

Trabajo Práctico N° 4

Parte 1: Análisis de la base de hogares y tipo de ocupación

A partir de una visión general de la EPH hogar se observaron variables que podrían describir la desocupación y servirían para mejorar las predicciones. Estas pertenecen al apartado “estrategias del hogar” cuyo código empieza con la letra “V” y hacen referencia a si las personas del hogar vivieron de determinados ingresos en el último mes (por ejemplo, por pedir préstamos a familiares). Estas variables podrían indicar situaciones de vulnerabilidad económica o falta de ingresos (por lo tanto, la dependencia hacia agentes externos). Otra variable que sería interesante utilizar es I13 ya que indica si alguna de las habitaciones del hogar es utilizada como lugar de trabajo, aunque una limitación de tomar dicha variable es que podría estar representando a personas ocupadas pero muchas en situación de informalidad.

1.1 Limpieza de la base

Para comenzar con el filtrado de datos, se realizó el mismo procedimiento que para la base EPH hogar. Primero se seleccionó la columna “región” de cada una de las bases (2004 y 2024) y los datos correspondientes al Gran Buenos Aires. Luego se utilizó el comando *str.lower()* para pasar cada uno de los nombres de las columnas a imprenta minúscula, de modo que estas pudieran ser comparables. Se obtuvo un total de dos columnas (*idimph* y *pondih*) no coincidentes entre 2004 y 2024. Estas fueron eliminadas para evitar datos faltantes al momento de unir las bases pertenecientes a diferentes años. Se creó un único *dataframe* de EPH hogar que luego se unió mediante *merge* con la base de datos EPH individual, considerando como parámetros indicadores el código de usuario y número de hogar. A continuación se comprobó si los valores claves estaban duplicados (ya que en la base EPH individual la variable ‘*codusu*’ se repite para cada componente familiar), como así también la coincidencia de datos entre ‘*codusu*’ y ‘*nro_hogar*’ a través de *isin()*. Se creó un *for* para recorrer todas las columnas del *dataframe* y se eliminaron las columnas repetidas, para luego poder eliminar las columnas que sólo contenían valores faltantes.

En relación con el tratado de valores extraños, se realizó un *for* que recorrió cada columna del *dataframe* y comprobó si los datos eran numéricos o no. Para aquellos que sí lo eran, se recorrió luego cada fila y a los datos completados por “...”, 0, -9 o 9 (número representativo de la respuesta “no sabe/no responde”) se los reemplazó por NaN. Posteriormente estas filas fueron eliminadas con el método *dropna()*.

Para finalizar con la limpieza de datos, se obtuvieron los nombres de las columnas que contenían algún NaN y aquellas que no. Las que no tenían ningún NaN se conservaron, mientras que algunas que sí tenían (y eran irrelevantes para el análisis, entre ellas las del código ‘*iv*’ que contenían características de la casa, por ejemplo, el tipo de techo) fueron eliminadas. Otras variables que consideramos que podrían ser importantes para la predicción del desempleo fueron conservadas (por ejemplo, si la familia ocupa algún espacio de la casa

para trabajar o el régimen de tenencia del hogar). Además, consideramos importante eliminar los NaN de *'itf'* ya que resulta imposible que la familia no perciba ningún ingreso (a nivel jubilación, transferencia del estado) y de *'estado'* porque era relevante conocer si la persona estaba ocupada/desocupada o era menor de 10 años. Sin embargo aunque quedaron columnas con valores NaN, se decidió no eliminarlas para determinar si solo con la eliminación de filas la cantidad de datos era suficiente para poder incluirlas al análisis.

1.2 Creación de variables

En este apartado se crearon tres variables que no se encontraban en la base de datos original pero consideramos que serían útiles para predecir a los individuos desocupados. En primer lugar optamos por desarrollar una variable llamada *'proporcion_trabajando'*, que justamente como menciona el nombre, es la proporción de personas que trabajan en el hogar en relación con la cantidad de personas que viven ahí. Primero se filtró el *dataframe* a partir de *'estado'==1*, que hace referencia a las personas en estado activo. Luego se agrupó el total de personas por hogar (a partir del *dataframe* completo) y se guardaron esos datos en una variable llamada *'total_personas_hogar'*. Se realizó lo mismo para el total de trabajadores por hogar (con el nuevo *dataframe*), cuyos datos se guardaron en *'trabajadores_por_hogar'*. Posteriormente se arrojaron estos datos en un nuevo *dataframe* a partir de un merge entre las variables previamente creadas. Para obtener la proporción deseada se dividió la cantidad de personas trabajando en el total de las personas para cada *'codusu'* y *'nro_hogar'*. La variable *'proporcion_trabajando'* podría ser un buen indicador para estimar el desempleo porque indicaría el nivel de participación laboral que aporta casa hogar. Además, varios hogares con una proporción cercana a cero podrían ser señaladores de zonas geográficas desfavorecidas, aunque para esto se necesitaría precisar más la región abordada.

