TRABAJO PRÁCTICO N°4 - CLASIFICACIÓN Y REGULARIZACIÓN DE DESOCUPACIÓN USANDO LA EPH

Big Data y Aprendizaje Automático para Economistas

GRUPO N°1: Pilar Valdez y José Antonio López

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

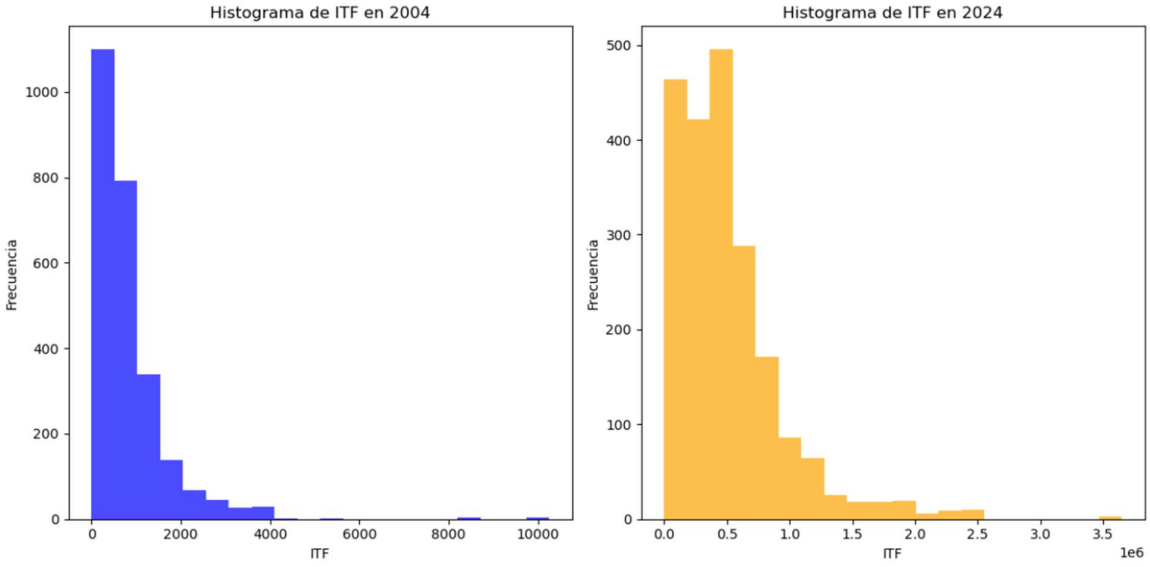
Trabajo sobre las Bases de Datos

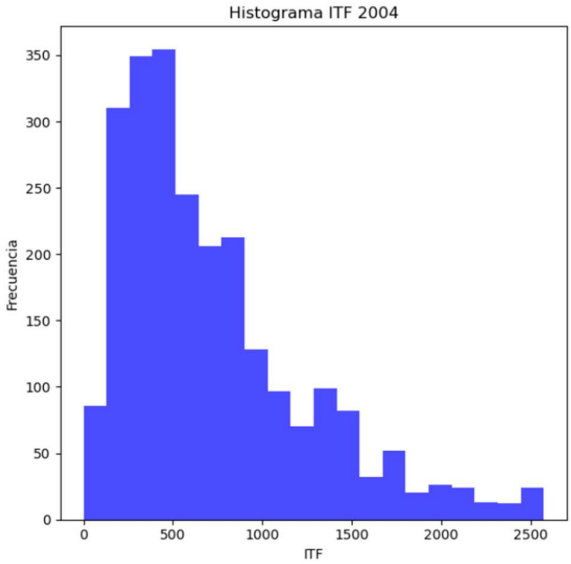
Las bases de datos tenian dos formatos diferentes, .dta y .xls, con Phyton pudimos trabajar esos diferente formatos en un solo programa.

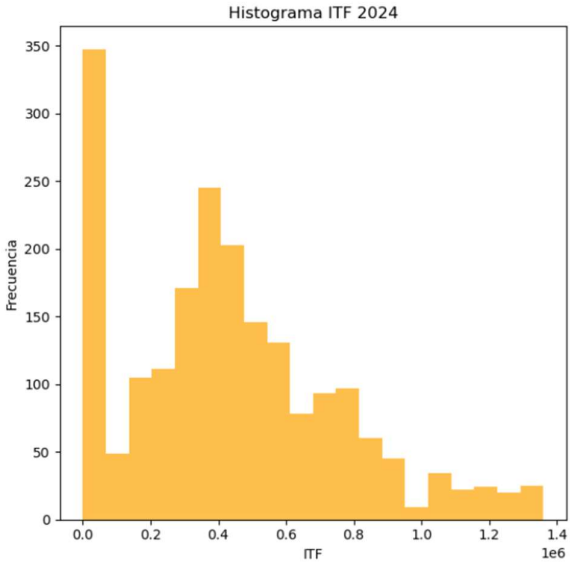
Por otro lado, las variables con respuestas categóricas en el año 2004 se cargaban como lista de datos en la Base, en tanto en 2024 se cargan como numericas. De manera que realizamos la unificación con Jupyter. Podemos mencionar que este trabajo se hizo sobre variables como Estado, Nivel de Educación, Sexo, etc.

