INVESTIGACIÓN e INNOVACIÓN en INGENIERÍAS
ISSN2344-8652

Algoritmo greedy para la predicción del índice de servicio de pavimento basado en agrupación y regresión lineal

Greedy algorithm to predict the pavement serviceability index based on clustering and linear regression

Francisco Anacona Campo **Carlos Cobos Lozada** **Martha Mendoza Becerra**
Universidad del Cauca, Colombia

OPEN ACCESS

Recibido: 22/09/2020
Aceptado: 22/10/2020
Publicado: 23/11/2020

Correspondencia de autores:
javieranacona@unicauca.edu.co

Resumen

Objetivo: Proponer un algoritmo CLR (Clusterwise Linear Regression) que realiza agrupamiento divisivo de muestras de segmentos de pavimentos utilizando modelos de regresión lineal y define automáticamente el número de agrupaciones con el fin de predecir el índice de capacidad de servicio del pavimento (pavement serviceability index, PSI). **Metodología**: Basado en el proceso de investigación iterativa propuesto por Pratt se desarrollaron dos ciclos de mejora del algoritmo propuesto. El primer ciclo permitió obtener una versión inicial, aplicarlo sobre los datasets de entrenamiento y prueba y observar las mejoras que se debían realizar. **Resultados:** Se obtuvo un modelo compuesto por tres grupos de muestras de segmentos de pavimento con sus correspondientes modelos de regresión lineal multivariable (atributos mixtos) que permiten predecir el PSI de una muestra de pavimento. **Conclusiones:** El modelo se obtuvo con menor tiempo de cómputo (15,6 veces menos tiempo que el reportado por el estado del arte) y presenta mejores resultados en sencillez en comparación con los modelos lineales y no lineales reportados en la literatura, además, en calidad tiene resultados similares (incluso mejores en algunas métricas) al modelo lineal y es competitivo frente al modelo no lineal.

Palabras clave: Regresión lineal, agrupamiento, pavimento, algoritmo.

Abstract

Objective: To propose a CLR (Clusterwise Linear Regression) algorithm that carries out a divisive grouping of pavement segment samples using linear regression models and automatically defines the number of groupings in order to predict the pavement serviceability index (PSI). **Methodology:** Based on the iterative research process proposed by Pratt, two improvement cycles of the proposed algorithm were developed. The first cycle allowed us to obtain an initial version, apply it to the training and test datasets, and observe the improvements that had to be made. **Results:** A model composed of three groups of pavement segment samples was obtained with their corresponding multivariate linear regression models (mixed attributes) that allow predicting the PSI of a pavement sample. **Conclusions:** The model was obtained with less computing time (15.6 times less time than that reported by the state of the art) and presents better results in simplicity compared to the linear and non-linear models reported in the literature, in addition, it has quality results similar (even better in some metrics) to the linear model and is competitive against the non-linear model.

Keywords: Linear Regression, clustering, pavement, algorithm.

Como citar (IEEE): 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", vol. 8, n°3, pp. 119-134, 2020. DOI: https://doi.org/10.17081/invinno.8.3.4708

Introducción

La gestión apropiada de los pavimentos es clave para la reducción de accidentes de tránsito, la disminución de costos de reparación y mantenimiento, y el aumento de la vida útil de los tramos de pavimento. El deterioro del pavimento depende de varios factores, las propiedades especificas del pavimento, el paso del tiempo, las condiciones de tráfico, el clima, entre otros [1]. Una herramienta útil para la adecuada gestión del pavimento, son los modelos de predicción del comportamiento del pavimento, los cuales deben ser precisos y calibrados a las diferentes condiciones a las que están expuestas las rutas viales [2].

Los modelos que miden el comportamiento del pavimento pueden predecir el índice de capacidad de servicio del pavimento (Pavement Serviceability Index, PSI), una medida con la cual se pronostica el deterioro del pavimento y se puede alertar anticipadamente cuales rutas deben ser intervenidas y el tipo de procedimiento que se les debe realizar [3]. Existen diferentes métodos de modelamiento empíricos y mecanicistas. La literatura reporta varios modelos que incluyen pocas variables explicativas y otros modelos que por ejemplo usan redes neuronales artificiales, que producen resultados con calidad insuficiente cuando se aplica a datos reales [4], por lo que no son utilizados en la práctica. En otros trabajos se han formulado modelos más complejos, sin embargo las estimaciones de la importancia de las variables, en algunos casos, tienen resultados opuestos a los esperados [5]. Además, en estos trabajos se requiere una intensiva participación de expertos que ayuden a discriminar variables de poca relevancia con el fin de mejorar la predicción del PSI. Para poder construir estos modelos, se necesita tomar muestras historias de diferentes segmentos de pavimentos y con ellos crear un conjunto de datos de entrenamiento [6].

El PSI se calcula por segmentos de carretera, ya que cada segmento a pesar de ser construido con el mismo material y en fechas similares, puede estar expuesto a diferentes condiciones y haber sido sujeto a diferente número de reparaciones o mantenimientos [7]. Por otro lado, determinar el PSI en forma individual es un proceso complejo y costoso, por esto, se han desarrollado diferentes investigaciones, donde se agrupan segmentos de pavimento con factores y condiciones similares, y de esta manera se obtiene un modelo para cada grupo que puede ser aplicado individualmente a los segmentos que lo conforman o que son similares.

La literatura reporta dos tipos de modelos de predicción: probabilísticos y deterministas. Un modelo determinista es una función matemática formal y precisa que busca predecir el PSI [8]. Un modelo probabilístico hace referencia a una función de probabilidad que busca predecir el PSI basado en un análisis del rendimiento histórico de tramos de pavimentos. Este es el modelo utilizado en este artículo. Entre los modelos probabilísticos se han propuesto métodos de agrupación empíricos usando modelos lineales y no lineales del comportamiento del pavimento, entre ellos se destaca la generación de modelos de regresión lineal por clusterwise (CLR) [9], una técnica que combina agrupamiento y regresión para crear subconjuntos de observaciones mutuamente excluyentes de un conjunto de datos [10] y el uso de algoritmos meta-heurísticos para ajustar los valores de los modelos de predicción con el fin de hacerlos más precisos [11,12,1314]. En estos trabajos se busca obtener modelos lineales o no lineales de cada grupo basado en recocido simulado o variaciones de este, que sean más precisos y que puedan ser aplicados a nuevos segmentos de carretera con el fin de predecir el valor de PSI para cada grupo.