En segundo lugar se creó una variable llamada *'tasa'*. Para simplificar el manejo de datos, del *dataframe* original se extrajeron tres columnas que serían fundamentales para esta nueva variable: *'v3_m'* (monto de ingreso de indemnización por despido), *'v5_m'* (monto de ingreso por subsidio/ayuda social), *'itf'* (ingreso total familiar). La idea principal fue similar a la construcción de la primera variable. Se realizó una suma del ingreso por indemnización y por subsidio o ayuda social por código de usuario. Luego se dividió cada suma por el ingreso total familiar y este resultado fue guardado en la variable *'tasa'*. Posteriormente, al *dataframe* madre se le agregó una nueva columna llamada *'tasa_parcial'* que tomó el valor 1 para los valores de *'tasa'* superiores a 0.5 y 0 caso contrario. Este valor fue elegido arbitrariamente, partiendo de la idea de que si más de la mitad del ingreso familiar se explica por una transferencia monetaria, habría una gran chance de que algún componente familiar esté desempleado. La variable *'tasa_parcial'* podría ser útil ya que ayudaría a capturar la dependencia económica del hogar en relación con contribuciones externas. Una limitación de esta variable es que la ausencia de dependencia de subsidios no implica necesariamente que la persona está empleada, de hecho hay casi un 100% de coincidencia entre la tasa <0.5 y persona empleada y no tan buena coincidencia entre la tasa y los desempleados, pero es *proxy* interesante para considerar.

En tercer lugar se creó la variable *'prom_edad'* a partir del cálculo del promedio de edades dentro de un grupo familiar (por *'codusu'* y *'nro_hogar'*). Luego se creó un dataframe que contuvo solo las variables relacionadas al código de identificación familiar, la edad y una columna binaria llamada *'desocupado'*. Se seleccionaron sólo las filas en las que *'desocupado'* era igual a uno y por otro lado se filtraron los promedios de las edades por hogar menores a 20. A partir de esta información se calculó el porcentaje de datos con edad promedio menor a 20, que representó el 14.71% de la muestra total. A partir de esta variable se podrían establecer algunas relaciones con el desempleo, por ejemplo los hogares con promedios de edad más bajos (sobre todo jóvenes) indicarían que las personas están desempleadas. Sin embargo, dio una ocupación de 95% de menores de 20, esto se puede deber a la muy baja cantidad de desempleados comparada a la de empleados de la base de datos.

1.3 Estadísticas descriptivas

En esta sección se llevaron a cabo procedimientos de estadística descriptiva que creemos podrían ser relevantes para predecir la desocupación. En primer lugar fue necesario controlar que cada hogar estuviese representado una sola vez (por las variables *'codusu'* y *'nro_hogar'*). A partir de aquí se realizó una tabla que agrupó los hogares según la cantidad de personas que viven en ellos y se calculó la media de la proporción de trabajadores y la cantidad de hogares en cada grupo (es decir, grupos de 1, 2, 3 personas). Luego se creó un gráfico de barras (Figura 1) para la media de la proporción de trabajadores utilizando la librería seaborn. En este gráfico se puede observar que las proporciones más altas se ubican en hogares con pocas personas (1,2), lo que es lógico ya que quienes viven solo probablemente necesiten trabajar (a menos que sean estudiantes) y este valor disminuye un poco a medida que se amplía el grupo familiar. Esto puede deberse a que los hogares de familias de 5, 6 integrantes tienen la presencia de niños o adultos mayores que no están en edad de trabajo. Resulta interesante observar que cuando el eje x=12, hay una caída en la proporción de trabajadores. Este caso en principio podría reflejar cierta dependencia económica ya que es un hogar con gran cantidad de participantes pero baja proporción de trabajadores. O se puede deber a un caso particular, deberá analizarse cuantos hogares de 12 integrantes se tomaron para calcular el promedio.

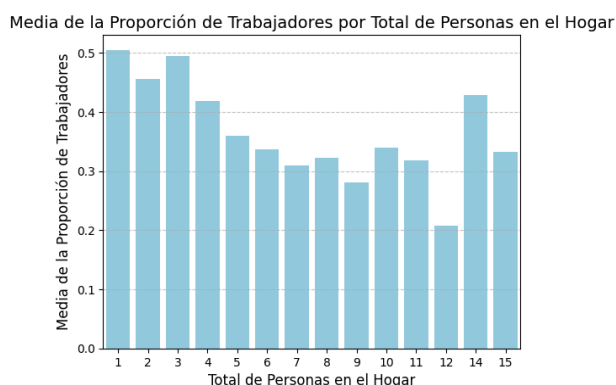


Figura 1: media de la proporción de trabajadores por total de personas en el hogar

Por otro lado se realizó un gráfico representativo del número de hogares con habitaciones destinadas al trabajo 2004 y 2024 (Figura 2). Primero se separó el *dataframe* original en dos *dataframes*: uno para cada año y también se eliminaron los valores faltantes de la columna *'ii3'* (número de habitaciones destinadas al trabajo). Se agruparon los datos

para cada año teniendo en cuenta la cantidad de habitaciones y se calculó el porcentaje de hogares en cada categoría respecto al total de hogares de cada año. Para visualizar los datos se crearon dos gráficos de barras y se observó que, del número de hogares destinados al trabajo, casi el 95% destina sólo una habitación tanto en 2004 como en 2024.

