

Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), mayo-junio 2025,
Volumen 9, Número 3.

https://doi.org/10.37811/cl_rcm.v9i1

UN MODELO MULTICLASIFICADOR PARA LA PREDICCIÓN DE LA CARGA DE ENFRIAMIENTO Y CALOR EN EDIFICIOS RESIDENCIALES

**MULTICLASSIFIER MODEL FOR PREDICTING COOLING
AND HEATING LOADS IN RESIDENTIAL BUILDINGS**

Luis Enrique, Ramirez Milla
Universidad Nacional del Santa, Perú

DOI: https://doi.org/10.37811/cl_rcm.v9i3.18273

Un Modelo Multiclasificador para la Predicción de la Carga de Enfriamiento y Calor en Edificios Residenciales

Luis Enrique Ramirez Milla¹Lramirez@uns.edu.pe<https://orcid.org/0000-0001-9064-5395>Universidad Nacional del Santa
Perú

RESUMEN

El estudio propone un modelo multiclasificador basado en algoritmos de machine learning con validación cruzada para predecir la carga de enfriamiento y calor en edificios residenciales en términos del coeficiente de determinación: R^2 . Con respecto al método se tiene que el conjunto de datos analizado es un dataset de Kaggle, el cual está compuesto de 768 registros, el mismo que consta de 8 variables de entrada y 2 variables de salida. El modelo incluye varias etapas de procesamiento y evaluación y se comparan diferentes técnicas propuestas para predecir la carga de enfriamiento y calor utilizando conjuntos de algoritmos considerando la evaluación del desempeño. Los mejores resultados se obtuvieron con el modelo multiclasificador 1 (Random forest regressor, Linear regresion, XGBoost y K-nearest Neighbors) para la “Carga de calefacción” ($R^2 = 99.75$) y con XGBoost para la “carga de enfriamiento” ($R^2 = 99.05$). Adicionalmente, se realizaron procesos de preprocesamiento de datos, como la normalización de variables, la detección y manejo de valores atípicos, y la división del conjunto de datos en subconjuntos de entrenamiento y prueba. La validación cruzada se aplicó para garantizar la robustez de los modelos y evitar el sobreajuste. La implementación se realizó en Python utilizando bibliotecas como Scikit-learn, XGBoost y Pandas. En enfoque demuestra que el uso combinado de múltiples algoritmos mejora la precisión en la predicción de variables térmicas, siendo una herramienta prometedora para la eficiencia energética en el diseño de edificios residenciales.

Palabras clave: modelo multiclasificador, carga de calor, carga de enfriamiento, edificios residenciales, Python

¹ Autor principal

Correspondencia: Lramirez@uns.edu.pe

Multiclassifier Model for Predicting Cooling and Heating Loads in Residential Buildings

ABSTRACT

The study proposes a multi-classifier model based on machine learning algorithms with cross-validation to predict the cooling and heating load in residential buildings in terms of the coefficient of determination: R^2 . Regarding the method, the analyzed dataset is a Kaggle dataset, which is composed of 768 records, consisting of 8 input variables and 2 output variables. The model includes several stages of processing and evaluation and different proposed techniques are compared to predict the cooling and heating load using sets of algorithms considering the performance evaluation. The best results were obtained with multi-classifier model 1 (Random Forest regressor, Linear regression, XGBoost and K-nearest Neighbors) for the “Heating load” ($R^2 = 99.75$) and with XGBoost for the “cooling load” ($R^2 = 99.05$). Additionally, data preprocessing processes were performed, such as variable normalization, outlier detection and handling, and splitting the dataset into training and testing subsets. Cross-validation was applied to ensure the robustness of the models and avoid overfitting. The implementation was carried out in Python using libraries such as Scikit-learn, XGBoost, and Pandas. The approach demonstrates that the combined use of multiple algorithms improves the accuracy in predicting thermal variables, making it a promising tool for energy efficiency in residential building design.

Keywords: multi-classifier model, heat load, cooling load, residential buildings, Python

Artículo recibido 17 abril 2025

Aceptado para publicación: 22 mayo 2025



INTRODUCCIÓN

Obras recientes

Roy et al. (2020) propusieron una técnica de *machine learning* para predecir la carga térmica en edificios residenciales usando datos de 768 muestras con 8 atributos. Utilizando R y Matlab, evaluaron algoritmos como Gaussian Process Regression (GPR), Gradient Boosting Machine (GBM) y Minimax Likelihood Machine Regression (MPMR), obteniendo un R^2 de 99%, lo que confirma la eficacia del *machine learning* en la optimización energética de edificios inteligentes.

Guo et al. (2023) propusieron combinar algoritmos de optimización hiperparamétrica con LightGBM para mejorar la predicción de cargas térmicas en edificios. Usando un dataset de 768 registros, construyeron cuatro modelos híbridos, destacando TPE-LightGBM por su alta precisión (R^2 de 99.81% en calefacción y 99.24% en refrigeración). Concluyeron que su enfoque es eficaz para evaluar la eficiencia energética en etapas tempranas del diseño de edificaciones.

Zhou et al. (2021) analizaron 15 algoritmos de *machine learning* para predecir la carga de calefacción en edificios, utilizando datos con 8 atributos y validación cruzada. GPR y Multivariate Adaptive Regression Splines (MARS) obtuvieron el mejor desempeño con un R^2 de 95.6%, concluyendo que estos modelos son precisos y eficaces para estimar la demanda energética.

