



FCFM

Maestría en Ciencias de Datos

Aprendizaje Automático

Aprendizaje Automático para evaluar el Riesgo de Crédito

Edwin Martín Romero Silva

November 21, 2023

Contents

1	Introducción	3
2	Objetivo del proyecto	3
3	Base de datos	3
4	Estadística Descriptiva	4
5	Metodología	6
5.1	Variable objetivo	6
5.2	Modelo Elegido	6
5.3	Métrica de Desempeño	7
5.4	Entrenamiento y Validación	8
5.5	Diseño de Experimentos	9
5.5.1	Parte 1 Hiperparámetros más influyentes	9
5.5.2	Parte 2 Mejor combinación de hiperparámetros	11
5.5.3	Parte 3 Desempeño Promedio	12
6	Interpretación de Resultados	13
7	Conclusión	14

1 Introducción

El crédito es un servicio financiero que permite a personas físicas o morales adquirir bienes, servicios o financiar proyectos mediante la obtención de fondos prestados. Esto implica que las entidades financieras confían en que el prestatario pueda realizar compras o inversiones con la obligación de devolver el monto del préstamo en el futuro, generalmente con intereses.

El machine learning está transformando la evaluación crediticia, la capacidad de analizar grandes volúmenes de datos de manera eficiente permite a las instituciones financieras la creación de modelos predictivos mas sofisticados para evaluar el riesgo crediticio de los clientes de manera más precisa y rápida. Estos modelos pueden identificar patrones que podrían pasar desapercibidos para los métodos tradicionales, lo que permite tomar decisiones crediticias más informadas.

2 Objetivo del proyecto

El objetivo del proyecto es desarrollar un modelo capaz de predecir el comportamiento crediticio de los clientes, que permita a las instituciones financieras evaluar el riesgo crediticio y tomar decisiones informadas. Al no utilizar información crediticia de los clientes para el desarrollo del modelo, permitirá evaluar incluso a clientes sin historial crediticio.

3 Base de datos

La base de datos elegida para desarrollar el modelo se llama Application Record. Contiene 20,508 registros de tarjetas de crédito y cada registro contiene información del cliente al momento de la solicitud de crédito. Luego de realizar limpieza en la base de datos y crear nuevas variables a partir de las ya existentes, la base de datos contiene 21 variables finales. Hay que resaltar que ninguna de las variables es referente a historial crediticio.

1. Variables Categóricas

- CatTipoIngresos
- CatOcupacion
- CatEducacion
- CatEdoCivil
- CatVivienda
- FlagWorkPhone
- FlagPhone
- FlagEmail
- FlagMasculino
- FlagAuto
- FlagCasa
- MALO

2. Variables Continuas

- CantidadHijos
- MontoIngresoTotal
- CantidadMiembrosFamilia
- Edad
- ExperienciaLaboral
- IngresoPorHijo
- EdadPrimerTrabajo
- IngresoAjustado2
- MediosComunicacion