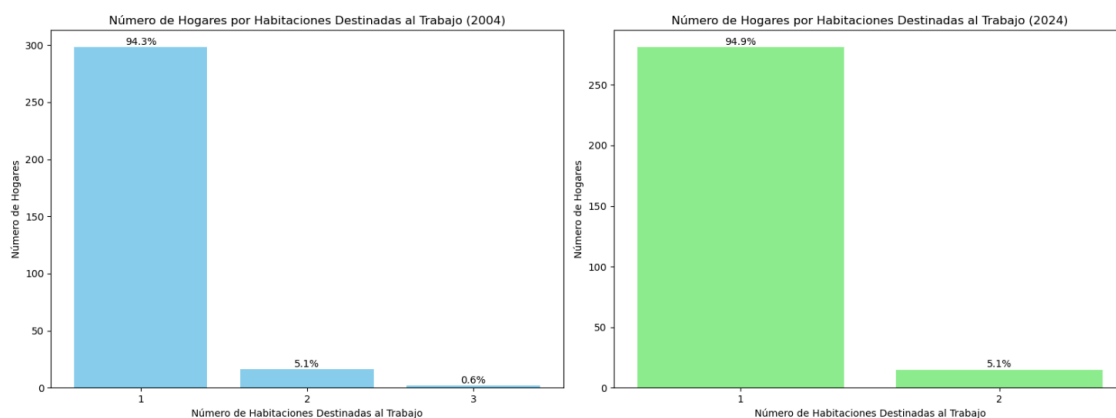


Figura 2: Número de hogares con habitaciones destinadas al trabajo en 2004 y 2024

Por último, se realizó *pie chart* (Figura 3) para representar la distribución porcentual que muestra la proporción de hogares con menores de 10 años que trabajan o no trabajan. Si bien este tipo de gráfico no es tan recomendable para visualizar muchas variables, en este caso solo se utilizaron dos ('trabajan' vs. 'no trabajan'). Se controlaron NaNs en columnas de interés 'ix_men10', 'v19_a' y 'v19_b' y se unificó el tipo de datos a numéricos. Luego se filtraron hogares con menores de 10 años y se creó una columna llamada 'trabajo' que se completó con los mismos datos que 'v19_a' (si hay menores de 10 años que aporten a los ingresos de la familia trabajando) para poder calcular la distribución porcentual de hogares que trabajan y no trabajan. Si bien el porcentaje de menores trabajando es bajísimo, esta información nos daría indicios sobre los hogares que cuentan con algún integrante desempleado. Este mismo análisis se replicó para la variable 'v19_b' que indica si hay menores de 10 años que aporten a los ingresos de la familia solicitando limosna pero el análisis se hizo considerando como total el grupo de niños que sí trabaja. Se obtuvo que un 60% de los niños que trabajan, piden limosna (Figura 4).

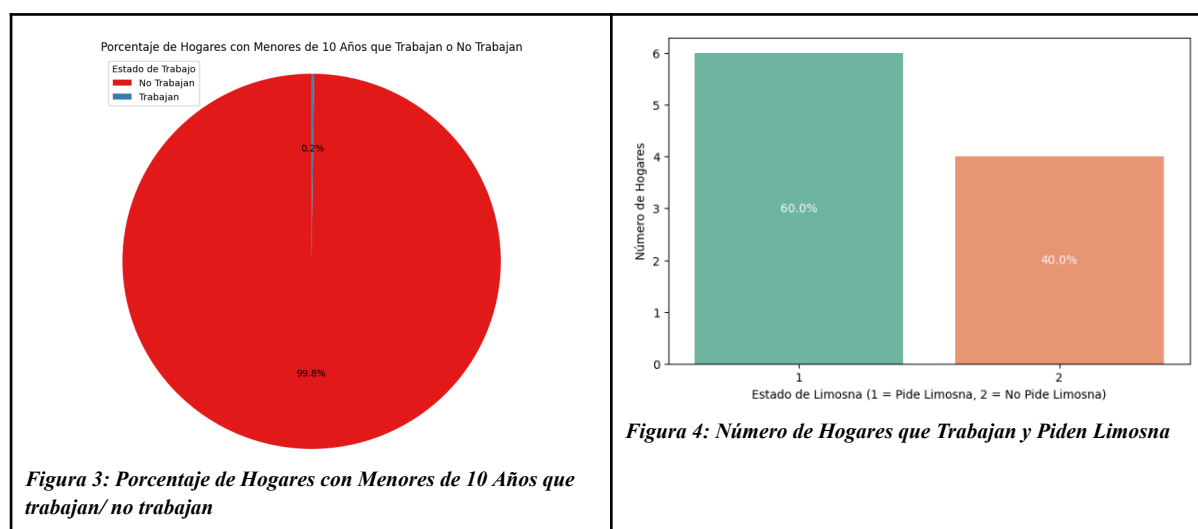


Figura 3: Porcentaje de Hogares con Menores de 10 Años que trabajan/ no trabajan

Figura 4: Número de Hogares que Trabajan y Piden Limosna

Parte 2: Clasificación y regularización

2.1 Preparación de la base

Se realizó una división del *dataframe* principal según el año de la EPH y también se trabajó con las respuestas de la EPH individual trabajadas en el TP3 con las variables dummy. Luego las predicciones se plantearon para cada año por separado pero el procedimiento fue el mismo.

