Maestría en Economía – Universidad Nacional de La Plata

Problem set 2 - Machine Learning

Alumnos: Ara Portentoso Francisco, Corradi Valentin, Di Placido Pedro

15 de diciembre de 2024

Se puede acceder al repositorio público donde se encuentran los códigos y demás elementos asociados al trabajo a partir del siguiente link:
https://github.com/dipla70/ML-TP2.git

Introducción

La pobreza representa uno de los principales desafíos económicos y sociales en países en desarrollo, ya que limita el acceso a servicios básicos como educación, salud y vivienda digna, afectando directamente las condiciones de vida de las personas. Según el Departamento Administrativo Nacional de Estadística (DANE), en 2018 la tasa de pobreza monetaria en Colombia alcanzó el 27,0%, mientras que la pobreza extrema se situó en un 7,2% a nivel nacional. Estas cifras resaltan la urgencia de implementar metodologías robustas que permitan medir, analizar y predecir la pobreza, con el objetivo de orientar políticas públicas más efectivas.

En este estudio, se aplican y comparan diversos modelos de aprendizaje automático para predecir la condición de pobreza de los hogares. Se utilizan modelos de regresión regularizada como Lasso y Elastic Net, conocidos por su capacidad para manejar problemas de colinealidad entre variables. Asimismo, se emplean técnicas de ensamble como Random Forest, Bagging Random Forest y AdaBoost, las cuales combinan múltiples árboles de decisión para mejorar la precisión de las predicciones. Además, se consideran enfoques tradicionales como Logit y CART (Classification and Regression Trees), con el fin de obtener una evaluación integral del desempeño de cada modelo, tomando como referencia el F1-Score, una métrica que balancea precisión y exhaustividad en las predicciones.

La medición tradicional de la pobreza por ingreso se basa en establecer un umbral de pobreza y una función indicadora. Si el ingreso de un hogar supera dicho umbral, no se considera pobre; de lo contrario, el hogar es clasificado como pobre. No obstante, este enfoque presenta ciertos desafíos prácticos. Fitzpatrick et al. (2018) sostienen que las encuestas nacionales de hogares, diseñadas para recolectar datos de consumo o ingreso, suelen ser complejas en su ejecución. Por su parte, McBride y Nichols (2018) argumentan que estas encuestas son costosas y demandan mucho tiempo, lo que limita su implementación frecuente.

Debido a estas dificultades, surge el interés en explorar alternativas más rápidas y menos costosas que permitan predecir la pobreza utilizando técnicas de machine learning. Estas técnicas aprovechan otras variables disponibles para alcanzar resultados precisos y eficientes. Existen trabajos previos que respaldan este enfoque. Por ejemplo, Solis-Salazar y Madrigal-Sanabria (2022) en *"A machine learning proposal to predict poverty"* utilizan XGBoost para predecir pobreza y concluyen que este enfoque ofrece un mejor balance entre errores de inclusión y exclusión. En un contexto similar, Quin Li

et al. (2022), en *"Is poverty predictable with machine learning? A study of DHS data from Kyrgyzstan"*, comparan XGBoost con GLM y encuentran que el primero supera al segundo en la mayoría de los casos, destacando la utilidad del machine learning para la selección de variables.

Otros estudios, como el de Hamzan et al. (2022), en *"Poverty prediction using machine learning approach"*, proponen el uso del Random Forest Regressor, logrando un destacado score de 0.9462. Finalmente, Hassan et al. (2024) en *"Machine learning study using 2020 SDHS data to determine poverty determinants in Somalia"*, encuentran que el modelo de Random Forest presenta la mejor performance, con un accuracy del 98,36%. Estos antecedentes evidencian la efectividad de los enfoques de machine learning para predecir la pobreza, ofreciendo soluciones rápidas, precisas y eficientes. En este trabajo, buscamos replicar este enfoque aplicando diferentes modelos de machine learning al contexto colombiano, evaluando su desempeño y utilidad para la predicción de la pobreza a nivel hogar.

Datos

El principal objetivo de este trabajo es construir un modelo que permita predecir la pobreza de los hogares. La clasificación de un hogar se realiza mediante la siguiente función indicadora: $\text{poor} = I(\text{Ing} < \text{Pl})$, donde Ing representa el ingreso del hogar y Pl es el umbral de pobreza. Para abordar este problema, se plantearon dos enfoques: clasificación, donde se predice un valor de 0 si el hogar no es pobre y 1 si lo es, o predicción del ingreso, evaluando luego si este se encuentra por encima o por debajo del umbral de pobreza.