Moayedi et al. (2019) estudiaron la predicción de la carga de calefacción en edificios eficientes usando algoritmos de machine learning como ElasticNet y Random Forest. Con un dataset de 768 edificaciones, Random Forest destacó con un R^2 de 99.89%, concluyendo que es una herramienta eficaz para estimar la demanda térmica en construcciones energéticamente eficientes.

Moradzadeh et al. (2020) propusieron una metodología para predecir las cargas de calefacción y refrigeración en edificios utilizando Multilayer perceptron (MLP) y Support vector regression (SVR), con un dataset de 768 muestras. SVR obtuvo el mejor R^2 (99.79%) para calefacción y 98.78% para refrigeración, concluyendo que ambos enfoques son eficaces para optimizar el consumo energético en edificaciones.



Algoritmos utilizados

Decision Tree

Es un algoritmo de machine learning diseñado para la toma de decisiones, el cual está compuesto por un nodo principal, diversos nodos internos y múltiples nodos terminales o de hoja, en donde los nodos de hoja indican los resultados de las decisiones, mientras que los demás nodos representan las pruebas basadas en las características, cuyo propósito es crear un árbol capaz de hacer predicciones precisas sobre datos no observadas (Zhou, 2021).

ElasticNet

Es un algoritmo robusto de machine learning que mezcla las características de Ridge y Lasso, incorporando la función estándar de pérdida de mínimos cuadrados, empleado en problemas típicos en conjuntos de datos de alta dimensionalidad, sobreajuste y multicolinealidad. (Mahdi et al., 2021)

K-Nearest Neighbors (KNN)

Es un método de machine learning no paramétrico que a diferencia de otros algoritmos que descartan los datos de entrenamiento después de construir el modelo, KNN conserva en memoria todas las instancias de entrenamiento, es así que cuando se presenta un nuevo ejemplo x que no ha sido visto previamente, el algoritmo identifica los k ejemplos de entrenamiento más cercanos a x y determina la etiqueta mayoritaria (en el caso de clasificación) o calcula la etiqueta media (en caso de regresión) (Burkov, 2019).

Lasso

Es un algoritmo de regresión regularizado basado en la minimización de la norma L_1 aplicada a la regresión lineal, en donde no existe una solución directa para Lasso, por lo que se requieren enfoques numéricos iterativos, como el descenso de gradiente, para resolverlo, como también técnicas de subgradiente, por lo que, Lasso es útil tanto para la selección de variables como para la estimación con penalización. (Jiang, 2021)

Random Forest

Es un modelo formado por un grupo de árboles construidos de manera recursiva. El término “Random” señala que cada árbol emplea una muestra aleatoria de los datos originales para el entrenamiento, a la



vez, en cada nodo de los árboles generados, se considera solo un subconjunto aleatorio de las características de entrada para efectuar las divisiones (Trappenberg, 2020)

Regresión Lineal

Representa un modelo tradicional de naturaleza lineal, también denominado ajuste polinómico y constituye uno de los enfoques lineales más básicos en machine learning, considerando un problema de regresión lineal, donde los datos de entrenamiento consisten en p ejemplos, cuya entrada tiene n dimensiones, representadas como $(x_i, i=1\dots p)$ y donde $x_i \in \mathbb{R}^n$. Por otro lado, la salida es unidimensional y se expresa como $(y_i, i=1\dots p)$ con $y_i \in \mathbb{R}$. (Joshi, 2020)

Ridge

Ridge-Regressor es un método de machine learning que se emplea principalmente cuando las variables independientes muestran una alta correlación entre sí. A diferencia de la regresión Lasso, este enfoque mantiene todos los coeficientes en el modelo. (Thevaraja et al., 2019)

XGBoost

XGBoost Regressor es una agrupación de árboles de decisiones en el que el modelo realiza una expansión aditiva de la función objetivo. Esto se consigue al minimizar una función de pérdida para regular la complejidad de los árboles. (Bentéjac et al., 2021)

Métrica utilizada

R² SCORE. Es una métrica que se refiere a la varianza de los valores predichos, dividida entre la suma de la varianza de las predicciones y la varianza esperada de los errores (Gelman et al., 2019). Esta métrica se expresa matemáticamente de la siguiente manera:

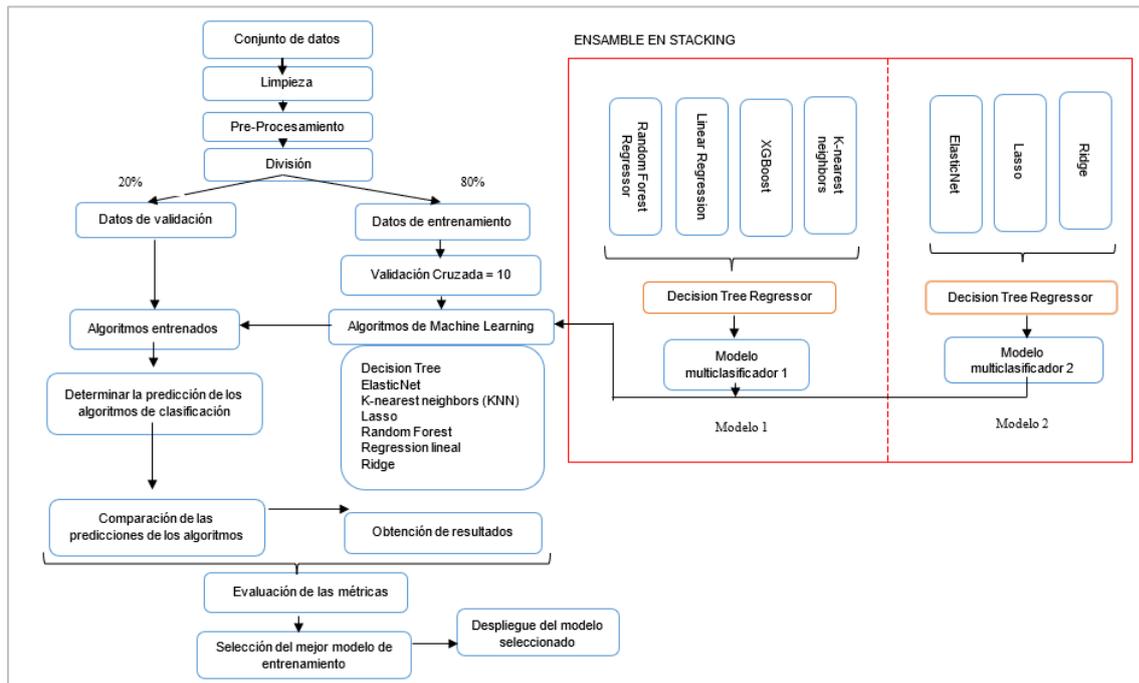
$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (1)$$

MATERIALES Y MÉTODOS

Aquí detallamos la metodología utilizada para la construcción del modelo multclasificador (Figura 1), comprendiendo las siguientes etapas: Conjunto de datos, Limpieza, preprocesamiento, división, algoritmos, comparación de predicciones, obtención de resultados, evaluación de la métrica, selección del mejor modelo y despliegue del modelo seleccionado.



Figura 1: Modelo multclasificador



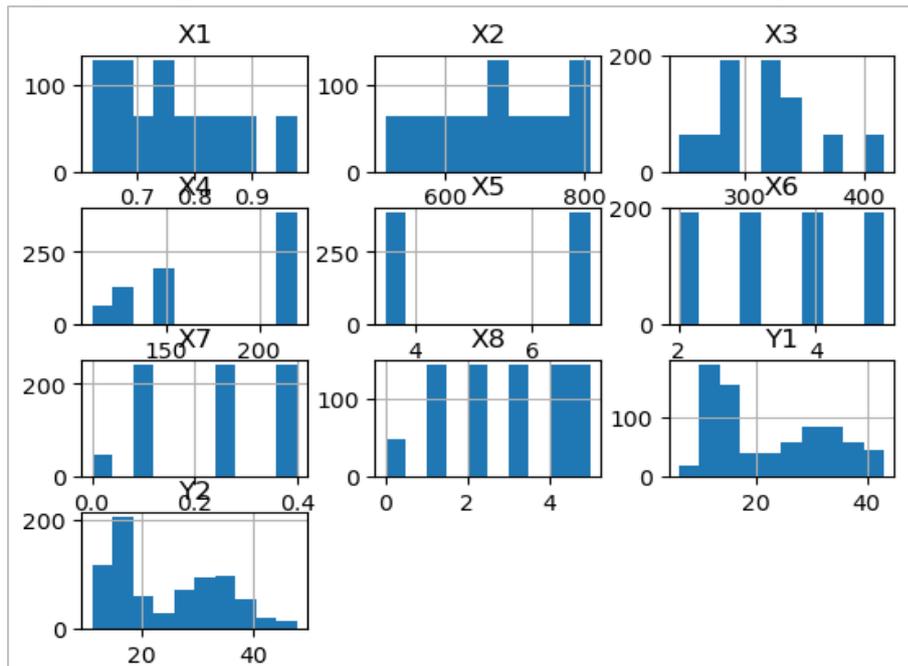
Conjunto de datos

Para la selección de los datos se recurrió a la base de datos de Kaggle de donde se obtuvo el dataset energy-efficiency-data-set. Este dataset comprende 768 registros de edificios residenciales con 10 atributos (8 variables predictoras y 2 variables predichas) descritos en la Tabla 1 y Figura 2.

Tabla 1: Variables del conjunto de datos sobre calefacción y enfriamiento

Variable	Descripción	Rol de la variable
X1	Compacidad o estanqueidad relativa de las medidas correspondientes.	Entrada
X2	Cantidad de espacio que abarca la superficie externa	Entrada
X3	Área de un edificio calculada multiplicando la altura del edificio, medida a partir del suelo hasta la línea del alero del techo por el ancho del edificio	Entrada
X4	Parte de un lote o parcela cubierta por la estructura del techo de todos los edificios, excluyendo los aleros y los voladizos, pero incluyendo los porches, patios y garajes cubiertos.	Entrada
X5	Distancia vertical desde el suelo hasta el punto más elevado de la habitación.	Entrada
X6	Ubicación en el entorno con referencia al tiempo, lugar y personas.	Entrada
X7	Área de la superficie interior de todas las ventanas acristaladas, incluida el área de la hoja, el bordillo u otros elementos estructurales, que encierran el espacio acondicionado.	Entrada
X8	Sistemas que utilizan cristal y otros materiales para cubrir espacios.	Entrada
Y1	Cantidad de energía térmica que sería necesario agregar a un espacio para mantener la temperatura en un rango aceptable.	Salida
Y2	Cantidad de energía térmica que sería necesario extraer de un espacio (enfriamiento) para mantener la temperatura en un rango aceptable.	Salida

Figura 2: Diagrama de frecuencias de las variables en el conjunto de datos



Respecto a la correlación de las variables, en la Figura 3 se observa una alta relación de las variables predictoras X1 - X8 con respecto a las variables predichas Y1 y Y2.