En primer lugar se generó un *dataframe* llamado '*df_prediccion_2024*' que, después de eliminar las columnas con más de un NaN y las columnas con datos no analizables (como por ejemplo, datos con el número 9 que corresponden a 'no responde') se fusionó con el *dataframe* '*datos_respondieron_2024_individual*' correspondiente a la EPH individual. A partir de la unión de ambos *dataframes* se controló que no hubieran columnas repetidas. Se recorrieron las columnas restantes y aquellas que contenían datos categóricos provenientes de la base hogar no analizada en el TP3 (como por ejemplo si la persona vive de jubilación, beca de estudio, alquiler, etc.) fueron pasadas a dummies eliminando la primera categoría para evitar multicolinealidad. Todo este procedimiento de limpieza de datos fue realizado para el año 2004 también.

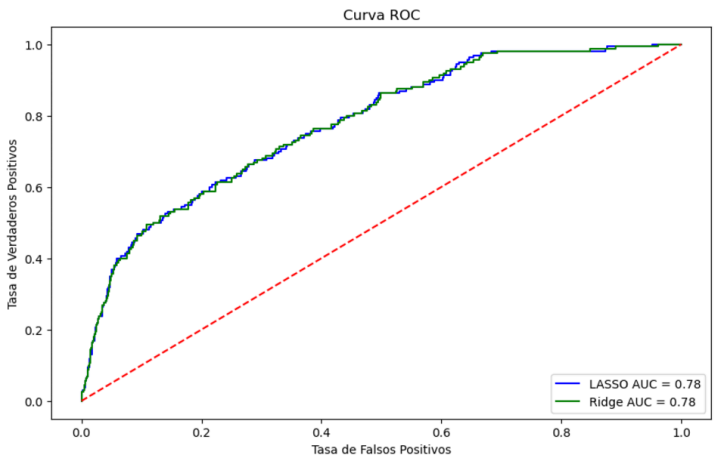
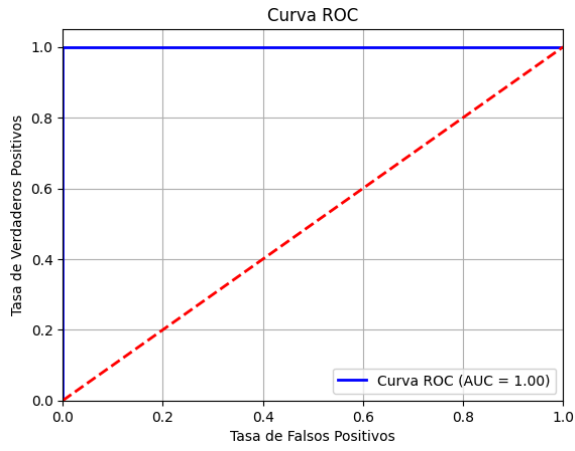
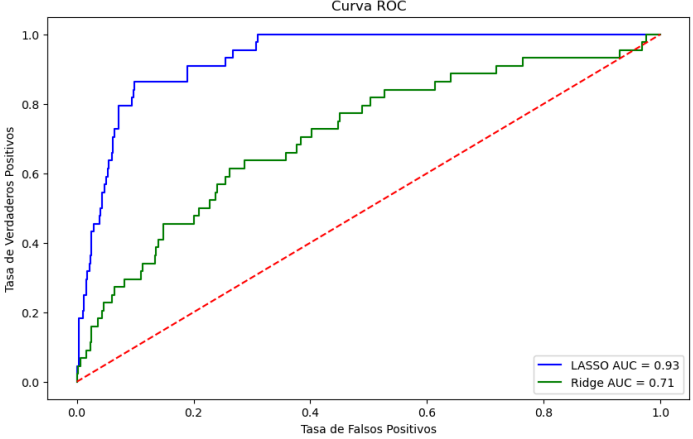
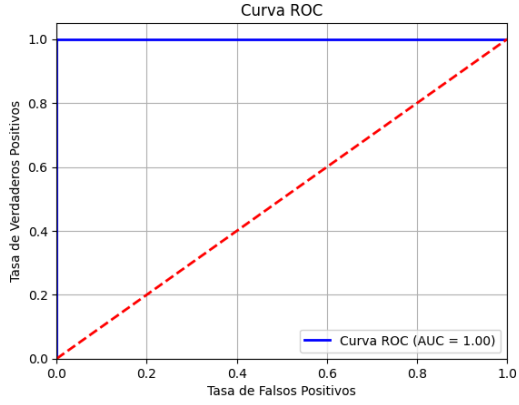
2.2 Predicciones

Se dividieron los datos en dos conjuntos: uno de entrenamiento (70% de observaciones) y otro de prueba (30%) mediante el comando *train_test_split* con el parámetro '*random_state*' configurado en 101.

