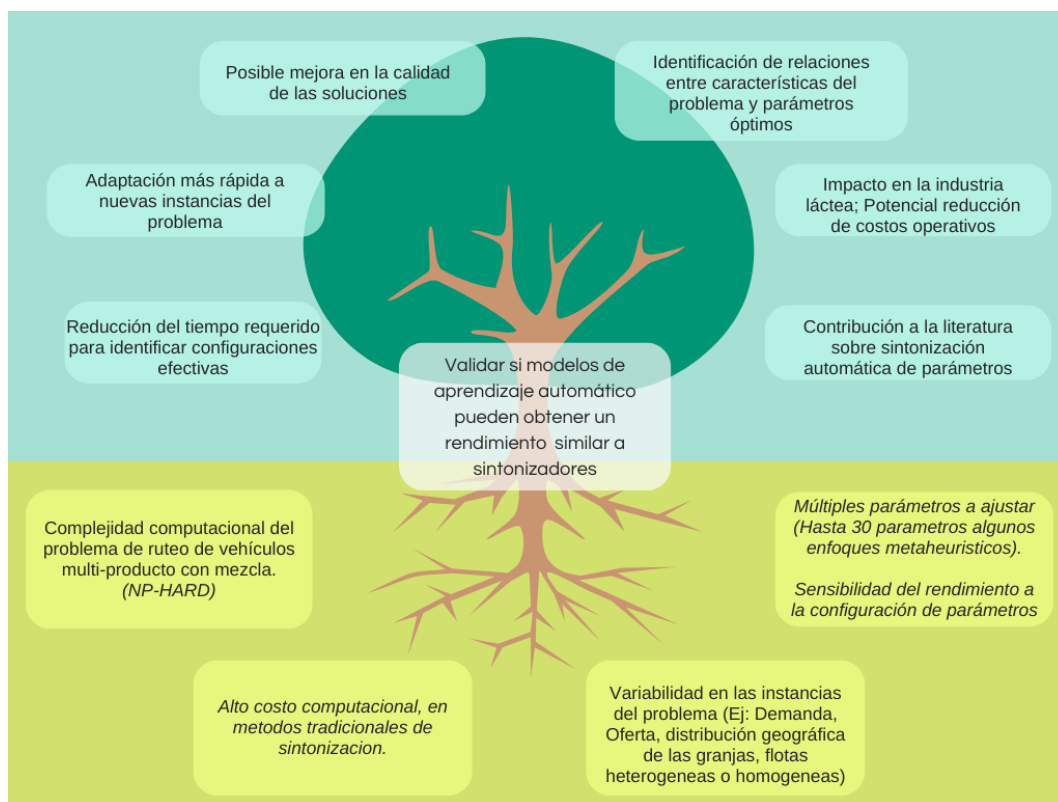


Figura 1: Árbol del Problema: Sintonización de metaheurísticas con aprendizaje automático para el ruteo de vehículos multiproducto con mezcla.

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se presenta el problema particular de *VRP* a resolver y sus características, seguido por los algoritmos metaheurísticos recientes que logran resolverlo. Luego se presentan herramientas y técnicas idóneas para la predicción de problemas logísticos tipo *VRP*; se presentan brevemente sus fundamentos teóricos y los distintos *trade-offs* a considerar. El capítulo continúa con una revisión en la que se presentan algunos de los fundamentos teóricos de modelos de aprendizaje automático clásico y de redes neuronales. Dichos fundamentos serán útiles para comprender el diseño propuesto posteriormente. Por último, se presentan técnicas de optimización de hiperparámetros que están ligadas a mejorar los modelos de aprendizaje automático para la predicción de parámetros.

2.1. El problema de ruteo de vehículos con mezclas (*VRPMPWB*)

El problema de Ruteo de Vehículos Multiproducto con mezclas (*VRPMPWB*) por sus siglas en inglés, es una variante compleja del clásico problema de ruteo de vehículos (*VRP*), que incorpora la posibilidad de mezclar productos durante el transporte. Esta variante tiene aplicaciones relevantes en diversas industrias, tales como la recolección de leche [Paredes-Belmar *et al.*, 2016], la distribución de combustibles [Avella *et al.*, 2004a] y la entrega de productos químicos [Fagerholt, 2000].

Las principales características que diferencian al *VRPMPWB* del *VRP* clásico incluyen:

- **Mezcla de productos:** Permite la combinación de productos durante el transporte, lo que puede afectar tanto la calidad como el valor del producto final. Esta característica añade una capa de complejidad adicional a la optimización de rutas y cargas [Paredes-Belmar *et al.*, 2016].
- **Demanda y recogida:** Los clientes pueden requerir tanto la entrega como la recogida de productos, lo cual complica la planificación de rutas al tener que satisfacer ambos tipos de requerimientos [Avella *et al.*, 2004a]. Esta característica no es directamente aplicable al contexto lácteo.
- **Impacto en la calidad:** La mezcla de productos puede afectar la calidad final, lo que es relevante en la recolección de leche. Este aspecto introduce una dimensión adicional a la función objetivo del problema [Paredes-Belmar *et al.*, 2016].
- **Capacidad variable:** La capacidad efectiva de los vehículos puede cambiar según cómo se mezclen los productos, lo que añade más complejidad a la planificación de rutas y cargas [Villagrán *et al.*, 2020].

- **Restricciones de compatibilidad:** (Opcional) Algunos productos no pueden mezclarse o deben ser transportados en compartimentos separados, debido a restricciones de seguridad, calidad o regulaciones [Fagerholt, 2000].

En el contexto específico de la recolección de leche, se han propuesto diversos modelos que abordan las particularidades de esta industria:

En [Paredes-Belmar *et al.*, 2016], los autores presentaron un modelo de programación entera mixta que considera diferentes calidades de leche y su impacto en el valor final del producto. En este enfoque, la mezcla de calidades se maneja discretamente, donde la calidad resultante de la mezcla está determinada por la leche de menor calidad en el tanque. Este modelo optimiza la recolección de leche maximizando el beneficio total.

En [Villagrán *et al.*, 2020], los autores introdujeron una variante más realista del problema, aplicando un enfoque de "*prize collecting*", donde no es necesario visitar todas las granjas. Este modelo refleja mejor las condiciones de la industria láctea, donde las decisiones de recolección dependen de la calidad y cantidad de leche disponible en cada granja. Además, incorpora restricciones de ventanas de tiempo y la posibilidad de realizar recolecciones parciales.

En [Montero *et al.*, 2022], los autores extendieron el problema introduciendo el concepto de mezcla gradual en el VRPMPWB, donde la calidad de la mezcla se calcula proporcionalmente al volumen y la calidad de cada componente. Este enfoque proporciona una representación más precisa del proceso de mezcla en la recolección de leche, permitiendo una optimización más detallada de las rutas y decisiones de mezcla.

La complejidad del VRP, y en particular del VRPMPWB, lo hace computacionalmente intratable para instancias de tamaño realista utilizando métodos exactos. En consecuencia, el uso de metaheurísticas se ha consolidado como el enfoque predominante para resolver este tipo de problemas [Montero *et al.*, 2022]. No obstante, el rendimiento de estas metaheurísticas depende en gran medida de una correcta configuración de sus parámetros, lo que resalta la necesidad de desarrollar métodos eficientes de sintonización de parámetros.

2.2. Metaheurísticas para el VRPMPWB

Debido a la complejidad computacional del VRPMPWB, que es un problema *NP-hard* [Lenstra y Kan, 1981], las metaheurísticas se utilizan ampliamente para resolver instancias de tamaño realista.

Un enfoque interesante para nuestro trabajo es el presentado por [Polat *et al.*, 2022], quienes propusieron un método de Búsqueda de Vecindad Variable (VNS) para resolver el problema de recolección de leche con restricciones realistas. Este algoritmo incorpora múltiples características del problema, incluyendo ventanas de tiempo y flota heterogénea, lo que re-

presenta un avance significativo en la modelización del VRP en el contexto de la recolección de leche.

Otro enfoque importante es el propuesto por [Montero *et al.*, 2022], quienes desarrollaron un algoritmo de Búsqueda Local Iterada (ILS) para resolver el problema de recolección de leche con mezcla gradual. Este método introduce el concepto de mezcla gradual en el VRPWGMB, donde la calidad de la mezcla se calcula proporcionalmente al volumen y la calidad de cada componente.

Por último, un enfoque relevante es el trabajo de [López-Castro *et al.*, 2023], quienes presentaron un método de Búsqueda Local Iterada *Multi-Start* (MS-ILS) para abordar el problema de recolección de leche con flota heterogénea, considerando criterios ambientales. Este algoritmo incorpora la minimización de emisiones de CO₂ como objetivo, lo que representa un avance significativo en la inclusión de aspectos de sostenibilidad en el VRP. El MS-ILS propuesto por López-Castro *et al.* se basa en el marco de trabajo de *ParamILS*, lo cual es relevante para nuestro estudio sobre la predicción de configuraciones óptimas de parámetros.

Estos enfoques son de particular interés para nuestro estudio, ya que nuestro trabajo se centra en desarrollar un método para predecir configuraciones óptimas de parámetros para algoritmos metaheurísticos aplicados al VRPWGMB. En este contexto, consideraremos diversos algoritmos y técnicas de sintonización para nuestro análisis y desarrollo.

Además, cabe destacar el trabajo de [Marmion *et al.*, 2013], quienes desarrollaron un enfoque de diseño automático de algoritmos de búsqueda local estocástica híbridos. Este enfoque utiliza una gramática para describir posibles instanciaciones de una metaheurística generalizada de búsqueda local, lo que podría ser relevante para futuras extensiones de nuestro trabajo en la predicción de parámetros para el VRP.

La evolución de estos modelos, desde el enfoque discreto de [Paredes-Belmar *et al.*, 2016] hasta el modelo más realista de [Montero *et al.*, 2022], muestra la creciente complejidad y precisión en el modelado del VRPMPWB en el contexto de la recolección de leche. Estos avances destacan la importancia de desarrollar metaheurísticas eficientes y adaptables que puedan manejar las complejidades adicionales introducidas en los modelos más recientes.

La tabla 1 presenta un resumen de acercamientos metaheurísticos de la literatura para resolver el VRPMGWB.

Referencia	Características VRP								Objetivo	Método de solución
	MF	HF	MC	TT	MP	TW	PD	GB		
[Claassen y Hendriks, 2007]	•								(Min.) Desviaciones en niveles de demanda	Exacto
[Caramia y Guerriero, 2010]		•	•						(Min.) Tiempo de preparación y transporte	Búsqueda Local
[Dayarian et al., 2015]	•								(Min.) Costo total	Generación de Columnas
[Lahrichi et al., 2013]	•	•	•		•				(Min.) Costo de transporte	Búsqueda Tabú
[Paredes-Belmar et al., 2016]		•							(Max.) Ganancias	Exacto y Heurísticas
[Masson et al., 2016]		•				•			(Min.) Distancia	Búsqueda Larga de Vecindario Adaptativa
[Sethanan y Pitakaso, 2016]		•	•				•		(Min.) Costo total	Evolución Diferencial
[Paredes-Belmar et al., 2017]	•								(Max.) Ganancias	ACO
[Villagrán et al., 2020]		•							(Min.) Costo total	GRASP
[Huang et al., 2019]	•								(Min.) Tiempo total	Cobertura de Conjuntos
[Polat et al., 2019]			•						(Min.) Distancia	Exacto
[Chokanat et al., 2019]			•			•			(Min.) Costo total	Evolución Diferencial
[Villagrán et al., 2020]									(Max.) Ganancias	Búsqueda Local Iterada
[Ruiz-Meza et al., 2020]	•	•					•		(Min.) Costo total	Exacto
[Montero et al., 2022]		•						•	(Min.) Costo total	Búsqueda Local Iterada
[Polat y Topaloglu, 2022]			•						(Min.) Costo total del viaje	Búsqueda Local Iterada
[Polat et al., 2022]		•	•			•			(Min.) Costo total	Búsqueda de Vecindario Variable
[Erdem, 2022]	•					•			(Min.) Costo total energético	Búsqueda de Vecindario Variable
[López-Castro et al., 2023]		•							(Min.) Emisiones de CO ₂	Búsqueda Local Iterada

Tabla 1: Características de modelado de problemas VRP para recolección de leche. Adaptada de [López-Castro et al., 2023]. MF: Multifábricas, HF: Flota heterogénea, MC: Multicompartimento, TT: Camión y tráiler, MP: Multiperiodo, TW: Ventanas de tiempo, PD: Recogida y entrega, GB: Mezcla global.

2.3. Técnicas de Sintonización de Parámetros para Problemas de Ruteo de Vehículos

La sintonización automática de parámetros para metaheurísticas que abordan el Problema de Enrutamiento de Vehículos (*Vehicle Routing Problem*, VRP) representa un desafío crítico dentro de la optimización combinatoria. Este campo ha experimentado una evolución significativa, transitando desde enfoques manuales hacia métodos cada vez más sofisticados y automatizados. Los principales avances en este ámbito pueden categorizarse en cinco grupos: métodos basados en búsqueda local, métodos basados en *racing*, métodos basados en modelos, métodos basados en población y tendencias emergentes.

En el contexto de los métodos basados en búsqueda local, ParamILS [Hutter et al., 2009] des-

taca como una contribución fundamental. Este algoritmo, diseñado específicamente para la configuración automática de algoritmos, opera alternando entre fases de perturbación, donde se introducen cambios aleatorios controlados a la configuración actual, y fases de búsqueda local, durante las cuales se explora exhaustivamente el vecindario de la configuración mediante cambios unitarios en los parámetros.

Una característica distintiva de ParamILS es su mecanismo de *capping* adaptativo, que permite terminar anticipadamente evaluaciones poco prometedoras utilizando un límite dinámico basado en el rendimiento de la mejor configuración encontrada hasta el momento. Variantes posteriores, como FocusedILS [Hutter *et al.*, 2011], incorporan estrategias de intensificación para asignar más recursos a configuraciones que exhiben un desempeño sobresaliente.

Por otro lado, los métodos basados en *racing* han demostrado ser particularmente efectivos en escenarios con presupuestos computacionales limitados. Irace [López-Ibáñez *et al.*, 2016] se posiciona como el representante más destacado de esta categoría, implementando un proceso iterativo que combina muestreo probabilístico y eliminación progresiva de configuraciones. El componente de muestreo genera nuevas configuraciones de acuerdo con una distribución que favorece a las configuraciones élite identificadas en iteraciones previas, mientras que el mecanismo de eliminación emplea el test estadístico de Friedman para comparar y descartar configuraciones inferiores, concentrando así los recursos en las alternativas más prometedoras.

En el ámbito de los métodos basados en modelos, la optimización bayesiana ha emergido como un enfoque predominante. SMAC3 [Lindauer *et al.*, 2022] se destaca por su uso innovador de bosques aleatorios como modelos sustitutos para predecir el rendimiento de configuraciones no evaluadas. Estos modelos generan predicciones agregando las estimaciones individuales de cada árbol en el bosque, y son utilizados para guiar el proceso de selección de nuevas configuraciones maximizando una función de mejora esperada.

Los métodos basados en población, como los algoritmos evolutivos y la optimización por enjambre de partículas, han demostrado un desempeño sobresaliente cuando se dispone de recursos para evaluaciones paralelas.

Los algoritmos evolutivos [Eiben *et al.*, 1999] mantienen una población de configuraciones que evoluciona a través de operadores de selección, cruce y mutación, con diseños especializados para manejar tanto parámetros numéricos como categóricos. Por su parte, la optimización por enjambre de partículas [Kennedy y Eberhart, 1995] modela las configuraciones como partículas que se desplazan en el espacio de búsqueda, actualizando sus posiciones de acuerdo con su experiencia individual y la información compartida por el enjambre.

Más recientemente, han surgido tendencias prometedoras como el uso de técnicas de aprendizaje automático para predecir configuraciones efectivas a partir de las características de las instancias del problema [Zhang *et al.*, 2020], la configuración *online* que ajusta dinámicamente los parámetros durante la ejecución del algoritmo [Kadioglu *et al.*, 2010], y los métodos híbridos que integran estrategias complementarias como *racing* y optimización bayesia-

na para aprovechar sus fortalezas individuales [Ansótegui *et al.*, 2015].

Estos avances apuntan hacia el desarrollo de sistemas de configuración más robustos y eficientes, capaces de adaptarse a las particularidades de cada instancia del VRP, reducir los tiempos de sintonización y mejorar la transferibilidad de las configuraciones a través de diferentes escenarios.

La investigación actual en este campo se orienta hacia el desarrollo de métodos más eficientes y robustos que puedan manejar espacios de configuración de mayor escala y complejidad, acelerar la identificación de configuraciones de alto rendimiento, mejorar la transferibilidad de las configuraciones entre diferentes tipos de instancias y aprovechar el conocimiento acumulado para guiar la búsqueda de manera más informada. Esto incluye la incorporación de conocimiento del dominio en el proceso de configuración.

La tendencia hacia la automatización del diseño algorítmico sugiere que los métodos de configuración automática seguirán ganando importancia en el futuro del desarrollo de algoritmos para VRP. En última instancia, estos esfuerzos contribuyen a potenciar la aplicabilidad y el impacto de las metaheurísticas en la resolución de problemas desafiantes de optimización combinatoria como el VRP.

La elección del método de sintonización más apropiado depende de factores como la naturaleza de los parámetros a configurar —que pueden ser continuos, discretos o categóricos—, el presupuesto computacional disponible, las características del problema que se resuelve, los objetivos específicos del proceso de sintonización —rendimiento medio, robustez, tiempo de configuración—, entre otros.

2.4. Predicción de Parámetros mediante Aprendizaje Automático Supervisado

La predicción de parámetros mediante aprendizaje automático ha emergido como una alternativa prometedora para abordar el costoso proceso de configuración algorítmica. Esta aproximación se basa en la extracción de características relevantes de las instancias del problema para predecir configuraciones de parámetros efectivas.

Los árboles de decisión y *Random Forests* han mostrado buenos resultados en la optimización de hiperparámetros, especialmente cuando se combinan con técnicas de búsqueda bayesiana [Hutter *et al.*, 2011]. Su capacidad para manejar tanto variables categóricas como numéricas los hace particularmente útiles en este contexto.

Las *Support Vector Machines* (SVM) se han aplicado exitosamente en la predicción de parámetros, especialmente cuando se utilizan *kernels* no lineales para capturar relaciones complejas entre características y parámetros óptimos [Mohammed y El Kafhali, 2022].

La regresión logística y el análisis discriminante gaussiano (GDA) por sus siglas en inglés, han demostrado ser efectivos en la clasificación de configuraciones de parámetros, particularmente cuando se combinan con técnicas de ensamblado [Yasmin *et al.*, 2024].

Los modelos bayesianos, incluyendo *Naive Bayes*, han sido utilizados en contextos donde la interpretabilidad del modelo es importante y cuando se dispone de conocimiento previo sobre las relaciones entre características y parámetros [López-Ibáñez *et al.*, 2016].

Las técnicas de *boosting*, particularmente en combinación con árboles de decisión, han mostrado resultados prometedores en la predicción de parámetros, mejorando la robustez de las predicciones [Chen y Guestrin, 2016].

Las redes neuronales han demostrado capacidad para capturar relaciones complejas entre las características de las instancias y sus configuraciones óptimas de parámetros. Sin embargo, su efectividad está fuertemente condicionada por la disponibilidad de conjuntos de datos de entrenamiento suficientemente grandes y diversos [Vanschoren, 2018].

Un aspecto crítico en la predicción de parámetros es la selección y extracción de características relevantes de las instancias. En [Rasku *et al.*, 2016], los autores proponen un conjunto comprensivo de características para problemas de ruteo de vehículos, incluyendo métricas espaciales, de agrupamiento y relacionadas con ventanas de tiempo.

Los resultados experimentales sugieren que la predicción de parámetros mediante aprendizaje automático puede ser más efectiva cuando:

- Se dispone de un conjunto diverso de instancias de entrenamiento que cubra adecuadamente el espacio de características.
- Las características extraídas capturan efectivamente los aspectos relevantes de las instancias para la configuración de parámetros.
- El algoritmo objetivo muestra sensibilidad a sus parámetros, pero mantiene cierta estabilidad en su comportamiento [Kerschke *et al.*, 2018b].

Es importante notar que la efectividad de esta aproximación puede verse limitada cuando se trabaja con conjuntos de instancias muy heterogéneos o con volúmenes bajos de datos respecto a la complejidad de las funciones o comportamientos a predecir.

2.4.1. XGBoost: Marco de Potenciación de Gradiente

El algoritmo *Extreme Gradient Boosting* (XGBoost) representa una implementación optimizada y escalable del marco de potenciación de gradiente (*Gradient Boosting*), desarrollado

por [Chen y Guestrin, 2016]. Su eficacia reside en la construcción de un modelo de ensamblado de manera secuencial. A diferencia de métodos como *Random Forest*, donde los estimadores base (árboles) se construyen de forma independiente, la potenciación de gradiente es un proceso aditivo en el que cada nuevo árbol se entrena para corregir los errores residuales del conjunto de árboles previamente entrenados.

Formalmente, dado un conjunto de datos $(x_i, y_i)_{i=1}^N$, el objetivo es aprender una función $F(x)$ que minimice una función de pérdida diferenciable $L(y, F(x))$. El modelo se construye de forma aditiva:

$$\hat{y}_i^{(T)} = \sum_{t=1}^T f_t(x_i) = \hat{y}_i^{(T-1)} + f_T(x_i)$$

donde f_t es un árbol de decisión entrenado en la iteración t . En cada paso, el nuevo árbol f_T se elige para minimizar la función objetivo, que incluye la pérdida y un término de regularización Ω para controlar la complejidad del modelo:

$$\text{Obj}^{(T)} = \sum_{i=1}^N L(y_i, \hat{y}_i^{(T-1)} + f_T(x_i)) + \Omega(f_T)$$

XGBoost utiliza una aproximación de Taylor de segundo orden para la función de pérdida, lo que permite una optimización más precisa y rápida. El término de regularización $\Omega(f)$ penaliza tanto el número de hojas en el árbol como la magnitud de sus pesos, previniendo el sobreajuste.

Dependiendo de la naturaleza de la tarea, la función de pérdida L varía. Una métrica común para regresión es el Error Cuadrático Medio (MSE), definido como:

$$L(y, \hat{y}) = (y - \hat{y})^2 \quad (1)$$

Esta función de pérdida es convexa y diferenciable, lo que facilita la optimización mediante descenso de gradiente. Su derivada proporciona una medida directa del error residual que debe corregirse en cada iteración.

Para clasificación binaria se usa la función de pérdida conocida como logística, basada en la log-verosimilitud:

$$L(y, p) = -[y \log(p) + (1 - y) \log(1 - p)] \quad (2)$$

donde

$$p = \sigma(\hat{y}) = \frac{1}{1 + e^{-\hat{y}}} \quad (3)$$

es la probabilidad predicha obtenida mediante la función sigmoide. Esta función penaliza fuertemente las predicciones incorrectas con alta confianza.