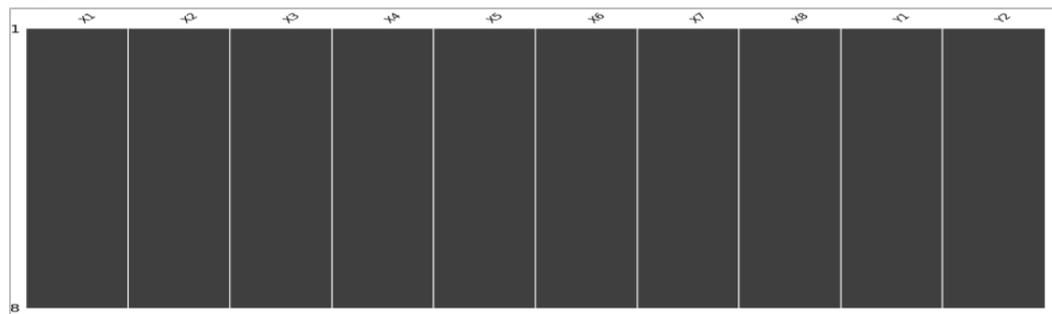
Figura 3: Matriz de correlación



Limpieza

La revisión de la información se llevó a cabo mediante comandos en Python, lo que permitió confirmar que no existen datos faltantes, como se muestra en la Figura 4. Además, no fue necesario aplicar transformaciones, ya que los valores presentes en las características son adecuados para llevar a cabo los experimentos con los algoritmos propuestos.

Figura 4: Escaneo de datos faltantes



Preprocesamiento

Se evaluaron los datos del conjunto para identificar la necesidad de posibles transformaciones y adecuar su representación a un formato óptimo para tareas de minería de datos. Sin embargo, en este caso particular, los datos ya se encontraban en condiciones apropiadas, por lo que no fue necesario realizar ajustes adicionales

División de los datos

Se realizó una partición del conjunto de datos, asignando el 80% para el entrenamiento y el 20% restante para la prueba. Esta estrategia tuvo como finalidad evitar que el modelo se limite a memorizar los patrones del conjunto de entrenamiento, lo que podría causar sobreajuste (overfitting), y, en cambio, favorecer su capacidad de generalización al realizar predicciones precisas sobre datos no vistos previamente.

Validación cruzada

Durante el entrenamiento del modelo, se aplicó un esquema de validación cruzada con diez repeticiones, con el fin de obtener una evaluación más robusta y generalizable del rendimiento.

Algoritmos

La Figura 5 presenta los distintos parámetros asignados a los algoritmos durante su calibración, con el objetivo de identificar la configuración óptima para cada uno de ellos y en la Figura 6 se presenta la definición base para el modelo multclasificador.

Figura 5: Parametrización de los algoritmos

```

A) Árboles de decisión
parametros = {
    'max_leaf_nodes': [None, 5, 10, 15, 20],
    'max_depth': [None, 5, 10, 15, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

E) ElasticNet
parametros = {
    'alpha': [5.0, 10.0],
    'fit_intercept': [True],
    'copy_X': [True, False],
    'tol': [1e-5, 1e-4],
    'max_iter': [1000, 2000, 5000]
}

B) Regresión lineal
parametros = {
    'fit_intercept': [True, False],
    'copy_X': [True, False],
    'n_jobs': [-1, 1, 2],
    'positive': [True, False]
}

F) Random Forest
parametros = {
    'bootstrap': [True, False],
    'max_depth': [5, 10, 15],
    'min_samples_leaf': [1, 2, 4],
    'min_samples_split': [2, 5, 10],
    'max_features': ['sqrt', 'log2']
}

C) Ridge
parametros = {
    'alpha': [0.1, 1.0],
    'fit_intercept': [True, False],
    'copy_X': [True, False],
    'solver': ['auto', 'svd'],
}

G) XGBoost
parametros = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 6, 9],
    'learning_rate': [0.01, 0.1, 0.3],
    'gamma': [0, 0.1, 0.2]
}

D) Lasso
parametros = {
    'alpha': [5.0, 10.0],
    'fit_intercept': [True],
    'copy_X': [True, False],
    'tol': [1e-5, 1e-4],
    'max_iter': [1000, 2000, 5000]
}

H) KNN
parametros = {
    'n_neighbors': [3, 5, 7, 9],
    'weights': ['uniform', 'distance'],
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
    'leaf_size': [10, 30, 50],
    'p': [1, 2],
    'metric': ['euclidean', 'manhattan']
}

