**UNIVERSIDAD DE BUENOS AIRES - 2025**

**Taller de Programación**

**Trabajo Práctico N° 4**

**Clasificando pobres en la EPH: métodos de regularización y CART**

*A. Modelo de Regresion Logistica con Regularización: Ridge y LASSO*

1. *Visualización:* Grafiquen en dos paneles distintos los coeficientes de la penalidad de LASSO y Ridge para la grilla del parámetros de penalidad con e interpreten la regularización con cada penalidad. (*Hint*: en la función Logistic Regression la opción C es la inversa de la fuerza de penalidad, en lugar de alpha. Según el tamaño de su matriz, se recomienda probar con distintas opciones de solver para el uso eficiente computacional).

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

En esta sección analizamos la evolución de los coeficientes de la regresión logística penalizada frente a diferentes niveles de regularización. Para ello estimamos modelos con penalización L1 (LASSO) y L2 (Ridge) utilizando una grilla de valores del parámetro de penalidad de la forma λ = 10ⁿ, con n ∈ {−5, …, 5}. Dado que la implementación de LogisticRegression en sklearn utiliza C = 1/λ, valores pequeños de λ implican una penalización débil (C grande), mientras que valores elevados de λ corresponden a una penalización fuerte (C pequeño).

Los resultados muestran que, para valores muy pequeños de λ (10⁻⁵ a 10⁻²), los coeficientes prácticamente no difieren de los estimados por el modelo sin regularización. Esto es esperable, dado que la penalización es casi nula y el modelo reproduce el comportamiento del logit estándar. A partir de valores intermedios de penalización (λ ≈ 10⁰ = 1), LASSO comienza a ejercer un efecto notable: algunos coeficientes se reducen rápidamente y varios de ellos se vuelven exactamente cero. Este comportamiento refleja la propiedad distintiva de la penalización L1: LASSO no solo contrae los coeficientes sino que también realiza selección automática de variables.

Para valores grandes de penalidad (λ ≥ 10²), todos los coeficientes colapsan a cero, lo que indica que la penalización domina completamente la función de pérdida y el modelo se torna no informativo. Este patrón —coeficientes estables para λ pequeño, eliminación progresiva en λ moderado y colapso total en λ grande— es característico de LASSO.

El panel correspondiente a Ridge exhibe un comportamiento marcadamente diferente. Al igual que en LASSO, cuando λ es muy pequeño los coeficientes prácticamente no cambian. Sin embargo, al aumentar la penalidad, la contracción de los coeficientes es suave y continua, sin quiebres abruptos y, fundamentalmente, sin que ningún coeficiente se vuelva exactamente cero.

Incluso para λ extremadamente elevados (10³ a 10⁵), todos los coeficientes se acercan asintóticamente a cero, pero ninguno es eliminado. Esto confirma la principal propiedad de Ridge: la penalización L2 regulariza el modelo contrayendo los coeficientes, pero nunca realiza selección de variables. Por lo tanto, Ridge tiende a mejorar la estabilidad del modelo en presencia de colinealidad sin reducir la dimensionalidad efectiva.

La comparación entre ambos gráficos resalta las diferencias fundamentales entre las dos penalizaciones. En efecto, estas diferencias son visualmente claras: mientras que LASSO comienza a eliminar variables desde λ ≈ 10¹, Ridge solo reduce la magnitud de los coeficientes pero nunca los descarta.

El ranking de importancia muestra que variables como nivel\_ed\_6.0, ch03\_7.0, estado\_2.0, ch03\_5.0 y miembros\_hogar aparecen sistemáticamente con los coeficientes de mayor magnitud en los niveles de baja penalidad. Esto sugiere que son las variables más influyentes para predecir la probabilidad de que la persona haya sido pobre en 2025.

1. *Penalidad óptima por Cross-validation y visualización*: Usando Logistic Regression CV dividiendo la base 5 partes (5-fold), elijan la penalidad óptima en la grilla mencionada arriba. ¿Qué seleccionó en cada caso? Hagan un grafico de linea con puntos conectados mostrando la distribución del error de clasificaición promedio entre folds para cada . *Opcional*: para la regularización LASSO, generen otro plot de puntos conectados, para la proporción de variables ignoradas por el modelo en función de , es decir la proporción de variables para las cuales el coeficiente asociado es cero.

Con el objetivo de elegir el nivel óptimo de regularización para los modelos logísticos penalizados, se aplicó validación cruzada de 5 folds sobre la misma grilla utilizada previamente (λ = 10ⁿ, con n ∈ {−5, …, 5}). Para cada valor de λ, se estimó el error promedio de clasificación (1 – accuracy) y se seleccionó el parámetro que minimiza dicho error.