Los datos utilizados provienen de la encuesta "Medición de Pobreza Monetaria y Desigualdad 2018", elaborada por el Departamento Administrativo Nacional de Estadística (DANE) de Colombia. La base de datos original incluye información tanto a nivel de hogares como de personas. Si bien el objetivo principal es estimar la pobreza a nivel de hogar, la capacidad de emparejar a las personas con el hogar al que pertenecen permite extraer información adicional de la base de personas, lo que enriquece el análisis y mejora el proceso de predicción.

Dado que la muestra original ya se encontraba dividida en un conjunto de entrenamiento y un conjunto de prueba, el primer paso consistió en filtrar las variables para garantizar que estuvieran presentes en ambas muestras. La base de entrenamiento inicial contenía más variables que la de prueba, por lo que fue necesario asegurar consistencia al momento de testear el modelo.

Al momento de seleccionar las variables predictoras, se estableció como condición que estas estuvieran presentes en ambas bases (entrenamiento y prueba). Esta decisión fue crucial dado que no sería posible predecir la variable de interés si el modelo utilizara variables que no estuvieran disponibles en el conjunto de prueba. En el **Anexo** puede consultarse la lista completa de las variables seleccionadas en cada data frame.

Además, se incluyó la variable **'Pobre'** (pobreza del hogar) para el enfoque de estimación directa. Otro procedimiento importante fue la eliminación de variables duplicadas presentes en ambas bases (hogares y personas), con el objetivo de evitar

doble contabilización y reducir posibles sesgos en los resultados. Estas decisiones metodológicas aseguran la coherencia en el análisis y optimizan la capacidad predictiva de los modelos utilizados.
Durante este proceso, ademas se incorporaron nuevas variables con el objetivo de capturar información relevante a nivel de hogar, tomando como fuente los registros individuales de las personas. Estas transformaciones permiten generar nuevos predictores con la informacion disponible que podrian mejorar las estimaciones.

Una de las variables creadas es la **recepción de subsidios**, que se construyó a partir de la información individual de los miembros del hogar. Para ello, se identificó si al menos una persona dentro del hogar reportó haber recibido algún subsidio, lo cual permitió obtener una variable binaria que indica si el hogar tuvo acceso.

Otra variable generada fue la **proporción de hombres en cada hogar**. Para calcular esta proporción, se consideró la cantidad de hombres en el hogar en relación con el total de personas que lo integran. La variable resultante, denominada *prop_sexo*, permite analizar la composición de género dentro del hogar y su posible asociación con la condición de pobreza.

