Algoritmo greedy para predecir el índice de servicio de pavimento basado en agrupación y regresión lineal o no lineal

Ing. Francisco Javier Anacona Campo

Director: PhD. Carlos Alberto Cobos Lozada

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I+D en tecnologías de la información (GTI)
Área de interés en Gestión de la Información
Popayán, Febrero de 2023

Algoritmo greedy para predecir el índice de servicio de pavimento basado en agrupación y regresión lineal o no lineal

Ing. Francisco Javier Anacona Campo

Tesis presentada a la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca para obtener el título de

Magíster en Computación

Director: PhD. Carlos Alberto Cobos Lozada
Codirectora: PhD. Martha Eliana Mendoza Becerra

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I+D en tecnologías de la información (GTI)
Área de interés en Gestión de la Información
Popayán, Febrero de 2023

Resumen

Uno de los problemas más complejos en la gestión de pavimentos consiste en saber cómo mantener y reparar las carreteras con el menor costo posible y afectando en menor medida el servicio que ellas prestan. Una estrategia muy utilizada para definir cuando una carretera requiere mantenimiento o reparación es la que se basa en el índice de servicio del pavimento (Pavement Serviceability Index, PSI), dicho valor entre más cercano a 5 expresa que la carretera está en óptimas condiciones, mientras que al bajar a 0 quiere decir que la carretera está casi inservible. Predecir el valor del PSI es muy importante en este escenario, ya que esto permite planear a futuro que intervenciones se deben realizar en las carreteras, buscando que sean menos costosas que una rehabilitación completa. Para predecir este valor, en el estado del arte se encuentran diferentes técnicas, entre ellas las basadas en machine learning y más específicamente el modelo de clusterwise linear regresión (CLR) que reporta los mejores resultados. Los trabajos previos muestran que construir este modelo es costoso en tiempo y ese costo crece exponencialmente en la medida que el número de registros de entrenamiento aumenta. En esta investigación se presenta un algoritmo greedy que crea un modelo CLR con base en un conjunto de muestras de pavimentos tomados del departamento de transporte de Nevada en Estados Unidos. Este algoritmo es más eficiente que los presentados en el estado del arte (menor tiempo de construcción y mayor calidad en la predicción). El algoritmo permite definir grupos de tramos de carreteras con sus respectivos modelos de regresión lineal y los pesos de los atributos del dataset de entrenamiento para que en conjunto con el algoritmo 1-NN se facilite la asignación de un nuevo tramo de carretera al grupo adecuado y de esta forma predecir el valor de PSI con el modelo de regresión de dicho grupo.

Dedicatoria

Este trabajo de grado y todas las metas en mi vida están dedicadas a Gladys Campo y Bolívar Anacona mis padres, es mi manera de agradecerles por todos los años de trabajo y esfuerzo que dedicaron para darme educación, también por su infinito amor y paciencia. De ellos debo aprender muchos valores como la bondad, humildad, paciencia y solidaridad, gracias por siempre ayudarme a levantarme y seguir adelante.

Agradecimientos

Mis agradecimientos son para mi director el PhD Carlos Cobos por todo el tiempo que me dedicó para lograr buenos resultados en este trabajo, además por ser una persona integra a la que admiro mucho, también agradecerle al ingeniero civil Silvio Ramírez que me regaló parte de su tiempo y me ayudó con la definición de las variables del conjunto de datos y con el análisis de los modelos resultantes.

Tabla de Contenido

Índice de Tablas

Índice de Figuras

Índice de Anexos

CAPÍTULO 1

1 Introducción

1.1 Planteamiento del problema

El proceso de mantenimiento y reparación de las carreteras es realizado por ingenieros expertos que gestionan y predicen el deterioro de los pavimentos, además de los impactos económicos que pueden ocurrir al aplicar correctivos sobre estos. Los mencionados expertos se basan en modelos del deterioro de las carreteras que predicen el estado del pavimento a lo largo del tiempo. Estos modelos presentan diversos tipos de dificultades como la calibración de los modelos, las características geográficas [1], entre otros, lo que hace que el proceso sea demorado y costoso. Por esto, la capacidad de modelar y predecir la condición del pavimento con precisión es fundamental para el éxito de los sistemas de gestiónde pavimentos (Pavement Management Systems, PMS) [2].

Estos modelos de predicción del deterioro del pavimento deben incluir todos los factores que afectan al pavimento, además de los efectos de los mantenimientos pasados. Muchos de los modelos de deterioro que existen actualmente en la literatura son modelos que incluyen pocas variables explicativas y la mayoría no incorpora los efectos del mantenimiento. Dentro del estado del arte se evidencia que el desempeño de los modelos obtenidos es aceptable, pero se podría mejorar, tanto para el enfoque lineal como para el no lineal [3].

La medida del comportamiento del pavimento depende de unos parámetros fundamentales quese definen en [4] como la fiabilidad y la vida esperada de servicio de un segmento de carretera. La vida esperada de servicio es el mejor parámetro para determinar el comportamiento del pavimento y puede ser medida mediante la métrica denominada Pavement Serviceability Index (PSI) o índice de servicio del pavimento, que en términos generales determina la condición actual del pavimento [5].

Cada tramo de pavimento tiene características particulares dependiendo de su construcción, además, son afectados por diversos factores como el tráfico, el clima, el tipo de suelo, entre otros. Debido a estos factores se pueden encontrar segmentos de carretera de diferentes localizaciones que son similares entre sí, por esto, en este contexto se han utilizado técnicas de agrupamiento como la regresión lineal por clusterwise (Clusterwise Linear Regression, CLR) [6], en la que se crean grupos con muestras de pavimento que no necesariamente son similares entre sí, pero que en conjunto permiten mejorar las predicciones de los modelos de comportamiento. El concepto se ha generalizado y en la literatura también se encuentran propuestas con modelos de regresión no lineales (Clusterwise NonLinear Regression, CNLR) como el presentado en [7].

En la literatura se reporta desde finales de los 70's diversas técnicas que buscan mejorar la calidad de los grupos que se forman y los modelos que se obtienen. Para lograr esto, se busca reducir el error cuadrado medio, reducir el error absoluto medio, aumentar la correlación de los modelos o minimizar/maximizar criterios de información como Akaike (AIC) y BIC. Se ha hecho uso de algoritmos de agrupación como K-Means, Mapas Autoorganizativos (Self-Organizing Map, SOM), Expectation-Maximization (E-M), algoritmos basados en densidad y algoritmos difusos como Fuzzy C-Means mezclados con modelos lineales generalizados, o no lineales. También se han usado otras estrategias para agrupación basadas en ramificación y poda (Branch & Bound), heurísticas de intercambio de arranque múltiple y generación de columnas perturbadas. Se han usado enfoques basados en diversas metaheurísticas como el recocido simulado (Simulated Annealing, SA), la optimización basada en enjambres de partículas (Particle Swarm Optimization, PSO), los algoritmos genéticos, mezclas de genéticos con PSO y algoritmos multi objetivo como NSGA-II y MOPSO. Y también se han usado diversos clasificadores como K-NN, redes neuronales y Random Forest para la fase de predicción. La **sección 2** de este documento presenta en detalle el estado del arte del área.

Teniendo en cuenta lo anterior y que en el estado del arte no se había usado un algoritmo con enfoque greedy que en forma divisiva creara grupos basados en la correlación de los elementos (observaciones/registros) y sus correspondientes modelos lineales o no lineales de los grupos, en este trabajo surgió la siguiente pregunta de investigación: ¿Es posible obtener mejores modelos de regresión lineal o no lineal para la predicción del PSI mediante un algoritmo voraz (greedy) que

realice un proceso divisivo de grupos guiado por la correlación de los elementos a sus modelos en los grupos formados?

Con la anterior pregunta, la hipótesis principal (alternativa) del trabajo de grado correspondió a:

H1: Un algoritmo voraz (greedy) que realiza un proceso divisivo de grupos guiado por la correlación de los elementos a sus modelos lineales o no lineales permite mejorar la predicción del PSI en comparación con los reportados en el estado del arte basado en el error absoluto medio (MAE), error cuadrático medio (RMSE), error cuadrático medio normalizado (NRMSE) y el porcentaje de puntos predichos dentro de 15% del valor de PSI real.

Por otro lado, la hipótesis nula planteada para esta investigación es:

H0: Un algoritmo voraz (greedy) que realiza un proceso divisivo de grupos guiado por la correlación de los elementos a sus modelos lineales o no lineales NO permite mejorar la predicción del PSI en comparación con los reportados en el estado del arte basado en el error absoluto medio (MAE), error cuadrático medio (RMSE), error cuadrático medio normalizado (NRMSE) y el porcentaje de puntos predichos dentro de 15% del valor de PSI real.

1.2 Aportes del proyecto

Este trabajo de grado en modalidad investigación (tesis de maestría), realizó aportes de nuevo conocimiento para la comunidad académica y científica que trabaja en CLR aplicada a la predicción del PSI en dos aspectos, primero con la propuesta de un nuevo algoritmo con un enfoque voraz (codicioso o greedy) que entrega mejores resultados a los reportados en el estado del arte y segundo, con la obtención de grupos y modelos de predicción lineal del comportamiento del pavimento que son diferentes a los existentes con una precisión superior y que permite a la comunidad su análisis detallado. Es preciso comentar que, con estos resultados, se espera que en la práctica esto ayude a reducir costos en el mantenimiento de las vías.

El código y datos resultado de la investigación se han publicado en https://gitlab.com/pacho328/paviments-greedy-weigths-gbhs, buscando con ello que cualquier investigador, académico o industrial lo use para replicar los experimentos, probar otros datos de entrenamiento de PSI o inclusive usar en otros

problemas de regresión donde el modelo CLR pueda ser usado, como en medicina, marketing, economía, gobierno, educación, agricultura, entre otros.

1.3 Objetivos

A continuación, se presentan los objetivos tal y como fueron aprobados en el documento de anteproyecto por parte del Consejo de Facultad de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

1.3.1 Objetivo general

Proponer un algoritmo voraz para la predicción del índice de servicio de pavimento (PSI) basado en la agrupación inteligente de modelos de regresión lineal o no lineal de muestras de segmentos de pavimentos de carreteras de Nevada en Estados Unidos.

1.3.2 Objetivos específicos

1. Modelar un algoritmo voraz que realice la agrupación inteligente de modelos de regresión lineal (Clusterwise Linear Regression, CLR) siguiendo el Patrón de Investigación Iterativa propuesto por Pratt [8] para predecir el índice de servicio de pavimento (PSI).

2. Adaptar el algoritmo voraz previamente propuesto usando el PII para definir modelos de regresión no lineal (Clusterwise Nonlinear Regression, CNLR) buscando refinar la predicción del índice de servicio de pavimento.

3. Evaluar el algoritmo propuesto frente a dos propuestas del estado del arte, recocido simulado con regresión de todos los subconjuntos (ASR) presentada en [9] y recocido simulado con mínimos cuadrados ordinarios (OLS) presentada en [7], usando un conjunto de datos del departamento de transporte de Nevada (EE.UU) y comparar los resultados obtenidos con cuatro métricas, a saber, error absoluto medio (MAE), error cuadrático medio (RMSE), error cuadrático medio normalizado (NRMSE) y el porcentaje de puntos predichos dentro del 15% del valor de PSI real.

1.4 Resultados obtenidos

A continuación, se resumen los principales resultados de la presente investigación:

- **Monografía del trabajo de grado**. Corresponde al presente documento que contiene introducción (problema, justificación, objetivos y resultados), estado del arte, propuesta del algoritmo voraz basado en CLR, así como la experimentación y comparación de resultados obtenidos, conclusiones y trabajos futuros que el grupo de investigación espera desarrollar en el corto plazo, referencias y anexos.
- **Código fuente de la solución propuesta**: El algoritmo voraz (CLR) implementado en Java usando la librería de Weka. Está disponible en https://gitlab.com/pacho328/paviments-greedy-weigths-gbhs.
- **Artículo 1**: Artículo publicado en una revista nacional indexada en categoría B según el PUBLINDEX de MINCIENCIAS con la siguiente referencia: F. Anacona-Campo., C. Cobos-Lozada., M. Mendoza-Becerra. "Algoritmo greedy para predecir el índice de servicio de pavimento basado en agrupación y regresión lineal", volumen 8, número 3, pp. 119-134, 2020. DOI: https://doi.org/10.17081/invinno.8.3.4708. También disponible a través de https://revistas.unisimon.edu.co/index.php/innovacioning/article/view/4708. Ver **Anexo 1.**
- **Artículo 2**: Artículo en evaluación en revista internacional con los resultados finales de la investigación. Ver **Anexo 2.**

1.5 Estructura de la monografía

A continuación, se describe de manera general el contenido y organización de la presente monografía:

CAPÍTULO 1: INTRODUCCIÓN: Corresponde al presente capítulo, que introduce el tema, la pregunta de investigación, los aportes del trabajo, los objetivos (general y específicos) definidos para el proyecto y que previamente fueron aprobados, un resumen de los resultados obtenidos y finalmente se describe como está organizada la monografía.

CAPÍTULO 2: ESTADO DEL ARTE: Este capítulo presenta un estado del arte de trabajos e investigaciones relacionadas con CLR y otros relacionados más específicamente en su uso en la gestión de pavimentos.

CAPÍTULO 3: MODELO PROPUESTO: En este capítulo se presenta una descripción detallada de la solución propuesta, que incluye: descripción de los datos

utilizados en la propuesta, explicación del algoritmo greedy, se explican en detalle los componentes del algoritmo con un ejemplo y finalmente la complejidad computacional.

CAPÍTULO 4: EXPERIMENTACIÓN: En este capítulo se describe el proceso utilizado para la experimentación, el proceso de evaluación de los modelos resultantes, los resultados obtenidos y la comparación con el modelo base y los resultados reportados en el estado del arte.

CAPÍTULO 5: CONCLUSIONES Y TRABAJO FUTURO: En este capítulo se presentan las conclusiones obtenidas en este trabajo de grado y después se mencionan las ideas que el grupo de investigación espera trabajar en el corto plazo.

CAPÍTULO 6: REFERENCIAS BIBLIOGRÁFICAS: Este último capítulo contiene las referencias bibliográficas de los artículos y libros consultados durante la realización del proyecto.

CAPÍTULO 2

2 Estado del Arte

Los documentos relacionados con la temática de investigación se buscaron en Google Scholar, IEEEXplore, Scopus y ScienceDirect utilizando la cadena de búsqueda “("clusterwise" OR "cluster-wise" OR "cluster linear regression") AND (pavement OR asphalt OR concrete)”. Este proceso arrojó como resultado 68 artículos relacionados con clusterwise en general y otros que se enfocan a la aplicación de clusterwise en segmentos de pavimento. A continuación, se presentan los principales aportes de cada uno de estos trabajos ordenados cronológicamente en orden ascendente (más antiguos a más recientes).

En 1979 [6] se presentó el enfoque CLR que tiene como objetivo agrupar las observaciones o registros de un dataset, con el fin de minimizar la suma del error cuadrado de un modelo de regresión lineal de las observaciones que conforman cada grupo frente a sus datos reales. En esta propuesta se definen tres pasos principales. El paso 1 realiza una partición de las observaciones en k grupos donde cada grupo debe tener un mínimo número de registros l donde $l \ll m$. El paso 2 busca intercambiar de cada par de grupos un elemento de cada uno que permita minimizar la suma del error de los dos grupos (el grupo al que se le va a quitar un elemento debe tener más de l observaciones). El paso 3 hace que se repita el paso 2 mientras se encuentre una mejora. Este proceso tiene la desventaja de depender de la inicialización establecida en el paso 1.

En 1982 [10] se planteó una versión rápida para CLR basada en la propuesta anterior [6], la cual comparte el mismo objetivo y metodología, pero se diferencia en un pequeño pero significativo cambio. En este trabajo se permite que cada grupo tenga un número mínimo de observaciones diferente al de los otros grupos. Este proceso sigue teniendo la desventaja de depender de la inicialización establecida en el paso 1, pero los autores comentan que el algoritmo se puede ejecutar varias veces y de ellas seleccionar la mejor solución basada en la función objetivo (minimizar el

error cuadrado entre el valor predicho por el modelo de regresión y el valor real de cada observación).

En 1986 [11] se presentó un algoritmo CLR basado en las propuestas anteriores [6] y [10] pero la función objetivo que se optimiza se basa en la suma de los errores absolutos entre el valor predicho por el modelo de regresión del grupo al que pertenece la observación y el valor real de cada observación. Esta función objetivo corresponde con la norma L1 también conocida como la distancia de Manhattan mientras que los trabajos previos toman la normal L2 o distancia Euclidiana, donde L1 se considera una mejor forma de guiar el proceso de optimización.

En 1988 [12] presentó una metodología basada en mezclas condicionales y máxima verosimilitud para realizar CLR. Esta metodología estima simultáneamente modelos de regresión y la pertenencia de las observaciones a K grupos separados. La metodología se trabaja con el algoritmo Expectation-Maximization (E-M) [13] para la estimación de parámetros. Se realizó un análisis de Monte Carlo para evaluar la calidad de la metodología, utilizando conjuntos de datos sintéticos. Finalmente, se mencionan diversas áreas de aplicación utilizando este enfoque.

En 1989 [14] propusieron una metodología basada en recocido simulado (simulated annealing, SA) para CLR que agrupa simultáneamente las observaciones en un número predeterminado de grupos y estima los coeficientes de las funciones de regresión correspondientes a cada grupo. Este procedimiento comienza cuando se selecciona una solución factible que inicia aleatoriamente y calcula el valor de la función objetivo, después elige al azar un punto dentro del espacio m dimensional (m es la dimensión de la función objetivo), se calcula el valor de la función objetivo de la nueva solución, y para aceptar la nueva solución se establece una función de probabilidad que establece si se va a aceptar o no la solución, esta probabilidad depende de dos condiciones la primera retorna el valor de 1 si la nueva solución es mejor, y en caso contrario se define por el exponente de la multiplicación de valor de la temperatura por la diferencia entre los valores resultantes de las funciones objetivo. A este enfoque se le realizó un análisis de Monte Carlo que demuestran la adaptación consistente del procedimiento de recocido simulado para conjuntos de datos de varios tamaños y diferentes especificaciones reportando buenos resultados.

También en 1989 [15] presentaron una metodología de segmentación de beneficios enfocada en el consumidor mediante CLR. Este enfoque inicia desde una preclasificación aleatoria, después busca minimizar el error en dos fases, una de transferencia de elementos de un grupo a otro y otra de intercambio de dos elementos entre segmentos, buscando mejorar las estimaciones resultantes de las combinaciones de los pesos de los productos. Cada vez que un grupo se adiciona o se remueve, se calcula la matriz de suma de cuadrados y productos (SSP) del clúster en cuestión. Este método es adecuado para encontrar segmentos de consumidores y problemas que no tengan un dataset de gran tamaño. El trabajo fue evaluado con data sets sintéticos.

Siguiendo en 1989, en [16] proponen un enfoque Fuzzy Clusterwise Regression (FCR) para la segmentación de beneficios que cuenta con dos etapas, en la primera etapa se tienen unos valores parciales que se estiman utilizando modelos de regresión, y en la segunda etapa estas estimaciones se utilizan para agrupar las observaciones mediante métricas de análisis tales como análisis factorial y análisis discriminante . Este artículo también proporciona un análisis de la prueba de Monte Carlo con el fin de evaluar el rendimiento de este enfoque, obteniendo buenos resultados en comparación con la metodología de agrupación superpuesta (overlapping cluster regression, OCR) [14].

En 1993 [17] se investigó la regresión en agrupaciones difusas (FCR) buscando estimar agrupaciones difusas y las ponderaciones de los beneficios inferidos dentro de un grupo. Los datos consisten en evaluaciones generales de los consumidores (por ejemplo, preferencias) de K productos. Cada grupo tiene un vector diferente de importancia de peso, de antemano se conoce el número de grupos y todos los sujetos pueden tener membresía parcial en todos los grupos. Las membresías oscilan entre 0 y 1 e indican el grado de pertenencia del sujeto al grupo, donde 1 equivale al valor más alto de pertenencia. FCR estima la importancia de los beneficios percibidos del producto en cada uno de los grupos, y simultáneamente estima las membresías. El parámetro de peso difuso definido por el usuario influye en la medida en que los sujetos pertenecen a más de un grupo. El algoritmo se inicia con una partición difusa aleatoria, es decir, los pesos se eligen al azar. Luego, las ponderaciones de los beneficios se estiman dentro de cada agrupación utilizando un análisis de regresión de mínimos cuadrados y los nuevos valores de las membresías a partir de los residuos de regresión y estas estimaciones se utilizan

nuevamente para obtener nuevas estimaciones de las ponderaciones de los beneficios. El algoritmo FCR consiste en la ejecución de varias iteraciones con los procesos anteriores y se detiene en el momento que las membresías no cambian o el valor del error no disminuya. Después de validar con validación cruzada, esta propuesta superó el 90% del ajuste predictivo y superó el ajuste predictivo de la regresión general no agrupada.

En 1996 [18] se presentó un enfoque que modifica agrupaciones con el fin de mejorar la respuesta de una o varias variables exógenas(continuas o categóricas). Especialistas en marketing han empelado la segmentación basada en clústeres, pero han detectado problemas asociados con la relación de los grupos o segmentos con las variables (marcas de los productos o medidas de lealtad del proveedor). Este trabajo buscó grupos individuales internamente homogéneos y evidenció un comportamiento diferencial con respecto a una variable exógena (continua o categórica), para tratar este tema modificaron sistemáticamente la agrupación de K-means para mejorar la predicción. Este cambio está diseñado para respetar una restricción especificada por el usuario. Este método de segmentación realiza predicciones razonablemente precisas de una variable exógena y mantiene una alta similitud de los valores de los miembros en las variables a predecir dentro de cada grupo separado.

En 1999 [19] se utilizó un enfoque de programación matemática para el modelado de regresión por conglomerados (grupos). En este enfoque se adiciona un parámetro de heterogeneidad en la regresión tradicional y el análisis discriminante. Demuestra que la estimación del modelo de regresión por agrupaciones no es equivalente a resolver un modelo de programación de enteros mixtos no lineales (NMIP) [20], esto debido a los enfoques de los objetivos y la validez de los modelos propuestos, ya que al momento de calcular la estimación el modelo de programación lineal (original) se transforma en un modelo de programación no lineal simple con restricciones lineales.

En el 2002 [21] se propuso una agrupación por regresión lineal difusa de mínimos cuadrados para datos de entrada y salida difusos, es decir se utiliza un modelo de regresión difusa para evaluar la relación funcional entre las variables dependientes e independientes en un entorno difuso. En este artículo se proponen dos métodos de estimación, el primer enfoque es la distancia aproximada de mínimos cuadrados difusos y utiliza una fórmula de aproximación para el producto de dos números

difusos, el segundo enfoque utiliza la integración de la distancia de intervalo a sus agrupaciones de $nivel - w$ [22], en ambos enfoques se validó la suma de errores cuadrados de la regresión aplicados a dos ejemplos tomados de [22]. Ambos métodos sugieren un procedimiento alternativo y eficiente para estimar las variables de los modelos de regresión lineal difusa con salidas difusas, parámetros difusos.

En 2002 [23] propusieron formar un grupo como un subconjunto de datos que no contiene valores, a este procedimiento se le llamó agrupaciones de punto fijo (FPC) para la regresión lineal. En este procedimiento se calcula la varianza, los estimadores de regresión lineal y los puntos que se encuentran demasiado lejos del hiperplano de regresión y se declaran valores atípicos. Primero se define la escala del error mediante una ecuación de punto fijo y se clasificanlos grupos por las diferentes escalas de error, además las agrupaciones cuentan con 3 características, la primera en donde no es necesario establecer el número de agrupaciones, la segunda en donde no todos los puntos necesitan pertenecer a un grupo de regresión. Finalmente, las observaciones pueden pertenecer a más de un grupo. Este enfoque fue comparado con métodos que utilizan el estimador de máxima verosimilitud y concluyen que el método obtiene buenos resultados de estimación pero con un alto costo computacional.

