Problem Set 2

Big Data & Machine Learning

Profesor: Ignacio Sarmiento

Grupo 9
Santiago Melo - 202324572
Miguel Blanco - 202412541
Diana Lopera - 1130639521
Link repositorio: https://github.com/samelomo99/PS2_SM_MB_MB_DL

1. Introduction

La pobreza es un concepto complejo con una naturaleza multidimensional. Desde un enfoque monetario, un hogar se considera pobre si sus ingresos son insuficientes para cubrir sus necesidades básicas de alimentación, salud, educación y vivienda (Barrett y Carter, 2013). Por tanto, su erradicación constituye uno de los Objetivos de Desarrollo Sostenible, ya que afecta a una gran proporción de la población mundial, y representa una prioridad política en numerosos países. Actualmente, aproximadamente el 8,5 % de la población mundial vive en situación de pobreza extrema, mientras que el 44 % vive en pobreza, incluyendo un umbral específico para países de ingreso medio-alto (World Bank, 2024).

En el contexto colombiano, la pobreza presenta características particulares. Aunque los indicadores oficiales muestran una disminución en la pobreza monetaria —pasando del 42 % en 2008 al 27 % en 2018, según cifras del DANE—, persisten factores estructurales que dificultan avances sostenidos. La desigualdad de género, la inestabilidad del empleo y el acceso limitado a servicios esenciales aumentan el riesgo de caer o permanecer en situación de pobreza. Esto evidencia la complejidad del fenómeno más allá de la mera insuficiencia de ingresos (Hernández y Zuluaga, 2022).

Diversos estudios han identificado que los determinantes de la pobreza incluyen variables sociodemográficas y estructurales. Barrett y Carter (2013) sostienen que las limitaciones en el acceso al crédito y la incapacidad para acumular activos generan mecanismos auto-reforzantes —o "trampas de pobreza"— que impiden la mejora del bienestar a lo largo del tiempo. Lanjouw y Ravallion (1994) analizan cómo el tamaño del hogar, a menudo correlacionado con el número de dependientes, influye en las medidas de pobreza y bienestar per cápita. Banerjee y Duflo (2007), al explorar las vidas económicas de los pobres en diversos países, destacan la escasa posesión de activos y la falta de acceso a seguros y crédito formales como factores clave que exponen a los hogares pobres a una mayor inseguridad económica. Por su parte, Hernández y Zuluaga (2022), en su análisis de la vulnerabilidad a la pobreza multidimensional en Colombia, identifican que ser jefe de hogar femenino, estar en el autoempleo, experimentar la pérdida de empleo o la quiebra de un negocio familiar se correlacionan positivamente con la vulnerabilidad. En contraste, mayores años de escolarización y el acceso a crédito disminuyen dicha vulnerabilidad.

Frente a las limitaciones de los enfoques econométricos tradicionales, la incorporación de métodos de *machine learning* se presenta como una alternativa prometedora. Si bien variables como el número de integrantes del hogar, la edad, el sexo, el nivel educativo (especialmente del jefe de hogar, la madre y el esposo), la situación laboral, las condiciones de la vivienda, el acceso a servicios básicos (agua potable, saneamiento, electricidad, combustible para cocinar, número de habitaciones, gastos en servicios y alquiler) y la ubicación geográfica

son reconocidas como determinantes de la pobreza, el aprendizaje automático permite identificar de forma más precisa y contextualizada los predictores clave. Investigaciones recientes han demostrado que algoritmos como XGBoost, Random Forest y otros métodos de *boosting* pueden procesar grandes volúmenes de datos y detectar patrones complejos, facilitando una clasificación y predicción más acertada de la situación de pobreza (Sohnesen y Stender, 2017; Sani et al., 2018; Espana, 2022; Solís-Salazar y Madrigal-Sanabria, 2022; Hassan et al., 2024; Salvador, 2024).

En este sentido, identificar con precisión a las personas en situación de pobreza y emplear técnicas que reduzcan los costos de su medición es fundamental para diseñar intervenciones efectivas, asignar recursos de manera eficiente, comprender las causas estructurales del fenómeno, formular políticas públicas informadas y monitorear el progreso hacia los objetivos de desarrollo. Por tanto, este ejercicio empírico tiene como objetivo predecir la pobreza en Colombia utilizando diversas técnicas de *machine learning* (regresión lineal, logit, *elastic net*, árboles de decisión, Random Forest y *boosting*) para identificar la mejor estrategia de predicción de hogares pobres. La información proviene del DANE y de la misión "Empalme de las Series de Empleo, Pobreza y Desigualdad – MESEP", que comprende cuatro conjuntos de datos divididos en entrenamiento y prueba a nivel de hogar e individual.