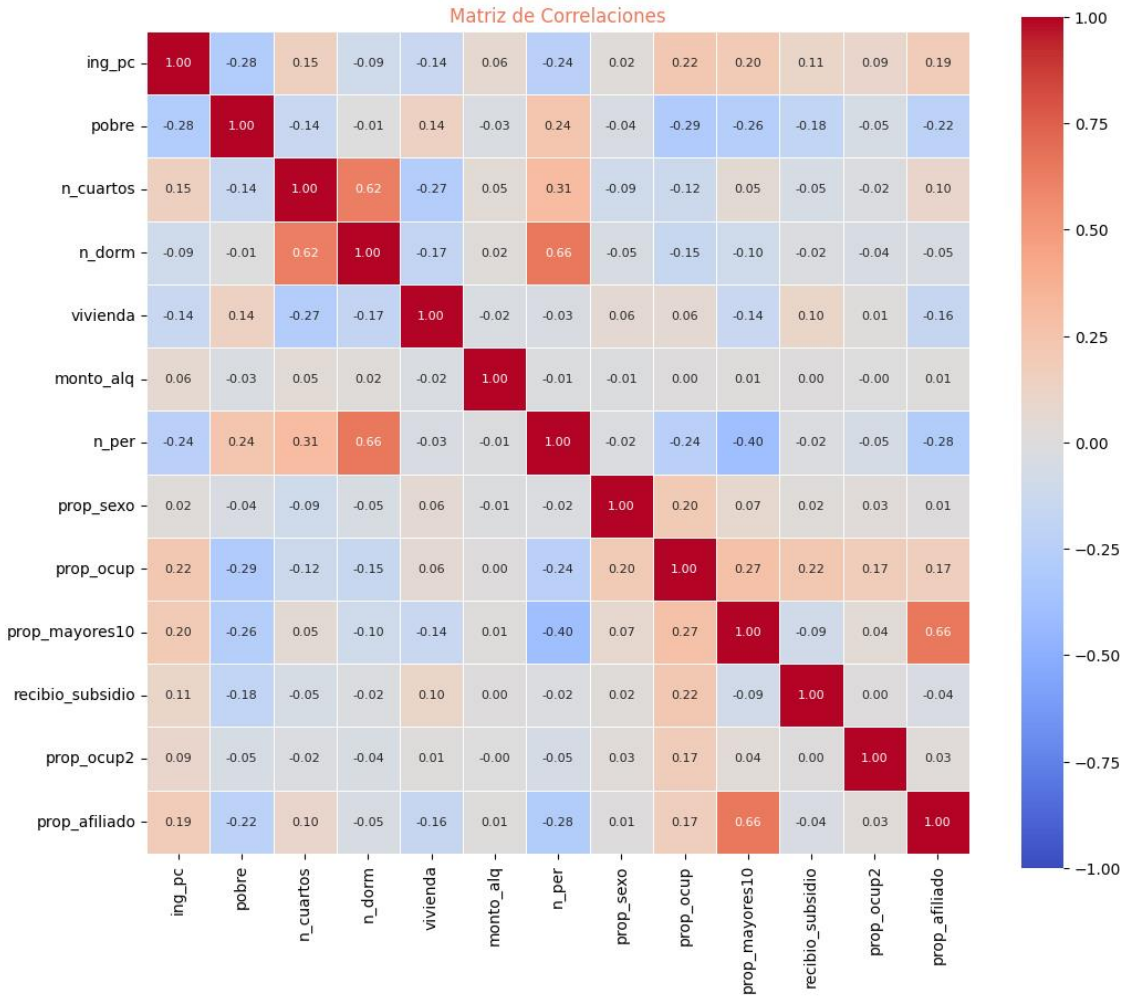
En cuanto a la actividad laboral general del hogar, se construyó la **proporción de personas ocupadas,** calculando el promedio de personas que reportaron estar ocupadas dentro del hogar. Esta variable refleja el nivel de actividad laboral agregado y proporciona información clave sobre la capacidad del hogar para generar ingresos a través del trabajo.

Por último, se incluyó la **proporción de personas mayores de 10 años** en cada hogar, capturando así la composición etaria de los hogares. Esta variable permite analizar la estructura demográfica de cada familia y evaluar la presencia de personas en edad productiva, lo que puede influir en las condiciones económicas del hogar.

Análisis de los datos

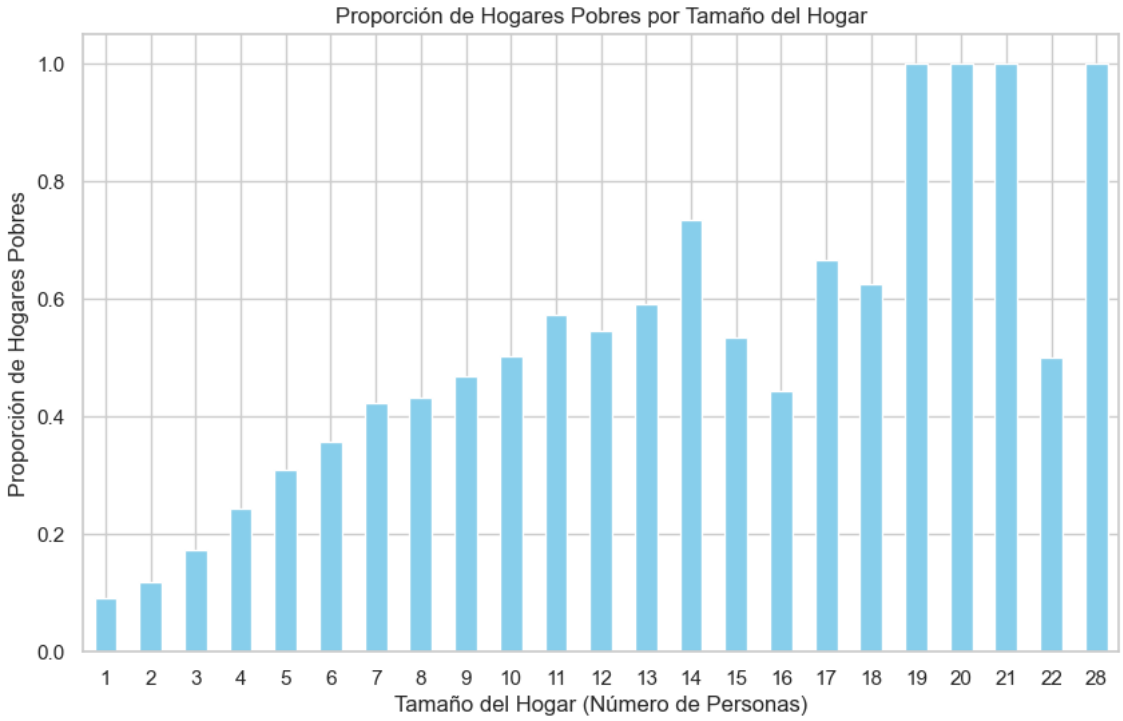
Con el objetivo de comprender mejor la información, realizamos un análisis exploratorio de los datos.