4 Estadística Descriptiva

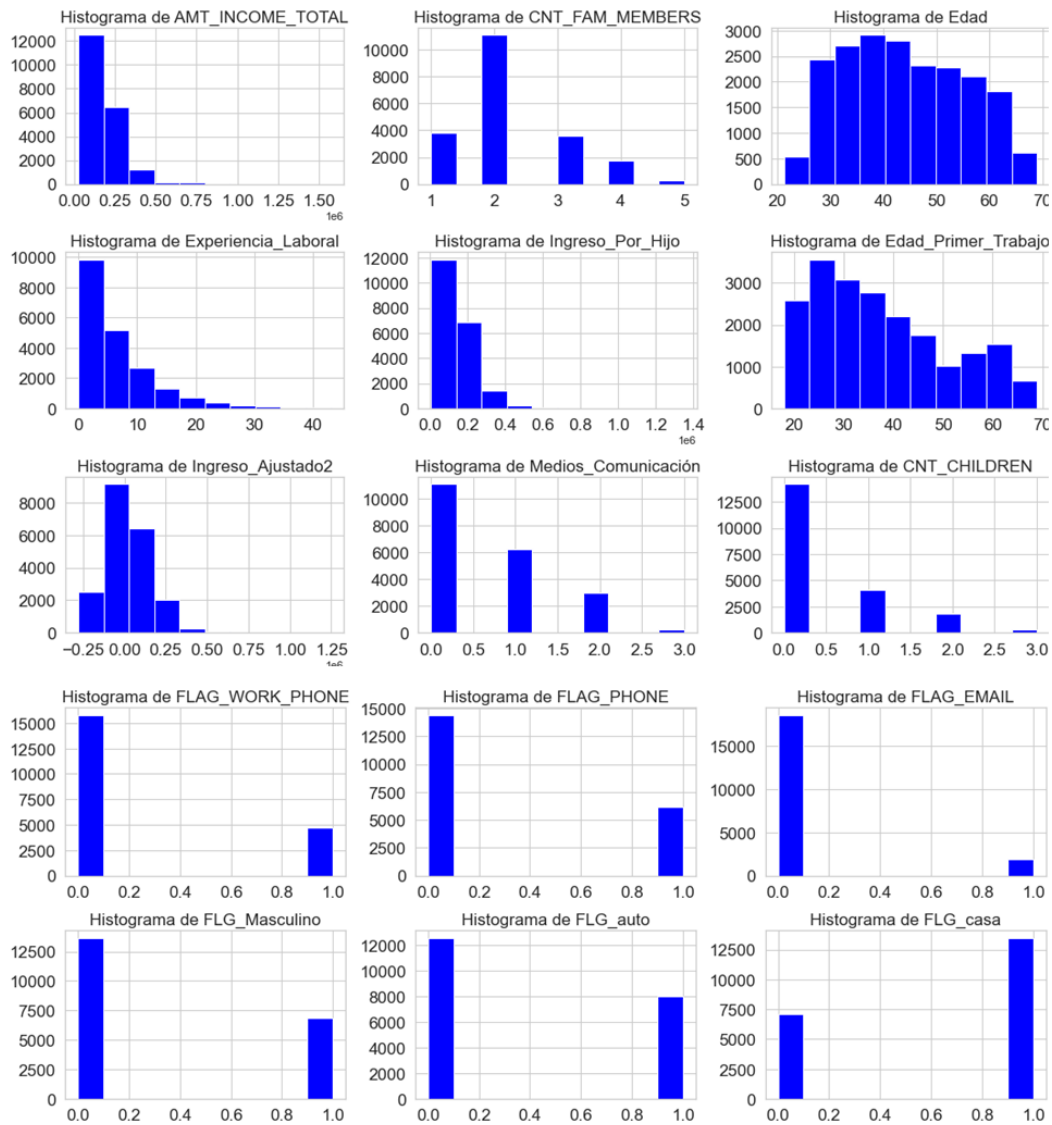


Figure 1: Histograma variables continuas

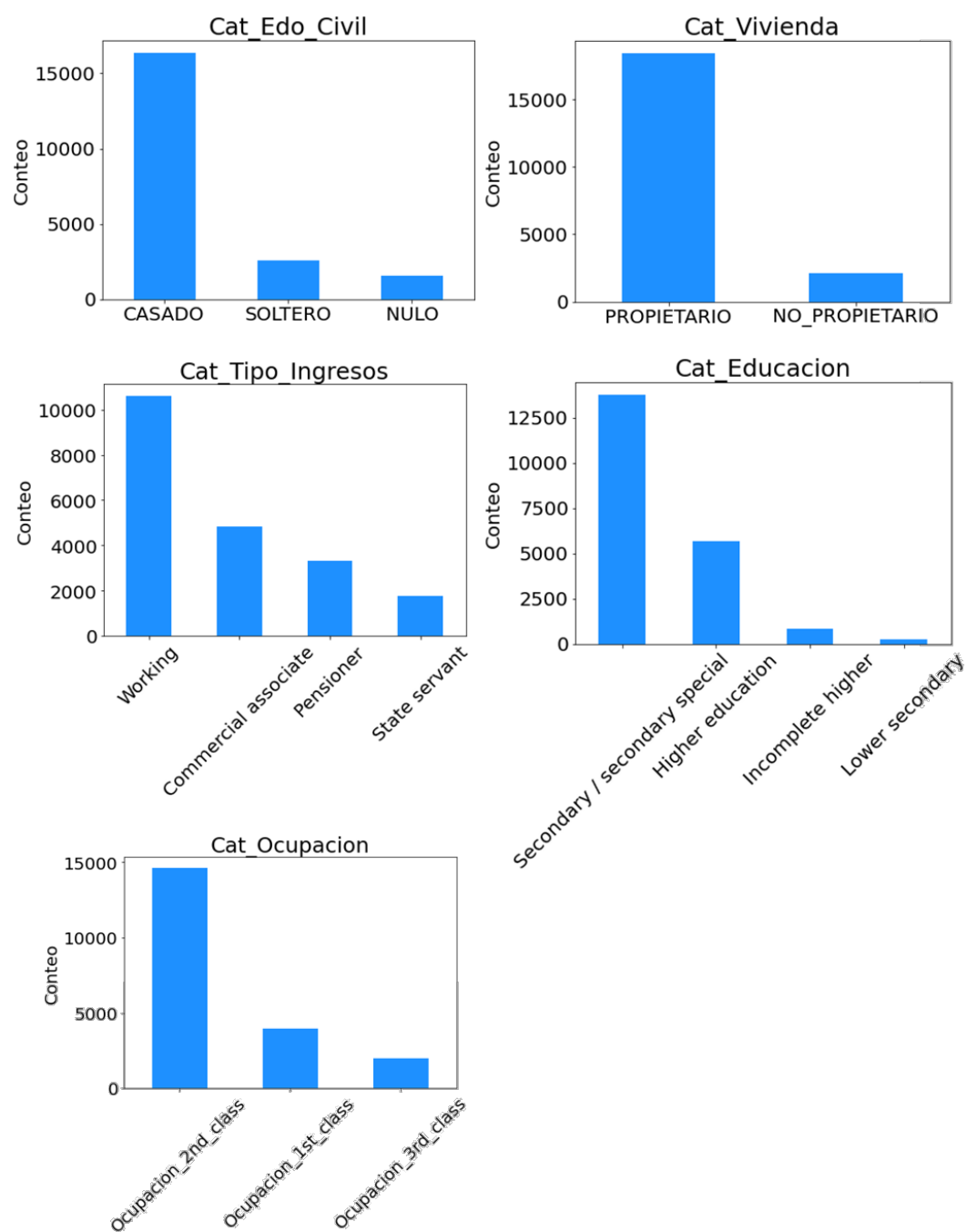


Figure 2: Gráfico de barras variables categóricas

5 Metodología

5.1 Variable objetivo

La variable objetivo del modelo es la variable MALO, esto significa que es la variable que el modelo trata de predecir.

Esta variable es binaria (0/1) y sirve para indicar el desempeño de una tarjeta de crédito. De forma que podemos identificar con un 1 a los clientes malos y con un 0 a los clientes buenos. Un cliente se considera ‘malo’ cuando cumple con lo siguiente: Atraso de 60 días o más, durante los primeros 24 meses desde que se coloca la tarjeta de crédito.

En esta base de datos cerca del .10 de los clientes están marcados como malos.

Estatus	Créditos	Porcentaje Créditos
0	18491	0.9016
1	2017	0.0984

Table 1: Conteo Clientes Malos.

5.2 Modelo Elegido

El algoritmo elegido para este modelo es el Xtreme Gradient Boosting, se utiliza la versión XGB Classifier, ya que estamos hablando de un problema de clasificación. Recordemos que el objetivo es predecir la variable binaria MALO. El XGB es un algoritmo de aprendizaje automático supervisado que se basa en la técnica de boosting. El boosting es una técnica de ensamble en la que se combinan N modelos simples (generalmente árboles de decisión) para crear un modelo más fuerte. El objetivo principal del boosting es mejorar el rendimiento del modelo, disminuyendo los errores del modelo anterior en iteraciones sucesivas. Es decir, el XGB funciona entrenando N modelos simples enfocándose en corregir los errores del modelo anterior y enfocándose en las instancias que el modelo previo clasificó incorrectamente. Este algoritmo es ampliamente utilizado y es conocido por su eficiencia y capacidad para manejar una variedad de problemas de regresión y clasificación.