En 2003 [24] se presentó un artículo sobre grupos, valores atípicos, regresión y grupos de puntos fijos (FPC). A diferencia de [23], en este enfoque se implementó LS-FPCV una propuesta que usa estimadores de grupos con diferentes distribuciones e implementa un procedimiento para encontrar todos los FPCV "sustanciales" con alta probabilidad. Consta de 6 pasos, primero inicializa los parámetros del algoritmo, segundo calcula los valores de la distribución de probabilidad de las variables, tercero genera los subconjuntos basados en el valor de probabilidad del paso anterior, cuarto busca las semejanzas entre los grupos, quinto calcula para cada grupo el índice de enlace único, finalmente, establece un valor final de FPCV al valor con mayor frecuencia. El enfoque propuesto se comparó con MLCLR [12] y MBGCN [25] y se concluyó que FPC puede superar a MLCLR en los conjuntos de datos donde la distribución de los datos es normal en las variables independientes, y puede ser mejor que MBGCN en distribuciones no normales.

También en 2003 [26] se planteó una metodología de regresión multicriterio (con varios criteriosy restricciones) [27] para la configuración de la agrupación buscando identificar agrupamientos que proporcionen segmentos homogéneos y buenos

valores de R^2 para cada una de las mediciones de variables dependientes. Se implementó una estrategia de solución heurística basada en SA para resolver el problema de optimización resultante y se obtuvo como resultado un enfoque viable para la agrupación multicriterio. Además, evidenciaron que las restricciones aplicadas al tamaño de los conjuntos de datos que pueden manejarse de manera efectiva. Adicionalmente, concluyen que en los ensayos empíricos las temperaturas altas con enfriamiento lento en SA aumento significativamente el costo computacional con muy poca mejora en la calidad de la solución.

En 2005 [28] se propuso una solución CLR usando un proceso estocástico donde para cada grupo, los estimadores de los coeficientes de regresión están dados por la regresión parcial de los mínimos cuadrados sobre las variables explicativas. Cada grupo se compone de dos partes, una de la varianza residual y la otra que representa la distancia entre las predicciones dadas por los modelos globales y locales. El número de clústeres se trata como desconocido y se analiza la convergencia del algoritmo. El enfoque se evaluó utilizando la suma de los errores cuadrados (SSE) como medida de ajuste y tomando un conjunto de datos de [29], dando como resultado mejores modelos que los obtenidos en [29].

En 2006 [30] se investigó la manera de predecir la condición del pavimento usando CLR, buscando ajustar los datos a más de una función y de esta manera modelar el deterioro de la condición del pavimento. La agrupación de los pavimentos utilizó los siguientes pasos: Primero, se identificó el tipo de pavimento (flexible, rígido… etc.), región geográfica y clima. Segundo, se identificó la variable dependiente como la clasificación de la condición del pavimento o pavement condition rating (PCR) y se establecieron las posibles variables explicativas. Tercero, se identificó la función de predicción (curva de potencia, función polinomial). Cuarto, se identificaron las variables explicativas significativas utilizando regresión gradual. Quinto, se desarrolló y analizó el modelo final, finalmente se hacen las predicciones de nuevos segmentos de pavimentos. En el paso quinto se realiza el proceso de agrupación que consta también de cinco pasos; primero se elige un punto base con PCR y la edad del pavimento, segundo, se calcula el valor el $PCR1{,}0$ para cada grupo, tercero, se calcula la membresía de dos grupos consecutivos $Z1{,}0$ y $Z2{,}0$, cuarto, se calcula el $PCRt{,}1$ y $PCRt{,}2$, finalmente se predice el PCR utilizando la siguiente fórmula $PCR = Z1{,}0PCRt{,}1 + Z2{,}0PCRt{,}2$. Para la asignación de un nuevo pavimento a un grupo es necesario calcular la distancia con K-nn, adicionalmente la cantidad de grupos

es definida por el SSE, es decir, si se adiciona un nuevo grupo y el error aumenta, este grupo no se toma en cuenta. Este enfoque se recomienda en los casos en que los tipos pavimento no están bien definidos, pero finalmente concluyen que no se puede demostrar la validez del modelo propuesto.

También en 2006 [31] se planteó un análisis de regresión lineal difusa por conglomerados o fuzzy clusterwise linear regression model (FCWLR) con una variable de salida difusa simétrica. Con este enfoque un elemento puede pertenecer a dos o más grupos, esto debido a que los parámetros de regresión y los grados de membresía se pueden calcular simultáneamente minimizando una única función objetivo, de esta manera se busca la clasificación de un conjunto de unidades y su interpolación, además este proceso mide la incertidumbre en el proceso de asignación, permitiéndole al modelo difuso tener en cuenta las posibles relaciones lineales entre el tamaño de los spreads (simetría) y la magnitud de los centros estimados. La bondad de ajuste del modelo FCWLR se basó en la medida $R2$.

En 2007 [32] investigan sobre la segmentación del mercado basada en enlaces de satisfacción del cliente (cliente-lealtad). Este enfoque genera modelos de mezcla condicional que permiten la clasificación probabilística simultánea de observaciones en segmentos subyacentes y la estimación de modelos de regresión que explican las medias y las variaciones de la variable dependiente dentro de cada uno de esos segmentos (o grupos). Este modelo se aplica a la relación entre lealtad y satisfacción del cliente. El procedimiento de agrupación que genera esto modelos funciona de la siguiente manera, inicia asignando el elemento (encuestado) i-ésimo a un grupo usando la regla de Bayes con la probabilidad estimada que resulta en una agrupación difusa de los encuestados en los grupos. Luego, se estiman las proporciones de mezcla, coeficientes, varianzas y probabilidades de membresía. El número de grupos se estima a través del criterio de información Bayesiano (BIC) [33]. Finalmente, se utiliza el análisis de correspondencia multivariado para analizar la asociación entre el consumidor y los segmentos, e identificar cada subgrupo.

En 2008 [34] se estudió el sobre ajuste en CLR. El procedimiento de evaluación comparó los resultados de los datos observados con los obtenidos en un conjunto de permutaciones aleatorias de las medidas de respuesta para un problema de tamaño modesto con N = 60 objetos y K = 3 grupos donde el número de particiones factibles es más de 7.06 X1027. En el trabajo incorpora una fase de intercambio por pares, separando la variación en la variable de respuesta que explica los modelos

de regresión dentro del clúster. Como resultado de este enfoque se evidenciaron pocos beneficios más allá de lo que podría lograrse dividiendo a los sujetos basándose solamente en la medida de respuesta, Esto debido a que el enfoque es propenso al sobre ajustarse, los análisis fueron basados en el problema propuesto en [27].

También en 2008 [35] se utilizó un análisis probabilístico de las clasificaciones de deterioro en el pavimento con el método de regresión por clústeres. Este enfoque utilizó una agrupación por mezcla, donde cada observación se asigna al grupo basado en el valor máximo de membresía. Cada vez que se actualiza la función objetivo y la probabilidad, se calcula una probabilidad posterior, cada predicción realizada de cada grupo es el resultado del producto del porcentaje del pavimento sobre la cantidad total de tipos de pavimento. El modelo propuesto se comparó con un modelo de Markov [36] que consiste en clasificaciones de agrietamiento discreto. Como resultado, se obtuvo que el modelo propuesto proporciona mejores predicciones que el modelo de Markov, generando un RMSE de 3,5 en comparación con un 13 del modelo de Markov. El modelo propuesto predijo correctamente cinco de las siete calificaciones, mientras que el modelo de Markov no predijo correctamente ninguna de ellas.

Adicionalmente, en 2008 [37] se presentó la optimización del sistema de remedición de aguas subterráneas basado en el riesgo que esto supone para la salud, a través de CLR utilizando un método de análisis de conglomerados paso a paso (Stepwise Cluster Analysis, SCA) el cual está compuesto por 5 pasos. En el primer paso se genera un criterio con el fin de determinar si las observaciones sé pueden dividir en dos grupos, además se comprueba si los dos grupos obtenidos pueden ser fusionados en uno; el segundo paso divide los grupos en términos de los puntos óptimos de corte; el tercer paso verifica si alguno de los subgrupos puede clasificarse en uno bajo un criterio probabilidad clasificación donde el error sea inferior a un nivel aceptable (5%) denominado grupo de consejo; el cuarto paso utiliza el análisis de regresión para capturar las relaciones de cada uno de los grupos generados; finalmente, los resultados arrojados por CLR ayudaron al proceso de optimización al evaluar el riesgo para la salud del sistema de remedición de aguas subterráneas, ya que permiten realizar aproximadamente 3.000 simulaciones por segundo.

En 2009 [38] proponen un sistema de recomendación híbrido combinando sistemas de recomendación colaborativos. El procedimiento de agrupación inicia al instanciar la calificación del usuario, después se procede a calcular el peso y varianza de cada grupo con mínimos cuadrados ponderados, finalmente, se obtiene una matriz que contiene las características para cada elemento. Este enfoque tiene la ventaja de predecir nuevos elementos y proporciona un marco para explicar sus predicciones.

También en 2009 [39] se implementó un algoritmo de clusterwise para determinar las relaciones de entrada-salida (lineales) de algunos procesos de fabricación utilizando Fuzzy logistic controller basado en dos enfoques, el primero denominado entropy-based fuzzy clustering (EFC) y el segundo con un algoritmo más clásico denominado Fuzzy C-meas (FCM) utilizado en el proceso de agrupación y un algoritmo genético para la optimización de los grupos. Ambos modelos se probaron en los procesos de mecanizado de flujo abrasivo y soldadura por gas inerte de tungsteno. Al realizar las pruebas con 50 casos y evaluando el porcentaje de la desviación de las predicciones, los resultados mostraron que EFC obtuvo mejores resultados que FCM debido a que es más flexible y con ello obtuvo mejores agrupaciones.

En 2010 [40] se presentó una combinación entre la regresión de cadena [41] para estimar los coeficientes de regresión y la agrupación difusa regularizada con el fin de determinar sistemáticamente un grado óptimo de difuminación en las membresías, de modo que estas varíen gradualmente de 0 a 1 mientras se maneja la multicolinealidad entre las variables. Este procedimiento agrega un parámetro de penalización para los coeficientes de regresión y para las membresías de los grupos, además tiene como objetivo la reducción de los valores absolutos de las estimaciones de mínimos cuadrados de los coeficientes de regresión hacia cero en cada grupo, esto genera sesgo, pero a cambio logra variaciones más pequeñas de las estimaciones entre los grupos. Como resultado, este enfoque tiende a dar estimaciones más confiables de los coeficientes de regresión en presencia de multicolinealidad, pero tiene una limitación en el tiempo de cálculo, ya que el número de parámetros a probar aumenta rápidamente con un aumento en el número de grupos, además, corre el riesgo de sobre ajustarse.

También en 2010 [42] proponen una adaptación de la regresión por grupos con valores de intervalo; el enfoque propuesto combina el algoritmo de agrupamiento dinámico con el método de regresión de centro y rango con el fin de identificar la

partición de los datos y los modelos de regresión relevantes, uno para cada grupo de modo que un criterio de adecuación que mide el ajuste entre los grupos y sus prototipos (representación inicial del grupo) se minimiza localmente. La particularidad de este tipo de método es que el prototipo de cada grupo está representado por el hiperplano dado por la relación de la regresión lineal entre la variable dependiente y las variables predictoras independientes.

Siguiendo en 2010 [43] se presentó un algoritmo de agrupamiento híbrido de refinamiento (REHCA) con una metodología que trabaja en tres fases; en la primera fase, obtiene una partición de los datos en grupos utilizando un algoritmo de agrupamiento basado en densidad, en la segunda fase, cada uno de los grupos basados en densidad se divide en un número fijo de grupos, luego, cada conjunto de clústeres lineales se procesa mediante un procedimiento de refinamiento que elimina los valores atípicos y fusiona los pares de clústeres lineales que se ajustan al mismo modelo. Los grupos obtenidos después de esta fase se reagrupan y se ejecuta el procedimiento de refinamiento para producir el agrupamiento final. La aplicación de refinamiento en dos fases reduce el número de pares de clústeres a evaluar, también mejora las posibilidades de que se fusionen los pares de clústeres correctos. Un paso final del post procesamiento identifica los puntos que se encuentran cerca de otros grupos y, por lo tanto, podrían asignarse incorrectamente. Las líneas de regresión se vuelven a calcular sin tener en cuenta estos puntos para lograr estimaciones más precisas, en esta fase también se identifican valores atípicos; finalmente, se realiza una comparación de esta propuesta con LGA [44] usando GAP [45] en datos del mundo real y sintéticos "realistas", y se evidencia que REHCA obtiene una estimación de clúster precisa y eficiente.

Continuando en el 2010, en [46] se propuso una regresión lineal robusta por conglomerados a través de recortes; este artículo es una extensión de la metodología TCLUST [47] y se basa en cómo se controlan los parámetros de dispersión k en lugar de los valores propios de las matrices de covarianza k. Esta metodología adiciona unos parámetros de dispersión que se actualizan utilizando los residuos cuadrados medios de los grupos y las proporciones de observaciones en cada grupo se utilizan para actualizar los pesos. Se debe resolver un problema de programación matemática cuadrática para controlar la relación entre la dispersión del grupo; los pasos de agrupación se repiten N veces con diferentes inicializaciones aleatorias retornando la mejor solución obtenida. Además, se

implementa una restricción que limita la relación de dispersión de grupo; estos procedimientos mejoran las asignaciones en los grupos e incorpora algunas observaciones descartadas erróneamente.

Adicionalmente, en 2010 [48] se presentó una clase de modelos de regresión por conglomerados con variable de respuesta difusa y variables explicativas numéricas, que incorpora agrupamiento difuso. Este enfoque busca evitar el problema de heterogeneidad en la regresión difusa subdividiendo el conjunto de datos en grupos homogéneos y realizando una regresión difusa separada en cada grupo. De esta manera puede estimar simultáneamente los parámetros de regresión y el grado de pertenencia de cada observación a cada agrupación mediante la optimización de una única función objetivo. Esta metodología cuenta con una variable dependiente difusa que no es simétrica, además, este diseño genera modelos de regresión lineal con mejor ajuste. Se ejecutaron múltiples pruebas comparándose con otras propuestas del estado del arte midiendo el valor de SSE dando como resultado un menor valor de SSE que la mayoría de las propuestas.

En 2011 [49] se presentó un enfoque de mínimos cuadrados ponderados para la regresión por conglomerados que utiliza regresión robusta. Inicialmente, se define el número de grupos usando el criterio de información de Akaike (AIC), después se realiza una ponderación aleatoria para determinar una partición de inicio y a continuación, se determinan los M-estimadores (reduciendo los valores extremos de la variable dependiente) usando los mínimos cuadrados. Se usó como función objetivo el coeficiente de determinación R^2 y para no quedar atrapado en óptimos locales,el método se inicia varias veces desde particiones aleatorias. Este método es validado a través de simulaciones de Monte Carlo donde se compara con el enfoque de mezcla finita [5 0] (regresión por conglomerados) y se obtiene como resultado que en la mayoría de casos el enfoque propuesto supera al enfoque de mezcla finita.

También en 2011 [51] se utilizó FCM con vectores de regresión buscando predecir la fuerza del concreto basado en un conjunto de datos que registra la mezcla de varios componentes tales como cemento, agua, plastificante, entre otros. El enfoque empleado consistió en aplicar FCM al conjunto de datos, modificar su membresía desde un mapeo de características seguido por la construcción de los modelos de vectores de regresión, además de utilizar un algoritmo genético para el afinamiento de los parámetros. Esta propuesta fue comparada con dos variantes de una red

neuronal ANFIS que corresponde a un método híbrido para optimizar parámetros y dos variantes de una función fuzzy con estimación de mínimos cuadrados (IFF- LSE y FF-SVRs). La propuesta obtuvo el menor (mejor) valor RMSE promedio.

Siguiendo en 2011 [52] se realizó un método de predicción de las concentraciones de PM10 en el aire mediante regresión por conglomerados. Las mediciones fueron tomadas al noroeste de París, Francia. Para la selección de variables se utilizó Random Forest, para el proceso de agrupación primero se calculó la expectativa condicional de la probabilidad del registro, luego se calcularon los parámetros que maximizan dicha probabilidad. El número de componentes es generalmente desconocido y necesita ser estimado, para ello usan un enfoque clásico que consiste en ajustar modelos de varios componentes y compararlos utilizando el criterio de información Bayesiano BIC [34]. Con este método se demostró que es posible pronosticar con precisión la concentración media diaria de PM10 ajustando una función de predictores meteorológicos y la concentración promedio de PM10 medida el día anterior.

En 2012 [53] se desarrolló un sistema de pronóstico basado en regresión lineal por conglomerados para caracterizar los comportamientos de disolución de líquidos densos en faseno acuoso (DNAPL) en medios porosos, en donde muchas variables pueden ser continuas o discretas y las relaciones entre ellas son inherentemente no lineales. Se propuso un modelo CLR para abordar tales complejidades discretas y no lineales, que consta de cinco pasos: establecimiento de criterios para dividir y fusionar grupos, división de grupos, fusión de grupos, análisis de regresión y predicción. Esta propuesta es la misma utilizada en [37] el proceso de pronóstico que se divide en dos fases: búsqueda de grupos de consejos [37] y estimación de variables de respuesta. Esta propuesta en comparación con el modelo MLR convencional de CLR tuvo un rendimiento superior para predecir comportamientos de disolución de DNAPL bajo complejidades discretas no lineales.

También en 2012 [54] proponen un método de regresión clusterwise llamado Regresión SOMwise que combina una estructura de mapa autoorganizado (SOM) y un perceptrón multicapa MLP. Cada unidad de la estructura SOM tiene como objetivo construir un modelo que reconozca varios grupos en los datos, donde la dependencia entre los predictores y la respuesta es variable de un grupo a otro. Esta metodología trabaja con una neurona ganadora y minimiza el riesgo de que el MLP quede atrapado en óptimos locales. Este método se probó en un problema de

consumo de electricidad y el conjunto de datos incluye un muestreo para cada estado de EE. UU con tres variables independientes (precio de la electricidad, ingreso per cápita y precio del gas) y una variable de respuesta (consumo de electricidad) y concluye que el método trabaja bien en un entorno real comparado con metodologías como regresión lineal de punto fijo (FPC) y máxima probabilidad bajo un modelo de mezcla de regresión (MLRM).

En 2012 [55] además se propone una estrategia de ramificación y poda para resolver el problema de regresión por conglomerados. Es una extensión del algoritmo repetitivo de ramificación y poda (RBBA) [56], [57]. Este método cuenta con 3 características fundamentales: secuenciación de observaciones, optimización de la ramificación y poda de un número limitado de subconjuntos que generan una optimización rápida del conjunto de datos, además una implementación eficiente de cálculos incrementales que elimina la mayoría de las redundancias. Se realizaron evaluaciones con datos reales y sintéticos, dando como resultado que todos los componentes del algoritmo propuesto proporcionan mejoras significativas en los tiempos de procesamiento y cuando se combinan, generalmente proporcionan el mejor rendimiento en comparación con la metodología de optimización lógica-cuadrática mixta por CPLEX.

En 2013 [58] presentan una extensión funcional de regresión difusa por grupos que estima simultáneamente membresías difusas de grupos y funciones de coeficiente de regresión para cada grupo. Es decir, permite que las variables dependientes y predictoras sean funcionales, además de las membresías difusas, los coeficientes de regresión por conglomerados se estiman minimizando una función objetivo que adopta un enfoque de expansión para aproximar datos funcionales. Para lograr esto se establecen dos índices qué miden si los clústeres están separados entre sí en función de las membresías de clúster; estos índices son el índice de rendimiento difuso (FPI) y la entropía de clasificación normalizada (NCE). Los autores realizaron simulaciones para demostrar el rendimiento superior del método propuesto en comparación con su contraparte no funcional y para examinar el rendimiento de varias medidas de validez del grupo y seleccionar el número óptimo de grupos. El método se probó con conjuntos de datos reales de 35 estaciones meteorológicas canadienses para ilustrar su utilidad empírica.

Siguiendo el 2013, en [59] se propuso una regresión lineal multivariante para datos heterogéneos con el fin de identificar la estructura de agrupamiento de los datos y

ajustar las posibles relaciones lineales dentro y entre los grupos, es decir, un modelo de regresión dentro del clúster que requieren variables independientes para identificar grupos homogéneos. En este trabajo tienen en cuenta la posible presencia de una relación lineal entre grupos, ajustando los centroides de los grupos, que actúan como filtros al eliminar la variabilidad no relevante. Esta metodología cuenta con 3 puntos importantes, primero, minimizar la variación de las covariables dentro del clúster, segundo, maximizar la variación entre grupos vista desde la regresión entre grupos y tercero, minimizar la variación dentro de cada grupo resultante.

En el mismo 2013, en [60] se presentó un algoritmo para minimizar una función no convexa no uniforme en regresión lineal por agrupaciones, este algoritmo divide de manera incremental todo el conjunto de datos en grupos que se pueden aproximar fácilmente. Este procedimiento está divido en los siguientes pasos: inicialmente se considera solo un clúster y se encuentra una función de regresión lineal para todo el conjunto de datos, después se genera un buen punto departida para resolver problemas de optimización global, posteriormente se procede a aplicar el algoritmo de Späth [10] a partir de la solución encontrada en el paso anterior, luego se elige la solución que tenga menos parámetros y finalmente se define un criterio de parada. Las pruebas se realizaron en veinte conjuntos de datos pequeños y siete entre medianos y grandes. Se comparó con el algoritmo de Späth de inicio múltiple para la regresión lineal por conglomerados, dando como resultado soluciones significativamente más precisas y considerablemente más rápidas.

En 2014 [61], proponen un método de predicción de la demanda de efectivo en cajerosautomáticos mediante la agrupación y la regresión con redes neuronales. En este artículo realizan el proceso de agrupamiento de los diferentes cajeros utilizando el algoritmo Taylor-Butina´s [62] el cual costa de cinco pasos; inicialmente se construye la tabla de umbral-vecino más cercano usando similitudes en ambas direcciones, después se encuentran los puntos únicos, es decir, puntos de datos (cajeros automáticos) con una lista vacía de vecinos más cercanos, después se calcula el punto de datos con la lista de vecinos cercana más grande. Este punto tiende a estar en el centro de la k-ésima (k clústeres) región más densamente ocupada del espacio de datos. El punto de datos junto con todos sus vecinos dentro de su región de exclusión constituye un clúster. El punto de datos se convierte en el punto de datos representativo para el clúster, después se eliminan todos los

elementos del clúster de todas las listas de vecinos más cercanos y se repite el proceso hasta que no existan puntos de datos con una lista de vecinos más cercanos que no esté vacía y finalmente, se asignan los puntos de datos restantes, es decir, los singletons falsos se asignan al grupo que contiene su vecino más cercano. Como resultado se obtuvo que al agrupar los centros de cajeros similares se reduce la tarea computacional en la fase de pronóstico, mejorando así las predicciones de demanda de efectivo,además, el enfoque propuesto de agrupamiento seguido de la predicción arrojó valores de error porcentual absoluto medio simétrico mucho más pequeños que el enfoque tradicional de predicción directa en toda la muestra sin agrupamiento.

También en 2014 [63] se trabajó en un óptimo global con CLR para la generación de columnas mejoradas con heurísticas, secuenciación y optimización de un subconjunto. Este trabajo tuvo como fin resolver problemas de agrupamiento. El procedimiento que proponen ejecuta las heurísticas (Mixed logical-quadratic, Branch and Bound, Greedy Heuristics, Multistart ExchangeHeuristic, Generating Perturbed Columns), si estas no encuentran las columnas, se realiza una búsqueda a fondo con subconjuntos incrementales, las observaciones son ordenadas por variables duales y de inclusión por reglas de ramificación. Para esta propuesta se utilizó un conjunto de datos compuesto por 100 observaciones creadas sintéticamente de 2 a 6 líneas. Con base en estas líneas, los conjuntos de datos se generan con cantidades incrementales de perturbación de la distribución normal y las variables independientes son aleatorias. Se realizó una evaluación con todas las heurísticas anteriormente mencionadas y se encontró que el método degeneración de columnas solo se vuelve competitivo a medida que aumenta el número de líneas.