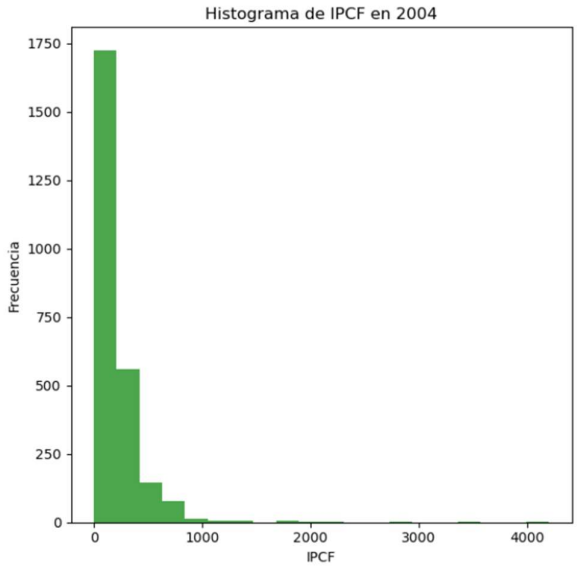
Una vez realizado lo anterior se juntaron las cuatro Bases de Datos anteriores en un solo dataframe. Para hacer un Merge entre las bases de Individuos y la de Hogares utilizamos las columnas CODUSU y Número de Hogar.

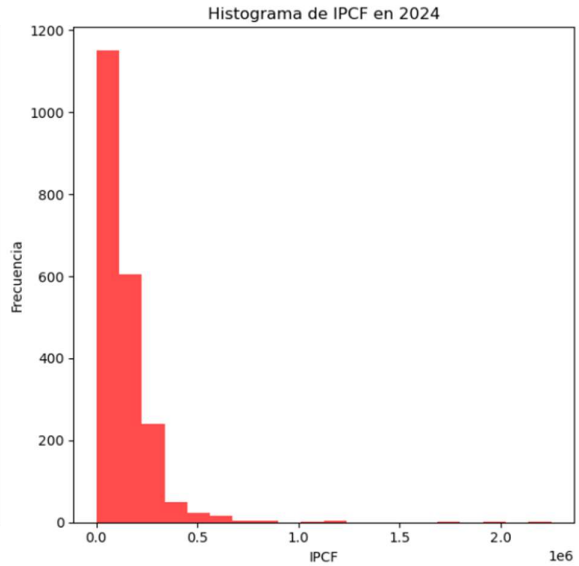
Presentamos a continuación los histogramas de ITF Ingreso total familiar, IPCF Ingreso Per capita familiar y Nivel Educativo.