El gráfico de error de clasificación para LASSO muestra una trayectoria en forma de “U” extendida. Para λ muy pequeños (10⁻⁵ a 10⁻²), el error permanece prácticamente constante y cercano al mínimo. A partir de λ ≥ 10⁰, el error aumenta de manera marcada, coincidiendo con el inicio del proceso de eliminación de variables. Para λ grandes (≥ 10²), el error se dispara, reflejando que el modelo queda excesivamente penalizado y pierde capacidad predictiva.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

El mínimo error de clasificación para LASSO se obtiene en λ = 10⁻3 (C=1000) con un error promedio de 0,2735. Cabe destacar que, en este valor óptimo de penalidad, LASSO no elimina ninguna variable (0 de 36). Esto indica que, en este caso, la regularización L1 mejora marginalmente el ajuste sin inducir un modelo más esparso.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

El comportamiento del error de clasificación para Ridge es similar en forma, aunque presenta una trayectoria más suave. Para λ muy pequeños (10⁻⁵ a 10⁻³), el error permanece estable. El mínimo se alcanza en λ = 10⁻². A partir de λ ≥ 10⁰, el error aumenta gradualmente, reflejando un exceso de contracción de los coeficientes. El mínimo error para Ridge se obtiene en λ = 10⁻2 (C=1000) con un error promedio de 0,2736. El modelo nunca elimina variables, consistente con las propiedades de la penalización L2.

Los resultados de CV permiten concluir que los dos modelos logran prácticamente el mismo error mínimo (LASSO = 0,2735, Ridge = 0,2736). En efecto, la diferencia es muy pequeña, por lo que la elección entre L1 y L2 no afecta de manera sustantiva la performance predictiva sobre este conjunto de datos.

1. *Estimación con y comparación de coeficientes*: Estimen una **Regresión Logística** usando *X\_train* de respondieron\_2025 sin penalidad, con penalidad L1 y L2 siendo la penalidad óptima elegida en el ítem anterior. Exporten una tabla con las siguientes columnas: (i) los coeficientes estimados para cada variable sin penalidad, (ii) coeficientes de penalidad L1, y (ii) coeficientes de penalidad L2. Interpreten los resultados de la tabla y conteste: ¿cómo son los coeficientes de la regularización con respecto a logit sin penalidad? ¿La penalidad L2 elimina variables de las que incluyeron en su matriz *X\_train*?

En esta sección estimamos tres modelos logísticos utilizando el conjunto de entrenamiento:

* Logit sin penalización.
* Logit con penalización L1 (LASSO) utilizando la penalidad óptima seleccionada por CV (λ = 10⁻³).
* Logit con penalización L2 (Ridge) utilizando su penalidad óptima (λ = 10⁻²).

Observando los resultados, comparamos los coeficientes estimados, la magnitud de los cambios introducidos por cada tipo de regularización y la eventual eliminación de variables.

Para empezar, observamos que ninguna variable fue eliminada por LASSO en la penalidad seleccionada por CV. Esto significa que, aunque LASSO tiene capacidad para fijar coeficientes iguales a cero, la validación cruzada determinó que la penalidad que minimiza el error de clasificación es muy débil como para inducirnos a la elección de un modelo más esparso. Por lo tanto, los modelos finalizados con L1 y L2 utilizan exactamente las mismas 36 variables que el logit sin penalización.

Los siguientes gráficos permiten apreciar visualmente las diferencias entre los tres modelos.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Como puede observarse, LASSO (L1) genera cambios algo mayores que Ridge, pero en general moderados. El cambio promedio absoluto con respecto al modelo sin penalización es igual a 0,091. No se observan diferencias en los signos ni en la importancia relativa de las variables. Algunas variables muestran contracciones relativamente fuertes (p. ej., cat\_inac\_2.0, estado\_3.0, ch03\_10.0), aunque sin llegar a cero.

Por su parte, Ridge (L2) aplica un grado de contracción más suave. En este caso, el cambio promedio absoluto es de 0,0367. Esto es consistente con el efecto típico de la penalización L2, que reduce la varianza del modelo sin inducir sparsity. En todos los casos, los coeficientes Ridge mantienen su signo y orden aproximado respecto al logit sin penalizar.

Los coeficientes más relevantes del modelo sin penalizar (como nivel\_ed\_6.0, estado\_2.0, ch03\_7.0, ch03\_3.0, etc.) se mantienen como los predictores de mayor magnitud tanto en LASSO como en Ridge. Por lo tanto, las dos regularizaciones preservan la estructura central del modelo. En otras palabras, los predictores dominantes siguen siendo los mismos, y la regularización solo introduce pequeños ajustes en las magnitudes.