Siguiendo en 2014 [64] se estudió la heterogeneidad en el deterioro del pavimento basado en un modelo de regresión lineal por conglomerados. El objetivo de la agrupación fue maximizar la variación dentro del grupo en lugar de confiar en los criterios observados, es decir, la heterogeneidad no observada, se manifiesta en coeficientes a nivel de segmento que se diferencian en su magnitud y signo. Este enfoque busca generar un modelo CLR para la predicción de PSI. Este procedimiento describe el efecto de las variables explicativas en toda la población y supone que la población se compone de K grupos, también admite la asignación simultánea de grupos y la estimación de un conjunto de coeficientes para cada uno de los segmentos. El modelo se estima con el objetivo de minimizar la suma residual

de cuadrados (RSS) y utiliza el algoritmo de intercambio de Späth para resolver el problema de optimización y esta operación a su vez depende de la estimación de los coeficientes del modelo que se calculan utilizando las expresiones de forma cerrada de OLS. El modelo propuesto se evaluó con un conjunto de datos de 131 pavimentos de prueba de La Asociación Americana de Oficiales de Carreteras Estatales y Transportes (AASHO). Los resultados del estudio confirman que la heterogeneidad no observada es significativa en el conjunto de datos. La evaluación del modelo sugiere que el efecto de esta heterogeneidad es capturado adecuadamente por los coeficientes a nivel de segmento.

En 2015 [65] se planteó un modelo de regresión inteligente (Intelligent Regression Model, IRM) con el fin de predecir la respuesta promedio del historial de terremotos con el objetivo de evaluar el diseño de una estructura de concreto reforzada. Para esto realizaron agrupaciones por k-means y generaron modelos por predicción, realizando combinaciones de SA, k-means y wavelet weighted least squares support vector machine (WWLS-SVM), utilizando los registros en función del espectro de diseño iraní con el tipo de suelo III correspondiente al servicio geológico de los Estados Unidos, lo que dio como resultado a WWLS-SVM como el mejor enfoque, ya que el tiempo de optimización se reduce significativamente. Los resultados lograron una solución óptima con el costo óptimo y un error pequeño en las predicciones, basándose en la media del error porcentual absoluto (MAPE), la raíz del error cuadrático medio relativo (RRMSE) y el coeficiente de determinación (R^2).

También en 2015 [66] se investigó acerca de la forma de detectar variables que están causando diferencias en la estructura de los componentes entre diferentes grupos y propone dos heurísticas basadas en CLR (lower-bound, cutoff congruence) de detección con el fin de determinar variables periféricas de una manera sistemática y objetiva. Las heurísticas se basan en el análisis simultáneo de componentes por agrupaciones, que se presentó como una herramienta útil para capturar las similitudes y diferencias en las estructuras de componentes entre los grupos. Los datos con los cuales se trabajó son datos hipotéticos, pero para la aplicación se utilizaron datos del International College Survey (ICS) 2001 y se tomaron 10.018 registros de participantes de 48 diferentes países. La investigación determina que lower-bound tiene un rendimiento superior debido a que la cantidad de falsos positivos fue mínima.

Siguiendo en 2015, en [67] se propone un algoritmo para la regresión lineal en clústeres basado en técnicas de suavizado con un enfoque que divide de forma incremental todo el conjunto de datos en grupos que se pueden aproximar fácilmente mediante una función de regresión lineal. Este procedimiento se lleva a cabo de la siguiente manera, se genera una solución inicial, después se calcula el modelo de regresión lineal a todo el conjunto de datos, se aumenta el tamaño de los grupos y se calcula la regresión lineal a cada grupo, después se crea un nuevo conjunto de datos el cual contiene solo los valores que cumplan con un criterio establecido (minimizar o maximizar la función objetivo). El algoritmo se evaluó utilizando varios conjuntos de datos para el análisis de regresión y se comparó con los algoritmos Späth de múltiples inicios incrementales.

Además, en 2015 [68] se presentó un algoritmo incremental para resolver problemas de CLR. Este algoritmo inicia con un grupo que divide sus elementos con el fin de adicionar un nuevo grupo, en el cual se aplica una suavización hiperbólica de una función incrementada, además utiliza otros parámetros buscando reducir el espacio de búsqueda de los valores de los coeficientes del grupo nuevo y este procedimiento se repite hasta el número k de grupos establecido por el usuario. Este enfoque fue probado usando conjuntos de datos con solucionesconocidas como red wine quality, housing, forest fire, entre otros tomados del repositorio de la UCI [72] y demostró ser capaz de encontrar soluciones globales o casi globales si los conjuntos de datos son lo suficientemente densos, pero falla si un conjunto de datos no es denso o contiene valores atípicos.

En 2017 [69] se planteó una metodología para la agrupación difusa y robusta a través de recortes y restricciones que se basa en verosimilitudes de probabilidad. La robustez del método se logra recortando una proporción fija de observaciones, en un proceso que es autodeterminado por el conjunto de datos. También definen restricciones en las dispersiones de los conglomerados para obtener problemas matemáticamente bien definidos y para evitar la detección de agrupaciones falsas. El proceso se resume en 3 pasos principales; el procedimiento se inicializa varias veces seleccionando aleatoriamente los parámetros iniciales, el segundo paso está compuesto por 3 subprocesos, que inicia calculando los valores de las membresías, después recorta las observaciones ordenándolas y estableciendo un punto de corte, finalmente se actualizan los parámetros, estos pasos se repiten hasta cumplir con

el criterio de parada establecido, el tercer paso evalúa la función objetivo retornando el mejor valor obtenido.

También en 2017, en [70] investigaron acerca de la combinación de regresión lineal por grupos y K- means con una ponderación automática de pesos en las variables explicativas (WCLR) que tiene como objetivo proporcionar agrupaciones más homogéneas. Este método es capaz de seleccionar las variables explicativas importantes en el proceso de agrupamiento y la implementación del algoritmo tiene cuatro pasos: representación, ponderación, modelado y asignación, los cuáles buscan minimizar una función objetivo. La representación proporciona la solución óptima para el cálculo de los representantes de los conglomerados. La ponderación busca la solución óptima para el cálculo del vector de ponderaciones de relevancia de las variables. El modelado proporciona una solución óptima para el cálculo del vector de parámetros de los modelos de regresión y finalmente en la asignación se proporciona la solución óptima para la partición del clúster. Se realizó una comparación del método propuesto con 8 diferentes conjuntos de datos, entre ellos wine quality, glass y concrete tomados de repositorio de la UCI [71] y otras técnicas de CLR como KPLANE [72]. Como resultado, WCLR tuvo el mejor desempeño de manera significativa con respecto a las otras metodologías.

Siguiendo con el 2017, en [73] se presentó una metodología de regresión clusterwise con covariantes principales (PCCR). Para el funcionamiento de esta metodología se establece un número de grupos K y de pesos. El objetivo de PCCR es encontrar una matriz de ponderación W, una matriz de partición C, una matriz de carga PX y K vectores de ponderación de regresión, de modo que se minimiza la función de pérdida de mínimos cuadrados, con una restricción definida. El algoritmo encargado de obtener estos datos viene dado por 3 pasos; primero inicializa la matriz de partición C, segundo inicializa una matriz de carga PX, tercero actualiza consecutivamente las membresías del clúster de las unidades de nivel 2, dentro de este paso se actualiza la matriz de ponderación calculando los nuevos pesos, después se procede a recalcular los vectores condicionales y finalmente a verificar el valor de la función de pérdida para mejorar. Esta metodología fue comparada con una estrategia secuencial (es decir, primero extraer componentes de X; luego, predecir en función de estos componentes), dando como resultado que el enfoque propuesto es claramente superior en 4 puntos fundamentales, primero los componentes de PCCR revelan mejor los vínculos entre estos conjuntos de

variables, segundo una solución PCCR muestra mayores diferencias en los pesos de regresión específicos del clúster que la solución secuencial, tercero PCCR produce una agrupación más perspicaz, y finalmente cuando se realiza una validación cruzada de diez folders, la solución PCCR se generaliza mejor.

Continuando con el 2017, en [74] se usó la técnica CLR para la predicción de la lluvia mensual en Victoria, Australia. Este enfoque adoptó una formulación de optimización no convexa no uniforme de CLR propuesto previamente en [60] aplicando el algoritmo de CLR con técnicas de suavizado [67] para resolverlo. Este algoritmo divide gradualmente todo el conjunto de datos en subconjuntos. El algoritmo comienza con una función lineal y resume la estructura subyacente de los datos agregando dinámicamente una función lineal en cada iteración, para ello se siguen los siguientes pasos; primero se calcula una función de regresión lineal con todo el conjunto de datos, iniciando el grupo en j = 1, en el segundo paso toma el conjunto j = j + 1 y calcula un conjunto de soluciones iniciales para la j-ésima función lineal, tercero se calcula el conjunto de soluciones para el problema CLR auxiliar a partir de cada solución inicial, cuarto, se calcula el conjunto de soluciones para el problema de CLR utilizando soluciones para el del CLR auxiliar, quinto, se elige la solución con el valor más bajo de la función objetivo como la solución global aproximada al problema, finalmente, si j alcanza el límite establecido entonces se detiene, de lo contrario, regresa al segundo paso. Este estudio fue validado utilizando datos de lluvia con cinco variables meteorológicas de entrada durante el período de 1889 a 2014 de ocho estaciones meteorológicas geográficamente dispersas. El rendimiento de la predicción del método CLR se evaluó comparando los valores de lluvia observados y pronosticados utilizando cuatro medidas de precisión de pronóstico. El método propuesto también se comparó con el CLR utilizando el marco de máxima verosimilitud mediante el algoritmo de maximización de expectativas, regresión lineal múltiple, redes neuronales artificiales y las máquinas de soporte vectorial, y los resultados demuestran que el algoritmo propuesto supera a los otros métodos en la mayoría de las ubicaciones.

En el mismo 2017, en [75] se desarrolló un algoritmo para resolver el problema de regresión de desviaciones mínimas absolutas lineales por grupos, donde la función objetivo se representa como una diferencia de funciones convexas $fk\,(x, y) = fk1(x, y) - fk2(x, y)$. La propuesta plantea una optimización no suave buscando iterar uno a uno los valores de los pesos y realizando la diferencia entre funciones convexas

con el fin de minimizar el error; por cada iteración realizada se aumenta el número de grupos y se evalúa si mejoro o no con respecto a la versión anterior, el algoritmo propuesto se probó utilizando conjuntos de datos artificiales y del mundo real y arrojo como resultados que el esfuerzo computacional requerido no aumenta fuertemente dependiendo del tamaño de un conjunto de datos y puede obtener soluciones de buena calidad, pero en la mayoría de los casos, no hay garantía de que estas soluciones sean soluciones globales.

Otra propuesta en 2017 [76] usó CLR y las diferencias individuales en los efectos de la motivación académica en la intención de estudiantes de educación superior por abandonar sus estudios. El procedimiento de agrupamiento y estimación aplicado para este modelo se estima maximizando una función de probabilidad utilizando el algoritmo iterativo de maximización de expectativas (E-M) [13]. E-M consta de dos pasos, en el paso de estimación el algoritmo estima la probabilidad de que cada individuo pertenece a cada grupo y en el paso de maximización se estiman los coeficientes de regresión realizando K funciones de regresión. Estos pasos se repiten hasta que se alcanza algún criterio de convergencia. Cuando diferentes grupos están presentes en los datos, CLR ajusta una función de regresión lineal específica para cada grupo, junto con una estimación de las probabilidades de los conglomerados posteriores para cada individuo. Para agrupar a los individuos, cada uno de ellos se asigna al grupo utilizando la probabilidad más alta. Como se desconoce el número de grupos, se evalúan de uno a cinco grupos y se utiliza el Criterio Bayesiano de Información (BIC) [33] para determinar el modelo óptimo. En general, se escoge el modelo con el BIC más bajo. Adicionalmente para la evaluación se realizaron 1000 arranques aleatorios. El principal objetivo fue encontrar grupos de estudiantes que se diferencien entre sí en relación con las diferentes formas de motivación y la intención de abandonar sus estudios. Como resultado, se descubrió que los estudiantes diferían en cuatro tipos de motivación académica, que afectaban la intención de desertar sus estudios de educación superior.

También en 2017 [77] se utilizaron algoritmos de agrupación con regresión lineal generalizada (generación de columnas (CG) [78], GA adaptado de Lloyd´s [79] con K-means y un algoritmo modificado de [6]) sobre un conjunto de datos de una unidad de mantenimiento de stock (stock-keeping unit, SKU) de una cadena de supermercados. Entre los datos se encuentra productos, tamaño de producto, tipo

y fabricante. La comparación dio como resultado que CG puede resolver pequeñas instancias de manera óptima y es ligeramente mejor que GA Lloyd´s en tiempo de ejecución y en una métrica denomina OptGap, métrica que se deriva de SSE.

Además, en 2017 [80] proponen un CLR exhaustivo para un sistema de gestión de pavimentos que tiene como objetivo determinar un número óptimo de grupos de pavimento. Esta metodología establece membresías de las muestras de pavimento y determina variables explicativas significativas para cada grupo y los coeficientes de regresión estimados para cada modelo obtenido. Lo anterior fue soportado por Simulated Annealing (SA) obteniendo una combinación de subconjuntos con el fin de identificar la colinealidad. Este procedimiento obtuvo 6 modelos y para la evaluación de los modelos se utilizó un conjunto de datos con atributos como edad, promedio de tráfico diario, temperatura mínima, temperatura máxima, entre otras. Las muestras fueron tomadas en Nevada, Estados Unidos. Esta investigación arrojó como resultado un 74% de puntos dentro del ± 15% del valor de predicción y un RMSE de 0,47.

Terminado el 2017, en [81] se realizó un estudio comparativo sobre el uso de algoritmos metaheurísticos para la planificación del mantenimiento de carreteras desde la perspectiva de un país en desarrollo. El objetivo de los algoritmos fue minimizar el costo del ciclo de vida de una red de carreteras y maximizar al mismo tiempo la condición del pavimento, manejando dos enfoques, el primero con algoritmos mono objetivo: algoritmos genéticos (GA) [82], PSO [83], y la combinación de los dos (GAPSO), el segundo enfoque con el uso de algoritmos multi objetivo: NSGAII (Non-domination Sorting Genetic Algorithm II) [84] y MOPSO (multi-objective particle swarm optimization) [85]. El estudio se realizó con 8 secciones de pavimentos tomados de la red de transporte rural de la provincia de Khuzestan (Irán). Los resultados obtenidos por este estudio indican que el PSO proporciono un mejor rendimiento del pavimento con respecto a los otros algoritmos de un solo objetivo. Los algoritmos multi objetivo superaron a los algoritmos de un solo objetivo debido a que pueden equilibrar ambos objetivos. Finalmente, el algoritmo NSGAII tuvo un mejor rendimiento que MOPSO.

En 2018 [86] se investigaron tres diferentes métodos para la regresión difusa y robusta en conglomerados. Los tres métodos tienen en común un enfoque de máxima verosimilitud, recortes y restricciones. El primer método se deriva del modelado de la variable de respuesta y las variables explicativas a través de un

componente ajustado en cada grupo, el segundo es un enfoque difuso que tiene en cuenta las relaciones lineales subyacentes dentro de cada grupo con términos de error distribuidos y condicionados a las variables explicativas, el tercer enfoque también es difuso y utiliza pesos ponderados para cada uno de los grupos siendo de gran utilidad al momento de detectar valores atípicos en las variables explicativas que pueden actuar como puntos de “mala influencia”.

Continuando el 2018, en [87] se investigó un nuevo enfoque de micro grupos para regresiones de mínimos cuadrados parciales (PLS) en agrupaciones. Esta metodología la denominaron Micro-Batch Clusterwise Partial Least Squares (mb-CW-PLS) donde se combinan el PLS con dos niveles de agrupación (macro y micro) y enfoques de optimización de micro-grupo para encontrar la estructura subyacente de las observaciones y proporcionar a cada macro grupo de observaciones su propio conjunto de coeficientes de regresión. Este procedimiento se basa en la estrategia divide y vencerás donde los micro grupos se calculan utilizando k-means, definiendo el valor de k como la relación del tamaño del conjunto de datos al tamaño deseado de los micro grupos. Para realizar un cambio en un macro grupo se mueve todo el micro grupo y no un solo elemento, el cálculo de la calidad del macro grupo se realiza calculando la suma de los errores de los micro grupos que lo componen. La propuesta se comparó con PLS, OLS, regresión Ridge, LASSO y Random Forest con diferentes conjuntos de datos. La propuesta supera a estos métodos, en algunos casos por un margen sustancial, en términos de RMSE.

También en 2018 [88] se propone un enfoque de programación de diferencias convexas no uniformes para la regresión lineal por conglomerados. La función objetivo en este problema se representa como una diferencia de funciones convexas, buscando resolver eficientemente los problemas de CLR en grandes conjuntos de datos. Se propone un enfoque incremental para generar soluciones de partida con una buena calidad inicial. En este artículo se tuvieron en cuenta 20 conjuntos de datos pequeños y conjuntos de datos sintéticos, el objetivo fue demostrar el rendimiento del algoritmo en función del número de observaciones, atributos, agrupaciones y el nivel de perturbación. La cantidad de variables de entrada en estos conjuntos de datos varía de 5 a 11 y el número de datos de 1.030 a 45.730. La propuesta se comparó con el algoritmo NOBIA-CLR [67], dando como resultado que NOBIA-CLR produce soluciones más precisas que el algoritmo propuesto, sin embargo, la propuesta requiere un esfuerzo computacional

significativamente menor. La calidad de las soluciones en ambos algoritmos es muy similar.

Siguiendo el 2018, en [89] se planeó una combinación entre Modelos de Regresión basados en Fuzzy C-means (FCRM) con FCM con pesos automatizados (FWCLR), donde se realiza una combinación difusa de cada grupo junto con un procedimiento heurístico para ajustar automáticamente los hiperparámetros con el fin de obtener modelos que puedan ser mejores al realizar una predicción. Para lograr esto, se inicializa la matriz de membresía de manera aleatoria y luego se inicia un proceso iterativo comenzando con un prototipo de clúster, es decir se establecen un determinado grupo al cual se le definen unos pesos iniciales y sus coeficientes; a continuación, se establece la mejor combinación de pesos para este grupo. En el paso 3 se calculan los coeficientes de la regresión lineal del modelo, como paso 4 se calculan los valores de las membresías del grupo resultante. Finalmente, El algoritmo termina cuando la matriz de membresías no realiza cambios en sus valores. Este estudio se comparó con Multiple Linear Regression (MLR), CLR, KPLANE [72] modificado y Weighted Clusterwise Linear Regression (WCLR), utilizando diferentes conjuntos de datos entre ellos: Forest fires, glass, wine y red wine quality obtenidos desde el repositorio de UCI [71], en donde FWCLR obtuvo mejores o similares resultados que MLR, CLR, KPLANE y WCLR, comparándose con las medidas de MAE y RMSE.

Asimismo, en 2018 [90] se exploraron tres enfoques diferentes de CLR, el primero utilizó el algoritmo KPLANE [72] que combina CLR con k-means, el segundo utiliza pesos para los clústeres y obtiene un promedio final para todos los modelos de los clústeres (CLR-p) y el tercero utiliza CLR con restricciones de similitud (CLR-c). Para esta propuesta usaron 3 conjuntos de datos: Boston housing, abalone y autompg del repositorio de la UCI [71] y se compararon con la regresión de soporte vectorial (SVR) y el Random Forest (RF) tomando en cuenta las medidas de error cuadrático medio (MSE), el R^2 y el tiempo de ejecución. Los resultados mostraron que CLR-c resultara ser el más efectivo cuando se conocen las restricciones.

Otra investigación en 2018 [7] propone una regresión generalizada por conglomerados para la estimación de grupos óptimos de registros de pavimento y sus modelos de comportamiento. Para buscar una combinación óptima de observaciones, utilizan el algoritmo SA y para buscar los mejores parámetros de los modelos utilizan una regresión de todos los subconjuntos (ASR). Además,

identifican la colinealidad entre los parámetros. Proponen dos enfoques, uno lineal y otro no lineal, buscando reducir el error de las predicciones del PSI de un conjunto de datos de muestras tomadas en carreteras de Nevada, Estados Unidos. Entre los atributos recolectados se encuentra la edad, promedio diario de tráfico, promedio diario de camiones, elevación, entre otras, estas variables de tipo categóricas y continuas. Realizan las evaluaciones con 3.005 registros y obtienen como resultado que las predicciones de PSI del enfoque no lineal fueron más exactas, obteniendo un total de 5 grupos finales y un porcentaje de 81% de los puntos dentro del ± 15% del valor PSI real, lo que significa una mejor calidad que un enfoque lineal previo [81] donde se habían obtenido 6 grupos finales con un porcentaje de acierto de 74% de puntos dentro del ± 15% del valor PSI real. El valor de RMSE del modelo no lineal fue de 0,41 superando el trabajo previo en [80].

Otro trabajo de 2018 [9] utilizó generación simultánea de grupos óptimos de pavimento y modelos lineales de comportamiento, utilizando muestras de pavimento de Nevada, Estados Unidos. Este trabajo se enfocó, primero, en obtener un número óptimo de grupos, segundo, en asignar los segmentos al grupo más apropiado y tercero, en calcular el coeficiente de regresión para las variables reduciendo el error. Para llevarlo a cabo este trabajo se usó SA para la obtención de los vecinos de cada grupo y con ordinary least square (OLS) se buscaron los mejores modelos de regresión. El entrenamiento fue realizado con 14.637 registros y las evaluaciones de los modelos con 3.005 (17% del total de registros disponibles) registros de carreteras de Nevada (Estados Unidos). Con esta propuesta se superaron los resultados obtenidos en [80] pasando de obtener un RMSE de 0,47 a 0,44; sin embargo no se supera al enfoque no lineal de [7] que obtiene un valor de 0,41.

Terminando el 2018 [91] se propuso un método de agrupación con regresión multi bloque (estructura de bloques compuesta por las variables de la función y una cantidad desconocida de grupos) regularizada (CW.rMBREG). Esta propuesta abarca 2 métodos: clusterwise multiblock PLS (CW.MBPLS) y clusterwise multiblock redundancy analysis (CW.MBRA) [92]. Este trabajo combina la agrupación y un modelo basado en componentes (multi-bloque) asociado con un criterio bien definido para optimizar, es decir se predice un conjunto de datos y a partir de K conjuntos de datos explicativos, todas las variables cuantitativas se miden en las mismas N observaciones. Para darle el enfoque regularizado se imponen algunas restricciones para tratar con la estructura multi-bloque de los datos y para manejar

posibles matrices de bloques explicativos mal condicionados (es decir, cuando las variables superan en número a las observaciones dentro del grupo o en caso de multicolinealidad). Este trabajo se evaluó con una simulación y un ejemplo real tomando datos de la calidad de aire interior y se compararon con los métodos CW.MBRA y CW.MBPLS calculando la medida RMSE. Dando como resultado que CW.rMBREG obtuvo una mejor calidad de la predicción y facilitó la interpretación de datos complejos mal condicionados.

En 2019 [93] se propuso una metodología que obtiene modelos mediante la imposición de restricciones sobre las varianzas y covarianzas de las variables de respuesta en las subpoblaciones. Para esto, usaron métodos para el análisis de la regresión lineal multivariante basado en mezclas de regresiones gaussianas no relacionadas, lo que permite usar un vector diferente de regresores para cada variable dependiente. En este trabajo se utilizó el estimador ML con modelos de tamaños de muestras variables y niveles de superposición variables.

También en 2019 [94] se estudió un CLR para la estimación de la frecuencia de choques automovilísticos, donde todos los puntos de datos se agrupan en una sola función con características asociadas al sitio del accidente, se tiene como objetivo buscar grupos de parámetros para las funciones de estimación correspondiente y se utiliza SA para agrupar los datos y la estimación de máxima verosimilitud (MLE) para estimar los parámetros que componen las funciones de choque. Los datos utilizados en este estudio se extrajeron de Nevada Citation and Accident Tracking System (NCATS), Highway Performance Monitoring System (HPMS) y Traffic Records and Information Access (TRINA) del departamento de transporte de Nevada (Estados Unidos). Los datos contienen características de las carreteras, tráfico, intersecciones y accidentes recopiladas en el condado de Clark, la región más grande de Nevada. Los resultados de dos subtipos de sitios: segmentos divididos de varios carriles urbanos (SS1) e intersecciones señalizadas urbanas de cuatro vías (SS2), los resultados mostraron que las predicciones de CLR obtuvieron valores más bajos de RMSE que las predicciones de un clúster único (todo el dataset).

