

MÁSTER UNIVERSITARIO EN BIG DATA & BUSINESS ANALYTICS

TRABAJO FINAL DE MÁSTER

“Modelos de Machine Learning para la Predicción de Impago en el Sector
Financiero”

AUTORES:

Eva Meneses Soto, evamenso1@gmail.com
Ana María Quintero Estevez, aquinteroe@student.eae.es
María Regina Meiners de Alba, mmeinersd@student.eae.es
(Grupo nº 5)

Tutor:

Prof. D. José Ramón Vallejo Rodrigo, jvallejorodrigo@campus.eae.es

Madrid, 2025

RESUMEN

El presente Trabajo de Fin de Máster (TFM) se enfoca en el modelado predictivo del riesgo de crédito utilizando un conjunto de datos de préstamos personales. Nuestro objetivo principal es predecir la probabilidad de incumplimiento (default) de pago de los préstamos utilizando las características de los solicitantes. La metodología aplicada incluyó un análisis exploratorio de datos (EDA) y una rigurosa ingeniería de variables, con especial atención al tratamiento de valores nulos. De igual manera, se construyeron y evaluaron múltiples modelos de clasificación para seleccionar el algoritmo con mejor desempeño predictivo.

En este TFM se busca demostrar que una adecuada ingeniería de variables, sustentada en un tratamiento apropiado de los datos, incluidos los valores nulos, nos ayuda a desarrollar modelos de predicción de impagos más sólidos, aunque no todos los ajustes aplicados resulten en mejoras directas del desempeño. Además, se incluye el uso de herramientas de interpretabilidad como SHAP para darnos transparencia al funcionamiento del modelo y facilitar la comprensión del impacto que tiene cada variable en las predicciones. De igual manera, se incorpora un dashboard interactivo orientado a negocio que traduce los resultados del modelo en indicadores y alertas accionables, para la toma de decisiones más eficientes en la gestión del riesgo crediticio.

Palabras clave: Predicción de impago; Machine Learning; Interpretabilidad; EDA; Monitoreo

ABSTRACT

This Master's Thesis focuses on predictive modeling of credit risk using a dataset of personal loans. The main objective is to predict the probability of default on loans based on the characteristics of the applicants. The applied methodology included an exploratory data analysis (EDA) and a rigorous feature engineering process, with special attention to the treatment of missing values and the selection of the most relevant attributes. Multiple classification models were built and evaluated to determine the algorithm with the best predictive performance.

This Thesis aims to demonstrate that sound feature engineering, supported by consistent data treatment, including the handling of missing values, contributes to the development of more robust default prediction models, even if not all adjustments directly improve performance. Furthermore, interpretability tools such as SHAP are incorporated to bring transparency to the model's functioning and to facilitate understanding of how each variable impacts the predictions. An interactive business-oriented dashboard is also included, translating model results into actionable indicators and alerts to support more effective operational decision-making in credit risk management.

Key-words: Default prediction; Machine Learning; Interpretability; Exploratory Data Analysis; Monitoring

ÍNDICE

I. PARTE INTRODUCTORIA

1. INTERÉS DEL ESTUDIO.....	11
2. FINES Y OBJETIVOS.....	11
3. ESTADO DE LA CUESTIÓN.....	12
3.1. Definición de crédito bancario.....	12
3.2. Calificación crediticia.....	12
3.3. Revisión de la literatura de Default Bancario y Machine Learning.....	13
4. METODOLOGÍA.....	14

II. PARTE GENERAL

5. ARQUITECTURA.....	21
5.1. Origen de datos.....	21
5.2. Almacenamiento.....	21
5.3. Tratamiento de datos.....	22
5.4. Modelos.....	23
5.5. Monitorización y visualización.....	25
6. DESARROLLO Y ANÁLISIS.....	26
6.1. Análisis Exploratorio (EDA).....	26
6.1.1. Librerías Python.....	26
6.1.2. Variables.....	28
6.1.3. Preprocesamiento preliminar.....	29
6.1.4. Análisis univariante.....	30
6.1.5. Análisis Bivariante.....	40
6.2. Análisis de los valores nulos.....	57
6.3. Matriz de Correlaciones.....	63
6.4. Limpieza y tratamiento de datos.....	66
6.5. Modelos.....	69

6.5.1. Librerías en R.....	69
6.5.2. Recodificación de variables.....	70
6.5.3. División de datos.....	71
6.5.4. Métodos de Preprocesamiento para modelos.....	71
6.5.5. Validación Cruzada.....	77
6.5.6. Modelos Lineales Penalizados.....	77
6.5.7. Modelos No Lineales.....	80
6.5.8. Comparativa Modelo Lineal Seleccionado vs Modelo No Lineal.....	85
6.5.9. Interpretabilidad del modelo seleccionado.....	87
6.5.10. Comparativa modelo sin historical default.....	90
6.5.11. Interpretabilidad del modelo seleccionado sin historical_default.....	94
7. MONITOREO Y VISUALIZACIÓN DE DATOS.....	98
8. TRANSPARENCIA Y CUMPLIMIENTO NORMATIVO EN LA TOMA DE DECISIONES AUTOMATIZADAS.....	102
9. CONCLUSIONES.....	103
10. BIBLIOGRAFÍA.....	106
11. ANEXOS.....	108

ÍNDICE DE GRÁFICOS

Gráfico 6.1.1. - Boxplot customer_age.....	32
Gráfico 6.1.2 - Distribución de los Ingresos de los Clientes.....	33
Gráfico 6.1.3 - Distribución logarítmica de los Ingresos de los Clientes.....	33
Gráfico 6.1.4 - Distribución de los Ingresos de los Clientes (P99).....	33
Gráfico 6.1.5 - Boxplot de la duración del empleo.....	34
Gráfico 6.1.6 - Distribución del Importe del Préstamo.....	35
Gráfico 6.1.7 - Distribución logarítmica del Importe del Préstamo.....	35
Gráfico 6.1.8 - Distribución del Importe del Préstamo (P99).....	35
Gráfico 6.1.9 - Distribución del Tipo de Interés.....	36
Gráfico 6.1.10 - Boxplot del Tipo de Interés.....	36
Gráfico 6.1.11 - Frecuencia por Plazo de Préstamo.....	37
Gráfico 6.1.12 - Densidad del Historial crediticio.....	37
Gráfico 6.1.13- Frecuencia del Tipo de Vivienda.....	38
Gráfico 6.1.14 - Frecuencia del Motivo del Préstamo.....	38
Gráfico 6.1.15 - Frecuencia de Calificación Crediticia.....	39
Gráfico 6.1.16 - Proporción de Clientes con y sin Historial Crediticio.....	39
Gráfico 6.1.17 - Distribución del Estado del Préstamos (variable objetivo).....	40
Gráfico 6.1.18 - Edad de los clientes según el estado del préstamo.....	42
Gráfico 6.1.19 - Duración del Empleo según el estado del préstamo.....	44
Gráfico 6.1.20 - Ingresos según el estado del préstamo.....	46
Gráfico 6.1.21 - Monto del Préstamo según el estado del préstamo (P99).....	47
Gráfico 6.1.22 - Monto del Préstamo según el estado del préstamo (P99).....	48
Gráfico 6.1.23 - Duración del Préstamo según el estado del préstamo (P99).....	50
Gráfico 6.1.24 - Duración del Préstamo según el estado del préstamo (P99).....	51
Gráfico 6.1.25 - Motivo del Préstamo según el estado del préstamo (P99).....	53
Gráfico 6.1.26 - Calificación del Préstamo según el estado del préstamo (P99).....	55

Gráfico 6.1.27 - Proporción de Default según el historial de impago.....	56
Gráfico 6.2.1 - Relación entre nulos con historial_default.....	58
Gráfico 6.2.2 - Distribución de customer_age por missing_hist_default.....	59
Gráfico 6.2.3 - Distribución de cred_hist_length por missing_hist_default.....	60
Gráfico 6.2.4 - Proporción de loan_grade por missing_hist_default.....	61
Gráfico 6.2.5 - Distribución de tasa de interés por missing_hist_default.....	61
Gráfico 6.2.6 - Violin plot de la relación de tasa de interés y calificación crediticia.....	62
Gráfico 6.3.1 - Matriz de Correlaciones de variables numéricas.....	64
Gráfico 6.3.2 - Matriz V de Cramer.....	65
Gráfico 6.3.3 - Matriz de Correlaciones de variables numéricas y categorías convertidas..	66
Gráfico 6.5.1 - SHAP Summary Plot Variables Categóricas.....	87
Gráfico 6.5.2 - SHAP Summary Plot Variables Numéricas.....	88
Gráfico 6.5.3 - SHAP Bar Plot.....	89
Gráfico 6.5.4 - SHAP Bar Plot sin historical_default.....	94
Gráfico 6.5.5 - SHAP Summary Plot Variables Numéricas sin historical_default.....	95
Gráfico 6.5.6 - Dependencia SHAP Loan Amount.....	95