Por último, para clasificación multiclase se utiliza la *cross-entropy* basada en la función logística de la ecuación (2). Las probabilidades de clase se calculan como:

$$p_k = \frac{e^{\hat{y}_k}}{\sum_{j=1}^K e^{\hat{y}_j}} \quad (4)$$

Y la función de pérdida se define como:

$$L(y, \hat{y}) = - \sum_{k=1}^K y_k \log(p_k) \quad (5)$$

donde y_k es la codificación *one-hot* de la clase verdadera y K es el número total de clases. Esta formulación extiende naturalmente la pérdida logística al caso multiclase, manteniendo la interpretación probabilística.

2.4.2. Redes Neuronales Artificiales

Las redes neuronales artificiales son una clase de modelos de aprendizaje automático inspirados en la neurociencia computacional. Están constituidas por unidades de procesamiento interconectadas (neuronas) organizadas en capas.

Las Redes Neuronales *Feed-Forward* (FFNN), también conocidas como perceptrones multicapa (MLP), son la arquitectura canónica de red neuronal. En una FFNN, la información fluye de manera unidireccional desde una capa de entrada (L_0), a través de una o más capas ocultas ($L_1, \dots, L_{H-1}$), hasta una capa de salida (L_H). No existen conexiones cíclicas. La transformación de una capa $l - 1$ a una capa l se define matemáticamente como:

$$a^{(l)} = g(W^{(l)}a^{(l-1)} + b^{(l)}) \quad (6)$$

donde $a^{(l)}$ es el vector de activaciones de la capa l , $W^{(l)}$ es la matriz de pesos que conecta la capa $l - 1$ con la l , $b^{(l)}$ es el vector de sesgos (*bias*) para la capa l , y $g(\cdot)$ es una función de activación no lineal. La composición de estas transformaciones a través de múltiples capas

permite a la red aproximar funciones no lineales de alta complejidad, según el Teorema de Aproximación Universal [Nishijima, 2021].

- **Funciones de Activación:** Son funciones no lineales que se aplican a la salida de cada neurona. Su rol es fundamental para dotar al modelo de la capacidad de aprender relaciones no triviales en los datos.

1. ReLU (*Rectified Linear Unit*):

$$g(z) = \max(0, z) \quad (7)$$

Es computacionalmente eficiente y combate el problema del gradiente evanescente. Su derivada es constante (1 para $z > 0$), lo que facilita el flujo del gradiente durante la retropropagación.

2. Sigmoido:

$$g(z) = \frac{1}{1 + e^{-z}} \quad (8)$$

Comprime su entrada al rango $(0, 1)$, interpretándose como una probabilidad. Es susceptible a la saturación del gradiente, lo que puede ralentizar el aprendizaje.

3. Softmax:

$$g(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (9)$$

Es una generalización de la función sigmoide para múltiples clases. Transforma un vector de *logits* en una distribución de probabilidad sobre K clases mutuamente excluyentes.

4. Lineal:

$$g(z) = z \quad (10)$$

Es la función identidad. Se utiliza en la capa de salida de modelos de regresión, ya que no impone restricciones sobre el rango del valor de salida.

- **Funciones de Pérdida:** Cuantifican la discrepancia entre las predicciones del modelo ($\hat{y}$) y los valores objetivo reales (y). El objetivo del entrenamiento es minimizar esta función.

1. Entropía Cruzada Binaria:

$$L = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (11)$$

Derivada de la teoría de la información, es el estándar para problemas de clasificación binaria y multietiqueta.

2. Entropía Cruzada Categórica:

$$L = - \sum_{k=1}^K y_k \log(\hat{y}_k) \quad (12)$$

Es la extensión para clasificación multiclase, donde y_k es una variable binaria.

El entrenamiento de una red neuronal es un problema de optimización que busca los parámetros (pesos y sesgos) que minimicen la función de pérdida.

2.5. Sintonización de Hiperparámetros en Modelos de Aprendizaje Automático

La optimización de hiperparámetros en modelos de aprendizaje automático constituye un aspecto fundamental que determina significativamente su rendimiento. Estos hiperparámetros son variables de configuración externa que definen la arquitectura y el comportamiento del modelo durante el entrenamiento y la inferencia.

2.5.1. Métodos Tradicionales de Optimización

La búsqueda en *grid* y la búsqueda aleatoria representan los enfoques clásicos para la optimización de hiperparámetros. La búsqueda en *grid* explora sistemáticamente el espacio de hiperparámetros mediante una discretización predefinida, mientras que la búsqueda aleatoria muestrea configuraciones de manera estocástica. Si bien estos métodos son robustos y fáciles de implementar, su eficiencia disminuye significativamente al aumentar la dimensionalidad del espacio de búsqueda [Lindauer *et al.*, 2022].

2.5.2. Optimización Bayesiana

La optimización bayesiana ha emergido como una alternativa más sofisticada, implementada notablemente en el *framework* SMAC3. Este enfoque construye un modelo probabilístico del espacio de hiperparámetros, típicamente utilizando procesos gaussianos o bosques aleatorios, para guiar la búsqueda hacia regiones prometedoras. SMAC3 incorpora, además, técnicas de *racing* para evaluar eficientemente múltiples configuraciones, reduciendo significativamente el costo computacional del proceso de optimización [Lindauer *et al.*, 2022].

2.5.3. Técnicas Metaheurísticas

Los algoritmos metaheurísticos han demostrado ser particularmente efectivos en la optimización de hiperparámetros. El algoritmo *Firefly* (FFA), por ejemplo, ha mostrado resultados prometedores en la sintonización de redes neuronales y *Support Vector Machine* aprovechando su capacidad para mantener diversidad en la búsqueda mientras converge hacia óptimos globales. Los estudios empíricos sugieren que FFA puede superar a otros métodos metaheurísticos como los algoritmos genéticos (*Genetic Algorithms*, GA) y *Particle*

Swarm Optimization (PSO) en términos de calidad de solución y velocidad de convergencia [Mohammed y El Kafhali, 2022].

2.5.4. Multi-dimensional Bagging

El enfoque *Multi-dimensional Bagging* (MDB) representa una innovación reciente en la optimización de hiperparámetros. Esta técnica divide el espacio de búsqueda en subespacios más manejables, permitiendo una exploración más efectiva mientras mantiene la diversidad necesaria en los modelos resultantes. MDB ha demostrado ser particularmente efectivo cuando se combina con técnicas de optimización metaheurística, como se evidencia en los resultados experimentales presentados por [Yasmin *et al.*, 2024].

2.5.5. Consideraciones Técnicas

La selección del método de optimización debe considerar las características específicas del modelo y del problema. Por ejemplo, las redes neuronales típicamente requieren la optimización de múltiples hiperparámetros continuos como la tasa de aprendizaje y su decaimiento, mientras que los modelos basados en árboles involucran principalmente hiperparámetros discretos. La optimización bayesiana tiende a ser más eficiente para espacios de búsqueda de baja a media dimensionalidad, mientras que los métodos metaheurísticos pueden manejar mejor espacios de mayor dimensionalidad [López-Ibáñez *et al.*, 2016].

El proceso de optimización debe incorporar técnicas de validación cruzada para evitar el *overfit* a los datos de entrenamiento. Además, es crucial considerar la interacción entre diferentes hiperparámetros, ya que estos rara vez son independientes. Por ejemplo, en redes neuronales existe una fuerte interacción entre el tamaño de lote, la tasa de aprendizaje y el número de épocas de entrenamiento.

La tendencia actual en investigación apunta hacia el desarrollo de métodos que puedan adaptar dinámicamente la estrategia de búsqueda según las características del problema y el comportamiento observado durante la optimización. Esto incluye la consideración de múltiples objetivos simultáneos, como la precisión del modelo y el costo computacional, así como la incorporación de conocimiento específico del dominio para guiar más efectivamente la búsqueda.

2.6. Resumen

En este capítulo se presentó el Problema de Ruteo de Vehículos con Mezcla, sus particularidades y variantes. Se introdujeron los acercamientos metaheurísticos del estado del arte

que abordan este problema. Luego, se ofreció un resumen de los enfoques específicos aplicados a problemas VRP con mezcla. Además, se describieron las técnicas más relevantes del estado del arte para la sintonización de parámetros, como ParamILS, iRace, entre otras.

Posteriormente, se explicaron brevemente los algoritmos de aprendizaje automático supervisado más útiles para la predicción de parámetros. Finalmente, se presentaron consideraciones técnicas y recomendaciones propuestas por diversos investigadores sobre la selección, evaluación y combinación de estrategias de optimización.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. Introducción

En la presente sección se propone el diseño y validación de un *pipeline* o *framework* de aprendizaje automático orientado a la predicción automática de parámetros para algoritmos metaheurísticos, en el contexto de la optimización combinatoria. El objetivo principal es evaluar la viabilidad y efectividad de este tipo de herramientas como complemento o alternativa a los métodos tradicionales de sintonización automática. Mediante la construcción de modelos predictivos entrenados con características relevantes de las instancias, se busca anticipar configuraciones de parámetros que conduzcan a un buen desempeño algorítmico, reduciendo así el tiempo de experimentación y mejorando la eficiencia computacional en entornos reales y de investigación.

3.2. Selección de Metaheurísticas

Como punto de partida, se propone una evaluación preliminar de diversas metaheurísticas empleadas previamente en problemas similares de ruteo con mezcla gradual, con el objetivo de seleccionar una base algorítmica sólida para esta investigación. Se considerarán tres enfoques principales: la Búsqueda Local Iterada (ILS) de [Montero *et al.*, 2022], que ha demostrado efectividad en contextos reales en Chile; su variante Multi-Start (MS-ILS), desarrollada por [López-Castro *et al.*, 2023], que incorpora criterios ambientales en el diseño de rutas; y finalmente, el algoritmo de Búsqueda de Vecindad Variable (VNS), presentado por [Polat *et al.*, 2022], reconocido por su desempeño robusto en entornos realistas.

A partir de esta evaluación, se optará por emplear como base la metaheurística ILS propuesta por [Montero *et al.*, 2022], ya que se cuenta con acceso a la implementación original, así como experiencia previa en su utilización. Además, se dispone de un conjunto de instancias compatibles y validadas, lo que permite una integración fluida con el resto del flujo experimental de la presente investigación.

El proceso inicia con la generación de una solución factible mediante una heurística constructiva, la cual utiliza un componente voraz aleatorizado controlado por el parámetro RCL (*Restricted Candidate List*). A partir de esta solución inicial, el algoritmo comienza un proceso de refinamiento iterativo, donde busca mejorar inteligentemente la mejor solución encontrada a través de modificaciones locales. La búsqueda se ejecuta de forma continua hasta que se alcanza un límite de tiempo predefinido (TMAX), asegurando la convergencia hacia un resultado de alta calidad dentro del tiempo de cómputo asignado.

Desde una perspectiva técnica, el algoritmo implementa una metaheurística de Búsqueda Local Iterada (ILS) que utiliza una Búsqueda por Entornos Variables (VNS) como su componente de búsqueda local. El VNS explora de manera sistemática y aleatoria un conjunto de cuatro estructuras de vecindad distintas, definidas por los operadores: *Reorder-Nodes*, *Exchange-Nodes*, 2Opt y OR-Opt. Cuando la búsqueda local del VNS alcanza un óptimo local, se activa el mecanismo de perturbación del ILS, que modifica la solución actual para escapar de dicho óptimo. Este ciclo de búsqueda local (intensificación) y perturbación (diversificación) permite una exploración robusta y eficaz del espacio de soluciones.

3.3. Operadores de Búsqueda Local

Para la exploración del espacio de soluciones, se utilizan los siguientes operadores de búsqueda local, los cuales modifican una o dos rutas existentes para generar soluciones vecinas.

El operador *Or-opt*, también conocido como *relocate* intra-ruta o *string relocation*, opera dentro de una única ruta. Su función es seleccionar una secuencia de una o más visitas consecutivas y reubicarla en una posición diferente dentro de la misma ruta. Este operador es útil para optimizar la secuencia interna de visitas, buscando reducir costos asociados a la distancia o al tiempo de recorrido. Técnicamente, el operador evalúa las posibles posiciones de inserción para la secuencia seleccionada y elige aquella que ofrece la mayor mejora en la función objetivo. Su uso es esencial para refinar las rutas individuales y mejorar la calidad general de la solución [Or, 1976].

El operador *Exchange*, también conocido como *relocate* inter-ruta, modifica dos rutas mediante la transferencia de una única visita desde una ruta de origen hacia una posición específica en una ruta de destino. Se utiliza para corregir desequilibrios en la distribución de visitas, buscando minimizar costos operativos como la distancia total. El procedimiento identifica una visita candidata y evalúa su inserción en todas las posiciones posibles de otra ruta, seleccionando el movimiento que genere la mayor reducción de costo. Este enfoque es eficaz para balancear la carga de trabajo entre los vehículos y reducir la longitud de las rutas más largas [Laporte *et al.*, 2014].

El operador *Cross-exchange*, formalizado por [Taillard *et al.*, 1997], es una técnica de mejora aplicada a un par de rutas. Consiste en seleccionar una secuencia de visitas de una ruta y otra secuencia de igual longitud de una segunda ruta para luego intercambiarlas entre sí. Comúnmente, se utilizan secuencias de una a tres visitas. Al modificar múltiples elementos de ambas rutas de manera simultánea, este operador permite una exploración más amplia y potente del espacio de soluciones, contribuyendo a escapar de óptimos locales y a mejorar significativamente la calidad de las soluciones.

El operador *2-opt*, propuesto originalmente por [Croes, 1958], es una de las heurísticas de optimización local más clásicas y efectivas. Pertenece a la familia de operadores de intercambio de aristas (*edge exchange*) y opera sobre una única ruta. El método consiste en eliminar

dos aristas no adyacentes de la ruta, lo que la divide en dos subrutas, y luego reconectar estas subrutas invirtiendo el orden de una de ellas para crear una nueva ruta sin cruces. El objetivo es eliminar intersecciones en el trazado de la ruta, lo que casi siempre resulta en una reducción de la distancia total. A pesar de su simplicidad y complejidad computacional de $O(n^2)$, sigue siendo una técnica fundamental en la resolución del Problema del Vendedor Viajero (TSP) y el Problema de Ruteo de Vehículos (VRP).

3.4. Generación de Datos

Una vez seleccionada la metaheurística de referencia, se diseñará un proceso experimental sistemático que permita generar un conjunto diverso y representativo de instancias del problema. Este proceso utilizará herramientas de sintonización automática como *Irace*, configurado para explorar un amplio espacio de configuraciones de parámetros. De esta forma, se asegurará una adecuada cobertura del espacio de soluciones y una base de datos suficientemente rica para el posterior entrenamiento de modelos predictivos.

El paquete *irace* es un método para la configuración automática de algoritmos que implementa un enfoque de carreras iteradas para encontrar los parámetros de mejor rendimiento para un algoritmo sobre un conjunto de instancias de un problema. El concepto fundamental es la “carrera”, donde múltiples configuraciones de parámetros (candidatos) se ejecutan en paralelo. A medida que se obtienen resultados, *irace* aplica pruebas estadísticas, como la prueba de Friedman, para identificar y descartar tempranamente las configuraciones que muestran un rendimiento estadísticamente inferior.

Este proceso de eliminación secuencial permite concentrar el esfuerzo computacional en los candidatos más prometedores, haciendo que el proceso de sintonización sea mucho más eficiente que una búsqueda exhaustiva.

La siguiente figura 2 representa la interacción inicial de los distintos componentes para la generación de datos:



Figura 2: Diseño inicial de la *pipeline* de sintonización de parámetros

La principal fortaleza de *irace* radica en su naturaleza iterativa. Al finalizar cada carrera, el proceso no se detiene. En su lugar, analiza las configuraciones de élite (aquellas que sobrevivieron a la carrera) y utiliza esta información para actualizar un modelo de muestreo proba-

bilístico. Este modelo adaptativo se emplea para generar la siguiente población de configuraciones candidatas, sesgando la búsqueda hacia las regiones más prometedoras del espacio de parámetros identificadas en iteraciones anteriores. Este ciclo de ejecución, eliminación y muestreo adaptativo permite a *irace* refinar progresivamente la búsqueda, equilibrando de forma eficaz la exploración de nuevas configuraciones con la explotación de las de alto rendimiento, convergiendo así a una configuración de parámetros robusta y altamente optimizada.

La generación de datos contemplará la ejecución paralela de múltiples instancias con diferentes semillas, distribuidas en diversas tandas experimentales, con el objetivo de capturar la variabilidad inherente al comportamiento estocástico del algoritmo. Posteriormente, se realizará un proceso de preprocesamiento y limpieza mediante técnicas específicas que aseguren la calidad de los datos utilizados en la etapa de modelado. Además de los atributos clásicos de instancia, implementación y solución, se incorporarán características compuestas, así como información extraída de estudios previos relacionados, como el trabajo de [Barros Everett, 2024].

La extracción de datos comienza con consumir y procesar los archivos generados por el sintonizador *irace*. Se extraen las tres soluciones élite encontradas por el sintonizador, denotadas en las cajas color verde claro y oscuro en la figura 2 durante su ejecución. Luego, se ejecuta el algoritmo metaheurístico de manera independiente con los valores de los parámetros élite encontrados, representados en la relación desde la caja verde hacia la azul de la figura 3, para validar dichos resultados usando 20 semillas aleatorias. Se almacenan las configuraciones/soluciones no duplicadas, denotadas por la caja morada de la figura 3.

Este conjunto de datos será la base del conjunto de entrenamiento que se acoplará con los conjuntos de datos de características de agrupamiento, espaciales y particulares del problema de ruteo de vehículos multiproducto con mezcla.

La figura 3 presenta el esquema de validación de las soluciones encontradas por el sintonizador. Las cajas verdes representan las soluciones/parámetros élite encontrados y provienen de la fase de generación de datos. Se ejecutan las instancias con los parámetros élite y se comparan con las soluciones élite encontradas por *irace*. Este diseño es una decisión metodológica para asegurar la fiabilidad de las soluciones y parámetros encontrados.

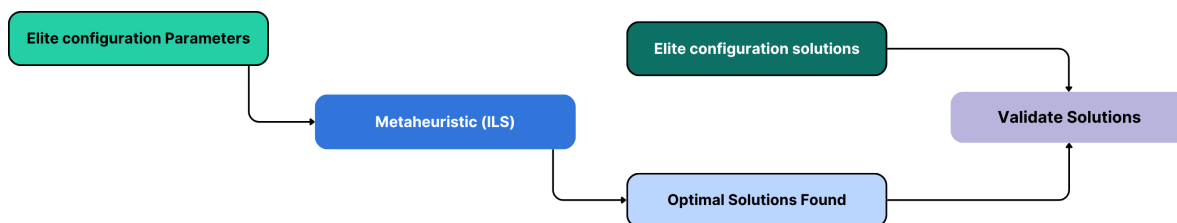


Figura 3: Diseño de alto nivel de la validación de soluciones

3.5. Ingeniería de Atributos

La fase de ingeniería de características, detallada en las secciones posteriores, ha sido fundamental para transformar los datos brutos de las instancias VRP de recolección de leche en un formato que facilita el aprendizaje y desempeño de los modelos.

El resultado de este proceso es un conjunto de datos con menor dimensionalidad y el mismo nivel de información. Este conjunto representa una cuantificación rigurosa de la estructura espacial, las propiedades de agrupamiento y las particularidades del mezclado inherentes a cada instancia.

Recientemente, ha surgido un interés significativo en la aplicación de técnicas de aprendizaje automático (ML) para mejorar los métodos tradicionales de resolución de VRPs [Kerschke *et al.*, 2018a, Zhao *et al.*, 2021]. Un área prometedora de investigación se centra en la construcción de procedimientos de predicción de parámetros de ML para meta-heurísticas, con el objetivo de adaptar automáticamente la configuración de estos algoritmos a las características específicas de cada instancia del problema.

La premisa es que la estructura inherente y la complejidad de una instancia VRP pueden cuantificarse a través de un conjunto de métricas descriptivas, las cuales pueden ser utilizadas por modelos de aprendizaje automático para predecir configuraciones paramétricas efectivas o seleccionar el algoritmo más adecuado.

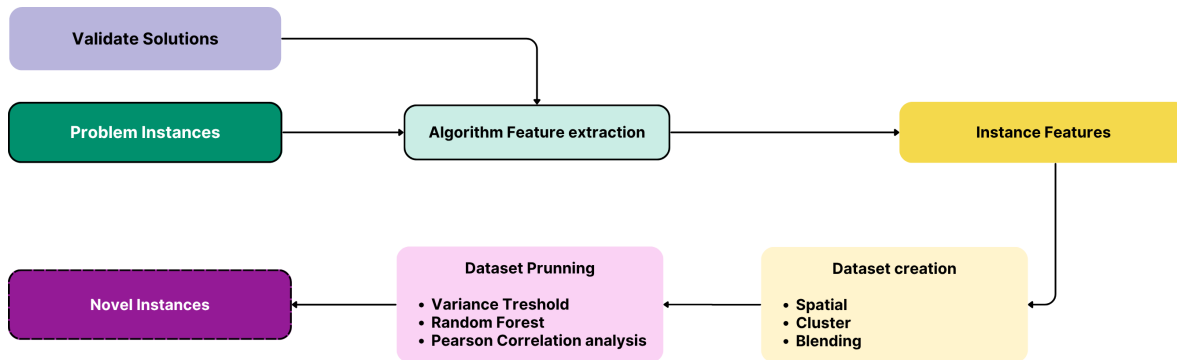


Figura 4: Diseño de alto nivel de la ingeniería de atributos

La figura 4 resume el diseño de alto nivel de la ingeniería de atributos, con sus componentes clave y los flujos del proceso. Cada componente del *pipeline* será explicado con más detalle en las subsecciones posteriores.

3.5.1. Características de la instancia

A continuación se describen los predictores de la instancia que se utilizarán para modelar el problema de recolección de leche. Algunas de estas características pueden resultar redun-

dantes o ser descartados en base a su aporte informativo.

La Tabla 2 lista los predictores que provienen directamente de los datos brutos de las instancias [Montero, 2021]. Es importante destacar que, a pesar de que algunas de estas características puedan resultar redundantes, su inclusión en la etapa inicial de modelado permite evaluar su aporte informativo y calcular predictores compuestos.

Característica	Descripción
No. de Clientes	Representa el total de granjas o clientes a atender.
No. de Productores	Representa el total de granjas o clientes (sin incluir el depósito).
No. de Camiones	Cantidad de vehículos disponibles para realizar la recolección.
Capacidad de Camiones	Capacidad máxima de carga de cada camión, expresada en litros.
Demanda por tipo de leche	Volumen requerido por cada tipo (A, B y C), expresado en miles de litros (leche_A_dem, leche_B_dem, leche_C_dem).
Producción	Vector que representa la producción en cada uno de los nodos o granjas.
Tipo de leche	Vector que representa la categoría de leche que vende cada granja.
Células Somáticas	Vector que representa la cantidad de células somáticas por litro de cada granja.

Tabla 2: Características de la instancia derivadas de los datos brutos

3.5.2. Características espaciales

La *estructura espacial* de una instancia de VRP, definida por la distribución geográfica de los clientes y la ubicación del depósito, es un determinante fundamental de su complejidad computacional y de la naturaleza de las soluciones óptimas [Smith-Miles y Lopes, 2012, Pillac *et al.*, 2013, Uchoa *et al.*, 2017]. Una distribución de clientes dispersa, por ejemplo, puede favorecer estrategias algorítmicas que prioricen la exploración (diversificación), mientras que instancias con alta densidad local o fuertes agrupaciones pueden beneficiarse de enfoques que exploten subproblemas o vecindarios de búsqueda más pequeños (intensificación). La topología espacial afecta directamente métricas clave como la longitud total de las rutas, el número esperado de vehículos necesarios y la eficiencia potencial de los operadores de movimiento utilizados por las metaheurísticas [Laporte, 2009].