En 2020 [95] se evaluó el uso de recortes (eliminación de registros) y restricciones en dispersiones con el fin de agrupar datos utilizando regresión lineal. Proponen TCLUST-REG, una versión adaptativa del modelo de recorte restringido de

ponderación por grupos (TCWRM) que consiste en eliminar del conjunto de datos una fracción de las unidades de datos "más periféricas" para obtener un recorte con la menor variación posible. Adicionalmente, se restringe el grupo y se dispersa para reducir la posibilidad de soluciones falsas. Las pruebas se enfocaron en el proceso de generación de datos y se realizaron 5 casos de estudios, 4 de ellos tomados de la literatura, el caso restante basado en datos reales. Para cada caso se analizó la mediana del índice de rand ajustado, dando como resultado que TCLUST-REG resulta ser menos sensible a los datos distribuidos en las variables explicativas, además, que el mecanismo de generación de datos mejora considerablemente los resultados presentados en [91].

Otra investigación en 2020 [96] presentó dos estimadores de la matriz de covarianza del estimador ML realizando un análisis de la regresión lineal a través de un modelo de regresión gaussiano multivariado por agrupaciones. El primer estimador se basa en la matriz derivada de segundo orden de la verosimilitud de los datos; el otro explota un enfoque sándwich [101]. También desarrollaron expresiones analíticas para el vector gradiente y la matriz derivada de segundo orden, esto hace posible calcular las dos estimaciones de la matriz de covarianza asintótica del estimador ML. Además de una evaluación numérica de los rendimientos de los estimadores propuestos en muestras finitas y la comparación con el estimador paramétrico basado en bootstrap, generaron mezclas finitas de modelos de regresión gaussianos. La propuesta fue evaluada en pequeños conjuntos de datos simulados y reales, lo que arrojo como resultado que los estimadores son capaces de capturar la dependencia lineal entre las muestras y los efectos lineales de los predictores en las muestras de observaciones provenientes de poblaciones heterogéneas desconocidas.

Siguiendo en 2020, en [97] proponen una metodología de agrupación con regresión lineal soportada por vectores. El conjunto de datos se divide en k grupos (definido por el usuario) y cada grupo es aproximado por una regresión lineal y se diferencia del tradicional CLR, ya que tienen en consideración la estimación de varias funciones de regresión. Los errores de regresión se definen utilizando $L_1 - Risk$ [89] y para formulaciones CLR existentes se aplica $L_2 - Risk$ [88] de esta manera los valores son menos sensibles a los valores atípicos, adicionalmente implementa una modificación denominada double bundle method (DBDC) [98] como un método de búsqueda local para la optimización de la diferencia no suave de convexos (DC)

concepto presentado en [88] que utiliza explícitamente una descomposición de DC de la función objetivo para aprovechar tanto la convexidad como la concavidad. Esta investigación trabajó con 3 valores para ε (épsilon) en SVR, con 6 conjuntos de datos sintéticos y 5 conjuntos de datos tomados de un entorno real con el fin de medir el RMSE. Los resultados muestran que este enfoque tiene un buen rendimiento, pero no es capaz de identificar los llamados grupos "verticales" (agrupaciones de tendencia vertical que se puede observar de manera gráfica), no está claro qué tan eficiente es el algoritmo como herramienta de predicción, además, no es recomendable para problemas de CLR de gran escala o para la clasificación supervisada.

También en 2020 [99] se propuso un enfoque de agrupamiento restringido para identificar unidades de flujo hidráulico (HFU) en depósitos de petróleo, esto debido a un problema presentado en las unidades de flujo hidráulico en yacimientos. En la investigación se buscaron modelos con continuidad uniforme de HFU. Los análisis fueron realizados sobre el conjunto de datos UNISIM- ID (modelo de reservorio para el campo petrolífero Namorado, Cuenca de Campos, Brasil). En el trabajo se propone como algoritmo base de agrupamiento a Slope Constrained Clustering (SCC) y 6 variantes, 5 de ellas basadas con restricciones (SCCPC, SCCCT, SCCNL, SCCPY) y una variante llamada 6 Neighbour. La estructura del algoritmo base contiene un bucle en el cual va llenando un grupo donde cada nuevo elemento se asigna al grupo si y solo si cumple con la medida de restricción establecida para todos los elementos en el grupo, en el caso de que no se cumpla con las condiciones, se crea un nuevo grupo con las nuevas muestras, esta sería la primera fase del proceso de clusterwise la segunda fase del proceso consiste en definir el modelo de regresión para cada grupo.

Finalmente, en 2020 [100] se trabajó en un método para estimar parámetros de una distribución de probabilidad basado en la estimación de la máxima verosimilitud, sobre modelos CLR con un enfoque escalar restringido. Se proponen 4 componentes escalares que interactúan con unos límites establecidos y bajo cada componente escalar tiene definido un algoritmo, este enfoque utiliza un método de eliminación basado en probabilidad llamado k-deleted y representa la calidad con la que ajustan los datos después de eliminar un registro. Esto es comparado con RGD [101],un método gaussiano para mezclas multivariadas, y se evalúan bajo el criterio de información Bayesiano (BIC) [33]. En este estudio se prueban la cantidad de

grupos que se obtienen en varios escenarios, además de realizar pruebas en dos aplicaciones empíricas, se registran las medidas de media, mediana, error cuadrado medio (MSE) y el índice adj-Rand. El método propuesto obtuvo mejores resultados en comparación con el método RGD en términos de precisión de las estimaciones de los parámetros y tiempo de ejecución, pero la calidad de los grupos fue muy similar.

CAPÍTULO 3

3 Modelo Propuesto

3.1 Introducción

Esta investigación propone un algoritmo voraz que crea un modelo CLR con base en un conjunto de muestras de pavimentos tomados del departamento de transporte de Nevada en Estados Unidos mismo dataset utilizando en la propuesta de Khadka [7]. Este algoritmo es más efectivo que los presentados en el estado del arte (mayor calidad en la predicción y menor tiempo de construcción). El algoritmo permite definir grupos de tramos de carreteras con sus respectivos modelos de regresión lineal y los pesos de los atributos del dataset de entrenamiento para que en conjunto con el algoritmo 1-NN y la distancia euclidiana ponderada se facilite la asignación de un nuevo tramo de carretera al grupo adecuado y de esta forma predecir el valor de PSI con el modelo de regresión de dicho grupo.

3.2 Trabajo base

Como trabajo base se tomó la propuesta de Khadka [7] que cuenta con una base de datos con muestras de pavimentos, y en la cual se adaptó el algoritmo recocido simulado, que busca la cantidad óptima de grupos utilizando el criterio de información bayesiano (BIC) como función objetivo. Con el BIC buscan un equilibrio entre la bondad del ajuste y la complejidad de los modelos, adicionalmente para buscar los mejores parámetros de los modelos utilizan una regresión de todos los subconjuntos (ASR) e identifican la colinealidad entre los parámetros. Finalmente, se selecciona el mejor modelo al explorar todas las combinaciones de las variables de entrada y se selecciona el número óptimo de clústeres.

3.3 Conjunto de datos

El conjunto de datos en total cuenta con 17643 registros que fueron tomados de una base de datos del departamento de transporte de Nevada. Estos datos se dividen en dos conjuntos, el primero corresponde al conjunto de entrenamiento compuesto

por 14.638 registros y el segundo corresponde al conjunto de validación compuesto por 3.005 registros. Cada registro está compuesto por 15 variables o atributos, 9 son explicativas continuas (numéricas), 5 explicativas categóricas y la variable objetivo que tiene por nombre PSI y es continua. En la

Tabla 1 se presenta el nombre de cada variable, la descripción y el tipo de dato.

Tabla 1. Descripción de las variables/atributos de los conjuntos de datos

Variable/Atributo	Descripción	Tipo de variable
PSI	Representa la medida del estado actual del registro del segmento de carretera entre 0 y 5.	Objetivo (continua)
1. AADT	El promedio anual de tráfico diario (Annual average daily traffic) a la que está sometido un segmento de carretera en una de las direcciones.	Explicativa continua
2. TRUCKS	El tráfico promedio de camiones a la que está expuesto el segmento de carretera.	
3. ELEVATION	Es la elevación en el punto medio de un segmento en metros.	
4. PRECIP	Representa la precipitación media anual en cm.	
5. MIN_TEMP	Es la temperatura media anual mínima del aire en grados centígrados.	
6. MAX_TEMP	Es la temperatura media anual máxima del aire en grados centígrados.	
7. WET_DAYS	Es el número total de días húmedos en el transcurso del año.	
8. FREEZE_THAW	Es el número ciclos totales de congelación o descongelación que experimenta un pavimento en el transcurso del año.	
9. RUT_DEPTH	Representa la profundidad promedio de los huecos del registro de un segmento de carretera.	
10. AGE	Es la edad o época en la que del registro del segmento de carretera tuvo una intervención por reparación o mantenimiento.	Explicativa categórica (factor)
11. NUMBER_OF_LANES	Representa el número de carriles que tiene un segmento de carretera.	
12. SYS_ID	Clasifica si el segmento pertenece al Sistema Nacional de Carreteras (NHS), al Programa de Transporte de Superficie (STP) o es una Ruta Interestatal (IR).	
13. F_CLASS	Representa el uso del segmento de carretera: (1) vías Interestatales o vías primarias, (2) vías expresas o autopistas (3) vías Arteriales principales, (4) vías arterias menores o secundarias, (5) vías colectoras principales, (6) vías colectoras secundarias, (7) vías locales.	
14. CATEGORY	Representa la prioridad de un segmento de carretera, utilizando factores como el volumen del tráfico y la frecuencia de actividades de mantenimiento y reparación. Los valores varían entre un rango de 1 y 5 siendo 1 el valor con mayor prioridad y 5 el valor con menos prioridad.	

3.4 Algoritmo Greedy propuesto

La solución propuesta incluye varias tareas en dos tiempos. En el proceso de entrenamiento se realiza lo siguiente: 1) la definición del modelo CLR con un algoritmo voraz y 2) la definición de los pesos de los atributos con una adaptación del algoritmo GBHS. En el proceso de validación o en su uso en producción, a partir de un registro al que se le espera definir su valor de PSI se debe realizar lo siguiente: 1) buscar el grupo donde se encuentra el registro más similar basado en el algoritmo 1-NN y los pesos de los atributos previamente definidos en entrenamiento y 2) aplicar el modelo de regresión del grupo a los datos del registro para predecir su valor de PSI.

Para presentar el algoritmo voraz que realiza el modelo CLR usando el conjunto de datos de entrenamiento a continuación se hace una explicación de los pasos u operaciones de este y en paralelo se ilustran dichos pasos con un ejemplo. El algoritmo tiene como objetivo usar los 14.638 registros de muestras de segmentos de pavimentos del conjunto de entrenamiento para construir un número no predeterminado de grupos, y cada grupo cuenta con un modelo de regresión que permite predecir el PSI. Se busca que cada modelo en cada grupo cuente con un valor alto de correlación (R^2) [102] al predecir el PSI y cuente con un número mínimo y máximo de registros. El algoritmo está compuesto de dos fases que interactúan entre sí dependiendo de los resultados de cada una de ellas. El **Pseudocódigo 3** que se presenta más adelante muestra el diagrama general del algoritmo, pero primero se explican la Fase 1 y la Fase 2 para luego explicar la interacción entre estas fases.

3.4.1 Fase 1

Esta fase se muestra en el **Pseudocódigo 1** (El cálculo de su complejidad se explica en una sección posterior). Se reciben como variables de entrada, el dataset con los registros de entrenamiento, el mínimo valor de correlación que debe tener un grupo para ser aceptado, el error promedio que se acepta para incluir los registros en un grupo, el error adicional que se permite aceptar cada vez que se busca crear un nuevo grupo (este permite crear grupos que tienen un poco menos calidad que en la iteración anterior) y finalmente el mínimo número de registros que debe tener un grupo. En la línea 1 se inicializa la variable que va a almacenar los grupos calidad (GruposCalidad). En la línea 2 se define un bucle “Mientras” que termina cuando el

dataset de entrada queda vacío, ya que dentro de este bucle se busca mover los registros desde este dataset hacia un grupo de calidad.

Entradas: • **Dataset:** con los datos de entrenamiento. • **CoeficienteCorrelacionMinimo:** El valor mínimo del coeficiente de correlación que debe tener un grupo para ser aceptado. Por defecto igual a 0,88. • **ErrorPromedioAceptado:** Valor que determina el error promedio aceptado de los grupos permitiendo seleccionar los registros que más se ajustan al modelo de regresión los cuales serán incluidos en un grupo de calidad. • **ErrorPromedioAdicional:** Valor de tolerancia del error que se ajusta cada vez que se crea un siguiente grupo y se reduce la expectativa de calidad de los registros que se incluirán en este. • **MínimoRegistros:** Número mínimo de registros que debe tener un grupo para ser aceptado. Se recomienda un valor entre el 10% y 20% del tamaño del conjunto de datos de entrenamiento. **Salidas**: • **GruposCalidad:** Una lista de grupos aceptados, cada uno de ellos con su modelo regresión. • **Huérfanos:** Registros que no fueron aceptados en ningún grupo de calidad.	**Complejidad** $\boldsymbol{O\left(p^2n + p^3 + nlog(n)\right)}$
Inicio	
1. GruposCalidad = Ø;	$O(1)$
2. **Mientras** Dataset.Tamaño != 0 **Hacer**	$O(c); c<<n$
3. Modelo = CalcularModeloRegresionLineal (Dataset)	$O(p^2n + p^3)$
4. MejoresRegistros = OrdenarDatosPorErrorDePrediccion (Dataset, Modelo, ErrorPromedioAceptado, MínimoRegistros)	$O(nlog(n))$
5. Grupo = CrearGrupo (Dataset, MejoresRegistros)	$O(n)$
6. ModeloGrupo = CalcularModeloRegresionLineal (Grupo)	$O(p^2n + p^3)$
7. **Si** ModeloGrupo.CoeficienteCorrelacion > CoeficienteCorrelacionMinimo **entonces**	$O(1)$
8. GruposCalidad.Adicionar (Grupo, ModeloGrupo)	$O(1)$
9. Dataset = Borrar (Dataset, Grupo.Registros)	$O(n)$
10. ErrorPromedioAceptado = ErrorPromedioAceptado + ErrorPromedioAdicional	$O(1)$
11. **Si no**	
12. Huérfanos = DataSet.ObtenerRegistros ()	$O(h)$
13. **Romper Mientras**	
14. **Fin Si**	
15. **Fin Mientras**	
16. **Retornar** GruposCalidad, Huérfanos	
Fin	

Pseudocódigo 1. Fase 1 del algoritmo propuesto para el entrenamiento.

En la línea 3 se genera un modelo de regresión lineal con los registros existentes en el conjunto de datos utilizando la función LinearRegression de Weka con los valores por defecto ("-S 0 -R 1.0E-8 -num-decimal-places 4"). Luego en la línea 4

se realiza el método OrdenarDatosPorErrorDePrediccion, este método primero revisa el tamaño del conjunto de datos de entrenamiento, si este tamaño no supera el mínimo número de registros (MínimoRegistros) termina retornando el conjunto de datos como lo recibió; de lo contrario se ejecutan las siguientes cuatro tareas principales, primero, predice el PSI para todos los registros del conjunto de datos, segundo, calcula por cada registro el error entre el valor predicho y el valor real, tercero, ordena los registros del conjunto de datos del que tiene menor error al que tiene mayor error y cuarto, selecciona los registros que en su error de predicción no superan el parámetro ErrorPromedioAceptado; si la cantidad de esos registros es menor al mínimo número de registros entonces lo completa con otros registros que cuenten con el menor error posible sin importar que superen el ErrorPromedioAceptado, al final este método retorna esos registros como los MejoresRegistros, es decir, aquellos que se ajustan mucho mejor al modelo previamente realizado.

Con estos MejoresRegistros, en la línea 5 se crea un grupo y en la línea 6 se crea un modelo de regresión para este grupo de registros. En la línea 7 se pregunta si el grupo que se creó tiene la calidad requerida, es decir, tiene un coeficiente de correlación superior al parámetro CoeficienteCorrelacionMinimo, en caso de que sea cierto (verdadero) se adiciona este grupo junto con su modelo a la lista de GruposCalidad (línea 8), se remueven dichos registros del conjunto de datos de entrenamiento (línea 9) y en la línea 10 se suma el valor de ErrorPromedioAdicional al ErrorPromedioAceptado para permitir que en la siguiente iteración se puedan incluir registros a un grupo de calidad con un poco más de error (un poco menor calidad) que en la iteración actual. En el caso de que el grupo que se creó no tenga la calidad requerida (línea 11), en la línea 12 los registros del conjunto de datos se pasan a la lista de huérfanos y se rompe el ciclo “mientras” (línea 13) que inicio en la línea 2. Este método finalmente retorna los grupos de calidad y los registros huérfanos que corresponde a registros que no se ingresaron a ningún grupo de calidad (línea 16).

En la **Figura 1** se puede observar como el conjunto de datos se procesa en cada una de las iteraciones (de izquierda a derecha) de la Fase 1. Inicialmente, se tiene el conjunto de datos de entrenamiento completo (14638 registros) y en la iteración 1 se obtiene el primer grupo representado en un bloque de color celeste, que se guarda en la lista de grupos de calidad (GruposCalidad) que se observa en la parte

superior derecha de la imagen, este proceso se repite hasta la iteración 3 donde se aprecia que se agrega el último grupo de calidad (quedando 3 grupos de calidad) y el dataset que se está procesando tiene en total 7713 registros. Es preciso observar que los grupos de calidad tienen valores altos en su coeficiente de correlación (CC) aunque en cada iteración este valor decrece. En la iteración 4, al crear el modelo de regresión y armar el grupo con los mejores registros, ese grupo (bloque con fondo en color naranja y letras en rojo) ya no supera el criterio de calidad basado en el parámetro CoeficienteCorrelacionMinimo y como consecuencia todos los registros del dataset en ese momento se mueven a un dataset de huérfanos y el dataset original queda vacío, con esto se termina la Fase 1.

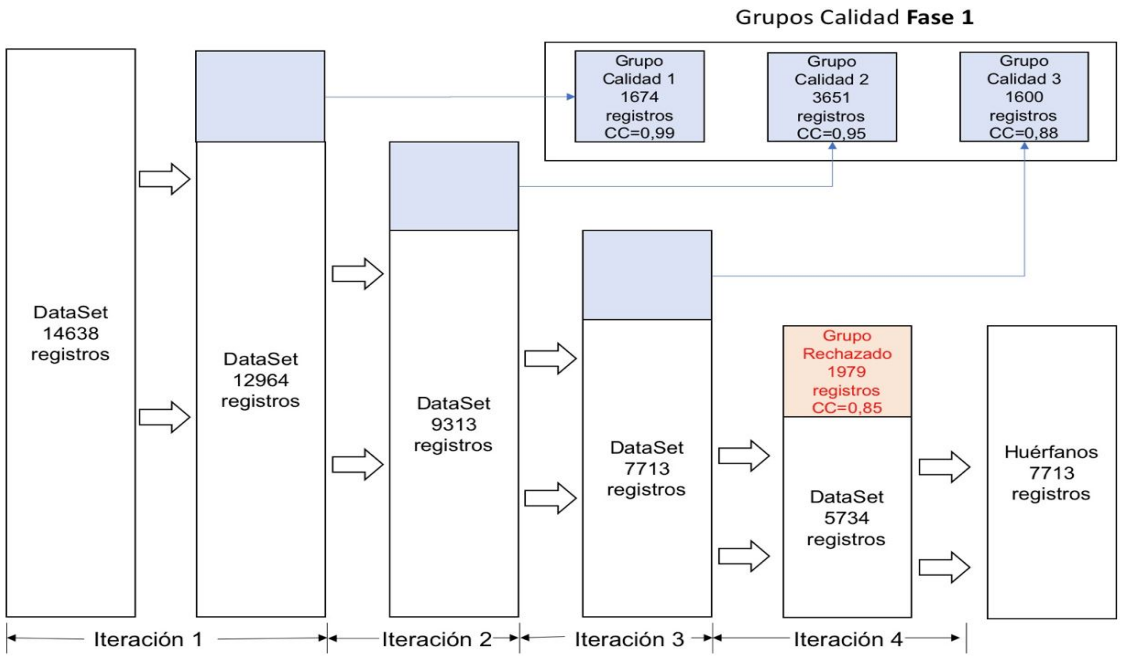

Figura 1. Representación del comportamiento de la Fase 1 con datos reales.

3.4.2 Fase 2

En el **Pseudocódigo 2** se puede observar la lógica de esta fase donde se distribuyen los registros del dataset de huérfanos a los grupos de calidad resultantes de la Fase 1. Como entradas de esta fase se reciben los grupos de calidad o GruposCalidad (cada grupo con su modelo de predicción de PSI) y el dataset con los registros huérfanos, es decir, aquellos que todavía no pertenecen a un grupo. Esta fase se inicia con un ciclo “para” que toma uno a uno los registros huérfanos (línea 1) y busca en cuál de los grupos de calidad encaja mejor, esto se hace con el

método BuscarMejorGrupo (línea 2). Dentro de este método se incluye temporalmente cada registro huérfano en cada grupo de calidad y se evalúa como varía el coeficiente de correlación, aquel que tenga el mejor impacto (porque mejora su calidad más que los demás o porque es el que menos pierde calidad en comparación con los otros) se considera como el mejor grupo (MejorGrupo). Teniendo el registro huérfano y el MejorGrupo se solicita al método IncluirRegistroEnGrupo (línea 3) que agregue el registro en el grupo y que actualice formalmente el modelo de predicción del grupo con el registro ya incluido. Este procedimiento termina cuando todos los registros huérfanos dejan de serlo y quedan asignados a un grupo de calidad.

Entradas: • **GruposCalidad:** Lista con los grupos de calidad (grupos con sus propios modelos de predicción) generados en la Fase 1. • **Huérfanos:** Lista con los registros que no fueron aceptados en ningún grupo de calidad en la Fase 1. **Salidas**: • **GruposCalidad:** Lista con los grupos de calidad modificados en esta fase. Cada grupo con su modelo de predicción.	**Complejidad** $\boldsymbol{O(h * GC * (p^2 n + p^3))}$
Inicio 1. **Para** i = ∅ **Hasta** Huérfanos.Tamaño **Hacer** 2. MejorGrupo, ModeloGrupo = BuscarMejorGrupo (GruposCalidad, Huérfanos[i]) 3. IncluirRegistroEnGrupo (GruposCalidad [MejorGrupo], Huérfanos[i]) 4. **Fin Para** 5. **Retornar** GruposCalidad **Fin**	$O(h); h<<n$ $O(GC * (p^2 n + p^3)); GC<<n$ $O(1)$

Pseudocódigo 2. Fase 2 del algoritmo propuesto para el entrenamiento

La **Figura 2** muestra tres (3) grupos de calidad con sus modelos (bloques superiores en color verde y celeste enmarcados en cuadros con borde negro) luego, muestra el resultado de incluir a modo de prueba (temporalmente) un registro huérfano a los 3 grupos y sus nuevos modelos. En este caso los cambios son muy pequeños y los bloques verdes que dicen “Nuevo Modelo” no alcanzan a mostrar dicha variación y por eso su CC (coeficiente de correlación) parece igual a los modelos originales. Más abajo en la figura se pueden ver las variaciones (-0.00001, -0.000002 y +0.000002) para cada uno de los tres modelos. De estos valores se selecciona el mejor grupo, que en este caso es el grupo de calidad 3 porque su calidad (CC) mejora en 0.000002. Luego este registro se incluye al grupo de calidad 3 y se actualiza el modelo, es decir, el Nuevo Modelo remplaza al anterior. Este proceso se repite para todos los registros huérfanos.

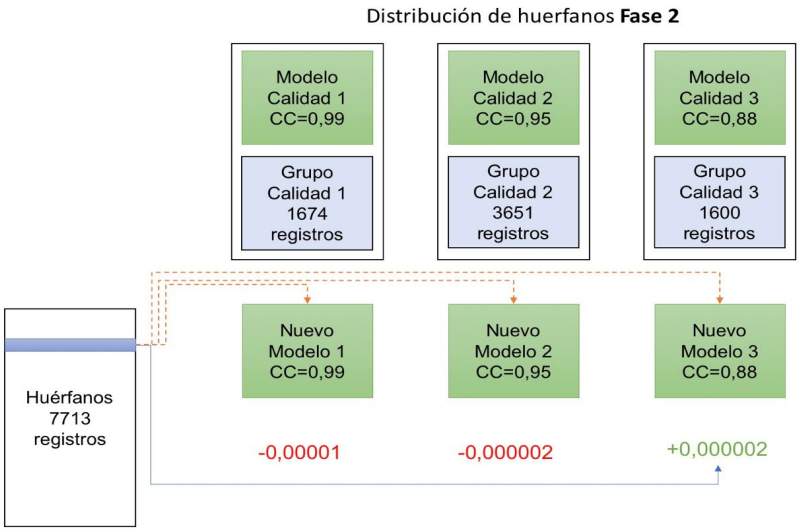

Figura 2. Asignación del grupo de calidad a un registro huérfano en la Fase 2.