Los resultados indican que el modelo de Random Forest fue el modelo predicción robusto y sensible para identificar hogares en situación de pobreza en Colombia. El cual fue ajustado a través de una validación cruzada de 5 pliegues y optimizado con hiperparámetros (mtry = 6, min.node.size = 50 y el criterio de impureza Gini), mostró un rendimiento destacado, con un AUC-ROC promedio superior a 0,84. Este desempeño, junto con una alta sensibilidad en la detección de casos positivos (beneficiarios), confirma la capacidad del modelo para capturar las complejas interacciones entre variables sociodemográficas y estructurales, superando en precisión a otros modelos tradicionales como la regresión logística y el elastic net.

En las siguientes secciones se describe brevemente el tratamiento dado al conjunto de datos y los principales hallazgos descriptivos. En la sección de modelos y resultados, se presentan los desempeños del modelo que obtuvo el mejor puntaje en la competencia de predicción en Kaggle, junto con un análisis comparativo con los demás algoritmos empleados. Finalmente, se expone una conclusión general del ejercicio.

2. Datos

2.1. Descripción de los datos

Este estudio se basó en los datos del Empalme de las Series de Empleo, Pobreza y Desigualdad (MESE). Para este caso, se generó una base de datos a nivel hogar con 164.960 observaciones para la base de entrenamiento, en la cual se condensaron las variables mencionadas a continuación. Junto a cada variable se describe el tratamiento dado.

Tal como se discutió en la introducción, las variables usadas para correr las distintas técnicas de clasificación de pobreza se fundamentan, en primer lugar, en la revisión de literatura sobre los determinantes estructurales de la pobreza. Esta evidencia muestra que la pobreza a nivel de hogar está explicada por dimensiones como la composición del hogar, características del jefe de hogar, capital humano, condiciones de vivienda, acceso a servicios y activos del hogar.

Variable	**Descripción**
Id	Representa el código identificador de cada una de las variables.
Dominio	Ciudad de residencia; se tomó la del jefe de familia.
Arrienda	Indica si el hogar paga renta por el lugar donde vive.
Pobre	Toma el valor de 1 si el hogar es pobre según la línea de pobreza de 2018, y 0 de lo contrario.
p5010	Número de habitaciones donde duermen los habitantes del hogar.
Nper	Número de personas en el hogar.
salud_jefe	"Sí"si el jefe de hogar está afiliado al régimen contributivo, "No"de lo contrario.
edad_jefe	Edad del jefe de hogar.
t_dependencia	Tasa de dependencia.
nmujeres	Número de mujeres en el hogar.
nmenores	Número de menores de edad en el hogar.
nocupados	Número de personas ocupadas en el hogar.
n_sin_educación	Número de personas sin nivel educativo.
H_Head_mujer	1 si el jefe de hogar es mujer, 0 de lo contrario.
clima_educ	Promedio de años de educación de los jefes de hogar.
ocup_jefe_informal	1 si el jefe de hogar tiene trabajo informal, 0 de lo contrario.
recibe_ayuda	1 si algún miembro del hogar recibe ayuda, 0 de lo contrario.

Cuadro 1: Descripción de variables del hogar

Durante el tratamiento de los datos, se notó que la base de datos de prueba no contenía registros para Bogotá, a diferencia de la base de entrenamiento. Por lo tanto, para algunos modelos se eliminaron las observaciones correspondientes a Bogotá.

2.2. Estadísticas descriptivas

En esta sección se presentan las estadísticas descriptivas de algunas de las variables más relevantes. En el Cuadro 1 se encuentra la descripción de las variables, mientras que en el Cuadro 2 se presenta la estadística de variables numéricas como el número de personas por hogar o la edad promedio del jefe de hogar.

Se puede evidenciar que variables como el clima educativo o la edad presentan una alta variabilidad, lo que refleja la diversidad de los hogares en la muestra.

Cuadro 2: Estadísticas descriptivas de variables numéricas

	Media	SD	Min	Max
P5010	1,99	0,90	1	15
Nper	3,29	1,77	1	28
edad_jefe	49,61	16,39	11	108
t_dependencia	0,35	0,33	0	1
nmujeres	1,74	1,18	0	14
nmenores	0,33	0,63	0	9
nocupados	3,29	1,77	1	28
n_sin_educacion	0,77	0,99	0	15
n_recibe_ayuda	0,18	0,45	0	5
clima_educ	6,11	2,85	0	15

Por otra parte, en las gráficas descriptivas de la Figura 1 se presenta la diferenciación entre hogares pobres y no pobres según algunas variables clave. Por ejemplo, se puede observar que la edad del jefe de hogar en los hogares pobres tiende a ser menor en comparación con el resto. El clima educativo es similar en ambos grupos, por lo que la variable de educación no parece tener un efecto diferenciador importante.