ÍNDICE DE TABLAS

Tabla 5.1.1 - Tamaño de la base de datos.....	21
Tabla 6.1.1- Librerías Python.....	27
Tabla 6.1.2 - Ajustes de tipo y formato aplicados a las variables del Loan Dataset.....	30
Tabla 6.1.3 - Descripción de la variables.....	31
Tabla 6.1.4 - Valores erróneo eliminados.....	41
Tabla 6.1.5 - Media y desviación típica de la edad por default y no default.....	42
Tabla 6.1.6 - T-Student de edad por default y no default.....	43
Tabla 6.1.7 - Media y desviación típica de la duración del empleo según default y no default	
44	
Tabla 6.1.8 - T-Student de la duración del empleo según default y no default.....	45
Tabla 6.1.9 - Media y desviación típica de los ingresos según default y no default.....	46
Tabla 6.1.10 - T-Student del Ingreso según default y no default.....	46
Tabla 6.1.11 - Media y desviación típica del Monto del Préstamo según default y no default.	
48	
Tabla 6.1.12 - T-Student de la Monto del Préstamo según default y no default.....	48
Tabla 6.1.13- Media y desviación típica de la Tasa de Interés o según default y no default	
49	
Tabla 6.1.14 - T-Student de la Tasa de Interés según default y no default.....	49
Tabla 6.1.15 - Media y desviación típica de la Tasa de Interés o según default y no default...	
50	
Tabla 6.1.16 - T-Student de la Tasa de Interés según default y no default.....	50
Tabla 6.1.17 - Frecuencia del Tipo de Vivienda según default y no default.....	52
Tabla 6.1.18 - Chi-Cuadrado del Tipo de Vivienda según default y no default.....	52
Tabla 6.1.19 - Frecuencia del Motivo del Préstamo según default y no default.....	53
Tabla 6.1.20 - Chi-Cuadrado del Tipo de Vivienda según default y no default.....	54
Tabla 6.1.21 - Frecuencia de la Calificación del Préstamo según default y no default.....	55
Tabla 6.1.22 - Chi-Cuadrado de la Calificación del Préstamo según default y no default...	
56	

Tabla 6.1.23- Chi-Cuadrado del historial de impago según default.....	57
Tabla 6.4.1 - Distribución de los Valores Nulos.....	68
Tabla 6.5.1- Librerías R.....	70
Tabla 6.5.2 - Ajustes de tipo aplicados a las variables del Dataset en R.....	71
Tabla 6.5.3 - Preprocesamiento modelos.....	72
Tabla 6.5.4- Grid Modelos Lineales Penalizados.....	77
Tabla 6.5.5- Hiperparámetros Modelos Lineales Penalizados.....	79
Tabla 6.5.6 - Resultados Modelos Lineales Penalizados.....	80
Tabla 6.5.7 - Grid Modelos No Lineales.....	81
Tabla 6.5.8 - Hiperparámetros Modelos No Lineales.....	83
Tabla 6.5.9 - Resultados Modelos No Lineales.....	85
Tabla 6.5.10- Resultados Modelos No Lineal vs Lineal Penalizado.....	86
Tabla 6.5.11- Resultados Random Forest Conjunto de prueba.....	87
Tabla 6.5.12- Hiperparámetros Modelos Sin Historical Default.....	92
Tabla 6.5.13 - Resultados Modelos Lineales Penalizados sin historical_default.....	92
Tabla 6.5.14- Resultados Modelos No Lineales.....	93
Tabla 6.5.15- Comparativa Modelos Lineales vs No Lineal sin historical_default.....	94
Tabla 6.5.16- Resultados Random Forest Conjunto de prueba sin historical_default.....	94

ABREVIATURAS

- AUC – Area Under the Curve
- BBDD – Bases de Datos
- BI – Business Intelligence
- CSV – Comma-Separated Values
- EDA – Exploratory Data Analysis (Análisis Exploratorio de Datos)
- KNN – k-Nearest Neighbours
- KPI – Key Performance Indicator
- ML – Machine Learning (Aprendizaje Automático)
- MARS – Multivariate Adaptive Regression Splines
- MIT – Massachusetts Institute of Technology
- ROC – Receiver Operating Characteristic
- ROC-AUC – Métrica combinada ROC + AUC
- SHAP – SHapley Additive exPlanations
- SVM – Support Vector Machine
- TFM – Trabajo de Fin de Máster
- DAX – Data Analysis Expressions

I.- PARTE INTRODUCTORIA

1. INTERÉS DEL ESTUDIO

La detección de impago de créditos bancarios es una problemática que afecta tanto a las instituciones financieras como a la propia economía de un país, es por esto que es muy importante poder realizar su detección a tiempo. Por lo tanto, ambos están en búsqueda de encontrar la mejor manera de predecir dichas situaciones y de esta manera evitar las repercusiones que ello conlleva.

El interés de este TFM radica en poder crear modelos estadísticos que sean capaces de detectar de manera adecuada la clasificación de los clientes a la hora de otorgar un crédito bancario. Se propone probar distintos modelos predictivos y aplicar técnicas de interpretabilidad de modelos para otorgar la mejor solución posible para dicho problema.

La creciente complejidad y los riesgos asociados al incumplimiento de crédito hacen urgente el desarrollo de herramientas nuevas y explicables para mejorar la gestión del riesgo. Este trabajo es necesario para contribuir a la estabilidad del sistema financiero, proporcionando modelos que sean precisos y también comprensibles para las partes interesadas.

2. FINES Y OBJETIVOS

A lo largo de este Trabajo de Fin de Máster, abordaremos el problema del incumplimiento de créditos bancarios personales, con el objetivo principal de optimizar la gestión del riesgo crediticio y facilitar la toma de decisiones estratégicas de negocio. Para ello, el desarrollo y evaluación de modelos estadísticos y de Machine Learning será el medio que nos permitirá identificar, anticipar y gestionar el riesgo de impago de manera más eficiente que con los enfoques tradicionales.

Para ello, en este trabajo, nos centraremos en tres partes principales:

- **Comparación de modelos predictivos lineales y no lineales:** consideraremos los modelos Lasso, Ridge y Elastic-Net que permitirán analizar y explicar la importancia y la relación de las distintas variables de manera lineal. Además, gracias a la implementación posterior de algoritmos estadísticos más avanzados y de aprendizaje automático como son Bagging, KNN, Random Forest, SVM y MARS, podremos captar las relaciones no lineales para aumentar la precisión de la predicción.
- **Interpretabilidad de los modelos:** utilizaremos la herramienta de SHAP para analizarla con el objetivo de cuantificar el efecto de cada variable en la predicción.
- **Monitoreo continuo y seguimiento a través de dashboards de negocio:** desarrollaremos dashboards interactivos para visualizar los principales KPIs de negocio relacionados con el riesgo crediticio, así como indicadores de desempeño del modelo y alertas ante posibles cambios en el dataset. De esta manera, se asegura un ciclo de mejora continua y una alineación permanente con los objetivos estratégicos de la organización.

3. ESTADO DE LA CUESTIÓN

3.1. Definición de crédito bancario

Un crédito bancario es un acuerdo financiero en el que una entidad, normalmente un banco, pone a disposición de un cliente una cantidad de dinero con la condición de que este lo devuelva en un futuro con los intereses generados (Crédito bancario - concepto y crédito para empresas).

3.2. Calificación crediticia

La calificación crediticia es una medida del riesgo de impago de una persona, empresa o gobierno. Consiste en medir la capacidad de pago del deudor en función de diversas

características como su historial financiero, el nivel de endeudamiento y la estabilidad económica del solicitante del préstamo.

La calificación crediticia suele estar otorgada por agencias especiales que se dedican a ello o por los propios bancos, cada una teniendo su propia escala. Generalmente las calificaciones más altas están representadas por una o varias "A" y significan que tiene una baja probabilidad de impago y las calificaciones más bajas por "D" que representan un alto riesgo de incumplimiento (*Calificación crediticia – qué es y para qué sirve*, 2015).

3.3. Revisión de la literatura de Default Bancario y Machine Learning

En este apartado buscamos exponer la literatura encontrada donde se tratan temas de default bancario y machine learning.

