UNIVERSIDAD COMPLUTENSE MADRID

ntic master
revolucionamos la comunicación

TRABAJO FINAL DE MÁSTER

MODELO DE CLASIFICACIÓN CREDITICIA

Autor: Joel Reyes L.

RESUMEN

Con el fin de optimizar los tiempos de evaluación crediticia y disminuir los índices de morosidad, la dirección de una empresa financiera ha solicitado al área de tecnológica crear un sistema inteligente que clasifique a los postulantes a crédito en tres diferentes niveles de riesgo: *bueno, estándar* y *malo*.

El objetivo de este documento es detallar las distintas etapas de desarrollo y selección de un modelo de clasificación crediticia, elemento central del mencionado sistema inteligente.

En términos generales, los modelos de clasificación crediticia son herramientas de *machine learning*[1] utilizadas para identificar la solvencia de individuos y entidades en diferentes niveles de riesgo a partir de distintos factores, como el historial de pagos, los ingresos, los activos y otros indicadores financieros.

[1] *https://www.ibm.com/es-es/topics/machine-learning*

I. ANTECEDENTES GENERALES

1. DEFINICIÓN DEL PROBLEMA

La dirección de una empresa financiera ha solicitado a su equipo de desarrollo tecnológico crear una herramienta que permita evaluar el riesgo crediticio. Esta herramienta deberá asistir a los ejecutivos en el proceso de aprobación de créditos, reducir la tasa de morosidad y asignar eficientemente recursos financieros.

1.1 OBJETIVO

El objetivo es desarrollar un modelo predictivo que clasifique a los nuevos solicitantes a crédito en diferentes niveles de riesgo, basándose en la información ingresada al momento de la postulación.

2. METODOLOGÍA

2.1 Público Objetivo

Este modelo está dirigido a personas que buscan obtener algún crédito de la empresa financiera.

2.2 Fuente de Datos

A lo largo de los años, esta empresa financiera ha podido recopilar datos bancarios básicos, además de mucha información asociada a la solicitud de créditos personales la cual se encuentra disponible en el sitio de *Kaggle.com* *Credit Score Classification*. Para mayor detalle ver **TABLA 3 Descripción general de variables**.

2.3 Control de versiones y desarrollo

En cuanto al desarrollo y el control de versiones, este puede revisarse en el repositorio *GitHub* *Trabajo Final de Master Data Scientist*.

2.4 Comprobaciones Iniciales

A fin de detectar posibles valores mal codificados, fuera de rango o valores perdidos no declarados, se ha realizado una comprobación inicial de los datos.

En esta revisión se identificaron inconsistencias en la tipificación de variables, originadas por la presencia de prefijos o sufijos con caracteres especiales en algunos valores numéricos, lo que provocó que la variable fuera catalogada como categórica en lugar de numérica.

Por otro lado, se detectaron valores perdidos no declarados. Las variables numéricas identificadas con este error fueron *Num_of_Loan*, el número de créditos, y el *Monthly_Balance*, el saldo actual en la cuenta. En cuanto a las variables no numéricas con este error podemos mencionar los campos *Occupation* y *Payment_Behaviour*.

2.5 Corrección de Errores Detectados

Para las variables numéricas con valores perdidos no declarados, los valores fueron reemplazados por *NaN*[2]. Los valores numéricos que incluían sufijos o prefijos fueron ajustados, eliminando dichas extensiones para que la variable pudiera ser tipificada de forma correcta.

Finalmente, aquellas variables de identificación única como: el identificador de registro (*ID*), el identificador de consumidor (*Customer*_ID), el nombre de la persona (*Name*) y el número de seguro social (*SNN*), fueron removidas del conjunto de datos debido a que no aportan valor a la predicción.

Las variables cualitativas fueron normalizadas, y aquellas con valores perdidos no declarados fueron reemplazados, en su mayoría, por *NaN*. En el caso de la variable *Occupation* el valor perdido no declarado fue reemplazo por el valor *Other*, haciendo alusión a *"Otro tipo de profesiones u ocupaciones"*. En el caso de la variable *Credit_History_Age*, esta fue convertida a numéricas (ver **TABLA 4 Errores y correcciones**).

[2] *NaN* "not a number", codificación estándar para valores perdidos o perdidos no declarados.

2.6 Descripción de Variable Objetivo

La variable objetivo, *Credit_Score*, está compuesta por tres categorías: *good*, *poor* y *standard*. La clase mayoritaria es *standard* con un total de 53.17% de los registros válidos, por otro lado, la clase minoritaria es *good*, con un 17.83%.

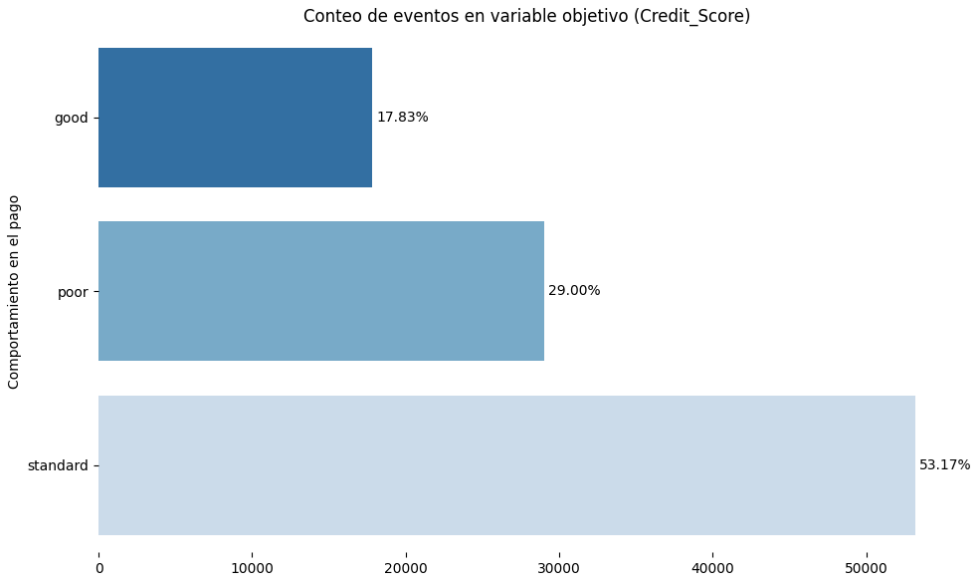

Fig. 1 Conteo de eventos de variable objetivo

2.7 Análisis Univariante

Después de corregir los errores detectados en la fase inicial de verificación, se puede proceder con un análisis descriptivo del conjunto de datos.

i Variables cuantitativas

En cuanto a las variables numéricas el porcentaje de registros no informados, o *missings*, es bastante bajo, siendo la variable *Monthly_Inhand_Salary* la que se lleva el mayor porcentaje con un 15%.

Durante el estudio se ha podido observar variables con coeficientes de variación mayores a 30%, es decir: *los datos no son homogéneos*, por lo tanto, la media no resulta representativa para estos grupos. Estas mismas variables muestran además valores positivos de asimetrías mayores a la unidad, lo que nos indica que las distribuciones se encuentran segadas hacia la derecha. Estos resultados en los coeficientes de variación y de asimetría se deben a los valores atípicos (*outliers*) extremos, los que alteran significativamente la distribución de estas

variables, esto aun cuando la presencia de valores atípicos no supera el 2% del total registros válidos. En **TABLA 5 Análisis descriptivo de variables numéricas** y **Fig. 12** es posible revisar los detalles.