En otras áreas también se usa CLR y conceptos aplicados allí se pueden trasladar al problema de predecir el PSI, a continuación se resumen algunos de ellos organizados por año de publicación.

Por ejemplo, para 2015 se seleccionado dos trabajos. En [15] donde se usa una combinación del algoritmo k-means combinado con recocido simulado y wavelet weighted least squares support vector machine para definir un algoritmo CLR que permite evaluar la respuesta de una estructura de concreto reforzada frente

a un terremoto. Y en [16] donde se propone un algoritmo incremental que en conjunto con técnicas de suavizado resuelve problemas CLR. El algoritmo divide incrementalmente el conjunto de datos en grupos que pueden aproximarse fácilmente mediante una función de regresión lineal. Presentan un procedimiento especial para generar una solución inicial e ir buscando el óptimo global en cada iteración.

En 2017, se incluyen tres trabajos previos, el primero, en [17] donde presentan el uso de CLR para la predicción de la precipitación mensual en Victoria (Australia) con datos que incluyen 5 variables meteorológicas medidas desde 1889 hasta 2014. Los resultados con CLR superan en diversas localizaciones al algoritmo de maximización de expectativas, la regresión lineal múltiple, una rede neuronal artificial y una máquina de soporte vectorial para modelos de regresión. El segundo en [18] donde proponen el uso de CLR para determinar la relación entre la motivación académica y la intención de abandonar estudios en instituciones de educación superior alemanas. Y el tercero en [19] donde proponen un CLR generalizado donde una entidad puede tener más de una observación e implementan y comparan tres algoritmos en un problema de mantenimiento de existencia de unidades empleado para pronosticar efectos de halo y canibalización en promociones utilizando datos reales de minoristas de una gran cadena de supermercados.

En 2018 se seleccionan cuatro trabajos, a saber: En [20] formulan el problema CLR como un problema de optimización no convexo y no suave utilizando la función de error de regresión al cuadrado. En este problema, la función objetivo se representa como una diferencia de funciones convexas y se definen condiciones de optimalidad. Presentan además un algoritmo con un enfoque incremental para generar soluciones y se evalúan con conjuntos de datos de diferente tamaño. En [21] se presenta un método CLR difuso basando en modelos de regresión difusos y Fuzzy C-means como algoritmo de clustering. Además definen un procedimiento heurístico para afinar los parámetros y obtener modelos más útiles en la tarea de predicción. En [22] complementan el CLR para definir el modelo usar en un nuevo datos con dos posibles estrategias, una con un proceso de clasificación (CLR predictivo) y otra basada en restricciones definidas por el usuario (CLR restrictivo). La experimentación muestra que los dos modelos son mejores al CLR tradicional y a los bosques aleatorizados en conjuntos de datos del repositorio de la UCI. Y en [23] proponen un algoritmo denominado CW.rMBREG que arma grupos y a la vez con un modelo basado en componentes que ellos denominan multi bloque que tiene un criterio bien definido obtienen buenos resultados en un entorno simulado de calidad de aire interior.

En 2019 se destacan dos trabajos. En [24] donde proponen un algoritmo CLR basado en recocido simulado acoplado con la estimación de la máxima verosimilitud con el objetivo de estimar choques de autos y obtienen mejores resultados que los reportados en el estado del arte. Y en [25] donde proponen un algoritmo denominado TCWRM para estimar valores basado en la metodología CLR. Esta propuesta se basa en un algoritmo previo denominado TCLUST-REG al cual le hacen adaptable la fase de creación del segundo recorte y obtienen resultados competitivos.

En 2020 se resaltan 4 investigaciones, a saber: En [26] modifican la matriz de covarianza que usa el algoritmo de maximización de expectativas comúnmente usado en CLR y logran mostrar que es útil y eficaz en aplicaciones prácticas. En [10] se propone un algoritmo CLR basado en regresión lineal de soporte vectorial y se evalua sobre seis datasets sintéticos, donde se muestra que el algoritmo es eficiente y robusto. En [27] se propone el uso de un algoritmo CLR restringido para abordar el problema de identificar unidades de flujo hidráulico en yacimientos de petróleo. Y en [28] se propone una forma de comparar modelos CLR basándose en un nuevo esquema de regularización de la máxima verosimilitud usada por el criterio de información Bayesiano (BIC).

El presente artículo muestra un algoritmo que permite definir automáticamente el número de agrupaciones (grupos) de un conjunto de datos de entrenamiento con información de segmentos de pavimento que incluye 14 variables independientes (10 continuas y 4 categóricas) y el valor real de PSI. Para cada grupo se crean modelos de regresión lineal que permiten predecir el valor de PSI de un segmento de pavimento. El algoritmo propuesto es greedy (voraz o codicioso), obtiene resultados competitivos en calidad comparados con el estado del arte, pero en tiempos de cómputo mucho menores que las propuestas más recientes.

El resto del artículo se encuentra organizado de la siguiente manera. La sección 2 describe la metodología seguida resaltando el origen de los conjuntos de datos de entrenamiento y prueba, junto con sus atributos y el número de instancias (registros) en cada uno, además del algoritmo de CLR (agrupamiento de las instancias basado en la calidad de los modelos de regresión lineal que se obtiene para cada grupo) que se realizó. La sección 3 muestra los resultados, presentando primero la forma como se definieron los modelos con el conjunto de datos de entrenamiento, la validación de los modelos obtenidos sobre el conjunto de datos de validación y la comparación de los resultados con el estado del arte. Finalmente, en la sección 4 se presentan las conclusiones del trabajo realizado y las líneas de trabajo futuro que el grupo de investigación espera desarrollar en el futuro cercano.

Metodología

Para el desarrollo de esta investigación se tomó como guía el Patrón de Investigación Iterativa (PII) propuesto por Pratt [29]. Un patrón diseñado para orientar el desarrollo de proyectos de investigación que involucran una solución computacional y que contempla la realización de ciclos compuestos de cuatro etapas: observación del problema, identificación del problema, desarrollo de la solución y prueba de la solución. Buscando que al terminar cada ciclo se entregue un artefacto o producto concreto.