Para empezar, la investigación realizada por José Manuel García y Walther Nagun Torres en 2023 desarrolla un modelo de aprendizaje automático con el objetivo de predecir el riesgo de impago en la cartera de consumo fiduciario de una institución financiera en Honduras. Ellos, utilizan los modelos supervisados Random Forest y XGboost para lograr identificar con una presión del 60% la probabilidad de impago.

En el artículo de Tanasuica se probaron varios algoritmos de aprendizaje automático supervisado para identificar al que clasificaba de mejor manera la base de datos de impago de empresas rumanas proveniente de una institución financiera no bancaria, posteriormente se utilizaron algoritmos no supervisados para identificar el grupo comparable al conjunto de empresas con default bancaria. Por último se utilizó el método SHAP para interpretabilidad de los algoritmos de ML lo cual llevó a identificar las señales de alerta claves que sirven como predictores de riesgo financiero (Tanasuica Zotic, 2024).

En la publicación de Qingyang se propone un modelo de aprendizaje supervisado y se utiliza SHAP para explicar la evaluación crediticia para comprender las contribuciones de las variables en el modelo y sus interacciones (Q. Xu et al., 2024).

En la investigación de Gramegna se toma información sobre PYMEs italianas, se entrena un modelo No Supervisado utilizando Random Forest y se mide que tan bien predicen

ambos modelos utilizando AUROC. Se aplica tanto SHAP como LIME para obtener la explicabilidad del modelo y se pasa a aplicar modelos no supervisados como lo son K-means y Clustering Espectral con los pesos de ambos métodos. Finalmente, se comparan ambos métodos LIME y SHAP para ver cuál proporciona mejores grupos y predicciones (Bussmann et al., 2025).

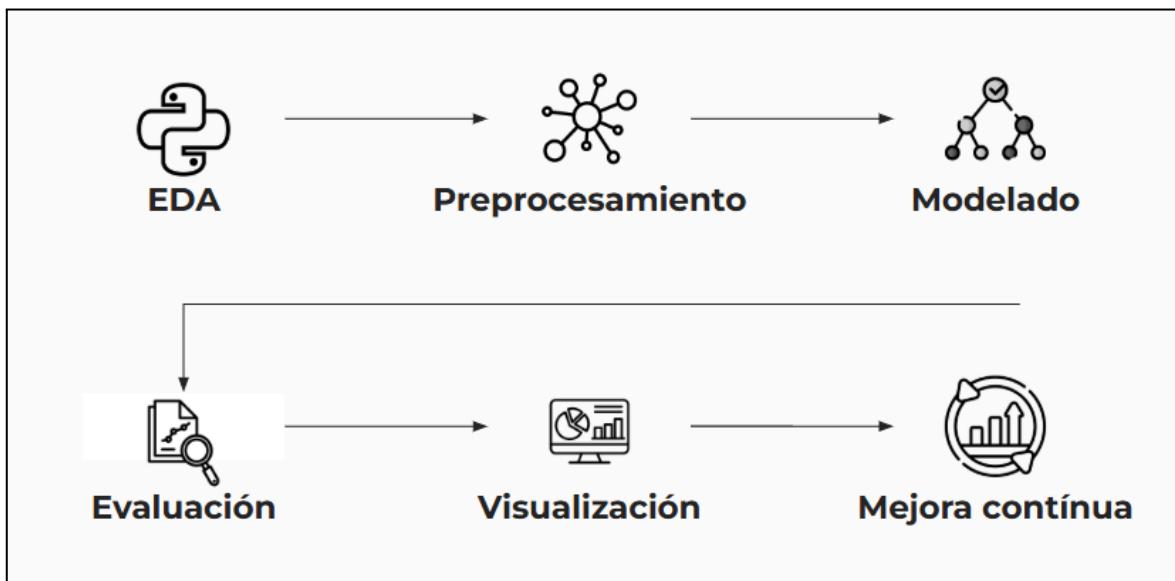
4. METODOLOGÍA

La metodología seguida para el presente trabajo es la siguiente:

- Análisis exploratorio de datos (EDA): en primer lugar, se realizó una exploración profunda del conjunto de datos con el objetivo de detectar errores, valores atípicos y patrones relevantes. Se utilizaron estadísticas descriptivas y visualizaciones gráficas para comprender la distribución de las variables y su relación con el objetivo de estudio (impago).
- Preprocesamiento: a partir del EDA, se procedió al tratamiento de los datos para prepararlos para la etapa del modelado.
- Modelado: esta etapa abarca la construcción y entrenamiento de modelos predictivos. Se compararon modelos lineales penalizados (Ridge, Lasso, Elastic Net) y modelos no lineales (Random Forest, Bagging, SVM, KNN, MARS), detallando las estrategias de ajuste de hiperparámetros utilizadas (grids específicos y validación cruzada).
- Evaluación: se comparó el desempeño de los modelos utilizando métricas adecuadas al problema de clasificación, con especial énfasis en la métrica f-measure, dada la importancia de equilibrar precisión y sensibilidad en contextos de riesgo. Además, se aplicaron herramientas de interpretabilidad como SHAP para analizar el impacto individual de las variables en las predicciones.
- Visualización: se desarrollaron visualizaciones orientadas tanto al área de negocio como técnica para realizar el seguimiento.

- Mejora continua: finalmente, se propone un esquema de mantenimiento y actualización del modelo en producción.

Figura 4.1- Metodología seguida



Fuente: elaboración propia

Modelos de predicción

En el presente estudio se estarán probando distintos algoritmos Supervisados de Machine Learning para de esta manera seleccionar el que realiza una mejor predicción sobre nuestra variable target (current_Loan_Default). Los algoritmos entrenados y probados en nuestro modelo de clasificación fueron los siguientes: Lasso, Ridge, Elastic-Net, Bagging, KNN, Random Forest, SVM y MARS. El lenguaje de programación utilizado para construir y probar los modelos fue R al igual que para posteriormente utilizar el modelo de SHAP para realizar la interpretabilidad del algoritmo con el mejor desempeño.

Para el entrenamiento de los modelos, se utilizarán técnicas de validación cruzada para evaluar el rendimiento del modelo y evitar el sobreajuste. Además, se llevará a cabo un ajuste de hiperparámetros mediante Grid Search para identificar la combinación óptima de parámetros que maximice la métrica de rendimiento, como la f1-measure (Adugna et al., 2024).

A continuación se compartirán las características principales de los algoritmos utilizados en el estudio.

Ridge

La regresión Ridge, también llamada regresión Cresta, es un método de estimación de los coeficientes de modelos de regresión múltiple en escenarios en los que las variables independientes están muy correlacionadas.

El algoritmo busca minimizar la siguiente ecuación:

$$RSS_{Ridge} = SSR + \lambda \sum_{i=1}^p \beta_i^2 , \quad SSR = \sum_{j=1}^n (\hat{y}_i - \bar{y})^2$$

Se trata de una regresión penalizada en la que a través del hiperparámetro λ se define el grado de regularización de las variables a mayor λ más se penaliza el tamaño de los coeficientes β_i en relación a la suma de cuadrados residual (SSR). Este tipo de algoritmos al ser penalizados se suelen utilizar cuando se dispone de bases de datos anchas con una gran cantidad de predictores o features muy correlacionados entre ellos. Habitualmente los coeficientes β_i nunca llegan a hacerse cero por lo que la regresión Ridge no realiza selección automática de features o predictores.

Lasso

Al igual que Ridge se trata de una regresión penalizada utilizando el hiperparámetro λ , sin embargo, en este caso la ecuación que se quiere minimizar es la siguiente:

$$RSS_{Lasso} = SSR + \lambda \sum_{i=1}^p |\beta_i| , \quad SSR = \sum_{j=1}^n (\hat{y}_i - \bar{y})^2$$

Con este algoritmo se pueden dar soluciones esquina donde el parámetro β_i estimado se anula por lo que permite realizar selección automática de predictores o features (cuando el $\hat{\beta}_i$ estimado se hace cero).

Elastic-Net

Se trata de un algoritmo que es una combinación de Ridge y Lasso. El propósito de este modelo es minimizar la siguiente función:

$$RSS_{Elastic-Net} = SSR + \lambda_1 \sum_{i=1}^p \beta_i^2 + \lambda_2 \sum_{i=1}^p |\beta_i| , \quad SSR = \sum_{j=1}^n (\hat{y}_i - \bar{y})^2$$

En este caso se cuenta con dos hiperparámetros λ con los cuales se permite realizar selección automática de features como en el algoritmo Lasso y se maneja de manera adecuada con variables predictoras muy correlacionadas como en el algoritmo Ridge.