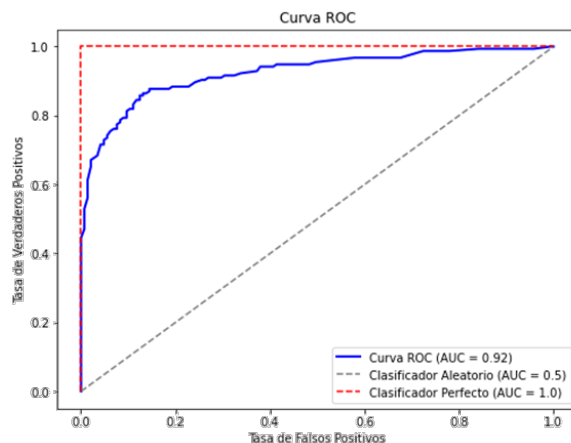
5.3 Métrica de Desempeño

Las métricas de error o de desempeño nos permiten medir que tan bueno es el modelo para cumplir su objetivo y de esta forma tomar decisiones. Se eligió el AUC (Área bajo la curva) ya que se adapta perfectamente a este problema de clasificación binaria. El AUC funciona de la siguiente manera:

Primero, la curva ROC es una representación gráfica que se utiliza comúnmente en problemas de clasificación para evaluar el rendimiento de un modelo, particularmente en modelos binarios. Esta curva muestra la relación entre dos tasas:

- Tasa de Verdaderos Positivos (TPR) : Mide la capacidad del modelo para identificar instancias positivas. Esta es la proporción de positivos reales que el modelo clasifica correctamente como positivos.
- Tasa de Falsos Positivos (FPR) : Mide la tasa de errores en los que el modelo clasifica incorrectamente instancias negativas como positivas. Esta es la proporción de negativos reales que el modelo clasifica incorrectamente como positivos.

La curva ROC traza la TPR en el eje vertical y la FPR en el eje horizontal a medida que se ajusta el umbral de decisión del modelo, tal como se muestra en la siguiente gráfica:



El AUC es el área debajo de la curva ROC. Cuanto mayor sea el AUC, mejor será el rendimiento del modelo. El AUC varía de 0 a 1, donde:

- AUC menor a 0.5 indica que el modelo es peor que una decisión aleatoria.
- AUC igual a 0.5 indica que el modelo tiene un rendimiento similar al aleatorio.
- AUC mayor a 0.5 indica que el modelo tiene cierta capacidad de discriminación y es mejor que una decisión aleatoria.
- AUC igual a 1 indica un clasificador perfecto que clasifica todas las instancias positivas antes que las negativas.

Para ilustrar el uso de la métrica en este proyecto, se calculó la curva ROC en el conjunto de datos para evaluar el rendimiento del XGB Classifier. Este análisis se realizó tanto en la muestra de entrenamiento (.70) como en la muestra de prueba (.30), utilizando los parámetros por defecto del modelo.