En esta investigación se desarrollaron dos ciclos. El primer ciclo permitió obtener varios productos, a saber: 1) una revisión de la literatura que se obtuvo usando las pautas presentadas por [30]; 2) la consecución de dos algoritmos del estado del arte implementados en R y su ejecución para comprobar su buen funcionamiento y la revisión de la repetibilidad de los resultados reportados en la literatura sobre los conjuntos de datos usados en el proyecto; 3) La realización de una primer versión del algoritmo (o versión inicial) con enfoque greedy y su aplicación sobre los datasets de entrenamiento y prueba, lo que permitió definir las mejoras que se debían realizar.

En el segundo ciclo se obtuvo la versión final del algoritmo greedy que se esperaba obtener, se le afinaron los parámetros (dos en esta versión final) y se compararon los resultados con los reportados en el estado del arte usando varias métricas (MEA, RMS, NRMS, la cantidad de predicciones que se encuentran en el rango del valor de PSI real ±15% y el tiempo). A continuación se describen los conjuntos de datos de entrenamiento y validación usados y se presenta en detalle el algoritmo propuesto.

Descripción de los conjuntos de datos

Los datos de entrenamiento y prueba se tomaron del departamento de transporte de Nevada [13]. El conjunto de datos de entrenamiento cuenta con 14638 instancias recolectados desde el año 2001 hasta el año 2010. El conjunto de datos de validación cuenta con 3005 instancias recolectados de 2011 a 2012. Durante el preprocesamiento de los conjuntos de datos se excluyeron las variables SAMPLE_ID y TIME_PERIOD debido a que no aportan información relevante al proceso de agrupación (el primero es el identificador del

número de muestra de un determinado segmento y el segundo es la cantidad de muestras tomadas para un segmento). A continuación, en la Tabla 1 se describen las variables que fueron tomadas en cuenta junto con su descripción.

Tabla 1. Coeficiente correlacional promediado

Variable	Descripción	Tipo de variable
PSI	Representa la medida del estado actual de la muestra del segmento de carretera entre 0 y 5.	De clase (numérica)
1. AGE	Es la edad o época en la que la muestra del segmento de carretera tuvo una intervención por reparación o mantenimiento.	Explicativa continua
2. 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.	
3. TRUCKS	El tráfico promedio de camiones a la que está expuesto el segmento de carretera.	
4. ELEVATION	Es la elevación en el punto medio de un segmento en metros.	
5. PRECIP	Representa la precipitación media anual en cm.	
6. MIN_TEMP	Es la temperatura media anual mínima del aire en grados centígrados.	
7. MAX_TEMP	Es la temperatura media anual máxima del aire en grados centígrados.	
8. WET_DAYS	Es el número total de días húmedos en el transcurso del año.	
9. FREEZE_THAW	Es el número ciclos totales de congelación o descongelación que experimenta un pavimento en el transcurso del año.	
10. RUT_DEPTH	Representa la profundidad promedio de los huecos de la muestra de un segmento de carretera.	
11. NUMBER_OF_LANES	Representa el número de carriles que tiene un segmento de carretera.	Explicativa categórica (factor)
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.	

Fuente: Elaboración Propia

Algoritmo propuesto para pronosticar PSI

El algoritmo tiene como objetivo agrupar las 14638 instancias de muestras de segmentos de pavimentos en el conjunto de entrenamiento con el fin de obtener los modelos de predicción de PSI que sean más precisos para cada grupo. El pseudocódigo se muestra en la Figura 1 y a continuación se explica en detalle.

Lo primero que realiza el algoritmo (línea 1), es cargar el dataset (10 variables continuas, 4 variables categóricas y la variable objetivo PSI) desde un archivo ARFF (formato nativo de Weka). En la línea 2 se inicializa la lista de GruposAceptados, una lista donde se organizan las instancias en grupos (clústeres) y todos estos grupos cumplen con un coeficiente de correlación mínimo esperado (CoeficienteCorrelacionMinimo) que ha sido definido por el usuario con el parámetro correspondiente. En esta misma línea 2, se inicializa la lista de ModelosAceptados, una lista de los modelos de regresión lineal de cada uno de los grupos aceptados que están registrados en la lista (GruposAceptados). Cada modelo, cuenta con los coeficientes para cada una de las variables explicativas, si la variable fue o no seleccionada, el coeficiente de correlación, entre otros valores.

Las líneas 4 a 15 encierran la fase 1 del proceso, un proceso iterativo de división del dataset en grupos de la mayor calidad posible. Este proceso termina cuando el grupo que se encuentra no tiene la calidad suficiente para ser aceptado en la lista de GruposAceptados.

Figura 1. Pseudocódigo del algoritmo propuesto para el entrenamiento

Entradas:
- NombreDataSet: Nombre del archivo ARFF con los datos.
- MinNumRegPorGrupo; Mínimo número de instancias que debe tener un grupo para ser aceptado.
- ErrorPromedioAceptado: Permite seleccionar las filas que más se ajustan al modelo de regresión lineal y serán incluidas en un grupo de mayor calidad.
- ErrorPromedioAdicional: Cada vez que se crea un siguiente grupo se reduce la expectativa de calidad de las instancias que se incluirán en este.
- CoeficienteCorrelacionMinimo: El mínimo coeficiente de correlación que debe tener un grupo para ser aceptado. Por defecto igual a 0,88.

Salidas:
- GruposAceptados: Una lista de grupos en archivos ARFF.
- ModelosAceptados: Modelos de regresión lineal de cada uno de los grupos con sus atributos seleccionados, coeficientes y valores de correlación.

```
    Inicio
1.      DataSet = CargarDatos (NombreDataSet)
2.      GruposAceptados = Ø; ModelosAceptados = Ø
3.      // Fase 1: Creación de grupos base de la mayor calidad posible
4.      Hacer
5.          Modelo = CalcularModeloRegresionLineal (DataSet)
6.          MejoresInstancias = OrdenarDatosPorErrorDePrediccion (DataSet, Modelo,
                                ErrorPromedioAceptado) //menor a mayor
7.          Grupo = CrearGrupo (DataSet, MejoresInstancias)
8.          ModeloGrupo = CalcularModeloRegresionLineal (Grupo)
9.          Si ModeloGrupo.CoeficienteCorrelacion > CoeficienteCorrelacionMinimo Hacer
10              GruposAceptados.Adicionar(Grupo)
11              ModelosAceptados.Adicionar(ModeloGrupo)
12              DataSet = DataSet – Grupo.Instancias
13              ErrorPromedioAceptado = ErrorPromedioAceptado + ErrorPromedioAdicional
14          Fin Si
15      Mientras ModeloGrupo.CoeficienteCorrelacion <= CoeficienteCorrelacionMinimo
16      // Fase 2: Asignación de instancias huérfanas al modelo donde reducen menos calidad
17      Si GruposAcetados.Total >= 2 Hacer
18          Para i = 0 Hasta DataSet.Filas Hacer
19              IdG = BuscarMejorGrupo (GruposAceptados, DataSet [i])
20              IncluirInstanciaEnGrupoAceptado (GruposAceptados[IdG], DataSet[i])
21              ModelosAceptados [IdG] = CalcularModeloRegresionLineal (GruposAceptados[IdG])
22          Fin Para
23      Retornar GruposAceptados, ModelosAceptados
    Fin
```

