INTELIGENCIA ARTIFICIAL

PROYECTO FINAL:
Modelo de clasificación de riesgo crediticio: Un enfoque con regresión logística multiclase (softmax)

Docente:

Dr. Anabel Martín González

Equipo 2:

Luis Alfredo Cota Armenta – LCC – 9° semestre

Carlos A. Ruiz Domínguez – LCC – 9° semestre

Edgar Sabido Cortés – LCC – 9° semestre

Fecha de entrega:

02 de diciembre del 2024

CONTENIDO

INTRODUCCIÓN 3
METODOLOGÍA 4
1. Modelo Matemático 4
Relación lineal 4
Función de activación softmax 4
2. Optimización con optimizador Adam 5
Media móvil del gradiente (Momentum) 5
Media móvil del gradiente al cuadrado (RMSProp) 5
Correcciones de sesgo 5
Actualización de los parámetros 6
3. Regularización L2 (Ridge) 6
L2(Ridge) 6
4. Función de pérdida: Entropía cruzada 7
5. Matriz de confusión y métricas de evaluación 7
Matriz de confusión 7
Precision 9
Recall 9
F1-score 9
Support 10
Accuracy 10
Macro promedios (macro average) 10
Promedios ponderados (weighted average) 11
Curva de aprendizaje 11
Curva ROC y AUC 12
RESULTADOS 14
Descripción de la base de datos 14
EDA 14
Pipeline del procesamiento de datos 31
Gráficas de entrenamiento 35
Conjunto de entrenamiento 35

Conjunto de validación 35
Curva de aprendizaje: Entrenamiento vs. Validación 36
Resultados del desempeño del entrenamiento 36
Métricas del conjunto de entrenamiento 36
Métricas del conjunto de validación 37
Resultados del desempeño de la prueba 38
Métricas 38
Gráficas ROC y AUCs 39
Gráfica de la distribución de probabilidades predichas por clase 42
Ejemplos de muestras mal clasificadas 43
CONCLUSIÓN 46
Resumen del algoritmo implementado 46
Resultados del desempeño obtenidos en la prueba 48
Explicación de los errores obtenidos 50
Lo aprendido en el proyecto 51

INTRODUCCIÓN

Modelo de Regresión Logística para Riesgo Crediticio

Descripción del Problema

El riesgo crediticio representa uno de los mayores desafíos para las instituciones financieras. Este problema se define como la probabilidad de que un prestatario no cumpla con sus obligaciones de pago, ya sea de forma parcial o total. Dado que las características financieras, sociales y demográficas de los clientes varían considerablemente, predecir el comportamiento de pago es una tarea compleja. Factores como el historial crediticio, los ingresos, el empleo y las condiciones macroeconómicas pueden influir significativamente en la capacidad de los solicitantes para cumplir con sus obligaciones financieras.

La falta de herramientas adecuadas para evaluar el riesgo puede llevar a decisiones erróneas, como la aprobación de créditos a clientes con alta probabilidad de incumplimiento o la negativa a otorgar préstamos a personas solventes. Estas malas decisiones no solo incrementan las pérdidas financieras para las instituciones, sino que también afectan su reputación y relación con los clientes.

Importancia del Problema

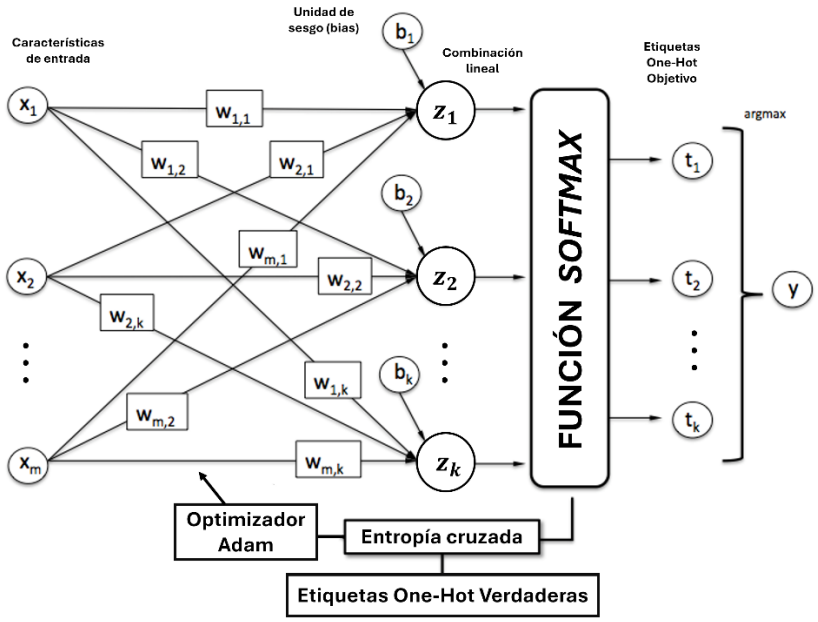
Abordar el riesgo crediticio es fundamental tanto para las instituciones financieras como para la economía en general. Desde el punto de vista organizacional, una gestión deficiente del riesgo crediticio puede resultar en altos niveles de cartera vencida, lo que impacta negativamente la liquidez, la rentabilidad y el cumplimiento de las normativas del sector financiero.

Por otro lado, desde una perspectiva social, la falta de acceso al crédito debido a evaluaciones inexactas puede limitar el crecimiento económico, afectando tanto a individuos como a empresas. Una evaluación eficiente del riesgo crediticio contribuye a una asignación justa de los recursos financieros, promoviendo un sistema más equitativo y funcional.

METODOLOGÍA

1. Modelo Matemático

El núcleo del modelo es la *regresión logística multiclase*. El modelo tiene la siguiente forma:



Relación lineal

Se calcula una combinación lineal para cada una de las características de entrada de la siguiente forma:

$$z_j = W_j^T X + b$$

Donde:

z_j: Combinación lineal de la característica *j*.

$X \in \mathbb{R}^{n \times m}$: Conjunto de datos con *n* muestras y *m* características.

$W_j^T \in \mathbb{R}^{m \times k}$: Transpuesta de la matriz de pesos para las *k* clases de la característica *j*.

$b \in \mathbb{R}^k$: Vector de sesgos o *bias* para las k clases.

Función de activación softmax

Convierte las salidas lineales z_i en probabilidades, pues se trata de la generalización de la función sigmoide. Se calcula de la siguiente manera:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Donde:
z_i : La puntuación lineal calculada para la clase i, es decir, $z_i = XW_i + b_i$.
e^{z_i}: La exponencial de z_i, que transforma las puntuaciones en valores positivos.
K : Número total de clases.
$\sum_{j=1}^{K} e^{z_j}$: Suma de las exponenciales de todas las puntuaciones, usada para normalizar las probabilidades.

Esto garantiza que las probabilidades estén en el rango [0,1] y la suma de las probabilidades para todas las clases sea 1.

2. Optimización con optimizador Adam

El optimizador Adam (**Adaptive Moment Estimation**) fue utilizado para actualizar los parámetros del modelo, como los **pesos** y el **sesgo**, a lo largo del proceso de entrenamiento, de manera que fuese más eficiente, estable y robusto en comparación con otros métodos como el gradiente descendente clásico.
.

Media móvil del gradiente (Momentum)

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

Donde:
m_t : Promedio móvil del gradiente en el paso t .
β_1: Tasa de decaimiento exponencial para el promedio móvil (valor común: 0.9).
m_{t-1}: Valor del promedio móvil en el paso anterior.
g_t: Gradiente calculado en el paso t.

Media móvil del gradiente al cuadrado (RMSProp)

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Donde:
v_t: Promedio móvil del gradiente al cuadrado en el paso t.
β_2: Tasa de decaimiento exponencial para este móvil (valor común: 0.999).
v_{t-1}: Valor del promedio móvil al cuadrado en el paso anterior.

Correcciones de sesgo

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t} \, , \widehat{v_t} = \frac{v_t}{1 - \beta_2^t}$$

Donde:
$\widehat{m_t}$, $\widehat{v_t}$: Valores ajustados para corregir el sesgo inicial.
β_1: Tasa de decaimiento exponencial para el promedio móvil (valor común: 0.9).

β_2: Tasa de decaimiento exponencial para este móvil (valor común: 0.999).

Actualización de los parámetros

$$W_{t+1} = W_t - \frac{n\widehat{m_t}}{\sqrt{\hat{v}} + \varepsilon}$$

Donde:
W_t: Valor de los pesos en el paso t.
n: Tasa de de aprendizaje, controla el tamaño de los pasos.
ε: Término pequeño para evitar divisiones por cero (valor común: 10^{-8}).
$\widehat{m}, \hat{v}$: Valores ajustados del promedio móvil del gradiente y su cuadrado.

3. Regularización L2 (Ridge)

Regularizar ayuda a limitar la complejidad del modelo penalizando los valores de los pesos. Por su parte, la regularización L2 agrega la suma de los cuadrados de los coeficientes del modelo a la función de pérdida, por lo que tiende a reducir los valores de los coeficientes, pero no los elimina completamente. Esto ayuda a distribuir el impacto de cada característica de manera más uniforme, lo cual puede ser beneficioso si se sospecha que muchas características tienen algún grado de relevancia.

L2(Ridge)

$$L_{reg} = \text{Pérdida} + \frac{\lambda}{2}\|W\|^2 = \text{Pérdida} + \frac{\lambda}{2}\sum_{i=1}^{n} w_i^2$$

Donde:
L_{reg}: Término de penalización.
n: Número total de pesos del modelo.
w_i: Pesos del modelo.
λ: Coeficiente de regularización, controla la intensidad de la penalización.
$\|W\|^2$: Es la notación de la norma L2 (o norma Euclidiana) del vector de pesos *W*, que es esencialmente $\sqrt{\sum_{i=1}^{n} w_i^2}$, pero al cuadrado para la regularización, por lo que se escribe simplemente como $\sum_{i=1}^{n} w_i^2$.

4. Función de pérdida: Entropía cruzada

La entropía cruzada mide la discrepancia entre las probabilidades predichas ($\hat{y}$) y las verdaderas (y):

$$L = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} y_{ij}\log\left(\widehat{y_{ij}}\right)$$

Donde:
L: Valor de la pérdida.
n: Número de muestras.
K: Número de clases.
y_{ij}: Etiqueta verdadera para la muestra i y clase j. Toma el valor de 1 si pertenece a esa clase y 0 si no.
$\widehat{y_{ij}}$: Probabilidad predicha para la muestra i y j.
$\log\left(\widehat{y_{ij}}\right)$: Penalización cuando $\widehat{y_{ij}}$ es baja para la clase correcta.

5. Matriz de confusión y métricas de evaluación

Matriz de confusión

La matriz de confusión es una herramienta fundamental para evaluar el rendimiento de un modelo de clasificación, ya que permite detectar clases que el modelo predice mal consistentemente.

		Clasificación predicha			
	Clases	***P1$_P$***	***P2$_P$***	***P3$_P$***	***P4$_P$***
Clasificación real	***P1$_R$***	**TP**			
	P2$_R$		**TP**		
	P3$_R$			**TP**	
	P4$_R$				**TP**

Sus componentes son:
True Positives (TP): Son las instancias predichas correctamente como una clase específica.
False Positives (FP): Se trata de las instancias predichas como una clase específica, pero que realmente pertenecen a otra clase.
False Negatives (FN): Se refiere a las instancias que pertenecen a una clase específica, pero se predicen como otra clase.
True Negatives (TN): Corresponde a las instancias que no pertenecen a la clase específica y tampoco se predicen como tal.

Para poder calcular los valores TP, TN, FP y FN de las diferentes clases de la matriz de confusión anterior, es necesario tener en cuenta que:

TP o *Verdaderos Positivos*: Son los casos en los que la clase predicha y la clase real son las mismas. Estos valores están ubicados en la diagonal principal. Para hallar los TP de una clase P_k específica basta con tomar su valor en la celda correspondiente dentro de la diagonal principal:

$$TP_{P1} = (P1_R, P1_P)$$
$$TP_{P2} = (P2_R, P2_P)$$
$$TP_{P3} = (P3_R, P3_P)$$
$$TP_{P4} = (P4_R, P4_P)$$

FP o *Falsos Positivos*: Son los casos en los que la clase predicha es una clase específica, pero la clase real es otra. Estos valores están ubicados en la columna correspondiente a la clase predicha, excluyendo la diagonal principal. Para hallar los FP de una clase P_k específica se tiene que realizar su suma correspondiente de entre las siguientes:

$$FP_{P1} = (P2_R, P1_P) + (P3_R, P1_P) + (P4_R, P1_P)$$
$$FP_{P2} = (P1_R, P2_P) + (P3_R, P2_P) + (P4_R, P2_P)$$
$$FP_{P3} = (P1_R, P3_P) + (P2_R, P3_P) + (P4_R, P3_P)$$
$$FP_{P4} = (P1_R, P4_P) + (P2_R, P4_P) + (P3_R, P4_P)$$

FN o *Falsos Negativos*: Son los casos en los que la clase real es una clase específica, pero se predice otra clase. Estos valores están ubicados en la fila correspondiente a la clase real, excluyendo la diagonal principal. Para hallar los FN de una clase P_k específica se tiene que realizar su suma correspondiente de entre las siguientes:

$$FN_{P1} = (P1_R, P2_P) + (P1_R, P3_P) + (P1_R, P4_P)$$
$$FN_{P2} = (P2_R, P1_P) + (P2_R, P3_P) + (P2_R, P4_P)$$
$$FN_{P3} = (P3_R, P1_P) + (P3_R, P2_P) + (P3_R, P4_P)$$
$$FN_{P4} = (P4_R, P1_P) + (P4_R, P2_P) + (P4_R, P3_P)$$

TN o *Verdaderos Negativos*: Son los casos en los que ni la clase predicha ni la clase real son la clase específica. Están fuera de las filas y columnas correspondientes a una clase específica. Para hallar los TN de una clase P_k específica se tiene que realizar su suma correspondiente de entre las siguientes:

$$TN_{P1} = Total\ de\ instancias - (TP_{P1} + FP_{P1} + FN_{P1})$$
$$TN_{P2} = Total\ de\ instancias - (TP_{P2} + FP_{P2} + FN_{P2})$$
$$TN_{P3} = Total\ de\ instancias - (TP_{P3} + FP_{P3} + FN_{P3})$$
$$TN_{P4} = Total\ de\ instancias - (TP_{P4} + FP_{P4} + FN_{P4})$$

Donde:
TP_{Pk}: Representa los TP de una clase específica P_k.
FP_{Pk}: Representa los FP de una clase específica P_k.
FN_{Pk}: Representa los FN de una clase específica P_k.
TN_{Pk}: Representa los TN de una clase específica P_k.
Pk_R: Representa la Clase Real de la clase específica P_k o la fila de la matriz.
Pk_P: Representa la Clase Predicha de la clase específica P_k o la columna de la matriz.
(Pk_{1_R}, Pk_{2_P}): Representa la celda dentro de la matriz y P_{k1} y P_{k2} son las respectivas clases específicas o fila y columna de la matriz.
Con los resultados anteriores es posible calcular las siguientes métricas describen mejor el rendimiento del modelo:

Precision

Mide la exactitud de las predicciones positivas del modelo. La precisión es especialmente útil cuando el costo de una predicción positiva falsa es alto. Se calcula como:

$$Precision = \frac{TP}{TP + FP}$$

Donde:
TP: Verdaderos positivos.
FP: Falsos positivos

Recall

También llamado *Exhaustividad* o *Sensibilidad*, mide la capacidad del modelo para identificar todas las instancias positivas. Es crucial cuando es importante capturar todos los casos positivos. Se calcula como:

$$Recall = \frac{TP}{TP + FN}$$

Donde:
TP: Verdaderos positivos.
FN: Falsos negativos.

F1-score

Es la *media armónica* de la *precisión* y el *recall*. Proporciona un balance entre ambos, especialmente útil cuando hay un desequilibrio de clases y es necesario considerar tanto la precisión como el recall. Es útil cuando necesitas un equilibrio

entre precisión y recall y no puedes sacrificar uno por el otro. Pero es particularmente valioso en situaciones donde tanto los falsos positivos como los falsos negativos tienen un costo significativo. Se calcula como:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Support

El support de una clase específica representa la cantidad total de instancias que hay de esta.