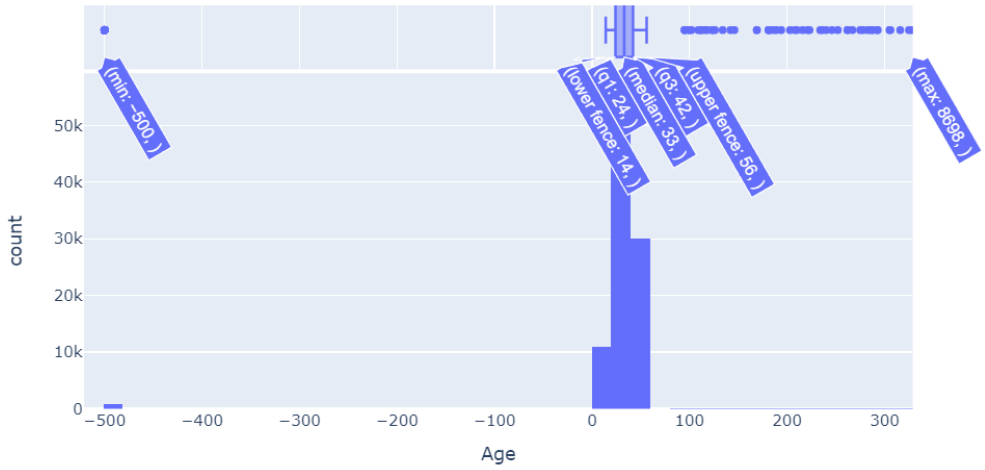

Fig. 2 Inspección gráfica de la variable Age, caso valores atípicos extremos

ii Variables cualitativas

Respecto a las variables cualitativas, se identifican principalmente valores categóricos con amplitud de hasta 16 diferentes categorías. Sólo dos de estas variables presentan valores perdidos no declarados. Para la variable *Credit_Mix* el porcentaje de valores perdidos alcanza un 20.2%, y en el caso de *Payment_Behaviour* este alcanza el 7.6%. Una situación interesante se presenta con la variable *Type_of_Loan*, la cual consta de 3533 registros únicos, debido a que cada persona puede tener más de un producto crediticio, sin embargo, sólo existen 9 tipo de productos crediticios diferente, lo que nos permite realizar una codificación organizadas por producto. Para más detalles ver **TABLA 6 Análisis descriptivo variables categóricas**.

2.8 Análisis Bivariante

Se ha realizado un análisis bivariante, con el propósito de identificar de antemano las relaciones marginales de las variables con la variable objetivo, y así obtener una idea de cuáles podrían ser influyentes en nuestro modelo.

i Correlación entre variables cuantitativas

Para estimar la correlación con la variable objetivo se realizado un reemplazo ordinal de esta.

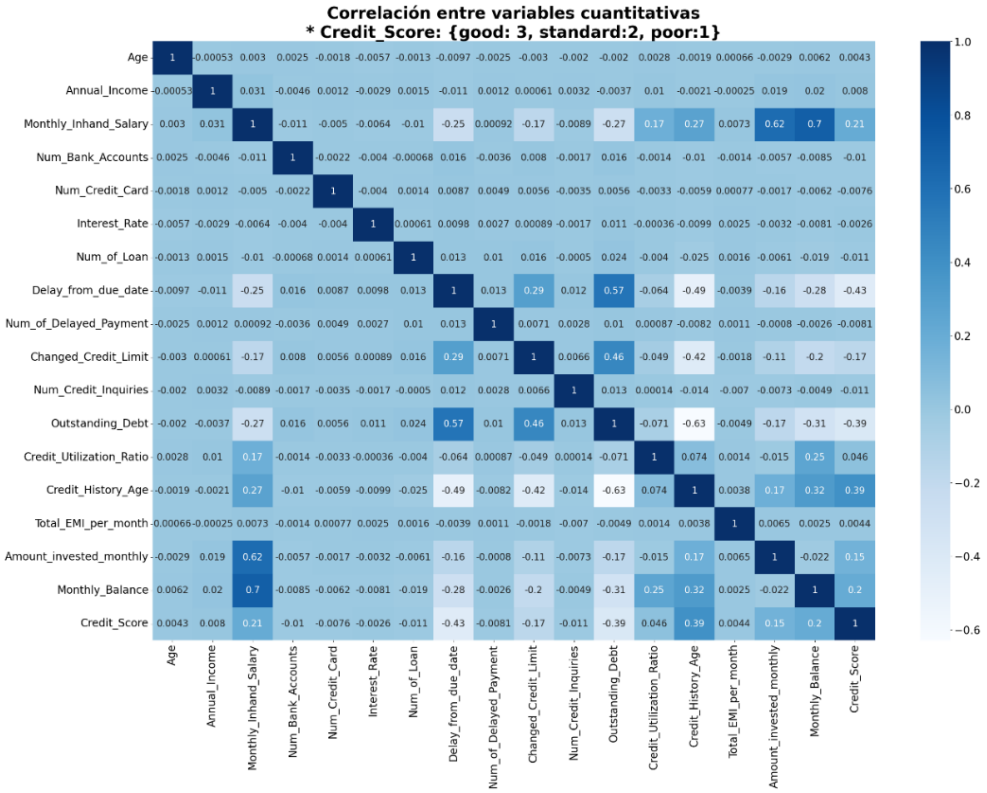

Fig. 3 Correlaciones entre variables cuantitativas y *Credit_Score*

Las tres variables numéricas más influyentes observadas son *Credit_History_Age*, *Outstanding_Debt* y *Delay_from_due_date*.

ii Correlación entre variables cualitativas

Para establecer una correlación entre las variables cualitativas y la variable objetivo se ha utilizado la *V de Cramer*[3] como métrica de evaluación.

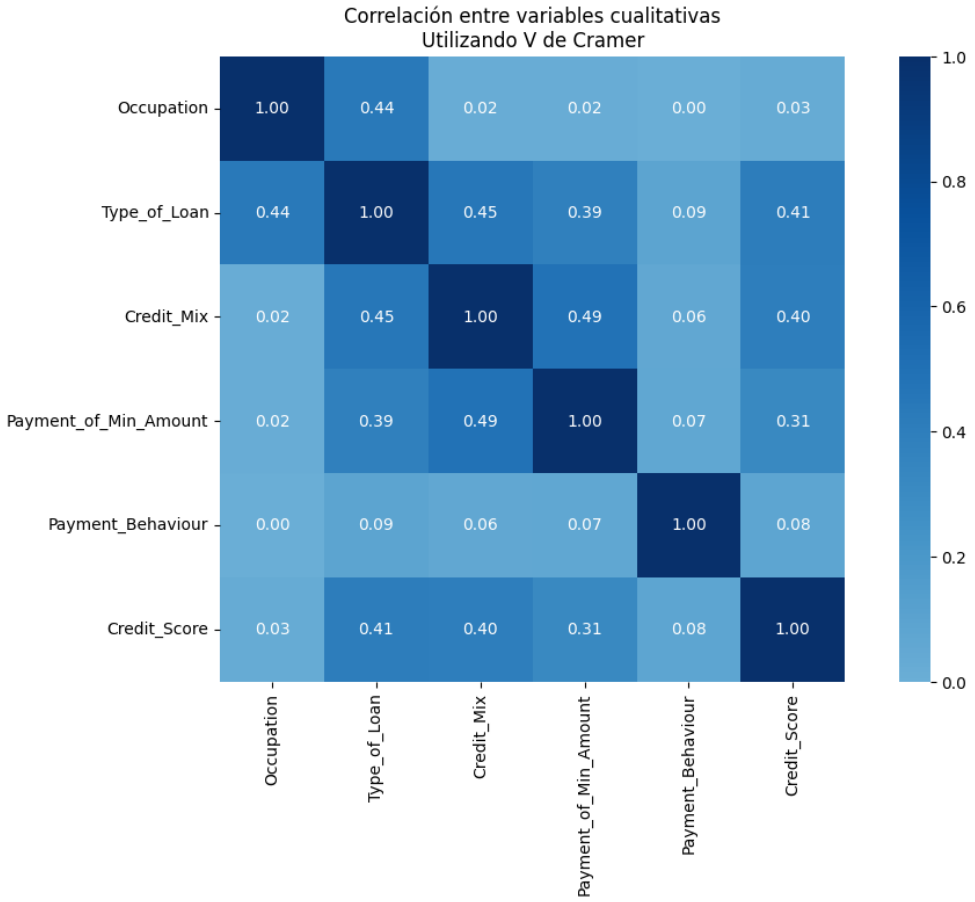

Fig. 4 Correlaciones entre variables cualitativas y *Credit_Score*

Inicialmente, las variables con mayor influencia observadas son *Type_of_Loan*, *Credit_Mix* y *Payment_of Min_Amout*.