La variable dependiente seleccionada fue '*desocupado*' (toda variable relacionada a esta fue eliminada como '*estado*', '*cat_inac*' y '*pp02h*'). Esta representa la condición laboral de los individuos y se asignó al vector *y_train* para el conjunto de entrenamiento y a *y_test* para el conjunto de prueba. Las variables independientes, es decir, aquellas utilizadas como predictores, se asignaron a *y_train* y *y_test*, que contienen las características restantes luego de eliminar la columna '*desocupado*'. Adicionalmente, a *x_train* y *x_test* se les añadió una columna de unos para incluir el término de intercepto en el modelo, asegurando así la correcta especificación de la regresión logística que fue el modelo a implementar para predecir.

Por otro lado, a diferencia del TP3, al modelo de regresión se le implementaron las técnicas de regularización LASSO y Ridge con $\lambda=1$ para evitar el sobreajuste y mejorar la generalización del modelo al agregar un término de penalización a la función de costo 0, ayudando a controlar la magnitud de los coeficientes del modelo.

Para evaluar el desempeño del modelo se utilizó el *accuracy* y una matriz de confusión. Además, para evaluar la capacidad discriminativa del modelo, se calculó la curva ROC y el área bajo la curva (AUC). Los resultados de los modelos y la comparación con el modelo utilizado en TP3 (corregido) se detallan en la siguiente tabla (Tabla 1):

	Regresión Logística con Lasso Y Ridge		Regresión Logística TP3
2004	 <p>Curva ROC</p> <p>— LASSO AUC = 0.78 — Ridge AUC = 0.78</p>		 <p>Curva ROC</p> <p>— Curva ROC (AUC = 1.00)</p>
	<p>Matriz de Confusión y precisión del modelo Lasso:</p> $\begin{bmatrix} 2008 & 4 \\ 156 & 4 \end{bmatrix}$ <p>AUC = 0.78 Accuracy = 0.926</p>	<p>Matriz de Confusión y precisión del modelo Ridge</p> $\begin{bmatrix} 2010 & 2 \\ 156 & 4 \end{bmatrix}$ <p>AUC = 0.78 Accuracy = 0.927</p>	<p>Matriz de confusión y precisión del modelo</p> $\begin{bmatrix} 2120 & 0 \\ 0 & 151 \end{bmatrix}$ <p>AUC = 1.0 Accuracy = 1</p>
2024	 <p>Curva ROC</p> <p>— LASSO AUC = 0.93 — Ridge AUC = 0.71</p>		 <p>Curva ROC</p> <p>— Curva ROC (AUC = 1.00)</p>
	<p>Matriz de confusión y precisión del modelo Lasso</p> $\begin{bmatrix} 1113 & 12 \\ 35 & 9 \end{bmatrix}$ <p>AUC = 0.93 Accuracy: 0.95</p>	<p>Matriz de confusión y precisión del modelo Ridge</p> $\begin{bmatrix} 1125 & 0 \\ 44 & 0 \end{bmatrix}$ <p>AUC = 0.71 Accuracy: 0.96</p>	<p>Matriz de confusión y precisión del modelo</p> $\begin{bmatrix} 1614 & 0 \\ 0 & 90 \end{bmatrix}$ <p>AUC = 1 Accuracy = 1</p>

Año 2024:

En términos de precisión, LASSO obtuvo un *accuracy* de 0.95, mientras que Ridge mostró un valor ligeramente superior de 0.96. Sin embargo, cuando se evaluó la capacidad discriminativa mediante la curva ROC, LASSO superó a Ridge, indicando una mayor tasa de verdaderos positivos a lo largo de los diferentes umbrales. Por el contrario, la curva ROC de Ridge fue más cercana a la recta diagonal, lo que refleja una menor capacidad para distinguir entre las categorías positiva y negativa.

En cuanto al AUC, LASSO alcanzó un valor de 0.93, mientras que Ridge obtuvo un AUC de 0.71, lo cual indica que LASSO es más eficaz para discriminar entre las clases, a pesar de tener una precisión ligeramente menor. Ridge, por su parte, no identificó correctamente ninguna clase positiva, lo que sugiere dificultades para clasificar los VP.

Por último, al implementar los modelos de regresión logística con regularización L1 (LASSO) y L2 (Ridge) utilizando $\lambda=1$, se observó una diferencia notable respecto a los resultados obtenidos en el TP3, donde no se aplicó regularización. En el TP3, los modelos sin regularización lograron un *accuracy* de 1.0 y un AUC de 1.0, un desempeño que, aunque parezca ideal, es un indicativo de sobreajuste (*overfitting*). Probablemente el modelo aprendió patrones demasiado específicos del conjunto de datos de entrenamiento, lo que afectaría su capacidad para generalizar a nuevos datos. Este fenómeno es común en conjunto de datos sesgados o con poca diversidad de clases, como pudo haber sido el caso de ausencia de desempleados.

Año 2004:

Tanto LASSO como Ridge obtuvieron resultados similares: 0.926 y 0.927 respectivamente y un AUC de 0.78 para ambos, lo que indica una discriminación moderada entre las clases. Sin embargo, aunque ambos presentan precisión y AUC similares, tienen una capacidad limitada para distinguir de manera óptima entre las clases ya que el AUC se encuentra considerablemente por debajo de 1.0. Estas observaciones se corresponden con las matrices de confusión que muestran que los modelos fueron más efectivos para clasificar correctamente las instancias negativas, al igual que en el TP3, pero tuvieron dificultades para identificar correctamente las clases positivas.