Las características espaciales buscan cuantificar esta estructura geométrica, traduciendo la disposición de los nodos en medidas numéricas interpretables. Esto incluye la densidad de

clientes, la forma del área de servicio, la dispersión respecto al depósito y entre los clientes, así como la heterogeneidad de estas distribuciones.

El trabajo de [Rasku *et al.*, 2016, Rasku *et al.*, 2019] ha sido crucial en la identificación y validación de un conjunto completo de características espaciales para problemas de ruteo de vehículos, abarcando descriptores de la distribución de nodos, propiedades geométricas del conjunto de clientes y métricas basadas en la vecindad más cercana. Este marco proporciona la base teórica y metodológica para la selección de características espaciales en este estudio [Barros Everett, 2024].

Las características espaciales seleccionadas están diseñadas para capturar los aspectos topológicos y de distribución de la instancia que son influyentes en la dificultad del problema de ruteo de vehículos y en el rendimiento de los algoritmos. Para la mayoría de las características espaciales basadas en distribuciones de valores (e.g., distancias, demandas, etc.), se calculan la media, la desviación estándar y el coeficiente de variación para proporcionar una visión completa tanto de la tendencia central como de la dispersión y heterogeneidad de la distribución.

A continuación se presenta la Tabla 3, la cual resume las características espaciales consideradas en este estudio, junto con la justificación de por qué cada una es relevante desde una perspectiva de optimización de ruteo y predicción algorítmica.

Característica	Justificación
Número de clientes	Determina el tamaño del problema, afectando la complejidad y el costo computacional [Bachnas <i>et al.</i> , 2014, Vidal <i>et al.</i> , 2013, Uchoa <i>et al.</i> , 2017].
Distancia promedio entre depósito y clientes	Mide la dispersión respecto al depósito, afectando la longitud media de rutas [Vidal <i>et al.</i> , 2012].
Distancia promedio entre clientes	Estima la dispersión global entre clientes y qué tan compactos espacialmente son.
Coeficiente de variación (CV) de distancia centroide-clientes	Evalúa la uniformidad espacial de los clientes respecto a su centroide; un CV alto indica posibles datos atípicos o zonas dispersas.
Relación demanda promedio versus capacidad	Estima el uso típico de la capacidad vehicular por cliente [Uchoa <i>et al.</i> , 2017].
Relación desviación estándar (DE) de demandas versus capacidad	Refleja la variabilidad de la demanda y su impacto en el balanceo de carga.
Relación clientes versus vehículos	Estima la carga promedio por vehículo y la intensidad del servicio [Uchoa <i>et al.</i> , 2017].
Distancia promedio al vecino más cercano	Indica la densidad local promedio; valores bajos sugieren proximidad entre clientes [Kerschke <i>et al.</i> , 2018a].
CV de distancia al vecino más cercano	Mide la heterogeneidad de la densidad local; un CV alto sugiere la coexistencia de zonas densas y dispersas [Kotthoff <i>et al.</i> , 2015].
Densidad de la red	Número de clientes por unidad de área; refleja cuán poblada está la región de servicio.
Distribución espacial de la demanda	Evalúa si la demanda está concentrada en ciertas zonas, lo que afecta la planificación de rutas.
Centralidad del depósito	Mide la cercanía relativa del depósito a los clientes; alta centralidad facilita rutas eficientes.
Aspecto del cuadro delimitador	Indica la forma del área de servicio: si es compacta (casi cuadrada) o alargada y estrecha. Esto determina si el diseño de rutas debe ser zonal/compacto o en forma de corredores lineales.
Circularidad de la distribución de clientes	Evalúa si la dispersión de clientes es radial y balanceada alrededor del centroide [Rasku <i>et al.</i> , 2016].

Tabla 3: Características espaciales

3.5.3. Características basadas en centroides

En esta subsección se detalla el cálculo de los indicadores fundamentales usados para obtener los predictores espaciales listados.

Para el cálculo de predictores espaciales y de agrupamientos es necesario conocer la ubicación de los puntos que conforman cada uno de los casos de prueba del problema. A partir de los datos en las instancias de prueba del VRP es posible la generación del plano en dos dimensiones, que se realiza transformando una matriz de distancias en un conjunto de coordenadas (x, y) , de manera que las distancias entre los puntos aproximen las distancias originalmente especificadas.

Inicialmente, se posiciona el depósito en el origen $(0, 0)$ y se coloca el segundo nodo a una distancia fija desde el depósito, en una dirección determinada de forma aleatoria. Para los nodos siguientes, se calcula la distancia entre el nodo a posicionar y aquellos ya colocados, usando estos valores para generar “círculos” de posibles posiciones, cada uno centrado en un nodo ya ubicado con radio correspondiente a la distancia requerida.

El algoritmo busca interceptar dichos círculos, ya que los puntos de intersección representan candidatas naturales para la colocación del nuevo nodo, garantizando que se cumplan las restricciones de distancia. En caso de encontrar una o más intersecciones, se promedian sus coordenadas y se ajustan levemente con un desplazamiento aleatorio para evitar una distribución demasiado rígida; y si no se logra hallar intersecciones válidas, se recurre a un posicionamiento aleatorio basado en un pequeño paso a partir del último nodo colocado.

Este enfoque, que combina geometría y elementos de aleatoriedad, permite generar un plano 2D que respete en la medida de lo posible las distancias estipuladas en la matriz original de distancias basadas en los costos de combustible.

Cálculo de Centroides En el contexto del Problema de Ruteo de Vehículos (VRP), el cálculo de centroides es fundamental para caracterizar la distribución espacial de las granjas. Se calculan dos tipos principales de centroides que permiten extraer características importantes para la optimización del ruteo.

El centroide no ponderado representa el centro geométrico de todas las ubicaciones de las granjas. Se calcula como la media aritmética de todas las coordenadas de las granjas:

$$\mathbf{c}_{\text{no-ponderado}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad (13)$$

donde n corresponde al número de granjas y $\mathbf{x}_i$ al vector de coordenadas (x_i, y_i) de la granja i .

El centroide ponderado tiene en cuenta la producción de leche de cada granja, dando mayor influencia a las granjas con mayor producción:

$$\mathbf{c}_{\text{ponderado}} = \frac{\sum_{i=1}^n p_i \mathbf{x}_i}{\sum_{i=1}^n p_i} \quad (14)$$

donde p_i es el valor de producción de la granja i .

Distancia del centroide a la planta Esta característica mide la distancia normalizada entre el centroide no ponderado y la planta procesadora. Se calcula como:

$$d_{\text{centroide-planta}} = \frac{\|\mathbf{c}_{\text{no-ponderado}} - \mathbf{x}_{\text{planta}}\|}{D} \quad (15)$$

donde D es la diagonal del rectángulo delimitador (utilizada para normalización). Esta medida indica qué tan cerca o lejos está el centro de la distribución de granjas respecto a la planta procesadora.

Distancia media del centroide a las granjas Calcula la distancia promedio desde el centroide no ponderado a todas las granjas:

$$\bar{d}_{\text{centroide-granjas}} = \frac{1}{n} \sum_{i=1}^n \frac{\|\mathbf{c}_{\text{no-ponderado}} - \mathbf{x}_i\|}{D} \quad (16)$$

También se calcula la desviación estándar ($\sigma_{\text{centroide-granjas}}$) y el coeficiente de variación ($CV_{\text{centroide-granjas}} = \frac{\sigma_{\text{centroide-granjas}}}{\bar{d}_{\text{centroide-granjas}}}$) de estas distancias para caracterizar la dispersión espacial.

Centralidad de la planta Esta medida evalúa qué tan central está ubicada la planta procesadora con respecto a la distribución de las granjas del problema.

$$C_{\text{planta}} = 1 - \frac{d_{\text{centroide-planta}}}{d_{\text{centroide-planta}} + \bar{d}_{\text{centroide-granjas}}} \quad (17)$$

Un valor cercano a 1 indica que la planta está muy cerca del centroide de las granjas, mientras que un valor cercano a 0 indica que está lejos.

Estas características basadas en centroides capturan propiedades espaciales importantes de la distribución de granjas en relación con la planta procesadora, proporcionando información

útil para la caracterización de instancias y la optimización del problema de ruteo de vehículos asociado.

Para garantizar que estas características sean comparables entre instancias de diferentes tamaños y escalas geográficas, es crucial normalizarlas. La técnica empleada consiste en dividir las características espaciales basadas en distancia por la longitud de la diagonal del cuadro delimitador de la instancia [Rasku *et al.*, 2019, Kerschke *et al.*, 2018a].

Este enfoque logra la invarianza de escala, permitiendo que los modelos de aprendizaje automático se centren en la estructura relativa del problema en lugar de su tamaño absoluto. Si bien esta es una estrategia común y efectiva, para otras características —como las basadas en conteos o demandas no relacionadas con distancias— se deben aplicar otras formas de normalización.

La figura 5 muestra la distribución en el plano x-y de 10 instancias del problema. Dichas distribuciones fueron calculadas en base a los costos de transporte de cada instancia. El conjunto considerado contempla instancias pequeñas de a lo más 36 granjas, mientras que la superficie máxima a cubrir por los camiones es de 14 400 unidades cuadradas. En concreto, las instancias 1, 3, 7 y 9 presentan una distribución más compacta de sus granjas, mientras que las instancias 2, 4, 5, 8 y 10 presentan mayor dispersión.

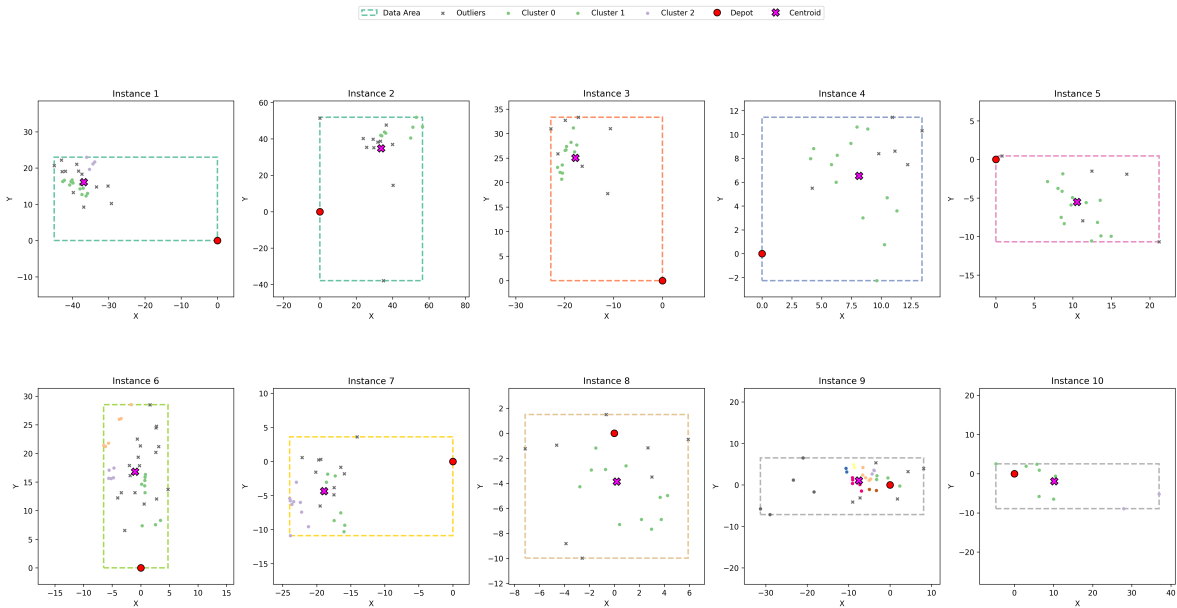


Figura 5: Plano cartesiano de las 10 primeras instancias que conforman el conjunto de datos de entrenamiento con área de operación entrelineas.

3.5.4. Características basadas en agrupamiento

Proceso de agrupamiento Además de la distribución espacial bruta, la forma en que los clientes se agrupan naturalmente debido a su proximidad geográfica constituye otra dimensión crucial de la estructura de una instancia del problema de ruteo de vehículos. La identificación de estos grupos puede revelar subproblemas naturales dentro de la instancia, sugerir descomposiciones efectivas, informar la construcción de soluciones iniciales (por ejemplo, asignando vehículos a grupos) o guiar el diseño de operadores de búsqueda local que operen dentro o entre grupos [Bello *et al.*, 2017]. Diferentes estructuras de agrupamiento implican diferentes retos para los algoritmos de ruteo y pueden requerir distintas estrategias de búsqueda.

Los algoritmos de agrupamiento basados en densidad, como OPTICS (*Ordering Points To Identify The Clustering Structure*), son particularmente adecuados para identificar estructuras de agrupamientos de forma arbitraria y densidades variables, a diferencia de métodos basados en centroides como *K-Means* [Campello *et al.*, 2013, Kriegel *et al.*, 2011].

OPTICS no produce una partición única, sino que construye un diagrama de alcanzabilidad que representa la estructura jerárquica de los datos, permitiendo la extracción de grupos a diferentes niveles de densidad. Para derivar características de agrupamiento significativas y consistentes, se utilizó *dynamicOPTICS* [Ankerst *et al.*, 1999], una variante que automatiza la selección de la estructura de grupos óptima basada en la optimización de una función de calidad de agrupamiento Q . Esta función busca un equilibrio entre la compacidad intra-grupos, la separación inter-grupos y la minimización de datos atípicos, definida por:

$$Q = \alpha \cdot D_{\text{out}} - \beta \cdot D_{\text{in}} - \gamma \cdot OR \quad (18)$$

donde D_{out} es la distancia promedio entre grupos (separación), D_{in} es la distancia promedio dentro de los grupos (compacidad), OR es el ratio de puntos clasificados como datos atípicos, y α, β, γ son coeficientes de ponderación [Halkidi *et al.*, 2001].

Las características de agrupamiento se derivan directamente de la estructura de grupos identificada por el algoritmo *dynamicOPTICS*. Estas métricas cuantifican propiedades de los grupos detectados, la presencia de datos atípicos y la calidad general de la estructura de agrupamiento, ofreciendo una perspectiva sobre la organización interna de los clientes que complementa las características espaciales globales.

La Tabla 4 lista las características de agrupamiento utilizadas en este estudio y su relevancia para entender la estructura de la instancia del problema de ruteo de vehículos y su impacto potencial en el rendimiento algorítmico.

Característica	Justificación
Ratio de grupos	Proporción de grupos respecto al total de clientes; sugiere fragmentación o concentración en la estructura espacial [Mester y Bräysy, 2007].
Ratio de datos atípicos	Proporción de clientes sin grupo. Identifica puntos aislados que dificultan la planificación directa.
Número promedio de clientes por grupo	Estima el tamaño típico de grupos, útil para considerar la división de la instancia en subzonas [Kerschke <i>et al.</i> , 2018a].
CV del número de clientes por grupo	Mide la variación en el tamaño de los grupos, importante para evaluar el desafío del balanceo de carga por vehículo [Campello <i>et al.</i> , 2013].
Distancia promedio intra-grupos	Evalúa qué tan compactos son los grupos internamente. Valores bajos favorecen rutas locales más eficientes.
Distancia promedio inter-grupos	Mide qué tan separados están los clústeres entre sí. Mayor distancia simplifica la asignación entre grupos pero puede aumentar el recorrido total [Halkidi <i>et al.</i> , 2001].
Índice de Silueta Promedio	Mide la calidad del agrupamiento según la cohesión dentro de los grupos y la separación entre ellos. Valores altos indican una estructura más definida.
Índice de Davies-Bouldin	Proporciona una relación entre la dispersión interna de los grupos y la separación entre sus centroides. Valores bajos indican un mejor agrupamiento [Davies y Bouldin, 1979].
Elongación de grupos	Indica si los clústeres tienden a ser alargados. Formas elongadas pueden dificultar la optimización de rutas internas.
Convexidad de grupos	Evalúa si los grupos tienen formas aproximadamente convexas. Formas irregulares complican la zonificación o asignación a rutas.
Distribución espacial de grupos	Indica si los grupos están distribuidos de manera uniforme o concentrados en ciertas regiones del área de servicio.

Tabla 4: Características derivadas del agrupamiento

Estas características ofrecen una visión cuantitativa de la estructura granular de la instancia del problema de ruteo de vehículos, complementando la perspectiva global proporcionada por las características espaciales. Al entender cómo los clientes se agrupan y las propiedades de estos grupos, los modelos de aprendizaje automático pueden inferir información sobre la dificultad local y global del problema, así como predecir qué estrategias algorítmicas (ej.: basadas en descomposición por clústeres) podrían ser más efectivas.

3.5.5. Características de Mezclado

Para analizar problemas de ruteo de vehículos con la complejidad adicional del mezclado de leche, es crucial cuantificar las propiedades inherentes de la instancia que afectan las decisiones de recolección y consolidación. Estas características estáticas clave describen la distribución de calidad respecto a células somáticas y volumen de la leche entre productores, influyendo directamente en las oportunidades y desafíos del mezclado para cumplir con los requisitos de los diferentes tipos de leche (A, B, C).

Característica	Descripción
Índice de Calidad (IC)	Dispersión relativa del conteo celular entre productores. Indica heterogeneidad de calidad.
Factor Contraste Calidad-Volumen (FCCV)	Relación entre la variabilidad de calidad y la variabilidad de volumen a nivel productor.
Porcentaje Producción Grado A Potencial (PPGAP)	Porcentaje del volumen total de leche con conteo celular $\leq$ cota A. Representa la cantidad de leche de alta calidad disponible antes del mezclado. (type_1_production_pct)
Porcentaje Producción Grado B Potencial (PPGBP)	Porcentaje del volumen total de leche con cota A $<$ conteo celular $\leq$ cota B. Leche de calidad intermedia. (type_2_production_pct)
Porcentaje Producción Grado C Potencial (PPGCP)	Porcentaje del volumen total de leche con conteo celular $>$ cota B. Leche de baja calidad. (type_3_production_pct)
Proximidad a Cota A (PCA)	Cuán cerca está el conteo celular promedio de la instancia de la cota A. Indica la "distancia" promedio a un umbral clave de calidad.
Proximidad a Cota B (PCB)	Cuán cerca está el conteo celular promedio de la instancia de la cota B. Indica la "distancia" promedio a otro umbral clave de calidad.
Índice de Complejidad de Mezclado (BCI)	Un índice agregado que evalúa la dificultad del mezclado considerando diversidad de calidades, variabilidad y alineación con la demanda.
Ratio de Alineación Calidad-Demanda (QDAR)	Relación entre la calidad promedio de la leche disponible y la calidad ponderada requerida por la demanda de los diferentes tipos de leche.
Dispersión Espacial de Calidad (SQD)	Cuantifica cómo la calidad de la leche varía espacialmente entre productores, considerando autocorrelación espacial y densidad.

Tabla 5: Características estáticas clave relacionadas con el mezclado

La Tabla 5 presenta las características propuestas relacionadas con el mezclado que se derivan directamente de los datos de la instancia.

Las características de mezclado se calculan directamente a partir de los datos de la instancia y resumen aspectos cruciales sobre la calidad y volumen de la leche disponible para el mezclado. A continuación, se detallan las principales métricas:

- **Índice de Calidad (IC):** Representa la dispersión relativa del conteo celular entre productores.

$$IC = \frac{\text{desviación estándar del conteo celular}}{\text{media del conteo celular}}$$

Un valor bajo indica homogeneidad en la calidad, mientras que un valor alto sugiere gran heterogeneidad, lo que puede ofrecer más oportunidades o mayores desafíos para el mezclado.

- **Factor Contraste Calidad-Volumen (FCCV):** Relaciona la variabilidad de calidad con la variabilidad de volumen a nivel productor.

$$FCCV = \frac{\text{desviación estándar del conteo celular}}{\text{desviación estándar de la producción}}$$

Indica si la variabilidad en calidad o en volumen es el factor dominante a considerar en la planificación de rutas y mezclado.

- **Proximidad a Cota A (PCA) y Proximidad a Cota B (PCB):** Miden la distancia del conteo celular promedio de la instancia a los umbrales clave de calidad ($Cota_A$, $Cota_B$).

$$PCA = \text{media del conteo celular} - Cota_A$$

$$PCB = \text{media del conteo celular} - Cota_B$$

Valores cercanos a cero o negativos para PCA/PCB indican que la calidad promedio está cerca o por debajo de los umbrales deseados, sugiriendo la viabilidad del mezclado. Valores positivos indican que la calidad promedio está por encima de los umbrales, señalando la necesidad de estrategias de dilución agresivas.

- **Índice de Complejidad de Mezclado (BCI):** Un índice agregado que combina la diversidad de tipos de leche, la variabilidad de calidad y la alineación con la demanda.

$$BCI \propto \text{diversidad de leche} \times \text{variabilidad de calidad} \times (1 + \text{alineación con demanda})$$

Busca capturar la dificultad general para satisfacer la demanda a través del mezclado, considerando la materia prima disponible y los requerimientos. Valores altos sugieren una instancia compleja donde el mezclado es crítico y desafiante.

- **Ratio de Alineación Calidad-Demanda (QDAR):** Compara la calidad promedio disponible con una calidad de referencia ponderada por la demanda de cada tipo de leche.

$$QDAR = \frac{\text{media del conteo celular ponderada por producción}}{\sum_{k \in \{A,B,C\}} Cota_k \times \text{ratio de demanda tipo } k}$$

Un valor cercano a 1 indica una buena correspondencia entre la calidad disponible y la requerida. Valores > 1 sugieren que la calidad disponible es, en promedio, inferior a la necesaria por la demanda.

- **Dispersión Espacial de Calidad (SQD):** Cuantifica cómo la calidad (conteo celular) varía geográficamente entre los productores, considerando la autocorrelación espacial.

$$\text{SQD} \propto \text{variabilidad de calidad} \times \frac{\text{autocorrelación espacial de calidad—distribución espacial general}}{\text{distancia promedio al vecino más cercano}}$$

Esta característica es fundamental para la planificación de rutas de recolección que busquen optimizar el mezclado a bordo o en el depósito. Un SQD bajo sugiere que productores cercanos tienden a tener calidades similares, facilitando el ruteo por zonas de calidad. Un SQD alto puede indicar una mezcla espacial de calidades, complicando la asignación de productores a rutas para lograr los objetivos de mezclado.

Las características de instancia, categorizadas en espaciales, de agrupamiento y de mezclado, ofrecen una visión multifacética de la estructura y complejidad de un problema de ruteo de vehículos con mezcla.

Las características espaciales como densidad, dispersión y forma nos informan sobre la *geometría del problema*. Numerosas relaciones respecto al ruteo de vehículos pueden verse interpretadas a través de métricas espaciales.

Las características de agrupamiento —como ratio de clústeres, ratio de datos atípicos y capacidad/separación de grupos— revelan la *estructura interna natural* basada en proximidad geográfica. Un alto ratio de grupos pequeños y numerosos datos atípicos puede indicar una instancia muy fragmentada, donde la asignación de clientes a rutas o grupos iniciales es un desafío primordial.