Fuente: Elaboración Propia

En la línea 5 se genera un modelo de regresión lineal para el Dataset, este modelo se crea con la función LinearRegression de Weka usando los valores por defecto ("-S 0 -R 1.0E-8 -num-decimal-places 4"), esto es, haciendo selección de atributos con el algoritmo MD5, un Ridge (factor que regula el sobreajuste de los coeficientes) de 1,0E-8 y 4 números decimales para la salida de los resultados. Con base en este Modelo obtenido, en la línea 6 se ordenan las instancias del Dataset de menor a mayor error absoluto de predicción, es decir, el valor absoluto de la diferencia entre el valor predicho por el modelo y el valor real de cada instancia en el DataSet. Ya ordenado el DataSet se calcula el número de instancias (MejoresInstancias) ubicados en las primeras filas del DataSet que tienen un error absoluto promedio menor que el parámetro ErrorPromedioAceptado, parámetro que define el usuario. Este valor de MejoresInstancias debe ser mayor o igual al parámetro MinNumRegPorGrupo y obviamente menor igual al total de filas en el DataSet.

En la línea 7 se crea un grupo (un dataset más pequeño) con las primeras MejoresInstancias del DataSet completo. En la línea 8, a este Grupo se le crea un modelo de regresión lineal (ModeloGrupo) con la misma configuración previamente explicada para Weka. Este ModeloGrupo puede cumplir o no con la mínima calidad requerida (parámetro CoeficienteCorrelacionMinimo), lo cual se pregunta en la línea 9; en caso de tener un coeficiente de correlación superior al mínimo, la variable Grupo y el ModeloGrupo se agregan a la lista de GruposAceptados y a la lista de ModelosAceptados en las líneas 10 y 11 respectivamente.

Continuando con la línea 12, al Dataset se le quitan las instancias del grupo que se formó y que aprobó el criterio de calidad, de esta forma se busca en la siguiente iteración, encontrar un grupo de calidad para el resto del DataSet que entra a esa iteración. Para la formación del siguiente grupo, se relaja el valor de ErrorPromedioAceptado, esto es, se le agrega el valor del parámetro ErrorPromedioAdicional (línea 13), con lo cual se logra que en la línea 6 puedan entrar un mayor número de filas al grupo candidato a ser aceptado en la siguiente iteración.

Este ciclo hacer mientras se rompe (línea 15) cuando no se encuentra un grupo que cumpla con la calidad mínima para ser parte de los GruposAceptados.

Cuando se termina la fase 1 (líneas 4 a 15), se inicia la fase 2 (líneas 17 a 22), en la que se busca asignar las instancias huérfanas (las que terminaron quedando en el DataSet), aquellas instancias que quedaron sin grupo porque el grupo al que pertenencia con cumplía con la calidad mínima para ser aceptado. Lo primero que se evalúa, es que existan por lo menos dos (2) GruposAceptados (línea 17). Puede ocurrir que el usuario defina parámetros muy altos para el CoeficienteCorrelacionMinimo y muy bajos para el ErrorPromedioAceptado, y como resultados no se cree ningún grupo o uno solo.

Asegurada la existencia de por lo menos dos (2) GruposAceptados, en las líneas 18 a 22 se recorren uno a uno las instancias huérfanas, se les busca (línea 19) el grupo donde tienen menos efectos negativos en el coeficiente de correlación (o en su defecto que tenga efectos positivos) para incluir la instancia en ese grupo (línea 20) y proceder a actualizar el modelo del grupo en la línea 21.

Al finalizar la fase 2 (líneas 17 a 22) se tienen todas las instancias del DataSet original organizados en los GruposAceptados y cada uno de estos grupos tienen sus modelos de regresión lineal (ModelosAceptados) actualizados con las instancias de cada grupo, que son las dos variables que retorna el algoritmo.

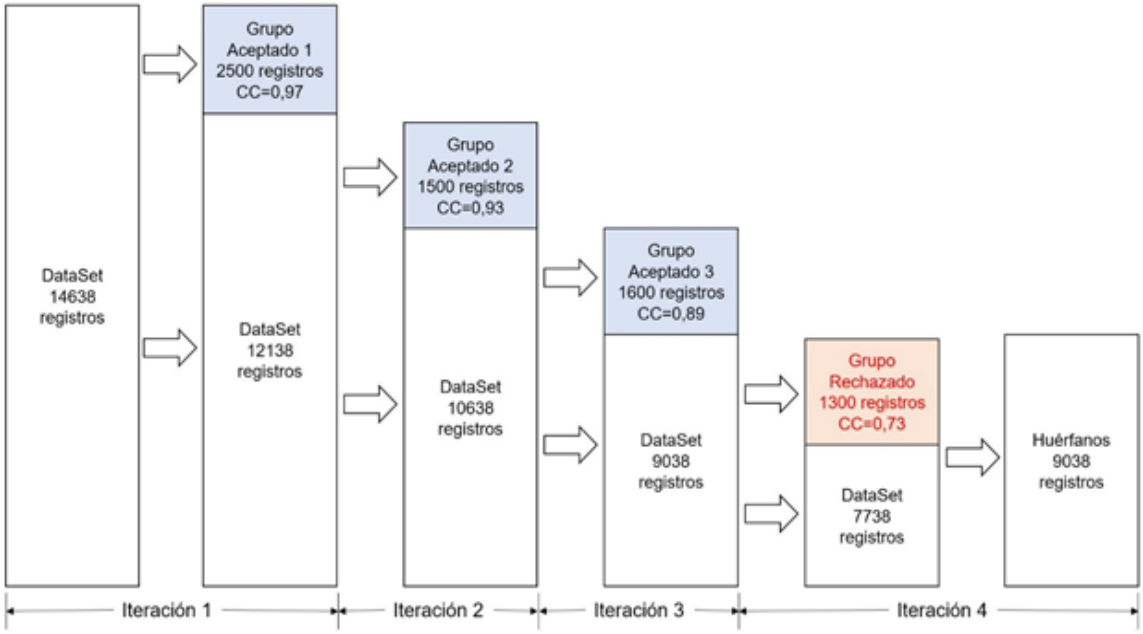
La Figura 2 muestra un ejemplo de la fase 1 del algoritmo. En la iteración 1 se inicia con un DataSet de 14638 instancias, el cual se divide en dos, un primer grupo que por su alta calidad (Coeficiente de Correlación, CC, de 0,97) se acepta y se incluye en GruposAceptados y el resto del DataSet (12138 instancias) que se convierte en el origen de datos para la iteración 2. En esta segunda iteración, el DataSet se vuelve a dividir en dos, un grupo de 1500 instancias que también tiene alta calidad (0,93) por lo que entra en los GruposAceptados,