La **Figura 3** muestra los grupos de calidad después de asignar todos los registros huérfanos; se puede observar que en algunos grupos el coeficiente de correlación bajo y que en todos, la cantidad de registros aumentó. Al sumar la cantidad de registros de todos los grupos el resultado es 14638, que corresponde al número total de registros del conjunto de datos de entrenamiento. Es preciso mencionar que el número de registros huérfanos queda reducido a 0 al terminar la Fase 2.

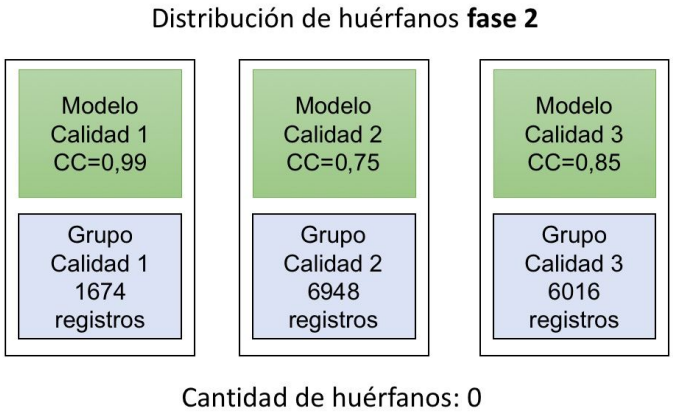

Figura 3. Grupos de calidad después de finalizada la Fase 2 con datos reales.

3.4.3 Algoritmo principal

El algoritmo voraz (greedy) propuesto, utiliza conjuntamente la Fase 1 y la Fase 2 como se puede ver en el **Pseudocódigo 3**. El algoritmo tiene como entradas los nombres de los archivos (en formato ARFF, nativo e Weka) con los datos de entrenamiento y validación, el CoeficienteCorrelacionMinimo explicado previamente en la Fase 1, el número mínimo y máximo de registros (MínimoRegistro, MáximoRegistros) que se espera albergue cada grupo, el ErrorPromedioAceptado y el ErrorPromedioAdicional que también fueron explicados en la Fase 1; además los parámetros del algoritmo GBHS (ParametrosGBHS), que contiene el tamaño de la memoria armónica y el número de iteraciones que ejecutará el algoritmo GBHS para definir los pesos de los atributos.

En la línea 1 se cargan los conjuntos de datos de entrenamiento y validación, después en la línea 2 se realiza la Fase 1 que utiliza el conjunto de datos de entrenamiento, el error promedio aceptado, el error promedio adicional y el mínimo número de registros en los grupos, conforme se explicó anteriormente, y retorna los grupos de calidad (GruposCalidad) y los registros huérfanos. En la línea 3 se valida que la cantidad de grupos resultantes entregados por la fase 1 sea mayor a 1, en caso contrario el grupo resultante pasa a ser denominado grupo definitivo como se puede ver en la línea 24 (significa que es mejor no agrupar el dataset con el enfoque propuesto, por esto, se debe tomar el modelo con el dataset como un todo y asignar pesos iguales a todos los atributos - línea 25 -, ya que no tiene sentido calcularlos, puesto que no se cuenta con la división de los datos en mínimo dos grupos). En caso de que, si se cuente con más de 1 grupo, en la línea 4 se valida si la cantidad de registros huérfanos es 0; en tal caso los grupos de calidad (GruposCalidad) resultantes de la fase 1 se asignan como los grupos definitivos (GruposDefinitivos). Esta variable de GruposDefinitivos es una de las salidas que el algoritmo retornará.

En la línea 5 comienza un proceso iterativo que se realiza un número no determinado de veces, ya que se ejecuta hasta que el número de registros huérfanos sea igual a 0. Después, en la línea 6 se procede a distribuir los registros huérfanos ejecutando la Fase 2 descrita anteriormente: esta fase retorna los grupos de calidad (GruposCalidad) pero con los registros huérfanos distribuidos dentro de ellos. A continuación, en la línea 7 se inicializa vacía la variable para almacenar los grupos definitivos (GruposDefinitvos) y en la línea 8 se inicializa vacía la variable que almacenará los nuevos registros huérfanos (NuevosHuerfanos) que se

obtendrán de la subdivisión de los GruposCalidad cuando estos tengan más registros que los definidos por el parámetro MáximoRegistros.

Entradas: • **NombreDataset:** Nombre del archivo con los datos de entrenamiento. • **NombreDatasetValidación:** Nombre del archivo con los datos de validación. • **CoeficienteCorrelacionMinimo:** El valor mínimo del coeficiente de correlación que debe tener un grupo para ser aceptado. Por defecto igual a 0,88. • **MínimoRegistros:** Número mínimo de registros que debe tener un grupo para ser aceptado. Por defecto igual a 1600 para el conjunto de datos de entrenamiento de pavimentos según una restricción definida en la formulación del problema (Khadka, Paz, Arteaga, et al., 2018). • **MáximoRegistros:** Número máximo de registros que debe tener un grupo definitivo para ser aceptado, el valor puede ser modificado por el usuario y por defecto es 3000. • **ErrorPromedioAceptado:** Valor establecido que determina el error promedio aceptado de los grupos que permite seleccionar los registros que más se ajustan al modelo de regresión lineal y serán incluidas en un grupo de mayor calidad. • **ErrorPromedioAdicional:** Valor de tolerancia del error que se ajusta cada vez que se crea un siguiente grupo y se reduce la expectativa de calidad de los registros que se incluirán en este. • **ParametrosGBHS**: Lista de parámetros relacionados con el algoritmo GBHS que realiza la definición de los pesos o ponderaciones de los atributos. Se detallan más adelante. **Salidas**: • **GruposDefinitivos:** Una lista de grupos finales, cada uno de ellos con su modelo regresión.	**Complejidad**
Inicio	
1. Dataset, DatasetValidacion = CargarDatos (NombreDataset, NombreDatasetValidacion)	$O\ (n)$
2. GruposCalidad, Huérfanos = Fase1 (Dataset, ErrorPromedioAceptado, ErrorPromedioAdicional, MínimoRegistros)	$O\left(p^2n\ +\ p^3\ + nlog(n)\right)$
3. **Si** GruposCalidad.Tamaño > 1 **entonces**	$O\ (1)$
4. **Si** Huérfanos.Tamaño == Ø **entonces** GruposDefinitivos = GruposCalidad	$O\ (1)$
5. **Mientras** Huérfanos.Tamaño != Ø **Hacer**	$O\ (c);\ c<<n$
6. GruposCalidad = Fase2 (GruposCalidad, Huérfanos)	$O(h * GC * (p^2n + p^3)); h < n$
7. GruposDefinitivos = Ø;	$O\ (1)$
8. NuevosHuérfanos = Ø	$O\ (1)$
9. **Para** i = 0 **Hasta** GruposCalidad.Tamaño **Hacer**	$O\ (GC)$
10. **Si** GruposCalidad[i] > MáximoRegistros **entonces**	$O\ (1)$
11. DatasetGrupo = GruposCalidad[i].ObtenerDatos ()	$O\ (n)$
12. GruposCalidad2, HuérfanosGrupo = Fase1 (DatasetGrupo, ErrorPromedioAceptado, ErrorPromedioAdicional, MínimoRegistros)	$O(GC * (p^2n\ + p^3 + n\ * log(n)))$
13. NuevosHuérfanos.Adicionar (HuérfanosGrupo)	$O\ (1)$
14. GruposDefinitivos.Adicionar (GruposCalidad2)	$O\ (1)$
15. **Si no**	
16. GruposDefinitivos.Adicionar (GruposCalidad[i])	$O\ (1)$
17. **Fin Si**	
18. **Fin Para**	
19. Huérfanos = Copiar (NuevosHuérfanos)	$O\ (h);\ h<<n$
20. GruposCalidad = Copiar (GruposDefinitivos)	$O\ (GD)$
21. **Fin Mientras**	
22. Pesos = GBHS (ParametrosGBHS)	$O\ ((HMS + NI)\ *\ n^2 * p)$
23. **Si no**	
24. GruposDefinitivos = Huérfanos	$O\ (1)$

25. Pesos = PesosIguales ()	$O\ (p)$
26. **Fin Si**	
27. **Retornar** GruposDefinitivos, Pesos	$O\ (1)$
Fin	

Pseudocódigo 3 Algoritmo principal.

En la línea 9 se realiza un ciclo que recorre todos los grupos de calidad y verifica (línea 10) que cumplan con el número máximo de registros (MaximoRegistros). Esto se hace buscando que uno o más grupos no acaparen la mayoría de los registros en el dataset de entrenamiento (Si el usuario no quiere realizar este control puede colocar un número grande o en su defecto el número de registros del dataset de entrenamiento). En el caso de que el grupo no supere el valor de MaximoRegistros (línea 15) entonces este grupo de calidad se agrega a la lista de GruposDefinitivos (línea 16). Si el grupo si supera el valor de MaximoRegistros entonces los registros de este grupo se copian en una variable (DatasetGrupo) (línea 11) para que en la línea 12 se aplique la Fase 1 (dividiendo este grupo en otros grupos de menor tamaño con buena calidad y en algunos huérfanos). Este proceso entrega nuevamente los grupos de calidad (GrupoCalidad2) con sus modelos de regresión y los registros huérfanos (HuérfanosGrupo); y estos datos se guardan en las variables de NuevosHuérfanos y GruposDefinitivos (líneas 13 y 14). Después que se finalice el ciclo “para” y se recorran todos los grupos de calidad, se actualizan los huérfanos como los nuevos huérfanos del proceso de mejora de los grupos. Además, los grupos de calidad para la siguiente iteración se toman como una copia de los GruposDefinitivos que van hasta el momento (línea 20). Al regresar a la línea 5 si hay registros huérfanos se repite el proceso de la línea 6 a la 20. Finalmente, cuando el número de registros huérfanos sea 0 se sale del ciclo “mientras” y se procede a hacer el proceso de definición de los pesos de los atributos (línea 22). Estos pesos serán utilizados junto con los GruposDefinitivos (Grupos de registros con sus modelos de predicción) en el proceso de evaluación o en producción.

3.4.4 Definición de pesos con GBHS

Esta tarea es relevante por dos situaciones importantes: 1) Los datos de las columnas (atributos) no se han normalizado (por ejemplo en el rango entre 0 y 1 para los atributos continuos), ya que los modelos de predicción basados en regresión lineal multivariable no requieren este preprocesamiento, pero al usar 1NN para definir a qué grupo debería ser asignado un registro nuevo este proceso si es requerido y 2) Los grupos que se obtienen con el algoritmo voraz que crea el modelo

CLR se centran en que los modelos de predicción se ajusten a los datos de cada grupo y no a que los elementos de un grupo sean similares entre sí, como si lo hace un algoritmo de clustering tradicional. Es por esto por lo que se hace necesario definir la forma como un nuevo registro (de evaluación o de producción) debe ser asignado a un grupo y luego con el modelo de dicho grupo predecir su PSI. Para cumplir con esta tarea se usa el algoritmo 1NN basado en la distancia euclidiana ponderada [103]. Para definir los pesos o ponderaciones de los atributos (tarea que permite reducir el efecto de ponderación que da el rango natural de los atributos en el dataset) se usa la metaheurística de la Mejor Búsqueda Armónica Global (Global-best Harmony Search, GBHS).

GBHS se basa en la imitación del proceso de improvisación musical, donde los músicos de Jazz improvisan los tonos de sus instrumentos en busca de un estado perfecto de armonía [104]. Esta metaheurística además incorpora el concepto de inteligencia de enjambre para mejorar la calidad de los resultados de su proceso de búsqueda. GBHS tiene una estructura base que se conoce como la memoria armónica (Harmony Memory, HM) que es similar a la población de un algoritmo genético, y tiene dos tareas fundamentales: 1) La inicialización de la memoria armónica (en algoritmos genéticos corresponde a la inicialización de la población) y 2) el proceso de evolución, que se hace con un enfoque estacionario donde la principal operación se conoce como la improvisación (en algoritmos genéticos corresponde a la generación de un descendiente) que crea una nueva solución con características de múltiples padres y algunas características pueden ser generadas aleatoriamente en el espacio de búsqueda factible; esto última como si fuera una mutación.

GBHS requiere que el usuario defina ciertos parámetros previo a su ejecución, estos son: 1) El tamaño de la memoria armónica (Harmony Memory Size, HMS) que en algoritmos genéticos corresponde al tamaño de la población; 2) el número de improvisaciones (Number of Improvisations, NI) que define el máximo número de nuevas soluciones o descendientes que se generaran en el proceso de búsqueda; 3) La tasa de consideración de la memoria armónica (Harmony Memory Consideration Rate, HMCR) que en el proceso de improvisación (creación de una nueva solución o descendiente) define la probabilidad de tomar los valores de los pesos de las soluciones ya existentes en la memoria armónica; 4) Los parámetros ParMin (Pitch Adjustment Rate Minimum) y ParMax (Pitch Adjustment Rate

Maximum) que se usan para definir la probabilidad de tomar el valor de un peso perteneciente a la mejor armonía (solución) de la memoria armónica [105], este parámetro es clave para darle la capacidad a la metaheurística de comportarse como un enjambre; 5) Los grupos de calidad obtenidos en el modelo CLR por el algoritmo voraz y el número de atributos (parámetro p) independientes en el dataset (todos excepto PSI). GBHS retorna los pesos de la mejor solución, esto es, los pesos que permiten que la mayor cantidad de registros obtengan un PSI predicho que está en el rango del PSI real ±15% del PSI real, es decir el valor predicho se parece mucho al valor real.

El **Pseudocódigo 4** resume las tareas que hace la adaptación de GBHS para definir los pesos de los atributos. Las líneas 1 a 5 (ciclo "para") se encargan de generar la memoria armónica, esto es, un conjunto de HMS (este parámetro se recomienda como un número entero entre 5 y 20) soluciones o armonías, donde cada una de estas armonías está compuesta por un vector (índice base cero) de tamaño p con los pesos para cada atributo del dataset y un valor de Fitness o calidad de esos pesos en relación con el dataset de entrenamiento. En la línea 2 el método GenerarPesosAleatorios se encarga de generar el vector de pesos asignando en cada celda o casilla del vector un número aleatorio entre cero y uno. Como estos pesos al final deben sumar uno, se deben normalizar, para esto se suman todos los valores inicialmente generados aleatoriamente entre 0 y 1 de las casillas del vector de pesos y luego cada valor de cada casilla se divide por esa suma. La **Figura 4** muestra un ejemplo de creación de pesos aleatorios para una armonía. En la parte superior se muestran los números generados aleatoriamente entre cero y uno sin estar normalizados en su conjunto. Luego en la parte superior derecha se muestra la suma de estos valores y en la parte inferior se muestra como los pesos se dividen por esa suma para dejar los pesos aleatorios normalizados, es decir, sumando uno.

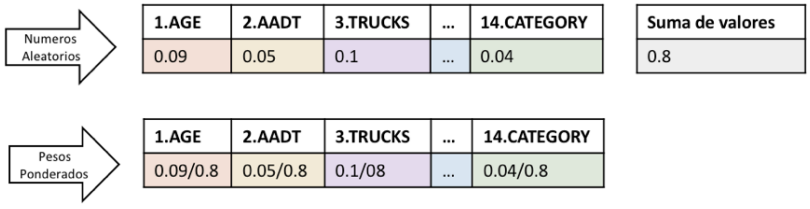

Figura 4. Representación de la distancia ponderada con los atributos del dataset.

Entradas: • **HMS**: Tamaño de la memoria armónica • **NI:** Número de improvisaciones que realiza GBHS. • **HCMR:** Tasa de consideración de memoria armónica. • **ParMin:** Tasa de ajuste de tono mínima**.** • **ParMax:** Tasa de ajuste de tono máxima. • **GruposCalidad**: Grupos de calidad reportados por el algoritmo voraz • **p:** Número de atributos en el dataset de entrenamiento.	**Complejidad** $\boldsymbol{O}\ ((\boldsymbol{HMS} + \boldsymbol{NI}) * \boldsymbol{n^2} * \boldsymbol{p})$
Inicio	
// Inicialización de la Memoria Armónica	
1. **Para** i = 1 **Hasta** HMS **Hacer**	$O\ (HMS)$
2. Pesos = GenerarPesosAleatorios ()	$O\ (p)$
3. Fitness = EvaluarCalidadEnModelo (Pesos)	$O\ (n^2 * p)$
4. HM = AdicionarArmonia (Pesos, Fitness)	$O\ (1)$
5. **Fin Para**	
6. HM = OrdenarMemoriaArmonica **(**HM**)** //De mejor a peor armonía	$O\ (HMS * \log HMS)$
// Evolución estacionaria con un descendiente por iteración	
7. **Para** i = 1 **Hasta** NI **Hacer**	$O\ (NI)$
// Improvisación	
8. Pesos = vector de p pesos en cero	$O\ (p)$
9. PAR = ParMin + (ParMax-ParMin) * (i / NI) // Tasa de ajuste de tono	$O\ (1)$
10. **Para** j = 0 **Hasta** p - 1 **Hacer** // número de pesos	$O\ (p)$
11. **Si** r1 < HMCR **Hacer** // r1 es un número aleatorio entre 0 y 1	$O\ (1)$
12. Pesos [j]= HM [r2, j] // Toma el valor de una armonía en HM // r2 es un número entero aleatorio entre 0 y HMS - 1	$O\ (1)$
13. **Si** r3 < PAR **Hacer** // r3 es un número aleatorio entre 0 y 1	$O\ (1)$
14. Pesos [j]= HM [0, j] // Toma el valor de la mejor armonía	$O\ (1)$
15. **Fin Si**	
16. **Si no**	
17. Pesos [j] = r4 factible // Valor aleatorio en el rango del atributo	$O\ (1)$
18. **Fin Si**	
19. **Fin Para**	
20. Pesos = Normalizar (Pesos)	$O\ (p)$
21. Fitness = EvaluarCalidadEnModelo (Pesos)	$O\ (n^2 * p)$
// Remplazo	
22. **Si** HM[HMS].Fitness **es peor que** Fitness **Hacer**	$O\ (1)$
23. HM[HMS] = (Pesos, Fitness)	$O\ (1)$
24. HM = OrdenarMemoriaArmonica (HM)	$O\ (HMS * \log HMS)$
25. **Fin Si**	$O\ (1)$
26. **Fin Para**	
27. **Retornar** HarmonyMemory [0].Pesos	
Fin	

Pseudocódigo 4 Algoritmo para definición de pesos basado en GBHS.

En la línea 3 y más adelante en la línea 21 se calcula la calidad de los pesos en relación con el modelo CLR generado hasta el momento con el algoritmo voraz (método EvaluarCalidadEnModelo). Evaluar la calidad implica tomar uno a uno los registros del dataset de entrenamiento como registro de prueba, todos los registros

del dataset en este momento están repartidos en los Grupos de Calidad del modelo CLR, luego para cada registro de prueba se le busca cuál es el registro más parecido (algoritmo 1NN) en todos los grupos de calidad, sin comprarse con el mismo, y cuando se encuentra ese otro registro más parecido, se aplica el modelo de predicción de ese grupo para obtener el PSI predicho. El registro más parecido se calcula usando la distancia euclidiana ponderada usando los pesos de la actual solución (variable Pesos). Usando el valor del PSI predicho y el valor del PSI real de cada registro se define si el PSI predicho está en el rango del PSI real ±15% del PSI real, si esto es así se cuenta como dentro del rango y al final se cuentan cuantos registros están en el rango y se divide en el número total de registros y se multiplica por 100 para obtener un porcentaje. Este porcentaje se convierte en el Fitness de la armonía.

Ya contando con los pesos y su calidad (Fitness) esta nueva solución o armonía se agrega a la memoria armónica (línea 4). Repitiendo este proceso HMS veces se logra la inicialización de la memoria armónica (líneas 1 a 5). Luego, la memoria armónica se ordena de mayor a menor con base en el valor de Fitness (línea 6), logrando con esto que la mejor solución quede en la primera posición del vector (posición cero usando vectores con índice base cero) y la peor solución (la de menor calidad) en la posición HMS - 1. La mejor solución es la que tiene mayor Fitness (aptitud o calidad) porque entre más cantidad de predicciones se hagan dentro del ±15% del valor real de cada tramo de carretera, mejor se ajustan los pesos. La **Figura 5** muestra un ejemplo de una memoria armónica compuesta por 10 armonías.

La segunda parte del algoritmo realiza la mejora de las soluciones encontradas hasta el momento. Como se mencionó previamente, la improvisación es la operación más importante de GBHS, pero luego de que se crea una nueva solución con esta operación se debe decidir si esta armonía debe ingresar a la memoria armónica o si se debe descartar (operación de remplazo), y si esta nueva armonía debe entrar a la memoria, se debe decidir cuál otra armonía debe salir de esta (en este caso la peor solución), para mantener así el tamaño de la memoria armónica constante.

0	1.AGE	2.AADT	...	14.Category	± 15 %
	0.09/0.8	0.05/0.8	...	0.04/0.8	90.1
1	1.AGE	2.AADT	...	14.Category	± 15 %
	0.21/0.8	0.75/0.8	...	0.01/0.8	88.8
...	1.AGE	2.AADT	...	14.Category	± 15 %
	...	...	...	...	...
9	1.AGE	2.AADT	...	14.Category	± 15 %
	0.99/0.8	0.25/0.8	...	0.24/0.8	80.3

Figura 5. Representación de la memoria armónica.

El bucle comprendido en las líneas 7 a 26 corresponde al número de improvisaciones que se realizarán en el proceso de mejora de la memoria armónica y búsqueda de la mejor solución posible. La operación de improvisación se realiza desde la línea 8 hasta la 20. En la línea 8 se crea el vector de Pesos con los p campos inicializados en cero. Luego en la línea 9 se calcula el valor de la tasa de ajuste de tono (PAR) de acuerdo con los parameros ParMin, ParMax, NI y el número actual i de la improvisación. El valor de PAR inicia con el valor ParMin y termina con el valor ParMax en un incremento lineal, esto busca que al principio el algoritmo tenga en cuenta con muy baja probabilidad los valores de los pesos de la mejor solución encontrada, pero en la medida que aumentan las iteraciones de búsqueda, esta probabilidad aumente y de esta forma las nuevas armonías que se improvisen se creen con más pesos de la mejor solución (armonía) encontrada hasta el momento en cada iteración (comportamiento de enjambre).

La operación de improvisación aplica tres reglas (líneas 11 a 18). Estas reglas se aplican a cada uno de los pesos del vector de Pesos (bucle de la línea 10). La primera regla que se puede aplicar ocurre con una probabilidad basada en el parámetro HMCR (este paramero se recomienda entre 0.7 y 0.95) y corresponde a la asignación del valor del peso de una armonía aleatoriamente seleccionada de la memoria armónica (línea 12). Como se observa en la línea 12 el valor r2 corresponde a un número entero aleatoriamente seleccionado entre 0 y HMS – 1, incluidos los valores límites. La segunda regla es la que da el comportamiento de enjambre al tomar el valor del peso de la mejor solución (armonía) de la memoria armónica (línea 14) y ocurre con una probabilidad PAR (se recomienda que este parámetro varíe entre un mínimo de 0.3 y un máximo de 0.7). La última regla se aplica con una probabilidad 1 – HMCR y consiste en generar aleatoriamente el valor

del peso entre 0 y 1. Esta última regla genera diversidad en el proceso de búsqueda y las dos primeras están diseñadas para que la nueva solución use valores de diferentes padres de soluciones ya encontradas.

Terminado el proceso de creación de la nueva armonía con base en el operador de improvisación, se procede a normalizar (línea 20) la nueva solución con base en el mismo proceso explicado previamente en el método GenerarPesosAleatorios (ejemplificado en la **Figura 4**) y luego se procede a calcular la calidad de los pesos ya normalizados para la nueva armonía (línea 21).