Las características de mezclado abordan las *particularidades operativas* del problema de recolección de leche con mezcla. PCA y PCB son indicadores directos de cuán lejos está la calidad promedio de los umbrales objetivo, señalando la necesidad e intensidad del mezclado.

Un PCA positivo alto, por ejemplo, indica que la leche promedio está significativamente por encima del límite para grado A, requiriendo una fuerte estrategia de dilución con leche de mejor calidad.

El indicador SQD es particularmente relevante, ya que cuantifica la correlación espacial de la calidad. Un valor alto de SQD implica que la calidad varía significativamente entre productores espacialmente cercanos, lo que complica la formación de rutas que puedan convenientemente recolectar leche para su posterior mezclado homogéneo o estratificado.

BCI y QDAR resumen la complejidad general y la alineación con los requerimientos de demanda, ofreciendo métricas agregadas sobre la dificultad del problema desde la perspectiva del mezclado.

En conjunto, estos indicadores permiten caracterizar la instancia, anticipar los principales desafíos de ruteo y mezclado, y guiar la selección o configuración de algoritmos para abordar eficazmente su complejidad inherente.

3.5.6. Características Eliminadas

El conjunto de datos original contiene 890 filas y 92 columnas, lo que conlleva riesgos de maldición de dimensionalidad así como ineficiencia computacional. Así que se descartaron indicadores irrelevantes como identificadores, predictores de demanda, proporciones de leches y características redundantes o altamente correlacionadas. Para reducir la dimensionalidad, se combinan tres criterios en el proceso de selección que permiten generar un conjunto de datos reducido:

1. **Relevancia de predictores en modelos basados en árboles (CART y XGBoost).** Solo se retiene variables que figuren entre las 45 más importantes en ambos métodos. CART evalúa la reducción de impureza Gini y XGBoost evalúa la ganancia promedio en la reducción de la función de pérdida.
2. **Umbral de varianza.** Se elimina variables con varianza $< 0,01$, pues aportan escasa señal discriminativa entre instancias. Este método identificó 27 características con escasa varianza.
3. **Correlación.** Para pares de variables con correlación $|r| > 0,95$, se conserva la variable considerada más interpretable o teóricamente relevante para el problema de ruteo de vehículos, y se descarta la otra, reduciendo así la multicolinealidad. Este enfoque permitió eliminar 3 dimensiones más de el conjunto de datos.

En base a estos métodos de limpieza se procedió a eliminar las características que estuvieran en común tanto en los métodos de CART, XGBoost y Umbral de varianza. Precedido por eliminación de identificadores y características que los investigadores sabían que no proveían valor a los modelos de ML, finalizado con el análisis de correlación.

El proceso en su conjunto permitió reducir el conjunto de características desde 92 a 39 características removiendo 53 dimensiones de el *dataset* original. La selección de características resultante busca mejorar la capacidad predictiva con la interpretabilidad y eficiencia computacional para las etapas posteriores de modelado.

A continuación se presenta la matriz de correlación entre las distintas características. Para una mejor visualización de la matriz de correlación se presenta en base a el conjunto de datos obtenido a través de la selección de características.

Se aprecia una mayor correlación entre predictores de un mismo tipo (espaciales, agrupamiento, mezcla) lo cual es de esperarse y también valores altos entre características de agrupamiento y espaciales lo cual tiene sentido ya que algunos de ellos son calculados en base a los mismos estadísticos.

Respecto a los parámetros del algoritmo a predecir, RCL no presenta una correlación significativa con ningún otro atributo lo cual puede ser indicativo de un bajo impacto en el desem-

SINTONIZACIÓN DE METAHEURÍSTICAS MEDIANTE APRENDIZAJE AUTOMÁTICO PARA PROBLEMAS DE RUTEO DE VEHÍCULOS MULTI-PRODUCTO CON MEZCLAS.

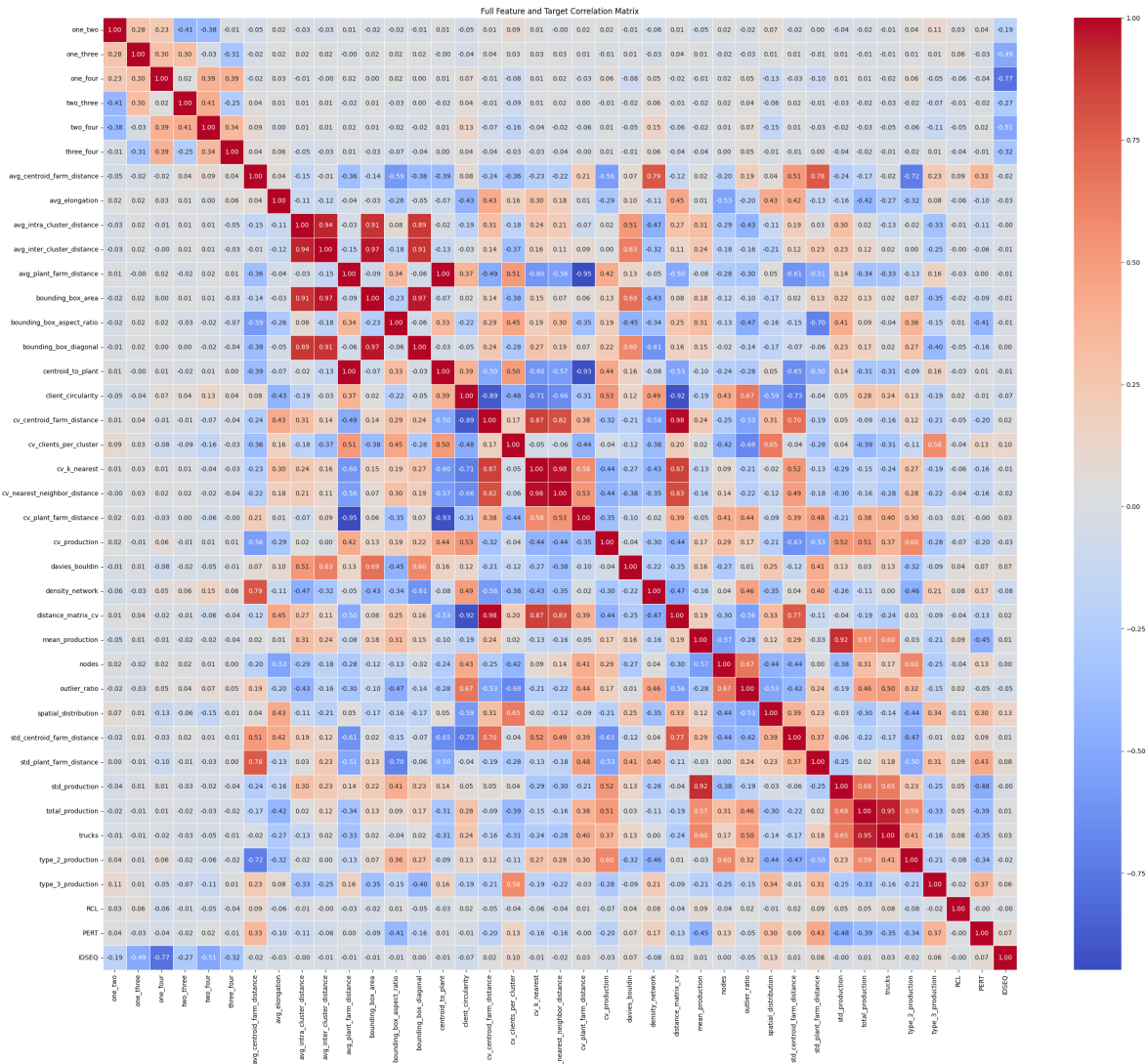


Figura 6: Matriz de correlación perteneciente al conjunto de datos analizado.

peño de la metaheurística, o falta de mayor volumen de datos para una mayor certeza a los valores de varianza reales.

El parámetro PERT define la cantidades de operadores o movimientos con el fin de proporcionar diversificación al algoritmo metaheurístico, se puede apreciar una correlación $\sim 0,4$ en características como *bounding_box_aspect_ratio* que puede deberse a que hay una relación entre la cantidad de operadores y el aspecto o forma de el terreno que contiene las granjas. a su vez puede denotar una relación entre el numero de operadores con el tamaño de la instancias. Otras caractetisticas identificadas son *mean_production*, *std_production*, *total_production* lo cual denota una relación interesante en torno a predictores de producción. Por último se reconocen correlaciones leves en torno a $\sim 0,35$ para los predictores que representan la producción de leche para tipo 2 y 3 de cada instancia.

El parámetro IDSEQ presenta correlaciones moderadas a intensas con los parámetros binarios que representan la activación de un par de operadores pues cada valor de IDSEQ puede ser mapeado a estas.

3.6. Diseño Propuesto de Modelos de Aprendizaje Automático

Esta sección detalla el diseño y las decisiones metodológicas en torno a la implementación de los modelos de aprendizaje.

3.6.1. Arquitectura General del Sistema

El procedimiento de aprendizaje automático propuesto adopta una estrategia heterogénea basada en árboles y redes neuronales artificiales. El diseño prioriza la simplicidad, portabilidad e interoperabilidad, implementando un flujo de trabajo modular que se ejecuta mediante pasos independientes organizados en cuatro fases secuenciales: (i) generación de datos, (ii) ingeniería de atributos, (iii) entrenamiento y optimización de modelos, y (iv) predicción de instancias no vistas.

En la figura 7 se contextualiza cómo este componente entra en el diseño del procedimiento de predicción de parámetros de este trabajo de título.

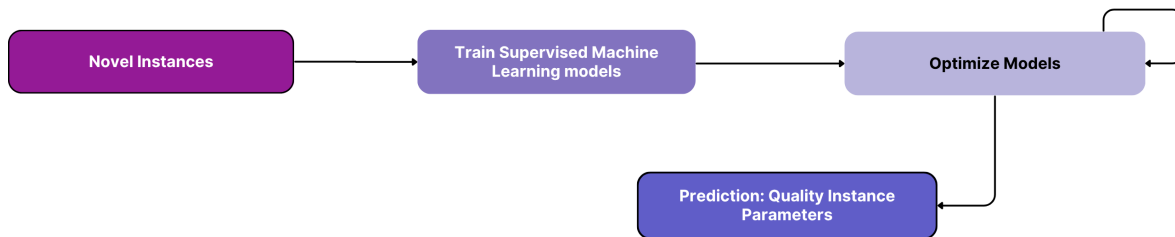


Figura 7: Diseño de alto nivel del entrenamiento y optimización de los modelos de aprendizaje.

La arquitectura dual permite la evaluación comparativa de algoritmos de aprendizaje automático clásicos versus redes neuronales bajo condiciones experimentales idénticas. Cada rama procesa independientemente el mismo conjunto de datos de entrada: una rama basada en potenciación de gradiente extremo (XGBoost) y otra fundamentada en redes neuronales *feed-forward*.

La decisión de implementar seis modelos especializados —tres por familia— se fundamenta en criterios de simplicidad de gestión y en la naturaleza intrínsecamente distinta de los problemas abordados. Esta segmentación permite la aplicación de funciones de pérdida específicamente diseñadas para cada tipo de tarea predictiva: regresión continua, clasificación multiclase y clasificación binaria multiple

Cada modelo puede optimizarse independientemente según las características estadísticas particulares de su dominio de aplicación.

La implementación tecnológica se sustenta en Python, integrando `scikit-learn` para la orquestación del procedimiento y preprocesamiento de datos, `XGBoost` para los modelos de potenciación de gradiente, y `Keras/TensorFlow` para la implementación de redes neuronales.

3.6.2. Diseño e implementación de modelos `XGBoost`

La rama `XGBoost` fue diseñada siguiendo principios de especialización funcional, donde cada modelo se optimiza específicamente para las características estadísticas y distribucionales de su tarea objetivo.

Predicción de RCL El parámetro RCL (*Restricted Candidate List*) presenta una naturaleza continua que requiere la predicción de valores en el dominio de los números reales positivos. Se implementa un `XGBRegressor` configurado con función objetivo de Error Cuadrático Medio (MSE). Esta función de pérdida, definida en la ecuación 1, es apropiada para tareas de regresión debido a su propiedad de no sesgo y su capacidad para penalizar proporcionalmente las desviaciones significativas mediante el término cuadrático. El algoritmo optimiza iterativamente los residuales del modelo mediante potenciación de gradiente, construyendo árboles de decisión que corrigen progresivamente los errores de predicción.

Clasificación multiclase para predicción de PERT La predicción de PERT constituye un problema de clasificación multiclase donde cada instancia pertenece exclusivamente a una de las categorías predefinidas. Se emplea `XGBClassifier` con función objetivo de entropía cruzada categórica (ecuación 12), que optimiza la verosimilitud logarítmica conjunta del modelo. Las probabilidades de clase p_k se calculan mediante la transformación softmax de la ecuación 9, generando una distribución de probabilidad normalizada sobre todas las clases posibles y garantizando la consistencia probabilística de las predicciones.

Clasificación multietiqueta para características binarias Las características binarias presentan independencia estadística entre etiquetas, permitiendo la descomposición del problema multietiqueta en múltiples problemas binarios independientes. Se utiliza la estrategia `MultiOutputClassifier` de `scikit-learn`, encapsulando un `XGBClassifier` configurado con función de pérdida logística binaria (ecuación 11), donde las probabilidades se obtienen mediante la función sigmoide de la ecuación 8. Esta arquitectura entrena seis clasificadores binarios especializados, cada uno optimizado independientemente mediante la minimización de la divergencia de Kullback-Leibler binaria [Shlens, 2014], lo que permite el tratamiento diferenciado y eficiente de cada característica.

3.6.3. Diseño e implementación de la rama de redes neuronales

El diseño de la rama neuronal prioriza la flexibilidad y la capacidad de aproximación no lineal, aprovechando el teorema de aproximación universal [Nishijima, 2021] para capturar patrones complejos en los datos.

Para cada uno de los modelos propuestos, se optó por usar Adam (*Adaptive Moment Estimation*) como el optimizador de cada red neuronal, ya que ofrece ventajas como simplicidad y eficiencia. Es un algoritmo de optimización de primer orden que adapta la tasa de aprendizaje para cada parámetro. Mantiene una estimación del primer momento (la media, m_t) y del segundo momento (la varianza no centrada, v_t) de los gradientes:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla L(\theta_{t-1}) \quad (19)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla L(\theta_{t-1}))^2 \quad (20)$$

Las estimaciones corregidas de sesgo $\hat{m}_t$ y $\hat{v}_t$ se utilizan para actualizar los parámetros θ :

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (21)$$

donde η es la tasa de aprendizaje, y β_1, β_2 son factores de decaimiento. El algoritmo Adam es robusto y eficiente, lo que lo ha convertido en el optimizador *de facto* para muchas aplicaciones de aprendizaje profundo.

Regresión neuronal para predicción de RCL La arquitectura neuronal implementada utiliza capas densas completamente conectadas organizadas en una estructura *feed-forward* secuencial mediante la API `keras.Sequential`. Las capas ocultas emplean activación ReLU (ecuación 7), aprovechando sus propiedades de gradiente constante para mitigar el problema del gradiente evanescente durante la retropropagación. La capa de salida utiliza activación lineal (ecuación 10) para permitir predicciones en todo el dominio real. El modelo se optimiza mediante el algoritmo Adam con función de pérdida MSE (ecuación 1), donde el algoritmo adapta automáticamente la tasa de aprendizaje para cada parámetro mediante la estimación de momentos del gradiente, proporcionando robustez ante gradientes ruidosos en espacios de alta dimensionalidad.

Clasificación multiclase neuronal para predicción de PERT La arquitectura neuronal mantiene capas ocultas con activación ReLU (ecuación 7) para preservar las propiedades de flujo de gradiente, mientras que la capa de salida implementa activación softmax (ecuación 9). Esta configuración genera una distribución de probabilidad válida sobre el espacio de clases, donde cada elemento de la salida representa la probabilidad condicional de pertenencia

a una clase específica. El modelo se optimiza mediante entropía cruzada categórica (ecuación 12) como función de pérdida y el algoritmo optimizador.

Clasificación multietiqueta neuronal para características binarias La arquitectura implementa seis neuronas de salida con activación sigmoide independiente (ecuación 8), permitiendo que cada salida represente una probabilidad binaria independiente en el rango $[0, 1]$. Esta configuración mantiene la independencia estadística entre etiquetas mientras permite el aprendizaje de representaciones compartidas en capas ocultas. El modelo utiliza entropía cruzada binaria (ecuación 11) como función de pérdida. El algoritmo optimizador Adam gestiona la optimización conjunta de todos los parámetros, balanceando automáticamente la contribución de cada tarea binaria al gradiente global durante la retropropagación.

La figura 8 representa de forma concisa los componentes principales y las familias de modelos de aprendizaje usados durante el desarrollo experimental.

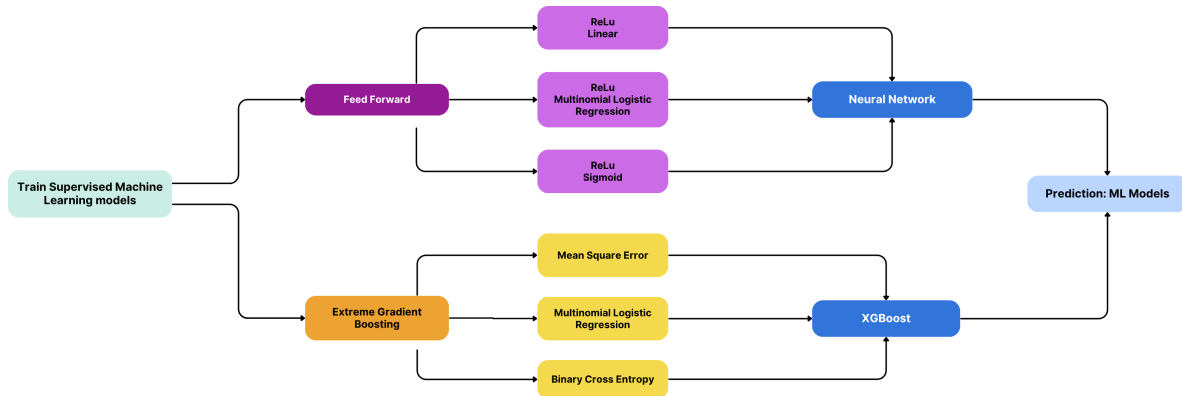


Figura 8: Diseño de alto nivel de la arquitectura particular de los modelos de aprendizaje automático.

3.6.4. Consideraciones metodológicas: modelos multiclase binarios

Dada la descomposición del parámetro IDSEQ en 6 operadores binarios independientes que representan la activación de pares de operadores, esta modificación genera 2^6 valores posibles. Sin embargo, sabemos de antemano que la permutación de 4 operadores es $4!$, lo que genera 24 órdenes posibles. Para restringir nuestra predicción a los valores válidos para la metaheurística, optamos por la minimización de el *Sum Squared Errors* (SSE) respecto a los vectores de etiquetas válidas. El vector que presenta menor error es elegido como el valor de IDSEQ. En otras palabras, comparamos el vector de probabilidades predicho por cada modelo con los 24 vectores estáticos válidos.

Este enfoque añade una capa de complejidad, pero proporciona interpretabilidad sobre los pares de operadores, lo cual es valioso.

3.7. Resumen

En este capítulo se presentó el algoritmo metaheurístico base del diseño experimental propuesto, el cual contiene componentes metaheurísticos de ILS y VNS. Además, este pertenece al estado del arte para la resolución del VRPMPWB.

Luego, se abordó la generación y validación de datos a través del sintonizador *Irace*. Seguido de esto, se realizó una ingeniería de atributos que organiza las características de los problemas en tres categorías fundamentales: espaciales, de agrupamiento y de mezcla. Se prosiguió con una limpieza o *pruning* del conjunto de características en pos de una reducción de dimensionalidad considerable.

Por último, se presentó la arquitectura de los modelos de aprendizaje automático, dividida en modelos de ensamble *XGBoost* y redes neuronales, en los cuales se diseñaron tres modelos por cada familia, especializados para cada tarea a predecir.

CAPÍTULO 4

EXPERIMENTOS Y RESULTADOS

4.1. Metaheurísticas objetivo

El algoritmo metaheurístico objetivo es el ILS propuesto por [Montero *et al.*, 2022]. Este es la base del experimento para el cual se entrenarán y validarán modelos de aprendizaje automático para la predicción de sus parámetros. Los parámetros que se buscan predecir u optimizar son *RCL*, *PERT* e *IDSEQ*. Las instancias de entrenamiento y validación corresponde a 5 mientras que las de prueba corresponden a [11,12,13,14,16,18,22,a62,a80], todas estan disponibles en [Montero, 2021].

4.2. Configuración Experimental y Arquitecturas de Modelos

La generación del conjunto de datos de entrenamiento se realiza mediante la ejecución del sintonizador Irace sobre el algoritmo metaheurístico ILS. Este proceso se llevó a cabo sobre 10 instancias distintas (véase 5), con 5 repeticiones para cada una, y se ejecutaron 20 ejecuciones paralelas utilizando diferentes semillas. Los resultados no repetidos de estas ejecuciones se usan para la construcción de los conjuntos de datos.

Posteriormente, a este conjunto de datos base se le enriqueció mediante la adición de características de instancias y la ingeniería de atributos, introduciendo nuevas características compuestas de tipo espacial, de agrupamiento y de mezcla. Finalmente, se aplicó un método de poda basado en la varianza para refinar el conjunto de características y reducir la dimensionalidad.

Los conjuntos de datos resultantes de este proceso han sido procesados con herramientas como *Jupyter Notebook*, utilizando bibliotecas de *Python* como *Pandas*, *NumPy*, *Matplotlib*, *Scikit-learn* y *SciPy*.

Para la construcción y medición de los modelos predictivos se usaron *API's* de bibliotecas como *xgb*, *Scikit-learn*, *Seaborn* y *TensorFlow/Keras*. Los datos se dividieron en conjuntos de entrenamiento y prueba. Se utilizaron semillas aleatorias para garantizar la reproducibilidad de las divisiones. La proporción del conjunto de prueba se estableció en 35 % para los modelos de Redes Neuronales (NN) y en 20 % para los modelos XGBoost (XGB).

En cuanto al preprocesamiento de las características, las variables numéricas fueron estandarizadas utilizando *StandardScaler* de *Scikit-learn*. Es importante destacar que este escalador se ajustó exclusivamente con los datos del conjunto de entrenamiento para prevenir cualquier fuga de información (*data leakage*) hacia el conjunto de prueba.

4.3. XGBoost

4.3.1. Análisis de hiperparámetros de XGBoost

Se realizó un barrido sistemático de hiperparámetros para optimizar el rendimiento de los modelos XGBoost previo a la fase de predicciones. Este proceso se centró en tres hiperparámetros clave: `learning_rate` (tasa de aprendizaje), `n_estimators` (número de árboles en el ensamble) y `max_depth` (profundidad máxima de cada árbol). Los resultados de estos análisis se presentan gráficamente a continuación.