contando ahora con dos grupos aceptados. El resto del DataSet se convierte en el origen de datos de la iteración 3. En esta tercera iteración, así como en la anterior se ha disminuido un poco el criterio de selección de las instancias que forman el grupo, razón por la cual, es normal que se acepten grupos con un coeficiente de correlación menor. El grupo que se obtiene en esta iteración tiene 1600 instancias y un coeficiente de correlación de 0,89 que todavía le alcanza para hacer parte de los GruposAceptados, ya que el CoeficienteCorrelacionMinimo de este ejemplo es de 0,88. El resto del DataSet ahora se convierte en la entrada de la siguiente iteración. En esta iteración 4, el grupo que se armó con base en las filas que tienen menor error absoluto en relación con el modelo lineal del Dataset, tiene 1300 instancias, pero su coeficiente de correlación es menor a 0,86, por esta razón no ingresa a los GruposAceptados y se detiene la fase 1 del algoritmo. En este caso lo que quedo del DataSet original sigue a la fase 2 del algoritmo, y las instancias de este DataSet ahora se denominan instancias huérfanas, ya que no pertenecen a ninguno de los GruposAceptados.

En la fase 2 del algoritmo, las instancias huérfanas del DataSet se toman una a una y se revisa en cuál de los GruposAceptados se deben incluir, se incluyen y se actualiza el modelo del grupo seleccionado. La Figura 3 muestra cómo se procesa una instancia huérfana (instancia número 15), en este caso la instancia de color azul oscuro. La instancia se incluye en el primer grupo aceptado y se genera un Modelo de Regresión Lineal del grupo con este pequeño cambio, luego la instancia se remueve. En este caso el modelo reporta un coeficiente de correlación de 0,96, lo que implica una pérdida de calidad ya que el modelo del grupo antes tenía un coeficiente de 0,97. Esta perdida (-0,01) se calcula de la misma forma para el segundo grupo (-0,02) y para el tercer grupo (0,01). En este caso el último grupo reporta que es el mejor lugar donde debe incluirse la instancia 15, ya que es el menor valor de perdida, que, en este caso corresponde a una ganancia o mejora del modelo. La instancia entonces se agrega al tercer grupo de aceptados y se actualiza el modelo de regresión lineal de dicho grupo.

Cuando se termina la agrupación de las instancias (fase 1 y 2 de entrenamiento), se obtienen los Grupos y sus correspondientes modelos. Estos son la base para el proceso de predicción del valor de PSI para nuevas instancias. La Figura 4 muestra la función que permite predecir el valor de PSI para una instancia nueva con base en los resultados del entrenamiento. En este caso a la Instancia de entrada se le calcula la distancia euclidiana con cada una de las instancias de cada uno de los GruposAceptados y se toma el identificador del grupo (MejorGrupo) que tiene la instancia con menor valor de distancia, en resumen, se busca cual instancia de los Grupos Aceptados tiene la menor distancia euclidiana (la instancia más parecida, operando de manera similar al algoritmo K-NN con K=1) y se define en que grupo (MejorGrupo) esta. Ya con el grupo identificado, en la línea 12 se le aplica a la instancia el Modelo de Regresión Lineal del grupo y se predice el valor de PSI, que corresponde al valor que retorna esta función.

Figura 2. Ejemplo de la fase 1 del algoritmo propuesto

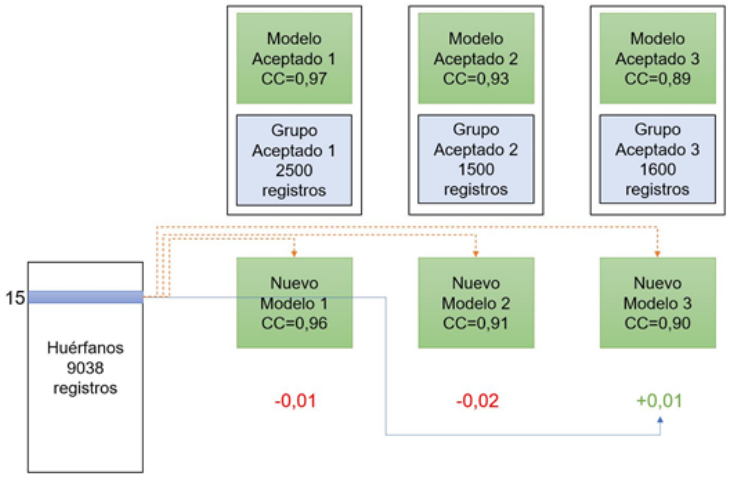


Fuente: Elaboración Propia

El proceso de validación consiste en ejecutar la predicción de cada una de las instancias del dataset de validación para obtener el valor de PSI predicho y compararlo con el valor PSI real o registrado en el dataset, de allí se procede a calcular diferentes métricas de calidad como son el error cuadrático medio (Mean Squared Error, MSE), la raíz del error cuadrático medio (Root-Mean-Squared Error, RMSE), la raíz del error cuadrático medio normalizada (Normalized Root-Mean-Square Error, NRMSE), el error absoluto medio (Mean Absolute Error, MAE) y la cantidad (o porcentaje) de instancias cuyas predicciones se encuentran en el rango del valor real ±15% de este valor.