Bagging

El presente algoritmo, también conocido como agregación de bootstrap, es un método de aprendizaje por conjuntos en el cuál se selecciona una muestra aleatoria de datos de un conjunto con reemplazo y estima árboles de decisión a cada muestra generada. En donde un árbol de decisión genera una partición de datos en subgrupos con valor de respuesta similar. Con el objetivo de lograr ramas lo más homogéneas posible.

El algoritmo sigue la siguiente ecuación:

$$\widehat{f}_{bag} = \frac{1}{b} (\widehat{f}_1(X) + \widehat{f}_2(X) + \dots + \widehat{f}_b(X))$$

Lo que significa que el modelo toma las predicciones individuales y calcula su promedio para obtener la predicción final del modelo combinado.

Random Forest

Este algoritmo es una variante de Bagging el cuál utiliza una cantidad establecida mediante un parámetro de árboles decorrelados. Este algoritmo en cada división del árbol de decisión elige entre un conjunto aleatorio pequeño (`mtry`) de variables independientes para realizar la separación de los datos. La ventaja sobre Bagging es que este algoritmo logra zonas de nuestro espacio de variables que bagging no logra, siendo muy útil para variables altamente

correlacionadas. Cabe destacar que este algoritmo sigue la misma fórmula que bagging con la excepción de la selección de los nodos.

KNN

El algoritmo de vecinos cercanos utiliza la proximidad para hacer clasificaciones y predicciones, busca los K puntos más cercanos a un punto concreto para poder inferir su valor. En este método no se hace ninguna suposición sobre la distribución de las variables independientes. La similaridad entre dos individuos se calcula en función de medidas de distancia, siendo la distancia euclídea la más utilizada:

$$d(x_i, y_j) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2}$$

SVM

Este algoritmo encuentra un hiperplano óptimo que separe las clases en un espacio de múltiples dimensiones. El hiperplano mencionado es seleccionado de manera que la distancia entre los puntos más cercanos (vectores soporte) de ambas clases sea la máxima posible. Si las clases son linealmente separables el algoritmo encuentra el hiperplano para hacer la clasificación, en caso de no serlo, se utilizan funciones kernel para que las observaciones sean linealmente separables en un espacio de mayor dimensión. La función utilizada en el presente algoritmo es la siguiente:

$$f(x^*) = \beta_0 + \sum_{i \in S} \alpha_i K(x^*, x_i)$$

En la cuál S es el conjunto de vectores soporte, α son los coeficientes asociados a dichos vectores y K es la función kernel utilizada.

MARS

El propósito de este algoritmo es utilizar modelos lineales para percibir relaciones no lineales, esto lo realiza mediante cambios de pendiente a través de distintas funciones en

nodos. El algoritmo toma cada punto de los datos como un nodo potencial y elige los puntos que minimizan la métrica objetivo la cual en casos de clasificación puede ser la entropía o el Índice Gini. Después de elegir los nodos se puede realizar una poda en la que se eliminan los nodos que contribuyen en menor medida a la acuracidad de las predicciones para que de esta manera mejorar su capacidad de generalizar y simplificar el modelo. El algoritmo sigue la siguiente función:

$$y_i = \beta_0 + \sum_{i=1}^p \beta_i \cdot h_i(x)$$

En donde $h_i(x)$ son las funciones hinge en las que en caso de que nuestra observación esté dentro del intervalo toman un valor de 1 y en caso contrario el valor es 0. En el caso de β_i este se refiere al coeficiente asociado a las funciones hinge.

Una técnica importante que se utilizará durante la fase de entrenamiento de algoritmos de aprendizaje automático es el tuning de hiperparámetros. Una de las técnicas más comunes para esto es el Grid Search, que busca la mejor combinación de parámetros del modelo dentro de un rango predefinido.

Grid Search simula un rango completo de posibles combinaciones de parámetros y evalúa el rendimiento del modelo para cada conjunto de valores. Para hacer esto, evalúa cada combinación de los parámetros a través de un proceso de validación cruzada. Esto se hace con el objetivo de identificar la combinación de parámetros que produce la mejor métrica de rendimiento, como la ¹f1-measure, en el modelo.

Después de probar todas las combinaciones, el Grid Search proporciona la combinación óptima de parámetros. Finalmente, el modelo se entrena nuevamente utilizando la combinación de parámetros seleccionada (Tanasuica Zotic, 2024).

¹ **f1-measure (f-meas):** mide el equilibrio entre la sensibilidad y la precisión, es decir, entre la probabilidad de que los eventos que predice un modelo como positivos, realmente lo sean y la probabilidad de que un positivo sea detectado de manera correcta por un modelo.

SHAP

Los modelos de aprendizaje automático son descritos como una “caja negra”, es por esto que el modelo de SHAP (SHapley Additive Explanations) nos puede ofrecer la interpretabilidad de estos modelos complejos. El modelo SHAP calcula cuánto cambia la predicción de un modelo cuando se incluye o excluye una variable específica realizando todas las combinaciones posibles de variables y determina el impacto promedio en la predicción final de manera global.

II.- PARTE GENERAL

5. ARQUITECTURA

5.1. Origen de datos

El dataset utilizado en este proyecto fue obtenido de Kaggle, bajo el nombre “Loan Dataset”. Esta base de datos estructurada y relacional, desarrollada por el Massachusetts Institute of Technology (MIT), contiene 32.586 registros y 13 columnas, organizados en un formato tabular que facilita tanto su almacenamiento como su análisis.

Número de Filas	32.586
Número de Columnas	13

Tabla 5.1.1 - Tamaño de la base de datos

Como base de datos relacional, cada columna representa atributos de los clientes, como su edad, ingresos, historial crediticio o estado actual del préstamo, mientras que cada fila corresponde a un cliente único. Esta estructura relacional asegura coherencia, evita redundancias y permite realizar consultas eficientes sobre los datos.

Esta combinación de características convierte al dataset en un recurso sólido y representativo para abordar el problema del default bancario a través de análisis exploratorios y modelización predictiva.

5.2. Almacenamiento

El almacenamiento de datos constituye un pilar fundamental en cualquier proyecto de análisis, ya que permite garantizar la integridad, disponibilidad y trazabilidad de la información utilizada a lo largo del proceso. En nuestro caso, aunque el análisis se ha realizado directamente a partir de un archivo CSV descargado desde Kaggle, consideramos relevante destacar el valor que habría aportado la implementación de una base de datos relacional como PostgreSQL en el desarrollo del trabajo.

Contar con una solución como PostgreSQL habría permitido organizar y gestionar los datos de forma más eficiente y segura, facilitando la actualización, el control de versiones y el acceso centralizado desde distintas herramientas. Dada su alta estabilidad y rendimiento, PostgreSQL resulta especialmente adecuado para trabajar con datasets de tamaño medio como el nuestro, sin comprometer la velocidad de consulta ni la escalabilidad del proyecto.

Además, su integración directa con lenguajes como Python y R, a través de librerías como psycopg2 y RPostgres, habría simplificado el flujo de trabajo al permitir conectarse directamente a la base de datos desde los notebooks de análisis, eliminando la necesidad de archivos intermedios y reduciendo el riesgo de inconsistencias. Asimismo, implementar una política de backups automatizados habría ofrecido una capa adicional de seguridad ante cualquier error de manipulación o fallo del sistema.

En definitiva, aunque por limitaciones prácticas no se ha incorporado una base de datos en este proyecto, reconocemos su utilidad y la mejora sustancial que podría haber supuesto a nivel operativo y analítico. Para futuros desarrollos o despliegues reales de modelos predictivos, contar con un sistema robusto y bien estructurado de almacenamiento es una decisión altamente recomendable

5.3. Tratamiento de datos

Este apartado constituye una parte fundamental en el desarrollo de nuestro Trabajo de Fin de Máster ya que garantiza la calidad y la fiabilidad de la información utilizada en los modelos predictivos. Así, en este estudio realizamos un proceso de limpieza, preprocesamiento y análisis exploratorio con el fin de optimizar el uso de los datos a la hora de hacer la modelización.

Para esto, utilizamos Python y R como herramientas principales gracias a la amplia variedad de librerías que ayudan al manejo, análisis y visualización de datos. Antes de esto, descargamos el archivo CSV desde Kaggle para posteriormente cargarlo en Python:

- *Python en Google Colab*: donde realizaremos el Análisis Exploratorio de Datos (EDA) y parte de la limpieza y tratamiento de datos con el uso de librerías como

Pandas, Matplotlib y Seaborn que ayudarán a realizar análisis detallado y gráficos dinámicos.

- *R con RStudio*: para terminar de hacer el tratamiento y prepararlo para el modelado.