En síntesis, la regularización mejora ligeramente la robustez y estabilidad del modelo, pero no altera su estructura fundamental ni selecciona variables adicionales, dado que los niveles de penalidad que optimizan la capacidad predictiva son relativamente bajos.

*B. Árboles de Decisión*

1. Estimen un árbol de decisión podado (CART) eligiendo el hiperparámetro de costo de complejidad del árbol (ccp\_aplha) con 10-fold Cross-Validation. En un gráfico de línea, muestre el error de clasificación en función de la grilla de valores que uso para ccp\_aplha. Comenten los resultados.

Para determinar el nivel óptimo de poda en el árbol de decisión, en esta sección utilizamos el hiperparámetro de costo de complejidad (ccp\_alpha), que controla la penalización por complejidad del árbol. Un valor mayor de ccp\_alpha produce árboles más simples (mayor poda), mientras que valores cercanos a cero permiten estructuras más profundas. Estimaomos 546 árboles diferentes, cada uno asociado a un valor de ccp\_alpha único extraído del proceso de coste-complejidad del árbol sin podar. Para cada uno se calculó el error promedio de clasificación mediante validación cruzada de 10 folds.

Los resultados son los siguientes:

* ccp\_alpha óptimo encontrado: 0,000575
* Error de clasificación mínimo: 0,2866
* Rango de errores en la grilla: [0,2866, 0,3499]

El gráfico CV vs *ccp\_alpha* muestra un patrón típico en árboles de decisión. Para ccp\_alpha ≈ 0 (sin poda), el error de CV es relativamente alto (≈ 0,35), lo que evidencia sobreajuste (el árbol sin restricciones se adapta excesivamente a las particularidades del entrenamiento). A medida que aumenta ccp\_alpha, el error cae muy rápidamente hasta alcanzar un mínimo alrededor de 0,000575, donde el árbol logra un mejor balance entre complejidad y capacidad predictiva. Por último, para valores mayores de ccp\_alpha (>0.001) el error comienza a aumentar nuevamente, lo que indica que la poda excesiva simplifica demasiado el árbol, produciendo un ajuste subóptimo.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

1. *Visualización del árbol podado por cross-validation:* En un panel A, muestren el gráfico del árbol que obtuvieron e interpreten. En un panel B,muestren en un gráfico la importancia de cada uno de los predictores. ¿Podría decirse que los coeficientes de las variables menos importantes son los que LASSO “achicó” a cero?

Diagrama

El contenido generado por IA puede ser incorrecto.

Con el ccp\_alpha óptimo se estimó el árbol final. Las clases predichas son pobre y no pobre.

Las características del árbol óptimo son las siguientes.

Profundidad: 8

Número de nodos: 65

Número de hojas: 33

Accuracy en entrenamiento: 0,7287

El árbol podado muestra una estructura clara y coherente. La variable raíz es miembros\_hogar. Es decir, la primera división, la más informativa para separar las clases, es el número de miembros en el hogar. La segunda variable más utilizada en niveles superiores es máximo nivel educativo del hogar (nivel\_ed\_6.0). Luego aparecen divisiones asociadas a al tipo de inactividad (cat\_inac), condición de actividad (estado), edad y estado civil (ch06, ch07) y categoría de ocupación (cat\_ocup). La lógica del árbol es consistente con los patrones de pobreza: tamaño del hogar y nivel educativo generan divisiones gruesas entre pobres y no pobres, mientras que las categorías laborales refinan los nodos inferiores.

El siguiente gráfico muestra la importancia relativa de las variables por reducción total de Gini:

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Como puede observarse, miembros\_hogar domina fuertemente el árbol (≈ 59% de la importancia total). Esto es coherente con la división raíz y refleja un predictor altamente discriminante para la condición de pobreza. El nivel educativo y la condición de inactividad también tienen pesos significativos, lo cual coincide con los determinantes socioeconómicos esperados. Variables como ch04 y cat\_ocup tienen un peso muy bajo (<1%), lo que sugiere que aportan poca capacidad de discriminación en un modelo no lineal basado en particiones.

Obsérvese que, aunque LASSO no eliminó variables en la penalidad óptima, cuando la penalidad era más fuerte eliminaba primero las mismas variables que el árbol asigna menor importancia. Esto indica coherencia entre ambos métodos y refuerza la interpretación de que ciertas variables aportan mucha menos información que otras.