Con la nueva armonía (pesos y calidad o fitness) se realiza el operador de remplazo (líneas 22 a 25). Primero. En la línea 22 se evalúa si la peor solución de la memoria armónica cuenta con una menor calidad que la obtenida con la nueva armonía, si esto es así se procede a remplazar la peor solución con la nueva armonía (línea 23) y luego a ordenar la memoria armonía de la mejor a la peor solución basado en su calidad (línea 24) de la misma forma como se hace en la línea 6. El proceso de ordenamiento se puede remplazar por un proceso en el cual se busque la posición de la mejor y la peor solución, esto último es más rápido computacionalmente hablando, aunque su aporte no es realmente significativo sabiendo que el tamaño de la memoria armónica es un número entero pequeño, normalmente menor igual a 20. Finalmente, la línea 27 retorna los pesos que se encuentran en la mejor solución que se encontró al terminar el proceso de búsqueda y optimización.

Terminado el proceso de definición de pesos el modelo CLR queda completo. El modelo CLR queda definido por un número K de grupos de calidad, cada grupo de calidad cuenta con los registros que lo conforman (los 14 atributos independientes y el valor de PSI real como variable dependiente) y su modelo de predicción. Además del vector de pesos para los atributos independiente del dataset que permite hacer la búsqueda del registro más parecido en los grupos usando el algoritmo 1NN y la distancia euclidiana ponderada para definir a qué grupo pertenece un nuevo registro.

3.4.5 Proceso de evaluación del modelo CLR obtenido

El proceso de predicción toma el conjunto de datos de validación (DatasetValidacion) y a cada registro le busca el registro del modelo CLR (compuesto por 1. grupos con sus modelos de regresión y 2. pesos ponderados de los atributos) más parecido usando 1-NN y la distancia euclidiana ponderada con

los pesos del modelo. Obtenido el registro más parecido se toma el modelo de regresión al que está asociado ese registro y con este modelo se le calcula el PSI predicho al registro de validación. Con el PSI predicho y el PSI real del dataset de validación se calculan diversas métricas (P15, MSE, MAR, RMSE y NRMSE) que son el resultado del proceso de evaluación (ver **Pseudocódigo 5**).

El **Pseudocódigo 5** muestra en detalle el proceso. Comienza recorriendo el conjunto de datos de validación (DatasetValidacion) con un ciclo “para” en la línea 1, dentro del ciclo se llama la función MinDistanciaGrupo que recibe el registro actual del conjunto de datos de validación, los grupos definitivos y los pesos de los atributos del modelo CLR, calcula cuál es el registro más similar y retorna el grupo al que ese registro pertenece; este procedimiento se explica más adelante. En la línea 3 se obtiene el PSI real (variable Psi) del registro actual del dataset de validación. En las líneas 4 a 9 se calcula iteración tras iteración el mayor (PsiMax) y menor (PsiMin) valor de PSI real en todos los registros del dataset de validación, datos que se requieren para el cálculo del NRMSE. A continuación se calcula el valor de PSI predicho, es decir, se aplica el modelo de regresión lineal del grupo que fue seleccionado en la línea 2 con los datos del registro actual de validación. Después en las líneas 11 a 13 se cuenta el registro en la estadística P15 si el valor de PSI predicho se encuentra en el rango del PSI real ±15%. En la línea 14 se va guarda el valor del PSI real menos el PSI predicho elevado al cuadrado para calcular la métrica MSE. En la lineal 15 se guarda el valor absoluto del PSI real menos el PSI predicho para calcular la métrica MAE. Una vez terminados todos los registros del conjunto de datos de validación, se procede a calcular y retornar las métricas requeridas de la línea 17 a la 22 realizando las operaciones pertinentes para cada una de ellas.

Para calcular la mínima distancia y obtener el grupo al cual corresponde el registro del conjunto de datos de validación se utiliza el método MinDistanciaGrupo, este procedimiento se puede ver en el **Pseudocódigo 6**. En la línea 1 se inicia con un ciclo “para”, que recorre cada uno de los grupos definitivos. Dentro de este ciclo se recorren uno a uno todos los registros del grupo definitivo actual (línea 2). Dentro de este ciclo interno se llama a la función DistanciaPesos con los parámetros del RegistroValidacion, el registro actual del grupo definitivo actual (GrupoDefinitivo [i].Registro [j]) y los pesos del modelo CLR; esta función se explica en el siguiente párrafo. Luego en las líneas 4 a 7 se calcula cuál es el mejor grupo (MejorGrupo),

aquel donde se encuentra el registro que tiene la menor distancia euclidiana con el registro de validación. Finalmente en la línea 10 se retorna el grupo seleccionado con su modelo de regresión que hacen parte del modelo CLR.

Entradas: • **DatasetValidacion:** Conjunto de datos de validación en formato ARFF • **GruposDefinitivos:** Grupos finales, cada uno de ellos con su modelo de regresión del modelo CLR. • **Pesos:** Pesos entregados por GBHS como parte del modelo CLR. **Salidas:** • **P15:** Porcentaje de registros cuyas predicciones se encuentran en el rango del valor real ±15% de este valor. • **MSE:** Error cuadrático medio (Mean Squared Error) • **MAE:** Error absoluto medio (Mean Absolute Error, MAE) • **RMSE:** Error cuadrático medio (Root-Mean-Squared Error, RMSE) • **NRMSE:** Raíz del error cuadrático medio normalizada (Normalized Root-Mean-Square Error, NRMSE)	**Complejidad** $O\,(m * n * p)$
Inicio	
1. **Para** i = 0 **Hasta** DatasetValidacion.Tamaño **Hacer**	$O\,(m)$
2. GrupoSeleccionado = **MinDistanciaGrupo** (DatasetValidacion [i], GruposDefinitivos, Pesos**)**	$O\,(n * p)$
3. Psi = DatasetValidacion [i].ObtenerPSI ()	$O\,(1)$
4. **Si** i = 0 **OR** Psi > PsiMax **Entonces**	$O\,(1)$
5. PsiMax = Psi	$O\,(1)$
6. **Fin Si**	
7. **Si** i = 0 **OR** < PsiMin **Entonces**	$O\,(1)$
8. PsiMin = Psi	$O\,(1)$
9. **Fin Si**	
10. PsiPredicho = GrupoSeleccionado.PredecirPSI (DatasetValidacion[i])	$O\,(p)$
11. **Si** PsiPredicho > Psi*0.85 **Y** PsiPredicho < Psi*1.15 **Entonces**	$O\,(1)$
12. P15 = P15 + 1	$O\,(1)$
13. **Fin Si**	
14. MSE = MSE + (Psi - PsiPredicho)^2	$O\,(1)$
15. MAE = MAE + | Psi - PsiPredicho | // valor absoluto	$O\,(1)$
16. **Fin Para**	
17. P15 **=** (100.0*Inside15) / Dataset.Tamaño ()**;**	$O\,(1)$
18. MSE = MSE / DatasetValidacion.Tamaño ()	$O\,(1)$
19. MAE = MAE / DatasetValidacion.Tamaño ()	$O\,(1)$
20. RMSE = RaízCuadrada(MSE)	$O\,(1)$
21. NRMSE = RMSE / (PsiMax-PsiMin)	$O\,(1)$
22. **Retornar** P15, MSE, MAE, RMSE, NRMSE	$O\,(1)$
Fin	

Pseudocódigo 5. Algoritmo de evaluación de grupos definitivos.

Entradas: • **RegistroValidacion:** Registro individual del conjunto de datos de validación en formato ARFF • **GruposDefinitivos:** Grupos finales, cada uno de ellos con su modelo de regresión del modelo CLR. • **Pesos:** Pesos entregados por GBHS como parte del modelo CLR. **Salidas:** • **GrupoSeleccionado:** Grupo al cual pertenece el registro más parecido al registro de validación	**Complejidad** $O\ (n * p)$
Inicio	
1. **Para** i = 0 **Hasta** GruposDefinitivos.Tamaño **Hacer**	$O\ (GD)$
2. **Para** j = 0 **Hasta** GrupoDefinitivos [i].Tamaño **Hacer**	$O\ (Rs);\ Rs \approx n/GD$
3. Distancia = DistanciaPesos (RegistroValidacion, GrupoDefinitivos [i].Registro [j], Pesos)	$O\ (p)$
4. **Si** (i = 0 **AND** j = 0) **OR** Distancia < DistanciaMin **Entonces**	$O\ (1)$
5. DistanciaMin = Distancia	$O\ (1)$
6. MejorGrupo = i	$O\ (1)$
7. **Fin SI**	
8. **Fin Para**	
9. **Fin Para**	
10. **Retornar** GrupoDefinitivo [MejorGrupo]	$O\ (1)$
Fin	

Pseudocódigo 6 Algoritmo para definir el grupo con el que se debe predecir el PSI a un registro de validación o nuevo (MinDistanciaGrupo).

Para hallar la distancia (DistanciaPesos) entre dos registros se necesitan el registro de validación, un registro de algún grupo definitivo del modelo CLR y los pesos del modelo obtenidos con GBHS. En **Pseudocódigo 7** se muestra el procedimiento. En la línea 1 se inicializa la variable Distancia, luego en la línea 2 se realiza un ciclo para recorrer las variables o atributos que componen cada registro de datos (sin incluir el PSI). Después se pregunta si la variable actual es numérica (línea 2), en el caso de que sea verdadero entonces a la distancia se le suma el resultado de la variable actual del RegistroValidacion menos el RegistroGrupo al elevarla al cuadrado multiplicado por el peso de la variable (línea 4). En caso contrario, ósea que la variable no sea numérica (línea 5), en la línea 5 se verifica si las variables RegistroValidacion[i] y RegistroGrupo[i] son diferentes, en caso de ser diferentes se suma solo el valor del peso de esa variable a la distancia. Este proceso se realiza para cada variable del registro y al final se retorna la raíz cuadrada de la distancia obtenida (línea 10).

Entradas: • **RegistroValidacion**: Registro individual del conjunto de datos de validación. • **RegistroGrupo:** Registro individual de un grupo definitivo. • **Pesos:** Pesos entregados por GBHS como parte del modelo CLR. **Salidas:** • **Distancia:** La distancia entre los dos registros de entrada.		**Complejidad** $O(p)$
	Inicio	
1.	Distancia = 0	$O(1)$
2.	**Para** i = 0 **Hasta** Pesos.Tamaño **Hacer**	$O(p)$
3.	**Si** RegistroGrupo.Attributos [i] es numérico **entonces**	$O(1)$
4.	Distancia = Distancia + Pesos [i] * ((RegistroValidacion [i] – RegistroGrupo [i])^2)	$O(1)$
5.	**Si no**	
6.	**Si** RegistroValidacion [i] != RegistroGrupo [i]	$O(1)$
7.	Distancia = Distancia + Pesos[i]	$O(1)$
8.	**Fin Si**	
9.	**Fin Si**	
10.	**Fin Para**	
11.	**Retornar** RaízCuadrada (Distancia)	$O(1)$
	Fin	

Pseudocódigo 7. Algoritmo para calcular la distancia entre dos registros (DistanciaPesos).

3.5 Complejidad computacional

El algoritmo está compuesto por la Fase 1, la Fase 2 y el cálculo de los pesos de los atributos con GBHS. A continuación, se muestra el cálculo de la complejidad computacional para cada uno de estos procesos. En la **Tabla 2** primero se presentan los parámetros utilizados para el cálculo de dicha complejidad.

3.5.1 Fase1

En el **Pseudocódigo 1** en la columna de la derecha se presenta la complejidad computacional línea por línea. En la línea 1 la complejidad es un valor constante $O(1)$, luego en la línea 2 la complejidad computacional depende de un ciclo "mientras" y es igual a $O(c); c \ll n$, después en la línea 3 se utiliza la complejidad computacional de la regresión lineal de Weka que corresponde a $(p^2n + p^3)$ [106]. La complejidad computacional del bloque comprendido entre las líneas 3 a 14 estaría dado por $(p^2n + p^3) + nlog(n) + n + (p^2n + p^3) + 1 + 1 + n + 1 + h$, que se puede resumir en $2(p^2n + p^3) + nlog(n) + 2n + 3 + h$. Lo anterior se encuentra

dentro de un ciclo “mientras” que depende del tamaño del conjunto de datos y este tamaño se modifica internamente por cada iteración y se define de la siguiente manera $c * (2(p^2n + p^3) + nlog(n) + 2n + 4)$, obtenido así una complejidad total para la Fase 1 igual a $\boldsymbol{O\left(p^2n + p^3 + n * log(n)\right)}$ dado que c es un número muy pequeño en comparación con n.

Tabla 2. Descripción de parámetros para el cálculo de la complejidad.

Parámetro	Descripción
n	Cantidad de registros (filas) en el conjunto de datos de entrenamiento
m	Cantidad de registros (filas) en el conjunto de datos de validación
p	Número de atributos (variables) independientes de los conjuntos de datos (todos excepto el PSI)
GC	Cantidad de grupos de calidad
GD	Cantidad de grupos definitivos
h	Cantidad de registros huérfanos
c	Número no predeterminado de ciclos a realizar
HMS	Tamaño de la memoria armónica en GBHS
NI	Número de improvisaciones (descendientes) a generar en GBHS

3.5.2 Fase 2

La complejidad de esta fase 2 se puede ver por cada línea en la columna derecha del **Pseudocódigo 2**. El bloque de líneas 2 y 3 tienen una complejidad de $GC * (p^2n + p^3) + 1$. Este cálculo se encuentra dentro de un ciclo “para” que comprende a las líneas 1 y 5 y está regido por el valor de $O(h); h < n$ que corresponde a la cantidad de registros huérfanos resultantes de la fase anterior, de esta manera el cálculo de la complejidad del método corresponde $\boldsymbol{O(h * GC * (p^2n + p^3))}$**.**

3.5.3 Definición de pesos con GBHS

La complejidad computacional de cada línea se puede ver en la columna derecha del **Pseudocódigo 4**. La inicialización de la memoria se realiza HMS veces y el bloque interior (líneas 2 a 4) realiza unas operaciones con complejidad computacional de $(p + n^2 * p + 1)$, por lo que este bloque de líneas 1 a 5 cuesta en total $O(HMS * (p + n^2 * p + 1))$. En la línea 6 se ordena la memoria armónica con un costo de $O\,(HMS * \log HMS)$ que en general es mínimo ya que la memoria armónica es pequeña, al ser HMS un valor pequeño (entre 5 y 20). Con lo anterior se puede decir que la inicialización tiene un costo de $O(HMS * n^2 * p)$. El bloque de

improvisaciones se realiza NI veces y agrupa las líneas de la 7 a la 26 donde el costo importante lo establece la línea 21, que corresponde a la evaluación de la calidad del vector de pesos del improviso y tiene un costo de $O\ (n^2 * p)$. Es así como la fase de improvisación o evolución tiene un costo total de $O\ (NI * n^2 * p)$. En este bloque también se desestima el costo del ordenamiento de la memoria armónica por ser mínimo. Sumando el costo de las dos fases (inicialización e improvisación) el costo total de la definición de pesos es de $\boldsymbol{O((HMS + NI) * n^2 * p)}$.

3.5.4 Construcción completa del modelo CLR

El **Pseudocódigo 3** también muestra en la columna derecha la complejidad de las operaciones realizadas. La complejidad de este algoritmo la gobiernan unas cuantas líneas, a saber: 1) La línea 2 que tiene una complejidad de $O\left(p^2n\ +\ p^3 + nlog(n)\right)$; 2) La línea 6 que tiene una complejidad de $O(h * GC * (p^2n + p^3))$ y se realiza en el marco de un ciclo con c iteraciones donde este número es muy pequeño, por lo que el costo de este bloque de lógica es de $O(h * GC * (p^2n + p^3))$; 3) La línea 12 con una complejidad de $O(GC * (p^2n\ +\ p^3 + nlog(n)))$ que se ejecuta GC veces (línea 9) en el marco de un ciclo con c iteraciones donde este número es muy pequeño, por lo que el costo de este bloque de lógica es de $O(GC^2 * (p^2n\ +\ p^3 + nlog(n)))$; Y la línea 22 que tiene una complejidad de $O\ ((HMS + NI) *\ n^2 * p)$. De estos 4 componentes, el primero y el tercero se pueden agrupar en uno sólo que al unirse con el segundo da como resultado $O\ ((GC^2 + h * GC) * (p^2n\ +\ p^3) + GC^2 * n * log(n)))\ donde\ h < n\ y\ GC \ll n$ por lo que la expresión se puede reducir a $O\ (h * GC * (p^2n\ +\ p^3) + GC^2 * n * log(n)))\ donde\ h < n\ y\ GC \ll n$. A este resultado se agrega el último componente y se tiene la siguiente complejidad $\boldsymbol{O\ (h * GC * (p^2n\ +\ p^3) + GC^2 * n * log(n)) + (HMS + NI) *\ n^2 * p)\ donde\ h < n\ y\ GC \ll n}$.

3.5.5 Uso del modelo CLR en producción

Dado un nuevo registro, predecir su valor de PSI implica llamar al método MinDistanciaGrupo (ver **Pseudocódigo 6**) que tiene un costo de $O\ (n * p)$ y aplicar el modelo de regresión del grupo seleccionado que tiene una complejidad de $O\ (1)$. Por lo que la complejidad del uso del modelo CLR propuesto en un entorno de producción es equivalente a $\boldsymbol{O\ (n * p)}$.

3.5.6 Evaluación del modelo CLR

El proceso de evaluación consiste en evaluar m registros del dataset de validación, donde el costo de evaluar un registro es de $O\ (n * p)$ y teniendo m registros en el dataset de validación el costo total es de $O\ (m * n * p)$ más los costos asociados a los cálculos de las métricas de calidad (MSE, MAE, RMSE y NRMSE) que son mínimos, por lo que el costo total del proceso de evaluación es de $\boldsymbol{O}\ (\boldsymbol{m} * \boldsymbol{n} * \boldsymbol{p})$.

CAPÍTULO 4

4 Experimentación

El algoritmo se implementó usando el entorno de desarrollo integrado (IDE) NetBeans en la versión 8.2 con el lenguaje de programación Java versión "1.8.0.131" y la librería Weka-stable-3.8.0.jar [107]. La librería de Weka sirvió para la lectura de datos, procesamiento de datos, definición del modelo, entre otros. El código fuente de esta implementación se encuentra disponible en https://gitlab.com/pacho328/paviments-greedy-weigths-gbhs donde también se encuentran las librerías necesarias y un archivo "readme.md" con los pasos para su uso.

4.1 Afinamiento de parámetros

Para establecer la configuración de los parámetros de entrada del algoritmo greedy se probaron 288 combinaciones que resultan de iterar el parámetro MínimoRegistros con un valor inicial de 1.400 hasta 1.600, el parámetro MáximoRegistros de 2.800 hasta 3.200, ambos parámetros con un incremento de 200 por cada iteración. Además, el parámetro ErrorPromedioAceptado que inicia con un valor de 0,02 hasta un valor de 0,035 y el parámetro ErrorPromedioAdicional que inicia en 0,03 hasta el valor de 0,065, ambos parámetros con incremento de 0,005 por iteración. En la **Tabla 3** se muestran los valores de los parámetros de entrada que generaron el mejor resultado.

Tabla 3 Valores de los parámetros de entradas del algoritmo greedy

Parámetro Algoritmo Greedy	Valor
MínimoRegistros	1.600
MáximoRegistros	3.000
ErrorPromedioAceptado	0,02
ErrorPromedioAdicional	0,065

En la **Tabla 4** se muestran los valores utilizados en GBHS para esta investigación. Valores que son tomados de las recomendaciones encontradas en el estado del arte.

Tabla 4 Valores de los parámetros de GBHS

Parámetro GBHS	Valor
HMS	10
NI	100
HCMR	0.85
ParMin	0.1
ParMax	0.4

4.2 Clusterwise NonLinear Regression

Para realizar el estudio de una propuesta no lineal, en esta investigación inicialmente se realizó una transformación a los datos aplicando logaritmo natural a cada uno de los atributos numéricos de los data sets de entrenamiento y validación (17.643 registros), después se modificó la lógica de validación y el código fuente del algoritmo greedy propuesto, adicionando una operación de antilogaritmo en el proceso de validación, con estos cambios se vuelve a lanzar el algoritmo greedy, dando como resultado el mismo dataset de entrenamiento. Los resultados del uso del modelo no lineal (CNLR) no fue satisfactorio ya que no puedo realizar más de 1 grupo, por ende este enfoque no se tomó en cuenta en la comparación con el modelo lineal propuesto y los reportados en el estado del arte.

4.3 Resultado obtenido y comparación

Iniciando la experimentación se realizó un modelo base utilizando la función de regresión lineal de la librería de Weka sin realizar ninguna modificación a los conjuntos de datos de entrenamiento y validación. Este proceso arrojó 3.005 valores de predicción de PSI correspondientes a cada registro del conjunto de datos de validación; con esto se calculó el porcentaje de registros que se encuentran en el rango del PSI real ±15% de este valor con un resultado igual a 89%. Revisando la literatura, se observó que este porcentaje supera los trabajos previos de Khadka et al. tal como se muestra en la **Figura 6**. Esta figura presenta los resultados (PSI real vs PSI predicho) organizados en tres partes, la parte (a) muestra el modelo base lineal que se desarrolló en esta investigación, la parte (b) muestra los resultados obtenidos por [7] usando 6 grupos con modelos lineales (6 grupos) con un resultado

de 74% y la parte (c) muestra el enfoque no lineal de [7] con 5 agrupaciones con un resultado de 81%. Se evidencia que los enfoques previamente presentados por Khadka et al. no superan al modelo base lineal que cuenta con un 89% de puntos dentro del rango de ±15% del valor PSI real.

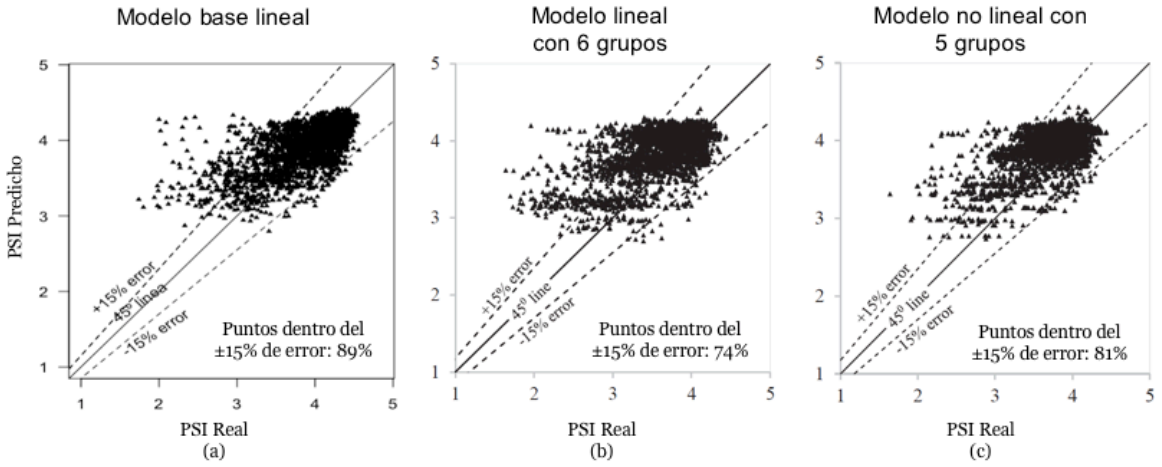

Figura 6. Modelo base lineal (a), modelo lineal (CLR) con 6 grupos propuesto por Khadka et al.(b) y modelo no lineal (CNLR) con 5 grupos propuesto por Khadka et al.

A continuación, en la **Figura 7** se puede observar en la parte (a) la gráfica correspondiente a los resultados del algoritmo propuesto en este documento. El algoritmo generó 6 grupos con modelos de regresión lineal y obtuvo un **95%** de puntos dentro del ±15% del PSI real. Los resultados obtenidos con el modelo CLR propuesto en este trabajo de grado son superiores a la propuesta lineal (CLR) de [7] en un 21%, a la propuesta no lineal (Clusterwise No Lineal Regression, CNLR) de [7] en un 14% y al modelo base lineal en un 6%. Es preciso comentar que los modelos propuestos por Khadka et al. se desarrollaron con el algoritmo de recocido simulado que evoluciona agrupaciones con un número específico de grupos; luego analizan los resultados de distintos números de grupos y se selecciona el de mejor calidad basada en el indicador BIC (Bayesian Information Criterion).