```

Figura 6: Definición base para el modelo multclasificador

```

A) Modelo multclasificador 1
# definiendo el modelo base
level0 = list()
level0.append(('rl', LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, positive=False)))
level0.append(('knn', KNeighborsRegressor(algorithm='ball_tree', leaf_size=10, metric='manhattan', n_neighbors=3, p=1, weights='uniform')))
level0.append(('rf', RandomForestRegressor(max_depth=10, max_features='log2', min_samples_leaf=4, min_samples_split=5)))
level0.append(('xgboost', XGBRegressor(gamma=0, learning_rate=0.3, max_depth=3, n_estimators=300)))

level1 = DecisionTreeRegressor(max_depth=10, max_leaf_nodes=None, min_samples_leaf=1, min_samples_split=2)

# define the stacking ensemble
model_Y2 = StackingRegressor(estimators=level0, final_estimator=level1)

B) Modelo multclasificador 2
# definiendo el modelo base
level0 = list()
level0.append(('rd', Ridge(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=None, solver='svd')))
level0.append(('lasso', Lasso(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=1000, tol=0.0001)))
level0.append(('ElasticNet', ElasticNet(alpha=0.1, copy_X=True, fit_intercept=True, l1_ratio=0.9, max_iter=1000, tol=0.0001)))

level1 = DecisionTreeRegressor(max_depth=10, max_leaf_nodes=None, min_samples_leaf=1, min_samples_split=2)

# define the stacking ensemble
model_Y2 = StackingRegressor(estimators=level0, final_estimator=level1)

```

Comparación de predicciones

Una vez ejecutados los algoritmos de Machine Learning y obtenidos los resultados correspondientes a cada una de las métricas consideradas, se procedió a su tabulación, análisis y comparación. Este proceso resulta esencial para evaluar el desempeño de los modelos y seleccionar el algoritmo que ofrezca mayor precisión.

Obtención de resultados

La obtención de resultados en el contexto de Machine Learning se basó en la evaluación y medición del rendimiento del modelo tras su fase de entrenamiento y validación.

Este proceso no solo consistió en aplicar el modelo sobre un conjunto de prueba, sino también en analizar la métrica clave que permitió valorar su desempeño.

Evaluación de la métrica

La evaluación de métricas en el ámbito de Machine Learning constituyó un elemento clave para analizar el rendimiento de los modelos y respaldar decisiones fundamentadas respecto a su eficacia, optimización y aplicabilidad en contextos reales. En el marco del presente estudio, los resultados obtenidos por cada algoritmo fueron analizados mediante técnicas estadísticas de comparación múltiple. Para ello, se utilizaron pruebas como Análisis de Varianzas (ANOVA) de un factor y Tukey HSD (Honest Significant Difference), que permitieron identificar diferencias significativas en el desempeño de los modelos evaluados.

Selección del mejor modelo

La selección del modelo de entrenamiento más adecuado en Machine Learning constituyó una etapa crucial en el desarrollo del sistema, en la que se consideraron múltiples factores para asegurar su efectividad. El modelo elegido no solo mostró un buen desempeño en el conjunto de entrenamiento, sino que también evidenció una sólida capacidad de generalización al enfrentarse a datos no previamente observados.

Despliegue de mejor modelo

Se realizó el proceso de despliegue, que permitió poner el modelo entrenado a disposición en un entorno de preproducción.

El objetivo principal del despliegue fue integrar el modelo dentro de un sistema operativo funcional, destinado a su utilización por usuarios finales o como parte de otros sistemas automatizados. La Figura 7 presenta la implementación del sistema diseñado para ofrecer una interfaz intuitiva, además se observa el ingreso de datos, a partir de los cuales el sistema realiza la predicción de la carga (Figura 8).



Figura 7: Interfaz del sistema

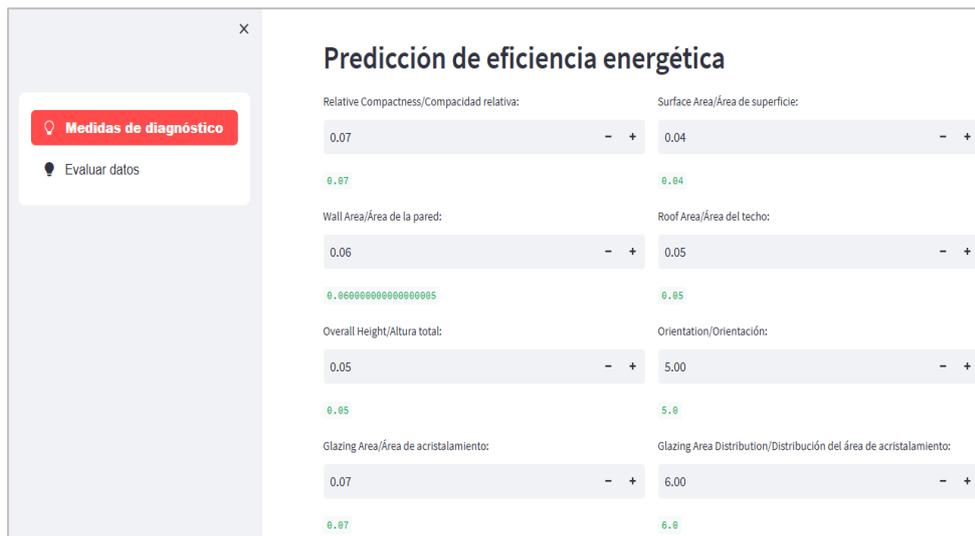
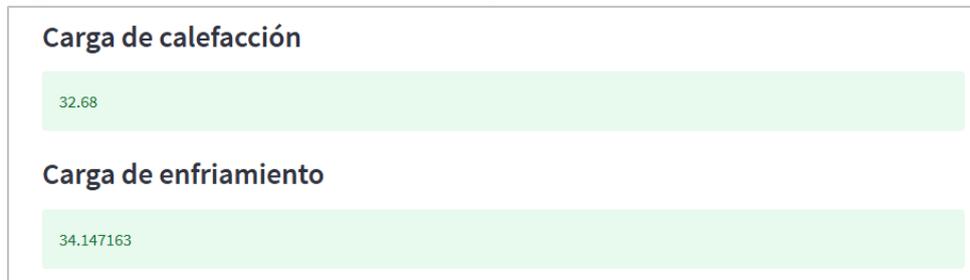