Accuracy

Indica qué porcentaje de predicciones del modelo fueron correctas. Se calcula como:

$$Accuracy = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}}$$

Macro promedios (macro average)

Los macro promedios calculan las métricas precisión, recall y F1-score individualmente para cada clase y luego promedian estos valores. No tienen en cuenta el tamaño de cada clase, por lo que cada clase tiene el mismo peso en el promedio, independientemente de cuántas instancias tenga.

Macro Precision

$$Macro\ Precision = \frac{1}{K}\sum_{i=1}^{K} \text{Precision}_i$$

Macro Recall

$$Macro\ Recall = \frac{1}{K}\sum_{i=1}^{K} \text{Recall}_i$$

Macro F1-score

$$Macro\ F1 - score = \frac{1}{K}\sum_{i=1}^{K} \text{F1} - \text{Score}_i$$

Donde:
K: Es el número total de clases.
n_i: Es el número de instancias o support de la clase i.

Promedios ponderados (weighted average)

Los promedios ponderados tienen en cuenta el soporte (el número de instancias de cada clase). Esto significa que las métricas de clases con más instancias tienen más peso en el promedio.

Weighted Precision

$$Weighted\ Precision = \sum_{i=1}^{K} \left(\frac{n_i}{N} \times \text{Precision}_i\right)$$

Weighted Recall

$$Weighted\ Recall = \sum_{i=1}^{K} \left(\frac{n_i}{N} \times \text{Recall}_i\right)$$

Weighted F1-score

$$Weighted\ F1-score = \sum_{i=1}^{K} \left(\frac{n_i}{N} \times \text{F1}-\text{score}_i\right)$$

Donde:
K: Es el número total de clases.
n_i: Es el número de instancias u support de la clase i.
N: Es el número total de instancias o la sumatoria de los support de las *K* clases.

Curva de aprendizaje

Esta herramienta visual y analítica permite evaluar el rendimiento de un modelo de machine learning a lo largo del tiempo, a medida que se entrena con más datos. Además, la curva de aprendizaje permite:

El diagnóstico del rendimiento del modelo

- **Sesgo alto**: Las curvas de entrenamiento y validación se estabilizan en un valor de error alto. Esto indica que el modelo no es lo suficientemente complejo.
- **Varianza alta**: La curva de error de entrenamiento es baja, mientras que la de validación es alta. Esto indica sobreajuste.

La optimización del modelo

- **Ajuste de hiperparámetros**: Permite identificar si se necesitan más datos de entrenamiento, si se deben ajustar los hiperparámetros o si se debe cambiar la complejidad del modelo.

- **Evaluación del Tamaño del Conjunto de Datos**: Ver si agregar más datos de entrenamiento mejora el rendimiento o si el modelo ya ha alcanzado su límite.

El monitoreo del proceso de entrenamiento

- **Seguimiento del Progreso**: Ver cómo cambia el error de entrenamiento y validación a lo largo de las épocas.
- **Detección de Problemas**: Identificar problemas como el estancamiento en el entrenamiento, donde el error no disminuye con más datos.

Componentes de la curva de aprendizaje

Una curva de aprendizaje típica incluye dos curvas principales:

- **Error de Entrenamiento**: Muestra el error del modelo en el conjunto de datos de entrenamiento a medida que se entrena con más datos o épocas. Tener una disminución continua sugiere que el modelo está aprendiendo de los datos.
- **Error de Validación**: Muestra el error del modelo en el conjunto de datos de validación a medida que se entrena con más datos o épocas. Idealmente, debería disminuir hasta un punto y luego estabilizarse. Si aumenta, indica sobreajuste.

Curva ROC y AUC

Curva ROC

La curva Receiver Operating Characteristic (ROC) es un gráfico que muestra la capacidad de un modelo para distinguir entre clases. Para un modelo de *clasificación binaria*, la curva ROC se construye representando la *Tasa de Verdaderos Positivos (TPR)* contra la *Tasa de Falsos Positivos (FPR)* a diferentes umbrales de decisión.

Sus componentes son:

TPR o Tasa de verdaderos positivos

Mide la sensibilidad del modelo. Se calcula como:

$$TPR = \frac{TP}{TP + FN}$$

FPR o Tasa de falsos positivos

Mide los errores al clasificar los negativos. Se calcula como:

$$FPR = \frac{FP}{FP + TN}$$

AUC o Área bajo la curva

El propósito de la AUC es dar un valor que resuma el rendimiento de la curva ROC. Este valor representa la probabilidad de que el clasificador asigne una puntuación más alta a una muestra positiva que a una negativa. Un AUC de 1 indica un clasificador perfecto, mientras que un AUC de 0.5 indica un clasificador aleatorio.

Su interpretación es:

- **AUC = 1:** Clasificador perfecto.
- **0.5 < AUC < 1:** Buen rendimiento del clasificador. Cuanto más cerca de 1, mejor.
- **AUC = 0.5:** Clasificador aleatorio.
- **AUC < 0.5:** Peor que un clasificador aleatorio (inversión de clases).

Se calcula como:

$$AUC = \int_0^1 TRP(FPR)\, d(FPR)$$

En el caso específico de este problema que consta de 4 clases (*clasificación multiclase*) se tendrían que generar 4 curvas ROC One-vs-Rest que representen:

1. Clase P1 vs. No P1 (P2 + P3 + P4).
2. Clase P2 vs. No P2 (P1 + P3 + P4).
3. Clase P3 vs. No P3 (P1 + P2 + P4).
4. Clase P4 vs. No P4 (P1 + P2 + P3).

Luego, se tendrían que calcular las AUC_{Pk} para cada una de estas curvas y tomar un promedio para tener una métrica general del rendimiento. Que en el caso específico de este problema serían:

$$AUC_{prom} = \frac{AUC_{P1} + AUC_{P2} + AUC_{P3} + AUC_{P4}}{4}$$

RESULTADOS

Descripción de la base de datos

EDA

Todos los datos utilizados para este problema se extrajo de la bases de datos en Kaggle, se puede acceder a través del siguiente enlace: Credit Risk Modeling||Classification problem📣📣📣

Ahora pasaremos explicar dicha base datos:

Estos datos se refieren al riesgo crediticio, que implica evaluar la probabilidad de que un prestatario incumpla con un préstamo u obligación crediticia. El conjunto de datos incluye una variedad de variables que se utilizan normalmente en la evaluación del riesgo de crédito.

Para atacar este problema tendremos dos bases de datos:

1. El primer conjunto de datos que tenemos es el conjunto interno del banco, credit_risk_file_1.csv, el segundo conjunto, credit_risk_file_2.csv, es propocionado por **Credit Information Bureau (India) Limited (CIBIL)**, que es una de las principales agencias de crédito en India. Proporciona informes de crédito y puntajes crediticios basados en la información recopilada de varios prestamistas y entidades financieras.

2. La variable objetivo es Approved_Flag, que contiene 4 categorías ['P1', 'P2', 'P3', 'P4'], segregando al cliente en clases para dar crédito. P1 es la categoría en la que el banco puede dar crédito fácilmente a ese cliente, mientras que P4 es la categoría en la que no es buena idea dar crédito a ese cliente, ya que puede aumentar las cuentas de morosidad (activos improductivos) del banco.

3. Ahora, la siguiente tabla contiene todas la variables para nuestro primer conjunto de datos el proporcionado por el banco, es el siguiente:

Nombre de la variable	Descripción	Tipo de dato
Total_TL	Total de líneas/cuentas comerciales en Bureau	Entero
Tot_Closed_TL	Total de líneas/cuentas comerciales cerradas	Entero
Tot_Active_TL	Total de cuentas activas	Entero

Total_TL_opened_L6M	Total de cuentas abiertas en los últimos 6 meses	Entero
Tot_TL_closed_L6M	Total de cuentas cerradas en los últimos 6 meses	Entero
pct_tl_open_L6M	Porcentaje de cuentas abiertas en los últimos 6 meses	Flotante
pct_tl_closed_L6M	Porcentaje de cuentas cerradas en los últimos 6 meses	Flotante
pct_active_tl	Porcentaje de cuentas activas	Flotante
pct_closed_tl	Porcentaje de cuentas cerradas	Flotante
Total_TL_opened_L12M	Total de cuentas abiertas en los últimos 12 meses	Entero
Tot_TL_closed_L12M	Total de cuentas cerradas en los últimos 12 meses	Entero
pct_tl_open_L12M	Porcentaje de cuentas abiertas en los últimos 12 meses	Flotante
pct_tl_closed_L12M	Porcentaje de cuentas cerradas en los últimos 12 meses	Flotante
Tot_Missed_Pmnt	Total de pagos atrasados	Entero
Auto_TL	Recuento de cuentas de automóviles	Entero
CC_TL	Recuento de cuentas de tarjetas de crédito	Entero
Consumer_TL	Recuento de cuentas de bienes de consumo	Entero
Gold_TL	Recuento de cuentas de préstamos de oro	Entero
Home_TL	Recuento de cuentas de préstamos para vivienda	Entero
PL_TL	Recuento de cuentas de préstamos personales	Entero
Secured_TL	Recuento de cuentas aseguradas	Entero
Unsecured_TL	Recuento de cuentas no aseguradas	Entero
Other_TL	Recuento de otras cuentas	Entero
Age_Oldest_TL	Antigüedad de la cuenta abierta más antigua	Entero
Age_Newest_TL	Antigüedad de la cuenta abierta más reciente	Entero

4. Conjunto de variables que tenemos en los datos proporcionados por el CIBIL:

Nombre de la variable	Descripción	Tipo de dato
time_since_recent_payment	Tiempo transcurrido desde el pago reciente realizado	Entero
time_since_first_deliquency	Tiempo transcurrido desde la primera morosidad (pago atrasado)	Entero
time_since_recent_deliquency	Tiempo transcurrido desde la morosidad reciente	Entero
num_times_delinquent	Número de veces moroso	Entero
max_delinquency_level	Nivel máximo de morosidad	Entero
max_recent_level_of_deliq	Nivel máximo reciente de morosidad	Entero
num_deliq_6mts	Número de veces que se ha morado en los últimos 6 meses	Entero
num_deliq_12mts	Número de veces que se ha morado en los últimos 12 meses	Entero
num_deliq_6_12mts	Número de veces morosos entre los últimos 6 y 12 meses	Entero
max_deliq_6mts	Nivel máximo de morosidad en los últimos 6 meses	Entero
max_deliq_12mts	Nivel máximo de morosidad en los últimos 12 meses	Entero
num_times_30p_dpd	Número de veces 30+ dpd	Entero
num_times_60p_dpd	Número de veces 60+ dpd	Entero
num_std	Número de pagos estándar	Entero
num_std_6mts	Número de pagos estándar en los últimos 6 meses	Entero
num_std_12mts	Número de pagos estándar en los últimos 12 meses	Entero

num_sub	Número de pagos por debajo de la norma	Entero
num_sub_6mts	Número de pagos por debajo de la norma en los últimos 6 meses	Entero
num_sub_12mts	Número de pagos por debajo de la norma en los últimos 12 meses	Entero
num_dbt	Número de pagos dudosos	Entero
num_dbt_6mts	Número de pagos dudosos en los últimos 6 meses	Entero
num_dbt_12mts	Número de pagos dudosos en los últimos 12 meses	Entero
num_lss	Número de cuentas de pérdidas	Entero
num_lss_6mts	Número de cuentas de pérdidas en los últimos 6 meses	Entero
num_lss_12mts	Número de cuentas de pérdidas en los últimos 12 meses	Entero
recent_level_of_deliq	Nivel reciente de morosidad	Entero
tot_enq	Total de consultas	Entero
CC_enq	Consultas sobre tarjetas de crédito	Entero
CC_enq_L6m	Consultas sobre tarjetas de crédito en los últimos 6 meses	Entero
CC_enq_L12m	Consultas sobre tarjetas de crédito en los últimos 12 meses	Entero
PL_enq	Consultas sobre Préstamos Personales	Entero
PL_enq_L6m	Consultas de Préstamos Personales en los últimos 6 meses	Entero
PL_enq_L12m	Consultas de préstamos personales en los últimos 12 meses	Entero

time_since_recent_enq	Tiempo transcurrido desde la última consulta	Entero
enq_L12m	Consultas en los últimos 12 meses	Entero
enq_L6m	Consultas en los últimos 6 meses	Entero
enq_L3m	Consultas en los últimos 3 meses	Entero
MARITALSTATUS	Estado civil	Cadena
EDUCATION	Nivel educativo	Cadena
AGE	Edad	Entero
GENDER	Género	Carácter
NETMONTHLYINCOME	Ingreso Mensual Neto	Entero
Time_With_Curr_Empr	Tiempo con el empleador actual	Entero
pct_of_active_TLs_ever	Porcentaje de cuentas activas de la historia	Flotante
pct_opened_TLs_L6m_of_L12m	Porcentaje de cuentas abiertas en los últimos 6 meses a 12 meses	Flotante
pct_currentBal_all_TL	Porcentaje del saldo actual de todas las cuentas	Flotante
CC_utilization	Utilización de la tarjeta de crédito	Flotante
CC_Flag	Bandera de tarjeta de crédito	Booleano
PL_utilization	Utilización de Préstamos Personales	Flotante
PL_Flag	Bandera de Préstamo Personal	Booleano
pct_PL_enq_L6m_of_L12m	Porcentaje de consultas PL en los últimos 6 meses a los últimos 12 meses	Flotante
pct_CC_enq_L6m_of_L12m	Porcentaje de consultas CC en los últimos 6 meses a los últimos 12 meses	Flotante
pct_PL_enq_L6m_of_ever	Porcentaje de consultas PL en los últimos 6 meses hasta siempre	Flotante
pct_CC_enq_L6m_of_ever	Porcentaje de consultas CC en los	Flotante

	últimos 6 meses a siempre	
max_unsec_exposure_inPct	Exposición máxima no garantizada en porcentaje	Flotante
HL_Flag	Bandera de Préstamos para Vivienda	Booleano
GL_Flag	Bandera de préstamo de oro	Booleano
last_prod_enq2	Último producto solicitado	Cadena
first_prod_enq2	Primer producto solicitado	Cadena
Credit_Score	Puntaje de crédito del solicitante	Entero
Approved_Flag	Prioridad levels (Objetivo col)	Cadena

Notas importantes de ambos conjuntos de datos:

1. Los tipos de datos presentes en el primer conjunto son: *enteros* y *flotantes*. Mientras que los tipos de datos presentes en el segundo conjunto son: *enteros*, *flotantes*, *carácter*, *cadenas* y *booleanos*.

2. La forma del conjunto de datos interno del banco es (51336, 26).

3. La forma del conjunto de datos cibil es (51336, 62).

4. La columna común en ambos conjuntos de datos es **PROSPECTID**, que es un ID único para cada cliente.

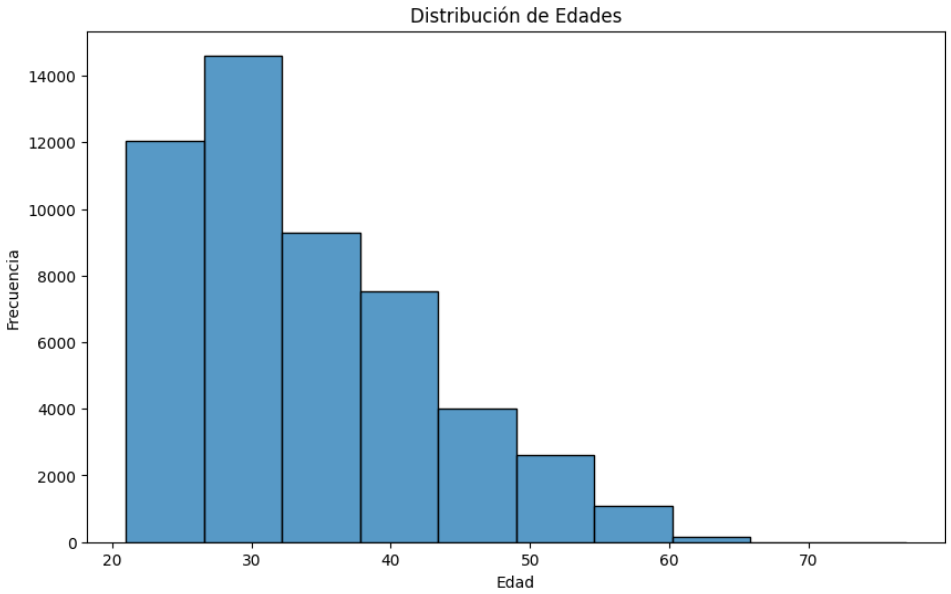
5. El valor "-99999" en ambos conjuntos de datos son valores **nulos**.