[3] *https://www.ibm.com/docs/es/cognos-analytics/11.1.0?topic=terms-cramrs-v*

2.9 Transformaciones

En *Machine Learning* (ML), algunos algoritmos admiten el uso de variables categóricas, pero la mayoría solo acepta valores numéricos. Adicionalmente, se ha comprobado que el empleo de valores numéricos mejora el rendimiento al entrenar y predecir con modelos de ML.

Debido a estos factores, se han implementado una serie de transformaciones que procederemos a revisar a continuación:

TABLA 1 Resumen de transformación a variables categóricas

Variable	Transformación	Comentarios
Occupation	LabelEncoder	No se requiere orden predeterminado
Type_of_Loan	MultiLabelBinarizer	Requiere multietiqueta
Credit_Mix	OrdinalEncoder	Requiere jerarquía
Payment_of_Min_Amount	LabelEncoder	No se requiere orden predeterminado
Payment_Behaviour	OrdinalEncoder	Requiere jerarquía
Credit_Score	OrdinalEncoder	Requiere jerarquía

Es importante mencionar que se realizaron varias transformaciones, como la tramificación de variables numéricas (ver **TABLA 7 Resultados antes/después de tramificación**) o la reducción de dimensionalidad, a fin de mejorar el rendimiento. Estas transformaciones fueron descartadas debido a su efecto negativo en el rendimiento de los modelos.

2.10 Imputaciones

i Valores atípicos (Outliers)

Se evaluaron distintas estrategias de manejo de valores atípicos, entre las que se cuentan: la *winsorización*[4], el reemplazo por valores estadísticos de posición central, como la media y la mediana, y el reemplazo por valores perdidos. Resultando la eliminación de estos registros, como la mejor alternativa debido a las mejoras en el rendimiento de los algoritmos de clasificación.

[4] *Winsorización*: Técnica estadística que se utiliza para limitar los valores extremos en el conjunto de datos

ii Valores perdidos (Missings)

Se utilizaron diversas técnicas para manejar los valores perdidos, incluyendo la imputación simple con medidas de tendencia central y la imputación iterativa basada en cadenas de Markov. Sin embargo, la mejor opción resultó ser la imputación *KNN* de vecinos más cercanos.

2.11 Preparación Previa al Modelamiento

Antes de proceder con el modelado, se realizaron los pasos finales: escalamiento, balanceo y división del conjunto para entrenamiento y prueba.

Después de completar las transformaciones e imputaciones, y considerando que los datos estaban en diferentes escalas, se llevó a cabo un proceso de escalamiento.

Para mejorar el desbalance de datos, se empleó una técnica de aumento de datos conocida como *SMOTE*[5]. Este método permite crear nuevos registros pertenecientes a las clases minoritarias, equilibrando así el conjunto de datos para su entrenamiento.

Finalmente se llevó a cabo una separación estratificada del conjunto para asegurar proporciones iguales entre los subconjuntos de entrenamiento y prueba.

[5] SMOTE: Técnica de Sobre muestreo Sintético de Minorías

II. MODELAMIENTO Y SELECCIÓN

Durante esta fase, se probaron y evaluaron varios algoritmos de *Machine Learning* y algunas arquitecturas de redes neuronales comúnmente empleadas para clasificación. Luego de una evaluación de rendimiento, los algoritmos que más destacaron fueron llevados a una etapa final de ajuste y evaluación detallada.

1. PRUEBAS, EVALUACIONES Y COMPARACIONES

1.1 Prueba y Evaluación de Algoritmos de Machine Learning

Para identificar el algoritmo con mejor rendimiento, en términos de exactitud (*accuracy*) y desviación estándar, se utilizó una grilla de evaluación de modelos, donde se emplearon las técnicas de estratificación de datos, y la validación cruzada repetida. A continuación, podemos revisar gráficamente los resultados obtenidos.

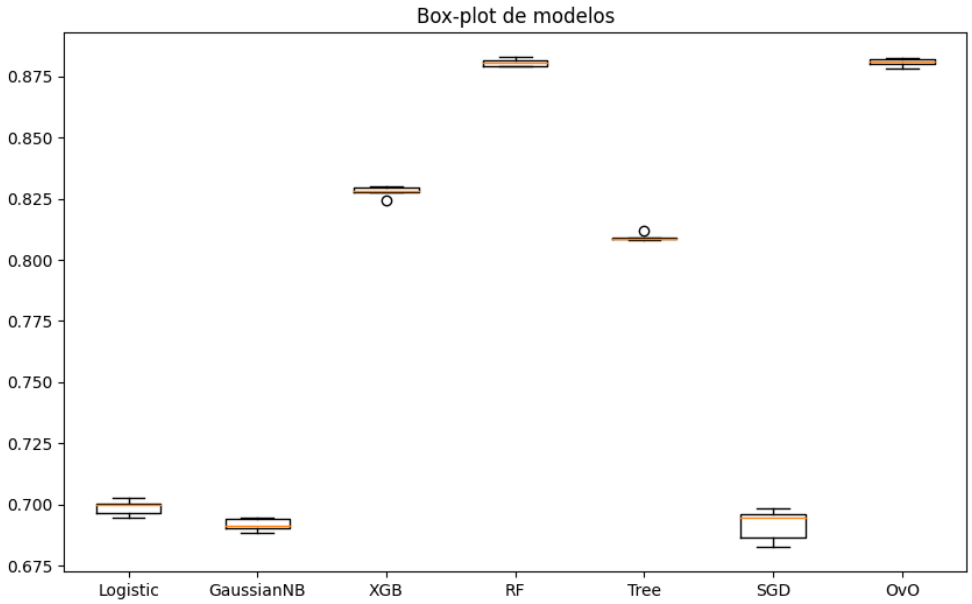

Fig. 5 Box plot de modelos de ML

El algoritmo con mejor rendimiento fue *Random Forest*[6] *(RF)*, con una exactitud de 88.0% y una desviación estándar de apenas 0.15%. El resultado completo de este experimento se encuentra en la **TABLA 8 Resumen grilla de modelos ML**.

1.2 Prueba y Evaluación Red Neuronal

Se utilizo un modelo secuencial de cinco capas ocultas, cada una con más de 120 neuronas. Además, hemos incorporado técnicas como el *dropout* y el *batch normalization*[7] para prevenir el sobreajuste.

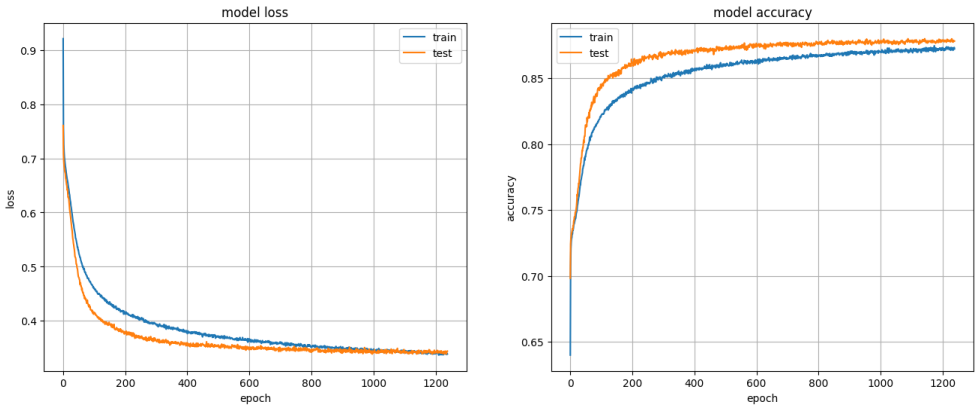

Fig. 6 Función de perdida y precisión, evaluación de la red neuronal de clasificación

Los resultados obtenidos tanto en la función de pérdida como en la exactitud, la que alcanzo también un 88%, demuestran la eficacia de este enfoque.

1.3 Comparación de Modelos de Clasificación

Ambos modelos tienen notablemente rendimiento. Aun cuando ambos tienen la misma exactitud, las métricas de precisión, sensibilidad y F1 Score, que es un resumen de ambas métricas, favorecen ligeramente al algoritmo de ML, debido a la consistencia de resultados de todas las clases, mostrando una clara ventaja al momento de identificar a la clase estándar.