La implementación del algoritmo se realizó en Java versión "1.8.0_131" y se usó Weka-stable-3.8.0.jar [31] para obtener los modelos de regresión lineal junto con el proceso de selección de atributos. El código fuente del algoritmo se puede consultar en https://gitlab.com/pacho328/agrupacion-greedy-pavimentos.

Figura 3. Ejemplo de la fase 2 del algoritmo propuesto



Fuente: Elaboración Propia

Figura 4. Pseudocódigo de la función para la predicción de PSI

Entradas:
- Instancia: Instancia a la cual se le desea predecir el valor de PSI.
- GruposAceptados: Una lista de grupos en archivos ARFF.
- ModelosAceptados: Modelos de regresión lineal de cada uno de los grupos con sus atributos seleccionados, coeficientes y valores de correlación.

Salidas:
- PSIPredicho: Valor de PSI predicho para la instancia de entrada.

```
Inicio
1.    MenorDistancia = Double.Max_Value
2.    MejorGrupo = 0
3.    Para i = 0 Hasta GruposAceptados.Tamaño Hacer
4.        Para j= 0 Hasta GruposAceptados[i].Instancias.Tamaño Hacer
5.            distancia = DistanciaEuclidiana (Instancia, GruposAceptados[i].Instancia[j])
6.            Si distancia < MenorDistancia Hacer
7.                MenorDistancia = distancia
8.                MejorGrupo = i
9.            Fin Si
10        Fin Para
11    Fin Para
12    PSIPredicho = AplicaModeloRegresionLineal (GruposAceptados[MejorGrupo], Instancia)
13    Retornar PSIPredicho
Fin
```

Fuente: Elaboración Propia

Resultados

Modelos obtenidos en la fase de entrenamiento

El afinamiento de los parámetros del algoritmo se realizó usando un enfoque de grilla. Los parámetros que se afinaron fueron ErrorPromedioAceptado y ErrorPromedioAdicional tomando para el primero, valores desde 0,02 hasta 0,07 y para el segundo desde 0,03 hasta 0,07 respectivamente, con lo cual se generaron 99 pruebas. El parámetro CoeficienteCorrelacionMinimo se dejó en un valor por defecto de 0,88 y para la obtención de los modelos de regresión lineal con selección de atributos se dejaron los valores por defecto de Weka [31]. En cada iteración se evaluaron los modelos resultantes con el conjunto de datos de validación y se realizó el cálculo del porcentaje de puntos en el rango del [PSI real] - 15% y el [PSI real] + 15%. Como resultado se obtuvo la Tabla 2 donde se muestran 63 de los 99 resultados obtenidos y se resalta en negrita el mejor resultado obtenido (76%). Esta tabla muestra el porcentaje de los valores PSI predichos que se encuentran dentro de ± 15% del valor del PSI real, para cada una de las combinaciones evaluadas partiendo de las variables de ErrorPromedioAdicional y ErrorPromedioAceptado

Tabla 2. Porcentaje de puntos dentro de ± 15% de error

Error Promedio Adicional	Error Promedio Aceptado						
	0,02	0,025	0,03	0,035	0,04	0,045	0,05
0,03	69,3	72,8	72,3	74,3	70,9	69,3	72,1
0,035	70,6	69,9	70,3	71,7	73,9	69,3	72,1
0,04	70,2	75,2	75	73,5	71,8	70,8	72,1
0,045	73,2	76	73,3	74,3	72,3	72,6	73,5
0,05	72,3	74,5	73,8	74,7	72,3	73,4	73,7
0,055	71,2	71,2	73,9	71,5	71,2	72,1	71,8
0,06	72	72,7	74,2	74,6	74,2	68,9	69,3
0,065	72	72,7	72,3	74,6	74,5	74,7	70,3
0,07	72	72,6	72,9	74,7	74,7	73,7	70,5

Fuente: Elaboración Propia

Los 99 resultados encontraron entre 2 y 5 grupos válidos, siendo 3 grupos el valor más frecuente con 47 apariciones, seguido por 2 grupos con una frecuencia de aparición de 32 veces. El algoritmo, en 8 ocasiones no armó grupos, sino que dejo el dataset completo en un solo grupo.

Los resultados de la mejor solución obtenida se muestran en la Tabla 3. Las variables no seleccionadas en cada grupo (3 en total) tienen un símbolo "-" como coeficiente. Las variables aadt y elevation (escritos en rojo) pueden ser eliminados del conjunto de entrenamiento, ya que ningún modelo las utilizó. Todos los grupos tienen un coeficiente de correlación final superior al 80% (parte baja de la tabla), de hecho, el coeficiente de correlación ponderado de los 3 grupos es de 84,1%. Además, se muestran los resultados de la fase 1, resaltando que los grupos aceptados tenían sólo (ni) 2340, 1813 y 1980 instancias cada uno, y que al incluir las instancias huérfanas crecieron a 5015, 2223 y 7400 instancias respectivamente. Este ingreso les costó calidad a todos los grupos, el grupo 1 paso de 99,3% a 81%, perdiendo 18,3% en el coeficiente de correlación. Para el segundo grupo la perdida fue poca, solo un 1,3%, hecho asociado a la poca cantidad de instancias huérfanas que entraron en este grupo. Finalmente, para el grupo 3 que fue el que más recibió instancias huérfanas el porcentaje perdido fue de 8,8%.

Validación y comparación

El comportamiento de los modelos de predicción se validó con el conjunto de pruebas mencionado anteriormente con el fin de predecir el atributo PSI. Para este procedimiento se tomaron las 3005 instancias de prueba y para cada instancia se buscó cual instancia del conjunto de datos de entrenamiento tenía una mayor similitud basados en la distancia euclidiana (algoritmo 1-nn con distancia euclidiana) y de esa manera saber el grupo al que pertenece esa instancia y aplicar el modelo de regresión lineal correspondiente, según los coeficientes presentados en la Tabla 3.

La Figura 5 muestra en la parte (a) los resultados (PSI vs PSI predicho) usando los 3 modelos lineales obtenidos con el algoritmo propuesto. La parte (b) muestra los resultados obtenidos por [12] usando 6 modelos lineales (6 grupos), este modelo se obtuvo con un algoritmo de recocido simulado que evoluciona agrupaciones en forma discreta. Una variación de ese algoritmo se usó para obtener modelos no lineales y como resultado, la parte (c) de la misma figura muestra los resultados con las 5 agrupaciones obtenidas. Esta figura además muestra el porcentaje de resultados que se encuentran en el rango del valor de PSI real ±15%. La propuesta de este trabajo es superior a la propuesta lineal (b) previamente propuesta en 2% y está por debajo de la propuesta no lineal en un 5%.