Sabiendo cuales son las variables para cada uno de las dos bases de datos con las que vamos a trabajar, pasaremos ahora a hacer EDA (Exploratory Data Analysis), para saber la distribución de los datos y poder entender un poco mas sobre la misma.

Primero vamos a ver el rango de edades de los clientes que tiene el banco:
Código:

```
# Crear el histograma simple
plt.figure(figsize=(10, 6))
sns.histplot(df_merge['AGE'], bins=10, kde=False)
plt.title('Distribución de Edades')
plt.xlabel('Edad')
plt.ylabel('Frecuencia')
plt.show()
```

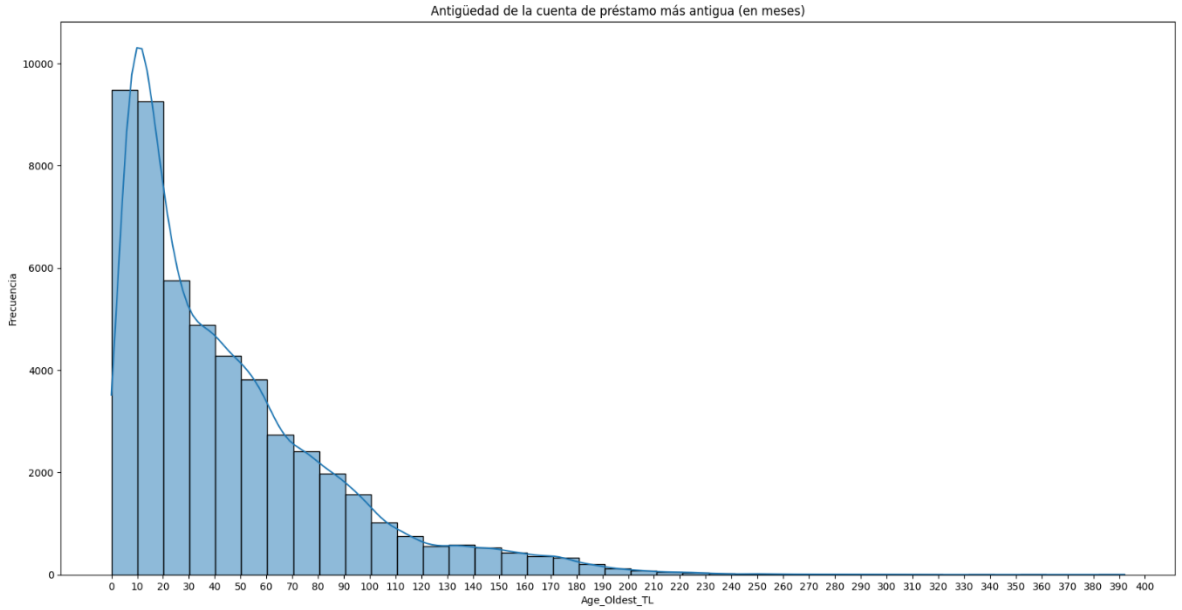
Resultado:



Por lo que podemos notar, el rango de edades oscila entre los 20 a 40 años, vemos que son mayormente adultos jóvenes los clientes que solicitan un préstamos, esto nos da una idea de que si la mayoría de los clientes estan en edad de trabajar o son jubilados, esto podría influir en nuestro modelo de clasificación.

ANTIGÜEDAD DE LA CUENTAS DE PRÉSTAMO EN MESES

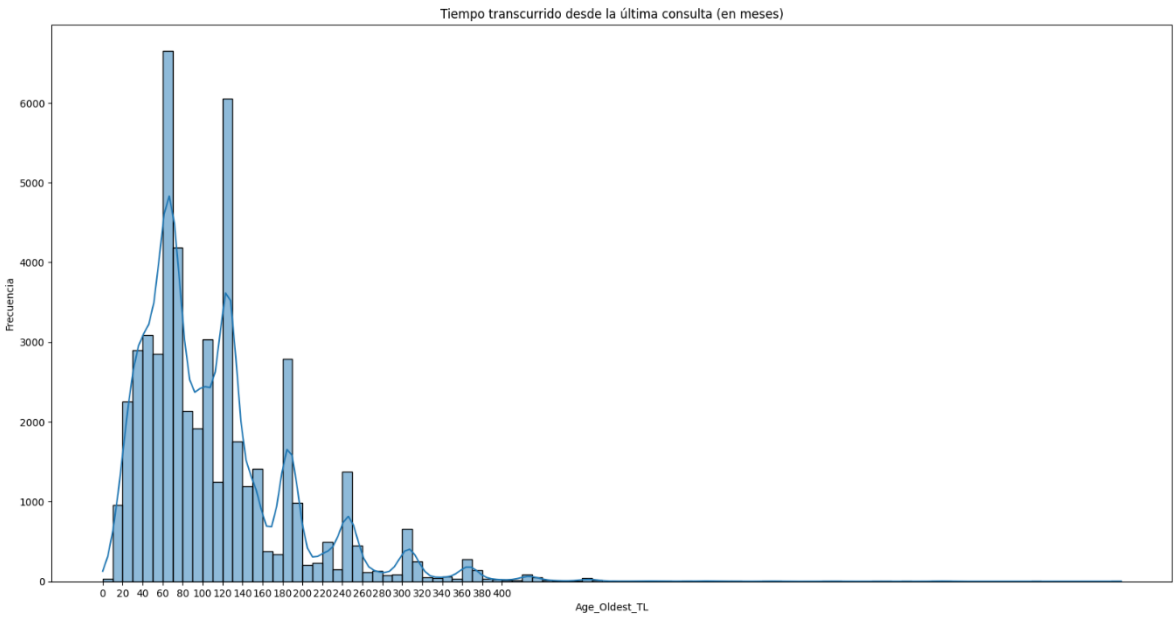
```
# Crear el histograma con saltos de 10 en 10
plt.figure(figsize=(20, 10))
sns.histplot(df_merge['Age_Oldest_TL'].dropna(), binwidth=10, kde=True)
plt.title('Antigüedad de la cuenta de préstamo más antigua (en meses)')
plt.xlabel('Age_Oldest_TL')
plt.ylabel('Frecuencia')
plt.xticks(range(0, int(df_merge['Age_Oldest_TL'].max()) + 10, 10))  # Configurar los saltos de 10 en 10 en el eje x
plt.show()
```



Por la gráfica anterior podemos notar que la antigüedad de las cuentas, en su mayoría, están entre los 0 a 20 meses, esto nos indica que las cuentas no son muy antiguas, mas si está en el rango 0 a 10, es decir que la mayoría no han cumplido el año y se puede tomar como clientes recientes.

TIEMPO TRANSCURRIDO DESDE LA ÚLTIMA CONSULTA (MESES)

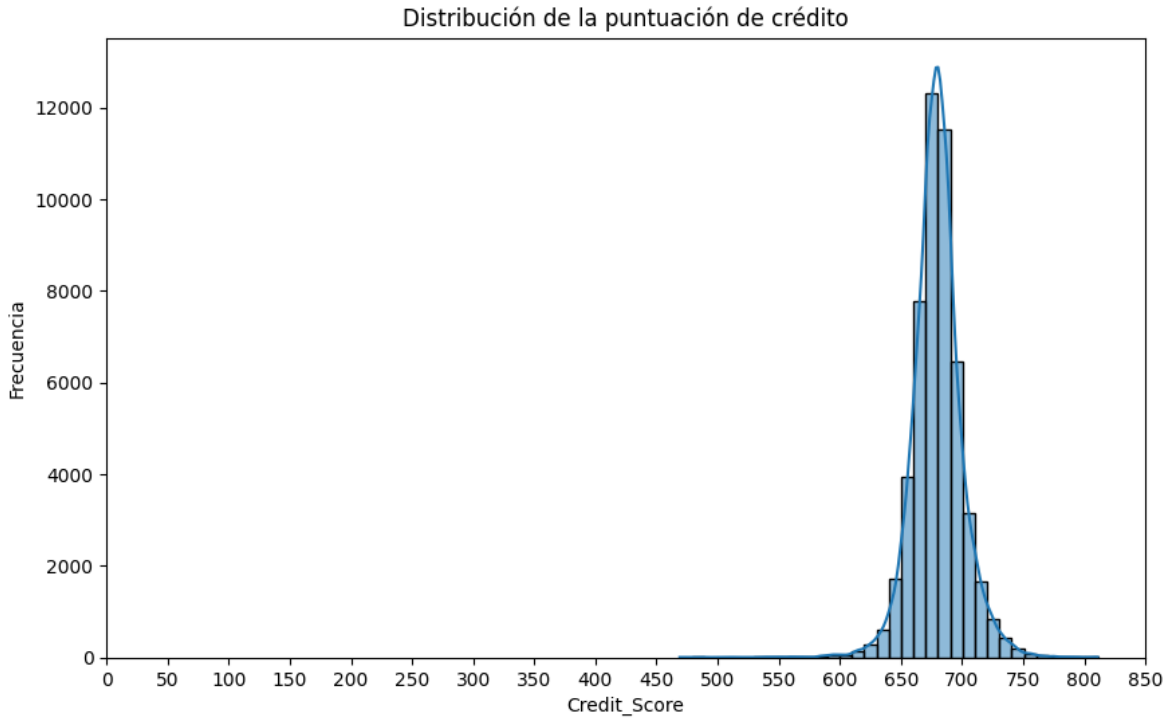
```
# Crear el histograma con saltos de 10 en 10
plt.figure(figsize=(20, 10))
sns.histplot(df_merge['Time_With_Curr_Empr'].dropna(), binwidth=10, kde=True)
plt.title('Tiempo transcurrido desde la última consulta (en meses)')
plt.xlabel('Age_Oldest_TL')
plt.ylabel('Frecuencia')
plt.xticks(range(0, int(df_merge['Age_Oldest_TL'].max()) + 20, 20))  # Configurar los saltos de 20 en 20 en el eje x
plt.show()
```



Es notable ver que las cuentas han tenido un tiempo considerable de chequeo, esto nos indica que no está el banco checando constantemente las cuentas de los clientes, omitir esto puede hacer que haya muchos clientes que no han pagado en mucho tiempo y tenga una fuga de dinero, o caso contrario, puede ser que hayan clientes que ya están al día con sus pagos y aun no se han reflejado en la base de datos y esto pueda ocasionar un sesgo.

DISTRIBUCIÓN DE LA PUNTUACIÓN DE CRÉDITO

```
# Reemplazar valores inadecuados con NaN
df_merge['Credit_Score'].replace(-99999, np.nan, inplace=True)

# Convertir la columna a tipo numérico (esto convertirá cualquier valor no numérico en NaN)
df_merge['Credit_Score'] = pd.to_numeric(df_merge['Credit_Score'], errors='coerce')

# Crear el histograma con saltos de 10 en 10
plt.figure(figsize=(10, 6))
sns.histplot(df_merge['Credit_Score'].dropna(), binwidth=10, kde=True)
plt.title('Distribución de la puntuación de crédito')
plt.xlabel('Credit_Score')
plt.ylabel('Frecuencia')
plt.xticks(range(0, int(df_merge['Credit_Score'].max()) + 50, 50))  # Configurar los saltos de 50 en 50 en el eje x
plt.show()
```

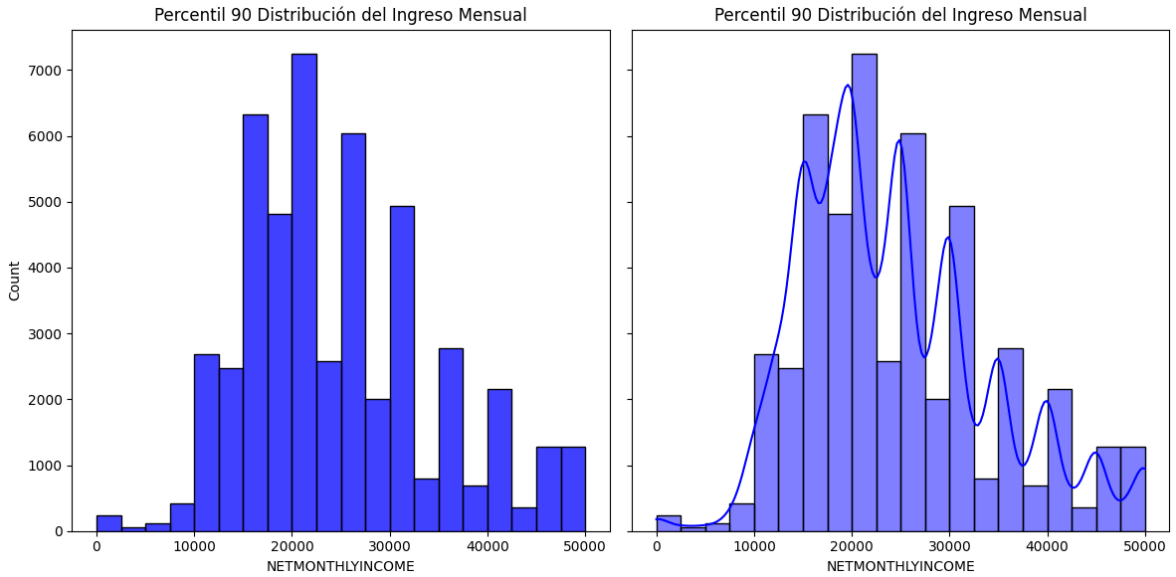


La mayor parte de la distribución de datos en la columna de puntaje de crédito se extiende entre 660 y 700, que se incluyen en la categoría P2 y es por eso que la mayoría de la categoría en la columna de destino es solo la categoría P2.

PERCENTIL 90 DISTRIBUCIÓN DEL INGRESO MENSUAL

Usamos el percentil 90 para ayudarnos a entender mejor la dispersión y la variabilidad de los datos. Por ejemplo, si el 90% de tus datos se agrupan en un rango estrecho, mientras que el 10% superior muestra una dispersión amplia, esto indica que hay una variabilidad significativa en los valores más altos.