A continuación, se presenta un resumen comparativo de ambos modelos:

[6] *Bosque aleatorio* https://www.ibm.com/mx-es/topics/random-forest
[7] Para más información puede consultar Técnicas de Regularización Básicas para Redes Neuronales

TABLA 2 Comparativa de rendimiento entre modelos

	Random Forest			Red Neuronal			
	precision	**recall**	**f1-score**	**precision**	**recall**	**f1-score**	**support**
good	0.89	0.94	0.92	0.88	0.97	0.92	8003
standard	0.87	0.79	0.83	0.90	0.72	0.80	8004
poor	0.88	0.91	0.90	0.86	0.95	0.90	8004
accuracy			0.88			0.88	24011
macro avg	0.88	0.88	0.88	0.88	0.88	0.88	24011
weighted avg	0.88	0.88	0.88	0.88	0.88	0.88	24011

2. OPTIMIZACIÓN DE MODELOS

Para el proceso de optimización de modelos se han utilizado las siguientes herramientas: *GridSearchCV*[8] y *Keras Tuner*[9]. Luego de ejecutar ambos procesos con distintas combinaciones de hiperparámetros para ambos modelos, los resultados fueron prácticamente iguales a los obtenidos durante la etapa anterior (ver **TABLA 9 Tuning: comparativa de modelos** y **Fig. 13**).

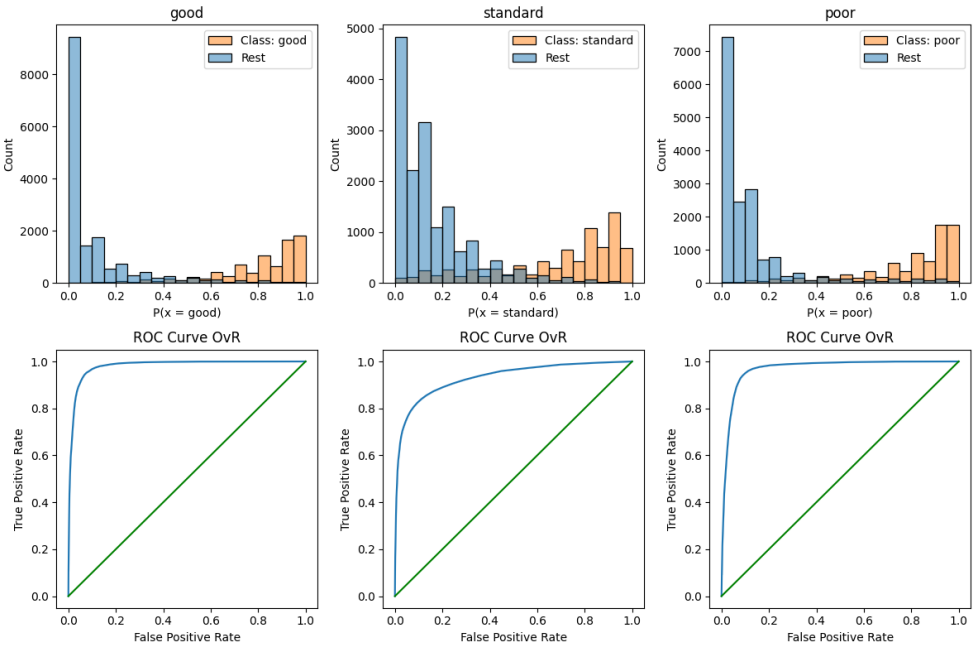

Fig. 7 Distribución de probabilidad para cada clase y sus curvas ROC

[8] Clase de *scikit-learn* que utiliza la técnica de validación cruzada repetida.
[9] Herramienta de optimización de hiperparámetros

3. MEJOR MODELO SELECCIONADO

Debido a un mejor rendimiento general, su menor complejidad y su bajo costo computacional se ha decido llevar a producción el modelo *Random Forest*.

En la **Fig. 7**, se pueden observar las palancas tras esta decisión; las distribuciones de probabilidades de cada clase frente al resto muestran un alto rendimiento, al igual que las curvas ROC. La clase estándar es la que muestra el rendimiento más bajo, evidenciando las dificultades del algoritmo para diferenciarla de las demás.

Un aspecto crucial a la hora de la selección es la explicabilidad del modelo. Utilizando la técnica de *feature importance*, podemos dar una puntuación a las variables que tienen mayor impacto en la predicción de nuestra variable objetivo.

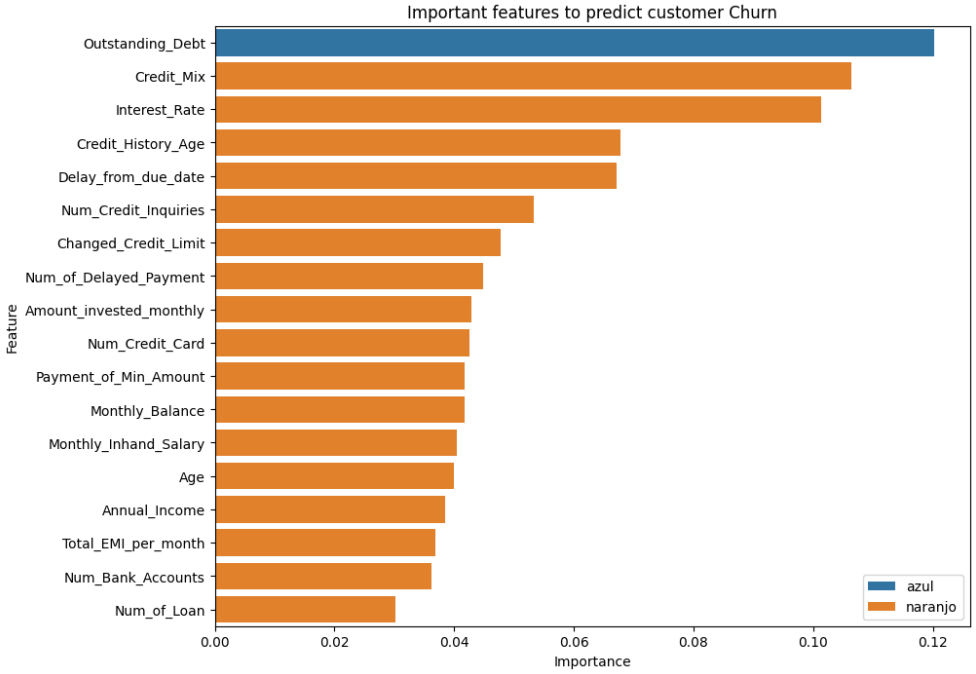

Fig. 8 Feature Importance de RandomForest

Las variables tres variables más relevantes al momento de hacer predicciones predecir son la deuda pendiente total (*Oustanding_Debt*), los tipos de crédito que aparecen en el perfil financiero del cliente (*Credit_Mix*) y la tasa de interés en las tarjetas de crédito (*Interest_Rate*). Todas estas variables reflejan el comportamiento del cliente respecto a un crédito, y demuestran su compromiso de pago ante una deuda.

III. PRODUCTIVIZACIÓN DEL MODELO Y MLOPS

La productivización es el proceso que lleva al modelo desde el entorno de pruebas hasta su implementación en el entorno de producción. Para que este proceso ocurra de manera efectiva y eficiente, y se asegure además la calidad, la estabilidad y la escalabilidad, se utilizan una serie de prácticas que automatizan y simplifican los flujos de trabajo y el despliegue, conocidas como practicas MLOps[10].

1. ALCANCE PRODUCTIVIZACIÓN

Debido a la complejidad y variedad de habilidades que se requiere para montar una solución de extremo a extremo (*end to end*) para un proyecto de ML, hemos acotado la productivización al desarrollo de un pipeline de modelamiento. Este pipeline cuenta con las etapas de preprocesamiento, entrenamiento/reentrenamiento, validación y versionamiento del modelo (ver **Fig. 10**). Para esto hemos utilizado la plataforma de analítica de *AWS*[11] llamada *Amazon SageMaker*[12].