Figura 8: Resultados obtenidos sobre la predicción



RESULTADOS

Resultados sobre R² Score - Calefacción

Tal como se observa en la Tabla 2, el modelo que alcanzó el valor más alto en la métrica R² Score para la predicción de la “carga de calefacción” en edificios residenciales fue el Modelo Multiclasificador 1, con un 99.75%. Le siguen Decision Tree con 99.67% y XGBoost con 99.05%. Posteriormente se ubican el Modelo Multiclasificador 2 con 97.62% y KNN con 97.28%, seguidos por Random Forest con 97.08%.

Por otro lado, Regresión Lineal y Ridge alcanzaron valores similares, con 90.86% y 90.84% respectivamente. Finalmente, Lasso obtuvo un 51.28%, y ElasticNet presentó el desempeño más bajo con un 43.80%.

Tabla 2: Resultados de la métrica R² Score (Calefacción)

CV	Decision Tree	Regresión Lineal	Ridge	Lasso	Elastic Net	KNN	Random Forest	XG Boost	Modelo multiclasi ficador 1	Modelo multiclasi ficador 2
2	99.67%	90.86%	90.84%	51.28%	43.80%	97.11%	96.98%	98.03%	99.74%	96.95%
3	99.64%	90.86%	90.84%	51.28%	43.80%	97.11%	97.14%	98.72%	99.74%	97.07%
4	99.67%	90.86%	90.84%	51.28%	43.80%	97.11%	97.18%	98.90%	99.77%	97.88%
5	99.67%	90.86%	90.84%	51.28%	43.80%	97.12%	97.09%	98.90%	99.75%	98.34%
6	99.65%	90.86%	90.84%	51.28%	43.80%	97.11%	97.29%	99.32%	99.72%	97.67%
7	99.67%	90.86%	90.84%	51.28%	43.80%	97.11%	97.16%	99.33%	99.74%	96.86%
8	99.67%	90.86%	90.84%	51.28%	43.80%	97.19%	96.75%	99.33%	99.76%	97.91%
9	99.67%	90.86%	90.84%	51.28%	43.80%	98.60%	96.96%	99.32%	99.74%	98.66%
10	99.67%	90.86%	90.84%	51.28%	43.80%	97.19%	97.17%	99.32%	99.75%	97.21%
11	99.68%	90.86%	90.84%	51.28%	43.80%	97.11%	97.11%	99.33%	99.76%	97.69%
Prec.	99.67%	90.86%	90.84%	51.28%	43.80%	97.28%	97.08%	99.05%	99.75%	97.62%

Según lo mostrado en la Tabla 3, el valor de significancia obtenido en la prueba ANOVA fue de 0.000, lo cual es inferior al umbral de 0.05. Esto permite rechazar la hipótesis nula de igualdad entre los grupos evaluados y concluir que existe evidencia estadísticamente significativa de que al menos algunas de las medias difieren entre sí.

Tabla 3: Prueba ANOVA de R² Score (Calefacción)

ANOVA de un factor					
	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Entre grupos	39592,826	9	4399,203	55992,140	,000
Dentro de grupos	7,071	90	,079		
Total	39599,897	99			

De acuerdo con los resultados de la prueba de Tukey (ver Tabla 4), y considerando un nivel de significancia de 0.05, se identificó que no existen diferencias significativas entre ciertos pares de algoritmos. Por ejemplo, la comparación entre Decision Tree y Modelo Multiclasi ficador 1, así como entre Regresión Lineal y Ridge, arrojó un valor de 1.000, lo que indica equivalencia estadística. Asimismo, la comparación entre KNN y Random Forest resultó en un valor de 0.872, y entre KNN y Modelo Multiclasi ficador 2 en 0.161, por lo que, en ambos casos, se acepta la hipótesis nula. No obstante, en otras comparaciones como Ridge frente a Decision Tree, o Lasso frente a XGBoost, se obtuvo un valor de 0.000, lo que indica la existencia de diferencias estadísticamente significativas entre los modelos evaluados.



Tabla 4: Resultados de prueba Tukey según la métrica R² Score (Calefacción) - pares de modelos sin diferencias significativas

Comparaciones múltiples						
Variable dependiente: valor HSD Tukey						
(I) algoritmo	(J) algoritmo	Diferencia de medias (I-J)	Desv. Error	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
Decision_tree	Modelo multclasificador 1	-,08100	,12535	1,000	-,4877	,3257
Regresion_LINEAL	RIDGE	,02000	,12535	1,000	-,3867	,4267
RIDGE	Regresion_Lineal	-,02000	,12535	1,000	-,4267	,3867
KNN	Modelo Multclasificador 2	-,34800	,12535	,161	-,7547	,0587
RANDOM_FOREST	KNN	-,19300	,12535	,872	-,5997	,2137
Modelo Multclasificador 1	DECISION_TREE	,08100	,12535	1,000	-,3257	,4877
Modelo Multclasificador 2	KNN	,34800	,12535	,161	-,0587	,7547

Resultados sobre R² Score - Enfriamiento

Tal como se presenta en la Tabla 5, el modelo que alcanzó el mayor valor en la métrica R² Score para la predicción de la carga de enfriamiento en edificios residenciales fue XGBoost, con un 99.05%. Le siguen Modelo Multclasificador 1 con 98.56% y KNN con 97.13%. Posteriormente se ubican Random Forest (97.04%) y Decision Tree (95.31%). A continuación, se encuentran Modelo Multclasificador 2 con 93.69%, Regresión Lineal con 88.62% y Ridge con 88.61%. Finalmente, los modelos con menor desempeño fueron Lasso, con 51.28%, y ElasticNet, con 44.24%.