El siguiente gráfico presenta una matriz de correlaciones que muestra las relaciones lineales entre diferentes variables del dataset. La correlación se mide a través del coeficiente de Pearson, con valores que oscilan entre -1 y 1. Los colores reflejan la magnitud y dirección de las correlaciones: los tonos rojos indican correlaciones positivas, mientras que los azules representan correlaciones negativas. Los valores cercanos a cero aparecen en colores más neutros, lo que sugiere una relación débil o inexistente entre las variables.



Si analizamos las correlaciones de la variable pobre vemos que la correlación negativa más destacada se observa con el ingreso per cápita (ing_pc), con un valor de -0.28, lo que refleja que a medida que aumenta el ingreso del hogar, disminuye la probabilidad de ser clasificado como pobre. De manera similar, la proporción de personas ocupadas (prop_ocup) presenta una correlación negativa de -0.29, indicando que en hogares con mayor participación laboral, la pobreza tiende a ser menor. La proporción de personas mayores de 10 años (prop_mayores10) también muestra una relación negativa moderada (-0.26), sugiriendo que una mayor presencia de individuos en edad de trabajar reduce la incidencia de pobreza. En contraste, variables asociadas al tamaño del hogar, como el número de personas (n_per) y el número de habitaciones ocupadas (n_dorm), tienen correlaciones positivas de 0.24 y 0.14, respectivamente, lo que sugiere que hogares más numerosos o con condiciones de mayor hacinamiento tienen una mayor probabilidad de ser pobres.

La cantidad de personas es una variable predictora importante para explicar pobreza. Esto implica que las familias mas pobres tienden a ser numerosas. Para ello, graficamos la proporción de hogares pobres por tamaño de hogar en el siguiente gráfico.



En hogares pequeños, la pobreza es menor, pero a partir de 7 personas, la proporción crece notablemente, superando el 50% en hogares más numerosos. En hogares con 19 o más personas, la pobreza alcanza casi el 100%, evidenciando que los hogares grandes son más vulnerables económicamente.

Metodología

A continuación se enumeran los modelos utilziados.

Regresión Lineal

La regresión lineal asume una relación lineal entre las variables independientes (X) y la variable dependiente (y). La ecuación básica es:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

Donde (β_0) es el intercepto, (β_j) son los coeficientes que indican la influencia de cada variable independiente, y (ϵ) es el término de error.

En el problema set decidimos no utilizar esta versión básica lineal para poder usar Lasso.

Ridge

La regresión Ridge introduce una penalización (L_2) para reducir el sobreajuste y manejar la multicolinealidad. Su función de pérdida es:

$$[L(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2]$$

Donde (λ) controla la magnitud de la penalización.

Lasso (Least Absolute Shrinkage and Selection Operator)

Lasso utiliza penalización (L_1), que puede forzar algunos coeficientes a ser exactamente cero, permitiendo la selección automática de variables. La función de pérdida es:

$$[L(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|]$$

Esto hace que Lasso sea adecuado para problemas con muchas variables redundantes.