Adicionalmente, aunque tanto los hogares pobres como los no pobres tienden a arrendar viviendas, se evidencia que una mayor proporción de los hogares pobres arrienda comparado con los no pobres. Finalmente, casi la mitad de los hogares pobres tiene una mujer como jefe de hogar, mientras que esta proporción se reduce en casi un 10 % en los hogares no pobres.

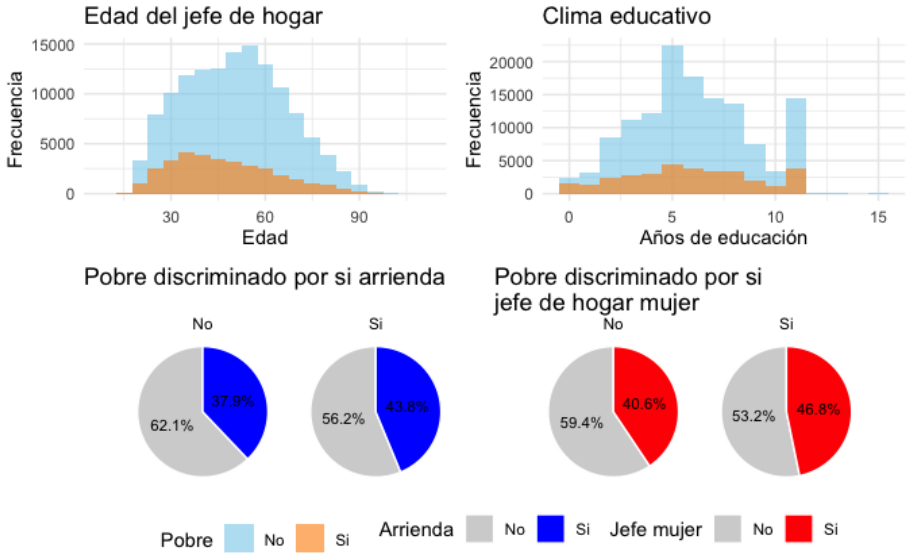

Figura 1: Gráfica descriptiva de algunas variables

3. Modelos y resultados

3.1. Selección y entrenamiento del modelo:

El modelo que alcanzó mejor resultado en la competencia pública de Kaggle fue un Random Forest entrenado con validación cruzada y métrica ROC.

El modelo fue entrenado a partir del paquete *caret*. Se empleó una validación cruzada de 5 *folders*, asegurando que la proporción de clases se mantuviera en cada subconjunto del entrenamiento. Esto permitió hacer estimaciones más robustas del desempeño y reducir el riesgo de overfitting.

Además, se utilizó el método *ranger*, el cual implementa el algoritmo de Random Forest de forma rápida, eficiente y escalable, y está diseñado para conjuntos de datos grandes y de alta dimensionalidad. En este sentido, esta metodología es mucho más rápida, ideal para *Big Data*, y eficiente computacionalmente.

Antes de correr el modelo con la totalidad de datos de la base *train*, se decidió dividir la misma en dos subconjuntos que denominamos *train_split* y *test_split*, para revisar la pertinencia del modelo, obtener la matriz de confusión y tener un valor aproximado del *Score F1*, antes de subirlo a la competencia pública. El resultado de la matriz de confusión fue la siguiente:

Cuadro 3: Matriz de confusión del modelo Random Forest (evaluación en test_split)

Predicción / Real	**0**	**1**
0	30.982	3.288
1	5.738	6.309

Con respecto a las especificaciones del modelo, se definió un control de entrenamiento *trainControl*, el cual contempla:

- **Cálculo de probabilidades de clase** (*classProbs = TRUE*), necesario para optimizar la métrica ROC.
- Combinación de *twoClassSummary* (para ROC, Sensitivity y Specificity) con *defaultSummary* (Accuracy y Kappa), lo que permitió monitorear múltiples indicadores de desempeño.

Por otra parte, la selección de hiperparámetros se realizó a través de una búsqueda en malla (*grid search*), utilizando 3 dimensiones. Las cuales se discutirán en la próxima sección.

- ***mtry:*** número de variables consideradas en cada división del árbol. Se probaron valores de 2, 4, 6 y 8.
- ***min.node.size:*** tamaño mínimo de nodos terminales, con valores entre 1 y 50.
- ***splitrule:*** se utilizó el criterio de impureza Gini.