El análisis para el año 2024 en comparación al TP3 también se aplica para el año 2004.

Conclusiones Generales:

El uso de regularización en LASSO (L1) y Ridge (L2) mejoró el desempeño del modelo. LASSO mantuvo una capacidad de precisión y AUC similar al TP3 al simplificar el modelo y eliminar características irrelevantes, evitando el sobreajuste. En cambio, Ridge mostró una ligera disminución en la precisión y el AUC debido a la penalización aplicada a los coeficientes, reduciendo la flexibilidad del modelo.

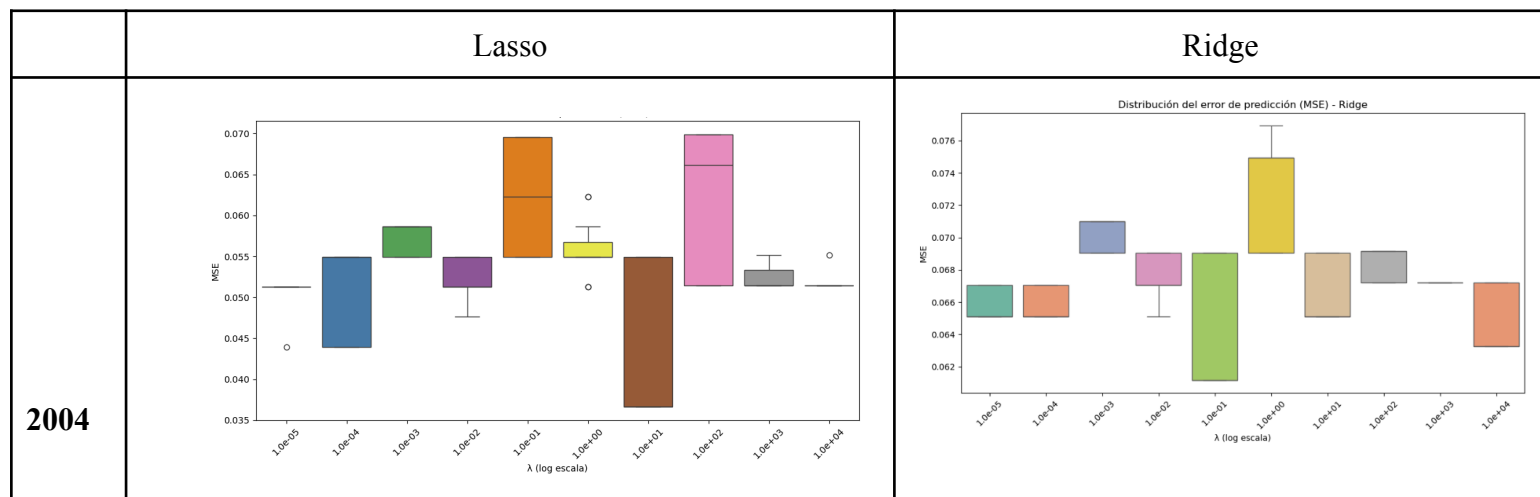
A pesar de estas ligeras reducciones en el desempeño de Ridge, ambos modelos con regularización demostraron una mayor capacidad de generalización. Aunque la precisión y AUC fueron ligeramente menores que en el TP3, la regularización resultó en modelos más robustos y menos propensos al sobreajuste. En conclusión, la implementación de LASSO y Ridge con $\lambda=1$ mejoró la capacidad de generalización demostrando que, aunque el modelo sin regularización logró un desempeño perfecto en los datos de entrenamiento, la regularización es crucial para evitar el sobreajuste y mejorar la robustez ante datos nuevos.