Amazon SageMaker cuenta con un servicio llamado *Sagemaker Pipelines*, esta herramienta nos permite crear, automatizar y gestionar flujos de trabajo de ML. A continuación, veremos un resumen de nuestro pipeline desplegado en dicho servicio.

[10] *https://aws.amazon.com/es/what-is/mlops/*
[11] *https://aws.amazon.com/es/*
[12] *https://aws.amazon.com/es/sagemaker/*

2. PASOS DEL PIPELINE

Nuestro pipeline de modelamiento, o cañería de modelamiento, consta de cinco pasos: *preprocesamiento*, *entrenamiento*, *evaluación*, *condición* y *registro*. Cada uno de estos pasos puede verse de forma gráfica en la viñeta de pipelines en la plataforma de *SageMaker*.

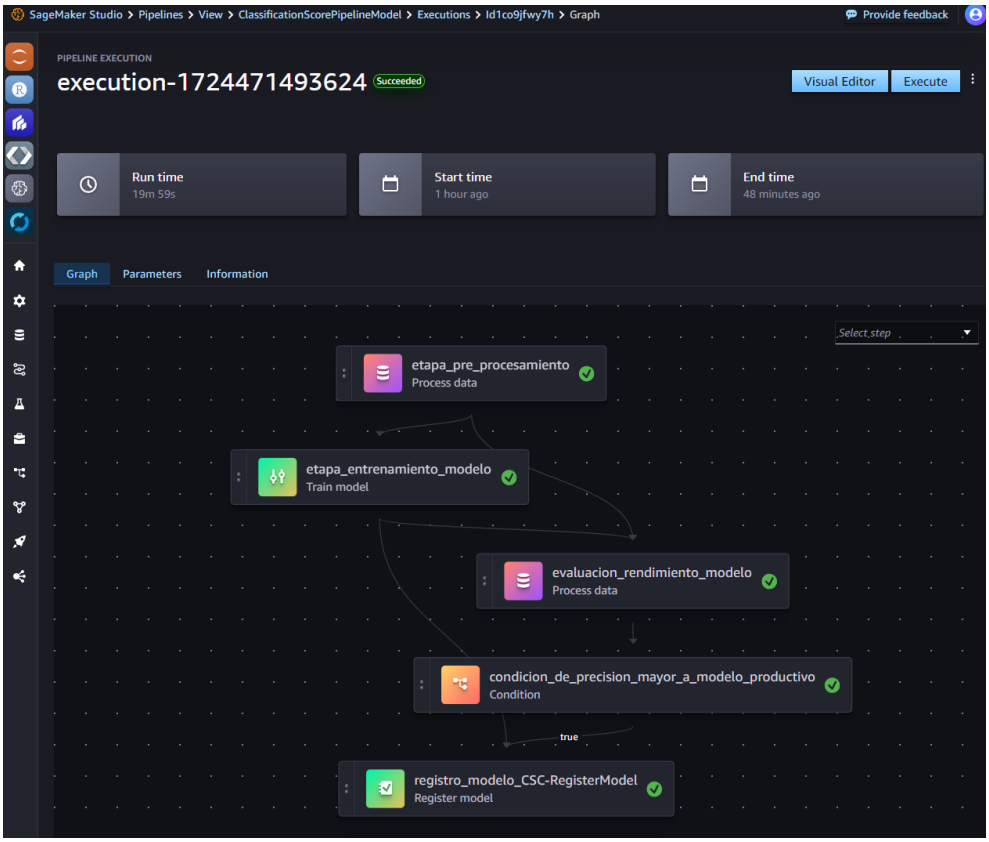

Fig. 9 SageMaker pipelines: cañería de modelamiento

2.1 Preprocesamiento

En esta etapa, hemos empleado la clase *SKLearnProcessor*, que carga una imagen *Docker*[13] proporcionada por *SageMaker*, la cual es comúnmente utilizada para el procesamiento de datos. Dentro de este contenedor ejecutamos el código de preprocesamiento (*preprocessing.py*) almacenado en la carpeta *code*. Aquí se implementa en un único script el proceso de limpieza de datos y de transformaciones efectuadas durante la etapa inicial de exploración y desarrollo del modelo.

[13] *https://www.docker.com/resources/what-container/*

2.2 Entrenamiento

En esta etapa se ha invocado un contenedor customizable de procesamiento, a través de la clase *ScriptProcessor*, aquí hemos cargamos una imagen especifica de *scikit-learn* para el entrenamiento del modelo *Random Forest*.

2.3 Evaluación

Aparte del preprocesamiento, la evaluación es una de las etapas más difíciles de programar debido a la gestión de archivos de métricas. En el caso del nuevo modelo, las métricas de evaluación son creadas y almacenadas, mientras que las métricas del modelo en producción deben ser identificadas y extraídas para la siguiente etapa de comparación. Esto se debe a que el pipeline de modelamiento está diseñado para el reentrenamiento periódico de modelos, con el fin de mantener el rendimiento de la predicción en el tiempo.

2.4 Condición

Para comparar una nueva versión del modelo con respecto a la que se encuentra en producción, se ha utilizado la métrica de exactitud (*accuracy*). En este caso, si el modelo entrenado tiene una exactitud superior a la del modelo en producción, automáticamente este se mueve al contenedor de modelos y se aprueba para su uso en inferencias.

2.5 Registro

Un aspecto clave para que un modelo y sus predicciones sean exitosas es tener varias versiones almacenadas. En este caso, la gestión de versiones del modelo de clasificación se lleva en el contenedor de modelos denominado *CreditScoreModel* (ver **Fig. 10**). Cada versión registrada tiene encadenado el modelo de clasificación y el de transformación (*StandarScaler*), que sale de la etapa de preprocesamiento. Esto último se hace para simplifica el proceso de inferencia, reduciendo el uso de código adicional para normalizar los nuevos datos de entrada.

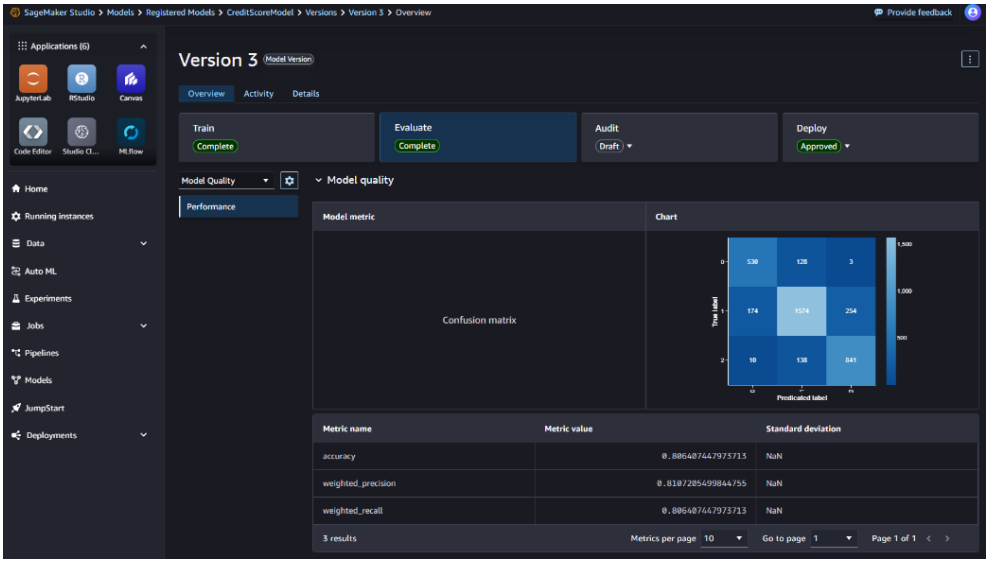

Fig. 10 Tercera versión del modelo de clasificación crediticia

3. DESPLIEGUE

En esta última etapa se ha desplegado un *endpoint* que permite la inferencia en tiempo real del modelo.