Tabla 5: Resultados de la métrica R² Score (Enfriamiento)

CV	Decision Tree	Regresión Lineal	Ridge	Lasso	Elastic Net	KNN	Random Forest	XG Boost	Modelo multclasificador 1	Modelo multclasificador 2
2	95.47%	88.62%	88.61%	51.28%	44.24%	97.11%	97.09%	98.03%	97.84%	91.77%
3	94.94%	88.62%	88.61%	51.28%	44.24%	97.11%	97.22%	98.72%	97.59%	93.82%
4	95.33%	88.62%	88.61%	51.28%	44.24%	97.11%	96.91%	98.90%	98.52%	92.92%
5	95.33%	88.62%	88.61%	51.28%	44.24%	97.12%	97.10%	98.90%	98.75%	93.43%
6	95.33%	88.62%	88.61%	51.28%	44.24%	97.11%	97.02%	99.32%	98.52%	94.23%
7	95.33%	88.62%	88.61%	51.28%	44.24%	97.11%	97.01%	99.33%	99.01%	94.66%
8	95.34%	88.62%	88.61%	51.28%	44.24%	97.19%	97.09%	99.33%	98.76%	94.01%
9	95.33%	88.62%	88.61%	51.28%	44.24%	97.11%	96.99%	99.32%	98.66%	93.58%
10	95.33%	88.62%	88.61%	51.28%	44.24%	97.19%	96.84%	99.32%	99.05%	94.72%
11	95.33%	88.62%	88.61%	51.28%	44.24%	97.11%	97.12%	99.33%	98.87%	93.77%
Prec.	95.31%	88.62%	88.61%	51.28%	44.24%	97.13%	97.04%	99.05%	98.56%	93.69%



Según lo indicado en la Tabla 6, el valor de significancia obtenido en la prueba ANOVA fue de 0.000, lo cual es inferior al umbral de 0.05. Esto permite rechazar la hipótesis nula de igualdad entre los grupos evaluados y concluir que existen diferencias significativas entre al menos algunas de las medias comparadas.

Tabla 6: Prueba ANOVA de R² Score (Enfriamiento)

ANOVA de un factor					
	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Entre grupos	36782,570	9	4086,952	34065,379	,000
Dentro de grupos	10,798	90	,120		
Total	36793,367	99			

De acuerdo con los resultados de la prueba de Tukey (Tabla 7) y considerando un nivel de significancia de 0.05, no se identificaron diferencias estadísticamente significativas entre algunos pares de algoritmos. En particular, las comparaciones entre Regresión Lineal y Ridge, así como entre KNN y Random Forest, arrojaron un valor de 1.000, indicando similitud en su desempeño. Asimismo, la comparación entre XGBoost y Modelo Multiclasificador 1 presentó un valor de 0.059, lo que también llevó a aceptar la hipótesis nula. No obstante, en otras comparaciones —como Ridge frente a Decision Tree o ElasticNet frente a KNN— se obtuvo un valor de 0.000, lo que evidencia diferencias estadísticamente significativas entre los modelos evaluados.

Tabla 7: Resultados de prueba Tukey según la métrica R² Score (Enfriamiento) - pares de modelos sin diferencias significativas

Comparaciones múltiples						
Variable dependiente: valor HSD Tukey						
(I) algoritmo	(J) algoritmo	Diferencia de medias (I-J)	Desv. Error	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
Regresion_Lineal	Ridge	,01000	,15490	1,000	-,4926	,5126
Ridge	Regresion_Lineal	-,01000	,15490	1,000	-,5126	,4926

DISCUSIÓN

Comparativa con investigaciones anteriores

El algoritmo de machine learning que mostró el mejor desempeño en la predicción de la carga térmica en edificios residenciales fue el Modelo Multiclasificador 1, con un R² Score de 99.75%.



En cuanto a la predicción de la carga de enfriamiento, el modelo más eficiente fue XGBoost, alcanzando un R² Score de 99.05%.

Estos resultados coinciden con estudios previos, como el de Zhou et al. (2021), quienes demostraron que los modelos GPR y MARS obtuvieron un R² de 95.6% para la carga de calefacción, concluyendo que los algoritmos de *machine learning* son herramientas eficaces para predecir el consumo energético en edificaciones. De manera similar, Moayedi et al. (2019) reportaron que el algoritmo Random Forest alcanzó un R² de 99.89%, evidenciando su alta precisión en la predicción de cargas térmicas en edificios energéticamente eficientes.