Se entrenaron múltiples combinaciones sobre el conjunto *train* y la métrica optimizada fue el AUC-ROC, el cual es más adecuado en contextos de clases desbalanceadas.

Por último, se hizo uso de un umbral de clasificación de 0,3 en lugar del 0,5 (predeterminado). Esta decisión se realizó como respuesta al desbalance de clases observado en los datos, en la cual ser clasificado como *Pobre* es minoritaria. De esta manera, se buscaba mejorar la sensibilidad del modelo sin sacrificar la especificidad. En este sentido, ajustar el umbral funciona como una estrategia complementaria al rebalanceo y permite priorizar la detección de positivos (personas en situación de pobreza).

Si bien no se aplicaron estrategias explícitas de rebalanceo como submuestreo o sobremuestreo, en su lugar, el modelo se enfocó en:

- Utilizar métricas apropiadas como ROC en lugar de Accuracy.
- Ajustar el umbral de decisión para controlar los falsos negativos.
- Asegurar balance en los pliegues de CV.

3.2. Ajuste de hiperparámetros:

Para el ajuste de hiperparámetros se utilizó un *grid search* con *Cross Validation* de 5 pliegues, enfocada en 3 hiperparámetros principales:

- ***mtry:*** Este hiperparámetro controla cuántas variables predictoras se consideran para dividir cada nodo en los árboles del bosque. Valores más bajos tienden a generar árboles más diversos, lo que puede generar la generalización del modelo, pero también aumentar la varianza. Por el contrario, valores más altos pueden hacer que los árboles se parezcan más entre si, reduciendo la varianza, a costa de un mayor sesgo.

  Una regla empírica común para elegir un buen punto de partida en clasificación es probar valores cercanos a la raíz cuadrada del número total de predictores. En este caso, con 17 variables predictoras, la raíz cuadrada es aproximadamente 4,1. Por esta razón, los valores de *mtry* entre 2 y 8 abarcan un rango cercano a dicho valor de referencia.

  Esta decisión también se basó en el costo computacional del modelo. Un número reducido de variables por división reduce el tiempo de entrenamiento y permite construir árboles más rápidamente, lo que permitía ajustar hiperparámetros y explorar más combinaciones.

  El mejor rendimiento se observó con $mtry = 6$, que mostró un equilibrio adecuado entre diversidad y precisión.

- ***min.node.size:*** Este parámetro regula la profundidad efectiva de los árboles. Al aumentarlo, se fuerza al árbol a detener las divisiones antes de llegar a nodos muy pequeños, lo que limita el sobreajuste y mejora la generalización. Por el contrario, un valor muy bajo permite árboles más profundos y específicos, que tienden a memorizar el conjunto de entrenamiento.

  Se probraron los valores de: 1, 5, 10, 20, 35 y 50; permitiendo una amplia variedad de profundización. El valor óptimo fue $min.node.size = 50$, indicando que, dada la naturaleza del conjunto de datos, era preferible evitar árboles demasiado profundos. Aunque esta selección sacrificó algo de sensibilidad, contribuyó en mejorar la especificidad y la estabilidad general del modelo.

- ***splitrule:*** Se mantuvo constante el criterio de partición $splitrule = gini$ como medida de la impureza de los nodos. Esta es la regla más comúnmente usada en tareas de clasificación, y su estabilidad y velocidad computacional la hacen preferible frente a otras alternativas como la entropía.

Cuadro 4: Desempeño del mejor modelo de Random Forest con validación cruzada y conjunto de evaluación

Métrica	**Modelo Principal (test)**	**Partición (test_split)**
AUC-ROC	0,8426	0,8203
Sensitivity	0,9603	0,6574
Specificity	0,3575	0,8437
Accuracy	0,8354	0,8051
Kappa	0,3880	0,4580

La mejor combinación fue *mtry* = 6, *min.node.size* = 50 y *splitrule* = *gini*, alcanzando un AUC-ROC promedio de 0,842 en la validación cruzada. Las métricas complementarias mostraron una sensibilidad de 0,96 y una especificidad de 0,35; lo que refleja un modelo con alta capacidad para identificar correctamente los casos positivos (beneficiarios), aunque con una tasa relativamente alta de falsos positivos.