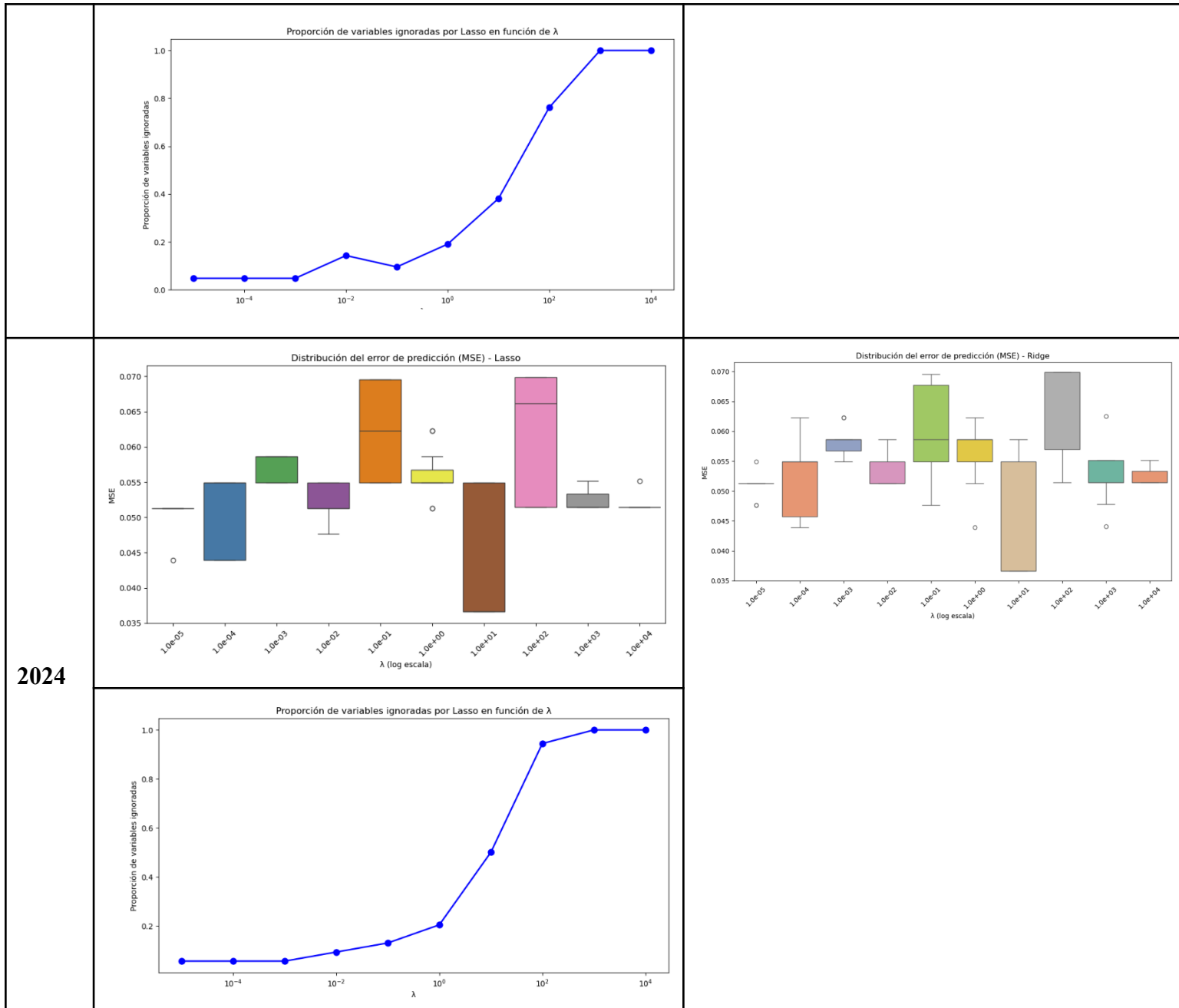
Cabe destacar que el parámetro λ regula la penalización aplicada a los coeficientes en Lasso y Ridge. En nuestro análisis utilizamos un $\lambda = 1$, pero realizamos una segunda ronda seleccionando el λ óptimo mediante *cross-validation*. Este método divide el conjunto de entrenamiento en k subconjuntos (folds), donde el modelo se entrena en $k-1$ folds y se evalúa en el fold restante, calculando el error promedio para determinar el valor óptimo de λ que evita el *underfitting* y el *overfitting*.

La elección de k es crucial porque influye directamente en la calidad de las estimaciones. Un k pequeño reduce la demanda de procesamiento (se entrena con pocos datos) pero aumenta la varianza de las estimaciones, haciéndolas menos estables. En cambio, un k grande utiliza casi todo el conjunto de datos para entrenar el modelo en cada iteración, lo que produce evaluaciones más precisas, pero a costa de que el modelo sea demasiado sensible a pequeñas variaciones en los datos.

Cuando $k = n$ (donde n es el número total de muestras), se realiza una estimación para cada muestra en el conjunto de datos. En este caso, el modelo se entrena con $n-1$ muestras. Este enfoque reduce el sesgo al usar cada punto de datos para evaluar el modelo y estimar de forma precisa el rendimiento. Sin embargo, tiene una alta varianza y es computacionalmente demandante, ya que requiere entrenar y evaluar el modelo n veces.

Vale la pena aclarar que no se utiliza el conjunto de prueba para elegir λ , ya que debe reservarse exclusivamente para evaluar el rendimiento final del modelo. Usarlo en la selección de λ implicaría ajustar el modelo a este subconjunto específico, lo que podría llevar a un sobreajuste y métricas que no reflejen la capacidad real de generalización del modelo.