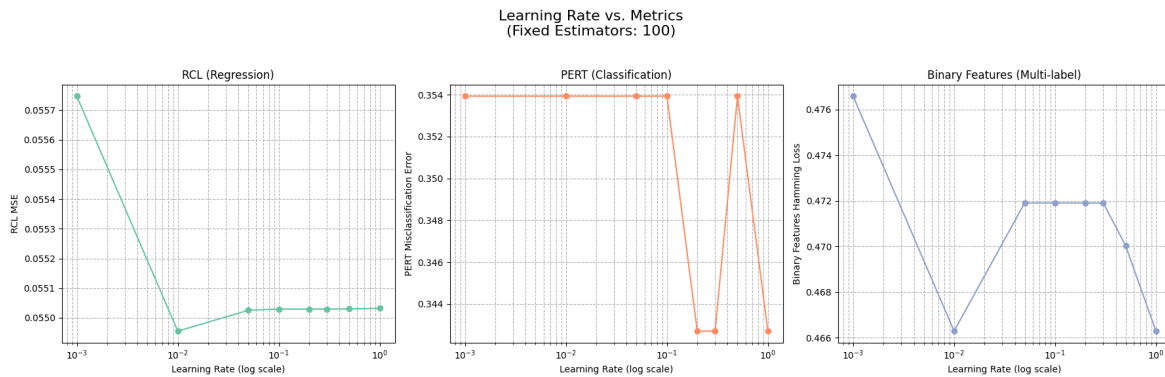


Figura 9: Impacto de la tasa de aprendizaje (`learning_rate`) en las métricas de rendimiento de los modelos XGBoost.

La Figura 9 muestra cómo se mide el impacto de la tasa de aprendizaje en la función de pérdida de cada modelo. Tanto para el modelo de regresión como el multi-etiqueta, el *learning rate* óptimo con 100 estimadores es 10^{-2} , ya que minimiza el *Mean Square Error* y el *Hamming Loss*, respectivamente.

Para el modelo de clasificación, los valores que minimizan el error de *misclassification* son 0.5 y 1. Se usará 0.5 como su óptimo para no dar saltos tan grandes en el entrenamiento final. Además, se percibe una tendencia a realizar saltos en la curva para los modelos clasificadores, mientras que el modelo de regresión presenta una tendencia a una curva más suavizada.

La Figura 10 muestra el impacto del número de estimadores, o en otras palabras, del número de árboles sobre las funciones de pérdida respectivas. Para el modelo de regresión, se aprecia una mejor minimización del error con un número de estimadores alrededor de 25, y luego se vuelve constante a partir de 50 estimadores. El modelo de clasificación del parámetro PERT presenta un comportamiento con saltos bien definidos en su curva; los valores ideales se encuentran en el rango de 80 a 200 estimadores y también con 500.

Dada la preocupación por el uso de recursos, se optará por 80 estimadores. En contraste, el modelo multi-etiqueta binario es indiferente al número de estimadores y constante en

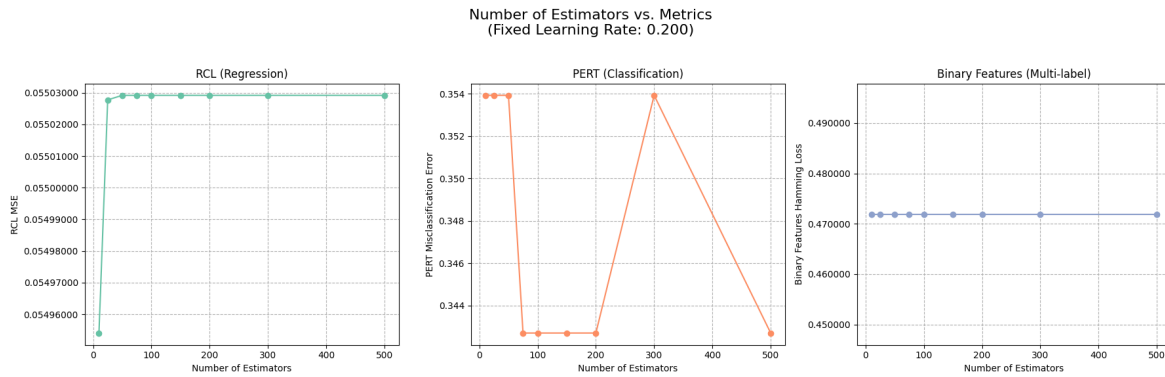


Figura 10: Impacto del número de estimadores ($n_estimators$) en las métricas de rendimiento de los modelos XGBoost.

todo su codominio, por lo que se trabajará con 5 estimadores para este caso. Esto puede ser indicativo de que no se está sacando provecho de la naturaleza de utilizar múltiples árboles, o que la variable aleatoria es muy simple y agregar más árboles solo genera gasto en recursos y sobreajuste.

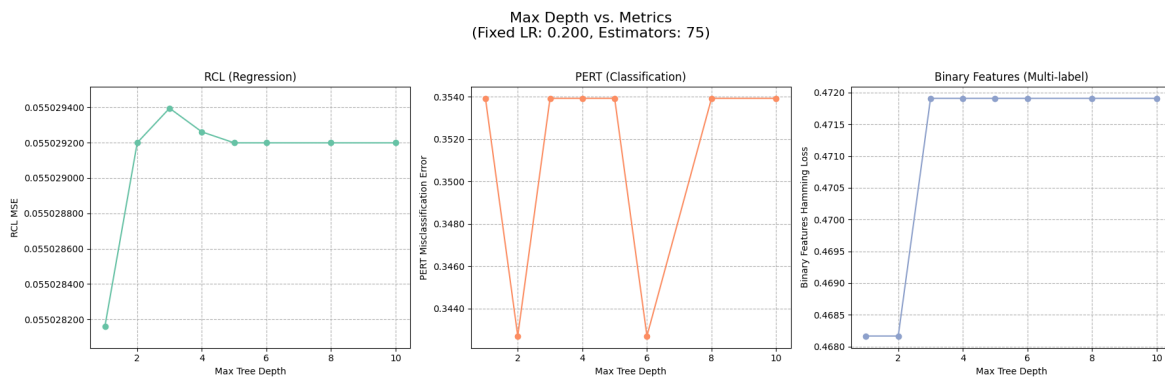


Figura 11: Impacto de la Profundidad Máxima del Árbol (max_depth) en las métricas de rendimiento de los modelos XGBoost.

La Figura 11 muestra el impacto de la profundidad máxima de los árboles en las métricas respectivas. El modelo de regresión presenta una tendencia al alza hasta una profundidad máxima de tres, luego decae levemente y se mantiene constante. El cambio en el MSE es casi irrelevante, ya que se analiza considerando 8 decimales de precisión y varían a lo más los últimos cuatro. Ya que con profundidad 1 generalmente existe un riesgo de *underfit*, se elige utilizar una profundidad máxima de 5 para el problema de regresión.

El caso del modelo de clasificación del parámetro PERT es más interesante: se nota un valor constante con bajadas considerables para dos valores de profundidad. Los valores óptimos son 2 y 6. Se realizarán las pruebas con 6. Por último, el modelo multietiqueta binario presenta una curva similar a la sigmoidal. Lo cual es útil es pos de la simplicidad.

4.3.2. Configuraciones de XGBoost

En base a los gráficos anteriores, los modelos XGBoost se configuraron con los siguientes hiperparámetros:

- **Modelo RCL (Regresión):** Configurado con una tasa de aprendizaje de 0.01, 25 estimadores y una profundidad máxima de 5 por árbol. Se utilizó la métrica de evaluación `rmse` y el objetivo `reg:squarederror`.
- **Modelo PERT (Clasificación Multiclase):** Los hiperparámetros finales fueron una tasa de aprendizaje de 0.5, 80 estimadores y una profundidad máxima de 6. Se empleó `mlogloss` como métrica de evaluación, con un objetivo `multi:softmax`.
- **Modelo de Características Binarias (Clasificación Multietiqueta):** Implementado mediante `MultiOutputClassifier` de Scikit-learn, que envuelve un estimador base `xgb.XGBClassifier`. Para cada clasificador base, se usó una tasa de aprendizaje de 0.01, 5 estimadores y una profundidad máxima de 2. La métrica fue `logloss` y el objetivo `binary:logistic`.

4.3.3. Importancia de Características en XGBoost

La importancia de características es una técnica que permite cuantificar la contribución de cada variable de entrada a las predicciones del modelo. En XGBoost, esta importancia se calcula típicamente basándose en la frecuencia con la que una característica se utiliza para realizar una división en los árboles (*gain*), el número total de veces que una característica aparece en los árboles (*weight*), o la ganancia promedio de las divisiones que utilizan la característica (*cover*).

La Figura 12 muestra el impacto de las características más importantes en las decisiones del modelo de regresión. Se puede apreciar que la producción media de la instancia es el predictor más importante para discernir en el modelo, además de la circularidad de los clientes y la desviación estándar de la distancia planta-granja. Estos suman alrededor de 0.9 de la importancia de características.

La Figura 13 muestra el impacto que toman ciertas características en la decisión del modelo de clasificación. Se puede apreciar que el coeficiente de variación de distancia al vecino más cercano es el predictor más importante para el modelo, explicando alrededor del 0.5 de las decisiones. Le siguen la desviación estándar de producción, el número de camiones, el ratio de datos atípicos y la desviación estándar de la distancia planta-granja. La suma acumulada de estas características representa alrededor de 0.96.

En base a esto, el modelo está fuertemente influenciado por predictores espaciales, de agrupamiento y características de producción.

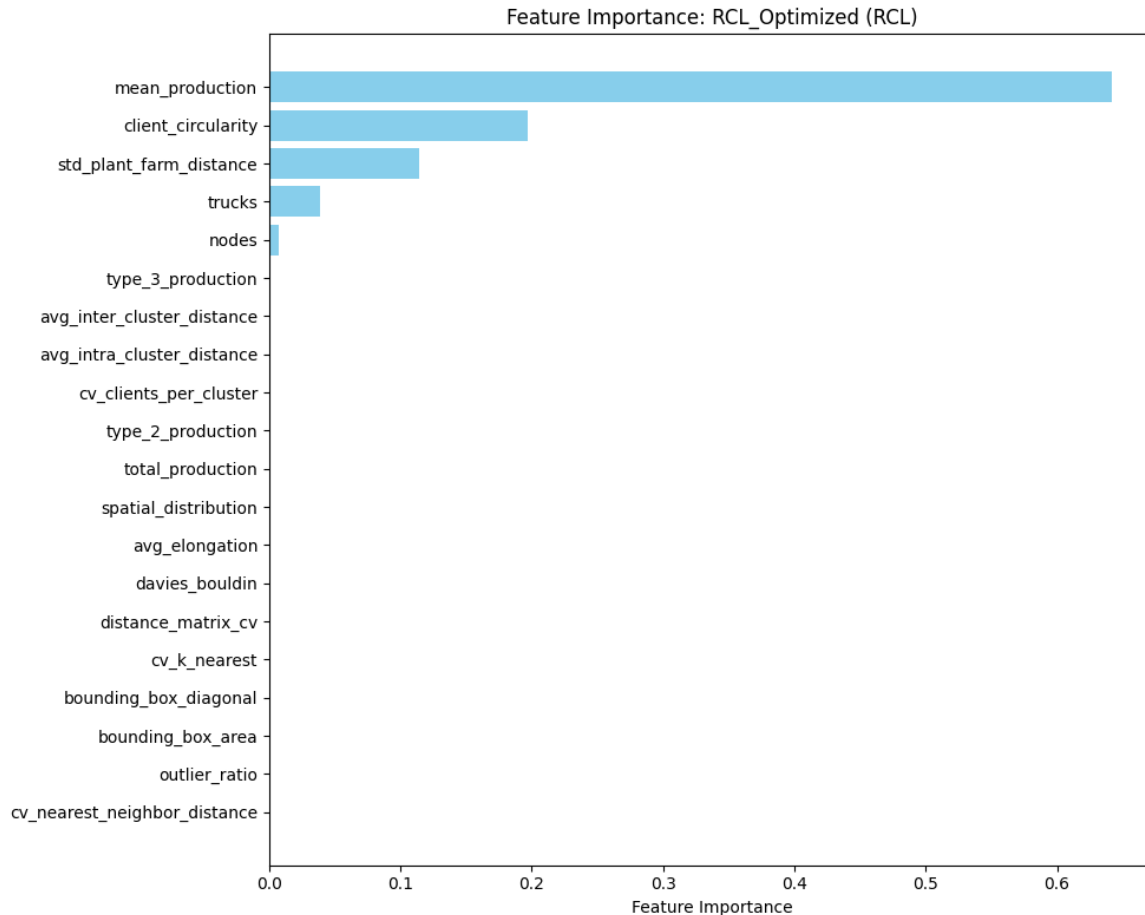


Figura 12: Importancia de las características para el modelo XGBoost optimizado de predicción del parámetro RCL.

La Figura 14 detalla la influencia de diversas características en la activación de los pares de operadores. Para las variantes del operador 2-opt, se observa que 2opt-Reorder es sensible a estadísticos espaciales compuestos y datos atípicos. Por su parte, 2opt-Exchange está determinado en un 80 % por la desviación estándar de la distancia planta-granja y el coeficiente de variación de clientes por grupo.

En el caso de 2opt-Or_opt, la activación depende de métricas de agrupamiento como el índice de Davies-Bouldin y la circularidad de los clientes, además de información espacial. Un predictor común y relevante para la activación de las variantes de 2-opt es la desviación estándar de la distancia entre la planta y las granjas.

Se identifica que el par Reorder-Exchange es influenciado principalmente por predictores de información global de los grupos, como el índice de Davies-Bouldin y estadísticos espaciales compuestos. Para Reorder-Or_opt, la elongación promedio y otros estadísticos espaciales compuestos acumulan el 95 % del puntaje de importancia, sugiriendo que la disposición geométrica de los grupos es fundamental para su activación.

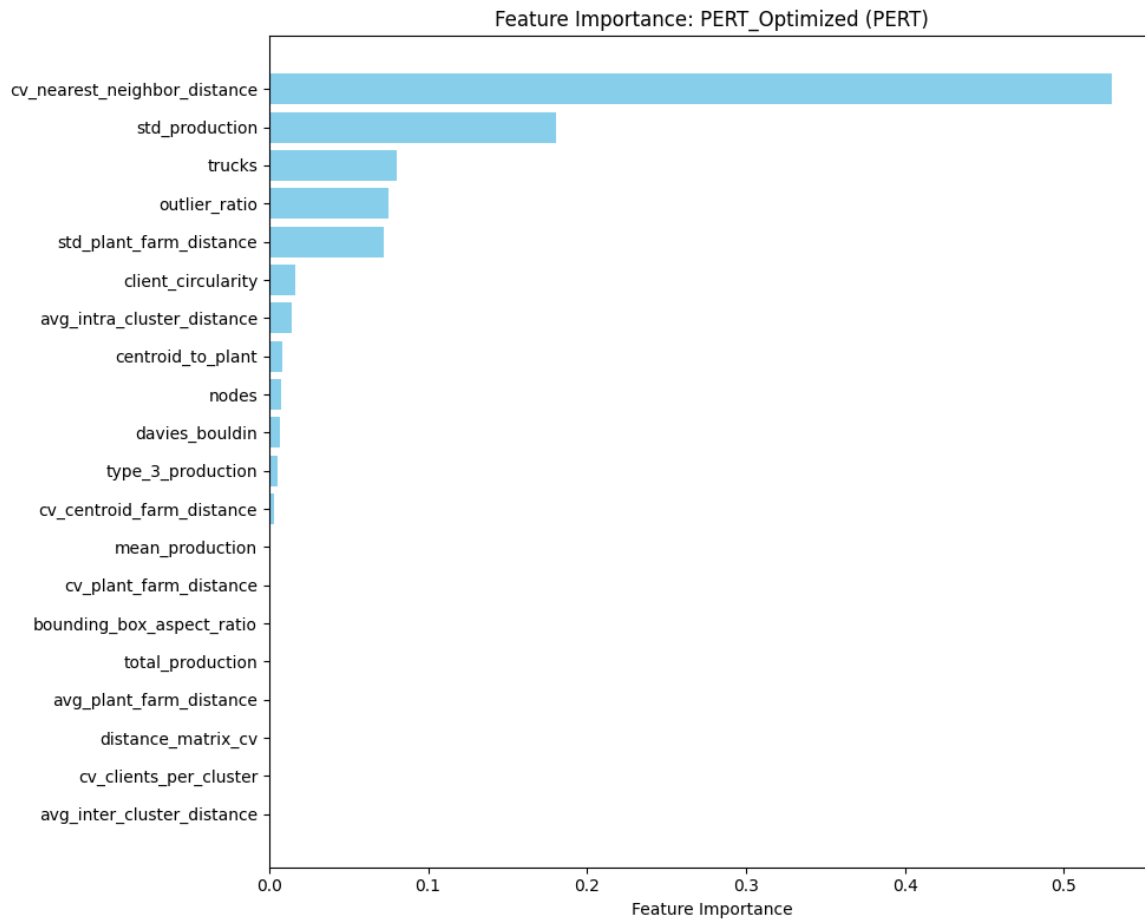


Figura 13: Importancia de las características para el modelo XGBoost optimizado de predicción del parámetro PERT.

De manera similar, en Exchange-Or_opt, la circularidad de los clientes y la distancia centroide-planta explican más del 95 % de la importancia acumulada. Esto permite inferir que el operador Or_opt requiere información estructural del clúster para ser efectivo.

Finalmente, se concluye que predictores simples como el número de granjas y camiones, a pesar de su simplicidad, son consistentemente relevantes para la mayoría de los operadores analizados.

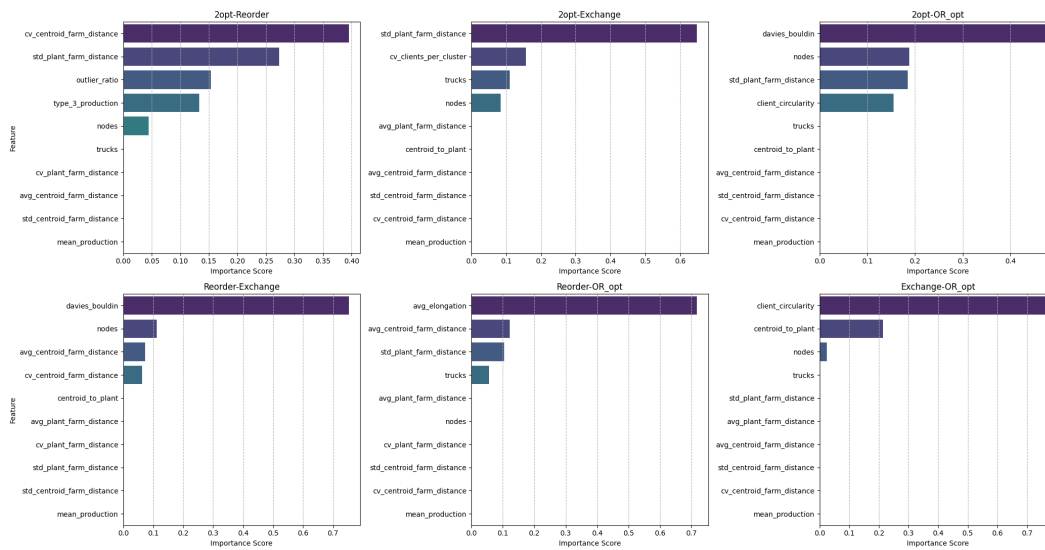


Figura 14: Importancia de las características para cada una de las seis salidas del modelo XGBoost de características binarias.

Resultados de los Modelos XGBoost

Tras el entrenamiento y la optimización de hiperparámetros, los modelos finales de XGBoost se evaluaron en el conjunto de prueba. A continuación, se presentan las métricas de rendimiento y las visualizaciones correspondientes.

4.3.4. Modelo de Regresión para RCL

El modelo de regresión muestra un rendimiento limitado en sus predicciones. El Error Absoluto Medio (MAE) es de 0.2025, lo que representa una desviación promedio moderadamente alta; de hecho, este error representa aproximadamente el 25.3 % del rango total de la variable objetivo. El Error Cuadrático Medio (RMSE) es ligeramente superior, con 0.2355, y la proximidad entre ambas métricas sugiere que el modelo es consistente en sus errores y no genera predicciones extremadamente inexactas o atípicas para grupos pequeños de datos.

Sin embargo, la métrica más reveladora es el coeficiente de determinación (R^2), que es de apenas 0.0083. Este valor tan bajo indica que el modelo es incapaz de explicar la variabilidad de los datos, capturando menos del 1 % de la varianza del parámetro RCL. Este pobre desempeño confirma lo que se esperaba, ya que es consistente con la baja correlación que se observó previamente entre las características de entrada y el objetivo a predecir.

Modelo Target	Métrica	Valor
RCL	MSE	0.0555
	RMSE	0.2355
	MAE	0.2025
	R2	0.0083
PERT	Accuracy	0.6573
	F1-score (Macro)	0.4765
	F1-score (Weighted)	0.6308
	Precision (Macro)	0.4973
	Precision (Weighted)	0.6273
	Recall (Macro)	0.4756
	Recall (Weighted)	0.6573
Binary Features (Overall)	Subset Accuracy	0.0337
	Hamming Loss	0.4766
	F1-score (Macro)	0.5740
	F1-score (Weighted)	0.5819
	Precision (Macro)	0.4382
	Precision (Weighted)	0.4446
	Recall (Macro)	0.8333
	Recall (Weighted)	0.8432
	Jaccard Score (Macro)	0.4382
	Jaccard Score (Weighted)	0.4446

Tabla 6: Métricas de rendimiento de los modelos XGBoost finales optimizados en el conjunto de prueba.

4.3.5. Modelo de Clasificación para PERT

El modelo desarrollado para predecir la característica discreta del parámetro PERT (con rango de 1 a 4) demuestra un rendimiento moderado, como se muestra en la Tabla 6.

Una exactitud (*Accuracy*) general de 0.6573 indica una clasificación correcta en aproximadamente dos tercios de los casos. El F1-score ponderado de 0.6308 y la precisión ponderada de 0.6273, junto con un Recall ponderado de 0.6573, sugieren un rendimiento relativamente equilibrado cuando se considera la distribución de clases.

Sin embargo, el F1-score macro de 0.4765 (acompañado de una precisión macro de 0.4973 y un Recall macro de 0.4756) es notablemente más bajo. Esto es indicativo de desequilibrios de clase, probablemente afectando el rendimiento en las clases menos representadas, lo cual se puede confirmar con la matriz de confusión.

Si bien la capacidad predictiva actual es sólida para las clases mayoritarias, estas métricas

sugieren que estrategias específicas para abordar el desequilibrio de datos podrían mejorar aún más el rendimiento general, especialmente para las clases subrepresentadas.

4.3.6. Modelo de Características Binarias para la Predicción de Secuencia de Operadores

El rendimiento global de los seis clasificadores binarios independientes se presenta en la sección “*Binary Features (Overall)*” de la Tabla 6, mientras que las métricas individuales por característica se detallan en la Tabla 7.

La Exactitud de subconjunto (*Subset Accuracy*) de 0.0337 (Tabla 6) indica que el modelo predice correctamente la combinación completa de las seis características binarias de forma simultánea en solo el 3.37 % de los casos. Esto subraya la dificultad de la tarea multi-etiqueta antes del paso de mapeo a IDSEQ. Consistentemente, una Pérdida de Hamming (*Hamming Loss*) de 0.4766 sugiere que, en promedio, casi la mitad de las etiquetas binarias individuales por instancia son incorrectas.