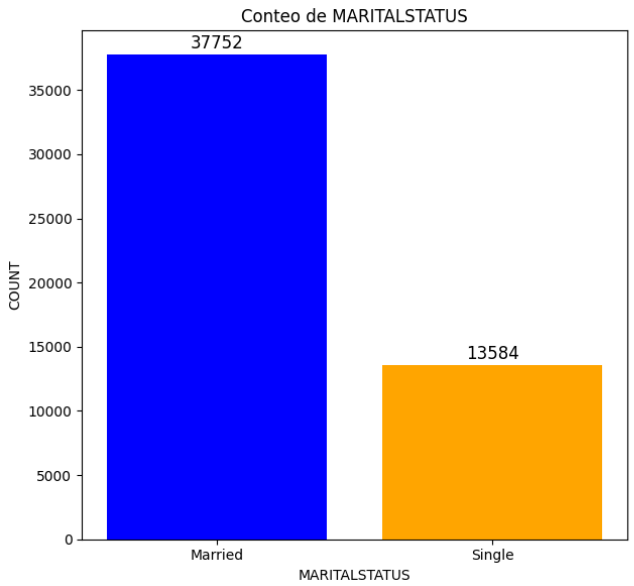
```python
# Eliminar valores extremos usando el percentil 95
upper_limit = np.percentile(df_merge['NETMONTHLYINCOME'], 95)
filtered_data = df_merge[df_merge['NETMONTHLYINCOME'] <= upper_limit]

# Configurar el tamaño de la figura
fig, axes = plt.subplots(1, 2, figsize=(12, 6), sharey=True)

# Histograma simple (izquierda)
sns.histplot(filtered_data['NETMONTHLYINCOME'], bins=20, kde=False, ax=axes[0], color='blue')
axes[0].set_title('Percentil 90 Distribución del Ingreso Mensual')
axes[0].set_xlabel('NETMONTHLYINCOME')
axes[0].set_ylabel('Count')

# Histograma con KDE (derecha)
sns.histplot(filtered_data['NETMONTHLYINCOME'], bins=20, kde=True, ax=axes[1], color='blue')
axes[1].set_title('Percentil 90 Distribución del Ingreso Mensual')
axes[1].set_xlabel('NETMONTHLYINCOME')

# Ajustar el diseño
plt.tight_layout()
plt.show()
```

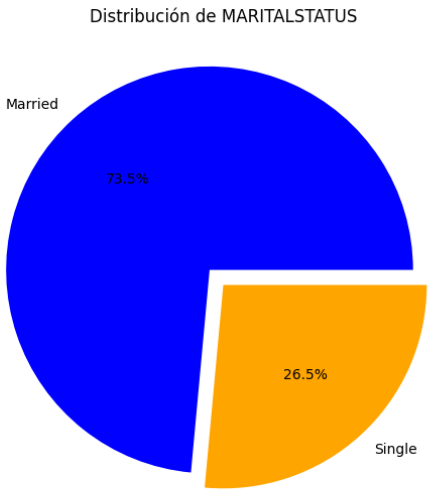


Esta columna ilustra que los ingresos salariales de la mayoría de las personas se encuentran entre 20 mil y 35 mil. Se puede observar que el banco se dirige principalmente a aquellas personas cuyos ingresos son inferiores a $ 50,000 por mes

GRÁFICO DE DISTRIBUCIÓN DEL ESTADO CIVIL

```python
# Contar la cantidad de ocurrencias por categoría en la columna MARITALSTATUS
status_counts = df_merge['MARITALSTATUS'].value_counts()
categories = status_counts.index
counts = status_counts.values

# Configurar la figura con subgráficos
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Gráfico de barras (izquierda)
axes[0].bar(categories, counts, color=['blue', 'orange'])
for i, count in enumerate(counts):
    axes[0].text(i, count + 500, str(count), ha='center', fontsize=12, color='black')
axes[0].set_title('Conteo de MARITALSTATUS')
axes[0].set_xlabel('MARITALSTATUS')
axes[0].set_ylabel('COUNT')

# Gráfico de pastel (derecha)
axes[1].pie(counts, labels=categories, autopct='%1.1f%%', colors=['blue', 'orange'], explode=(0.1, 0))
axes[1].set_title('Distribución de MARITALSTATUS')

# Ajustar el diseño
plt.tight_layout()
plt.show()
```

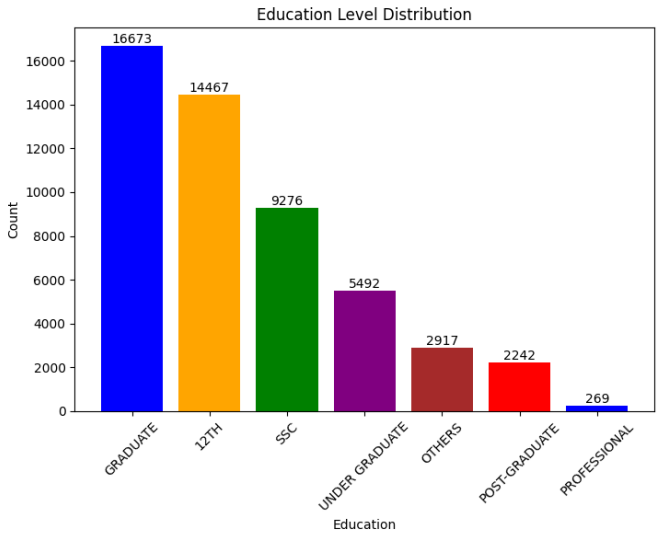


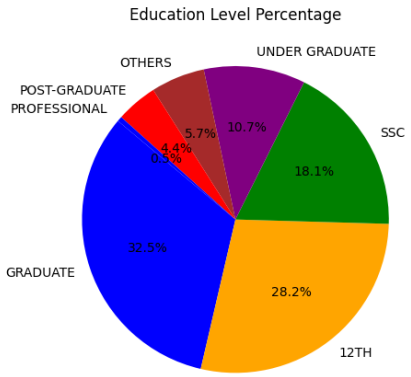


Esta columna indica que el 73.5% de las personas que solicitan el préstamo están casadas. Y esto es relevante ya que al contar con familia es mayor probabilidad de que paguen.

GRAFICO DE DISTRIBUCIÓN DE LA EDUCACIÓN

```python
#Contar las ocurrencias por categoría en la columna EDUCATION
education_counts = df_merge['EDUCATION'].value_counts()

# Configuración del gráfico de barras
plt.figure(figsize=(14, 6))

# Gráfico de barras
plt.subplot(1, 2, 1)
bars = plt.bar(education_counts.index, education_counts.values, color=['blue', 'orange', 'green', 'purple', 'brown', 'red'])
plt.title('Education Level Distribution')
plt.xlabel('Education')
plt.ylabel('Count')
plt.xticks(rotation=45)

# Agregar etiquetas con la cantidad encima de las barras
for bar in bars:
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), str(bar.get_height()),
             ha='center', va='bottom', fontsize=10)

# Gráfico circular
plt.subplot(1, 2, 2)
education_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140, colors=['blue', 'orange', 'green', 'purple', 'brown', 'red'])
plt.title('Education Level Percentage')
plt.ylabel('')  # Eliminar etiqueta del eje Y

# Mostrar las gráficas
plt.tight_layout()
plt.show()
```

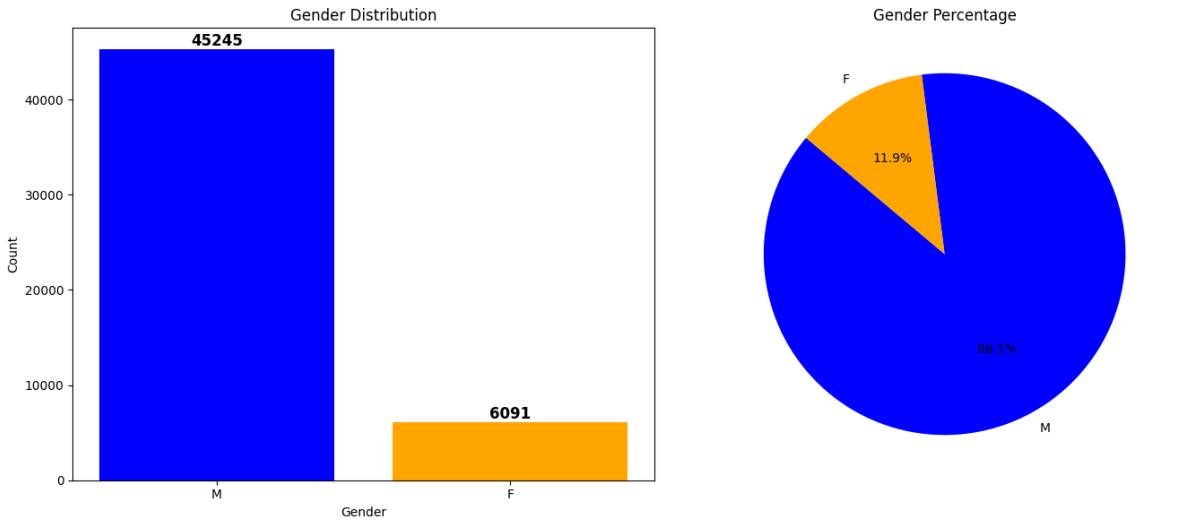




La población de graduados y 12º paso contribuye significativamente al conjunto de datos.

GRÁFICO DE DISTRIBUCIÓN DE GÉNERO

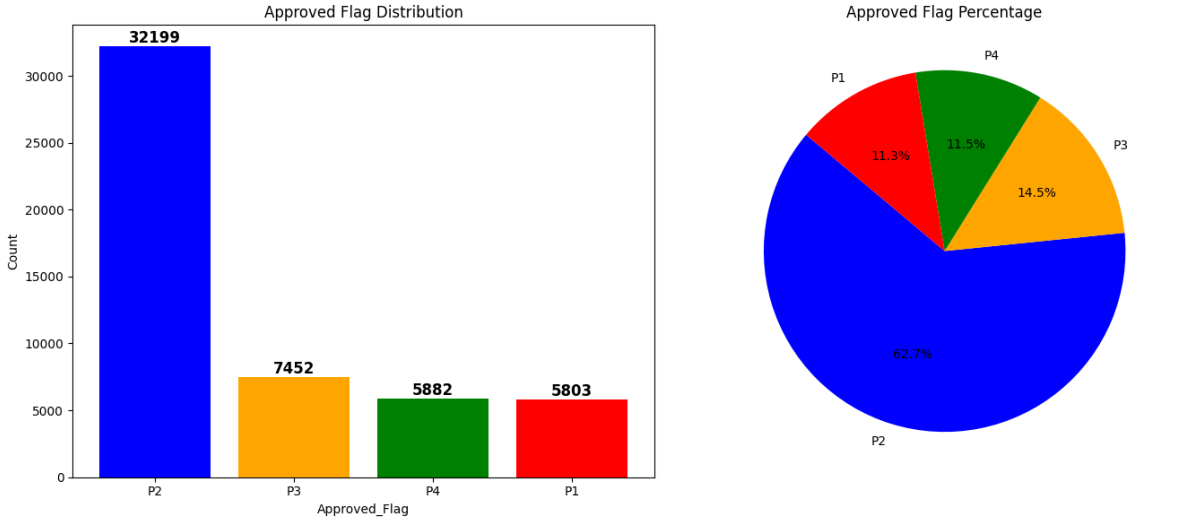
```
# Contar las ocurrencias por categoría en la columna GENDER
gender_counts = df_merge['GENDER'].value_counts()

# Configuración del gráfico de barras
plt.figure(figsize=(14, 6))

# Gráfico de barras
plt.subplot(1, 2, 1)
bars = plt.bar(gender_counts.index, gender_counts.values, color=['blue', 'orange'])
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.xticks(rotation=0)

# Agregar etiquetas con la cantidad encima de las barras
for bar in bars:
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), str(bar.get_height()),
             ha='center', va='bottom', fontsize=12, fontweight='bold')

# Gráfico circular
plt.subplot(1, 2, 2)
gender_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140, colors=['blue', 'orange'])
plt.title('Gender Percentage')
plt.ylabel('')  # Eliminar etiqueta del eje Y

# Mostrar las gráficas
plt.tight_layout()
plt.show()
```



En cuanto al género, el conjunto de datos muestra que el 88,1% de los solicitantes de préstamos son hombres, o podemos decir que el banco se dirige más a los candidatos masculinos que a los femeninos.

DISTRIBUCIÓN DE CATEGORÍAS DE VARIABLES OBJETIVO

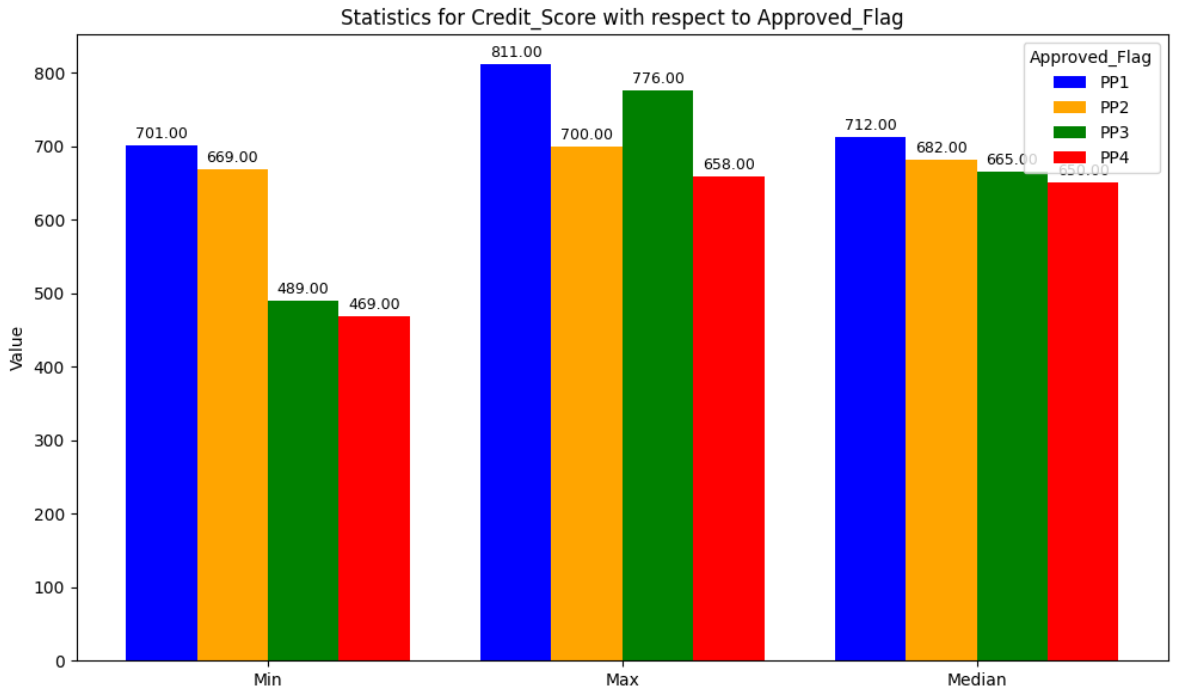
```python
# Contar las ocurrencias por categoría en la columna Approved_Flag
approved_flag_counts = df_merge['Approved_Flag'].value_counts()

# Configuración del gráfico de barras
plt.figure(figsize=(14, 6))

# Gráfico de barras
plt.subplot(1, 2, 1)
bars = plt.bar(approved_flag_counts.index, approved_flag_counts.values, color=['blue', 'orange', 'green', 'red'])
plt.title('Approved Flag Distribution')
plt.xlabel('Approved_Flag')
plt.ylabel('Count')
plt.xticks(rotation=0)

# Agregar etiquetas con la cantidad encima de las barras
for bar in bars:
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), str(bar.get_height()),
             ha='center', va='bottom', fontsize=12, fontweight='bold')

# Gráfico circular
plt.subplot(1, 2, 2)
approved_flag_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140, colors=['blue', 'orange', 'green', 'red'])
plt.title('Approved Flag Percentage')
plt.ylabel('')  # Eliminar etiqueta del eje Y

# Mostrar las gráficas
plt.tight_layout()
plt.show()
```



El 62.7% de las personas en el conjunto de datos pertenecen a la categoría P2 para la aprobación de préstamos. Siendo esta la que nos puede llegar a afectar al momento de entrenar nuestro modelo.