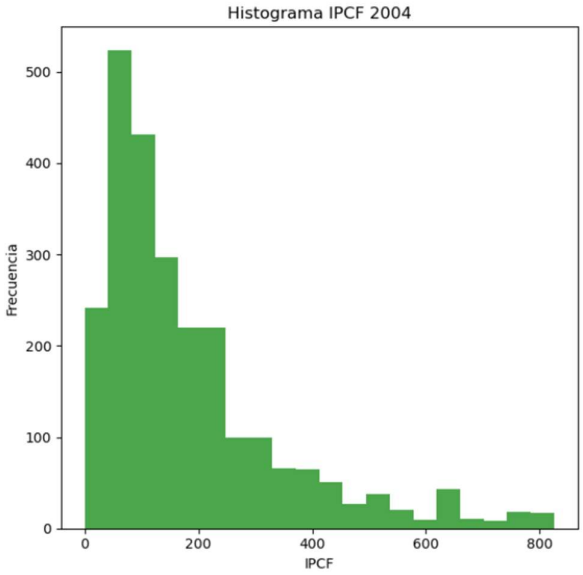


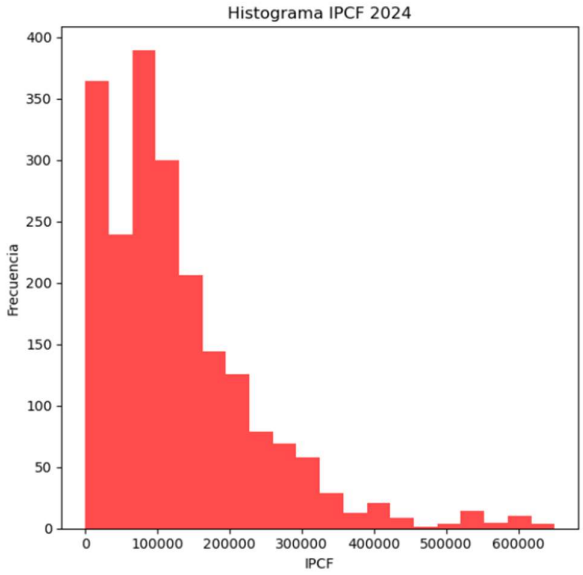

Histograma ITF 2004
Frecuencia
ITF
0
50
100
150
200
250
300
350
0
500
1000
1500
2000
2500

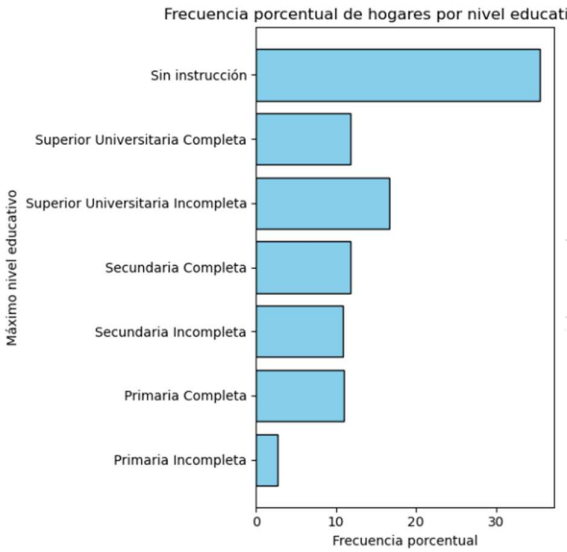

Histograma ITF 2024
Frecuencia
ITF
1e6
0
50
100
150
200
250
300
350
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4

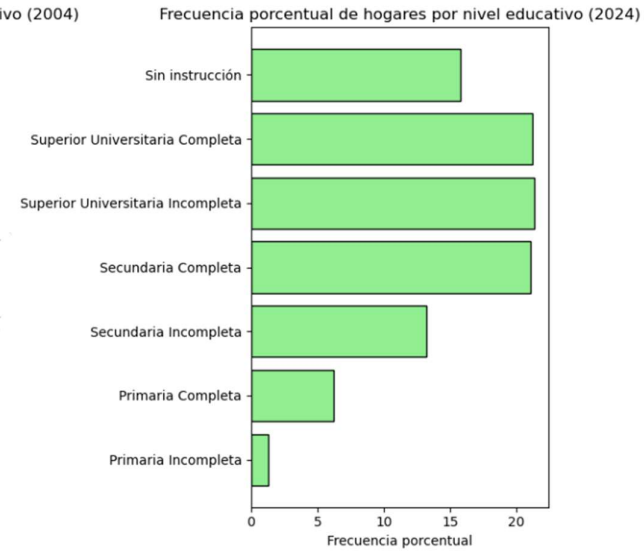

Histograma de IPCF en 2004
Frecuencia
IPCF
0
250
500
750
1000
1250
1500
1750
0
1000
2000
3000
4000

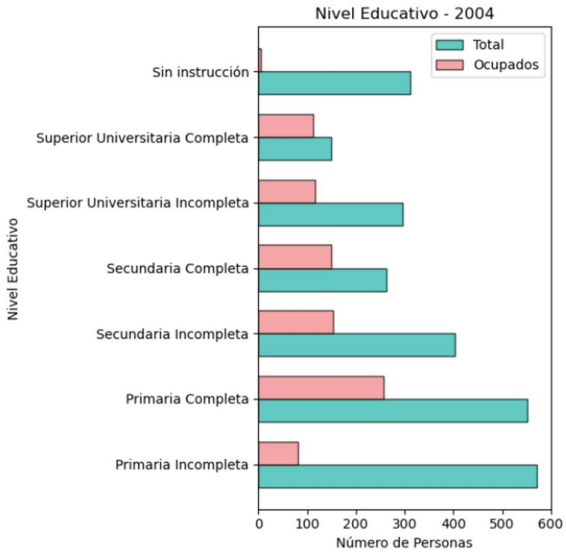

Histograma de IPCF en 2024
Frecuencia
IPCF
1e6
0
200
400
600
800
1000
1200
0.0
0.5
1.0
1.5
2.0