Elastic Net

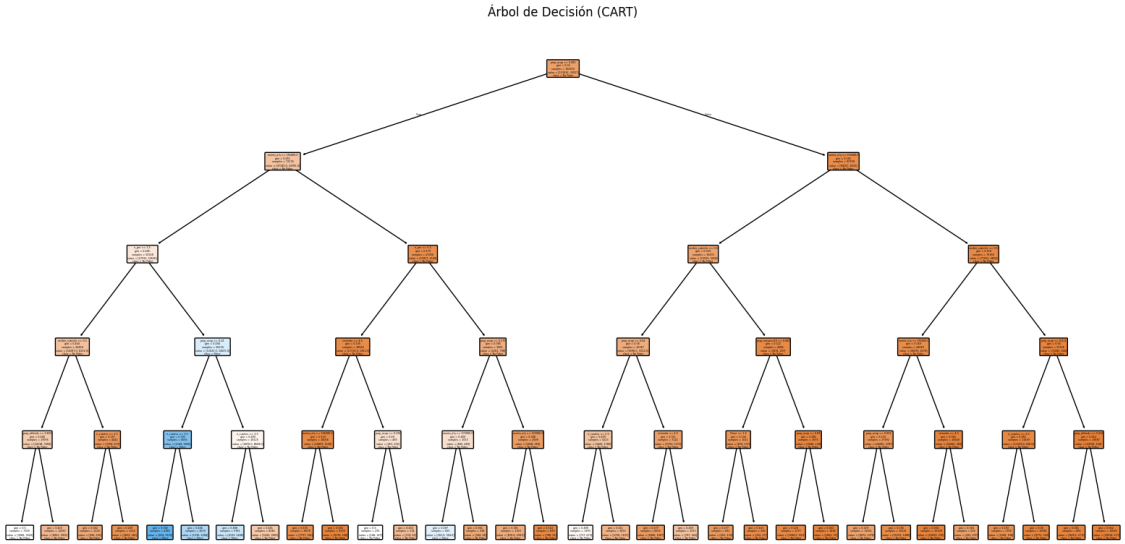
Elastic Net combina las penalizaciones (L_1) (Lasso) y (L_2) (Ridge). Su función de pérdida es:

$$[L(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2]$$

Elastic Net es útil cuando hay muchas variables correlacionadas, ya que combina las ventajas de Ridge y Lasso.

Árboles de Decisión

Los árboles de decisión dividen los datos en subconjuntos basados en condiciones sobre las variables independientes. El objetivo es minimizar una métrica como el MSE (para regresión) o el Gini Impurity (para clasificación) en cada división.



Random Forest

Random Forest utiliza el bagging para combinar múltiples árboles de decisión, mejorando la precisión y reduciendo el sobreajuste. La predicción final es:

$$[\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x)]$$

Donde (T) es el número de árboles y $(f_t(x))$ es la predicción del árbol (t).

Bagging (Random Forest)

Bagging (Bootstrap Aggregating) es una técnica de aprendizaje conjunto que se utiliza para mejorar la estabilidad y precisión de los modelos de aprendizaje automático al reducir el sobreajuste. Su enfoque principal es entrenar múltiples modelos (como árboles de decisión) en subconjuntos aleatorios de los datos de entrenamiento, generados mediante muestreo con reemplazo. La predicción final es una combinación de las predicciones de todos los modelos.

El algoritmo Random Forest es una extensión de Bagging que crea múltiples árboles de decisión, cada uno entrenado con un subconjunto aleatorio de los datos y un subconjunto aleatorio de características. La predicción final para clasificación es el voto mayoritario, mientras que para regresión es el promedio de las predicciones.

La fórmula para la predicción final en clasificación es:

$$\hat{y} = \text{modo}\{f_1(x), f_2(x), \ldots, f_T(x)\}$$

Y en regresión:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x)$$

Donde:
- T: Número de modelos base (árboles de decisión).
$f_t(x)$: Predicción del árbol \(t\) para la instancia \(x\).

El uso de Random Forest es especialmente útil cuando los datos tienen muchas variables, ya que reduce la varianza del modelo y mejora la generalización.

Adaboost

Adaboost (Adaptive Boosting) es otro método de aprendizaje conjunto que combina múltiples modelos base (usualmente árboles de decisión débiles) para crear un modelo más robusto. La idea principal es entrenar los modelos secuencialmente, dando mayor peso a las instancias mal clasificadas en iteraciones previas. Esto permite que el modelo final se enfoque en los datos más difíciles.

El peso de cada modelo base depende de su precisión, y las predicciones finales se obtienen mediante una combinación ponderada de los modelos individuales. La función de predicción en Adaboost se expresa como:

$$F(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x)\right)$$

Donde:
- $(F(x))$: Predicción final.
- (T): Número de modelos base.
- (α_t): Peso asignado al modelo (f_t), proporcional a su precisión.
- $(f_t(x))$: Predicción del modelo base t para la instancia x.

Adaboost es especialmente efectivo para modelos simples, como árboles de decisión con una sola división ("stumps"), y suele mejorar el rendimiento en problemas con datos desbalanceados.

Logit

El modelo Logit, también conocido como regresión logística, se utiliza para modelar relaciones entre un conjunto de variables independientes y una variable dependiente binaria. Es ampliamente utilizado en clasificación binaria. La idea principal es modelar la probabilidad de que una observación pertenezca a una de las dos categorías.