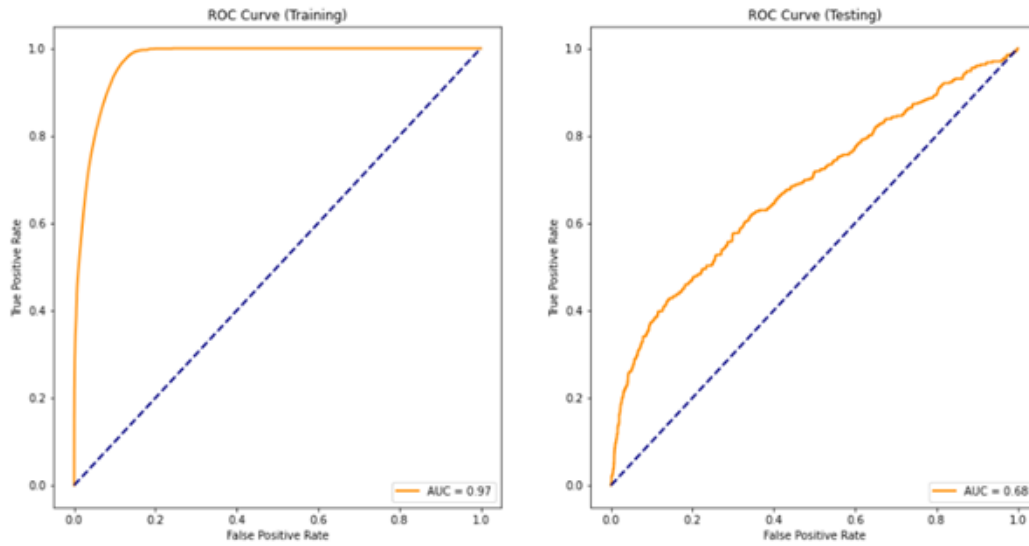


Figure 3: AUC XGB con hiperparámetros por defecto

5.4 Entrenamiento y Validación

Uno de los problemas más frecuentes en el desarrollo de un modelo es el sobreajuste. Se dice que un modelo presenta sobreajuste cuando tiene un gran desempeño al predecir en la base de datos de entrenamiento, pero no lo hace de la misma manera en datos nuevos. Para reducir el sobre ajuste se realizan experimentos con el modelo dividiendo los datos en 2 muestras, la de entrenamiento y la de validación. Buscando elegir la combinación de hiperparámetros que permitan alcanzar un buen desempeño tanto en la muestra de entrenamiento como en la de validación. A través del diseño de experimentos, se utilizan 3 combinaciones de porcentajes para segmentar nuestra base de datos, con la hipótesis de que en 1 de esas 3 combinaciones se reduzca de menor manera el sobre ajuste: 0.60 y 0.40, 0.70 y 0.30, 0.80 y 0.20.

5.5 Diseño de Experimentos

El diseño de experimentos de este proyecto se enfoca en encontrar la mejor combinación de hiperparámetros y porcentaje de muestra de entrenamiento que maximiza el desempeño del modelo XGB en la muestra Test, ya que se desea encontrar el mejor modelo que sea capaz de predecir en datos que nunca ha visto.

5.5.1 Parte 1 Hiperparámetros más influyentes

Primeramente, se crea una matriz con distintas combinaciones de hiperparámetros:

- `nestimators = [25, 50, 75, 100]`

Este hiperparametro hace referencia a la cantidad de árboles que tendrá nuestro modelo, la teoría dice que una gran cantidad de árboles aumenta el desempeño, pero aumenta el costo computacional.

- `subsample = [0.5, 0.7, 0.85, 1]`

Este hiperparametro indica el porcentaje de población que tomará cada árbol para entrenarse. La teoría dice que el no utilizar la totalidad de la muestra para cada árbol, le da diversidad a cada uno de ellos, lo que podría reducir el sobreajuste.

- `maxdepth = [3, 7, 12]`

Este hiperparámetro hace referencia a la profundidad de cada árbol..

- `learningrate = [0.05, 0.1, 0.15]`

Este hiperparámetro controla la tasa de aprendizaje del modelo, es decir, la magnitud con la que los ajustes se realizan durante cada iteración del entrenamiento.

- `frac = [0.60, 0.70, 0.80]`

Estos valores representan el porcentaje de la muestra de entrenamiento que se utiliza para entrenar el modelo. Un mayor porcentaje puede llevar a un modelo más robusto, pero es importante equilibrar esto con la capacidad de cómputo disponible. Este no es un parámetro propio del XGB, es un parámetro de la función `sample` de Python, que funciona para tomar muestras aleatorias. De igual manera se incluye en el experimento.

El resto de hiperparámetros mantienen su valor por Default.