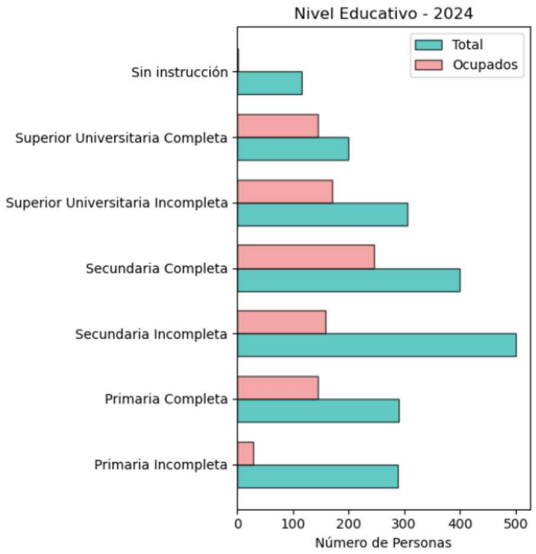

Histograma IPCF 2004
Frecuencia
IPCF
0
100
200
300
400
500
0
200
400
600
800


Histograma IPCF 2024
Frecuencia
IPCF
0
50
100
150
200
250
300
350
400
0
100000
200000
300000
400000
500000
600000


Frecuencia porcentual de hogares por nivel educativo (2004)
Máximo nivel educativo
Sin instrucción
Superior Universitaria Completa
Superior Universitaria Incompleta
Secundaria Completa
Secundaria Incompleta
Primaria Completa
Primaria Incompleta
0
10
20
30
Frecuencia porcentual

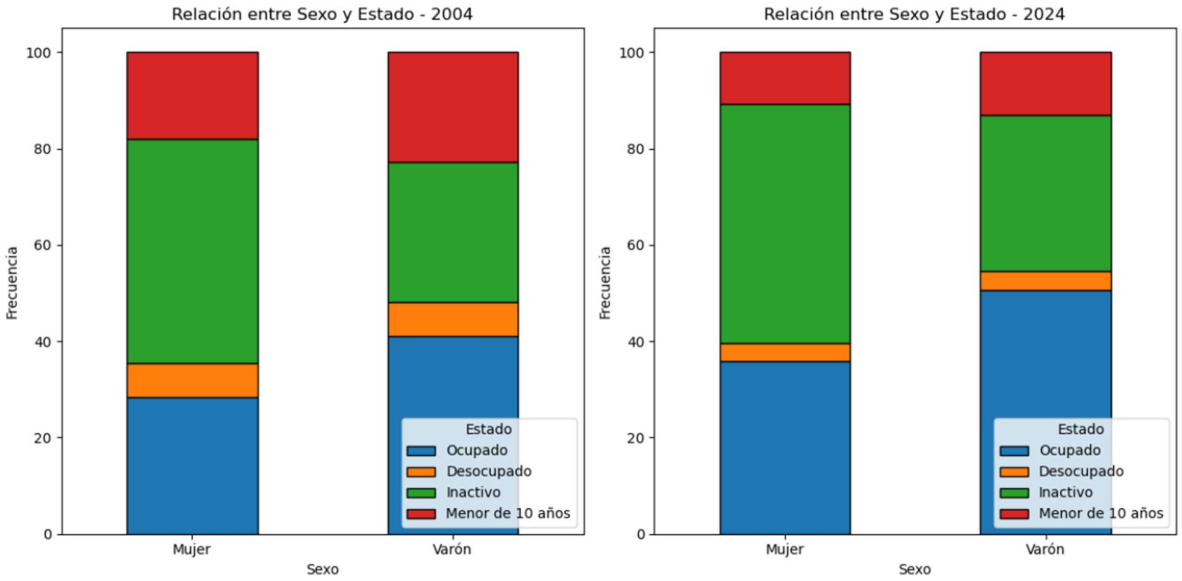

Frecuencia porcentual de hogares por nivel educativo (2024)
Sin instrucción
Superior Universitaria Completa
Superior Universitaria Incompleta
Secundaria Completa
Secundaria Incompleta
Primaria Completa
Primaria Incompleta
0
5
10
15
20
Frecuencia porcentual