La probabilidad de que la variable dependiente \(y\) sea igual a 1, dada una combinación lineal de las variables independientes \(X\), se expresa mediante la función logística:

$$P(y = 1|X) = \frac{1}{1 + e^{-\beta_0 - \beta_1 X_1 - \cdots - \beta_p X_p}}$$

Donde:
- P(y=1|X): Probabilidad condicional de la categoría positiva.
- β_0: Intercepto del modelo.
- $\beta_1, \ldots, \beta_p$: Coeficientes asociados a las variables predictoras.
$X_1, \ldots, X_p$: Variables predictoras.

La clasificación final se realiza utilizando un umbral. Por defecto, se asigna la categoría 1 si la probabilidad es mayor o igual a 0.5:

$$\hat{y} = \begin{cases} 1 & \text{si } P(y = 1|X) \geq 0.5 \\ 0 & \text{si } P(y = 1|X) < 0.5 \end{cases}$$

El modelo Logit se entrena maximizando la verosimilitud de los datos observados, lo que permite estimar los parámetros \(\beta\). Es adecuado para problemas donde la variable objetivo tiene dos categorías y las relaciones entre las variables independientes y la probabilidad logit son lineales.

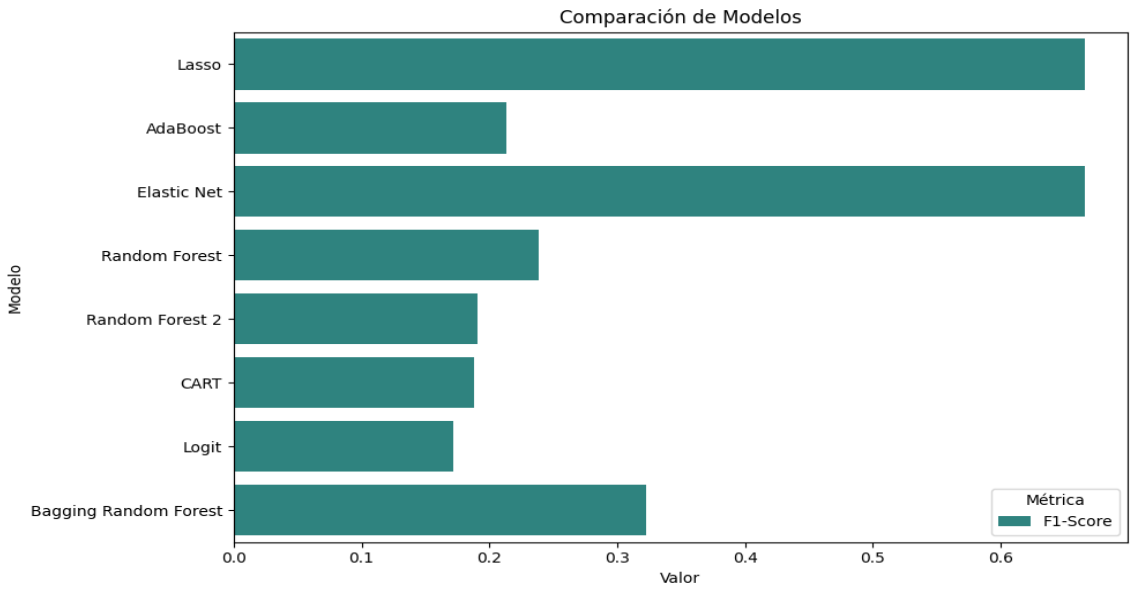
En el proceso de entrenamiento y evaluación de los modelos, se implementaron técnicas avanzadas de validación y ajuste de hiperparámetros para garantizar resultados robustos y evitar problemas de sobreajuste (overfitting). Se utilizó validación cruzada K-Fold para evaluar el desempeño de los modelos, dividiendo el conjunto de datos en 5 particiones

Adicionalmente, se aplicó Grid Search en combinación con K-Fold para optimizar los hiperparámetros de algunos modelos específicos. En particular, se utilizó para ajustar los parámetros del modelo Elastic Net mediante la función GridSearchCV, donde se exploraron diferentes combinaciones de l1_ratio y alpha. Esta búsqueda sistemática permitió encontrar los valores óptimos que maximizan la métrica F1-Score. De forma similar, en otros casos, se implementó Randomized Search para reducir el tiempo de cómputo y explorar un rango más amplio de hiperparámetros en modelos complejos como ElasticNetCV.