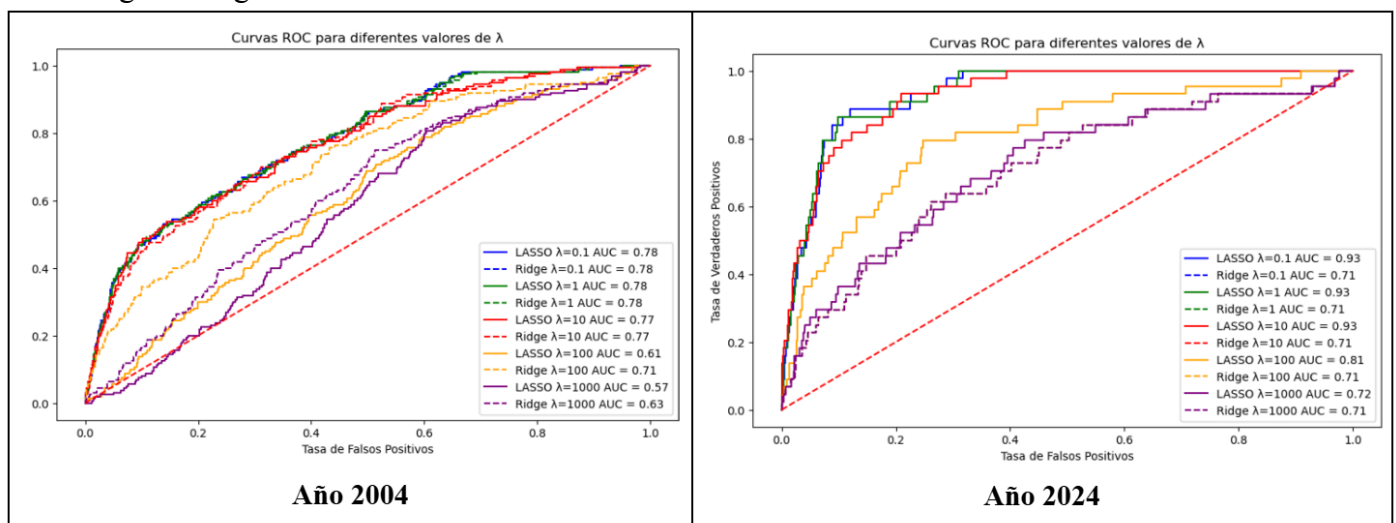


En términos generales, Lasso supera levemente a Ridge en cuanto a la efectividad de la predicción ya que tiende a tener un MSE promedio ligeramente menor en casi todos los valores de λ . En cuanto a la desviación estándar del MSE, Ridge es más sensible a los cambios en la regularización ya que muestra más variabilidad en su rendimiento, especialmente a valores más altos y medios de λ , mientras que Lasso parece ser más estable, con una DE más baja, lo que indica que produce resultados más consistentes.

En el caso de 2004, el valor óptimo de λ para ambos modelos es $1.0e-01$, cuyo MSE es 0.0446. Este valor parece ser el más adecuado en términos de error de predicción para ambos métodos, ya que balancea la regularización con el rendimiento. Además, con este valor de λ , las columnas a eliminar reportadas por el modelo Lasso son '*componente*', '*varón*' y '*prim_inc*'. Estas variables tienen coeficientes cercanos a cero, por lo tanto se eliminaron.

En el caso de 2024, el valor óptimo de λ para ambos modelos es $1.0e+01$, logrando el mejor balance entre regularización y capacidad predictiva. Al aumentar este valor, los modelos muestran un rendimiento decreciente, lo que sugiere que la regularización adicional reduce la capacidad de los modelos para ajustarse a los datos. Este valor de λ representa el punto de inflexión donde el modelo alcanza su mejor rendimiento en términos de MSE promedio. Con este valor de λ , las columnas a eliminar reportadas por Lasso son *decocur*, *gdecocur*, *total_personas*, *tasa_parcial*, *prom_edad_hogar*, *ingreso*, *con_pareja*, *sin_pareja*, *prim_inc*, *uni_inc*, *uni_comp*, *sin_instrucc*, *iv5*, *iv12_1*, *iv12_2*, *v21*, *v22*, *v7*, *v12*, *v13*, *v15*, *v16*, *v18*, *v19_a*, *v19_b*, *itf*, y *ch16*. Al parecer las variables que creímos importantes fueron descartadas para el modelo. Nos llamó la atención que muchas de las variables relacionadas al nivel educativo hayan sido descartadas en ambos años. Puede que sea común en el país que exista gente con formación desempleada y bastantes personas sin formación empleadas pero quizás con malas condiciones de trabajo.

Se probaron diferentes valores de λ para ambos años y se puede corroborar que coincide con el análisis realizado previamente donde el mejor λ es 10 para 2024 y 0.1 para 2004, por lo que la mejor regularización es Lasso para ambos años, a pesar de que Ridge alcance los mismos valores para 2024 con el λ óptimo como se puede observar en los siguientes gráficos.



El análisis de los modelos de regresión logística para los años 2004 y 2024 mostró diferencias en el comportamiento del parámetro de regularización λ . En 2004, aunque el menor error cuadrático medio (MSE) promedio fue de 0.0665 con $\lambda=0.1$ se seleccionó $\lambda=10$ por su mayor consistencia en distintas divisiones de los datos, lo que refleja un enfoque en la estabilidad y generalización del modelo. En 2024, $\lambda=10$ también fue óptimo, con un MSE promedio de 0.0446, menor que en 2004, sugiriendo un mejor ajuste del modelo a los datos más recientes. Comparando métodos de regularización, Ridge mantuvo todos los predictores con ajustes menores. LASSO se adapta mejor a los datos en cada caso, mostrando mayor capacidad para identificar relaciones relevantes. Cabe aclarar que las bases de los dos años utilizaron diferentes cantidad de variables ya que la ausencia de datos del 2004 dificulta la posibilidad de conservarlas. En consecuencia, Lasso elimina muchas más variables en 2024.