RELACIÓN DE LA VARIABLE TARGET CON OTRAS CARACTERÍSTICAS

```python
# Calcular estadísticas agrupadas por Approved_Flag
stats = df_merge.groupby('Approved_Flag')['Credit_Score'].agg(['min', 'max', 'median']).reset_index()

# Preparar los datos para la gráfica
labels = ['Min', 'Max', 'Median']  # Estadísticas
x = range(len(labels))  # Posiciones para las barras
colors = ['blue', 'orange', 'green', 'red']  # Colores para cada grupo

# Crear el gráfico de barras agrupadas
fig, ax = plt.subplots(figsize=(10, 6))

for i, flag in enumerate(stats['Approved_Flag'].unique()):
    values = stats.loc[stats['Approved_Flag'] == flag, ['min', 'max', 'median']].values[0]
    # Dibujar barras
    bars = ax.bar([p + i * 0.2 for p in x], values, width=0.2, label=f'P{flag}', color=colors[i])
    # Añadir valores encima de las barras
    for bar in bars:
        height = bar.get_height()
        ax.text(bar.get_x() + bar.get_width() / 2.0, height + 5, f'{height:.2f}', ha='center', va='bottom', fontsize=9)

# Personalizar el gráfico
ax.set_title('Statistics for Credit_Score with respect to Approved_Flag')
ax.set_xticks([p + 0.3 for p in x])
ax.set_xticklabels(labels)
ax.set_ylabel('Value')
ax.legend(title='Approved_Flag')

plt.tight_layout()
plt.show()
```

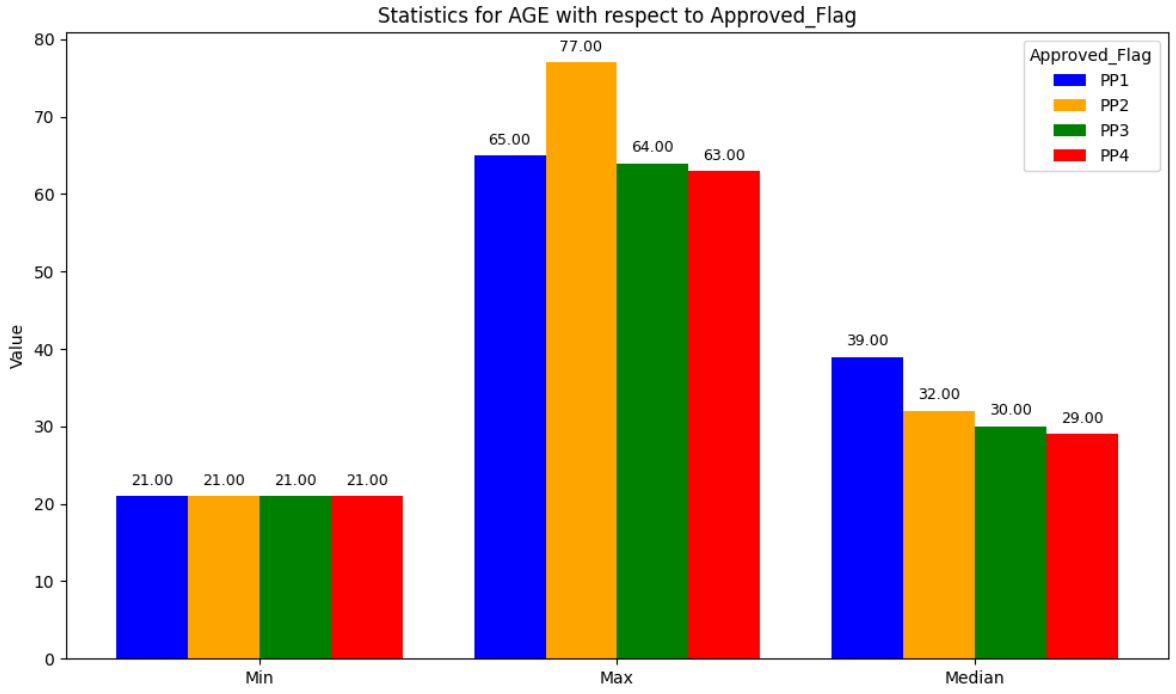


El puntaje de crédito mínimo y máximo en la categoría P3 es 489 y 776 respectivamente. Este rango indica que la categoría P3 crea mucha ambigüedad para que el modelo prediga el resultado con precisión. Para las categorías P1 y P2,

es más fácil de predecir para el modelo, ya que oscila entre (701, 811) y (669, 700), respectivamente.

VALORES MÍNIMOS, MÁXIMOS Y MEDIANOS DE EDAD PARA CADA CATEGORÍA

```python
# Calcular estadísticas agrupadas por Approved_Flag para la variable AGE
stats = df_merge.groupby('Approved_Flag')['AGE'].agg(['min', 'max', 'median']).reset_index()

# Preparar los datos para la gráfica
labels = ['Min', 'Max', 'Median']  # Estadísticas
x = range(len(labels))  # Posiciones para las barras
colors = ['blue', 'orange', 'green', 'red']  # Colores para cada grupo

# Crear el gráfico de barras agrupadas
fig, ax = plt.subplots(figsize=(10, 6))

for i, flag in enumerate(stats['Approved_Flag'].unique()):
    values = stats.loc[stats['Approved_Flag'] == flag, ['min', 'max', 'median']].values[0]
    # Dibujar barras
    bars = ax.bar([p + i * 0.2 for p in x], values, width=0.2, label=f'P{flag}', color=colors[i])
    # Añadir valores encima de las barras
    for bar in bars:
        height = bar.get_height()
        ax.text(bar.get_x() + bar.get_width() / 2.0, height + 1, f'{height:.2f}', ha='center', va='bottom', fontsize=9)

# Personalizar el gráfico
ax.set_title('Statistics for AGE with respect to Approved_Flag')
ax.set_xticks([p + 0.3 for p in x])
ax.set_xticklabels(labels)
ax.set_ylabel('Value')
ax.legend(title='Approved_Flag')

plt.tight_layout()
plt.show()
```



La edad mínima para todas las categorías es de 21 años. La edad máxima oscila entre los 65 y los 67 años. Se puede observar que para la categoría P1, la edad mediana es mayor en comparación con otras categorías y a medida que la categoría disminuye, la edad mediana también disminuye.

Observación de columnas numéricas y categóricas con respecto a la variable objetivo, es decir, Approved_Flag

1. El rango de categoría P1 es (701–811)

2. El rango de categoría P2 es (669–700)

3. El rango de categoría P3 es (489–776)

4. La categoría P3 de la variable objetivo es la categoría más ambigua. Esto se puede observar al observar el valor mínimo y máximo de la calificación crediticia para la categoría P3, que oscila entre 489 y 776, mientras que en el caso de P2, oscila entre 669 y 701.

5. Al ser P3 la categoría más ambigua, hace que el modelo disminuya su precisión a un nivel significativo.

6. La edad media de los que reciben préstamos de categoría P1 es un poco mayor que la de otras categorías. Por ejemplo, la edad mínima para la categoría P1 es de 39 años, mientras que para la categoría P2 es de 32 años, y para la categoría P3 es de 30 años. Por lo tanto, se puede observar que a medida que aumenta la edad, la aprobación del préstamo se vuelve más fácil.

Pipeline del procesamiento de datos

Para asegurar que los datos sean consistentes, bien formateados y estén correctamente escalados, con el fin de mejorar la capacidad del modelo para aprender patrones sin sesgos causados por diferencias de magnitud entre características fue necesario procesar los datos. El pipiline de este procesamiento se explica a continuación:

1) **Unión de los dos datasets:** Se unieron ambos datasets (*credit_risk_file_1.csv y credit_risk_file_2.csv*) en una relación 1-1 a través de la columna PROSPECTID.
2) **Eliminación de columnas irrelevantes**: Se eliminan columnas de cada conjunto de datos que no están presentes en la lista de columnas seleccionadas para garantizar consistencia en los datos procesados.
3) **Manejo de valores faltantes**: Se reemplazan los valores específicos marcados como *-99999* por valores *NaN* (Not a Number), que son reconocidos como datos faltantes. Las columnas con valores NaN son eliminadas posteriormente.

4) **Imputación de valores faltantes**: Los valores faltantes en las columnas numéricas se imputan utilizando la *media* de cada columna para preservar la consistencia y continuidad de los datos.
5) **Eliminación de columnas con baja varianza**: Se eliminan columnas numéricas cuya varianza sea menor a 1×10^{-6}. Ya que estas columnas no aportan información relevante para el modelo.
6) **Codificación de variables categóricas:** Las columnas categóricas (tipo string) se transforman en variables dummy mediante el método **one-hot encoding**. Esto asegura que el modelo pueda procesar datos categóricos correctamente.
7) **Separación de variables categóricas (dummy) y variables numéricas:** Una vez generadas las variables categóricas en su forma one-hot encoding, se procede a separarlas de las variables numéricas con el fin de evitar que les afecte la estandarización y posterior normalización de los datos.
8) **Estandarización:** Para las columnas numéricas se estandariza mediante se utiliza la clase StandardScaler de la biblioteca *scikit-learn* para ajustar y transformar los datos numéricos.
9) **Normalización:** La normalización implementada fue la llamada *RobustScaling* de la biblioteca de *scikit-learn* la cual utiliza el rango intercuartil (IQR) para escalar los datos, siendo robusto frente a outliers.
10) **Reconstrucción del dataset:** Se vuelven a unir las columnas numéricas estandarizadas/normalizadas y las variables dummy para formar el conjunto de datos final.
11) **Conversión de tipos:** Los conjuntos de datos se convierten al tipo *float* para garantizar compatibilidad con el modelo.
12) **Retorno de los datos procesados:** Finalmente, se retorna cada conjunto de datos.

De las **87** columnas de características presentes en la unión de ambos conjuntos de datos, después del procesamiento se el número de estas se redujo a **75**, que fueron las que finalmente pasaron al modelo para su entrenamiento.

Estas son:

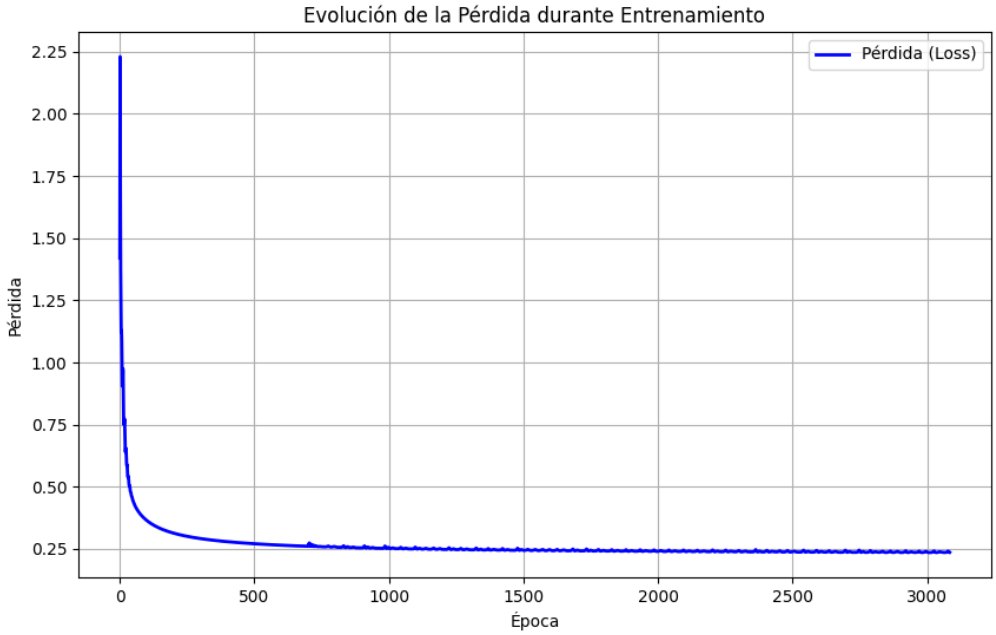
1. num_times_delinquent
2. max_recent_level_of_deliq
3. num_deliq_6mts
4. num_deliq_12mts
5. num_deliq_6_12mts
6. num_times_30p_dpd
7. num_times_60p_dpd
8. num_std
9. num_std_6mts

10. num_std_12mts
11. num_sub
12. num_sub_6mts
13. num_sub_12mts
14. num_dbt
15. num_dbt_6mts
16. num_dbt_12mts
17. num_lss
18. num_lss_6mts
19. num_lss_12mts
20. recent_level_of_deliq
21. AGE
22. NETMONTHLYINCOME
23. Time_With_Curr_Empr
24. pct_of_active_TLs_ever
25. pct_opened_TLs_L6m_of_L12m
26. CC_Flag
27. PL_Flag
28. pct_PL_enq_L6m_of_L12m
29. pct_CC_enq_L6m_of_L12m
30. pct_PL_enq_L6m_of_ever
31. pct_CC_enq_L6m_of_ever
32. HL_Flag
33. GL_Flag
34. Credit_Score
35. Total_TL
36. Tot_Closed_TL
37. Tot_Active_TL
38. Total_TL_opened_L6M
39. Tot_TL_closed_L6M
40. pct_tl_open_L6M
41. pct_tl_closed_L6M
42. pct_active_tl
43. pct_closed_tl
44. Total_TL_opened_L12M
45. Tot_TL_closed_L12M
46. pct_tl_open_L12M
47. pct_tl_closed_L12M
48. Tot_Missed_Pmnt
49. Auto_TL

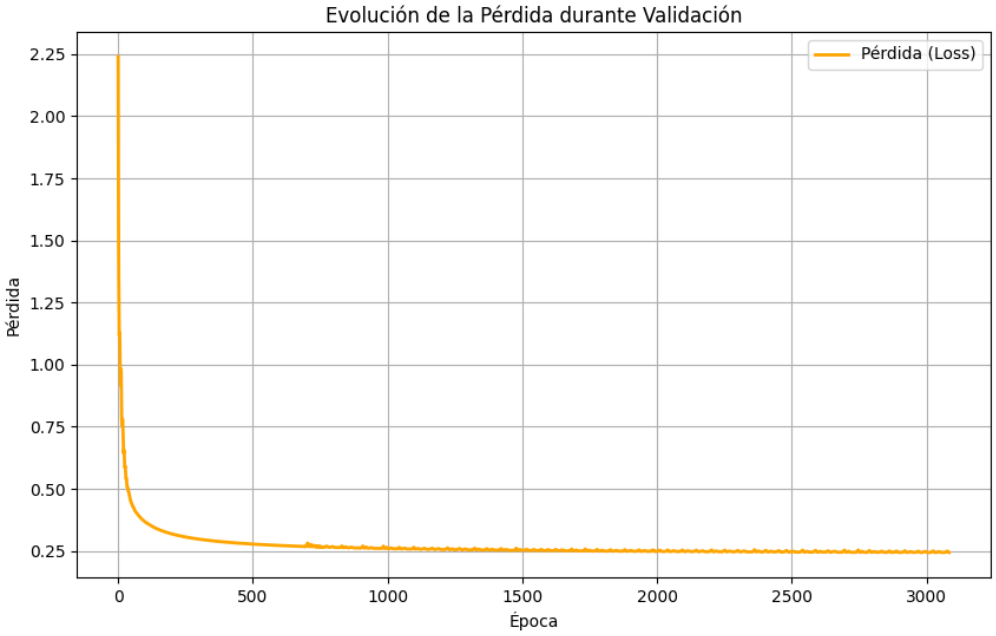
50. CC_TL
51. Consumer_TL
52. Gold_TL
53. Home_TL
54. PL_TL
55. Secured_TL
56. Unsecured_TL
57. Other_TL
58. MARITALSTATUS_Single
59. EDUCATION_GRADUATE
60. EDUCATION_OTHERS
61. EDUCATION_POST-GRADUATE
62. EDUCATION_PROFESSIONAL
63. EDUCATION_SSC
64. EDUCATION_UNDER GRADUATE
65. GENDER_M
66. last_prod_enq2_CC
67. last_prod_enq2_ConsumerLoan
68. last_prod_enq2_HL
69. last_prod_enq2_PL
70. last_prod_enq2_others
71. first_prod_enq2_CC
72. first_prod_enq2_ConsumerLoan
73. first_prod_enq2_HL
74. first_prod_enq2_PL
75. first_prod_enq2_others