A pesar de esto, las métricas agregadas como el F1-score ponderado de 0.5819 y el Recall ponderado de 0.8432 muestran un nivel moderado de rendimiento a nivel de predicción de las características individuales, considerando el soporte de clases. El alto recall (Macro y Ponderado) sugiere que los modelos tienden a identificar correctamente los casos positivos para las características binarias individuales, aunque esto puede venir a costa de una menor precisión (Macro de 0.4382 y Ponderada de 0.4446).

Característica Binaria (Operador)	F1-Score	Precision	Recall
2opt-Reorder	0.7240	0.5674	1.0000
2opt-Exchange	0.7055	0.5449	1.0000
2opt-OR_opt	0.6815	0.5169	1.0000
Reorder-Exchange	0.0000	0.0000	0.0000
Reorder-OR_opt	0.6566	0.4888	1.0000
Exchange-OR_opt	0.6766	0.5112	1.0000

Tabla 7: Métricas de rendimiento por característica individual para el Modelo XGBoost de Características Binarias Optimizado.

Al analizar las métricas por característica individual (Tabla 7), se observa un recall perfecto (1.0000) para la mayoría de las características, excepto para “*Reorder-Exchange*”, que el modelo no logra predecir en absoluto. Este recall perfecto para las otras cinco características sugiere que los modelos individuales rara vez fallan en identificar una característica cuando esta está presente. Sin embargo, la precisión, que varía entre 0.4888 y 0.5674 para estas cinco, indica una cantidad considerable de falsos positivos. Esto se refleja en los valores de F1-Score y Jaccard Score. La característica “*2opt-Reorder*” muestra el mejor balance con un F1-Score de 0.7240.

Estas métricas reflejan el rendimiento de los seis clasificadores binarios independientes; la eficacia final en la predicción de la permutación IDSEQ correcta está influenciada sustancialmente por la efectividad de la función de mapeo subsecuente que interpreta estas salidas binarias. La falla total en predecir “*Reorder-Exchange*” es un punto crítico a investigar, ya que podría deberse a una extrema rareza de esta característica en el conjunto de entrenamiento o a dificultades inherentes del modelo para aprenderla.

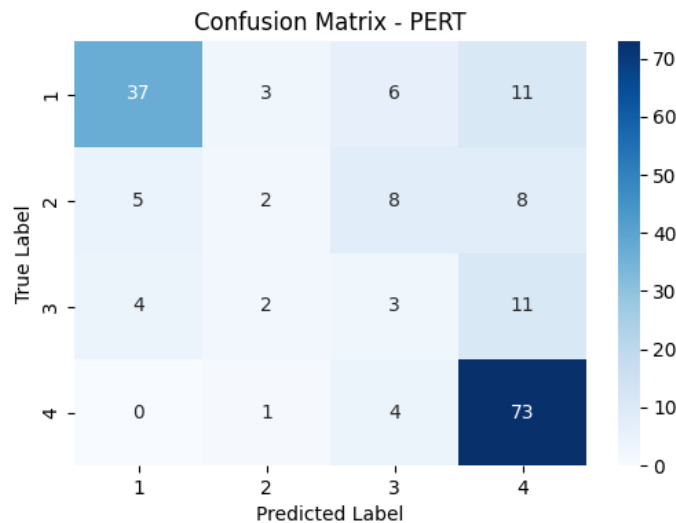


Figura 15: Matriz de Confusión para el modelo XGBoost de predicción de PERT (conjunto de datos reducido).

La Figura 15 muestra cómo se distribuyen las predicciones y el soporte de las distintas clases en una matriz de confusión. La matriz presenta un desbalance importante en su comportamiento. Predice correctamente más del 70 % de los casos de la clase 4, seguido por alrededor del 40 % de la clase 1. Luego, la clase 2 tiene muy mal desempeño. Esto se debe a que los modelos presentan sesgos hacia las clases 4 y 1, pues la mayoría de datos de entrenamiento presentan estas clases respectivas; es decir, existe un desbalance en el conjunto de datos, lo que transmite este comportamiento.

Sin embargo, en base a la Tabla 6 se nota que el *F1 Score ponderado* presenta un valor de 63 %, lo cual es decente y permite predecir correctamente 6 de cada 10 casos de forma general.

La Figura 16 muestra las predicciones del modelo para los pares de operadores en el conjunto de prueba, particularmente la matriz de confusión binaria para cada par de operadores.

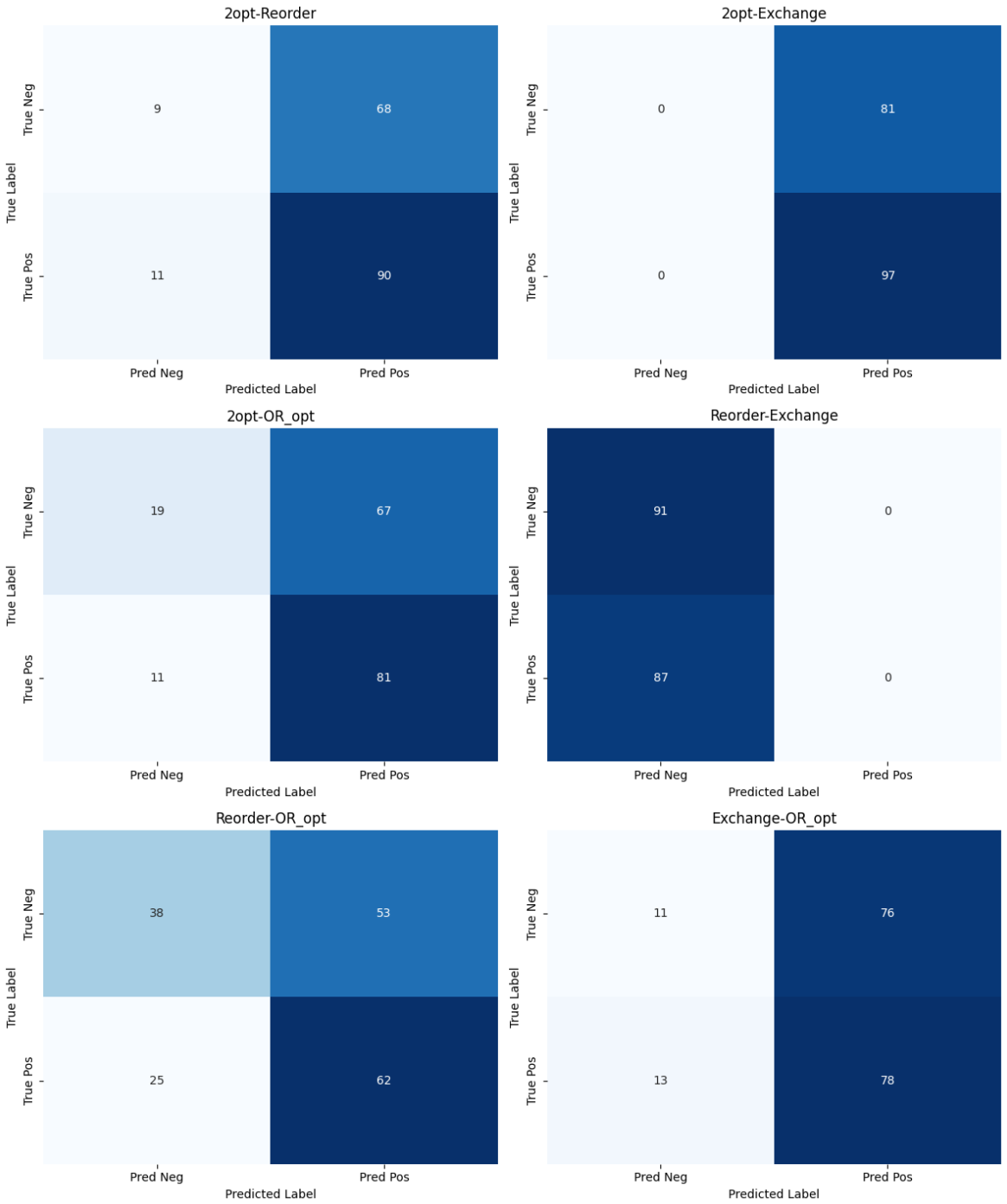


Figura 16: Matrices de Confusión Agregadas para cada una de las seis Características Binarias predichas por el modelo XGBoost.

Se aprecia que todos los modelos presentan un sesgo hacia alguna columna, es decir, tienen sesgos hacia la clase positiva o negativa.

Los modelos que deben predecir la activación de los operadores 2opt o OR_opt presentan un sesgo hacia la clase positiva o 1. En cambio, el modelo que debe predecir la activación de los operadores *Reorder-Exchange* tiene un sesgo total hacia la clase negativa o 0. Esto puede deberse a la distribución de esta clase en el conjunto de entrenamiento.

En base a esta figura, el mejor modelo es el que predice *Reorder-Or_opt*, ya que presenta la mejor distribución en su diagonal.

4.4. Redes Neuronales

Las redes neuronales (NN) son modelos computacionales inspirados en la estructura y función del cerebro humano. Consisten en capas interconectadas de "neuronas." nodos, que procesan información y aprenden patrones complejos a partir de los datos a través de la optimización de pesos sinápticos. En esta implementación, se utilizaron redes neuronales artificiales de tipo *Feed Forward*, particularmente se utilizó la implementación de *Sequential* de la librería *Keras* de *Tensorflow* que son adecuadas para tareas de regresión y clasificación.

4.4.1. Análisis de Hiperparámetros de Redes Neuronales

Se llevó a cabo un análisis de hiperparámetros para los modelos de redes neuronales. Este proceso es fundamental para encontrar arquitecturas y configuraciones de entrenamiento que generalicen bien a datos no vistos. La exploración de hiperparámetros para las redes neuronales incluyó la variación del número de capas ocultas, neuronas por capa, funciones de activación, tasas de aprendizaje, épocas y tamaños de lote, con el objetivo de identificar configuraciones robustas para cada tarea.

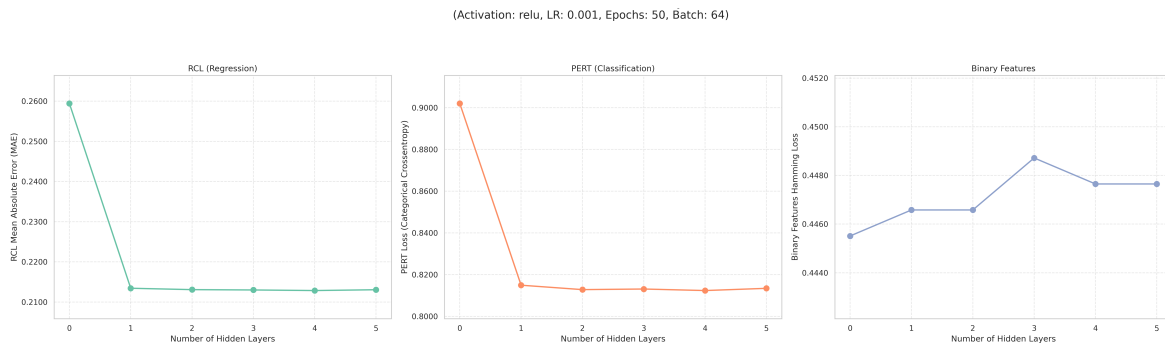


Figura 17: Impacto del número de capas ocultas en las métricas de rendimiento de las redes neuronales.

La Figura 17 muestra el impacto de la cantidad de capas ocultas respecto a las métricas respectivas de cada modelo. Para el modelo de regresión, el menor error absoluto se obtiene con cuatro capas ocultas, pero la ganancia es marginal respecto a dos capas, por lo que se prefiere simplicidad para la predicción de esta variable. De manera similar, la 'Cross Entropy Loss' del modelo de clasificación para el parámetro 'PERT' alcanza el mejor balance entre rendimiento y uso de recursos con dos capas ocultas.

El modelo multiclase para predecir el orden de los operadores obtiene el mejor rendimiento sin capas ocultas. Sin embargo, se prevé un posible *underfit* si el modelo no cuenta con capas internas que le permitan formar representaciones propias del orden de los operadores.

Se procede con una capa oculta, ya que permite al modelo captar mayor complejidad. A pesar de el leve aumento en 10^{-3} en la función de pérdida, el modelo continúa siendo simple.

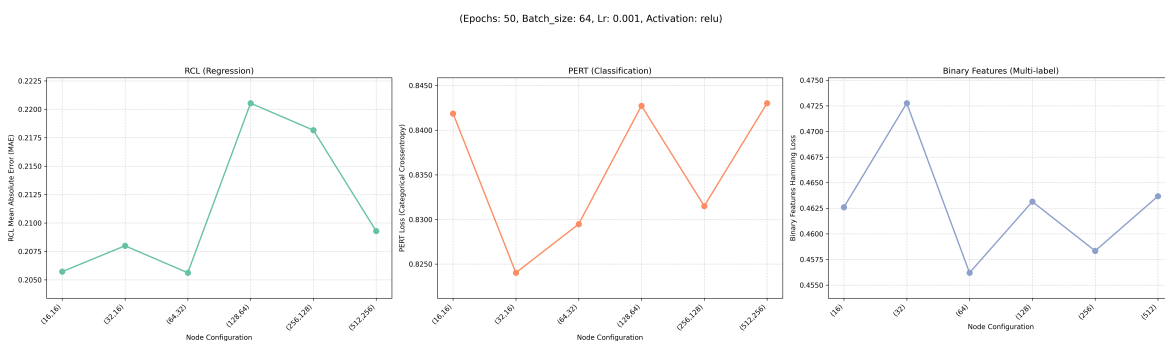


Figura 18: Impacto del número de neuronas en las capas ocultas respecto a las métricas de rendimiento de las redes neuronales.

La Figura 18 muestra el impacto de la cantidad de nodos o neuronas en cada capa oculta y cómo estos afectan las métricas de rendimiento. Para el modelo de regresión, se observa el mejor desempeño con la configuración (256,128), aunque puede resultar computacionalmente costosa. Sin embargo, la ganancia frente a la configuración (64,32) es marginal, lo que hace de esta última una opción más eficiente.

Para el modelo de clasificación que predice el parámetro 'PERT', se percibe una mejora en la métrica de pérdida a medida que aumenta la cantidad de nodos, aunque esta mejora se estabiliza después de (256,128).

Para el modelo multietiqueta, la configuración óptima es (256), ya que en esta se minimiza la *Hamming Loss*. El comportamiento de este modelo es distinto a los anteriores: empeora entre 16 y 32 nodos, mejora progresivamente hasta 256 y luego se mantiene constante hasta 512 nodos.

La Figura 19 compara las funciones de activación 'ReLU', 'Tanh' y 'Sigmoid' y cómo su uso impacta en las métricas de rendimiento. Para el modelo de regresión, las funciones que mejor minimizan el error absoluto son 'ReLU' y 'Sigmoid'. En el caso del modelo de clasificación para

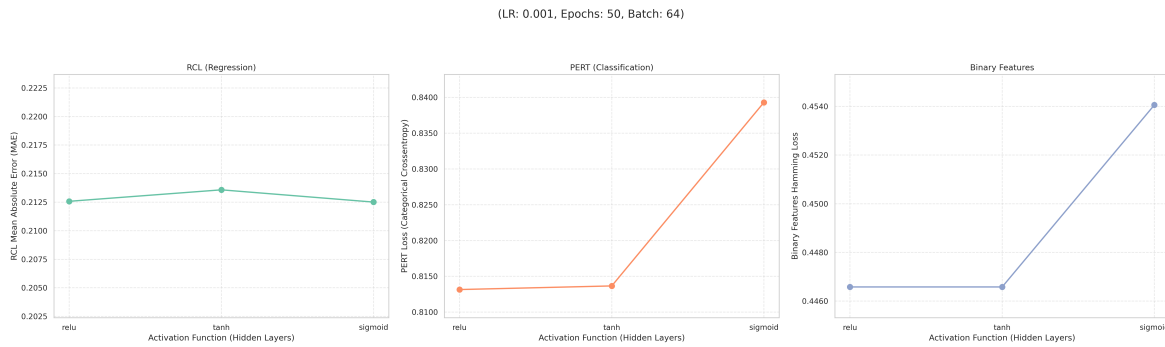


Figura 19: Impacto de la función de activación en capas ocultas (ReLU, Tanh, Sigmoid) en las métricas de rendimiento de las redes neuronales.

el parámetro 'PERT', se logra la mejor precisión con las funciones 'ReLU' y 'Tanh'. De manera similar, para el modelo binario multietiqueta, las funciones que minimizan la *Hamming Loss* son 'ReLU' y 'Tanh'. Dado que 'ReLU' demuestra los mejores resultados en los tres modelos y es preferida por su simplicidad, será la función de activación a utilizar en todos los modelos.

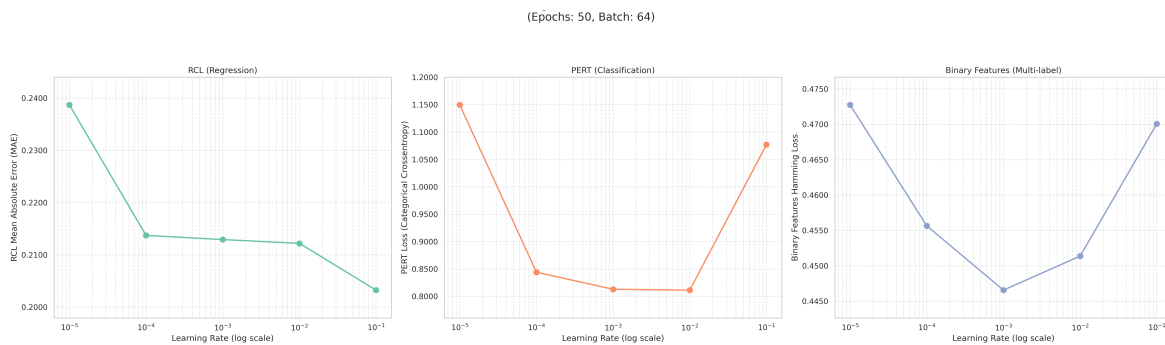


Figura 20: Impacto de la tasa de aprendizaje en las métricas de rendimiento de las redes neuronales (RCL: MAE, PERT: Accuracy, Binarias: Hamming Loss). Arquitectura fija: [128, 64], Activación ReLU, 50 épocas, lote 64.

La Figura 20 muestra el impacto de la tasa de aprendizaje en las métricas de rendimiento de cada modelo. Para el modelo de regresión, la mejor tasa de aprendizaje se obtiene en 10^{-1} . En cambio, para el modelo de clasificación del parámetro PERT, se observa el mejor desempeño con un valor de 10^{-2} . Para el modelo multietiqueta, el gráfico presenta un comportamiento en forma de 'V', con la minimización de la *Hamming Loss* en 10^{-3} .

La Figura 21 muestra el impacto del número de épocas sobre las funciones de pérdida. El gráfico de regresión presenta leve variación con el aumento de las épocas; se identifican dos mínimos locales del MAE en las épocas 10 y 100, respectivamente.

De forma similar, la variación de la función *Categorical Crossentropy* con el número de épocas es bastante estable. Alrededor de la época 25, la curva se vuelve asintótica. Se opta por 70 épocas como valor óptimo, logrando un equilibrio entre reducción de la pérdida y eficiencia

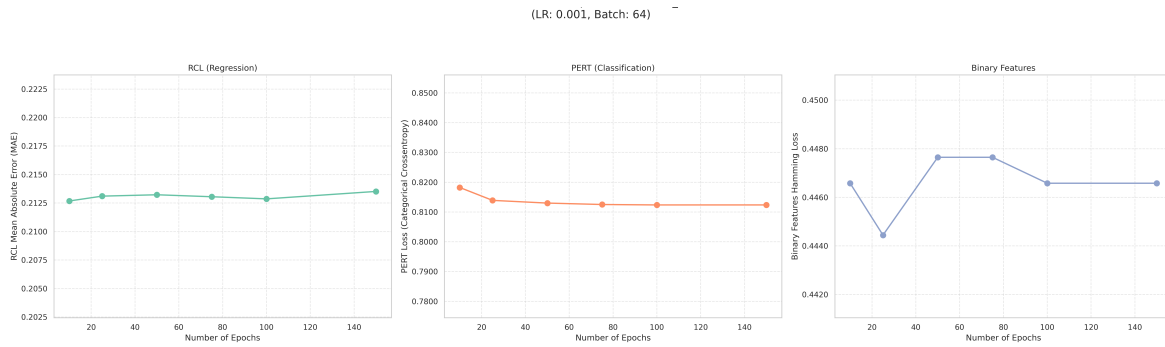


Figura 21: Impacto del número de épocas en las métricas de rendimiento de las redes neuronales (RCL: MAE, PERT: Accuracy, Binarias: Hamming Loss). Arquitectura fija: [128, 64], Activación ReLU, LR 0.001, lote 64.

computacional. El modelo que predice el orden relativo de los operadores, es decir, el modelo binario multietiqueta, presenta un comportamiento más complejo. La métrica se minimiza en la época 28. La curva evidencia alzas, mesetas y descensos, estabilizándose después de las 100 épocas.

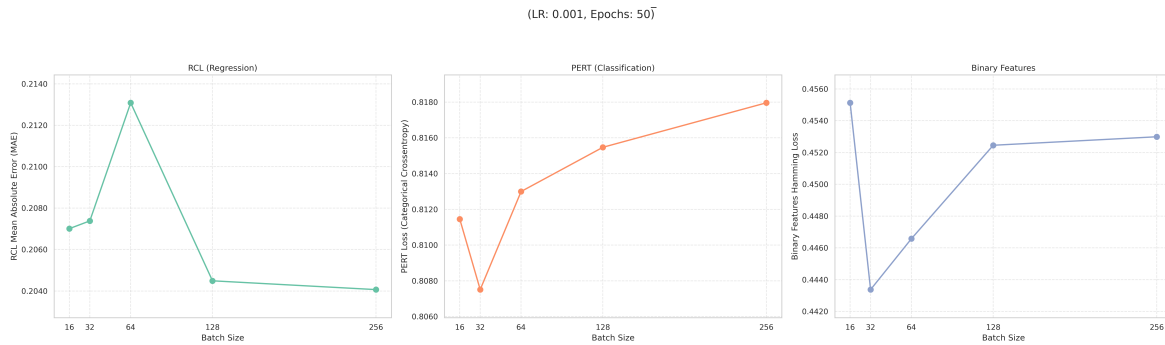


Figura 22: Impacto del tamaño del lote (*batch size*) en las métricas de rendimiento de las redes neuronales (RCL: MAE, PERT: Accuracy, Binarias: Hamming Loss). Arquitectura fija: [128, 64], Activación ReLU, LR 0.001, 50 épocas. El eje X se presenta en escala logarítmica.

En la Figura 22 se presenta el impacto del tamaño del lote sobre las métricas de rendimiento de cada modelo.

Para el caso del modelo de regresión que predice el parámetro RCL, se observa una mejoría con lotes más grandes, aunque la ganancia es marginal. Se opta por 128 como buen compromiso entre precisión y eficiencia.

En el caso del modelo de clasificación para el parámetro 'PERT', el tamaño de lote óptimo es 32, ya que luego tiende a mostrar una curva logarítmica creciente. De manera similar, pero con saltos más pronunciados, el modelo multietiqueta presenta su mejor rendimiento con un lote de 32. Luego, la métrica empeora hasta 128, perdiendo cerca de 10^{-2} unidades en la función de pérdida.