3.3. Análisis comparativo:

Para el ejercicio de predicción de pobreza se utilizaron diferentes técnicas, tales como OLS, Logit, Elastic Net, CART, Random forest y Boosting. Para todos los casos se utilizaron los datos de train, asignando el 70 % para el entrenamiento y el 30 % para la validación inicial. Asimismo, se aplicó una estrategia de validación cruzada para evaluar de forma robusta el rendimiento y la capacidad de generalización de cada modelo desarrollado. Este enfoque permitió comparar sistemáticamente diversos modelos, identificando el que presentaba mejores métricas en términos de precisión y error en datos no vistos. El objetivo principal de este proceso fue optimizar la selección del modelo que se subiría a la competencia de Kaggle. De esta manera, se evitó el desperdicio de envíos innecesarios. Los resultados de la matriz de confusión de cada modelo y los resultados de los 5 modelos de la competencia se presentan en el (Cuadro **??**)

Cuadro 5: Comparación de métricas: Logit vs. CART vs. OLS vs. XGBoost vs. Elastic Net

Métrica	**Logit**	**CART**	**OLS**	**XGBoost**	**Elastic Net**
Accuracy	0.8257	0.8118	0.8246	0.8282	0.3343
Precision	0.6452	0.5419	0.6370	0.6635	0.6616
Recall	0.2876	0.3879	0.2897	0.3534	0.3282
Specificity	0.9604	0.9179	0.9587	0.9529	0.3577
NPV	0.8434	0.8570	0.8436	0.8487	0.1222
FPR	0.0396	0.0821	0.0413	0.0471	0.6423
FNR	0.7124	0.6121	0.7103	0.6466	0.6718
F1	0.3978	0.4521	0.3987	0.4618	0.4403
F1 Kaggle	0.3959	0.4552	0.3742	0.4755	0.4412
Balanced_Accuracy	0.6240	0.6529	0.6242	0.6531	0.3430

La regresión logística es un referente por su interpretabilidad y bajo costo computacional, no obstante, tiene limitaciones frente a relaciones no lineales, transformaciones o relaciones complejas. Lo cual se evidencia cuando se implementaron técnicas más robustas, como la Elastic Net, que combina regularización para manejar, por ejemplo, correlaciones entre los predictores. Ahora bien, cuando introducen las técnicas basadas en árboles, como CART, se evidencia una mejora en el escore F1, quizá ocasionada por la flexibilidad que aporta al modelo al manejar variables categóricas desbalanceadas y capturar interacciones complejas. Por otro lado, cuando se realiza CART, este es optimizado con el tuning del parámetro cp, , especialmente al aplicar la poda (podar ramas que no aporten significativamente), lo que ayuda a reducir el sobreajuste. Sin embargo, CART presenta un menor desempeño que las técnicas de Random Forest y XGBoost. En principio, se sabe que CART presenta una gran desventaja: su alta sensibilidad a los datos de entrenamiento.

3.4. Importancia de las variables:

En el modelo de mejor desempeño (Random Forest), la importancia relativa de las variables fue evaluada utilizando el criterio de *impureza de Gini*, una métrica que mide cuánto contribuye cada variable a mejorar la pureza de los nodos en los árboles del modelo. Las variables con mayor capacidad para reducir la impureza fueron consideradas más relevantes para la predicción de la pobreza en los hogares.