Gráficas de entrenamiento

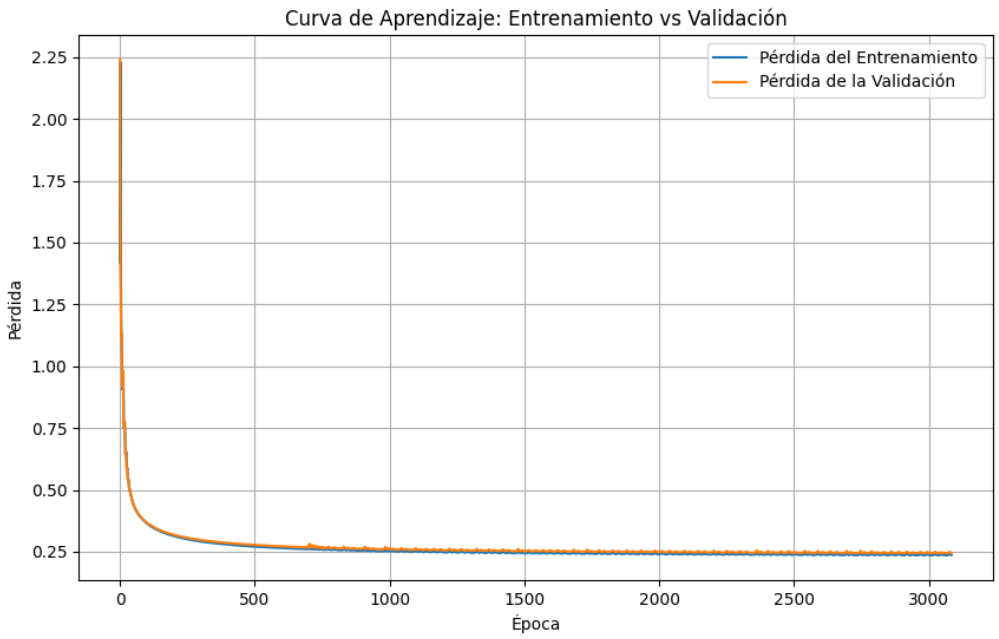
Conjunto de entrenamiento



Conjunto de validación

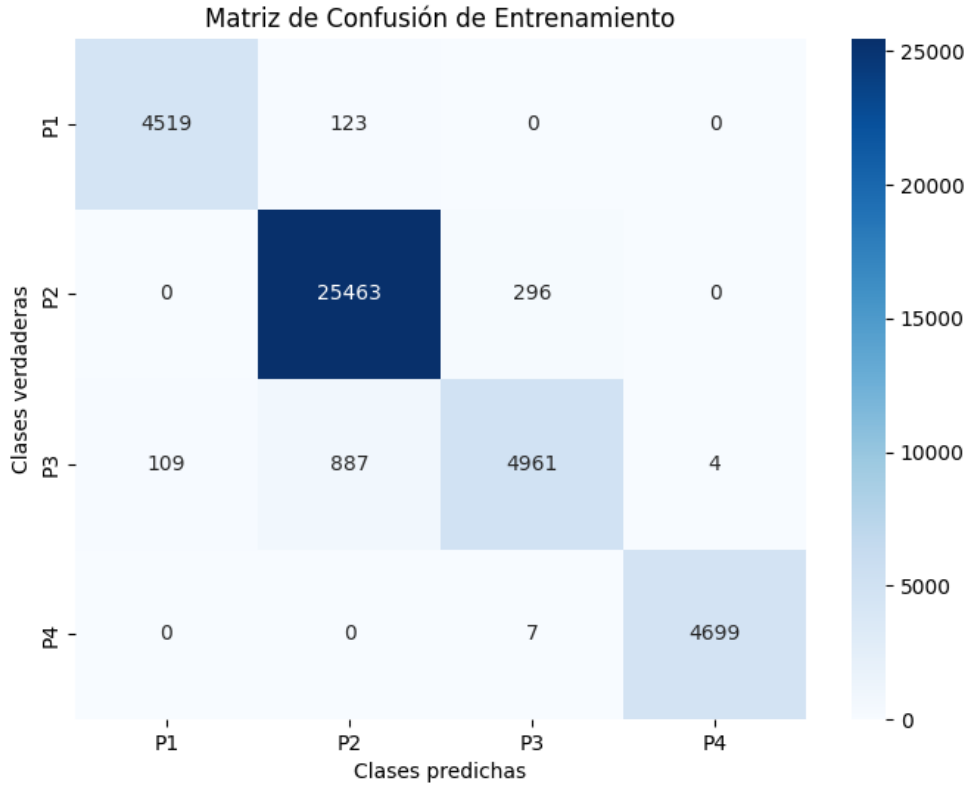


Curva de aprendizaje: Entrenamiento vs. Validación



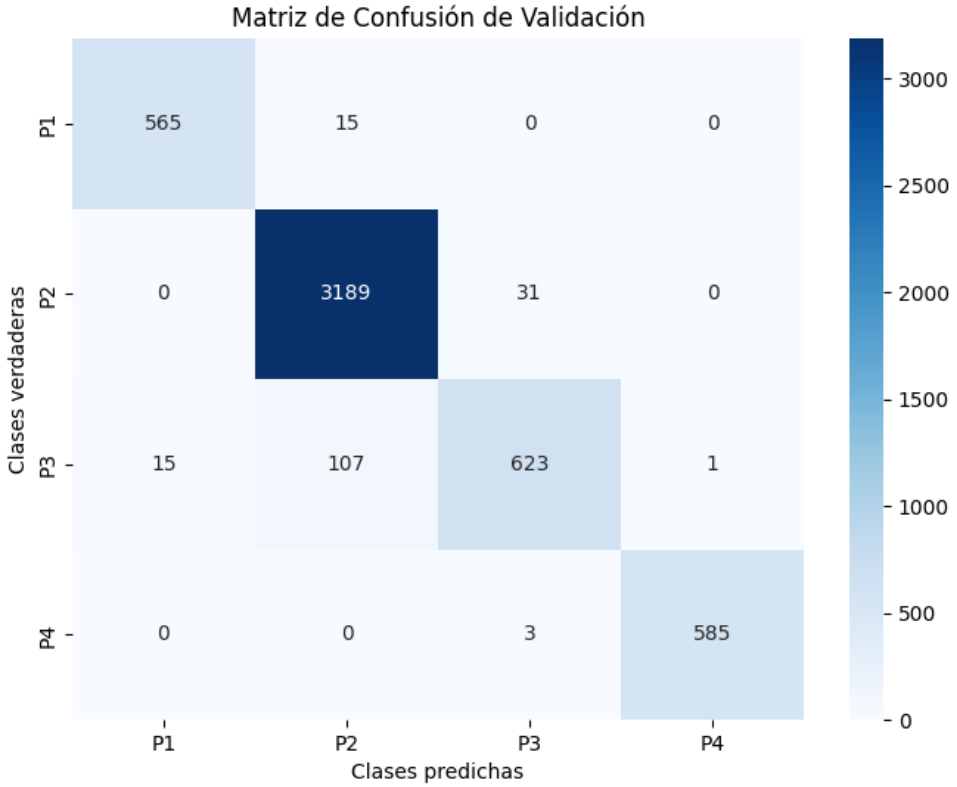
Resultados del desempeño del entrenamiento

Métricas del conjunto de entrenamiento



```
|----Matriz de Confusión----|
      P1     P2    P3    P4
P1  4519    123     0     0
P2     0  25463   296     0
P3   109    887  4961     4
P4     0      0     7  4699

|----Informe de Clasificación----|
    precision    recall  f1-score  support
P1   0.976448  0.973503  0.974973   4642.0
P2   0.961848  0.988509  0.974996  25759.0
P3   0.942439  0.832243  0.883920   5961.0
P4   0.999149  0.998513  0.998831   4706.0

Train Accuracy: 96.5277%

|----Promedios----|
Macro Avg    -> Precision: 0.9700, Recall: 0.9482, F1-score: 0.9582
Weighted Avg -> Precision: 0.9650, Recall: 0.9653, F1-score: 0.9645
```

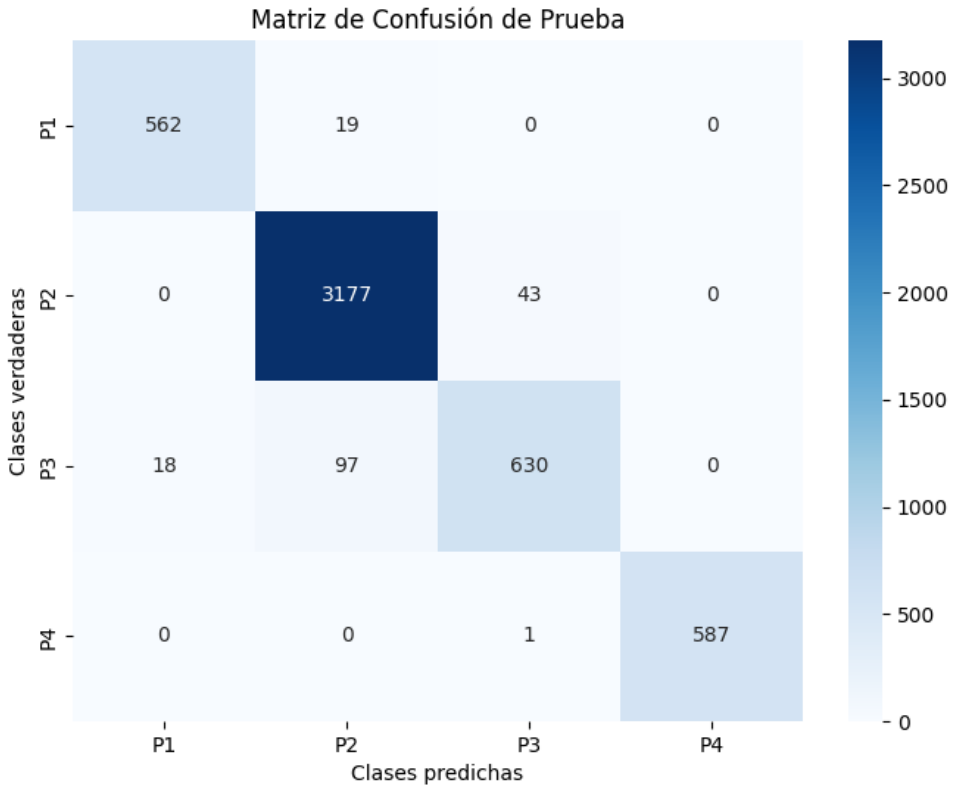
Métricas del conjunto de validación



```
|----Matriz de Confusión----|
     P1    P2   P3   P4
P1  565    15    0    0
P2    0  3189   31    0
P3   15   107  623    1
P4    0     0    3  585

|----Informe de Clasificación----|
    precision    recall  f1-score  support
P1   0.974138  0.974138  0.974138    580.0
P2   0.963153  0.990373  0.976573   3220.0
P3   0.948250  0.835121  0.888097    746.0
P4   0.998294  0.994898  0.996593    588.0

Validation Accuracy: 96.6498%

|----Promedios----|
Macro Avg    -> Precision: 0.9710, Recall: 0.9486, F1-score: 0.9589
Weighted Avg -> Precision: 0.9663, Recall: 0.9665, F1-score: 0.9657
```