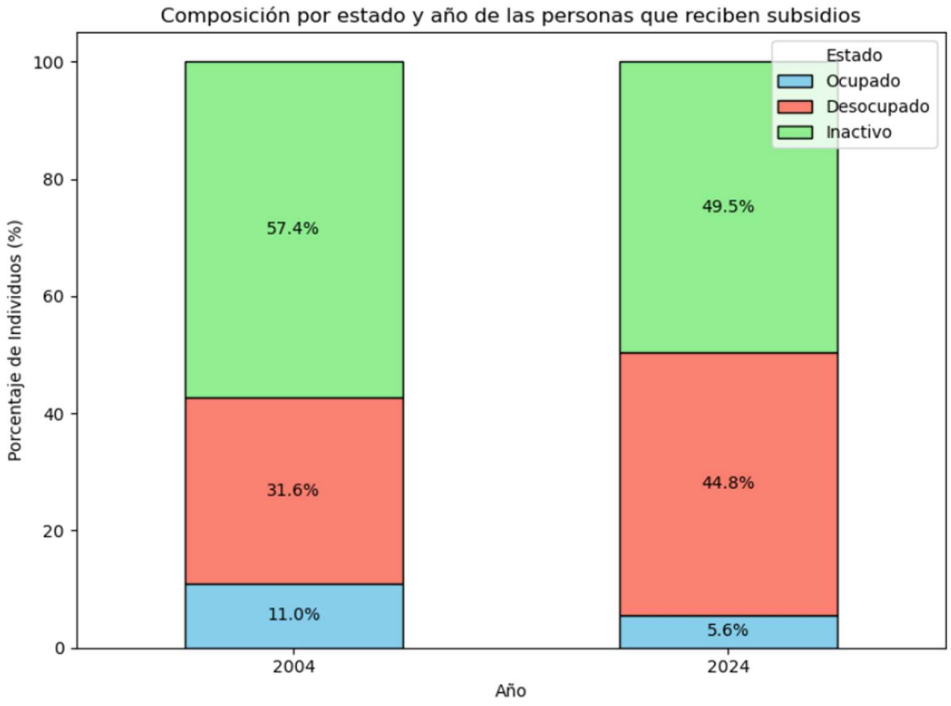


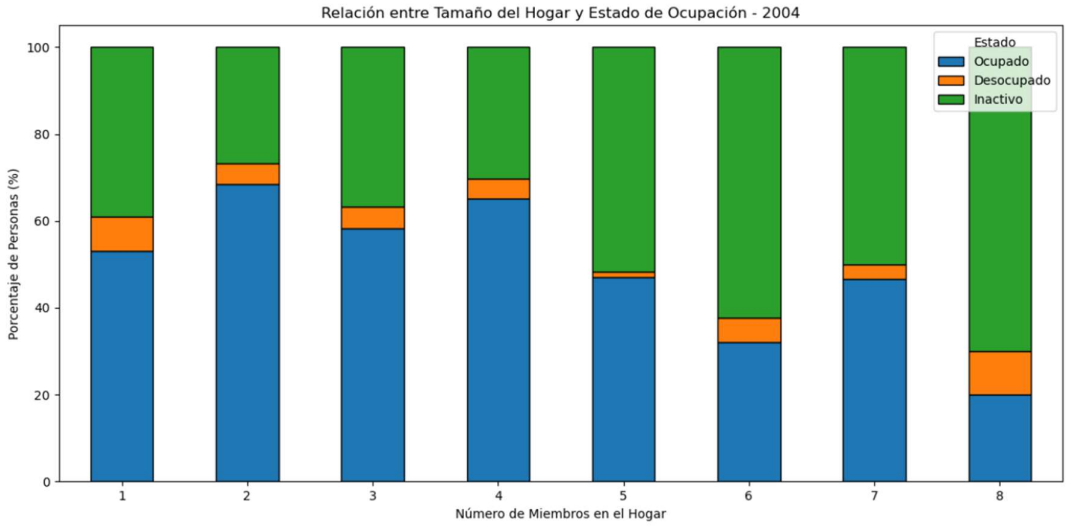


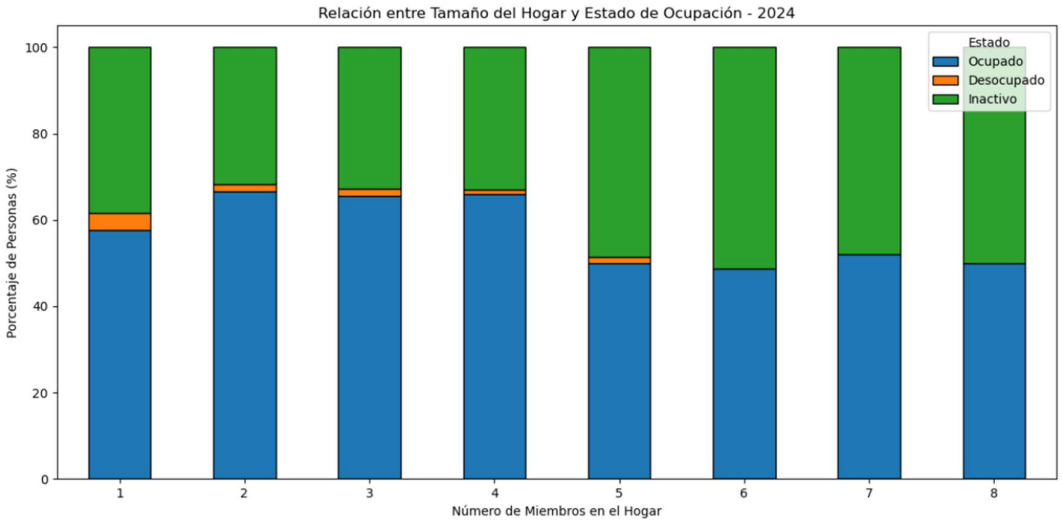
En Relación a las variables que esperamos sean buenos predictores de si una persona está Desocupada, esperamos que las principales sean Nivel de educación, Nivel de Educación, Genero y tamaño del Hogar, a continuación se presenta análisis para cada una.

ESTADO NIVEL_ED	Desocupado	Inactivo	Ocupado	Total
Sin instrucción	1	11	7	19
Primaria Incompleta	14	382	110	506
Primaria Completa	52	385	402	839
Secundaria Incompleta	47	543	313	903
Secundaria Completa	63	199	396	658
Superior Universitaria Incompleta	66	247	288	601
Superior Universitaria Completa	16	75	257	348
Total	259	1842	1773	3874


Relación entre Sexo y Estado - 2004
Relación entre Sexo y Estado - 2024
Frecuencia
Sexo
Mujer
Varón
Estado
Ocupado
Desocupado
Inactivo
Menor de 10 años


Composición por estado y año de las personas que reciben subsidios
Porcentaje de Individuos (%)
Año
2004
2024
Estado
Ocupado
Desocupado
Inactivo
57.4%
31.6%
11.0%
49.5%
44.8%
5.6%





Parte II Clasificación y Regularización

Selección de λ por validación cruzada:

La validación cruzada es una técnica que permite estimar el rendimiento de un modelo en datos no vistos, evitando el sobreajuste que podría ocurrir si utilizamos el conjunto de prueba para ajustar los hiperparámetros (como λ).

El proceso consiste en:

1. **Dividir los datos de entrenamiento:** Se divide el conjunto de entrenamiento en *k* subconjuntos (folds).
2. **Entrenar y evaluar:** Para cada valor de λ:
    - Se utiliza *k-1* folds para entrenar el modelo.