Este proceso es fundamental para asegurar la calidad y la validez de los datos ya que con esto podremos detectar variables con valores incompletos, detectar posibles sesgos y evaluar la representatividad del conjunto de datos y la variable objetivo. Con esto garantizamos que la información usada en los modelos predictivos sea confiable y adecuada.

5.4. Modelos

Para abordar nuestro problema de clasificación (default y no default) elegimos distintos modelos estadísticos para validar cual nos ofrece los mejores resultados. A continuación, se comparte el razonamiento detrás de la elección de cada uno de los modelos, asegurando que la metodología utilizada sea clara y justificable.

Modelos Lineales: se han elegido en el caso de que nuestros datos presenten fuertes correlaciones lineales. Entre ellos destacamos:

- En el caso del modelo **Ridge**, este se elige dada sus propiedades para reducir el sobreajuste (overfitting), permitiendo un mejor desempeño del modelo en datos fuera de la base de entrenamiento. Además, cuando existe fuerte correlación entre las variables independientes, Ridge distribuye de manera adecuada la relevancia de cada predictor, evitando que una sola variable domine el modelo y cause inestabilidad. A diferencia de Lasso, Ridge no elimina variables, lo cual puede ser beneficioso al conservar toda la información disponible para la predicción.
- Para **Lasso**, al igual que Ridge, este modelo maneja de manera adecuada la correlación entre variables. Sin embargo, este modelo evita el sobreajuste (overfitting) al realizar la selección de variables, eliminando las que considera irrelevantes. Dependiendo de la estructura de nuestros datos, esto puede ser

beneficioso al mejorar la interpretabilidad del modelo, aunque también podría generar una pérdida de información relevante.

- Por último, dentro de los modelos lineales consideramos adecuado utilizar el modelo de **Elastic-Net** el cuál es una combinación de Ridge y Lasso por lo que nos otorga las ventajas de ambos modelos permitiendo un balance entre la selección de variables (Lasso) y la reducción del sobreajuste sin eliminar predictores (Ridge). Sin embargo, en caso de no mostrar una mejora relevante contra dichos modelos se considerará más adecuado bajo el principio de la Navaja de Ockham² tomar el modelo más sencillo (Ridge o Lasso) para reducir la complejidad y el número de hiperparámetros a utilizar.

Modelos no Lineales: se eligieron para interpretar patrones complejos, interacciones y relaciones no lineales que presenten nuestros datos.

- **Bagging y Random Forest**, si bien Random Forest es una extensión de Bagging se decide probar con ambos modelos.
 - Bagging se considera útil si todas las variables son importantes, ya que en este caso no se excluirían aleatoriamente como en Random Forest.
 - Random Forest, por otro lado, ofrece un buen equilibrio entre sesgo y varianza, permitiendo una mejor generalización y un aprendizaje más robusto.

La combinación de estos modelos busca aprovechar los beneficios de los árboles de decisión, evitando el sobreajuste mediante técnicas de ensamblado.

- En cuanto a **KNN**, nuestro interés en probar dicho modelo es ver si clientes con características similares han realizado default. El presente modelo podrá agruparlos de manera adecuada y clasificar futuras observaciones mediante el cálculo de las distancias entre los datos.

² **Principio de la Navaja de Ockham:** este principio dice que en igualdad de condiciones, la explicación más simple suele ser la correcta.

- El modelo **SVM** se considera útil si existe una clara separación entre las personas que hacen y no hacen default. En este caso, SVM podría encontrar hiperplanos óptimos para separar ambas clases, asegurando una clasificación efectiva.
- Por último **MARS**, se elige dado que combina las características de los modelos lineales y no lineales tomando la simplicidad de los modelos lineales al igual que la flexibilidad que otorgan los no lineales para detectar relaciones más complejas entre nuestras variables creando relaciones lineales por segmentos (splines) sobre nuestro conjunto de datos.

El uso de la técnica como **SHAP** (SHApley Additive Explanations) es esencial para garantizar la interpretabilidad de los modelos no lineales. Esta técnica se considera fundamental debido a que:

- En entidades financieras, las decisiones deben ser explicables para cumplir con regulaciones y justificar la asignación de crédito.
- Mejora la transparencia y la confianza en el modelo, evitando que sea una "caja negra", es decir, que no se tenga un conocimiento del proceso de predicción.
- Permiten verificar que las decisiones del modelo no sean sesgadas, ayudando a garantizar equidad en el otorgamiento de crédito.
- Facilita la identificación de áreas de mejora en los modelos, contribuyendo a mejorar las predicciones.

5.5. Monitorización y visualización

Para el desarrollo de dashboards y visualización de resultados, elegimos Power BI debido a su balance entre usabilidad y funcionalidad avanzada. Esta herramienta permite crear dashboards interactivos y fácilmente compartibles, una ventaja clave frente a alternativas como Tableau o Python, ya que no requiere conocimientos avanzados de programación. Asimismo, Power BI ofrece integración fluida con distintas fuentes de datos y actualizaciones automatizadas, lo cual facilita la monitorización continua de los modelos.

Seguimiento de métricas clave y visualizaciones

Se diseñarán dos dashboards con enfoques diferenciados para públicos específicos: uno técnico, dirigido al equipo de Data Science, y otro estratégico, orientado al monitoreo de objetivos de negocio.

La combinación de ambos dashboards asegura que todos los actores involucrados, desde los científicos de datos hasta la alta dirección, cuenten con las herramientas necesarias para monitorizar el desempeño de los modelos, identificar áreas de mejora y alinear los resultados técnicos con los objetivos de negocio.

6. DESARROLLO Y ANÁLISIS

6.1. Análisis Exploratorio (EDA)

Antes de iniciar la fase de modelado, consideramos esencial realizar un análisis exploratorio de datos (EDA) sobre nuestro dataset. Esta etapa nos permitió comprender en profundidad la estructura y las principales características de la base de datos, identificar la presencia de valores atípicos y detectar patrones relevantes entre las variables. Además, el EDA resultó fundamental para garantizar la calidad, consistencia e integridad de los datos, sentando así una base sólida que nos asegurara un proceso de modelado fiable y adecuado a los objetivos del TFM.

6.1.1. Librerías Python

Todo el EDA se realizó Python sobre Google Colab, lo que nos permitió colaborar en un mismo cuaderno y acceder de forma directa a nuestro Google Drive grupal.

En la tabla que se muestra a continuación, mostramos las librerías utilizadas a lo largo de todo el análisis en Python:

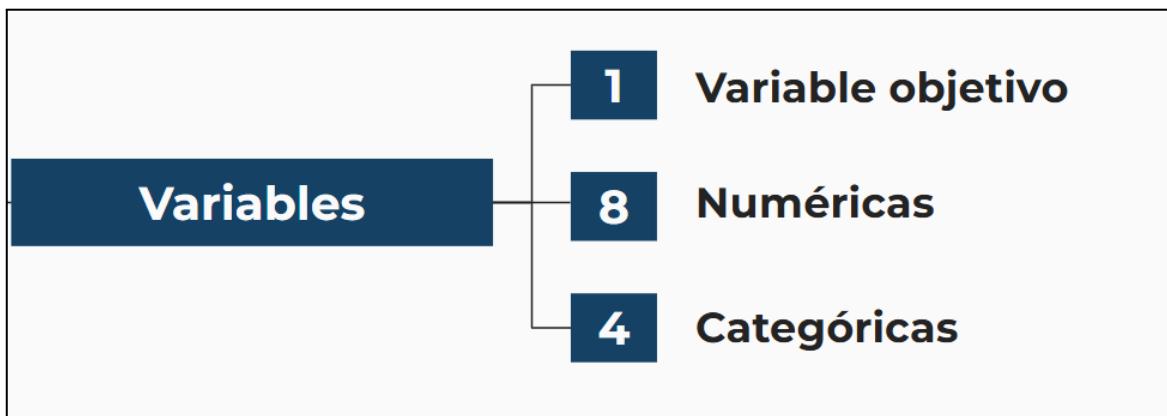
Librerías	Función principal en el EDA	Uso
google.colab.drive	Montar Google Drive como sistema de archivos	Lectura del CSV original y guardado de figuras/tablas finales
google.colab.files	Descargar archivos generados	Exportar CSV con los datos ya limpios
pandas	Manipulación y análisis tabular	Limpieza de formatos y cálculo de estadísticos
numpy	Operaciones numéricas vectorizadas	Conversión de tipos y cálculo eficiente de percentiles, medias, etc.
seaborn	Visualización estadística de alto nivel	Histogramas, violin-plots, barplots y mapas de calor con estilo uniforme
matplotlib.pyplot	Base gráfica y exportación	Ajuste de ejes/títulos y de los gráficos
IPython.display	Visualización de objetos enriquecidos	display() para renderizar DataFrames, gráficos
matplotlib.ticker	Control de los marcadores en los ejes	MaxNLocator para forzar que los marcadores sean
scipy.stats	Estadística descriptiva y contrastes de hipótesis	chi2_contingency (chi-cuadrado), ttest_ind (t-Student/Welch)
ydata_profiling	Generación automática de reportes exploratorios (EDA)	ProfileReport para crear un informe HTML con resúmenes estadísticos, histogramas, correlaciones, etc.