4.4.2. Configuraciones de Redes Neuronales (NN)

Se diseñaron arquitecturas de redes neuronales específicas para cada tarea predictiva. Las arquitecturas finales, resultantes del proceso de optimización y análisis experimental, fueron las siguientes. Todas ellas utilizaron el algoritmo optimizador Adam y la función de activación ReLU (*Rectified Linear Unit*, $f(x) = \max(0, x)$) en las capas ocultas:

- **Modelo RCL (Regresión):** Compuesto por dos capas ocultas con 64 y 32 neuronas, respectivamente. Se entrenó con una tasa de aprendizaje de 0.1, durante 10 épocas y con un tamaño de lote de 128. La capa de salida empleó una activación lineal.
- **Modelo PERT (Clasificación Multiclase):** Arquitectura de dos capas ocultas con 256 y 128 neuronas. Los parámetros de entrenamiento fueron una tasa de aprendizaje de 0.01, 25 épocas y tamaño de lote de 32. La activación *softmax* se usó en la capa de salida.
- **Modelo de Características Binarias (Clasificación Multietiqueta):** Configurado con una capa oculta de 256 neuronas. Se entrenó con una tasa de aprendizaje de 0.001, durante 28 épocas y con un tamaño de lote de 32. Cada una de las seis salidas utilizó una activación sigmoide.

Resultados de los Modelos de Redes Neuronales

Tras el entrenamiento y la optimización de hiperparámetros, los modelos finales de redes neuronales se evaluaron en el conjunto de prueba para medir su rendimiento. A continuación, se detallan las métricas obtenidas.

4.4.3. Modelo de Regresión para RCL

Las métricas de error para el modelo de regresión de RCL, como el *Root Mean Square Error* (RMSE) y el *Mean Absolute Error* (MAE) (Tabla 8), sugieren una magnitud promedio de error que podría considerarse moderada en el contexto del rango de valores de RCL. Es decir, el modelo presenta un error del orden de un cuarto a un tercio del dominio de los valores posibles de RCL, lo cual se considera un resultado moderado.

No obstante, la peor métrica es el coeficiente de determinación R^2 . Un valor de R^2 negativo, como el observado, es un indicio contundente de que el modelo no solo no explica la variabilidad de la variable RCL, sino que su desempeño es inferior al de un modelo basal que simplemente predijera la media histórica de RCL. Esto implica que el modelo actual no logra capturar la tendencia de los datos.

Modelo Target	Métrica	Valor
RCL	MSE	0.0655
	RMSE	0.2560
	MAE	0.2189
	R^2	-0.1911
PERT	Accuracy	0.6571
	F1-score (Macro)	0.4577
	F1-score (Weighted)	0.6181
	Precision (Macro)	0.4391
	Precision (Weighted)	0.5917
	Recall (Macro)	0.4868
	Recall (Weighted)	0.6571
Binary Features	Subset Accuracy	0.0609
	Hamming Loss	0.4460
	MAE (Probabilities)	0.4853
	Jaccard Score (Macro)	0.4310
	Jaccard Score (Weighted)	0.4348
	F1-score (Macro)	0.5987
	F1-score (Weighted)	0.6026
	Precision (Macro)	0.5504
	Precision (Weighted)	0.5520
	Recall (Macro)	0.6707
	Recall (Weighted)	0.6771

Tabla 8: Métricas de rendimiento de los modelos de redes neuronales finales en el conjunto de prueba, usando hiperparámetros personalizados.

Las posibles causas de este rendimiento deficiente incluyen una representación insuficiente de características predictivas (hipótesis respaldada por las bajas correlaciones observadas en análisis exploratorios previos), una posible escasez de datos para modelar relaciones complejas inherentes a RCL, o una relevancia intrínseca limitada de las características de entrada disponibles para la predicción de esta variable particular en el contexto del algoritmo meta-heurístico.

En consecuencia, y dada la ineficacia evidenciada por el R^2 negativo, este modelo no se considera adecuado para la tarea propuesta. Es relevante notar que, en comparaciones previas, un regresor XGBoost ($R^2 \approx 0,0083$), aunque también con un poder explicativo débil, demostró un rendimiento marginalmente superior con un consumo computacional menor.

4.4.4. Modelo de Clasificación para PERT

El modelo de clasificación para PERT alcanzó una exactitud (*accuracy*) general (Tabla 8) que indica un porcentaje considerable de clasificaciones correctas. Sin embargo, un análisis más profundo a través de las puntuaciones F1 revela matices importantes. El F1-score macro, que otorga igual peso a cada clase independientemente de su tamaño, sugiere un rendimiento global deficiente.

Clase	Precisión	Recall	F1-score	Soporte
1	0.7738	0.6842	0.7263	95
2	0.2500	0.3500	0.2917	40
3	0.0000	0.0000	0.0000	39
4	0.7326	0.9130	0.8129	138

Tabla 9: Métricas de rendimiento por clase para el modelo PERT, con hiperparámetros personalizados.

En contraste, el F1-score ponderado, que considera el número de instancias por clase, muestra una mejora, apuntando a un rendimiento promedio que supera una predicción aleatoria. Esta disparidad entre las puntuaciones F1 macro y ponderada es un fuerte indicio de un **desequilibrio de clases**, donde el modelo probablemente se desempeña mejor en las clases más frecuentes, un comportamiento también observado en análisis con otros algoritmos como XGBoost. Las métricas de precisión y *recall*, tanto en sus versiones macro como ponderada (Tabla 8), confirman esta interpretación.

La precisión ponderada, superior a la macro, indica que el modelo es más preciso al predecir instancias de las clases más comunes. De forma similar, una exhaustividad ponderada significativamente más alta que la macro revela que el modelo identifica un mayor porcentaje de las instancias positivas reales en dichas clases mayoritarias, mientras que su capacidad para recuperar instancias de clases minoritarias es limitada.

El desglose detallado del rendimiento por clase (presentado en la Tabla 9) es crucial: se observa un rendimiento aceptable para la Clase 1 y un rendimiento notablemente bueno para la Clase 4 (especialmente en exhaustividad), que son las clases con mayor soporte. Por el contrario, la Clase 2 muestra un rendimiento pobre y el modelo falla completamente en identificar instancias de la Clase 3.

Esta incapacidad para generalizar en las clases menos representadas es la principal causa del bajo F1-score macro y subraya el impacto negativo del desequilibrio de clases en el rendimiento global del clasificador PERT.

4.4.5. Modelo de Características Binarias para la Predicción de Secuencia de Operadores

El modelo de clasificación multietiqueta para las características binarias obtuvo una exactitud de subconjunto exacto 8) bajo. Esto indica que la predicción simultáneamente correcta de las seis etiquetas binarias para una instancia dada es infrecuente, lo cual es común en tareas multietiqueta debido a la complejidad combinatoria y la estricta naturaleza de esta métrica.

La pérdida de Hamming (*Hamming Loss*) (Tabla 8) refleja la fracción promedio de etiquetas incorrectamente predichas por instancia. El valor observado sugiere que una proporción considerable de etiquetas individuales son erróneas, indicando un rendimiento que apenas supera lo esperable por azar para cada etiqueta aislada. Este desempeño modesto puede atribuirse, en parte, a la débil correlación entre las características binarias y las variables de entrada. Adicionalmente, el Error Absoluto Medio (MAE) sobre las probabilidades predichas (Tabla 8) es relativamente alto, implicando que las salidas continuas del modelo (previas a la binarización) se encuentran, en promedio, alejadas de los valores objetivo (0 o 1). Esto puede llevar a umbrales de decisión subóptimos y errores en la clasificación final, posiblemente debido a una cantidad insuficiente de datos para un aprendizaje robusto de patrones complejos o a una arquitectura de modelo que no captura adecuadamente las interdependencias entre etiquetas.

Las puntuaciones F1 macro y ponderada (Tabla 8) son cercanas entre sí y apuntan a un rendimiento moderado a nivel de etiqueta individual. Su similitud sugiere que el rendimiento es relativamente homogéneo entre las diferentes etiquetas binarias o que el desequilibrio por etiqueta no afecta drásticamente el promedio ponderado en comparación con el macro. El *Jaccard Score* (Tabla 8), tanto en su versión macro como ponderada, indica un solapamiento moderado-bajo entre los conjuntos de etiquetas verdaderas y predichas, reforzando la idea de que, aunque algunas etiquetas puedan ser correctamente predichas, el acuerdo global por instancia es limitado.

Característica Binaria (Operador)	F1-Score	Precision	Recall
2opt-Reorder	0.6089	0.5892	0.6301
2opt-Exchange	0.5755	0.5316	0.6273
2opt-OR_opt	0.6967	0.5880	0.8547
Reorder-Exchange	0.4622	0.5225	0.4143
Reorder-OR_opt	0.6523	0.5602	0.7806
Exchange-OR_opt	0.5969	0.5112	0.7170

Tabla 10: Métricas de rendimiento por característica para el modelo *Binary Features*, con hiperparámetros personalizados.

Al analizar la precisión y la exhaustividad (*recall*) agregadas (Tabla 8), se observa que la exhaustividad es notablemente superior a la precisión. Esto sugiere que el modelo tiende a ser más inclusivo al predecir etiquetas positivas (es decir, es más "sensible" para detectar

positivos), lo que, si bien ayuda a capturar una mayor proporción de los verdaderos positivos, también incrementa la tasa de falsos positivos.

El rendimiento específico por cada característica binaria (detallado en la Tabla 10) es heterogéneo. Algunas relaciones secuenciales como *2opt-OR_opt* son predichas con un F1-score relativamente mejor, mientras que *Reorder-Exchange* presenta un rendimiento considerablemente más bajo. Esta variabilidad subraya que la capacidad del modelo para inferir estas relaciones depende de la relación específica, posiblemente debido a diferencias en su frecuencia o en la informatividad de las características de entrada para cada caso.

4.5. Resultados y análisis comparativo en instancias no vistas

A continuación, se ejecutan nueve instancias no vistas, estas son las instancias [c11, c12, c13, c14, c16, c18, c22, a62, a80] las cuales varían desde las 17 granjas hasta las 79 y desde un camión hasta cinco. Se encuentran disponibles en [Montero, 2021]. Esto asegura un conjunto de instancias de validación diverso, lo cual apunta a medir la capacidad de generalización de los modelos empíricamente. Se provee un análisis de las causas y consecuencias de estos resultados.

Modelo	Instancia	RCL	PERT	IDSEQ
XGB	c11	0.585260	4	2
XGB	c12	0.596275	3	2
XGB	c13	0.594210	4	2
XGB	c14	0.580117	1	2
XGB	c16	0.594210	2	2
XGB	c18	0.580117	1	2
XGB	c22	0.613733	1	2
XGB	a62	0.599892	4	2
XGB	a80	0.580117	4	2
Feed Forward	c11	0.668439	2	5
Feed Forward	c12	0.668439	4	19
Feed Forward	c13	0.650567	3	19
Feed Forward	c14	0.668439	1	5
Feed Forward	c16	0.668439	1	1
Feed Forward	c18	0.650567	1	5
Feed Forward	c22	0.668439	2	22
Feed Forward	a62	0.691567	2	8
Feed Forward	a80	0.680567	2	14

Tabla 11: Predicciones de modelos XGB y Feed Forward

En pos de mediciones más robustas, se ejecuta cada instancia con la metaheurística ILS 10 veces con distintas semillas. Dado que los modelos son independientes, también se optó

por evaluar instancias con los parámetros predichos por los mejores modelos de cada tipo en base a los resultados de la sección anterior, es decir, **RCL**, **PERT** proveniente de XGB y **IDSEQ** desde el modelo de red neuronal.

Los parámetros predichos por los modelos entrenados previamente se presentan en la tabla 11. Estos levantan suspicacia, ya que las instancias cuentan con variaciones importantes en el número de camiones y granjas. El parámetro RCL es predicho para ambas instancias con, a lo más, una dispersión de 0.1 unidades, lo cual tiene sentido ya que son relativamente cercanas. El parámetro PERT muestra un comportamiento variado, lo cual se sustenta en lo distintas que son las instancias; además, coinciden solo en los casos para aplicar un operador.

El modelo XGB tiene una tendencia a la clase 4, seguido de la clase 1, respectivamente. A su vez, el modelo de *Feed Forward* presenta una tendencia hacia la clase 2, seguido de la 1. IDSEQ, predicho por los modelos de XGB, presenta un sobreajuste hacia la clase 2 y muestra nula capacidad de generalización ante distintas instancias. En cambio, el modelo de redes neuronales presenta una adaptabilidad mucho mayor a las diferencias de las instancias, lo cual tiene más sentido.

Las tablas 12, 13 y 14 están basadas en la Tabla 5 de [Montero *et al.*, 2022], en la que se presenta la comparación del algoritmo *Branch & Cut* respecto al ILS en cuestión.

Las columnas presentadas como F y K son, respectivamente, el número de granjas y camiones. El valor T representa el tiempo total de ejecución del algoritmo de referencia *Branch & Cut*. IG % se refiere al *Integrality Gap*, que es una medida de la holgura o razón entre la solución entera óptima y la solución encontrada. Un valor menor indica menos holgura para encontrar mejores soluciones en el espacio de búsqueda. Z es el valor de la función objetivo correspondiente.

Grupo	#F	#K	T	IG(%)	Z	Mejor Z (%)	Peor Z (%)	Prom. Z (%)	T.T. (%)
11	17	1	0	0,00	128,40	-0,02	-0,02	-0,02	+92,630
12	29	3	14.400	0,22	648,20	0,00	-0,23	-0,06	-90,98
13	17	3	52	1,63	529,00	-0,01	-0,01	-0,01	+1688
14	28	4	14.400	0,88	1298,30	0,00	-0,55	-0,17	-93,16
16	23	2	40	0,00	361,40	0,00	0,00	0,00	+2250
18	29	5	14.400	8,15	1250,30	4,45	4,45	4,45	-93,15
22	19	4	1213	0,00	858,10	0,00	0,00	0,00	-22,87
a62	61	3	14.400	1,87	27.216	1,14	0,71	0,96	-75,06
a80	79	3	14.400	1,28	37.346	0,39	0,00	0,19	-60,02
Promedio	-	-	-	-	-	0,66	0,49	0,60	-

Tabla 12: ILS XGB (10 ejecuciones).

La tabla 12 denota, como resultado general, una mejora en las soluciones obtenidas debido a la predicción de parámetros. Notar que las instancias con IG % igual a 0.0, en teoría, no tienen espacio para mejorar soluciones, por lo que no debería haber valores mayores a 0.0 % de mejora en estos casos. Los incrementos se aprecian para las instancias 18, a62 y a80, las

cuales cuentan con IG mayores a cero. Además, se nota una capacidad de generalización decente, ya que el modelo fue entrenado con instancias pequeñas y aun así obtiene resultados positivos en instancias de mayor tamaño. Los peores resultados varían en torno a -0.55 %, lo cual es una pérdida menor en relación con la oportunidad de diversificar y los ahorros considerables en tiempo de ejecución, que varían en torno a [20 %-93 %], en instancias donde el algoritmo Branch & Cut no presenta valores atípicos.

Grupo	#F	#K	T	IG(%)	Z	Mejor Z (%)	Peor Z (%)	Prom. Z (%)	T.T. (%)
11	17	1	0	0,00	128,40	-0,02	-0,02	-0,02	+92,555
12	29	3	14.400	0,22	648,20	0,00	-0,23	-0,05	-91,34
13	17	3	52	1,63	529,00	-0,01	-0,01	-0,01	+1679
14	28	4	14.400	0,88	1298,30	0,00	-0,55	-0,13	-93,15
16	23	2	40	0,00	361,40	0,00	0,00	0,00	-2212
18	29	5	14.400	8,15	1250,30	4,45	4,45	4,45	-92,81
22	19	4	1213	0,00	858,10	0,00	0,00	0,00	-23,90
a62	61	3	14.400	1,87	27.216	1,26	0,91	1,06	-79,89
a80	79	3	14.400	1,28	37.346	0,48	0,12	0,28	-64,84
Promedio	-	-	-	-	-	0,68	0,52	0,62	-

Tabla 13: ILS Feed Forward (10 ejecuciones).

La tabla 13 permite confirmar los resultados obtenidos por las predicciones del modelo ILS, aunque presenta mejoras marginales respecto al modelo anterior, particularmente en instancias grandes. A su vez, confirma el ahorro considerable en tiempos de ejecución logrado por el algoritmo metaheurístico frente a sus pares. Se infiere que las mejoras marginales de estas ejecuciones se deben a una mejor adaptabilidad del modelo para predecir la permutación de los operadores. Este modelo, en promedio, presenta los mejores peores resultados, los cuales provienen de la menor pérdida en la calidad de las soluciones.

Grupo	#F	#K	T	IG(%)	Z	Mejor Z (%)	Peor Z (%)	Prom. Z (%)	T.T. (%)
11	17	1	0	0,00	128,40	-0,02	-0,02	-0,02	+92,689
12	29	3	14.400	0,22	648,20	0,00	-0,23	-0,05	-91,23
13	17	3	52	1,63	529,00	-0,01	-0,01	-0,01	+1698
14	28	4	14.400	0,88	1298,30	0,00	-0,44	-0,05	-86,27
16	23	2	40	0,00	361,40	0,00	0,00	0,00	-2228
18	29	5	14.400	8,15	1250,30	4,45	4,45	4,45	-93,20
22	19	4	1213	0,00	858,10	0,00	0,00	0,00	-23,87
a62	61	3	14.400	1,87	27.216	1,33	0,67	0,95	-75,93
a80	79	3	14.400	1,28	37.346	0,39	0,09	0,27	-59,83
Promedio	-	-	-	-	-	0,68	0,50	0,62	-

Tabla 14: ILS Merge (10 ejecuciones).

La tabla 14 presenta los resultados de mezclar las predicciones de RCL y PERT desde XGB, y de IDSEQ desde *Feed Forward*. La hipótesis para esto era que este modelo debía obtener

los mejores resultados, dado que contiene las mejores componentes en base a las métricas de rendimiento presentadas en las subsecciones 4.3.3 y 4.4.2. Esta combinación presenta resultados mejores que la ejecución de parámetros predichos por XGB puro, pero peores que los predichos por *Feed Forward*. Particularmente en la columna de Peor Z (%), aunque es una pérdida marginal de 0.02 unidades. Aunque cabe destacar mejores resultados en la instancia a62 para la columna Mejor Z (%), lo cual puede ser un indicio de que la combinación de parámetros predichos por distintos modelos puede resultar fructífera en instancias particulares. De todas formas, la hipótesis es rechazada.

Los resultados generales de las figuras 12, 13 y 14 son similares, lo cual es un punto a favor de la fiabilidad de estos. Notamos ganancias en las soluciones para 3/4 de las instancias con un IG % >1. También es importante tener presente que los resultados promedios están sesgados por los resultados obtenidos en la instancia 18, pero tienen sentido ya que presenta el mayor IG %.

A pesar de ser ganancias marginales en su mayoría, estas representan evidencia empírica de que los modelos de aprendizaje automático pueden predecir parámetros de algoritmos metaheurísticos que resuelven problemas de optimización combinatoria en pos de obtener soluciones de mayor calidad. El conjunto de resultados demuestra una capacidad de generalización decente a pesar de los resultados mostrados en la sección previa. Esto se debe a mejoras en los resultados en instancias con el doble o triple de granjas que las vistas en las instancias de entrenamiento.

Método	Mejor ganancia	Peor ganancia	Ganancia promedio	Tiempo total [s]
ILS (10 ejecuciones)	17,395,1	16,801,8	17,123,5	36,239
ILS (20 ejecuciones)	17,395,1	16,801,8	17,105,6	72,437
B&C	16,368,7	-	-	131,211

Tabla 15: Resultados para la instancia completa, ILS y B&C con la ganancia y tiempo total de los 25 clústeres añadidos. Extraída de [Villagrán et al., 2020]

La tabla 15 muestra el impacto del algoritmo metaheurístico ILS en instancias grandes o de producción respecto al algoritmo *Branch & Cut*. En base a los resultados presentados en 15, se aprecia una mejora sustancial respecto al algoritmo B&C, lo que implica que existe una mayor brecha en el desempeño en instancias grandes o de producción. Bajo el supuesto probable de que en instancias de gran tamaño existe una mayor holgura o potencial de optimización, esto se debe a espacios de búsqueda más grandes. Esto nos permite inferir que el potencial de los modelos de aprendizaje para la predicción de parámetros en problemas de optimización combinatoria es mayor, ya que se vuelve intratable la optimización manual o a través de sintonizadores para estas instancias.

4.6. Resumen

En el presente capítulo se abordó brevemente la metaheurística y la configuración del ambiente de ejecución. Luego se presentó un análisis simple de hiperparámetros tanto para XGB como para Feed Forward, lo que guió la elección de los hiperparámetros para el entrenamiento y *testing*. A continuación, se presentan los resultados de cada modelo en métricas generales, funciones de pérdida, entre otros. Se presenta también un análisis de causas y consecuencias. Por último, se realiza un análisis comparativo en contraste con el artículo de referencia propulsor del algoritmo metaheurístico base ILS, con el cual se obtiene evidencia para validar la hipótesis principal de los experimentos.

CAPÍTULO 5

CONCLUSIONES

La presente memoria aborda el desafío de la sintonización automática de parámetros para metaheurísticas, aplicado particularmente al Problema de Ruteo de Vehículos Multi-Producto con Mezcla Gradual (VRPMPWB). Este problema, inherente a la logística de recolección de leche, presenta una complejidad computacional significativa, siendo de naturaleza NP-hard. La eficacia de las metaheurísticas en su resolución depende fundamentalmente de una configuración adecuada de sus parámetros, un proceso que tradicionalmente ha sido costoso y manual.

La hipótesis central de este trabajo fue que los modelos de aprendizaje automático podían predecir configuraciones de parámetros, ofreciendo una eficacia similar a los métodos de sintonización tradicionales, pero con una notable reducción en el tiempo de configuración y sin comprometer la calidad de las soluciones.

Aunque cabe destacar que el costo al usar este enfoque de modelo de ML es trasladado al entrenamiento, validación y monitoreo de los modelos. La principal ventaja de este enfoque es su potencial de generalización y capacidad para abordar instancias de producción, lo que se vuelve intratable con métodos tradicionales. Los resultados de esta investigación, aunque con desafíos importantes, arrojan luz sobre el potencial y las limitaciones de este enfoque prometedor.

La generación de datos para el entrenamiento de los modelos se llevó a cabo de manera sistemática utilizando el sintonizador Irace, que, a través de carreras iteradas y pruebas estadísticas, identificó configuraciones de parámetros de alto rendimiento. Este proceso fue vital para asegurar una base de datos rica y representativa, capturando la variabilidad inherente al comportamiento de la metaheurística ILS.

El diseño de los modelos de aprendizaje automático adoptó una arquitectura dual y especializada: se optó por XGBoost y Redes Neuronales Feed-Forward, cada uno optimizado para el tipo específico de parámetro a predecir. Notar que esta elección permite una comparación entre modelos de aprendizaje automático clásico de la familia de ensambles y algo más contemporáneo como las redes neuronales. Para RCL, se utilizó un modelo de regresión; para PERT, un clasificador multiclase; y para las características binarias de IDSEQ, un clasificador multietiqueta. Esta modularidad no solo facilitó la gestión y depuración, sino que también permitió una evaluación comparativa directa del desempeño de ambos enfoques.