    - El fold restante se utiliza para evaluar el modelo y calcular una métrica de desempeño (por ejemplo, el error cuadrático medio).
3. **Calcular la métrica promedio:** Se calcula el promedio de la métrica de desempeño sobre todas las iteraciones.
4. **Seleccionar el mejor λ:** Se selecciona el valor de λ que obtuvo la mejor puntuación en la validación cruzada.

¿Por qué no usar el conjunto de prueba?

- **Sobreajuste:** Si utilizamos el conjunto de prueba para ajustar los hiperparámetros, estaríamos "entrenando" el modelo en los datos de prueba, lo que podría llevar a una sobreestimación del rendimiento del modelo en datos nuevos.
- **Propósito del conjunto de prueba:** El conjunto de prueba se utiliza únicamente para obtener una estimación imparcial del rendimiento final del modelo una vez que se ha seleccionado el mejor conjunto de hiperparámetros.

Implicancias de k en la validación cruzada

- **k pequeño:** Si k es muy pequeño, la varianza de la estimación del error puede ser alta, ya que cada fold contiene una proporción relativamente grande de los datos. Esto puede dificultar la selección del mejor modelo.
- **k grande:** Si k es muy grande, el tiempo de cálculo aumenta significativamente, ya que se deben entrenar muchos modelos. Además, si el conjunto de datos es pequeño, cada fold puede contener muy pocos datos, lo que puede afectar la estabilidad de las estimaciones.
- **k = n:** Si k es igual al número de muestras, cada fold contiene una sola muestra. Esto se conoce como "leave-one-out cross-validation" y puede ser computacionalmente costoso, especialmente para conjuntos de datos grandes. En este caso, el modelo se estima n veces.