Por otra parte, en el caso del Bagging Random Forest y otros modelos de árbol como Random Forest y CART, se utilizó la validación cruzada K-Fold sin la optimización explícita de hiperparámetros, centrándose en evaluar el rendimiento del modelo con los parámetros predeterminados. Finalmente, esta combinación de validación cruzada y búsqueda de hiperparámetros permitió comparar de manera objetiva el desempeño de cada modelo bajo condiciones consistentes y replicables.

Comparación de modelos

A continuación, se presenta una comparación del desempeño de distintos modelos de aprendizaje automático utilizados para predecir la condición de pobreza en los hogares. El F1-Score fue seleccionado como métrica principal, dado que permite evaluar el equilibrio entre precisión y exhaustividad en un problema de clasificación, lo cual es crucial en un contexto donde los errores de inclusión y exclusión pueden tener consecuencias significativas.



El gráfico muestra una comparación del rendimiento de varios modelos de aprendizaje automático en función del F1-Score, una métrica que balancea precisión y exhaustividad. El modelo Lasso destaca con el mayor F1-Score, superando el valor de 0.6, seguido de cerca por el modelo Elastic Net, que también alcanza un rendimiento sobresaliente. En contraste, otros modelos como Random Forest y Bagging Random Forest presentan un F1-Score intermedio, ubicándose alrededor de 0.3. Modelos como AdaBoost, Random Forest 2, CART y Logit muestran un desempeño relativamente más bajo, con valores de F1-Score por debajo de 0.3. En resumen, Lasso y Elastic Net sobresalen como las mejores opciones en términos de F1-Score, mientras que los demás modelos presentan un rendimiento considerablemente inferior.

Modelo ganador en Kaggle

El modelo Random Forest fue seleccionado como el mejor debido a su rendimiento superior en términos de F1-Score, mostrando una mejor capacidad para balancear precisión y exhaustividad en la predicción de pobreza. Basado en la combinación de múltiples árboles de decisión mediante bagging, este modelo mejora la generalización y reduce el riesgo de sobreajuste al entrenar árboles en subconjuntos aleatorios de datos y combinar sus predicciones.

Una de sus principales ventajas es su capacidad para capturar relaciones no lineales entre variables y la variable objetivo, sin requerir suposiciones estrictas sobre la

distribución de los datos. Esto lo hace ideal para contextos complejos como la predicción de pobreza. Sin embargo, presenta desventajas como su menor interpretabilidad frente a modelos más simples, como el Logit, y un mayor tiempo de entrenamiento en datasets grandes.

Es importante destacar que el F1-Score obtenido en Kaggle difiere del F1-Score calculado en nuestros resultados, lo cual podría deberse a diferencias en el manejo de la base de datos, las particiones de train y test, o la implementación del modelo.

Conclusiones

En este trabajo se implementaron y compararon diferentes modelos de aprendizaje automático con el objetivo de predecir la condición de pobreza en los hogares utilizando datos de la encuesta "Medición de Pobreza Monetaria y Desigualdad 2018" del DANE en Colombia. A través del análisis, se evidenció que los modelos Lasso y Elastic Net presentaron el mejor desempeño, alcanzando los valores más altos de F1-Score, destacando su capacidad para manejar variables colineales y seleccionar las más relevantes.
Los árboles de decisión (CART) son interpretables y capturan relaciones no lineales, pero tienden a sobreajustarse sin poda. Random Forest y Bagging Random Forest mejoran la precisión al combinar múltiples árboles, reduciendo la varianza, aunque sacrifican interpretabilidad y son más costosos computacionalmente. AdaBoost, al enfocarse en datos difíciles mediante boosting, logra mayor precisión en modelos débiles, pero es sensible a valores atípicos.

Finalmente, el modelo Logit es simple, interpretable y efectivo en relaciones lineales, aunque limitado frente a métodos más complejos para capturar no linealidades. En síntesis, cada modelo ofrece ventajas y desventajas que lo hacen adecuado según el contexto y los datos disponibles.

Apendice

Variables Seleccionadas

Base de Hogares (train_hogares):