Tabla 6.1.1- Librerías Python

6.1.2. Variables

La variable objetivo, sobre la que se centrará el análisis predictivo, es current_loan_status, que indica si el préstamo terminó en situación de default o no default.

Figura 6.1.1- Categorías de variables



Fuente: elaboración propia

El conjunto de variables disponible es el siguiente:

- **customer_id**: identificador único de cada cliente (número).
- **customer_age**: edad del cliente solicitante del préstamo (entero).
- **customer_income**: Ingreso anual declarado del cliente (convertido a variable numérica en libras esterlinas).
- **home_ownership**: tipo de tenencia de vivienda del cliente:
 - Rent: vivienda en renta.
 - Mortage: vivienda en hipoteca.
 - Own: vivienda propia.
 - Other: otro tipo de vivienda.
- **employment_duration**: duración del empleo actual del cliente en años (variable numérica).

- **loan_intent**: motivo declarado para la solicitud del préstamo (categorías:
 - Education: préstamo dedicado a la financiación de estudios.
 - Medical: préstamo dedicado a la cobertura de gastos médicos.
 - Venture: préstamos dedicados al emprendimiento o negocios nuevos.
 - Personal: préstamos dedicados a uso personal no especificado.
 - Debtconsolidation: préstamos dedicados a pagar deudas anteriores.
 - Homeimprovement: préstamos dedicados a reformas o mejoras del hogar.
- **loan_grade**: calificación crediticia asignada al préstamo, de mejor a peor calidad (A, B, C, D, E).
- **loan_amnt**: importe total solicitado para el préstamo (convertido a variable numérica en libras esterlinas).
- **loan_int_rate**: tipo de interés aplicado al préstamo, expresado como porcentaje (%).
- **term_years**: duración del préstamo en años.
- **historical_default**: indicador de si el cliente ha tenido impagos anteriores:
 - Y: el cliente tiene impagos anteriores.
 - N: el cliente no tiene impagos anteriores.
- **cred_hist_length**: longitud del historial crediticio del cliente en años (entero).
- **current_loan_status**: variable objetivo; indica si el préstamo terminó en default o no default (binario).

6.1.3. Preprocesamiento preliminar

Antes de realizar ningún análisis, realizamos un ajuste mínimo de formatos y tipos de datos que garantiza que el análisis descriptivo se calcule sobre valores coherentes. En el CSV original varias variables numéricas llegaban como texto (símbolos de libra, comas o porcentajes), además, aprovechamos, y cambiamos la variable objetivo, **current_loan_status** e **historical_default** a booleanas (True y False).

Columna	Tipo original	Tipo corregido	Transformación realizada
0	customer_id	object	Conversión a entero
1	customer_age	int64	Sin cambios
2	customer_income	object (coma)	Eliminación de comas y conversión
3	home_ownership	object	Reetiquetado como cadena
4	employment_duration	float	Conversión a entero
5	loan_intent	object	Reetiquetado como cadena
6	loan_grade	object	Reetiquetado como cadena
7	loan_amnt	object (f, ,)	Retirada de "f" y comas, conversión
8	loan_int_rate	object (%)	Retirada de "%", conversión
9	term_years	int64	Sin cambios
10	historical_default	object ("Y/N")	Mapeo Y→True, N→False
11	cred_hist_length	int64	Sin cambios
12	Current_loan_status	object ("DEFAULT/NO DEFAULT")	Mapeo DEFAULT→True, NO DEFAULT→False

Tabla 6.1.2 - Ajustes de tipo y formato aplicados a las variables del Loan Dataset

6.1.4. Análisis univariante

El análisis univariante constituye la primera fase de la exploración estadística ya que se dedica a examinar cada variable por separado con el fin de entender su comportamiento básico. Tal como vemos en la publicación realizada por Jordi Mas Elias (Mas Elias, 2019), esta fase inicial se apoya:

- *Medidas de tendencia central* (media, mediana, moda) para ubicar el valor típico.
- *Medidas de dispersión* (rango, varianza, desviación típica, IQR) para cuantificar la variabilidad.
- *Forma de la distribución* (asimetría, curtosis) ilustrada con histogramas y boxplots.
- *Tablas de frecuencias* para variables categóricas, que muestran la proporción de cada categoría.

En nuestro caso, comenzamos el análisis univariante aplicando `loan.describe().T` sobre las variables numéricas del *Loan Dataset*:

	count	mean	std	min	25%	50%	75%	max
customer_id	32583.00	16289.50	9405.92	1.00	8144.50	16288.00	24433.50	32581.00
customer_age	32586.00	27.73	6.36	3.00	23.00	26.00	30.00	144.00
customer_income	32586.00	66076.37	61980.29	4000.00	38500.00	55000.00	79200.00	6000000.00
employment_duration	31691.00	4.79	4.14	0.00	2.00	4.00	7.00	123.00
loan_amnt	32585.00	9756.25	21771.85	500.00	5000.00	8000.00	12200.00	3500000.00
loan_int_rate	29470.00	11.01	3.24	5.42	7.90	10.99	13.47	23.22
term_years	32586.00	4.76	2.47	1.00	3.00	4.00	7.00	10.00
cred_hist_length	32586.00	5.80	4.06	2.00	3.00	4.00	8.00	30.00

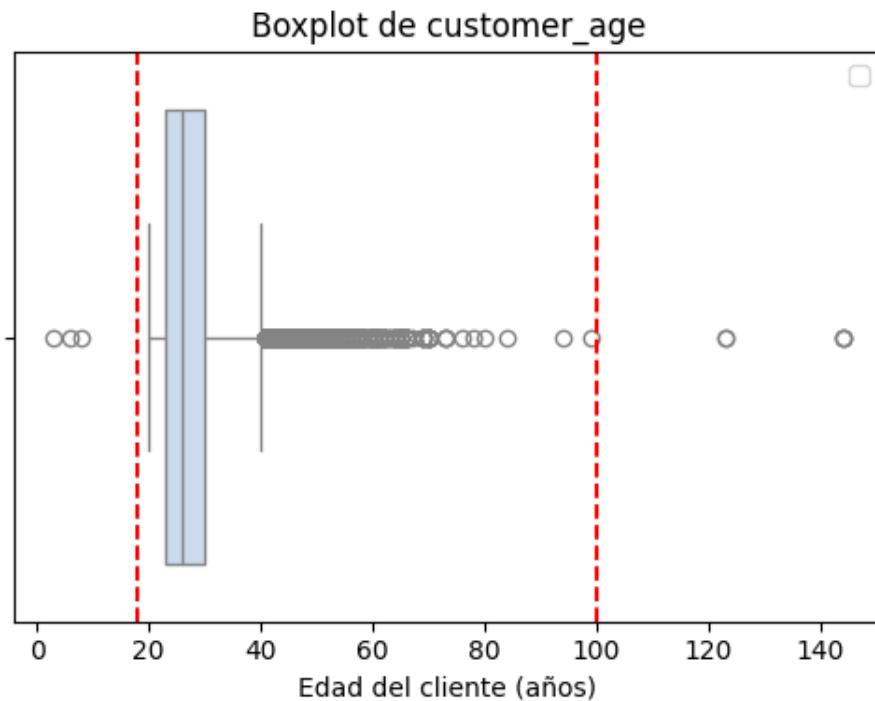
Tabla 6.1.3 - Descripción de la variables

En la [Tabla 6.1.3](#) se aprecian varios puntos llamativos sobre las variables numéricas del dataset:

Edad del solicitante

Como podemos observar, la edad media es de 27,7 años y la mediana de 26 años, con la mayoría concentrada entre 23 y 30 años. Sin embargo, se detectan valores extremos (mínimo de 3 y máximo de 144 años), claramente erróneos. El boxplot ([Gráfico 6.1.1](#)) confirma visualmente estos casos anómalos, destacando solicitudes por debajo de los 18 años y por encima de los 100 años.

Gráfico 6.1.1. - Boxplot customer_age



Ingresos anuales

La tabla descriptiva muestra un ingreso medio de aproximadamente 66.076 £. Sin embargo, la alta desviación estándar (cerca de 61980£) junto con el valor máximo registrado (de 6.000.000£) indican una distribución altamente asimétrica, con presencia de valores extremadamente elevados que no reflejan el comportamiento habitual de la mayoría de clientes. Además, el rango intercuartílico confirma que el 50% de los clientes declaran ingresos anuales entre 38.500£ y 79.200£.