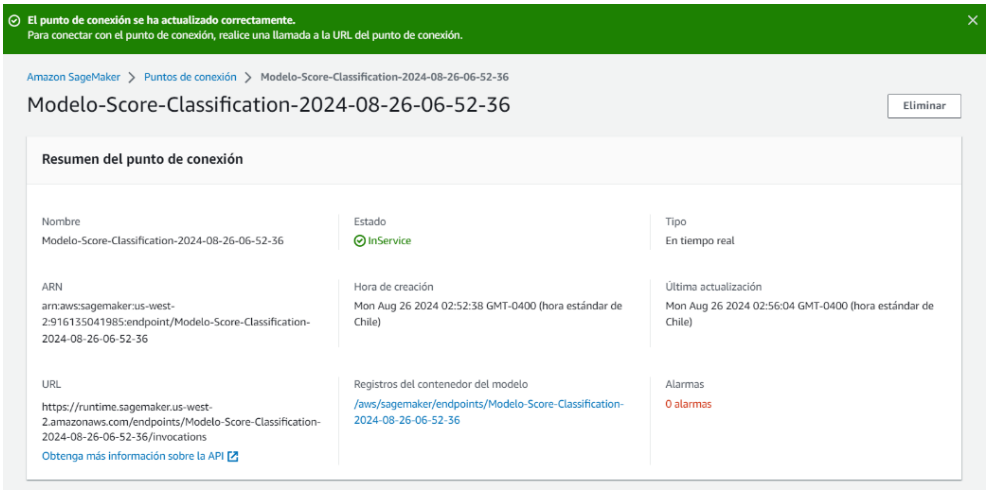

Fig. 11 Endpoint en servicio: Modelo Score Classification

Otra modalidad de despliegue es la *inferencia batch*[14]. Para llevarla a cabo, se puede emplear la clase *Transformer* especificando tanto la ubicación del modelo como la salida de los datos inferidos.

[14] *Inferencia batch:* inferencia por lotes.

IV. CONCLUSIONES

Para concluir, se pueden destacar tres aspectos principales de este desarrollo: la calidad de los datos, la elección del modelo y, finalmente, la puesta en producción del modelo.

En definitiva, contar con un conjunto de datos de calidad es esencial para garantizar la confiabilidad y coherencia en los resultados de cualquier modelo. Aunque existen diversas herramientas para limpiar y mejorar las características de las variables predictoras, es crucial que la entrada de datos se efectúe siguiendo un estándar de calidad, comprendido y aplicado por quienes gestionan y procesan nuevos productos crediticios. Debido a la alta suciedad del conjunto de datos, se recomienda que el equipo directivo establezca una estrategia de gobernanza de datos, lo cual contribuirá a optimizar el rendimiento de los modelos y a disminuir los tiempos y costos asociados al desarrollo y puesta en producción.

Aunque las redes neuronales ofrecen gran capacidad de aprendizaje y flexibilidad, su falta de interpretabilidad y elevado costo computacional son factores que le juegan en contra al momento de la selección. En nuestro análisis, comparamos una red neuronal con un algoritmo de aprendizaje automático y ambos tuvieron un 88% de precisión. Sin embargo, optamos por el modelo más simple, *Random Forest*, debido a su simplicidad y mejor interpretación de resultados. Una posible mejora en el rendimiento del resultado podría darse al utilizar alguna técnica de *stacking* para combinar ambos modelos y mejorar así las predicciones.

Finalmente, la fase de despliegue es donde el valor teórico del modelo se convierte en beneficios prácticos para la organización. Como pudimos ver, esta etapa presenta desafíos significativos, que van desde la selección de la arquitectura, hasta como aseguramos la sostenibilidad y calidad de los resultados para la organización. Es aquí donde muchas organizaciones, tanto grandes como pequeñas, enfrentan grandes dificultades. Para llevar a productivo este u otros modelos se recomienda utilizar servicios cloud totalmente gestionados como *Azure ML*, *Amazon Sagemaker* o *IBM Cloudpack*. Los cuales permiten que los equipos de desarrollo desplieguen modelos de manera sencilla, manteniendo la trazabilidad de los resultados a través de las herramientas de MLOps que estas plataformas ofrecen, reduciendo además los tiempos y costos de mantención de los modelos.

V. BIBLIOGRAFÍA

[1] N. Gift y A. Deza, Practical MLOps, 2021.

[2] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2019.

[3] M. Ekman, Learning Deep Learning, NVIDIA, 2021.

[4] S. Molin, Hands-On Data Analysis with Pandas, 2021.

[5] J. Arvin Lat, Machine Learning with Amazon SageMaker Cookbook, 2022.

[6] J. Simon, Learn Amazon SageMaker, 2021.

[7] P. Bruce y A. Bruce , Practical Statistics for Data Scientists, 2020.

VI. ANEXOS

1. DESCRIPCION DE VARIABLES

TABLA 3 Descripción general de variables

ID	Variable	Tipo	Descripción Inglés	Descripción Español
1	ID	string	Unique identification of an entry	Identificador único de registro
2	Customer_ID	string	Unique identification of a person	Identificador único de cliente
3	Month	string	Month of the year	Mes del año
4	Name	string	Name of a person	Nombre del cliente
5	Age	string	Age of the person	Edad del cliente
6	SSN	string	Social security number of a person	Número de Seguro Social
7	Occupation	string	Occupation of the person	Ocupación o título del cliente
8	Annual_Income	string	Annual income of the person	Ingreso anual del cliente
9	Monthly_Inhand_Salary	float	Monthly base salary of a person	Salario base mensual del cliente
10	Num_Bank_Accounts	int	Number of bank accounts a person holds	Número de cuentas bancarias que posee el cliente
11	Num_Credit_Card	int	Number of other credit cards held by a person	Número de tarjetas de crédito que posee el cliente.
12	Interest_Rate	int	Interest rate on credit card	Tasa de interés asociada a las tarjetas de crédito
13	Num_of_Loan	string	Number of loans taken from the bank	Número de préstamos que tiene con el banco
14	Type_of_Loan	string	Types of loan taken by a person	Tipo o categoría del préstamo
15	Delay_from_due_date	int	Average number of days delayed from the payment date	Retraso promedio en días en el pago desde la fecha de vencimiento
16	Num_of_Delayed_Payment	string	Average number of payments delayed by a person	Número promedio de pagos retrasados
17	Changed_Credit_Limit	string	The percentage change in credit card limit	Porcentaje de cambio en el limite de la tarjeta de crédito
18	Num_Credit_Inquiries	float	Number of credit card inquiries	Número de consultas de crédito realizadas por el cliente
19	Credit_Mix	string	Classification of the mix of credits	Variedad de tipos de crédito en el perfil financiero del cliente
20	Outstanding_Debt	string	Remaining debt to be paid (in USD)	Deuda total pendiente del cliente en dólares
21	Credit_Utilization_Ratio	float	Utilization ratio of credit card	Razón de utilización de la tarjeta de crédito
22	Credit_History_Age	string	Age of credit history of the person	Antigüedad del historial crediticio del cliente.
23	Payment_of_Min_Amount	string	Represents whether only the minimum amount was paid by the person	Comportamiento de pago con respecto al monto mínimo adeudado
24	Total_EMI_per_month	float	Monthly EMI (Equated Monthly Installments) payments (in USD)	Total de Cuotas Mensuales Equivalentes pagadas
25	Amount_invested_monthly	string	Monthly amount invested by the customer (in USD)	Cantidad invertida por el cliente mensualmente
26	Payment_Behaviour	string	Payment behavior of the customer (in USD)	Comportamiento de pagos del cliente en USD
27	Monthly_Balance	string	Monthly balance amount of the customer (in USD)	Saldo mensual en las cuentas financieras del cliente
28	Credit_Score	string	Represents the bracket of credit score (Poor, Standard, Good)	Representa el rango de puntuación crediticia (Pobre, Estándar, Buena)