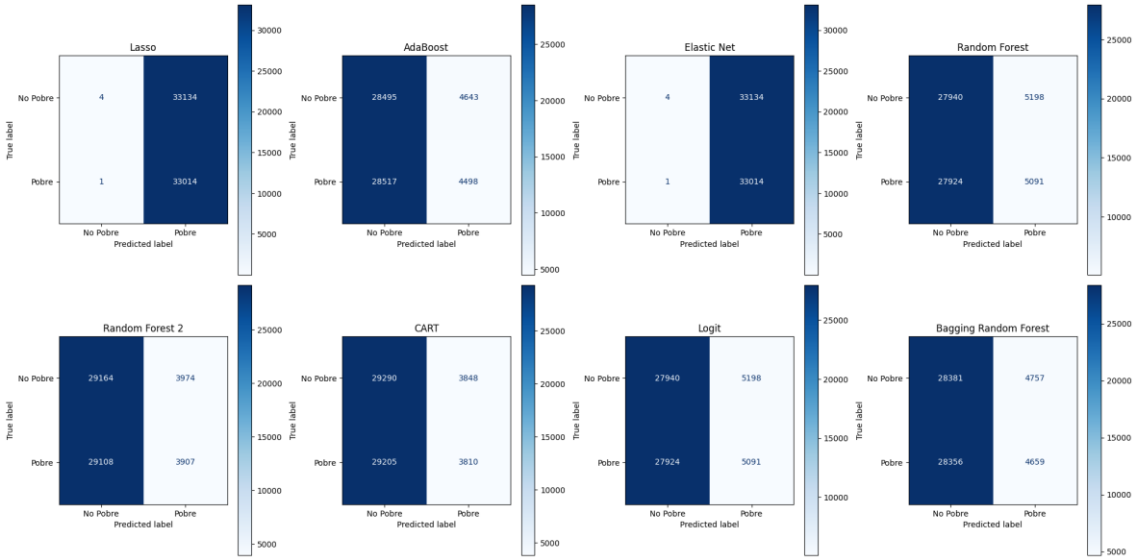
1. id: Identificador único de cada hogar.
2. Pobre: Variable binaria que indica si el hogar está en situación de pobreza (1: sí, 0: no).
3. Ingpcug: Ingreso per cápita del hogar, calculado como el ingreso total dividido por el número de personas.
4. P5010: Número de habitaciones utilizadas para dormir por las personas del hogar.
5. P5090: Régimen de tenencia de la vivienda (propia, alquilada, cedida, etc.).

6. P5130: Monto de alquiler pagado mensualmente (si corresponde).
7. Nper: Número total de personas que residen en el hogar.
8. Clase: Clasificación del hogar según su ubicación (cabecera o resto del área).

Base de Personas (train_personas):

1. id: Identificador único de cada individuo.
2. P6020: Sexo de la persona (1: masculino, 2: femenino).
3. P6040: Edad de la persona en años cumplidos.
4. P6050: Relación de parentesco con el jefe del hogar.
5. P6090: Afiliación a algún sistema de seguridad social.
6. P6210: Nivel educativo máximo alcanzado.
7. P6430: Tipo de empleo (asalariado, independiente, desempleado, etc.).
8. P6585s1: Indicador de si recibe algún tipo de subsidio alimentario.
9. P6585s3a1: Subsidio familiar (esta variable no se utilizó en análisis posteriores).
10. P7040: Dummy que indica si tiene ocupación secundaria (1: sí, 0: no).
11. P7045: Horas trabajadas en la ocupación secundaria.
12. P6800: Total de horas trabajadas por semana (no utilizada en los análisis finales).
13. P6870: Número de empleados en la empresa donde trabaja (no utilizada en los análisis finales).
14. Oc: Dummy que indica si la persona tiene ocupación (1: sí, 0: no).

Matrices de confusiones



BIBLIOGRAFÍA

James, G., Witten, D., Hastie, T., Tibshirani, R. y Taylor J. (2023). An introduction to Statistical Learning with applications in python.

Fitzpatrick, C., Bull, P., y Dupriez, O. (2018). Machine learning for poverty prediction: A comparative assessment of classification algorithms.

McBride, L. y Nichols, A. (2018). Retooling poverty targeting using out-of-sample validation and machine learning. *The World Bank Economic Review*

Solís-Salazar, M. y Madrigal-Sanabria, J. (2022). A machine learning proposal to predict poverty. *Tecnología en Marcha. 35*(4).

Li, Q., Yu, S., Échevin, D. y Fan, M (2022). Is poverty predictable with machine learnin? A study of DHS data from Kyrgyztan. *Socio-Economic Planning Sciences (81).*

Hamzah, S., Min, P., Gan, Y., Ong, T. y Sayeed, S. (2022). Poverty prediction using machine learning approach. *Journal of Southwest Jiaotong University.* 57(1).

Hassan, A.A., Muse, A.H. y Chesneau, C. (2024). Machine learning study using 2020 SDHS data to determine poverty determinants in Somalia. Nature.

James, G., Witten, D., Hastie, T., Tibshirani, R. y Taylor J. (2023). An introduction to Statistical Learning with applications in python.