Para entender mejor esta dispersión se construyeron diferentes visualizaciones. En primer lugar, vemos el [Gráfico 6.1.2](#), donde se observa que la gran mayoría de clientes concentra sus ingresos en los tramos más bajos, haciendo que los ingresos más altos apenas se aprecien visualmente. Para superar esta limitación y ver la distribución de una mejor manera, recurrimos a una escala logarítmica ([Gráfico 6.1.3](#)) que permite identificar con mayor claridad el carácter sesgado de la distribución observando un grupo minoritario con ingresos muy por encima del promedio.

Gráfico 6.1.2 - Distribución de los Ingresos de los Clientes

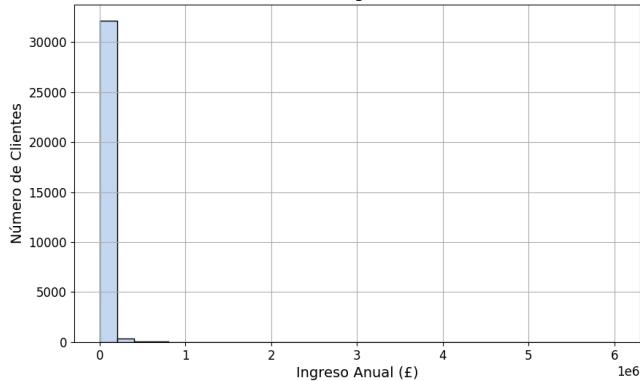
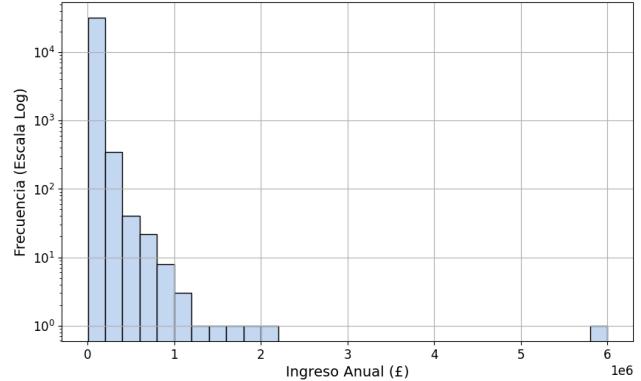
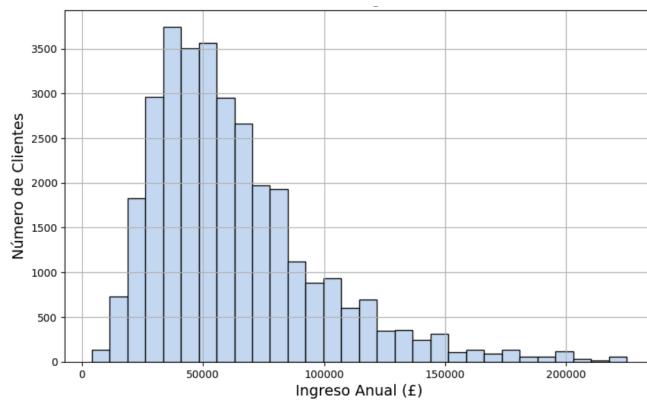


Gráfico 6.1.3 - Distribución logarítmica de los Ingresos de los Clientes



Finalmente, el [Gráfico 6.1.4](#) representa la distribución excluyendo los valores superiores al percentil 99 (P99), lo que nos permite visualizar con más detalle el comportamiento del grueso de la muestra. Aquí. confirmamos que los ingresos tienden a agruparse entre los 30.000£ y 70.000£, con una disminución progresiva en el número de clientes a medida que aumentan los ingresos.

Gráfico 6.1.4 - Distribución de los Ingresos de los Clientes (P99)



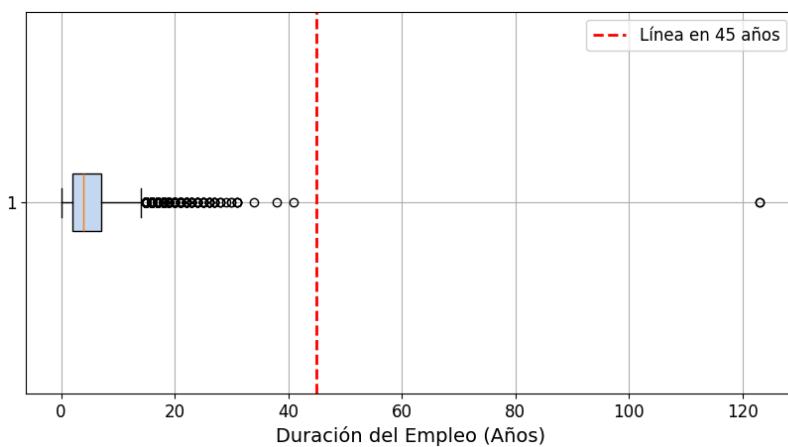
Antigüedad laboral

La mayoría de los clientes lleva entre 2 y 7 años en su puesto actual, con una antigüedad laboral promedio de 4,79 años. Este dato es coherente con la juventud general observada en

la base de datos, donde predominan solicitantes de entre 23 y 30 años de edad. No obstante, se registra un valor máximo de 123 años de antigüedad laboral, lo que sugiere errores de registro.

El boxplot ([Gráfico 6.1.5](#)) confirma esta situación: la mayoría de las observaciones se concentra en los valores esperados, pero se detectan numerosos valores atípicos, entre ellos el extremo de 123 años. Para facilitar su identificación, se ha trazado una línea discontinua en los 45 años, considerado un límite razonable dentro de una trayectoria laboral típica.

Gráfico 6.1.5 - Boxplot de la duración del empleo



Importe del préstamo

La tabla descriptiva muestra que el importe medio de los préstamos es de 9.756 £, acompañado de una desviación estándar elevada, superior a 21.700 £. Este nivel de dispersión queda evidenciado por el valor máximo registrado, que alcanza los 3.500.000 £, a pesar de que la mayoría de los préstamos se sitúan en un rango mucho más acotado, entre 5.000 £ y 12.000 £. Esta diferencia pone de manifiesto la presencia de valores extremos que alteran significativamente la distribución general de la variable.

Para analizar esta asimetría, se han utilizado diversos gráficos tal y como hicimos con la variable de los ingresos anuales. En el [Gráfico 6.1.6](#), se aprecia la distribución original de la variable y, observamos, como la mayor parte de los préstamos está en la parte de izquierda del gráfico. Sin embargo, este gráfico debido a su distribución asimétrica, no nos permite identificar como se comporta la muestra, por lo que optamos a hacer el [Gráfico](#)

[6.1.7](#) donde mostramos la distribución logarítmica. Este nos permite confirmar la asimetría de la muestra pero sigue sin permitirnos ver el comportamiento general.

Gráfico 6.1.6 - Distribución del Importe del Préstamo

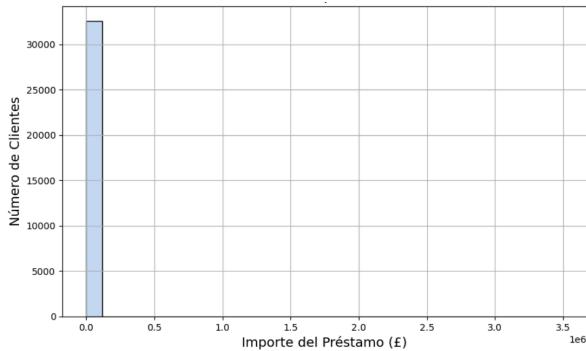
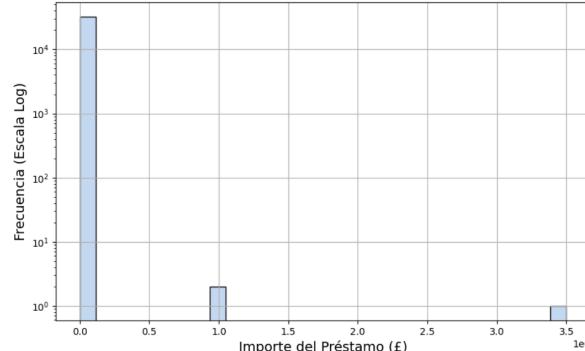
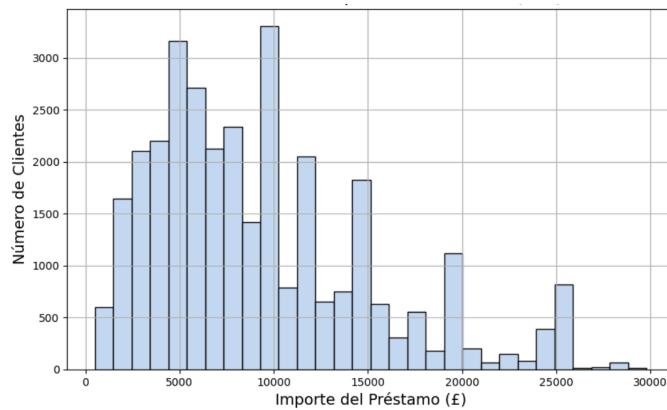


Gráfico 6.1.7 - Distribución logarítmica del Importe del Préstamo



El [Gráfico 6.1.8](#) nos permite ver la distribución del Importe del Préstamo una vez eliminados los valores del percentil 99. Esto nos permite observar con más detalle el comportamiento del conjunto principal de clientes, donde los préstamos más frecuentes oscilan entre las 4.000£ y las 10.000£.