Asimismo, Mohammed et al. (2021) encontraron que XGBoost logró un R² Score de 97.30% en la predicción de la carga de enfriamiento, destacando que el uso de métodos combinados puede mejorar significativamente la precisión del modelo. Por su parte, Sadeghi et al. (2020) reportaron que las redes neuronales profundas alcanzaron un R² Score de 99.44% en la predicción de carga de enfriamiento, concluyendo que los modelos de *machine learning* superan a otros enfoques tradicionales utilizados en la literatura.

CONCLUSIONES

Se evaluaron diez modelos, incluyendo dos diseñados mediante la técnica de ensamble Stacking, con el objetivo de predecir la carga de calefacción y enfriamiento en edificios residenciales y se identificó que las técnicas con mejor desempeño en términos de R² para la predicción de la carga térmica en edificios residenciales durante las pruebas fueron el 'Modelo Multiclasificador 1', con un 99.75%, para la carga de calefacción, y 'XGBoost', con un 99.05%, para la carga de enfriamiento. Posteriormente, mediante la prueba ANOVA de Friedman aplicada a los valores de R² y las pruebas post hoc, se obtuvieron nivel de significancia de 0.00. Esto indica que los modelos logran explicar un alto porcentaje de la variabilidad en la carga térmica de los edificios, lo que sugiere una alta capacidad predictiva en este contexto.

Además, estos resultados permiten concluir que el uso de modelos avanzados basados en *machine learning* no solo mejora la precisión en la estimación de cargas térmicas, sino que también ofrece una herramienta eficiente para apoyar la contratación de estas cargas en edificios residenciales.



Es importante destacar que los modelos mas precisos logran una generalización adecuada sin comprometer el rendimiento en lo datos de prueba, gracias al uso de validación cruzada y técnicas de preprocesamiento. Asimismo, la superioridad del modelo multclasificador puede atribuirse a la combinación de los algoritmos complementarios, lo que refuerza la utilidad de enfoques híbridos en este tipo de problemas. En conjunto, los hallazgos respaldan el potencial de estas técnicas como soporte para decisiones en términos de sostenibilidad y eficiencia energética.

REFERENCIAS BIBLIOGRAFICAS

- Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54, 1937-1967. <https://doi.org/10.1007/s10462-020-09896-5>
- Burkov, A. (2019). *The hundred-page machine learning book*. Canada: Andriy Burkov.
- Gelman, A., Goodrich, B., Gabry, J., & Vehtari, A. (2019). R-squared for Bayesian regression models. *The American Statistician*, 73(3), 307-309. <https://doi.org/10.1080/00031305.2018.1549100>
- Guo, J., Yun, S., Meng, Y., He, N., Ye, D., Zhao, Z., ... & Yang, L. (2023). Prediction of heating and cooling loads based on light gradient boosting machine algorithms. *Building and Environment*, 236, 110252. <https://doi.org/10.1016/j.buildenv.2023.110252>
- Jiang, H. (2021). *Machine learning fundamentals: A concise introduction*. Kingdom of England: Cambridge University Press. <https://doi.org/10.1017/9781108938051>
- Joshi, A. V. (2020). *Machine learning and artificial intelligence*. United States: Springer.
- Mahdi, G. J. M., Mohammed, N. J., & Al-Sharea, Z. I. (2021, May). Regression shrinkage and selection variables via an adaptive elastic net model. *Journal of Physics: Conference Series*, 1879(3), 032014. <https://doi.org/10.1088/1742-6596/1879/3/032014>
- Moayedi, H., Bui, D. T., Dounis, A., Lyu, Z., & Foong, L. K. (2019). Predicting heating load in energy-efficient buildings through machine learning techniques. *Applied Sciences*, 9(20), 4338. <https://doi.org/10.3390/app9204338>



- Mohammed, A. S., Asteris, P. G., Koopialipour, M., Alexakis, D. E., Lemonis, M. E., & Armaghani, D. J. (2021). Stacking ensemble tree models to predict energy performance in residential buildings. *Sustainability*, 13(15), 8298. <https://doi.org/10.3390/su13158298>
- Moradzadeh, A., Mansour-Saatloo, A., Mohammadi-Ivatloo, B., & Anvari-Moghaddam, A. (2020). Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings. *Applied Sciences*, 10(11), 3829. <https://doi.org/10.3390/app10113829>
- Roy, S. S., Samui, P., Nagtode, I., Jain, H., Shivaramakrishnan, V., & Mohammadi-Ivatloo, B. (2020). Forecasting heating and cooling loads of buildings: A comparative performance analysis. *Journal of Ambient Intelligence and Humanized Computing*, 11, 1253-1264. <https://doi.org/10.1007/s12652-019-01317-y>
- Sadeghi, A., Younes Sinaki, R., Young, W. A., & Weckman, G. R. (2020). An intelligent model to predict energy performances of residential buildings based on deep neural networks. *Energies*, 13(3), 571. <https://doi.org/10.3390/en13030571>
- Thevaraja, M., Rahman, A., & Gabirial, M. (2019, April). Recent developments in data science: Comparing linear, ridge and lasso regressions techniques using wine data. In *Proceedings of the international conference on digital image & signal processing* (pp. 1-6).
- Trappenberg, T. P. (2020). *Fundamentals of machine learning*. United Kingdom: Oxford University Press.
- Zhou, Y., Liu, Y., Wang, D., & Liu, X. (2021). Comparison of machine-learning models for predicting short-term building heating load using operational parameters. *Energy and Buildings*, 253, 111505. <https://doi.org/10.1016/j.enbuild.2021.111505>