Penalidad L1 (LASSO) y L2 (Ridge) en regresión logística

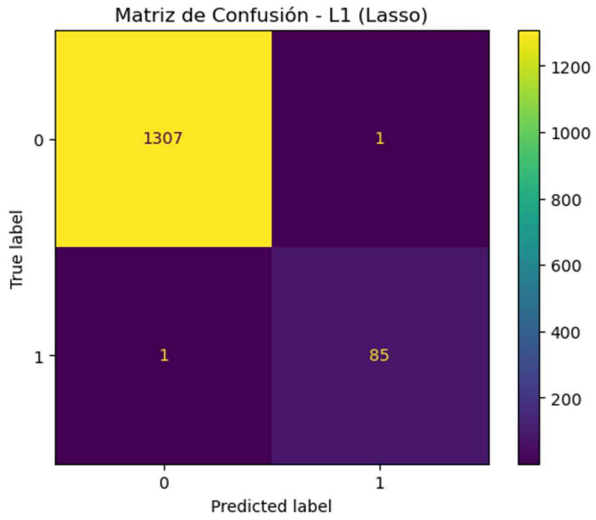
La penalización L1 (LASSO) tiende a reducir algunos coeficientes a cero, lo que puede conducir a la selección de características. La penalización L2 (Ridge) reduce el tamaño de todos los coeficientes, pero generalmente no los hace cero.

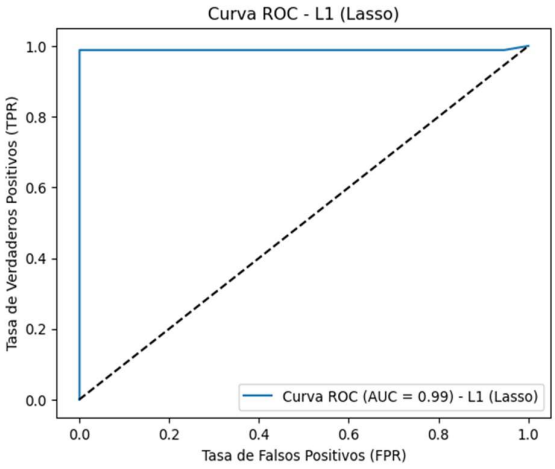
Para implementar estas penalizaciones en Python, utilizamos el parámetro penalty='l1' o penalty='l2' en el objeto LogisticRegression.

Implementación de la penalidad L1 y L2 en regresión logística

Vamos a entrenar una regresión logística con penalización L1 (Lasso) y L2 (Ridge) usando LogisticRegression de sklearn. Luego, evaluaremos los modelos a través de la matriz de confusión, la curva ROC, el AUC y la exactitud en cada uno de los años.

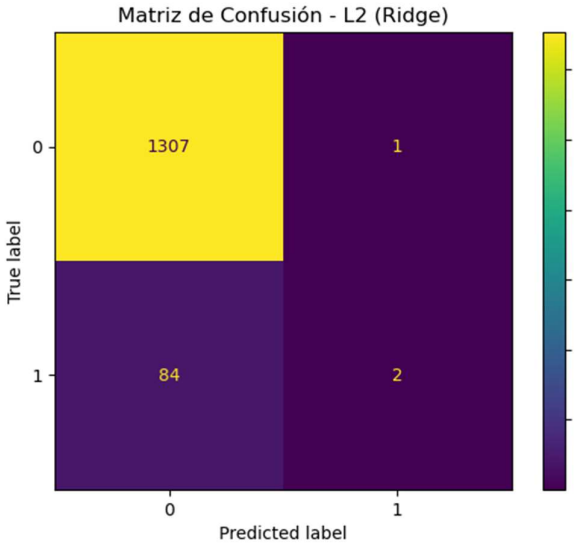
A continuación, los resultados:

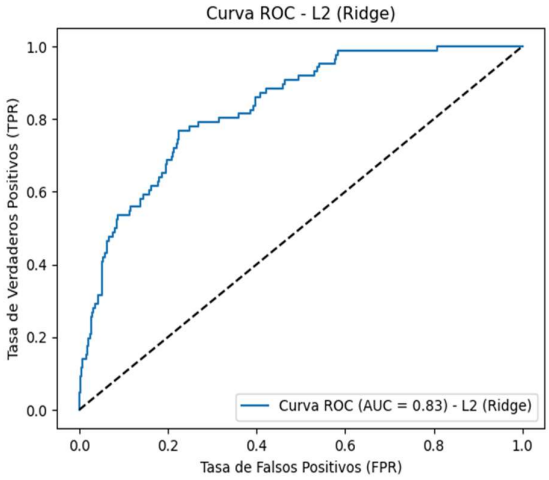




L1 (Lasso) - Exactitud: 0.9986

L1 (Lasso) - AUC: 0.9879





L2 (Ridge) - Exactitud: 0.9390

L2 (Ridge) - AUC: 0.8326

```
Modelo Lasso:
Mejores hiperparámetros: {'model__C': 10000.0}
Exactitud en test: 0.9992826398852224
Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1308
           1       1.00      0.99      0.99        86

    accuracy                           1.00      1394
   macro avg       1.00      0.99      1.00      1394
weighted avg       1.00      1.00      1.00      1394

Matriz de confusión:
 [[1308    0]
 [   1   85]]
```

```
Modelo Ridge:
Mejores hiperparámetros: {'model__C': 0.000774263682681127}
Exactitud en test: 0.9390243902439024
Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97      1308
           1       0.67      0.02      0.04        86

    accuracy                           0.94      1394
   macro avg       0.80      0.51      0.51      1394
weighted avg       0.92      0.94      0.91      1394

Matriz de confusión:
 [[1307    1]
 [  84    2]]
```

Vamos a interpretar los resultados de ambos modelos, el Lasso y el Ridge, en términos de sus hiperparámetros, precisión y las métricas de clasificación.

Modelo Lasso

1. Mejores hiperparámetros: - `{'model__C': 10000.0}` indica que el valor óptimo de `C` (que controla la regularización) es 10000. Un valor alto de `C` reduce la regularización, permitiendo que el modelo se ajuste mejor a los datos de entrenamiento.

2. Exactitud en test: - La precisión en el conjunto de prueba es de **0.9993**, o **99.93%**. Esto muestra que el modelo clasifica correctamente el 99.93% de las observaciones de prueba.

3. Reporte de clasificación:

- Clase 0 (etiqueta de la clase mayoritaria): - `Precision`, `Recall`, y `F1-score` son 1.00 en cada métrica, indicando que el modelo clasificó todos los ejemplos de esta clase sin errores.

- Clase 1 (etiqueta de la clase minoritaria): - La `Precision` es 1.00 y el `Recall` es 0.99, lo cual significa que el modelo identificó casi todos los ejemplos de la clase 1.

- Accuracy (Exactitud), **Macro avg**, y **Weighted avg** son prácticamente 1.00, indicando un rendimiento sobresaliente en ambas clases.

4. Matriz de confusión: - `[[1308, 0], [1, 85]]`: El modelo clasificó correctamente los 1308 casos de la clase 0 y 85 de los 86 de la clase 1, con solo un falso negativo en la clase 1.

Modelo Ridge

1. Mejores hiperparámetros: - `{'model__C': 0.000774263682681127}` indica que el valor óptimo de `C` es 0.00077, lo cual aplica una mayor regularización, restringiendo la complejidad del modelo.

2. Exactitud en test: - La exactitud en el conjunto de prueba es de 0.9390 o 93.9%, que sigue siendo buena pero inferior a la del modelo Lasso.

3. Reporte de clasificación:

- Clase 0 - `Precision`, `Recall`, y `F1-score` son cercanos a 1.00, indicando que el modelo maneja bien la clase mayoritaria.

- Clase 1: - `Precision` es 0.67, mientras que el `Recall` es de solo 0.02. Esto significa que el modelo identificó correctamente muy pocos ejemplos de la clase 1 (solo el 2%), por lo que su capacidad para capturar esta clase es baja.

- Macro avg y Weighted avg:

- Las métricas muestran la baja capacidad del modelo para clasificar adecuadamente ambas clases, en especial la clase minoritaria.

4. Matriz de confusión:

- `[[1307, 1], [84, 2]]`: El modelo clasificó correctamente casi todos los casos de la clase 0, pero solo 2 de los 86 casos de la clase 1, resultando en 84 falsos negativos.

Comparación

- Modelo Lasso es mucho más preciso y capaz de identificar ambas clases, con una precisión cercana al 100%.

- Modelo Ridge presenta una alta precisión para la clase mayoritaria (0) pero es ineficaz para la clase minoritaria (1) debido a la alta regularización.

- Esto indica que el modelo Lasso, con menor regularización, se adapta mejor a la complejidad de los datos, logrando capturar ambas clases de manera precisa.