	OP_ESTIMATORS	OP_SUBSAMPLE	OP_MAX_DEPTH	OP_LEARNING_RATE	OP_COLSAMPLE	OP_LOSS	OP_TRAIN_SAMPLE
0	25	0.5	3	0.05	1	log_loss	0.6
1	25	0.5	3	0.05	1	log_loss	0.7
2	25	0.5	3	0.05	1	log_loss	0.8
3	25	0.5	3	0.10	1	log_loss	0.6
4	25	0.5	3	0.10	1	log_loss	0.7
...
427	100	1.0	12	0.10	1	log_loss	0.7
428	100	1.0	12	0.10	1	log_loss	0.8
429	100	1.0	12	0.15	1	log_loss	0.6
430	100	1.0	12	0.15	1	log_loss	0.7
431	100	1.0	12	0.15	1	log_loss	0.8

432 rows × 7 columns

Figure 4: Matriz con todas las combinaciones de hiperparámetros

Al entrenar el modelo con las 432 combinaciones distintas de hiperparámetros y obtener su desempeño AUC tanto para la muestra Train como para la muestra Test. Se realizó un análisis para determinar cuáles hiperparámetros son los que tienen mayor influencia en el desempeño del modelo:

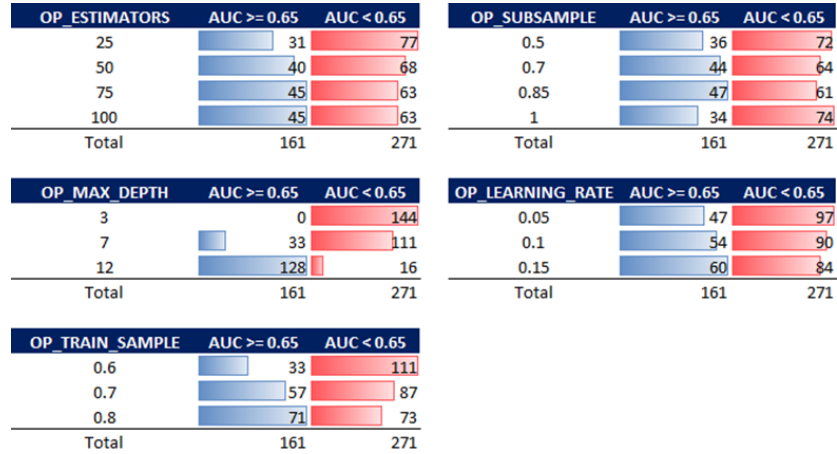


Figure 5: Resultados Primer Experimento

La conclusión de este primer experimento es que para este conjunto de datos, un mayor valor en el hiperparámetro nestimators nos permite obtener un desempeño (muestra Test) en el modelo XGB. De manera similar un mayor valor en maxdepth, un mayor valor en frac y un mayor valor en learningrate nos permite obtener un mejor desempeño en el modelo XGB.

También logra un mejor desempeño en los valores 0.7 y 0.85 para el hiperparámetro subsample, pero gráficamente parece que este no influye tan significativamente.

5.5.2 Parte 2 Mejor combinación de hiperparámetros

Tras obtener conclusiones sobre los valores de hiperparámetros que mejoran el desempeño del modelo, se realiza un segundo experimento. Esta vez dándole los siguientes valores a los hiperparámetros:

- nestimatros = [75, 100, 125, 150]
- subsample = [0.7, 0.85]
- maxdepth = [12, 15, 20]
- learningrate = [0.15, 0.2]
- frac = [.80]

Nuevamente, se realizó un análisis para determinar que valores en los hiperparámetros optimizan el desempeño del modelo en la muestra Test, en donde se concluye que la siguiente combinación es la mejor: nestimatros = 75, subsample = 0.85, maxdepth = 20, learningrate = 0.15, frac = 0.80.

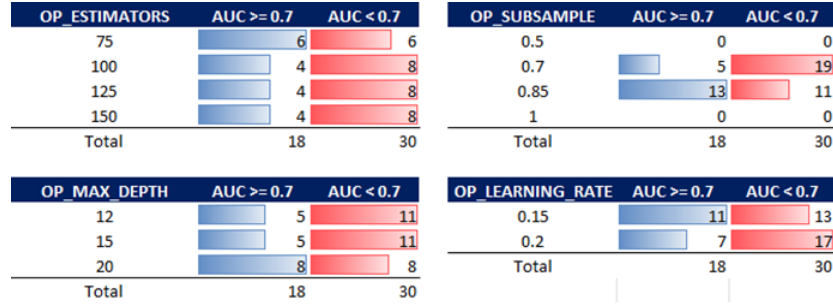


Figure 6: Resultados segundo experimento

De forma que no se eligió el modelo con mejor desempeño en Test, se eligió el modelo la mejor combinación de hiperparámetros definida anteriormente, el cual logra un desempeño de 0.7034.