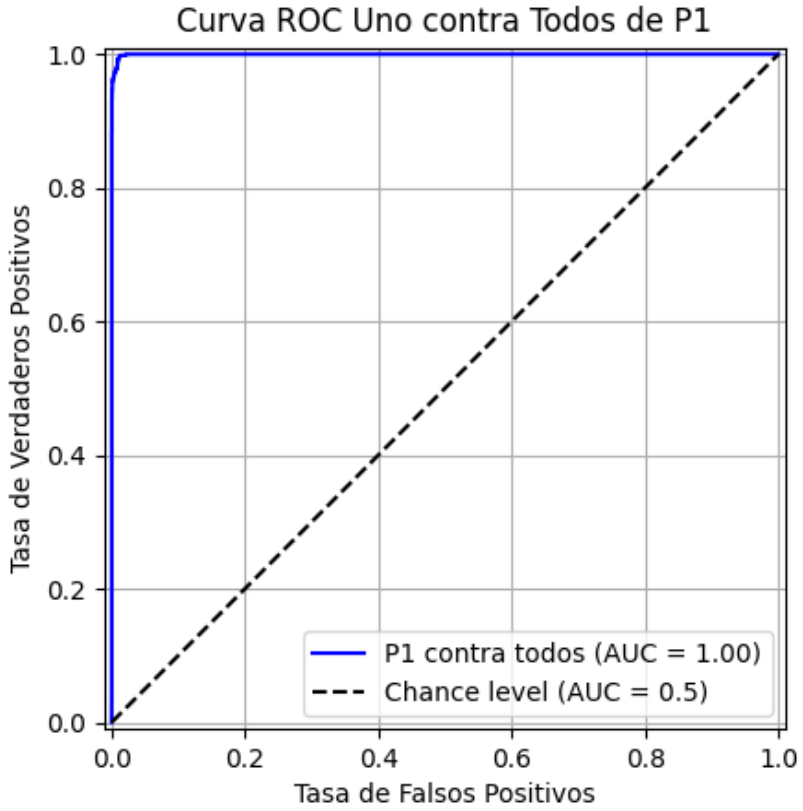
Resultados del desempeño de la prueba

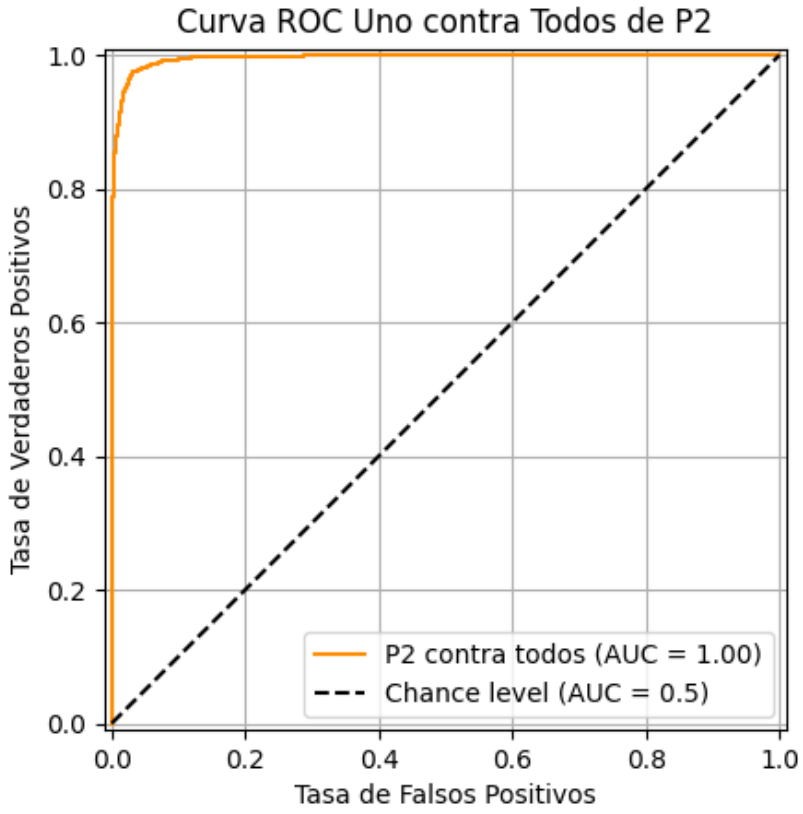
Métricas

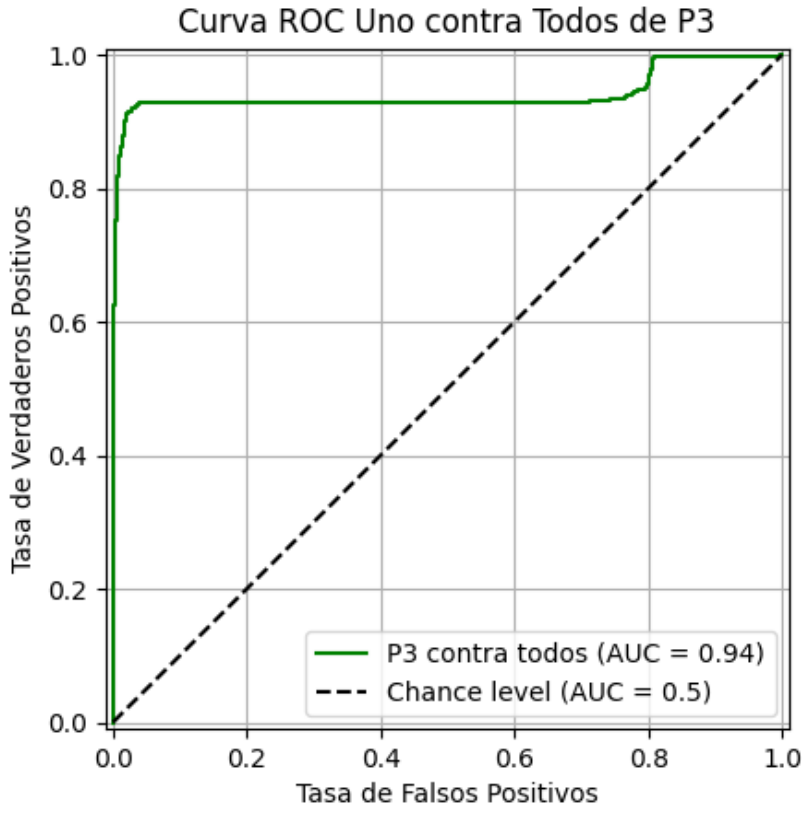


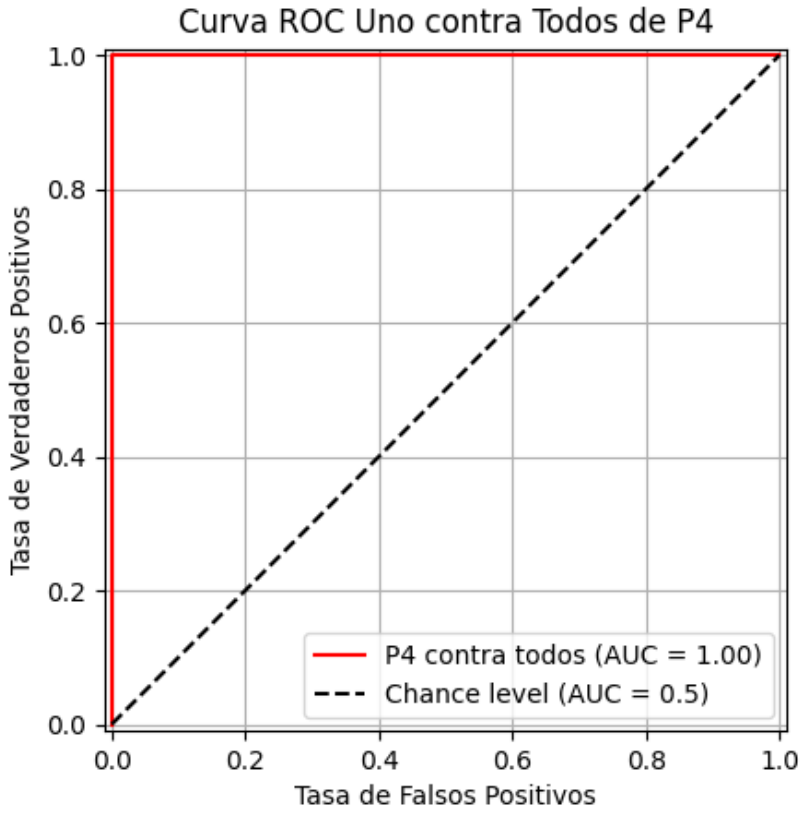
```
|----Matriz de Confusión----|
     P1    P2   P3   P4
P1  562    19    0    0
P2    0  3177   43    0
P3   18    97  630    0
P4    0     0    1  587

|----Informe de Clasificación----|
    precision    recall  f1-score  support
P1   0.968966  0.967298  0.968131    581.0
P2   0.964774  0.986646  0.975587   3220.0
P3   0.934718  0.845638  0.887949    745.0
P4   1.000000  0.998299  0.999149    588.0

Test Accuracy: 96.5329%

|----Promedios----|
Macro Avg    -> Precision: 0.9671, Recall: 0.9495, F1-score: 0.9577
Weighted Avg -> Precision: 0.9649, Recall: 0.9653, F1-score: 0.9647
```

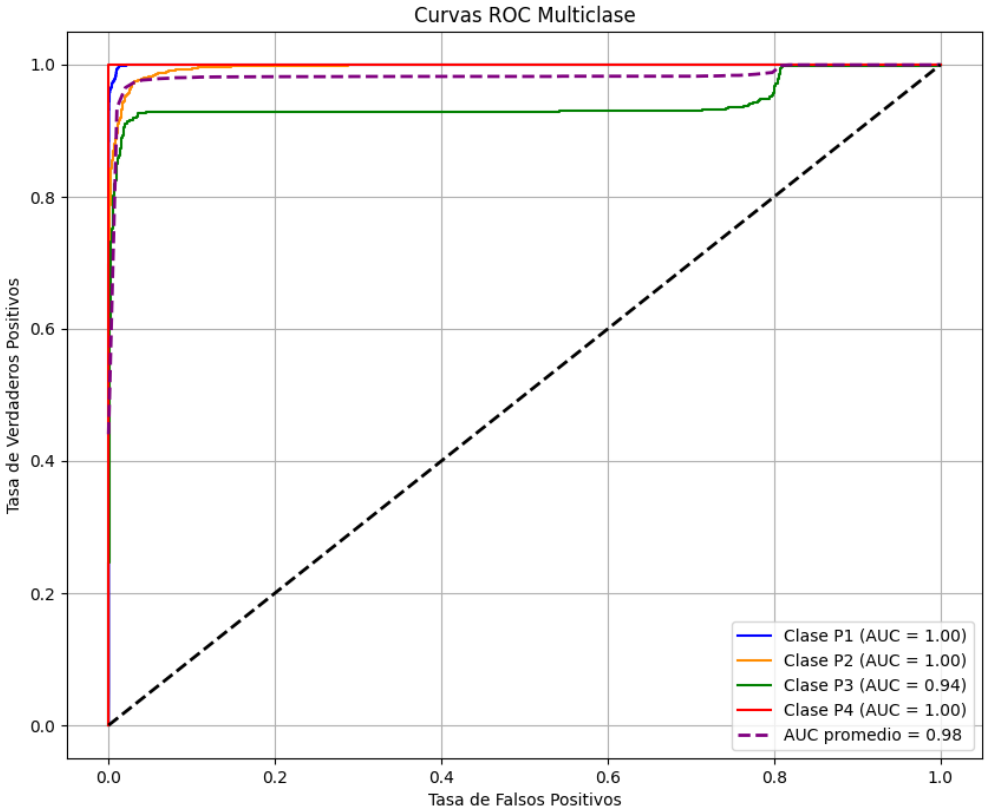
Gráficas ROC y AUCs



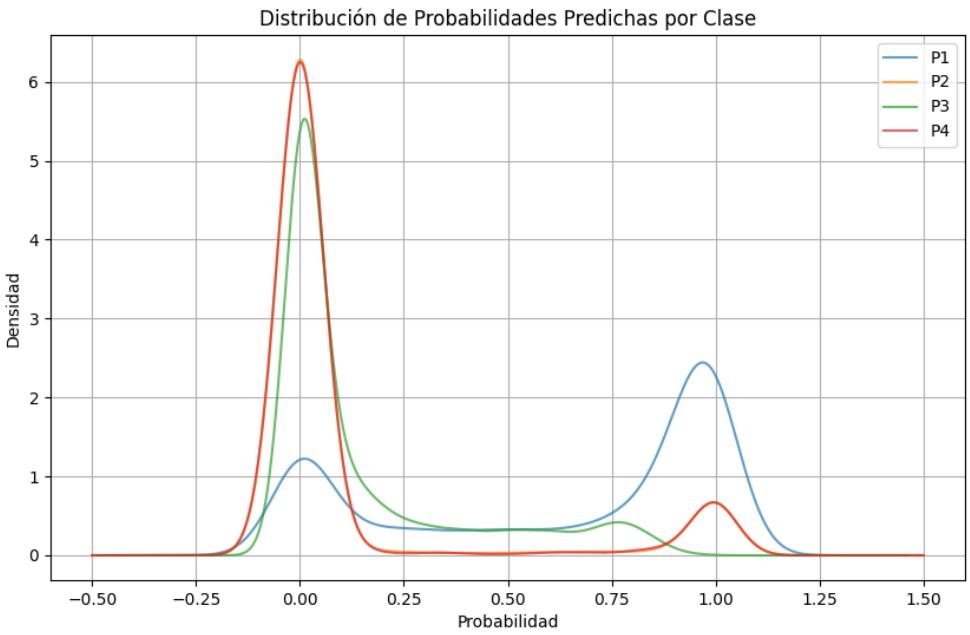

Curva ROC Uno contra Todos de P2
Tasa de Verdaderos Positivos
Tasa de Falsos Positivos
P2 contra todos (AUC = 1.00)
Chance level (AUC = 0.5)
0.0
0.2
0.4
0.6
0.8
1.0


Curva ROC Uno contra Todos de P3
Tasa de Verdaderos Positivos
Tasa de Falsos Positivos
P3 contra todos (AUC = 0.94)
Chance level (AUC = 0.5)
0.0
0.2
0.4
0.6
0.8
1.0


Curva ROC Uno contra Todos de P4
Tasa de Verdaderos Positivos
Tasa de Falsos Positivos
P4 contra todos (AUC = 1.00)
Chance level (AUC = 0.5)
0.0
0.2
0.4
0.6
0.8
1.0


Curvas ROC Multiclase
Tasa de Verdaderos Positivos
Tasa de Falsos Positivos
Clase P1 (AUC = 1.00)
Clase P2 (AUC = 1.00)
Clase P3 (AUC = 0.94)
Clase P4 (AUC = 1.00)
AUC promedio = 0.98
0.0
0.2
0.4
0.6
0.8
1.0

Gráfica de la distribución de probabilidades predichas por clase



Ejemplos de muestras mal clasificadas

```
Se encontraron un total de 178 elementos mal clasificados
```

CARACTERÍSTICAS	EJEMPLOS			
	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4
TRUE_CLASS	P3	P3	P3	P1
PREDICTED_CLASS	P2	P1	P1	P2
PROSPECTID	27183	25334	30669	39166
time_since_recent_payment	-99999	271	136	67
time_since_first_deliquency	-99999	-99999	29	35
time_since_recent_deliquency	-99999	-99999	6	2
num_times_delinquent	0	0	11	5
max_delinquency_level	-99999	-99999	75	19
max_recent_level_of_deliq	0	0	28	19
num_deliq_6mts	0	0	0	1
num_deliq_12mts	0	0	2	1
num_deliq_6_12mts	0	0	2	0
max_deliq_6mts	0	0	0	7
max_deliq_12mts	0	0	40	7
num_times_30p_dpd	0	0	5	0
num_times_60p_dpd	0	0	1	0
num_std	0	25	24	7
num_std_6mts	0	5	5	0
num_std_12mts	0	14	9	0
num_sub	0	0	0	0
num_sub_6mts	0	0	0	0
num_sub_12mts	0	0	0	0
num_dbt	0	0	0	0
num_dbt_6mts	0	0	0	0
num_dbt_12mts	0	0	0	0
num_lss	0	0	0	0
num_lss_6mts	0	0	0	0
num_lss_12mts	0	0	0	0
recent_level_of_deliq	0	0	9	7
tot_enq	1	1	4	7
CC_enq	0	0	0	0
CC_enq_L6m	0	0	0	0
CC_enq_L12m	0	0	0	0
PL_enq	0	0	0	3
PL_enq_L6m	0	0	0	3

PL_enq_L12m	0	0	0	3
time_since_recent_enq	2038	513	10	171
enq_L12m	0	0	2	4
enq_L6m	0	0	2	4
enq_L3m	0	0	2	0
MARITALSTATUS	Married	Married	Married	Married
EDUCATION	OTHERS	SSC	GRADUATE	GRADUATE
AGE	25	42	39	35
GENDER	M	M	F	M
NETMONTHLYINCOME	20000	23000	30000	20000
Time_With_Curr_Empr	106	110	47	185
pct_of_active_TLs_ever	1	0.25	0.5	0.2
pct_opened_TLs_L6m_of_L12m	1	0	1	1
pct_currentBal_all_TL	1	0.414	0.911	0.938
CC_utilization	-99999	-99999	-99999	-99999
CC_Flag	0	1	1	0
PL_utilization	-99999	0.414	-99999	0.938
PL_Flag	0	1	0	1
pct_PL_enq_L6m_of_L12m	0	0	0	1
pct_CC_enq_L6m_of_L12m	0	0	0	0
pct_PL_enq_L6m_of_ever	0	0	0	1
pct_CC_enq_L6m_of_ever	0	0	0	0
max_unsec_exposure_inPct	-99999	13.043	7.147	10
HL_Flag	0	1	1	0
GL_Flag	0	0	0	0
last_prod_enq2	others	others	ConsumerLoan	PL
first_prod_enq2	others	others	HL	AL
Credit_Score	702	702	702	702
Approved_Flag	2	2	2	1
Total_TL	1	4	10	5
Tot_Closed_TL	0	3	5	4
Tot_Active_TL	1	1	5	1
Total_TL_opened_L6M	1	0	1	1
Tot_TL_closed_L6M	0	0	1	2
pct_tl_open_L6M	1	0	0.1	0.2
pct_tl_closed_L6M	0	0	0.1	0.4
pct_active_tl	1	0.25	0.5	0.2
pct_closed_tl	0	0.75	0.5	0.8
Total_TL_opened_L12M	1	0	1	1
Tot_TL_closed_L12M	0	1	2	3
pct_tl_open_L12M	1	0	0.1	0.2

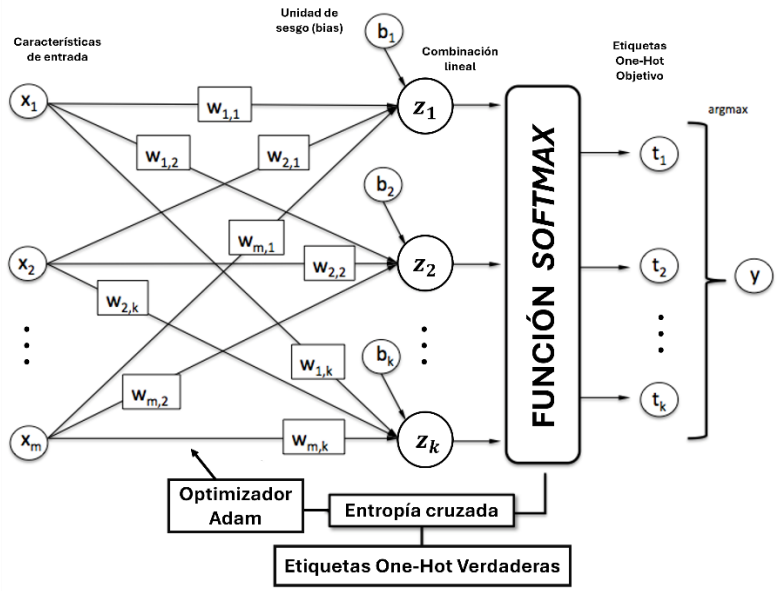
pct_tl_closed_L12M	0	0.25	0.2	0.6
Tot_Missed_Pmnt	1	0	1	0
Auto_TL	1	0	0	3
CC_TL	0	1	1	0
Consumer_TL	0	0	1	0
Gold_TL	0	1	5	0
Home_TL	0	0	0	0
PL_TL	0	1	0	1
Secured_TL	1	2	5	3
Unsecured_TL	0	2	5	2
Other_TL	0	1	3	1
Age_Oldest_TL	1	28	163	160
Age_Newest_TL	1	16	5	6

El resto de los elementos mal clasificados se encuentran dentro del archivo Elementos_Mal_Clasificados.csv que se encuentra dentro de la carpeta Datasets.

CONCLUSIÓN

Resumen del algoritmo implementado

El núcleo principal del modelo radica en una regresión logística multiclase (también denominada *softmax* o *multinomial*):



Dicho algoritmo utiliza una función de activación que es una generalización de la función *sigmoide* denominada ***softmax***:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Donde:
z_i : La puntuación lineal calculada para la clase i, es decir, $z_i = XW_i + b_i$.
e^{z_i}: La exponencial de z_i, que transforma las puntuaciones en valores positivos.
K : Número total de clases.
$\sum_{j=1}^{K} e^{z_j}$: Suma de las exponenciales de todas las puntuaciones, usada para normalizar las probabilidades.

Esta función garantiza que las probabilidades estén en el rango [0,1] y la suma de las probabilidades para todas las clases sea 1.

También, se implementó el algoritmo **Adam** (Adaptive Moment Estimation), para actualizar los parámetros del modelo, como los **pesos** y el **sesgo**, a lo largo del proceso de entrenamiento, de manera que fuese más eficiente, estable y robusto en comparación con otros métodos como el gradiente descendente clásico.