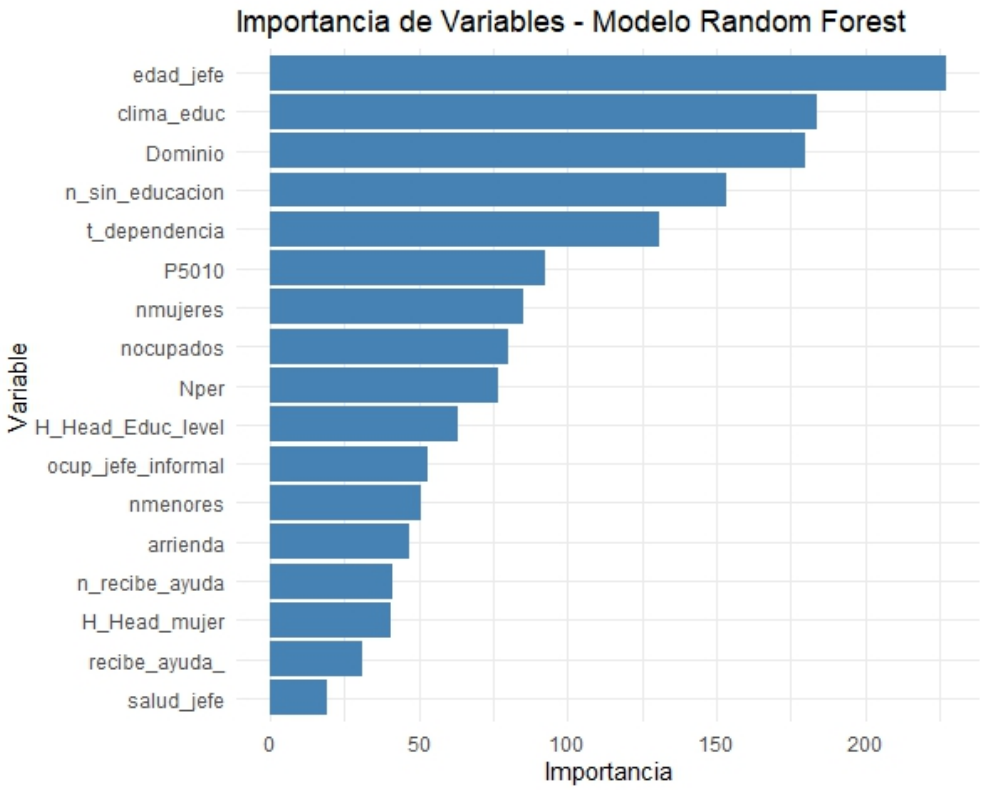

Figura 2: Importancia de las variables

Los resultados muestran que variables como la **edad del jefe de hogar** (`edad_jefe`), el **promedio de años de educación del hogar** (`clima_educ`), el **número de personas ocupadas** (`nocupados`) y la **tasa de dependencia** (`t_dependencia`) se destacan como las más importantes. Estas variables tienen una clara justificación empírica: por ejemplo, una mayor edad del jefe de hogar puede estar asociada con mayor estabilidad económica; más años de educación reflejan un mayor capital humano; y una mayor proporción de personas ocupadas en el hogar contribuye directamente a los ingresos. De esta manera, estas características mejoran sustancialmente la capacidad del modelo para distinguir entre hogares pobres y no pobres.

Asimismo, variables como el **número de menores** (`nmenores`) y la **afiliación del jefe de hogar al régimen contributivo de salud** (`salud_jefe`) también presentan un grado relevante de importancia, lo que sugiere que las condiciones demográficas y el acceso a servicios de salud son indicativos importantes del bienestar del hogar.

Por otro lado, variables como `H_Head_mujer` (si el jefe de hogar es mujer) o `recibe_ayuda` (si el hogar recibe ayudas) mostraron menor contribución relativa en el modelo. Aunque estos factores pueden influir en la pobreza, su capacidad de discriminación dentro del modelo es menor en comparación con las variables estructurales de educación, empleo y composición del hogar.

En conjunto, el análisis de importancia de variables refuerza la idea de que los factores educativos, demográficos y laborales son determinantes clave para entender la pobreza. Este resultado, además de estar respaldado por la teoría económica, está empíricamente sustentado por las puntuaciones de importancia derivadas del modelo, lo que valida su relevancia al momento de generar las predicciones sobre si un hogar es pobre o no.

4. Conclusiones

Este ejercicio permitió explorar de forma práctica cómo las técnicas de machine learning pueden aplicarse al análisis de la pobreza en Colombia. A partir de una base de datos amplia y representativa, se probaron distintos modelos para predecir si un hogar es pobre, teniendo en cuenta variables relacionadas con educación, composición del hogar, condiciones laborales y características del jefe de hogar.

Los resultados muestran que el modelo de *Random Forest* fue el que tuvo el mejor desempeño, especialmente en métricas como AUC-ROC y sensibilidad. Esto significa que el modelo no solo fue bueno prediciendo correctamente los casos, sino que también tuvo una buena capacidad para detectar hogares que efectivamente son pobres, lo cual es clave en contextos de política pública donde es más costoso no identificar correctamente a los beneficiarios.

El modelo fue ajustado cuidadosamente, utilizando validación cruzada y una búsqueda de hiperparámetros que permitiera equilibrar la especificidad y la sensibilidad. Aunque no se aplicaron técnicas explícitas de rebalanceo de clases, el uso de métricas adecuadas y la modificación del umbral de clasificación ayudaron a mejorar el rendimiento general del modelo en un escenario con clases desbalanceadas.

En comparación, modelos tradicionales como la regresión logística y Elastic Net tuvieron un buen comportamiento en algunas métricas, pero no alcanzaron los niveles de precisión ni la capacidad de generalización del *Random Forest*. Esto sugiere que, para problemas como este, donde las relaciones entre las variables no son lineales ni fácilmente modelables, las técnicas basadas en árboles pueden capturar mejor los patrones de los datos.

Además, el análisis de importancia de variables reveló que los factores que más aportan a la predicción son coherentes con la literatura sobre pobreza: edad del jefe de hogar, nivel educativo, número de personas ocupadas y tasa de dependencia. Esto le da mayor validez al modelo y sugiere que está aprendiendo relaciones relevantes en lugar de ajustarse aleatoriamente a los datos.