OP_ESTIMATORS	OP_SUBSAMPLE	OP_MAX_DEPTH	LEARNING_R	OP_TRAIN_SAMPLE	AUC_TRAIN	AUC_TEST
150	0.85	20	0.15	0.8	0.9756	0.7078
100	0.7	15	0.15	0.8	0.9743	0.7077
75	0.85	15	0.2	0.8	0.9749	0.7065
75	0.7	20	0.15	0.8	0.9743	0.7045
100	0.7	12	0.15	0.8	0.9729	0.7043
125	0.85	15	0.15	0.8	0.9752	0.7039
75	0.85	20	0.15	0.8	0.9749	0.7034
150	0.85	15	0.15	0.8	0.9759	0.7033
75	0.7	15	0.2	0.8	0.9742	0.7018
150	0.85	12	0.15	0.8	0.9754	0.7017
100	0.85	20	0.15	0.8	0.9753	0.7014
125	0.85	12	0.2	0.8	0.9750	0.7014
150	0.7	12	0.2	0.8	0.9747	0.7012
125	0.85	20	0.15	0.8	0.9755	0.7012
100	0.85	20	0.2	0.8	0.9760	0.7011
75	0.85	20	0.2	0.8	0.9755	0.7010
75	0.85	12	0.15	0.8	0.9719	0.7007

Figure 7: Modelos con mejor desempeño en el segundo experimento

5.5.3 Parte 3 Desempeño Promedio

Por último, utilizando el modelo elegido (Un XGB con $\text{nestimators} = 75$, $\text{maxdepth} = 20$, $\text{subsample} = 0.85$, $\text{learningrate} = 0.15$, $\text{frac} = 0.80$) nuevamente se entrena el modelo 50 veces, obteniendo un desempeño promedio de AUC de 0.7023, lo cual es muy cercano al desempeño esperado de 0.7034, lo cual es muy aceptable para un modelo de este tipo.

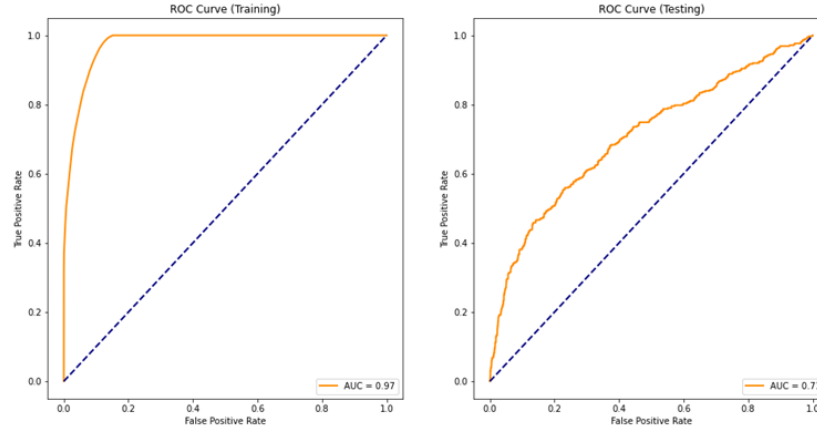


Figure 8: AUC modelo seleccionado

6 Interpretación de Resultados

Se utilizó la muestra Test para realizar un análisis en el cual se divide a la población en 3 rangos de probabilidades y se calculó lo siguiente:

- Créditos :Número de créditos por rango.
- Malos :Número de créditos malos por rango.
- Tasa Malos :Número de malos por rango/Número de Créditos por rango.
- Porcentaje Aprobación :Porcentaje de créditos aprobados acumulados desde el rango 1 hasta el rango n
- Tasa Malos acumulada:Tasa de Malos para la población acumulada desde el rango 1 hasta el rango n.