2. ERRORES Y CORRECCIONES

TABLA 4 Errores y correcciones

ID	Variable	Error Detectado	Transformación	
			Tipo	**Corrección**
1	ID	prefijo 0x	string	eliminada
2	Customer_ID	prefijo CUS_0	string	eliminada
3	Month	Sólo tiene 8 meses	string	sin intervención
4	Name		string	eliminada
5	Age	valores terminados en '_'	int	limpieza
6	SSN	valores '#F%$D@*&8'	string	remplazo
7	Occupation	valores '_______'	string	reemplazo
8	Annual_Income	valores terminados en '_'	float	limpieza
9	Monthly_Inhand_Salary		float	sin intervención
10	Num_Bank_Accounts		int	sin intervención
11	Num_Credit_Card		int	sin intervención
12	Interest_Rate		int	sin intervención
13	Num_of_Loan	valores terminados en '_'	int	limpieza
14	Type_of_Loan	cadena string contiene valores duplicados	string	limpieza
15	Delay_from_due_date		int	sin intervención
16	Num_of_Delayed_Payment	valores terminados en '_'	int	limpieza
17	Changed_Credit_Limit	valores '_'	float	reemplazo
18	Num_Credit_Inquiries		float	sin intervención
19	Credit_Mix	valores '_'	string	reemplazo
20	Outstanding_Debt	valores terminados en '_'	float	limpieza
21	Credit_Utilization_Ratio		float	sin intervención
22	Credit_History_Age	se puede convertir a flotante	string	conversión
23	Payment_of_Min_Amount	valores NM	string	sin intervención
24	Total_EMI_per_month		float	sin intervención
25	Amount_invested_monthly	valores '__10000__'	float	reemplazo
26	Payment_Behaviour	valores *!@9#%8*	string	reemplazo
27	Monthly_Balance	valor '__-333333333333333333333333333__'	float	reemplazo
28	Credit_Score		string	sin intervención

3. ANÁLISIS DESCRIPTIVO DE VARIABLES NUMÉRICAS

TABLA 5 Análisis descriptivo de variables numéricas

Variable	Registros Validos	Missings (%)	Media	Desviación std	Percentil			Valor Mín.	Valor Máx.	coeficiente de variación	Asimetría	% Outliers
					25%	50%	75%					
Age	100000	0,00	110,65	686,24	24	33	42	-500	8698	6,20	9,21	2,78
Annual_Income	100000	0,00	176415,7	1429618,05	19457,5	37578,61	72790,92	7005,93	24198062	8,10	12,51	2,78
Monthly_Inhand_Salary	84998	15,00	4194,17	3183,69	1625,57	3093,75	5957,45	303,65	15204,63	0,76	1,13	0,00
Num_Bank_Accounts	100000	0,00	17,09	117,4	3	6	7	-1	1798	6,87	11,20	1,32
Num_Credit_Card	100000	0,00	22,47	129,06	4	5	7	0	1499	5,74	8,46	2,27
Interest_Rate	100000	0,00	72,47	466,42	8	13	20	1	5797	6,44	9,01	2,03
Num_of_Loan	100000	0,00	3,01	62,65	1	3	5	-100	1496	20,81	15,98	4,35
Delay_from_due_date	100000	0,00	21,07	14,86	10	18	28	-5	67	0,71	0,97	4,00
Num_of_Delayed_Payment	92998	7,00	30,92	226,03	9	14	18	-3	4397	7,31	14,31	0,00
Changed_Credit_Limit	97909	2,09	10,39	6,79	5,32	9,4	14,87	-6,49	36,97	0,65	0,64	0,00
Num_Credit_Inquiries	98035	1,97	27,75	193,18	3	6	9	0	2597	6,96	9,79	0,00
Outstanding_Debt	100000	0,00	1426,22	1155,13	566,07	1166,16	1945,96	0,23	4998,07	0,81	1,21	5,27
Credit_Utilization_Ratio	100000	0,00	32,29	5,12	28,05	32,31	36,5	20	50	0,16	0,03	0,00
Credit_History_Age	90970	9,03	18,43	8,31	12	18,25	25,17	0,08	33,67	0,45	-0,05	0,00
Total_EMI_per_month	100000	0,00	1403,12	8306,04	30,31	69,25	161,22	0	82331	5,92	7,10	6,80
Amount_invested_monthly	91216	8,78	195,54	199,56	72,24	128,95	236,82	0	1977,33	1,02	2,56	0,00
Monthly_Balance	98791	1,21	402,55	213,93	270,11	336,73	470,26	0,01	1602,04	0,53	1,60	0,00

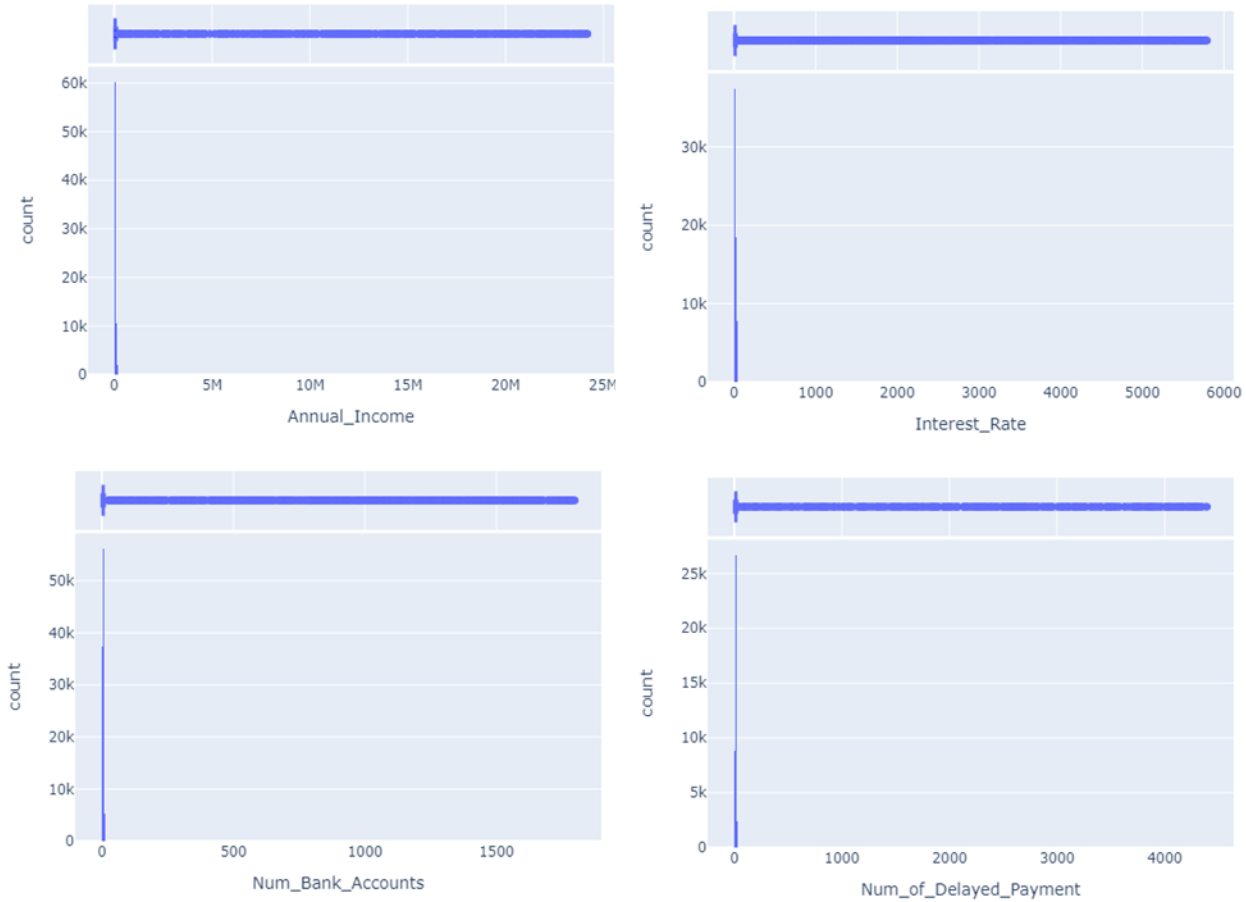

Fig. 12 Variables con *outliers* identificados durante el análisis descriptivo

4. ANÁLISIS DESCRIPTIVO VARIABLES CATEGÓRICAS

TABLA 6 Análisis descriptivo variables categóricas

Variable	Registros Validos	Missings (%)	Valores únicos	Valor más Frecuente	Frecuencia
Month	100000	0,00	8	january	12500
Occupation	100000	0,00	16	other	7062
Type_of_Loan	100000	0,00	3533	not specified	12816
Credit_Mix	79805	20,20	3	standard	36479
Payment_of_Min_Amount	100000	0,00	3	yes	52326
Payment_Behaviour	92400	7,60	6	low_spent_small_value_payments	25513

TABLA 7 Resultados antes/después de tramificación

Ranking Inicial	Feature	Tipo	Importance	Tramificado	Metrica	Ranking Obtenido[15]
6	Num_Credit_Card	numeric	0,042334	Si	indice	17
7	Num_Bank_Accounts	numeric	0,042107	Si	indice	10
9	Num_of_Delayed_Payment	numeric	0,038443	Si	indice	10
11	Monthly_Balance	numeric	0,033789	Si	indice	20
12	Monthly_Inhand_Salary	numeric	0,03372	Si	indice	21
13	Age	numeric	0,033707	Si	indice	21
14	Amount_invested_monthly	numeric	0,033201	Si	indice	20
15	Num_of_Loan	numeric	0,032884	Si	indice	19
16	Annual_Income	numeric	0,03253	Si	woe/indice	21
17	Credit_Utilization_Ratio	numeric	0,031424	Si	indice	20
18	Total_EMI_per_month	numeric	0,030743	Si	indice	20

[15] Se utilizo el *Feature Importance* de modelo como ranking de comparativa.