Desde una perspectiva de política pública, este tipo de herramientas puede ser muy útil para focalizar programas sociales, reducir errores de inclusión y exclusión, y hacer un uso más eficiente de los recursos. En lugar de depender únicamente de encuestas costosas o procesos de postulación, se podría complementar con predicciones automáticas que permitan identificar hogares vulnerables de forma más ágil.

En resumen, este ejercicio muestra que aplicar modelos de clasificación como *Random Forest* al problema de la pobreza no solo es factible, sino que puede aportar información valiosa para diseñar intervenciones más efectivas y bien dirigidas.

5. Apéndice

Cuadro 6: Métricas de desempeño para modelo OLS

Métrica	**Valor**
Accuracy	0.8246484
Kappa	0.3116546
Sensitivity	0.28973
Specificity	0.95867
Pos Pred Value	0.63697
Neg Pred Value	0.84356
Prevalence	0.20019
Detection Rate	0.05800
Detection Prevalence	0.09106
Balanced Accuracy	0.62420

Cuadro 7: Resultados de métricas para modelos para 7 especificaciones de logit (100 % de train)

model	**logit_m1**	**logit_m2**	**logit_m3**	**logit_m4**	**logit_m5**	**logit_m6**	**logit_m7**
AUC	0.889	0.907	0.937	0.938	0.937	0.927	0.917
Precision	0.814	0.814	0.843	0.842	0.845	0.830	0.835
Recall	0.981	0.977	0.961	0.959	0.961	0.968	0.961
F	0.890	0.888	0.898	0.897	0.899	0.894	0.893
AUCSD	0.002	0.001	0.001	0.001	0.001	0.001	0.001
PrecisionSD	0.001	0.001	0.001	0.001	0.001	0.000	0.001
RecallSD	0.001	0.002	0.002	0.001	0.001	0.001	0.001
FSD	0.001	0.001	0.001	0.001	0.001	0.001	0.001

Cuadro 8: Mejor especificación logit (con train 70 % datos) `logit_m4_prueba`

métrica	**logit_m4_prueba**
AUC	0.937046559
Precision	0.841030320
Recall	0.959699435
F	0.896451858
AUCSD	0.000933609
PrecisionSD	0.001742084
RecallSD	0.002351722
FSD	0.000950964

Cuadro 9: Resultados CART (70 %train) `cv_tree_1`

cp	ROC	Sens	Spec	Accuracy	Kappa	ROCSD	SensSD	SpecSD	AccuracySD	KappaSD	model
0.000	0.798328993	0.919258114	0.373578278	0.810016168	0.330413535	0.003823201	0.002475012	0.013862986	0.002485136	0.012082166	cv_tree_1
0.001	0.715817274	0.963564892	0.264005374	0.823517174	0.292213659	0.004754549	0.002312046	0.010590347	0.002580934	0.012113271	cv_tree_1
0.002	0.710979852	0.966715637	0.233939749	0.820018535	0.262663229	0.003583753	0.006905720	0.027514359	0.001449529	0.020644116	cv_tree_1
0.003	0.708350131	0.967018893	0.222520156	0.817974780	0.249344452	0.003546910	0.008423008	0.035420343	0.002092767	0.028199181	cv_tree_1
0.004	0.707069936	0.965048297	0.221959201	0.816286058	0.245340157	0.004668111	0.004613613	0.029613020	0.002704721	0.027287872	cv_tree_1
0.005	0.706892178	0.966834882	0.211359621	0.815593241	0.236155315	0.004457296	0.003125101	0.017203908	0.002274069	0.016854842	cv_tree_1
0.006	0.646460432	0.976796251	0.144693038	0.810215312	0.167338967	0.055217950	0.011603701	0.062656537	0.003578924	0.061403128	cv_tree_1
0.007	0.607329939	0.984321561	0.103386810	0.807963742	0.127312969	0.004236193	0.002520200	0.019525249	0.002237298	0.023024350	cv_tree_1
0.008	0.607329939	0.984321561	0.103386810	0.807963742	0.127312969	0.004236193	0.002520200	0.019525249	0.002237298	0.023024350	cv_tree_1
0.009	0.607224295	0.982859889	0.106112313	0.807340230	0.128374338	0.004191041	0.004962642	0.024156056	0.001396260	0.024792161	cv_tree_1
0.010	0.607224295	0.982859889	0.106112313	0.807340230	0.128374338	0.004191041	0.004962642	0.024156056	0.001396260	0.024792161	cv_tree_1
0.011	0.607154610	0.981614694	0.108966984	0.806915896	0.130028302	0.004135301	0.006137848	0.027042096	0.001062248	0.026507837	cv_tree_1
0.012	0.607076127	0.978214770	0.118138518	0.806032550	0.136807882	0.004307542	0.007760402	0.024343968	0.002909350	0.022057774	cv_tree_1
0.013	0.519901526	0.993752369	0.026086957	0.800031182	0.027523161	0.044501164	0.013970129	0.058332208	0.000510644	0.061543659	cv_tree_1
0.014	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.015	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.016	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.017	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.018	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.019	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.020	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.021	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.022	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.023	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.024	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.025	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.026	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.027	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.028	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.029	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1
0.030	0.500000000	1.000000000	0.000000000	0.799806015	0.000000000	0.000000000	0.000000000	0.000000000	0.000020853	0.000000000	cv_tree_1