Test						
Rango	Probabilidad	Créditos	Malos	Tasa Malos	%Aprobación Acumulada	Tasa Malos Acumulada
1	< 0.00895	1,641	76	4.6%	40%	4.6%
2	[0.00895, 0.07323)	1,640	118	7.2%	80%	5.9%
3	≥ 0.07323	821	196	23.9%	100%	9.5%
Total		4,102	390	9.5%		

Figure 9: Interpretación de resultados en la muestra Test

Se eligió a la muestra Test para este análisis ya que nos permite conocer el desempeño del modelo prediciendo en datos nuevos. El modelo elegido nos permite clasificar a los clientes en 3 rangos con Tasas de Malo distintas, en específico el rango 3 tiene una Tasa de Malos muy superior a la Tasa de Malo promedio (0.095). Este análisis le permite a una institución financiera elegir entre 2 escenarios distintos para mejorar la calidad de la cartera de crédito:

- Utilizar un punto de corte de 0.00895 para reducir la Tasa de Malos de 0.095 a 0.046, aprobando el 0.40 de los clientes.
- Utilizar un punto de corte de 0.07323 para permite reducir la Tasa de Malos de 0.095 a 0.059 aprobando el 0.80 de los clientes.

Ambos escenarios tienen sus ventajas y desventajas, y la elección depende de los objetivos de la institución financiera. Si el énfasis está en mejorar la calidad de la cartera a costa de una aprobación más selectiva, el primer escenario es el ideal. En cambio, si se prioriza aprobar a más clientes, aunque implique no mejorar tanto la calidad de la cartera, el segundo escenario sería más adecuado.

Esta capacidad de segmentar a los clientes en diferentes niveles de riesgo basados en la probabilidad de ser malos pagadores ofrece una valiosa herramienta para mejorar la toma de decisiones en la aprobación de créditos y la gestión del riesgo crediticio de una institución financiera.

7 Conclusión

El modelo seleccionado, demuestra un desempeño aceptable para predecir el comportamiento crediticio de los clientes, respaldado por la métrica del Área bajo la Curva (AUC). Permitirá a las instituciones financieras evaluar de manera efectiva el riesgo crediticio para la toma de decisiones informadas.

Al no depender de información crediticia, este modelo brinda la posibilidad de evaluar incluso a clientes sin historial crediticio, ampliando las oportunidades de evaluación y toma de decisiones para las instituciones financieras.

El diseño de experimentos nos permitió seleccionar los hiperparámetros que optimizan el desempeño del algoritmo Xtreme Gradient Boosting Classifier en nuestra base de datos y así seleccionar el mejor modelo.

Al dividir la población en tres rangos de probabilidades, se destacó la capacidad del modelo para clasificar a los clientes en diferentes niveles de riesgo. Esta segmentación permite a las instituciones financieras tomar decisiones estratégicas, ofreciendo dos escenarios claramente diferenciados:

- Utilizando un punto de corte de 0.00895, se logra reducir la Tasa de Malos del 0.095 al 0.046, priorizando la mejora en la calidad de la cartera, aunque aprobando solo el 0.40 de los clientes.
- Utilizando un punto de corte de 0.07323, se logra reducir la Tasa de Malos del 0.095 al 0.059, aprobando un 0.80 de los clientes.

Ambos escenarios presentan ventajas y desventajas, y la elección final depende de los objetivos de la institución financiera. Si se prioriza mejorar la calidad de la cartera, el primer escenario es la mejor opción. En cambio, si se busca aprobar a más clientes, aunque implique una mejora menos significativa en la calidad de la cartera, el segundo escenario es más apropiado.

References

- [1] Analytics Vidhya. (2023). Guide to AUC ROC Curve in Machine Learning. Recuperado de <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>
- [2] DataEvo. (2017). Machine Learning: del árbol de decisión al bosque de árboles con boosting (XGBoost). Recuperado de <https://www.youtube.com/watch?v=PN-8dTXh3DU&t=449s>
- [3] MiraltaBank. Riesgo de Crédito, modelos de Inteligencia Artificial IA. Recuperado de <https://www.miraltabank.com/riesgo-de-credito-modelos-de-inteligencia-artificial/>
- [4] kaggle. 2019. Credit Card Approval Prediction. Recuperado de <https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction/code/>