5. RESULTADO MODELOS MACHINE LEARNING

TABLA 8 Resumen grilla de modelos ML

Algoritmo	Librería	Precisión Media	Desviación Std
LogisticRegression	sklearn	0.698861	0.002786
GaussianNB	sklearn	0.691747	0.002441
XGBClassifier	xgboost	0.827969	0.001932
RandomForestClassifier	sklearn	0.880595	0.001469
DecisionTreeClassifier	sklearn	0.809136	0.001321
SGDClassifier	sklearn	0.691564	0.006093
OneVsOneClassifier	sklearn	0.880862	0.001457

6. TUNING DE MODELOS: RESULTADOS OBTENIDOS

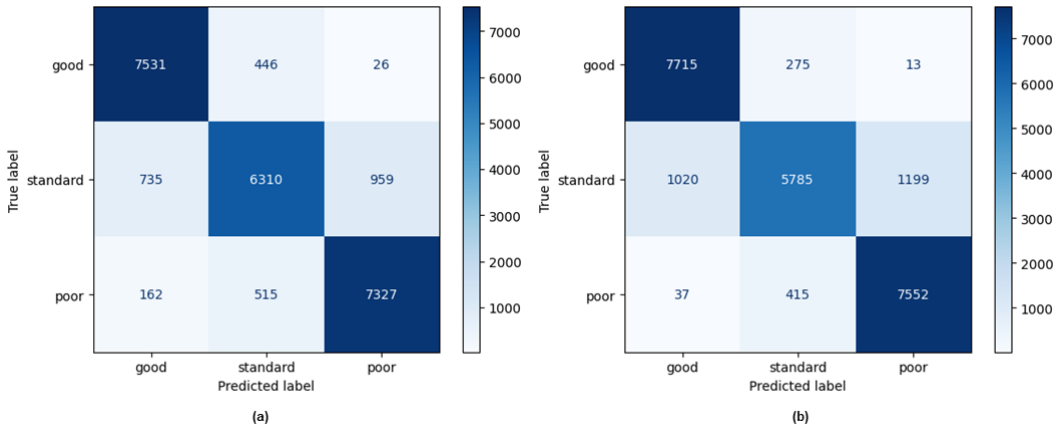

Fig. 13 Matriz confusión (a) Random Forest (b) Red Neuronal

TABLA 9 Tuning: comparativa de modelos

	Random Forest			Red Neuronal			
	precision	**recall**	**f1-score**	**precision**	**recall**	**f1-score**	**support**
good	0.89	0.94	0.92	0.88	0.96	0.92	8003
standard	0.87	0.79	0.83	0.89	0.72	0.80	8004
poor	0.88	0.92	0.90	0.86	0.94	0.90	8004
accuracy			0.88			0.88	24011
macro avg	0.88	0.88	0.88	0.88	0.88	0.87	24011
weighted avg	0.88	0.88	0.88	0.88	0.88	0.87	24011

7. PRODUCTIVIZACIÓN

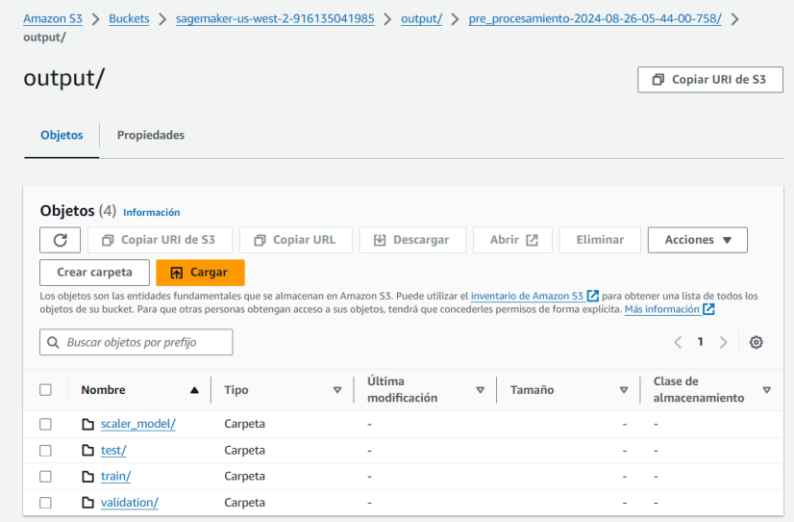

Fig. 14 Salida a *Bucket S3* de la etapa de preprocesamiento

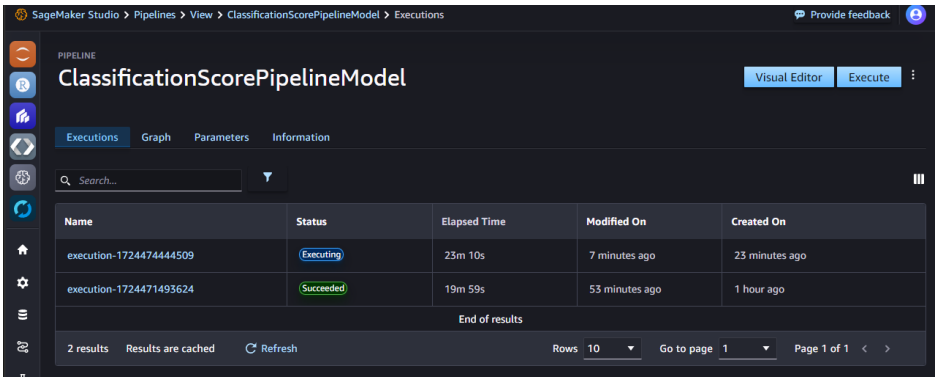

Fig. 15 Ejecución de los *SageMaker Pipelines*

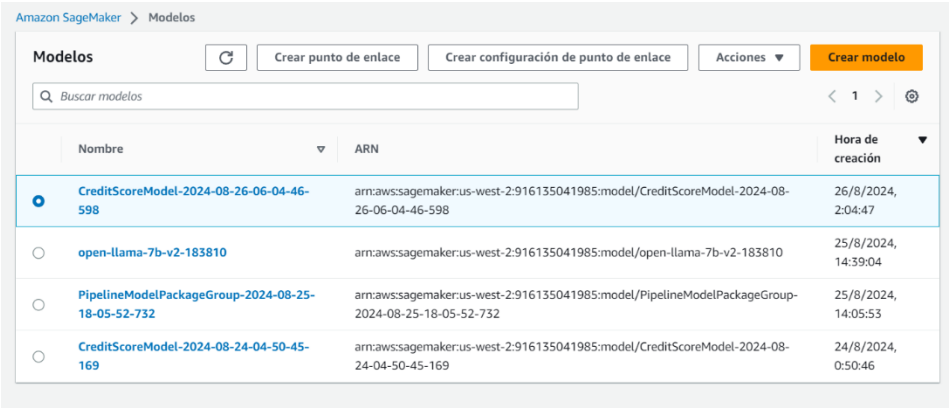

Fig. 16 Modelos disponibles para producción (creación de *endpoint*)