Cuadro 10: Métricas de desempeño para modelo XGboost (con matriz de confusión)

Métrica	**Valor**
Accuracy	0.8282
IC 95 % Accuracy	(0.8247, 0.8316)
No Information Rate	0.7920
P-Valor [Acc ¿ NIR]	¡ 2.2e-16
Kappa	0.37
McNemar's Test P-Valor	¡ 2.2e-16
Sensitivity	0.35336
Specificity	0.95292
Pos Pred Value	0.66348
Neg Pred Value	0.84871
Prevalence	0.20804
Detection Rate	0.07351
Detection Prevalence	0.11080
Balanced Accuracy	0.65314

Referencias

Banerjee, A. V., & Duflo, E. (2007). The economic lives of the poor. *Journal of Economic Perspectives*, *21*(1), 141-167. https://doi.org/10.1257/jep.21.1.141

Barrett, C. B., & Carter, M. R. (2013). The economics of poverty traps and persistent poverty: Empirical and policy implications. *The Journal of Development Studies*, *49*(7), 976-990. https://doi.org/10.1080/00220388.2013.785527

Espana, C. P. (2022). Clasificación de la pobreza en Bolivia, utilizando Random Forest y XGBoost. *Cuadernos de Investigación Económica Boliviana*, *5*(1), 73-98. https://www.researchgate.net/profile/Cristian-Paucara/publication/378521356_Clasificacion_de_la_pobreza_en_Bolivia_utilizando_Random_Forest_y_XGBoost/links/65de50dee7670d36abe2f35d/Clasificacion-de-la-pobreza-en-Bolivia-utilizando-Random-Forest-y-XGBoost.pdf

Hassan, A. A., Muse, A. H., & Chesneau, C. (2024). Machine learning study using 2020 SDHS data to determine poverty determinants in Somalia. *Scientific Reports*, *14*(1), 5956. https://www.nature.com/articles/s41598-024-56466-8.pdf

Hernández, J. E., & Zuluaga, B. (2022). Vulnerability to Multidimensional Poverty: An Application to Colombian Households. *Social Indicators Research*, *164*, 345-371. https://doi.org/10.1007/s11205-022-02961-2

Lanjouw, P. F., & Ravallion, M. (1994). *Poverty and household size* (inf. téc. N.º WPS 1332). World Bank Group. http://documents.worldbank.org/curated/en/641891468741345861

Salvador, E. L. (2024). Use of Boosting Algorithms in Household-Level Poverty Measurement: A Machine Learning Approach to Predict and Classify Household Wealth Quintiles in the Philippines. *arXiv preprint arXiv:2407.13061*. https://arxiv.org/pdf/2407.13061

Sani, N. S., Rahman, M. A., Bakar, A. A., Sahran, S., & Sarim, H. M. (2018). Machine learning approach for bottom 40 percent households (B40) poverty classification. *International Journal of Advanced Science and Engineering Information Technology*, *8*(4-2), 1698. https://core.ac.uk/download/pdf/325990481.pdf

Sohnesen, T. P., & Stender, N. (2017). Is random forest a superior methodology for predicting poverty? An empirical assessment. *Poverty & Public Policy*, *9*(1), 118-133. https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=2750208

Solís-Salazar, M., & Madrigal-Sanabria, J. (2022). A machine learning proposal to predict poverty. *Revista Tecnología en Marcha*, *35*(4), 84-94. https://www.scielo.sa.cr/scielo.php?pid=S0379-39822022000400084&script=sci_arttext&tlng=en

World Bank. (2024). *Poverty, Prosperity, and Planet Report 2024: Pathways Out of the Polycrisis*. https://doi.org/10.1596/978-1-4648-2123-3