Para evaluar el tiempo de ejecución de las propuestas, se recreó la implementación lineal de Khadka et al y se ejecutaron ambas propuestas en el mismo computador (n1-highcpu-8, 8 CPU virtuales, 7.2 GB de memoria de Google Cloud Platform). Esto arrojó como resultado que la implantación de [12] tomó 25 horas de ejecución solo para la agrupación 6, lo que implica que su ejecución demoraría aproximadamente 375 horas para armar grupos de 2 a 16 y seleccionar la mejor solución, conforme se propone en su artículo. La implementación propuesta por este articulo demoró 24 horas para todas las combinaciones realizadas (99) incluyendo el cálculo de las predicciones. El mejor resultado obtenido (ver Tabla 3) en la presente propuesta tan sólo tomó un tiempo de ejecución de 13 minutos, y generó mejores resultados, ya que 76% de las instancias están dentro del [PSI real] ± 15%. Además el modelo es más sencillo que el propuesto por [12] ya que tiene 3 grupos y el modelo lineal de Khadka et al. tiene 6 grupos.

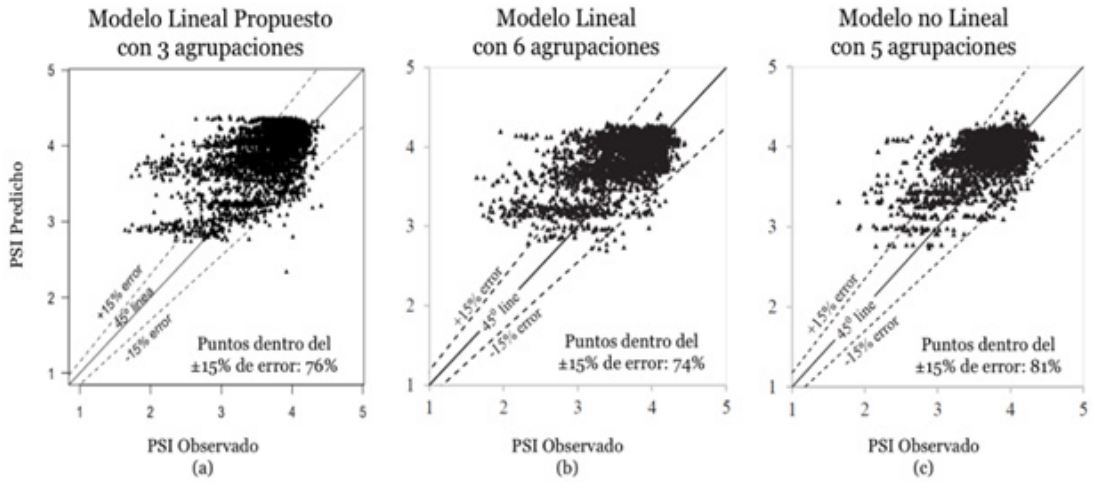
Tabla 3. Resultados de los modelos por grupo

Variable	Coeficientes			
	Grupo 1	Grupo 2	Grupo 3	
age=0	0,093	0,087	0,09	
age=1	-0,050	0,035	0,078	
age=2	-0,053	0,010	0,043	
age=3	-	-	0,026	
age=4	-0,069	-	-	
age=5	-0,041	-0,014	-0,087	
age=6	-0,142	-0,325	-0,080	
age=7	-	-	-0,450	
age=8	-	-	-0,470	
aadt	-	-	-	
trucks	0,000	-	-	
elevation	-	-	-	
precip	-0,032	-0,013	-0,017	
min_temp	0,010	-0,002	-0,012	
max_temp	-0,049	-0,012	0,016	
wet_days	0,001	-0,002	0,001	
freeze_thaw	-0,003	-0,000	-	
rut_depth	-1,039	-1,088	-0,939	
number_of_lanes=1	0,224	0,152	0,070	
number_of_lanes=2	-0,258	-0,157	-	
number_of_lanes=3	-	-0,018	-0,362	
sys_id=1	-0,204	0,295	0,304	
sys_id=2	0,132	-0,138	-0,086	
sys_id=3	0,049	-0,184	-0,224	
f_class=1	0,198	-0,192	-0,158	
f_class=2	-0,191	0,264	0,162	
f_class=3	-	0,176	0,068	
f_class=4	0,036	0,178	0,194	
f_class=5	-0,269	-0,111	-0,070	
f_class=6	-0,351	-0,558	-0,152	

f_class=7		-0,808	-0,183	0,348	
category=1		0,490	0,210	0,072	
category=2		-	0,020	0,113	
category=3		-0,168	-0,028	0,157	
category=4		0,102	-0,277	-0,527	
category=5		-0,758	-0,389	-0,121	
intercepto		7,543	5,176	3,345	Totales
Fase 1	CC	99.3%	96.3%	91.8%	96%
	ni.	2340	1813	1980	6133
Fase 2	CC	81%	95%	83%	84.1%
	ni	5015	2223	7400	14638

Fuente: Elaboración Propia

Figura 5. PSI predicho versus PSI observaciones



Fuente: Elaboración Propia

A continuación, la Tabla 4 muestra otras métricas de comparación del modelo lineal propuesto y de los modelos lineal y no lineal desarrollados por [12]. Los valores de los dos modelos lineales son muy similares, y los resultados del modelo no lineal son mejores (menores en este caso), lo que motiva a utilizar el algoritmo propuesto en este articulo para obtener modelos no lineales.

Tabla 4. Métricas del comportamiento de la predicción.

Métrica	Modelo Propuesto	Modelo Lineal	Modelo No Lineal
	(3 grupos)	(6 grupos)	(5 grupos)
MEA	0,37	0,36	0,33
RMSE	0,49	0,47	0,41
NRMSE	0,17	0,17	0,15

Fuente: Elaboración Propia

Conclusiones

En este trabajo se utilizó un enfoque diferente para obtener modelos de regresión lineal que agrupan muestras de segmentos de pavimentos. Para esto se diseñó un algoritmo greedy que realiza agrupaciones incrementales de los datos basado en el error de predicción que tienen las instancia en el marco de un modelo lineal. Los resultados obtenidos por el algoritmo propuesto superan al modelo de regresión lineal propuesto en el estado del arte en cuanto a simplicidad, menor número de agrupaciones, y en cuanto al porcentaje de instancias cuya predicción está en el rango del ±15% del valor de PSI real. Además, obtiene valores similares de MAE, RMSE y NRMSE. Por otro lado, el modelo de regresión no lineal presentado en el estado del arte obtiene mejores resultados en todas las medidas, pero su formulación es más compleja y cuenta con un mayor número de agrupaciones. En relación con el tiempo de ejecución (y la correspondiente complejidad computacional) se observa una disminución significativa con el algoritmo propuesto. En los modelos obtenidos, las variables aadt y elevation fueron ignoradas.