En nuestra implementación, Adam se utiliza para actualizar los *pesos* y *sesgos* (*bias*) del modelo en cada iteración de entrenamiento de la siguiente forma:

1. **Inicialización de los momentos:** Antes de comenzar el proceso de entrenamiento, se inicializan los momentos m_w, v_w para los pesos y m_b, v_b para los sesgos como vectores de ceros. Estos momentos se actualizan en cada iteración del entrenamiento de acuerdo con los gradientes calculados.
2. **Cálculo de los gradientes**: En cada época, calculamos los gradientes de la *función de pérdida* respecto a los pesos y sesgos. Estos gradientes son obtenidos usando la regla de la cadena, derivando la función de pérdida con respecto a los parámetros del modelo. Posteriormente, se calcula el gradiente de la pérdida en relación con los pesos (dw) y sesgos (db).
3. **Actualización de los momentos**: Utilizando los gradientes calculados, los momentos m_w, v_w, m_b y v_b se actualizan adaptativamente con los parámetros β_1 y β_2. Estos son los factores de decaimiento exponencial que determinan la contribución de los gradientes pasados en los momentos actuales. Los valores $\beta 1 = 0.9$ y $\beta 2 = 0.999$ son los valores predeterminados utilizados en esta implementación.
4. **Corrección de los momentos**: Debido a la inicialización de los momentos en ceros, los primeros momentos pueden ser sesgados hacia cero al principio del entrenamiento. Para corregir este sesgo, se aplicó una corrección en el cálculo de los momentos (m_w y v_w) y se normalizan por $\left(1 - \beta_1^{epoch}\right)$ y $\left(1 - \beta_2^{epoch}\right)$, respectivamente.
5. **Actualización de los parámetros**: Finalmente, los pesos y sesgos (bias) del modelo se actualizan de manera adaptativa, ajustando su valor según la tasa de aprendizaje η y la información proporcionada por los momentos corregidos, junto con un pequeño valor ε para evitar la división por cero:

$$\text{weights} = \text{weights} - \eta \cdot \frac{m_w}{\sqrt{v_w} + \varepsilon}$$

$$\text{bias} = \text{bias} - \eta \cdot \frac{m_b}{\sqrt{v_b} + \varepsilon}$$

Esta actualización adaptativa permite que las magnitudes de las actualizaciones varíen dinámicamente según la información del gradiente, lo que ayuda a acelerar el proceso de convergencia y mejora la estabilidad en el modelo.

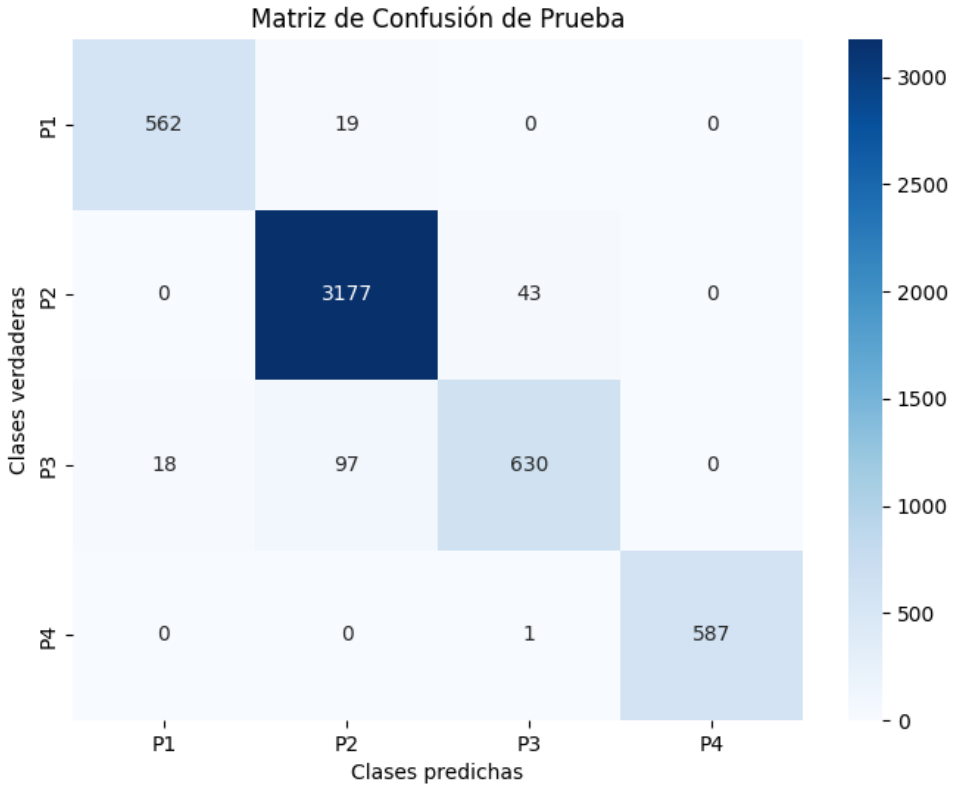
La función de costo utilizada para el modelo realizado es la entropía cruzada. Su objetivo es medir cuán bien las probabilidades predichas por un modelo se alinean con las etiquetas verdaderas. La función de la perdida de entropía cruzada es:

$$L = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} y_{ij}\log\left(\widehat{y_{ij}}\right)$$

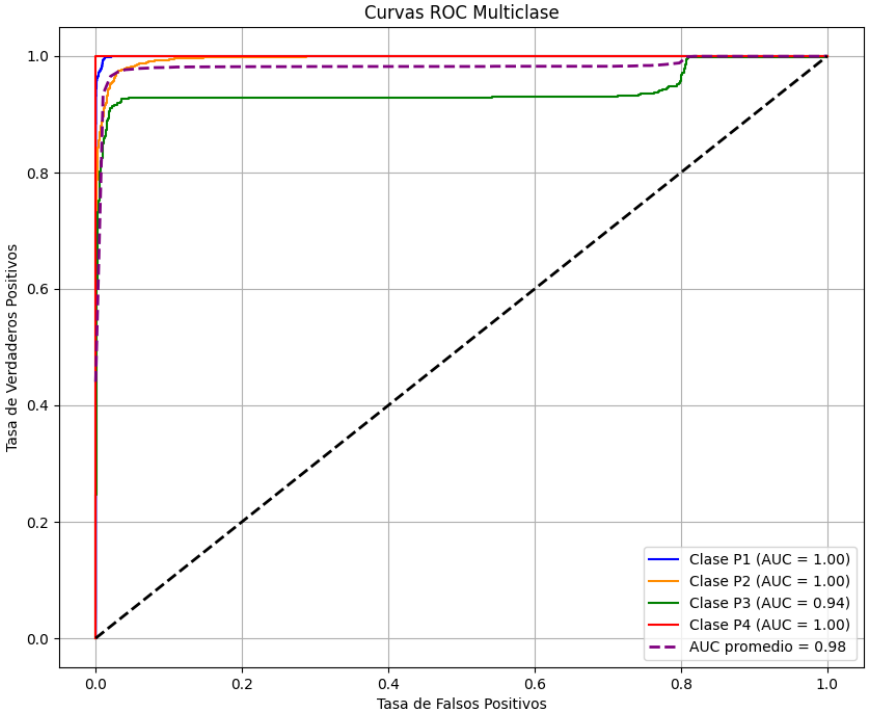
En donde:
L: Valor de la pérdida
n: Número de muestras.
y_{ij}: Etiqueta verdadera para la muestra i y clase j. Toma el valor de 1 si pertenece a esa clase y 0 si no.
$\widehat{y_{ij}}$: Probabilidad predicha para la muestra i y j.
$\log\left(\widehat{y_{ij}}\right)$: Penalización cuando $\widehat{y_{ij}}$ es baja para la clase correcta.

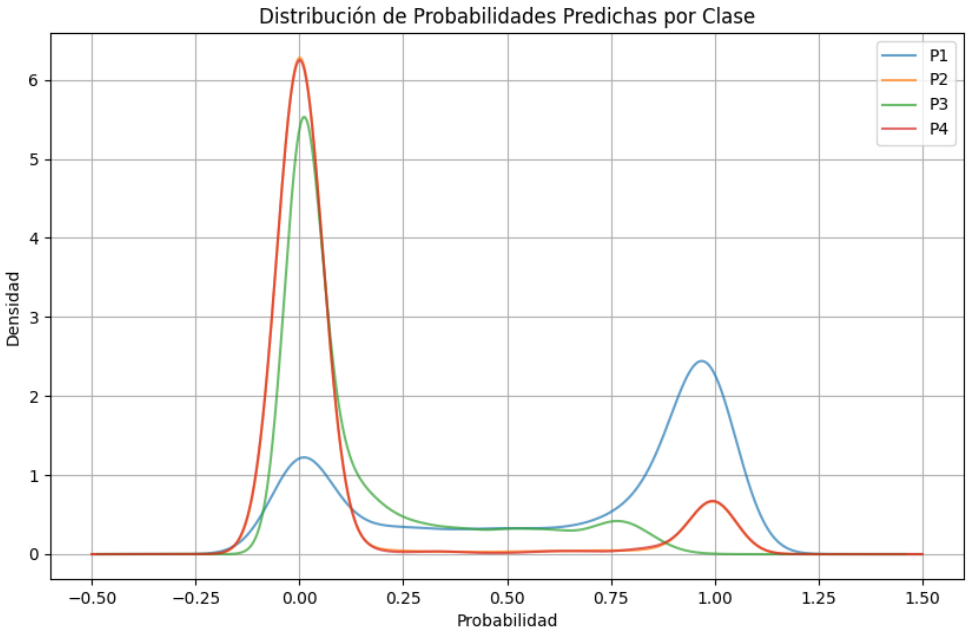
La pérdida *L* es un promedio de cuánto el modelo "se equivoca" al predecir las probabilidades de las clases correctas en todas las muestras. Para una muestra en particular, sólo importa el término $\widehat{y_{ij}}$ donde $y_{ij} = 1$ (es decir, la clase correcta), ya que $y_{ij} = 0$ anula los otros términos. Debido a que funciona bien en problemas de clasificación multiclase es que fue elegida esta función de pérdida.

Resultados del desempeño obtenidos en la prueba



```
|----Informe de Clasificación----|
    precision    recall  f1-score  support
P1   0.968966  0.967298  0.968131    581.0
P2   0.964774  0.986646  0.975587   3220.0
P3   0.934718  0.845638  0.887949    745.0
P4   1.000000  0.998299  0.999149    588.0

Test Accuracy: 96.5329%

|----Promedios----|
Macro Avg    -> Precision: 0.9671, Recall: 0.9495, F1-score: 0.9577
Weighted Avg -> Precision: 0.9649, Recall: 0.9653, F1-score: 0.9647
```





Explicación de los errores obtenidos

El principal motivo por el cual el modelo presentó errores al clasificar ciertas muestras de una clase específica, confundiéndolas con otras, radicó en varias limitaciones inherentes a los datos, las cuales son: falta de información relevante, similitud entre características de ciertas clases, y un desbalance significativo en el conjunto de datos.

En primer lugar, se identificó una falta de información en las muestras problemáticas, evidenciada por un alto número de valores iguales a -99999. Estos valores, que no aportan información útil, introdujeron ruido en el aprendizaje del modelo y afectaron negativamente su desempeño. Para mitigar este problema, los valores -99999 fueron transformados en NaN y, posteriormente, eliminados mediante la eliminación de columnas con valores nulos. Este proceso ayudó a mejorar la calidad de los datos y, por ende, el aprendizaje del modelo.

Sin embargo, también se observó una baja varianza entre las características de ciertas clases, lo que dificultó la capacidad del modelo para diferenciarlas. Particularmente, las clases P2 y P3 presentaron características menos distintivas en comparación con clases como P1 o P4, cuyas propiedades son más marcadas. La similitud entre P2 y P3 generó ambigüedad durante el proceso de clasificación, reduciendo la precisión del modelo al identificar correctamente estas clases.

Otro factor crítico fue el desbalance en el conjunto de datos, particularmente la baja representación de la clase P3. La escasez de ejemplos representativos de dicha clase limitó la capacidad del modelo para aprender patrones robustos y consistentes que permitieran clasificarla correctamente. El problema anterior se vio amplificado

cuando el desequilibrio en el conjunto de datos convergió con características poco distintivas, como fue el caso entre las clases P2 y P3, impactando directamente en el desempeño de nuestro modelo.

Por tanto, la combinación de estos factores (falta de información, baja varianza entre clases y desbalance de datos) fue determinante para que el modelo presentara errores en la clasificación. Aunque si bien, la implementación de medidas como la estandarización, normalización, tratamiento de valores nulos y el análisis de características mejoraron el aprendizaje del modelo y los resultados obtenidos, se reconoce que el desbalance del conjunto de datos sigue siendo un factor que limita su desempeño.

Como conclusión de lo anterior podemos mencionar que, la mejora en la calidad de los datos mediante técnicas de preprocesamiento fue clave para mitigar parte de los problemas observados, mejorando la capacidad del modelo para aprender patrones más representativos y diferenciables entre clases. No obstante, un análisis más profundo sobre el desbalance y la distintividad entre clases podría seguir optimizando el rendimiento del modelo.

Lo aprendido en el proyecto

La experiencia de trabajar en este proyecto nos permitió profundizar en la implementación práctica de un modelo de regresión logística multinomial. Aunque no teníamos antecedentes directos sobre este tema, como equipo nos apoyamos mutuamente para superar los retos. Cada integrante contribuyó con sus habilidades únicas, ya sea en la investigación, la programación o la validación del modelo, lo que permitió una distribución eficiente de las tareas.

En el proceso, aprendimos la importancia de la colaboración y la comunicación para resolver problemas complejos. Además, comprendimos el impacto que un modelo como este puede tener en la resolución de problemas de clasificación multiclase, lo que refuerza su utilidad en aplicaciones del mundo real. Este proyecto no solo fue un ejercicio técnico, sino también una experiencia formativa en cuanto a trabajo en equipo y adaptación frente a lo desconocido.

“El realizar la implementación del modelo de regresión logística multiclase (softmax) fue un desafío, ya que no se abordó en clase y requería investigar por nuestra cuenta. Por lo que, este proceso me permitió profundizar en conceptos importantes, como lo es el enfoque multinomial y la función softmax, que son fundamentales para resolver problemas de clasificación multiclase. Además, el desarrollar este modelo me ayudó a reforzar mis habilidades de trabajo en equipo y resolución de problemas, al mismo tiempo que adquirí nuevas herramientas y técnicas útiles para proyectos futuros. A causa de todo lo anterior es que me encuentro bastante satisfecho con

los resultados obtenidos, ya que logramos superar las dificultades iniciales y obtener una solución funcional que amplía lo aprendido en la asignatura.”

Edgar Sabido Cortés

“En este proyecto aprendí a implementar un modelo de regresión logística multiclase utilizando el enfoque softmax, lo que me permitió comprender conceptos como funciones de activación, optimización adaptativa con Adam y regularización L2. También desarrollé habilidades en análisis y preprocesamiento de datos, especialmente al manejar valores faltantes y datos desbalanceados, como en la clase P3, lo que reforzó la importancia de contar con datos bien estructurados. Trabajar en equipo fue fundamental para superar retos y coordinar esfuerzos, lo que mejoró mi capacidad de colaboración. Este proyecto me ayudó a aplicar conocimientos teóricos a un problema real y a valorar el impacto práctico de la inteligencia artificial en la evaluación del riesgo crediticio. Estoy satisfecho con los resultados obtenidos y seguro de que las habilidades adquiridas serán útiles en el futuro.”

Luis Alfredo Cota Armenta

“Este proyecto me ha permitido analizar y optimizar los conjuntos de datos de interés, asegurando que puedan ser utilizados de manera adecuada en modelos de regresión logística. A través de este proceso, he aprendido a manejar grandes volúmenes de datos, como los proporcionados por el conjunto de datos, las cuales incluyen información clave sobre el riesgo crediticio de los solicitantes de préstamos. Mi objetivo fue garantizar que los datos fueran adecuados para su uso en modelos predictivos, superando desafíos como la presencia de valores nulos y el desequilibrio en las clases de la variable objetivo "Approved_Flag".

Además, el proyecto me permitió estudiar y aplicar diversas técnicas de preprocesamiento de datos, como la limpieza, la imputación de valores faltantes y la normalización de variables. Durante este proceso, presté especial atención a la importancia de balancear los datos, dado que las clases P1, P2 y P4 eran significativamente más que las categorías P3, lo cual podría haber afectado la precisión del modelo.”

Carlos Antonio Ruiz Domínguez