Un pilar fundamental de este estudio fue la exhaustiva y robusta ingeniería de atributos. Se diseñaron y extrajeron características espaciales, de agrupamiento y de mezclado, proporcionando una visión holística de la complejidad intrínseca de cada instancia del VRPMPWB. Estas características no solo cuantifican la geometría del problema, sino que también revelan estructuras internas y propiedades relacionadas con la calidad y el volumen de la leche,

cruciales para las decisiones de ruteo y mezclado.

La decisión de desagregar el parámetro IDSEQ en pares de operadores binarios fue metodológicamente sólida, favoreciendo la interpretabilidad de los operadores y su impacto individual. El dataset obtenido a partir de un riguroso proceso de selección de características (basado en importancia en modelos de árbol, umbral de varianza y correlación), permitió reducir los niveles de dimensionalidad, contribuyendo a mitigar el ruido y la redundancia de la información.

5.1. Experimentos y Resultados

A pesar de la base metodológica, los resultados experimentales presentaron un panorama matizado. Para la predicción de RCL, el modelo de regresión mostró un coeficiente de determinación R^2 extremadamente bajo (0.0083 para XGBoost y -0.1911 para Redes Neuronales).

Esto indica que los modelos apenas explican la variabilidad en RCL y, en el caso de las Redes Neuronales, su desempeño fue incluso inferior a una predicción basada en la media histórica. Esta ineficacia sugiere que las características de entrada disponibles no son lo suficientemente informativas para predecir RCL, que este parámetro tiene una relevancia menor en el rendimiento de la metaheurística ILS para las instancias estudiadas, o que la cantidad de datos de entrenamiento fue insuficiente para capturar su comportamiento complejo. La baja correlación previamente observada entre RCL y otras características refuerza esta hipótesis.

Por otra parte, es bueno recalcar los resultados MAE de 0.2025 para XGBoost y 0.2189 para Feed Forward, donde se aprecia una dispersión de 0.2 unidades respecto al óptimo, lo cual no es tan malo, aunque hay margen de mejora. La predicción en instancias no vistas demuestra cierto consenso entre ambos tipos de modelos, con una diferencia de a lo más 0.1 unidades, lo cual aporta confiabilidad a esa predicción.

En cuanto a la predicción de PERT, ambos modelos (XGBoost y Redes Neuronales) mostraron un rendimiento moderado, con una exactitud general aceptable (alrededor del 65 %). Sin embargo, el análisis del F1-score reveló un desequilibrio de clases, donde el rendimiento es sólido para las clases mayoritarias (Clase 1 y 4), lo cual puede deberse a que Irace tiende a seleccionar valores de los bordes del rango para estos parámetros. Esto es un reflejo de la distribución de las clases en el conjunto de entrenamiento y sugiere que, aunque el modelo puede predecir correctamente una parte significativa de los casos, su capacidad de generalización se ve comprometida por la disparidad en la representación de clases.

Para la predicción de instancias no vistas, notamos consenso entre ambos tipos de modelos; en promedio presentan 1 unidad de diferencia. Además, se aprecian ciertos sesgos en ambos modelos: el modelo XGBoost presenta sesgos hacia las clases 4 y 1, como confirman las métricas de rendimiento bajo el dataset de evaluación. En cambio, el modelo de redes neuronales tiene una preferencia hacia las clases 2 y 1.

Para la predicción de las características binarias de IDSEQ (que representan la activación de pares de operadores), la exactitud de subconjunto fue muy baja (0.0337 para XGBoost y 0.0609 para Redes Neuronales), evidenciando la dificultad de predecir correctamente la combinación completa de las seis etiquetas binarias simultáneamente. La (Hamming Loss) de 0.4766 para XGBoost y 0.4460 para Redes Neuronales, y el F1-score ponderado (0.5819 para XGBoost y 0.6026 para Redes Neuronales), reflejan un rendimiento moderado a nivel de predicción individual de los operadores.

La predicción en las instancias no vistas para las características binarias presenta un desempeño pobre en el tipo de modelos de XGBoost, ya que se nota un overfit claro hacia la clase 2. Dicho modelo presenta altas dificultades para obtener comportamientos balanceados en la predicción de características binarias y puede deberse en parte a la decisión metodológica de separar esta variable en pares de operadores relativos.

Para el tipo de modelos de Redes Neuronales, la variedad de predicciones es mucho más rica que su par, lo que es sustentado por mejores métricas. Debido a la arquitectura de las redes neuronales, que comprenden relaciones entre variables más complejas y/o factores latentes, otra causa posible de esto es el menor uso de datos de entrenamiento para las redes neuronales (15 % menos que los modelos XGBoost), lo que favorece la generalización de los modelos.

Los modelos lograron un Recall alto, lo que significa que fueron efectivos en la detección de casos positivos. No obstante, la precisión fue considerablemente más baja, resultando en una alta tasa de falsos positivos. Un punto crítico fue la falla total en predecir la característica "Reorder-Exchange" por parte del modelo XGBoost.

Un hallazgo novedoso y relevante fue el análisis de importancia de características para la activación de los pares de operadores, revelando que predictores espaciales, de agrupamiento y de producción influyen en la activación de operadores específicos, lo cual no ha sido explorado en trabajos previos.

Los resultados bajo instancias no vistas presentan resultados marginalmente positivos. Estas mejoras, a pesar de ser en torno al 1 %, implican dos cosas: primero, que los modelos de ML sí pueden predecir parámetros metaheurísticos en problemas de optimización combinatoria, particularmente VRPs, validando así la hipótesis principal del experimento.

Segundo, a pesar de que estas mejoras marginales son obtenidas a través de modelos que presentan F1-scores bajos o moderados, nos permite inferir que con modelos de mayor calidad se obtendrán mejores parámetros y, por ende, mejores soluciones.

A su vez, esto implica que la sintonización de parámetros se vuelve más crítica y potencialmente más beneficiosa a medida que aumenta la complejidad de la instancia, especialmente en escenarios de producción, donde la capacidad de generalización de los modelos sale a relucir.

A pesar de que las métricas de validación de los modelos predictivos fueron moderadas, su aplicación para configurar la metaheurística en instancias no vistas resultó en mejoras marginales pero consistentes en la calidad de la solución. Esto sugiere que, aunque los modelos capturan una señal débil de la relación entre las características de la instancia y los parámetros óptimos, esta señal es suficiente para guiar al algoritmo hacia regiones del espacio de búsqueda de mayor calidad, superando a configuraciones menos informadas

5.2. Limitaciones y Direcciones Futuras

Mirando hacia el futuro, las recomendaciones para investigaciones futuras son claras y cruciales para avanzar en este campo. Es imperativo incrementar el volumen de datos de entrenamiento en "órdenes de magnitud mayores". Los modelos de aprendizaje automático, especialmente las redes neuronales, requieren grandes volúmenes de datos para aprender relaciones complejas y generalizar de manera fiable.

Se sugiere orientar la generación de datos y los modelos a instancias de mayor tamaño, similares a las que se encontrarían en entornos de producción. Estamos optimistas de que con un mayor volumen de datos y un marco robusto, podremos alcanzar F1-scores cercanos al 80 %. Estas ganancias, incluso si son marginales del 1 %, pueden representar miles de dólares en diferencias operacionales a lo largo de un periodo de tiempo para industrias importantes como la química, de combustibles y láctea.

Se recomienda el uso de técnicas de *Resample* o Bootstrap para aumentar la diversidad y cantidad de datos disponibles a menor costo. En esta materia, sería ideal contar con una fuente diversa de datos mediante el uso de dos o más sintonizadores para evitar sesgos en los datos provenientes de una sola fuente.

Otra recomendación es la aplicación de regularización en las funciones de pérdida, que se vuelve fundamental para prevenir el sobreajuste; en este ámbito se recomienda el uso de 'elasticnet' para mayor robustez. En una línea similar, se promueve intentar con GeLU (Gaussian Error Linear Unit) como función de activación, ya que cuenta con características matemáticas más robustas que su predecesora.

Asimismo, se recomienda la utilización de herramientas de aprendizaje automático más avanzadas como PyTorch y la implementación de sistemas de monitoreo como TensorBoard para una optimización y seguimiento más detallado del entrenamiento de los modelos.

Se aconseja el uso de técnicas de reducción de dimensionalidad como Sparse PCA (Análisis de Componentes Principales Disperso) o MDS (Escalamiento Multidimensional) en el caso de entrenar modelos con datasets con una dimensionalidad considerable. En esta línea, sería interesante medir la diferencia de modelos entrenados con dimensiones reducidas frente a otros que no, en el marco de la predicción de parámetros metaheurísticos.

En perspectivas generales, se recomienda indagar más en modelos de redes neuronales y sus distintas variantes, ya que estos presentaron los mejores resultados en instancias no vistas, así como una exploración y análisis de hiperparámetros más profundo para optimizar los modelos.

Es necesario reevaluar la sensibilidad y el rol de parámetros como RCL en el rendimiento de la metaheurística ILS y si las características de instancia actuales son suficientes para predecirlos con precisión. Dada la limitación de datos, una variante a explorar es el uso de modelos preentrenados con técnicas como *embedding*. Finalmente, es importante considerar el *tradeoff* entre el costo de entrenar, validar y monitorear modelos de producción, y las posibles ganancias en optimización que pueden generar a largo plazo.

En síntesis, este trabajo ha establecido un marco sólido para la sintonización de metaheurísticas mediante aprendizaje automático en problemas de ruteo con mezcla, demostrando la importancia de ciertas características en la activación de operadores. Aunque los resultados actuales indican un desempeño moderado en el conjunto de validación y mejoras marginales en instancias no vistas, aun así, el presente trabajo valida el uso de técnicas de aprendizaje automático para la predicción de parámetros en problemas de optimización combinatoria.

Los aspectos clave para investigaciones futuras son la expansión y mejora de los *datasets* de entrenamiento, la aplicación de ingeniería de atributos, junto con la exploración de técnicas de aprendizaje automático más sofisticadas.

ANEXOS

En los Anexos se incluye todo aquel material complementario que no es parte del contenido de los capítulos de la memoria, pero que permiten a un lector contar con un contenido adjunto relacionado con el tema.

REFERENCIAS BIBLIOGRÁFICAS

- [Ankerst *et al.*, 1999] Ankerst, M., Breunig, M., Kröger, P., y Sander, J. (1999). Optics: Ordering points to identify the clustering structure. volumen 28, pp. 49–60.
- [Ansótegui *et al.*, 2015] Ansótegui, C., Malitsky, Y., Samulowitz, H., Sellmann, M., y Tierney, K. (2015). Model-based genetic algorithms for algorithm configuration. En *IJCAI*, pp. 733–739.
- [Avella *et al.*, 2004a] Avella, P., Boccia, M., y Sforza, A. (2004a). Optimization of the transportation of gasoline in venezuela. *Transportation Science*, 38(4):490–500.
- [Avella *et al.*, 2004b] Avella, P., Boccia, M., y Sforza, A. (2004b). Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, 152(1):170–179.
- [Bachnas *et al.*, 2014] Bachnas, A., Tóth, R., Ludlage, J., y Mesbah, A. (2014). A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study. *Journal of Process Control*, 24(4):272–285.
- [Barros Everett, 2024] Barros Everett, T. W. (2024). Analysis of routing problem instances for metaheuristic parameter prediction using machine learning. Tesis de máster, Universidad Técnica Federico Santa María, Santiago, Chile. Memoria de Titulación para optar al título de Ingeniero Civil Informático.
- [Bello *et al.*, 2017] Bello, I., Pham, H., Le, Q. V., Norouzi, M., y Bengio, S. (2017). Neural combinatorial optimization with reinforcement learning.
- [Campello *et al.*, 2013] Campello, R. J. G. B., Moulavi, D., y Sander, J. (2013). Density-based clustering based on hierarchical density estimates. En Pei, J., Tseng, V. S., Cao, L., Motoda, H., y Xu, G., editores, *Advances in Knowledge Discovery and Data Mining*, pp. 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Caramia y Guerriero, 2010] Caramia, M. y Guerriero, F. (2010). A milk collection problem with incompatibility constraints. *Interfaces*, 40(2):130–143.
- [Chen y Guestrin, 2016] Chen, T. y Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754.
- [Chokanat *et al.*, 2019] Chokanat, P., Pitakaso, R., y Sethanan, K. (2019). Methodology to solve a special case of the vehicle routing problem: A case study in the raw milk transportation system. *AgriEngineering*, 1(1):75–93.
- [Claassen y Hendriks, 2007] Claassen, G. y Hendriks, T. (2007). An application of special ordered sets to a periodic milk collection problem. *European Journal of Operational Research*, 180(2):754–769.

- [Croes, 1958] Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812.
- [Davies y Bouldin, 1979] Davies, D. L. y Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.
- [Dayarian et al., 2015] Dayarian, I., Crainic, T. G., Gendreau, M., y Rei, W. (2015). A column generation approach for a multi-attribute vehicle routing problem. *European Journal of Operational Research*, 241:888–906.
- [Eiben et al., 1999] Eiben, A., Hinterding, R., y Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Trans. Evolutionary Computation*, 3:124–141.
- [Erdem, 2022] Erdem, M. (2022). Optimisation of the electric truck route for milk collection problem: a real case study. *Transportation Letters*, 15:1–18.
- [Fagerholt, 2000] Fagerholt, K. (2000). Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research*, 131(3):559–571.
- [Halkidi et al., 2001] Halkidi, M., Batistakis, Y., y Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145.
- [Huang et al., 2019] Huang, K., Wu, K.-F., y Ardiansyah, M. N. (2019). A stochastic dairy transportation problem considering collection and delivery phases. *Transportation Research Part E: Logistics and Transportation Review*, 129:325–338.
- [Hutter et al., 2011] Hutter, F., Hoos, H. H., y Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. En Coello, C. A. C., editor, *Learning and Intelligent Optimization*, pp. 507–523, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Hutter et al., 2009] Hutter, F., Hoos, H. H., Leyton-Brown, K., y Stützle, T. (2009). Paramils: an automatic algorithm configuration framework. En *Journal of Artificial Intelligence Research*, volumen 36, pp. 267–306.
- [Hutter et al., 2014] Hutter, F., Xu, L., Hoos, H. H., y Leyton-Brown, K. (2014). Algorithm runtime prediction: Methods & evaluation. En *Twenty-Eighth AAAI Conference on Artificial Intelligence*.
- [Kadioglu et al., 2010] Kadioglu, S., Malitsky, Y., Sellmann, M., y Tierney, K. (2010). Isac – instance-specific algorithm configuration. En *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, p. 751–756, NLD. IOS Press.
- [Kennedy y Eberhart, 1995] Kennedy, J. y Eberhart, R. (1995). Particle swarm optimization. En *1995 IEEE International Conference on Neural Networks Proceedings, Proceedings of International Conference on Neural Networks*, pp. 1942–1948, Perth. IEEE Australia Council.
- [Kerschke et al., 2018a] Kerschke, P., Hoos, H. H., Neumann, F., y Trautmann, H. (2018a). Automated algorithm selection: Survey and perspectives.

- [Kerschke *et al.*, 2018b] Kerschke, P., Preuss, M., Wessing, S., y Trautmann, H. (2018b). A comprehensive set of instance features for the vehicle routing problem with time windows. En *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18*, pp. 869–876. Association for Computing Machinery (ACM).
- [Kotthoff *et al.*, 2015] Kotthoff, L., Kerschke, P., Hoos, H., y Trautmann, H. (2015). Improving the state of the art in inexact tsp solving using per-instance algorithm selection. pp. 202–217.
- [Kriegel *et al.*, 2011] Kriegel, H.-P., Kröger, P., Sander, J., y Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240.
- [Lahrichi *et al.*, 2013] Lahrichi, N., Crainic, T. G., Gendreau, M., Rei, W., y Rousseau, L.-M. (2013). Strategic analysis of the dairy transportation problem. *Journal of the Operational Research Society*, 66.
- [Laporte, 2009] Laporte, G. (2009). Fifty Years of Vehicle Routing. *Transportation Science*, 43(4):408–416.
- [Laporte *et al.*, 2014] Laporte, G., Vidal, T., y Riera-Ledesma, J. (2014). A review of routing and inventory routing problems. *European Journal of Operational Research*, 239(1):1–15.
- [Lenstra y Kan, 1981] Lenstra, J. K. y Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227.
- [Lindauer *et al.*, 2022] Lindauer, M., Eggenberger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhopf, T., Sass, R., y Hutter, F. (2022). Smac3: A versatile bayesian optimization package for hyperparameter optimization.
- [López-Castro *et al.*, 2023] López-Castro, L. F., Solano-Charris, E. L., y Pagès-Bernaus, A. (2023). Environmental approach for the design of raw milk collection routes with a heterogeneous fleet. *Computers and Operations Research*, 142:105759.
- [López-Ibáñez *et al.*, 2016] López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., y Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.
- [Marmion *et al.*, 2013] Marmion, M.-E., Mascia, F., López-Ibáñez, M., y Stützle, T. (2013). Automatic design of hybrid stochastic local search algorithms. En Blesa, M. J., Blum, C., Festa, P., Roli, A., y Sampels, M., editores, *Hybrid Metaheuristics*, pp. 144–158, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Masson *et al.*, 2016] Masson, R., Lahrichi, N., y Rousseau, L.-M. (2016). A two-stage solution method for the annual dairy transportation problem. *European Journal of Operational Research*, 251(1):36–43.

- [Mester y Bräysy, 2007] Mester, D. y Bräysy, O. (2007). Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers Operations Research*, 34(10):2964–2975.
- [Ministerio de Agricultura de Chile, 1979] Ministerio de Agricultura de Chile (1979). Decreto 178, reglamento específico para la determinación de la calidad de la leche cruda. Promulgación: 05-JUN-1979, Publicación: 27-SEP-1979.
- [Mohammed y El Kafhali, 2022] Mohammed, T. y El Kafhali, S. (2022). Performance analysis of metaheuristics based hyperparameters optimization for fraud transactions detection. *Evolutionary Intelligence*, 17.
- [Montero, 2021] Montero, E. (2021). Gradual Milk Blending Instances. <https://github.com/elimail/Gradual-Milk-Blending-Instances>.
- [Montero et al., 2022] Montero, E., Paredes-Belmar, G., y Leonardini, O. (2022). Milk collection problem with blending and collection points. *ISA Transactions*, 122:294–311.
- [Muñoz y Smith-Miles, 2017] Muñoz, M. A. y Smith-Miles, K. (2017). Performance analysis of continuous black-box optimization algorithms via footprints in instance space. *Evolutionary Computation*, 25:529–554.
- [Nishijima, 2021] Nishijima, T. (2021). Universal approximation theorem for neural networks. *CoRR*, abs/2102.10993.
- [Or, 1976] Or, I. (1976). *Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking*. Tesis doctoral, Xerox University Microfilms, Ann Arbor, Mich. Published as microfilm dissertation.
- [Paredes-Belmar et al., 2017] Paredes-Belmar, G., Lüer-Villagra, A., Marianov, V., Cortés, C. E., y Bronfman, A. (2017). The milk collection problem with blending and collection points. *Computers and Electronics in Agriculture*, 134:109–123.
- [Paredes-Belmar et al., 2016] Paredes-Belmar, G., Marianov, V., Bronfman, A., Obreque, C., y Lüer-Villagra, A. (2016). A milk collection problem with blending. *Transportation Research Part E: Logistics and Transportation Review*, 94:26–43.
- [Pillac et al., 2013] Pillac, V., Gendreau, M., Guéret, C., y Medaglia, A. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225:1–11.
- [Polat et al., 2022] Polat, O., Berk Kalayci, C., y Topaloğlu, D. (2022). Modelling and solving the milk collection problem with realistic constraints. *Computers & Operations Research*, 142:105759.
- [Polat et al., 2019] Polat, O., Kalayci, C., Bilgen, B., y Topaloğlu, D. (2019). An integrated mathematical model for the milk collection problem. *Pamukkale Univ Muh Bilim Derg*, 25(9):1087–1096. doi: 10.5505/pajes.2019.06791.

- [Polat y Topaloğlu, 2022] Polat, O. y Topaloğlu, D. (2022). Collection of different types of milk with multi-tank tankers under uncertainty: a real case study. *TOP*, 30(1):1–33.
- [Popović et al., 2012] Popović, D., Vidović, M., y Radivojević, G. (2012). Vns-based heuristic for solving the vehicle routing problem with time windows and fluid quantity splitting. *Expert Systems with Applications*, 39(18):13085–13093.
- [Rasku et al., 2016] Rasku, J., Kärkkäinen, T., y Musliu, N. (2016). Feature Extractors for Describing Vehicle Routing Problem Instances. En Hardy, B., Qazi, A., y Ravizza, S., editores, *5th Student Conference on Operational Research (SCOR 2016)*, volumen 50 de *Open Access Series in Informatics (OASIs)*, pp. 7:1–7:13, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [Rasku et al., 2019] Rasku, J., Musliu, N., y Kärkkäinen, T. (2019). On automatic algorithm configuration of vehicle routing problem solvers. *Journal on Vehicle Routing Algorithms*, 2.
- [Ruiz-Meza et al., 2020] Ruiz-Meza, J., Espinosa, A. T., Ayala, D. M., y Ortega, G. M. (EasyChair, 2020). Vehicle routing problem with simultaneous pickup and delivery for milk collection in galeras, sucre. EasyChair Preprint 3750.
- [Sethanan y Pitakaso, 2016] Sethanan, K. y Pitakaso, R. (2016). Differential evolution algorithms for scheduling raw milk transportation. *Computers and Electronics in Agriculture*, 121:245–259.
- [Shlens, 2014] Shlens, J. (2014). Notes on kullback-leibler divergence and likelihood.
- [Smith-Miles y Lopes, 2012] Smith-Miles, K. y Lopes, L. (2012). Measuring instance difficulty for combinatorial optimization problems. *Computers Operations Research*, 39(5):875–889.
- [Taillard et al., 1997] Taillard, E. D., Gambardella, L. M., Gendreau, M., y Potvin, J.-Y. (1997). Adaptive memory programming: A new metaheuristic for combinatorial optimization. *Revue Suisse de Recherche Opérationnelle*, 7(3):1–22.
- [Uchoa et al., 2017] Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., y Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858.
- [Vanschoren, 2018] Vanschoren, J. (2018). Meta-learning: A survey. *CoRR*, abs/1810.03548.
- [Vidal et al., 2012] Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., y Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60:611–624.
- [Vidal et al., 2013] Vidal, T., Crainic, T. G., Gendreau, M., y Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. *Computers Operations Research*.

- [Villagrán *et al.*, 2020] Villagrán, J., Montero, E., y Paredes-Belmar, G. (2020). An iterated local search approach to solve the milk collection problem with blending. En *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8.
- [Yasmin *et al.*, 2024] Yasmin, A., Butt, W., y Daud, A. (2024). Ensemble effort estimation with metaheuristic hyperparameters and weight optimization for achieving accuracy. *PLOS ONE*, 19:e0300296.
- [Zhang *et al.*, 2020] Zhang, C., Chen, W., y Zhang, Z. (2020). *Learning Process Analysis from Learners' Perspective*. Beijing Normal University Press, Beijing.
- [Zhao *et al.*, 2021] Zhao, J., Mao, M., Zhao, X., y Zou, J. (2021). A hybrid of deep reinforcement learning and local search for the vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7208–7218.