La fase 1 del algoritmo propuesto tiene una baja complejidad, pero la fase 2, relacionada con la inclusión de las instancias huérfanas tiene una mayor complejidad computacional, ya que implica la ejecución de un algoritmo 1-nn y la evaluación de k modelos de regresión lineal para todas y cada una de las instancias. A pesar de que la fase 2 tiene este costo, se considera que la fase 1 es la que tiene más opciones de mejora, por ejemplo, hacer más adaptativa la conformación del grupo candidato que se espera sea aceptado, la eliminación o reducción de parámetros en el algoritmo y evitar reducir la calidad aceptada en cada iteración con un valor fijo. Finalmente, el grupo de investigación contempla además de hacer los cambios previamente comentados, hacer uso de esa nueva versión en la propuesta de un modelo no lineal.

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, 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,” Transp. Res. Rec., no. 1123, pp. 40–46, 1987.
3. M. Rodríguez Moreno, G. Thenoux Zeballos, and A. González Vaccarezza, “Evaluación probabilística del agrietamiento de pavimentos asfálticos en carreteras de Chile,” Rev. la Constr., vol. 12, no. 2, pp. 152–165, 2013, doi: 10.4067/s0718-915x2013000200012.
4. H. Ceylan, M. B. Bayrak, and K. Gopalakrishnan, “Neural networks applications in pavement engineering: A recent survey,” Int. J. Pavement Res. Technol., vol. 7, no. 6, pp. 434–444, 2014, doi: 10.6135/ijprt.org.tw/2014.
5. R. Ramaswamy and M. Ben-Akiva, “Estimation of Highway Pavement Deterioration form In-Service Pavement Data,” Transp. Res. Rec., vol. 1272, pp. 96–106, 1990, [Online]. Available: https://trid.trb.org/view/351896.
6. 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, 2006, doi: 10.3923/jas.2006.193.197.
7. N. Bandara and M. Gunaratne, “Current and future pavement maintenance prioritization based on rapid visual condition evaluation,” J. Transp. Eng., vol. 127, no. 2, pp. 116–123, 2001, doi: 10.1061/(ASCE)0733-947X(2001)127:2(116).

8. K. A. Abaza, "Deterministic performance prediction model for rehabilitation and management of flexible pavement," Int. J. Pavement Eng., vol. 5, no. 2, pp. 111–121, 2004, doi: 10.1080/10298430412331286977.

9. H. Späth, "Algorithm 39 Clusterwise linear regression," Computing, vol. 22, no. 4, pp. 367–373, 1979, doi: 10.1007/BF02265317.

10. K. Joki, A. M. Bagirov, N. Karmitsa, M. M. Mäkelä, and S. Taheri, "Clusterwise support vector linear regression," Eur. J. Oper. Res., vol. 287, no. 1, pp. 19–35, 2020, doi: 10.1016/j.ejor.2020.04.032.

11. 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, 1989, doi: 10.1007/BF02296405.

12. 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, 2020, doi: 10.1080/10298436.2018.1521970.

13. M. Khadka and A. Paz, "Comprehensive Clusterwise Linear Regression for Pavement Management Systems," J. Transp. Eng. Part B Pavements, vol. 143, no. 4, p. 04017014, 2017, doi: 10.1061/jpeodx.0000009.

14. 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, p. 2159865, 2018, doi: 10.1155/2018/2159865.

15. 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, 2015, doi: 10.1007/s11803-015-0009-2.

16. 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, 2015, doi: 10.1007/s11590-014-0749-3.

17. 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, 2017, doi: 10.1016/j.atmosres.2017.01.003.

18. 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, 2017, doi: 10.1080/21568235.2017.1357481.

19. 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, 2017, doi: 10.1287/ijoc.2016.0729.

20. 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, 2018, doi: 10.1080/10556788.2017.1371717.

21. 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 IEEE International Conference on Fuzzy Systems, 2018, vol. 2018-July, pp. 1–8, doi: 10.1109/FUZZ-IEEE.2018.8491476.

22. I. Gitman, J. Chen, E. Lei, and A. Dubrawski, "Novel Prediction Techniques Based on Clusterwise Linear Regression," arXiv Prepr. arXiv1804.10742, 2018, [Online]. Available: http://arxiv.org/abs/1804.10742.

23. S. Bougeard, V. Cariou, G. Saporta, and N. Niang, "Prediction for regularized clusterwise multiblock regression," in Applied Stochastic Models in Business and Industry, 2018, vol. 34, no. 6, pp. 852–867, doi: 10.1002/asmb.2335.

24. N. Veeramisti, A. Paz, M. Khadka, and C. Arteaga, "A clusterwise regression approach for the estimation of crash frequencies," J. Transp. Saf. Secur., pp. 1–31, 2019, doi: 10.1080/19439962.2019.1611681.

25. 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, 2019, doi: 10.1007/s11634-018-0331-4.

26. G. Galimberti, L. Nuzzi, and G. Soffritti, “Covariance matrix estimation of the maximum likelihood estimator in multivariate clusterwise linear regression,” Stat. Methods Appl., 2020, doi: 10.1007/s10260-020-00523-9.

27. 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, 2020, doi: 10.1016/j.petrol.2019.106732.

28. 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 Appl., vol. 29, no. 1, pp. 49–78, 2020, doi: 10.1007/s10260-019-00480-y.

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

30. B. Kitchenham and P. Brereton, “A systematic review of systematic review process research in software engineering,” Information and Software Technology, vol. 55, no. 12. Elsevier B.V., pp. 2049–2075, 2013, doi: 10.1016/j.infsof.2013.07.010.

31. E. Frank, M. A. Hall, and I. H. Witten, “The WEKA workbench,” in Data Mining, Morgan Kaufmann, 2016, p. 128.