La **Figura 7** muestra en la parte (b) el modelo base lineal y facilita el análisis visual con respecto al modelo propuesto en esta investigación; se evidencia en el gráfico del modelo lineal con 6 grupos propuesto que tiene los puntos más cerca de la línea diagonal, evidenciando sus mejores resultados. Buscando evaluar el uso de modelos no lineales, los datos de los data sets de entrenamiento y validación se transformaron usando logaritmo natural para todos los atributos numéricos para que

al crear el modelo de Weka se generará un modelo base no lineal. Los resultados de este modelo base no lineal tienen menor calidad que los otros dos modelos. Tomando estos datos transformados también se ejecutó el algoritmo propuesto en esta investigación y los resultados tampoco fueron mejores por lo que no se presentan. Es preciso comentar que para el proceso de evaluación de los modelos se aplicó el antilogaritmo a la variable PSI para comparar apropiadamente los resultados obtenidos.

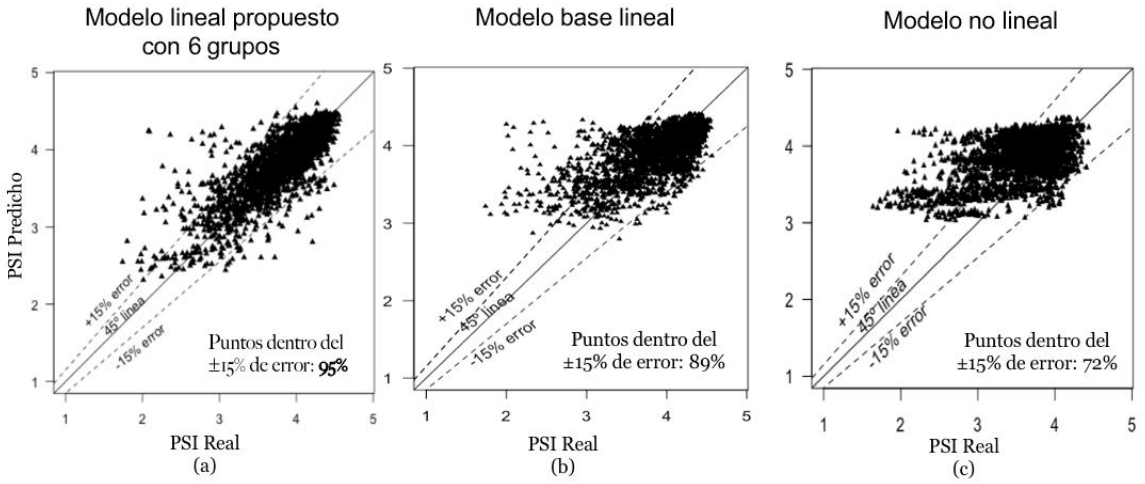

Figura 7 Modelo CLR propuesto (a) y modelo base (b), modelo no lineal propuesto (c).

La **Tabla 5** muestra otras métricas de comparación del modelo propuesto y de los modelos base lineal y no lineal y los modelos lineal y no lineal desarrollados por [7]. Los resultados están ordenados empezando con los mejores al lado izquierdo y los peores al lado derecho. Es preciso señalar que para las métricas MAE (Mean Absolute Error), RMSE (Root-Mean-Square Error) y NRMSE (Normalized RMSE) los valores más bajos reflejan mejores resultados. Por otro lado, para la métrica "±15% del PSI real" entre mayor sea el valor, mejor es el resultado del modelo. Debajo de cada valor de las métricas de los modelos con los que la propuesta se comparó, aparece entre paréntesis el porcentaje de mejora que logra la propuesta frente al estado del arte y los modelos base.

El modelo CLR lineal propuesto en este trabajo es más complejo que los modelos base (lineal y no lineal) ya que define 6 grupos, mientras que los modelos base tienen solo uno, es decir el dataset de entrenamiento completo, pero mejora en todas las métricas, entre un 36% y un 57.9% mejor en MAE, entre un 29% y un 49% mejor en RMSE, entre un 25% y 47.1% mejor en NRMSE y entre un 6.7% y un

31.9% en la métrica del "±15% del PSI real". En relación con los modelos CLR y CNLR del estado del arte, los resultados son aún mejores en todos los valores de las métricas, en relación con el modelo lineal no hay diferencia en número de agrupaciones, por lo que tiene el mismo nivel de complejidad y la diferencia con el modelo CNLR no es significativa es solo de 1 grupo, pero el modelo CNLR es más complejo en su interpretación por ser no lineal.

Tabla 5. Métricas del comportamiento de la predicción del PSI.

Medida	Modelo Lineal Propuesto con 6 agrupaciones	Modelo Base Lineal	Modelo CNLR Khadka et al. con 5 agrupaciones	Modelo CLR de Khadka et al. con 6 agrupaciones	Modelo Base No Lineal
MAE	**0,16**	0,25 (36%)	0,33 (51.5%)	0,36 (55.6%)	0,38 (57.9%)
RMSE	**0,25**	0,35 (29%)	0,41 (39%)	0,47 (46.8%)	0,49 (49%)
NRMSE	**0,09**	0,12 (25%)	0,15 (40%)	0,17 (47.1%)	0,17 (47.1%)
±15% del PSI real	**95%**	89% (6.7%)	81% (17.3%)	74% (28.4%)	72% (31.9%)

La **Tabla 6** muestra el número de registros de cada grupo y el coeficiente de correlación de los modelos de regresión lineal de cada grupo en el entrenamiento. Como se puede apreciar, todos los grupos cuentan con 1600 o más registros y tienen un coeficiente de correlación igual o superior a 95% excepto por un grupo que tiene un valor de 89%. Contar con esos valores de correlación altos permiten al modelo tener buenos resultados sobre el dataset de validación.

Tabla 6. Resultados de número de registros y coeficiente de correlación por grupo.

	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6
Número de registros	2750	2497	2610	1769	2618	2394
Coeficiente de correlación	98%	95%	89%	98%	96%	96%

Para evaluar el tiempo de ejecución de las propuestas, estas se ejecutaron en el mismo servidor (n1-highcpu-8, 8 CPU virtuales, 7.2 GB de memoria de Google Cloud Platform). La implementación de [7] que está programada en R tomó 25 horas de ejecución para construir un modelo con 6 grupos, esto implica que su ejecución

completa demoraría aproximadamente 375 horas (15.6 días) para evaluar desde 2 hasta 16 grupos y de esos modelos seleccionar la mejor opción conforme lo plantean los autores. La implementación del modelo CLR propuesta en este trabajo se ejecutó en 24 horas con 47 minutos probando diferentes combinaciones de valores de ErrorPromedioAceptado y ErrorPromedioAdicional (un total de 37 combinaciones) lo que significa que una sola ejecución tan solo demora 40 minutos, aproximadamente. Es claro que la construcción del modelo CLR propuesto en este documento, además de tener mejor calidad se demora mucho menos tiempo, tan solo el 0.7% del tiempo que ocupa la propuesta de Khadka et al. En el caso del modelo base se utilizó la herramienta Weka en la versión 3.9.4 en un computador con un procesador de 2.5 GHz Intel Core i5 de dos núcleos y 16 GB de memoria; la construcción del modelo base lineal demoró 0.82 segundos y 0.17 segundos para la evaluación. La construcción del modelo base no lineal demoró 0.32 segundos y 0.39 segundos para la evaluación (sin incluir el tiempo de la transformación de los datos).

Los coeficientes de cada grupo del modelo CLR obtenido con el algoritmo propuesto se presentan en la **Tabla 7** y en la siguiente sección se analizan.

4.4 Análisis de los coeficientes de los modelos

A continuación, se analiza el resultado de los modelos de cada grupo con relación al PSI. Las variables **aadt** y **trucks** no influyeron en ninguno de los modelos; se requieren análisis más profundos para definir si realmente no tienen relación directa con el PSI u otras variables ya incluidas en el dataset aportan más información que estas. La variable de elevación (**elevation**) no influye en el valor del PSI y en este sentido está conforme a lo expresado por los expertos del área. Para estas tres variables se pueden observar coeficientes de cero o cercanas a cero en los modelos de los seis grupos obtenidos.

Se puede observar que cuando un segmento de pavimento ha tenido una reparación o mantenimiento de dos años o menos (atributo **age**) su impacto en el valor de PSI es positivo en todos los grupos. También se puede notar en la mayoría de los grupos que el aumento en el tiempo de la última intervención (reparación o mantenimiento) este valor impacta negativamente sobre el valor de PSI, lo que en general corresponde con lo esperado. Es preciso notar que el grupo 2 con edad 7 y el grupo 3 con edades de 7, 8 y 9 no siguen esta tendencia, lo que puede deberse a la

combinación con otros factores registrados o no en el dataset como el tipo de intervención, el tipo de pavimento, la calidad de los materiales, entre otros.

El atributo de precipitaciones (**precip**) influye negativamente en el valor de PSI en todos los grupos, aspecto que corresponde a lo esperado, ya que a mayor cantidad de precipitaciones se deteriora más la calidad del pavimento.

El atributo de temperatura mínima (**min_temp**) afecta negativamente el PSI solo al grupo 6, en los demás grupos los afecta positivamente. El atributo de temperatura máxima (**max_temp**) influye positivamente en el PSI de los grupos 4 y 6 mientras que en los demás grupos lo afecta negativamente. Se espera que entre mayor sea la temperatura el pavimento se torne más frágil y se deteriore más, con lo que el PSI tiene una tendencia a bajar, es decir, con altas temperaturas existe más probabilidad de deterioro del pavimento. En ambos casos el grupo 6 sobresale como excepción a este comportamiento.

En la variable de días húmedos (**wet_days**) se observa que los grupos 2 y 3 tienen un efecto negativo en el PSI mientras que en los otros grupos es positivo. Estos dos grupos tienen en común que su valor de intercepto es relativamente más alto (7.9970 y 9.5581 respectivamente) que el de los otros grupos, los cuales están entre 2.2933 y 5.6360. Teniendo en cuenta que el dataset no contiene datos sobre el tipo de pavimento, es posible que el efecto de esta variable no contemplada este distorsionando en estos grupos el coeficiente de regresión para esta variable.

La congelación del pavimento (**freeze_thaw**) influye de manera negativa al PSI en cuatro de los seis grupos, lo que es esperado ya que al congelar y descongelar los materiales del pavimento se expanden sus componentes, lo que produce roces que deterioran la calidad del pavimento. A los grupos 1 y 2 esto no los afecta

Las abolladuras (**rut_depth**) influyen negativamente en todos los grupos, puesto que entre más abolladuras existan en un segmento de pavimento más será su deterioro y esto causa un efecto negativo en el valor del PSI. Este atributo tiene más influencia negativa que todos los atributos relacionados con el clima, como se ve en la **Figura 8**. La **Figura 9** muestra como las abolladuras influyen mucho en todos los grupos, más que el atributo de edad en la mayoría de los casos.

Cuando los segmentos de pavimentos son de una línea (**number of lanes = 1**) el efecto es positivo sobre el PSI, a diferencia de los segmentos de pavimento de dos

líneas (**number of lanes = 2**) y de tres líneas (**number of lanes = 3**), esto debido a que estos dos últimos tienen mayor tráfico de vehículos no solo de pasajeros sino de carga con mucho mayor peso.

Tabla 7. Resultados de los modelos por grupo

Variable	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6
age=0	0.1440	0.1789	0.2057	0.0737	0.1275	0.1491
age=1	0.0937	0.0912	0	0.0983	0.1424	0.0810
age=2	0.0262	-0.0375	-0.0919	0.0784	0.0890	0.0865
age=3	0.0334	0	-0.0914	0.0579	0.1207	0
age=4	-0.1083	-0.1847	-0.2438	-0.0192	0	0.0342
age=5	-0.0282	-0.0199	-0.1655	0.0203	0.0527	0
age=6	-0.1269	-0.1467	-0.2146	-0.0852	-0.0338	-0.0626
age=7	-0.1992	0.1527	0.8787	-0.5707	-0.7634	-0.4630
age=8	-0.3787	-0.1528	0.3899	-0.6063	-0.5892	-0.6439
age=9	-0.2333	-0.0512	0.7104	-0.7467	-0.6609	-1.2133
age=10	0	-1.2393		-1.3968	0	-0.7103
aadt	0	0	0	0	0	0
trucks	0	0.0001	0.0001	0	0	0.0001
elevation	0	-0.0001	-0.0001	0.0001	0.0001	0.0001
precip	-0.0224	-0.0128	-0.0134	-0.0510	-0.0477	-0.0051
min_temp	0.0166	0.0282	0.0025	0.0097	0.0193	-0.0201
max_temp	-0.0289	-0.0648	-0.0672	0.0031	-0.0226	0.0312
wet_days	0.0005	-0.0061	-0.0070	0.0087	0.0040	0.0025
freeze_thaw	0	0	-0.0055	-0.0009	-0.0007	-0.0007
rut_depth	-1.2954	-1.8379	-1.7141	-0.7785	-0.8373	-0.7546
number_of_lanes=1	0.1056	0.1913	0.1746	-0.0108	0	0.0724
number_of_lanes=2	-0.0879	-0.1635	-0.3065	0	-0.0608	-0.0386
number_of_lanes=3	-0.1294	-0.1970	0	0	0	-0.2081
sys_id=1 (IR)	0.0732	-0.2283	-0.4032	0.3667	0.3315	0.2262
sys_id=2 (NHS)	0	0.0924	0.1768	-0.1025	-0.1146	-0.0948
sys_id=3 (STP)	-0.0732	0.0987	0.0990	-0.3430	-0.2301	-0.1014
f_class=1	-0.1737	0.0344	0.1226	-0.2040	-0.1741	-0.2232
f_class=2	0.0685	-0.0498	-0.3919	0.1644	0.1566	0.3210
f_class=3	-0.0578	0	-0.0561	0.0312	0.1011	0.1173
f_class=4	0.0597	0.0588	0	0.3039	0.2199	0.1939
f_class=5	-0.2207	-0.1414	-0.1262	-0.0316	-0.1592	0.1270
f_class=6	-0.3695	0.0756	0.0926	-0.6217	-0.6018	-1.2740
f_class=7	0	-0.1185	-1.6062	0.4813	0.2779	0.4400
category=1	0.2163	0.4139	0.6372	0.0139	0.0877	0.2058
category=2	0.0731	0.0962	0.0636	0.0971	0.1557	0.1088
category=3	0.0150	-0.0824	-0.1620	0.1750	0.2189	0.0531
category=4	-0.2872	-0.0661	0.3988	-0.5428	-0.4961	-0.8923
category=5	-0.4758	-0.6437	-0.9798	-0.2331	-0.2714	-0.0807
intercepto	5.6360	7.9970	9.5581	3.1913	4.8759	2.2933

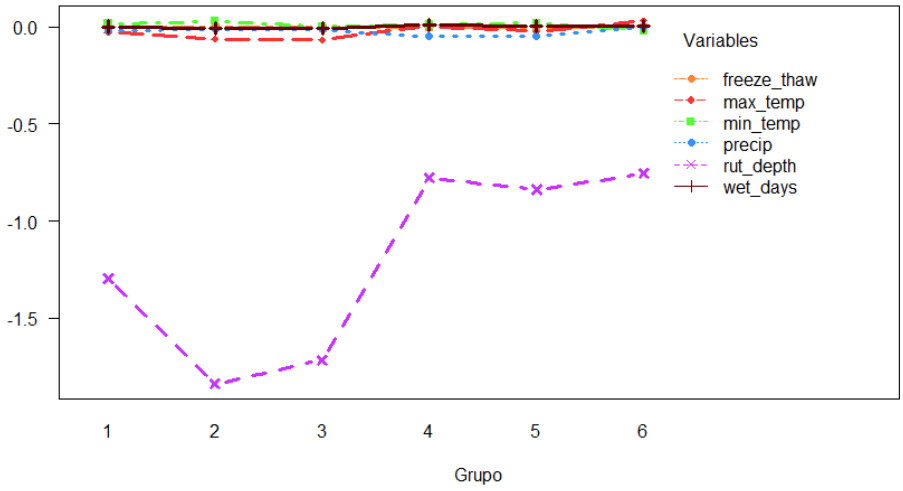

Figura 8 Rut_Depth versus clima (temperaturas, precipitaciones y días húmedos).

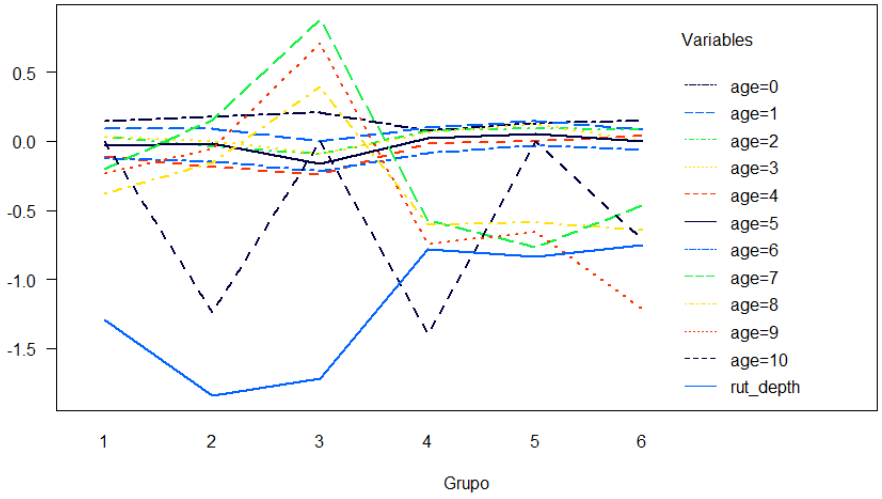

Figura 9 Rut_depth versus age.

Esta página ha sido dejada intencionalmente en blanco.

CAPÍTULO 5

5 Conclusiones y trabajo futuro

En esta investigación se propuso un algoritmo voraz que de forma incremental crea grupos de registros con sus modelos de regresión lineal asociados los cuales permiten predecir el valor de PSI de un tramo de pavimento basado en catorce (14) variables explicativas. Los modelos de predicción que cumplen un nivel mínimo de correlación se aceptan para formar parte de un grupo selecto y temporal de grupos de calidad (Fase 1). Luego los registros que no lograron formar parte de los grupos de calidad se incluyen en forma iterativa en los grupos de calidad donde mejore el coeficiente de correlación o disminuya en menor valor dicho coeficiente de correlación (Fase 2). Después las dos fases interactúan intercaladamente hasta que los grupos incluyan todos los registros y todos los grupos tengan un nivel mínimo de calidad (medida en coeficiente de correlación) y un número mínimo y máximo de registros. Para poder definir a qué grupo debe pertenecer un nuevo registro, primero se usa el algoritmo 1NN con la distancia euclidiana ponderada para identificar el registro más similar en sus características, luego con el modelo de regresión del grupo de ese registro similar se puede calcular el valor de PSI del nuevo registro. Para poder definir los pesos de las variables en el cálculo de la distancia euclidiana ponderada se adaptó el algoritmo GBHS.

Los resultados del modelo CLR propuesto en la presente investigación mostraron tener una mejor calidad que la presentada por un modelo base lineal, un modelo base no lineal y dos propuestas de la literatura que usan el enfoque CLR y CNLR. Los resultados se midieron con las métricas MAE, RMSE, NRMSE y el ±15% del PSI real. En todos los casos, el modelo CLR propuesto obtuvo mejores resultados; además de tener en cuenta la simplicidad de los modelos, ya que de esta manera la comprensión y el uso de modelos a nivel productivo es sencillo, sin necesidad de utilizar herramientas o frameworks complejos para hace uso productivo de los modelos.

El algoritmo propuesto obtuvo una mejora significativa en la reducción del tiempo de ejecución frente a las propuestas CLR y CNLR del estado del arte. La complejidad computacional en la Fase 1 es relativamente baja, mientras que la de la Fase 2 es alta, por lo que es preciso como trabajo futuro revisar formas más eficientes de realizar esta fase sin perder calidad en los resultados. La complejidad en la fase de uso productivo del modelo CLR propuesto es baja aunque también es susceptible de mejoras, por ejemplo haciendo uso de programación paralela.

En los modelos obtenidos, los atributos **aadt**, **truck** aparentan no tener relevancia en los modelos de regresión lineal obtenidos, esto se puede deber a la variable categoría que prioriza los segmentos de pavimentos y pueden tener mayor cantidad de reparaciones, que a su vez puede ser debido a la importancia de la carretera y el nivel de priorización del pavimento que se clasifica basado en la cantidad de tráfico de una vía. El atributo **elevation** no influye en el modelo ya que no afecta al deterioro del pavimento. Por otro lado, el atributo que más influye negativamente en todos los modelos son las abolladuras (**rut_depth**), es claro que entre más abolladuras tenga los segmentos de pavimento esto influirá de manera negativa en el valor del PSI.

Como trabajos futuros, se espera utilizar este algoritmo en otras aplicaciones o conjuntos de datos, por ejemplo, en la caficultura para poder predecir el rendimiento de los cultivos de café dependiendo de factores como el clima, características de las semillas, prácticas de cultivo y características del suelo. Siguiendo con el tema de pavimentos se espera aplicar este modelo utilizando datos de las carreteras de Colombia, se espera que los datos se puedan conseguir de las diferentes concesiones viales. Adicionalmente se espera modificar la metaheurística GBHS o utilizar otra metaheurística para obtener los pesos de los atributos.

CAPÍTULO 6

6 Referencias bibliográficas

[1] S. S. Jain, S. Aggarwal, and M. Parida, “HDM-4 Pavement Deterioration Models for Indian National Highway Network,” *J. Transp. Eng.*, vol. 131, no. 8, pp. 623–631, Aug. 2005, doi: 10.1061/(ASCE)0733-947X(2005)131:8(623).

[2] M. Y. Shahin, M. M. Nunez, M. R. Broten, S. H. Carpenter, and A. Sameh, *New techniques for modeling pavement deterioration*, no. 1123. 1987.

[3] R. Ramaswamy and M. Ben-Akiva, “Estimation of highway pavement deterioration from in-service pavement data,” *Transp. Res. Rec.*, vol. 1272, pp. 96–106, 1990.

[4] A. Alsherri and K. P. George, “Reliability Model for Pavement Performance,” *J. Transp. Eng.*, vol. 114, no. 3, pp. 294–306, May 1988, doi: 10.1061/(ASCE)0733-947X(1988)114:3(294).

[5] S. Terzi, “Modeling the Pavement Present Serviceability Index of Flexible Highway Pavements Using Data Mining,” *J. Appl. Sci.*, vol. 6, no. 1, pp. 193–197, Dec. 2005, doi: 10.3923/jas.2006.193.197.

[6] H. Späth, “Algorithm 39 Clusterwise linear regression,” *Computing*, vol. 22, no. 4, pp. 367–373, Dec. 1979, doi: 10.1007/BF02265317.

[7] M. Khadka, A. Paz, and A. Singh, “Generalised clusterwise regression for simultaneous estimation of optimal pavement clusters and performance models,” *Int. J. Pavement Eng.*, vol. 21, no. 9, pp. 1122–1134, Jul. 2020, doi: 10.1080/10298436.2018.1521970.

[8] K. S. Pratt, “Design Patterns for Research Methods: Iterative Field Research,” *AAAI Spring Symp. Exp. Des. Real*, no. 1994, pp. 1–7, 2009.

[9] M. Khadka, A. Paz, C. Arteaga, and D. K. Hale, “Simultaneous Generation of Optimum Pavement Clusters and Associated Performance Models,” *Math. Probl. Eng.*, vol. 2018, pp. 1–17, Dec. 2018, doi: 10.1155/2018/2159865.

[10] H. Späth, “A fast algorithm for clusterwise linear regression,” *Computing*, vol. 29, no. 2, pp. 175–181, Jun. 1982, doi: 10.1007/BF02249940.

[11] H. Späth, “Clusterwise linear least absolute deviations regression,”

Computing, vol. 37, no. 4, pp. 371–377, Dec. 1986, doi: 10.1007/BF02251095.

[12] W. S. DeSarbo and W. L. Cron, “A maximum likelihood methodology for clusterwise linear regression,” *J. Classif.*, vol. 5, no. 2, pp. 249–282, Sep. 1988, doi: 10.1007/BF01897167.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–22, 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.

[14] W. S. DeSarbo, R. L. Oliver, and A. Rangaswamy, “A simulated annealing methodology for clusterwise linear regression,” *Psychometrika*, vol. 54, no. 4, pp. 707–736, Sep. 1989, doi: 10.1007/BF02296405.