*C. Comparación entre métodos*

1. Computen la matriz de confusión (p>0,5), la curva ROC y las dos métricas que utilizó en el TP3. Comparen el desempeño de Regresiones Logísticas sin penalidad, KNN con K-CV (ambos en TP3), logit con penalidad LASSO y Ridge, y el árbol de decisión podado. Comenten los resultados. ¿Qué pueden decir del trade-off entre comunicación de los resultados y performance predictiva? En este caso, ¿hay una ventaja por utilizar un método no lineal? Incorporen dentro de la justificación (1 - *accuracy*) para argumentar.

En esta sección evaluamos el desempeño predictivo de cinco modelos:

* Regresión logística sin penalización
* Logit con LASSO (λ óptimo)
* Logit con Ridge (λ óptimo)
* Árbol de decisión podado (ccp\_alpha óptimo)
* KNN con K óptimo por CV

Para todos ellos se calcularon las métricas principales: matriz de confusión, accuracy, recall de la clase “pobre”, precisión, F1-score y AUC-ROC.

Gráfico, Gráfico de rectángulos

El contenido generado por IA puede ser incorrecto.

Comparando los errores en las matrices de confusion, se observa que los modelos logit (sin penalidad, LASSO, Ridge) son muy similares entre sí:

* Falsos negativos ~444–445
* Falsos positivos ~360–361
* Verdaderos positivos ~851–853
* Verdaderos negativos ~1145–1146

Esto significa que lost tres modelos logit equilibran razonablemente bien los dos tipos de error.

Por su parte, el árbol podado aumenta tanto los falsos negativos (470) como los falsos positivos (369). Es decir, es un modelo menos preciso y menos sensible (con menor AUC y accuracy).

Por último, el KNN empeora con respecto al logit la cuenta de falsos negativos (473) pero mejora la de verdaderos negativos (1153). Esto significa que KNN clasifica mejor a los no pobres, pero pierde sensibilidad en la clase pobre.

La máxima accuracy corresponde al modelo logit sin penalidad (0,7131), seguido muy de cerca por LASSO y Ridge. El árbol podado es el que menos acierta (0,7006), aunque con diferencia marginal. Las diferencias absolutas entre modelos son pequeñas. Es decir, todos los modelos tienen una performance muy similar, lo que indica un problema relativamente lineal y con estructura moderada en las variables.

| **Modelo** | **Accuracy** | **1 − Accuracy** |
| --- | --- | --- |
| Logit sin penalidad | 0,7131 | 0,2869 |
| LASSO | 0,7127 | 0,2873 |
| Ridge | 0,7127 | 0,2873 |
| KNN | 0,7052 | 0,2948 |
| Árbol podado | 0,7006 | 0,2994 |

El mejor modelo para identificar pobres (maximizar verdaderos positivos) es el Logit sin penalidad (65.8%).

| **Modelo** | **Recall pobres** |
| --- | --- |
| Logit sin penalidad | 0,6582 |
| LASSO | 0,6566 |
| Ridge | 0,6566 |
| Árbol podado | 0,6373 |
| KNN | 0,6350 |

Gráfico

El contenido generado por IA puede ser incorrecto.

Todos los modelos tienen AUC cercana a 0.78–0.79, con diferencias muy pequeñas. El mejor AUC lo logran LASSO/Ridge, pero la diferencia con logit es irrelevante desde el punto de vista predictivo.

Como conclusion general, podemos decir que los métodos no lineales no mejoran la performance predictiva. Logit, LASSO y Ridge obtienen las mejores métricas en conjunto (con diferencias mínimas entre sí); mientras que el árbol podado y KNN, pese a captar relaciones no lineales, no superan al logit. Esto significa que la estructura del problema es mayormente lineal y que los modelos más flexibles no aportan ganancias significativas. Pero además, los modelos lineales (Logit, LASSO, Ridge) se caracterizan por una mayor interpretabilidad y comunicabilidad (coeficientes con signo y magnitud claros, fáciles de justificar ante decisores públicos). Por lo tanto, no quedan dudas de la superioridad de los modelos lineales en el contexto del presente problema.

1. Recuerde que el Ministerio de Capital Humano está interesado en identificar a grupos vulnerables para dirigir los recursos de un programa de alimentos. ¿Cambió su respuesta con respecto a cuál es el “mejor” modelo para asignar recursos escasos a los más necesitados?

Dado que el Ministerio de Capital Humano busca identificar correctamente a personas pobres, priorizamos como métrica de evaluación el recall de la clase pobre. La idea es minimizar los falsos negativos, para evitar dejar sin ayuda a los grupos vulnerables. Por lo tanto, en vista de los resultados reportados, el mejor modelo para asignar recursos escasos a los más necesitados es la regresión logística sin penalización (lo que coincide con nuestra decisión previa).