[15] M. Wedel and C. Kistemaker, “Consumer benefit segmentation using clusterwise linear regression,” *Int. J. Res. Mark.*, vol. 6, no. 1, pp. 45–59, Sep. 1989, doi: 10.1016/0167-8116(89)90046-3.

[16] M. Wedel and J.-B. E. M. Steenkamp, “A fuzzy clusterwise regression approach to benefit segmentation,” *Int. J. Res. Mark.*, vol. 6, no. 4, pp. 241–258, Jan. 1989, doi: 10.1016/0167-8116(89)90052-9.

[17] J.-B. E. M. Steenkamp and M. Wedel, “Fuzzy clusterwise regression in benefit segmentation: Application and investigation into its validity,” *J. Bus. Res.*, vol. 26, no. 3, pp. 237–249, Mar. 1993, doi: 10.1016/0148-2963(93)90034-M.

[18] A. M. Krieger and P. E. Green, “Modifying Cluster-Based Segments to Enhance Agreement with an Exogenous Response Variable,” *J. Mark. Res.*, vol. 33, no. 3, p. 351, Aug. 1996, doi: 10.2307/3152131.

[19] K. Lau, P. Leung, and K. Tse, “A mathematical programming approach to clusterwise regression model and its extensions,” *Eur. J. Oper. Res.*, vol. 116, no. 3, pp. 640–652, Aug. 1999, doi: 10.1016/S0377-2217(98)00052-6.

[20] M. P. Kamat and L. Mesquita, “Nonlinear mixed integer programming,” in *Advances in Design Optimization*, Chapman \& Hall London, 1994, pp. 174–193.

[21] M.-S. Yang and T.-S. Lin, “Fuzzy least-squares linear regression analysis for fuzzy input–output data,” *Fuzzy Sets Syst.*, vol. 126, no. 3, pp. 389–399, Mar. 2002, doi: 10.1016/S0165-0114(01)00066-5.

[22] M. Sakawa and H. Yano, “Multiobjective fuzzy linear regression analysis for fuzzy input-output data,” *Fuzzy Sets Syst.*, vol. 47, no. 2, pp. 173–181, Apr. 1992, doi: 10.1016/0165-0114(92)90175-4.

[23] C. Hennig, “Fixed Point Clusters for Linear Regression: Computation and Comparison,” *J. Classif.*, vol. 19, no. 2, pp. 249–276, Dec. 2002, doi:

10.1007/s00357-001-0045-7.

[24] C. Hennig, “Clusters, outliers, and regression: fixed point clusters,” *J. Multivar. Anal.*, vol. 86, no. 1, pp. 183–212, Jul. 2003, doi: 10.1016/S0047-259X(02)00020-9.

[25] C. Fraley and A. E. Raftery, “How many clusters? Which clustering method? Answers via model-based cluster analysis,” *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.

[26] M. J. Brusco, J. D. Cradit, and A. Tashchian, “Multicriterion Clusterwise Regression for Joint Segmentation Settings: An Application to Customer Value,” *J. Mark. Res.*, vol. 40, no. 2, pp. 225–234, May 2003, doi: 10.1509/jmkr.40.2.225.19227.

[27] W. S. DeSarbo and D. Grisaffe, “Combinatorial optimization approaches to constrained market segmentation: an application to industrial market segmentation,” *Mark. Lett.*, vol. 9, no. 2, pp. 115–134, 1998, doi: 10.1023/A:1007997714444.

[28] C. Preda and G. Saporta, “Clusterwise PLS regression on a stochastic process,” *Comput. Stat. Data Anal.*, vol. 49, no. 1, pp. 99–108, Apr. 2005, doi: 10.1016/j.csda.2004.05.002.

[29] C. Preda and G. Saporta, “Regression PLS sur un processus stochastique,” *Rev. Stat. appliquée*, vol. 50, no. 2, pp. 27–45, 2002.

[30] Z. Luo and E. Chou, “Pavement Condition Prediction Using Clusterwise Regression,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1974, no. 1974, pp. 70–77, Jan. 2006, doi: 10.3141/1974-11.

[31] P. D’Urso and A. Santoro, “Fuzzy clusterwise linear regression analysis with symmetrical fuzzy output variable,” *Comput. Stat. Data Anal.*, vol. 51, no. 1, pp. 287–313, Nov. 2006, doi: 10.1016/j.csda.2006.06.001.

[32] X. Wang, P. Zhao, G. Wang, and J. Liu, “Market segmentation based on customer satisfaction-loyalty links,” *Front. Bus. Res. China*, vol. 1, no. 2, pp. 211–221, May 2007, doi: 10.1007/s11782-007-0013-0.

[33] G. Schwarz, “Estimating the Dimension of a Model,” *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, Mar. 1978, doi: 10.1214/aos/1176344136.

[34] M. J. Brusco, J. D. Cradit, D. Steinley, and G. L. Fox, “Cautionary Remarks on the Use of Clusterwise Regression,” *Multivariate Behav. Res.*, vol. 43, no. 1, pp. 29–49, Mar. 2008, doi: 10.1080/00273170701836653.

[35] Z. Luo and H. Yin, “Probabilistic Analysis of Pavement Distress Ratings with the Clusterwise Regression Method,” *Transp. Res. Rec. J. Transp. Res.*

Board, vol. 2084, no. 1, pp. 38–46, Jan. 2008, doi: 10.3141/2084-05.

[36] J. Yang, J. Lu, M. Gunaratne, and B. Dietrich, “Modeling Crack Deterioration of Flexible Pavements: Comparison of Recurrent Markov Chains and Artificial Neural Networks,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1974, no. 1, pp. 18–25, Jan. 2006, doi: 10.3141/1974-05.

[37] L. He, G. H. Huang, and H. W. Lu, “Health-Risk-Based Groundwater Remediation System Optimization through Clusterwise Linear Regression,” *Environ. Sci. Technol.*, vol. 42, no. 24, pp. 9237–9243, Dec. 2008, doi: 10.1021/es800834x.

[38] M. Kagie, M. van der Loos, and M. van Wezel, “Including item characteristics in the probabilistic latent semantic analysis model for collaborative filtering,” *AI Commun.*, vol. 22, no. 4, pp. 249–265, 2009, doi: 10.3233/AIC-2009-0467.

[39] N. A. Tushar and D. K. Pratihar, “Design of cluster-wise optimal fuzzy logic controllers to model input-output relationships of some manufacturing processes,” *Int. J. Data Mining, Model. Manag.*, vol. 1, no. 2, p. 178, 2009, doi: 10.1504/IJDMMM.2009.026075.

[40] H. W. Suk and H. Hwang, “Regularized fuzzy clusterwise ridge regression,” *Adv. Data Anal. Classif.*, vol. 4, no. 1, pp. 35–51, Apr. 2010, doi: 10.1007/s11634-009-0056-5.

[41] A. E. Hoerl and R. W. Kennard, “Ridge Regression: Biased Estimation for Nonorthogonal Problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970, doi: 10.1080/00401706.1970.10488634.

[42] F. D. A. T. De Carvalho, G. Saporta, and D. N. Queiroz, “A clusterwise center and range regression model for interval-valued data,” in *Proceedings of COMPSTAT 2010 - 19th International Conference on Computational Statistics, Keynote, Invited and Contributed Papers*, 2010, pp. 461–468. doi: 10.1007/978-3-7908-2604-3-45.

[43] W. J. K. Khalid Al-Begain, Dieter Fiems, *Analytical and Stochastic Modeling Techniques and Applications*, vol. 5157 LNCS. 2008.

[44] S. Van Aelst, X. (Steven) Wang, R. H. Zamar, and R. Zhu, “Linear grouping using orthogonal regression,” *Comput. Stat. Data Anal.*, vol. 50, no. 5, pp. 1287–1312, Mar. 2006, doi: 10.1016/j.csda.2004.11.011.

[45] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the Number of Clusters in a Data Set Via the Gap Statistic,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 63, no. 2, pp. 411–423, Jul. 2001, doi: 10.1111/1467-9868.00293.

[46] L. A. García-Escudero, A. Gordaliza, A. Mayo-Iscar, and R. San Martín, “Robust clusterwise linear regression through trimming,” *Comput. Stat. Data*

Anal., vol. 54, no. 12, pp. 3057–3069, 2010, doi: 10.1016/j.csda.2009.07.002.

[47] L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar, “A general trimming approach to robust cluster Analysis,” *Ann. Stat.*, vol. 36, no. 3, pp. 1324–1345, Jun. 2008, doi: 10.1214/07-AOS515.

[48] P. D’Urso, R. Massari, and A. Santoro, “A class of fuzzy clusterwise regression models,” *Inf. Sci. (Ny).*, vol. 180, no. 24, pp. 4737–4762, Dec. 2010, doi: 10.1016/j.ins.2010.08.018.

[49] R. Schlittgen, “A weighted least-squares approach to clusterwise regression,” *AStA Adv. Stat. Anal.*, vol. 95, no. 2, pp. 205–217, Jun. 2011, doi: 10.1007/s10182-011-0155-4.

[50] D. J. Hand, “Finite Mixture and Markov Switching Models by Sylvia Frühwirth-Schnatter,” *Int. Stat. Rev.*, vol. 75, no. 2, pp. 255–255, Aug. 2007, doi: 10.1111/j.1751-5823.2007.00015_8.x.

[51] S. S. Gilan, A. M. Ali, and A. A. Ramezanianpour, “Evolutionary Fuzzy Function with Support Vector Regression for the Prediction of Concrete Compressive Strength,” in *2011 UKSim 5th European Symposium on Computer Modeling and Simulation*, Nov. 2011, pp. 263–268. doi: 10.1109/EMS.2011.28.

[52] J.-M. Poggi and B. Portier, “PM10 forecasting using clusterwise regression,” *Atmos. Environ.*, vol. 45, no. 38, pp. 7005–7014, Dec. 2011, doi: 10.1016/j.atmosenv.2011.09.016.

[53] S. Wang, G. H. Huang, and L. He, “Development of a clusterwise-linear-regression-based forecasting system for characterizing DNAPL dissolution behaviors in porous media,” *Sci. Total Environ.*, vol. 433, pp. 141–150, Sep. 2012, doi: 10.1016/j.scitotenv.2012.06.045.

[54] J. Muruzábal, D. Vidaurre, and J. Sánchez, “SOMwise regression: a new clusterwise regression method,” *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1229–1241, Sep. 2012, doi: 10.1007/s00521-011-0536-3.

[55] R. A. Carbonneau, G. Caporossi, and P. Hansen, “Extensions to the repetitive branch and bound algorithm for globally optimal clusterwise regression,” *Comput. Oper. Res.*, vol. 39, no. 11, pp. 2748–2762, Nov. 2012, doi: 10.1016/j.cor.2012.02.007.

[56] H.-F. Köhn, “Branch-and-bound applications in combinatorial data analysis,” *Psychometrika*, vol. 71, no. 2, pp. 411–413, Jun. 2006, doi: 10.1007/s11336-005-1378-7.

[57] M. J. Brusco, “A Repetitive Branch-and-Bound Procedure for Minimum Within-Cluster Sums of Squares Partitioning,” *Psychometrika*, vol. 71, no. 2, pp. 347–

363, Jun. 2006, doi: 10.1007/s11336-004-1218-1.

[58] T. Tan, H. W. Suk, H. Hwang, and J. Lim, “Functional fuzzy clusterwise regression analysis,” *Adv. Data Anal. Classif.*, vol. 7, no. 1, pp. 57–82, Mar. 2013, doi: 10.1007/s11634-013-0126-6.

[59] D. Vicari and M. Vichi, “Multivariate linear regression for heterogeneous data,” *J. Appl. Stat.*, vol. 40, no. 6, pp. 1209–1230, Jun. 2013, doi: 10.1080/02664763.2013.784896.

[60] A. M. Bagirov, J. Ugon, and H. Mirzayeva, “Nonsmooth nonconvex optimization approach to clusterwise linear regression problems,” *Eur. J. Oper. Res.*, vol. 229, no. 1, pp. 132–142, Aug. 2013, doi: 10.1016/j.ejor.2013.02.059.

[61] K. Venkatesh, V. Ravi, A. Prinzie, and D. Van den Poel, “Cash demand forecasting in ATMs by clustering and neural networks,” *Eur. J. Oper. Res.*, vol. 232, no. 2, pp. 383–392, Jan. 2014, doi: 10.1016/j.ejor.2013.07.027.

[62] D. Butina, “Unsupervised Data Base Clustering Based on Daylight’s Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets,” *J. Chem. Inf. Comput. Sci.*, vol. 39, no. 4, pp. 747–750, Jul. 1999, doi: 10.1021/ci9803381.

[63] R. A. Carbonneau, G. Caporossi, and P. Hansen, “Globally Optimal Clusterwise Regression By Column Generation Enhanced with Heuristics, Sequencing and Ending Subset Optimization,” *J. Classif.*, vol. 31, no. 2, pp. 219–241, Jul. 2014, doi: 10.1007/s00357-014-9155-x.

[64] W. Zhang and P. L. Durango-Cohen, “Explaining Heterogeneity in Pavement Deterioration: Clusterwise Linear Regression Model,” *J. Infrastruct. Syst.*, vol. 20, no. 2, Jun. 2014, doi: 10.1061/(ASCE)IS.1943-555X.0000182.

[65] S. Gharehbaghi and M. Khatibinia, “Optimal seismic design of reinforced concrete structures under time-history earthquake loads using an intelligent hybrid algorithm,” *Earthq. Eng. Eng. Vib.*, vol. 14, no. 1, pp. 97–109, Mar. 2015, doi: 10.1007/s11803-015-0009-2.

[66] K. De Roover, M. E. Timmerman, and E. Ceulemans, “How to detect which variables are causing differences in component structure among different groups,” *Behav. Res. Methods*, vol. 49, no. 1, pp. 216–229, Feb. 2017, doi: 10.3758/s13428-015-0687-8.

[67] A. M. Bagirov, J. Ugon, and H. G. Mirzayeva, “Nonsmooth Optimization Algorithm for Solving Clusterwise Linear Regression Problems,” *J. Optim. Theory Appl.*, vol. 164, no. 3, pp. 755–780, Mar. 2015, doi: 10.1007/s10957-014-0566-y.

[68] A. M. Bagirov, J. Ugon, and H. G. Mirzayeva, “An algorithm for clusterwise

linear regression based on smoothing techniques," *Optim. Lett.*, vol. 9, no. 2, pp. 375–390, Feb. 2015, doi: 10.1007/s11590-014-0749-3.

[69] F. Dotto, A. Farcomeni, L. A. García-Escudero, and A. Mayo-Iscar, "Robust Fuzzy Clustering via Trimming and Constraints," in *Advances in Intelligent Systems and Computing*, vol. 456, 2017, pp. 197–204. doi: 10.1007/978-3-319-42972-4_25.

[70] J. Kreger, L. Fischer, S. Hasler, T. H. Weisswange, and U. Bauer-Wersing, "A Priori Reliability Prediction with Meta-Learning Based on Context Information," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10614 LNCS, 2017, pp. 200–207. doi: 10.1007/978-3-319-68612-7_23.

[71] A. Frank and A. Asuncion, "{UCI} Machine Learning Repository," 2010. http://archive.ics.uci.edu/ml/index.php (accessed Aug. 15, 2020).

[72] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *J. Glob. Optim.*, vol. 16, no. 1, pp. 23–32, 2000, doi: 10.1023/A:1008324625522.

[73] T. F. Wilderjans, E. Vande Gaer, H. A. L. Kiers, I. Van Mechelen, and E. Ceulemans, "Principal Covariates Clusterwise Regression (PCCR): Accounting for Multicollinearity and Population Heterogeneity in Hierarchically Organized Data," *Psychometrika*, vol. 82, no. 1, pp. 86–111, Mar. 2017, doi: 10.1007/s11336-016-9522-0.

[74] A. M. Bagirov, A. Mahmood, and A. Barton, "Prediction of monthly rainfall in Victoria, Australia: Clusterwise linear regression approach," *Atmos. Res.*, vol. 188, pp. 20–29, May 2017, doi: 10.1016/j.atmosres.2017.01.003.

[75] A. M. Bagirov and S. Taheri, "DC Programming Algorithm for Clusterwise Linear $${{\varvec{L}}}_\mathbf{1}$$ Regression," *J. Oper. Res. Soc. China*, vol. 5, no. 2, pp. 233–256, Jun. 2017, doi: 10.1007/s40305-017-0151-9.

[76] M. Rump, W. Esdar, and E. Wild, "Individual differences in the effects of academic motivation on higher education students' intention to drop out," *Eur. J. High. Educ.*, vol. 7, no. 4, pp. 341–355, Oct. 2017, doi: 10.1080/21568235.2017.1357481.

[77] Y. W. Park, Y. Jiang, D. Klabjan, and L. Williams, "Algorithms for Generalized Clusterwise Linear Regression," *INFORMS J. Comput.*, vol. 29, no. 2, pp. 301–317, May 2017, doi: 10.1287/ijoc.2016.0729.

[78] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-Price: Column Generation for Solving Huge Integer Programs," *Oper. Res.*, vol. 46, no. 3, pp. 316–329, Jun. 1998, doi: 10.1287/opre.46.3.316.

[79] U. Maulik and S. Bandyopadhyay, “Genetic algorithm-based clustering technique,” *Pattern Recognit.*, vol. 33, no. 9, pp. 1455–1465, Sep. 2000, doi: 10.1016/S0031-3203(99)00137-5.

[80] M. Khadka and A. Paz, “Comprehensive Clusterwise Linear Regression for Pavement Management Systems,” *J. Transp. Eng. Part B Pavements*, vol. 143, no. 4, p. 4017014, Dec. 2017, doi: 10.1061/JPEODX.0000009.

[81] A. Gerami Matin, R. Vatani Nezafat, and A. Golroo, “A comparative study on using meta-heuristic algorithms for road maintenance planning: Insights from field study in a developing country,” *J. Traffic Transp. Eng. (English Ed.*, vol. 4, no. 5, pp. 477–486, Oct. 2017, doi: 10.1016/j.jtte.2017.06.004.

[82] D. Whitley, “A genetic algorithm tutorial,” *Stat. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994, doi: 10.1007/BF00175354.

[83] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43. doi: 10.1109/MHS.1995.494215.

[84] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[85] J. E. Alvarez-Benitez, R. M. Everson, and J. E. Fieldsend, “A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts,” in *International conference on evolutionary multi-criterion optimization*, 2005, pp. 459–473. doi: 10.1007/978-3-540-31880-4_32.

[86] E. G. Gil, Eduardo, *The Mathematics of the Uncertain A Tribute to Pedro Gil*. 2018.

[87] B. Gaël, A. Hanane, B. Stéphanie, L. Mustapha, and N. Ndèye, “A New Micro-Batch Approach for Partial Least Square Clusterwise Regression,” *Procedia Comput. Sci.*, vol. 144, pp. 239–250, 2018, doi: 10.1016/j.procs.2018.10.525.

[88] A. M. Bagirov and J. Ugon, “Nonsmooth DC programming approach to clusterwise linear regression: optimality conditions and algorithms,” *Optim. Methods Softw.*, vol. 33, no. 1, pp. 194–219, Jan. 2018, doi: 10.1080/10556788.2017.1371717.

[89] R. A. M. da Silva and F. de A. T. de Carvalho, “On Combining Fuzzy C-Regression Models and Fuzzy C-Means with Automated Weighting of the Explanatory Variables,” in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Jul. 2018, vol. 2018-July, pp. 1–8. doi: 10.1109/FUZZ-IEEE.2018.8491476.

[90] I. Gitman, J. Chen, E. Lei, and A. Dubrawski, “Novel Prediction Techniques

Based on Clusterwise Linear Regression," *arXiv Prepr. arXiv1804.10742*, 2018.

[91] S. Bougeard, V. Cariou, G. Saporta, and N. Niang, "Prediction for regularized clusterwise multiblock regression," *Appl. Stoch. Model. Bus. Ind.*, vol. 34, no. 6, pp. 852–867, Nov. 2018, doi: 10.1002/asmb.2335.

[92] S. Bougeard, H. Abdi, G. Saporta, and N. Niang, "Clusterwise analysis for multiblock component methods," *Adv. Data Anal. Classif.*, vol. 12, no. 2, pp. 285–313, Jun. 2018, doi: 10.1007/s11634-017-0296-8.

[93] G. Galimberti and G. Soffritti, "Seemingly unrelated clusterwise linear regression," *Adv. Data Anal. Classif.*, vol. 14, no. 2, pp. 235–260, Jun. 2020, doi: 10.1007/s11634-019-00369-4.

[94] N. Veeramisti, A. Paz, M. Khadka, and C. Arteaga, "A clusterwise regression approach for the estimation of crash frequencies," *J. Transp. Saf. Secur.*, vol. 13, no. 3, pp. 247–277, Mar. 2021, doi: 10.1080/19439962.2019.1611681.

[95] F. Torti, D. Perrotta, M. Riani, and A. Cerioli, "Assessing trimming methodologies for clustering linear regression data," *Adv. Data Anal. Classif.*, vol. 13, no. 1, pp. 227–257, Mar. 2019, doi: 10.1007/s11634-018-0331-4.

[96] G. Galimberti, L. Nuzzi, and G. Soffritti, "Covariance matrix estimation of the maximum likelihood estimator in multivariate clusterwise linear regression," *Stat. Methods Appt.*, vol. 30, no. 1, pp. 235–268, Mar. 2021, doi: 10.1007/s10260-020-00523-9.

[97] G. C. Chow, "Maximum-likelihood estimation of misspecified models," *Econ. Model.*, vol. 1, no. 2, pp. 134–138, Apr. 1984, doi: 10.1016/0264-9993(84)90001-4.

[98] K. Joki, A. M. Bagirov, N. Karmitsa, M. M. Mäkelä, and S. Taheri, "Double Bundle Method for finding Clarke Stationary Points in Nonsmooth DC Programming," *SIAM J. Optim.*, vol. 28, no. 2, pp. 1892–1919, Jan. 2018, doi: 10.1137/16M1115733.

[99] G. P. Oliveira, M. D. Santos, and W. L. Roque, "Constrained clustering approaches to identify hydraulic flow units in petroleum reservoirs," *J. Pet. Sci. Eng.*, vol. 186, p. 106732, Mar. 2020, doi: 10.1016/j.petrol.2019.106732.

[100] R. Di Mari, R. Rocci, and S. A. Gattone, "Scale-constrained approaches for maximum likelihood estimation and model selection of clusterwise linear regression models," *Stat. Methods Appt.*, vol. 29, no. 1, pp. 49–78, Mar. 2020, doi: 10.1007/s10260-019-00480-y.

[101] R. Rocci, S. A. Gattone, and R. Di Mari, "A data driven equivariant approach to constrained Gaussian mixture modeling," *Adv. Data Anal. Classif.*, vol. 12,

no. 2, pp. 235–260, Jun. 2018, doi: 10.1007/s11634-016-0279-1.

[102] W. Aleadelat and K. Ksaibati, “Estimation of Pavement Serviceability Index Through Android-Based Smartphone Application for Local Roads,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2639, no. 1, pp. 129–135, Jan. 2017, doi: 10.3141/2639-16.

[103] T. Bailey and A. K. Jain, “A Note on Distance-Weighted k-Nearest Neighbor Rules,” *IEEE Trans. Syst. Man. Cybern.*, vol. 8, no. 4, pp. 311–313, 1978, doi: 10.1109/TSMC.1978.4309958.

[104] M. G. H. Omran and M. Mahdavi, “Global-best harmony search,” *Appl. Math. Comput.*, vol. 198, no. 2, pp. 643–656, May 2008, doi: 10.1016/j.amc.2007.09.004.

[105] F. Chao, D. Zhou, C.-M. Lin, C. Zhou, M. Shi, and D. Lin, “Fuzzy cerebellar model articulation controller network optimization via self-adaptive global best harmony search algorithm,” *Soft Comput.*, vol. 22, no. 10, pp. 3141–3153, May 2018, doi: 10.1007/s00500-017-2864-4.

[106] Siddhartha, “Time Complexities Of ML Algorithms,” *7 Hidden Layers*, 2021.

[107] team developer Waikato, “Weka 3.8 is the latest stable version,” *The workbench for machine learning*, 2019. https://www.cs.waikato.ac.nz/ml/weka/