Gráfico 6.1.8 - Distribución del Importe del Préstamo (P99)



Tipo de Interés

La variable tipo de interés presenta una media de 11 % aproximadamente, con la mayoría de los valores comprendidos entre el 7,9 % (percentil 25) y el 13,5 % (percentil 75). A pesar de que el rango total abarca desde 5,4 % hasta 23,2 %, la dispersión es moderada y concentrada en torno al centro de la distribución, como refleja el rango intercuartílico.

En el histograma ([Gráfico 6.1.9](#)), se observa que la distribución no es simétrica, sino que presenta un leve sesgo hacia la derecha. Los tipos de interés más comunes se sitúan entre el 9 % y el 13 %, siendo muy pocos los préstamos concedidos por encima del 17 %. Esta asimetría también queda reflejada en el boxplot ([Gráfico 6.1.10](#)), donde se identifican algunos valores atípicos por encima del 20 %, aunque no en una proporción alarmante.

Gráfico 6.1.9 - Distribución del Tipo de Interés

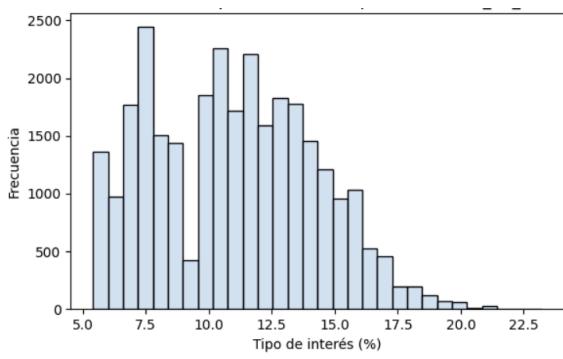
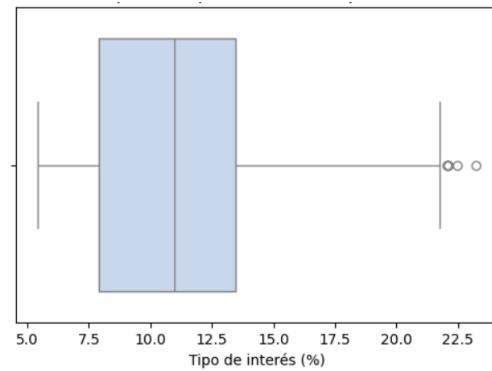


Gráfico 6.1.10 - Boxplot del Tipo de Interés

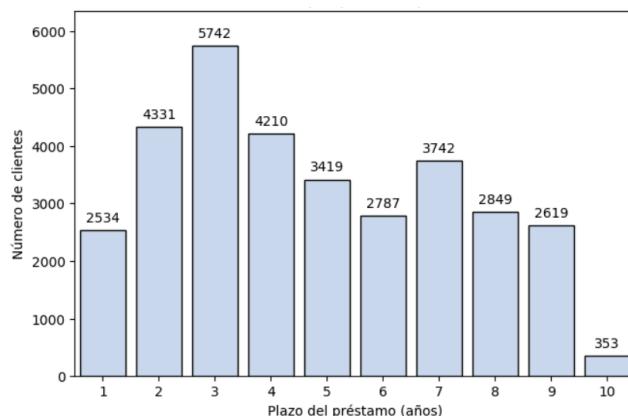


Plazo del Préstamo

El análisis del plazo de los préstamos revela una distribución claramente discreta, con valores comprendidos entre 1 y 10 años. Tal como muestra el [Gráfico 6.1.11](#), la duración más frecuente es de 3 años, con un total de 5.742 préstamos, seguida por los plazos de 2 y 4 años. A partir de ahí, la frecuencia disminuye progresivamente conforme aumenta la duración del préstamo, siendo el plazo de 10 años el menos habitual, con solo 353 casos registrados.

Este patrón indica que las entidades financieras tienden a conceder préstamos con plazos relativamente cortos, lo cual podría estar relacionado con políticas internas de riesgo o perfiles crediticios de los solicitantes a los cuales no tenemos acceso. Además, se observa que a medida que el plazo se alarga, la demanda de préstamos se reduce notablemente, lo que sugiere una preferencia por parte de las entidades o de los clientes hacia préstamos de menor duración.

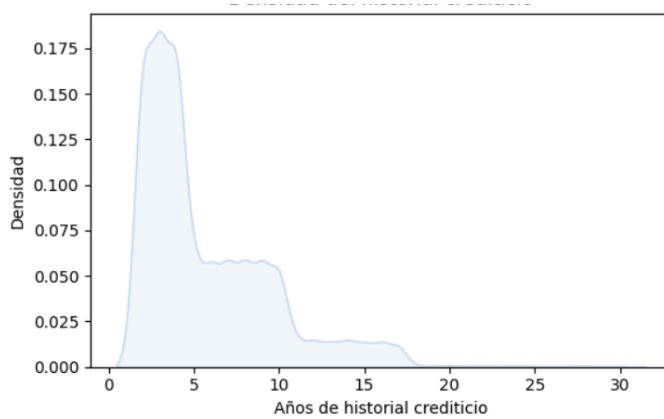
Gráfico 6.1.11 - Frecuencia por Plazo de Préstamo



Antigüedad del Historial Crediticio

La variable cred_hist_length refleja la antigüedad del historial crediticio de los clientes en años. El gráfico de densidad ([Gráfico 6.1.12](#)) muestra una distribución asimétrica hacia la derecha, con una fuerte concentración de clientes con historiales recientes. La mayoría de observaciones se sitúan entre los 2 y 8 años, siendo especialmente frecuentes los perfiles con entre 3 y 5 años de antigüedad crediticia. A medida que el historial se alarga, la densidad decrece notablemente, lo que indica que son pocos los clientes con historiales superiores a 15 años.

Gráfico 6.1.12 - Densidad del Historial crediticio



Una vez completado el análisis univariante de las variables cuantitativas, procedemos a examinar brevemente las variables categóricas presentes en el conjunto de datos. A través

de este, se busca identificar patrones relevantes en el perfil de los solicitantes y hacer hipótesis relacionadas con las variables numéricas.

En relación con el tipo de vivienda ([Gráfico 6.1.13](#)), la mayoría de los clientes viven en régimen de alquiler (50,5%) o tienen una hipoteca (41,3%), mientras que solo un 7,9% posee una vivienda propia. Esta distribución es coherente con los resultados obtenidos previamente sobre la edad y los ingresos de los clientes: la mayoría son personas jóvenes (mediana de 26 años) con historiales crediticios aún breves y salarios que tienden a concentrarse en los tramos bajos o medios, lo que limita su capacidad para acceder a la propiedad (*Véase Tabla 6.1.3*).

En cuanto al motivo del préstamo ([Gráfico 6.1.14](#)), las razones más frecuentes son la educación con un 19.8% de las observaciones, seguido de cerca por médicos (18,6%) y emprendimiento (17,5%). Estas finalidades también refuerzan la idea de una población joven y en desarrollo profesional, que solicita información para cubrir necesidades básicas o invertir en su futuro.

En relación con la calificación crediticia ([Gráfico 6.1.15](#)), destaca el predominio del grado A con un 48.1% de los casos, seguido de los grados B y C. La escasa presencia de observaciones D y E puede indicar una mayor prudencia por parte de la entidad financiera a la hora de conceder créditos a clientes con perfiles de mayor riesgo. Este dato, se podría relacionar con el análisis previo de los Tipos de Interés, donde se observaron valores moderados con pocos casos extremos.

Gráfico 6.1.13- Frecuencia del Tipo de Vivienda

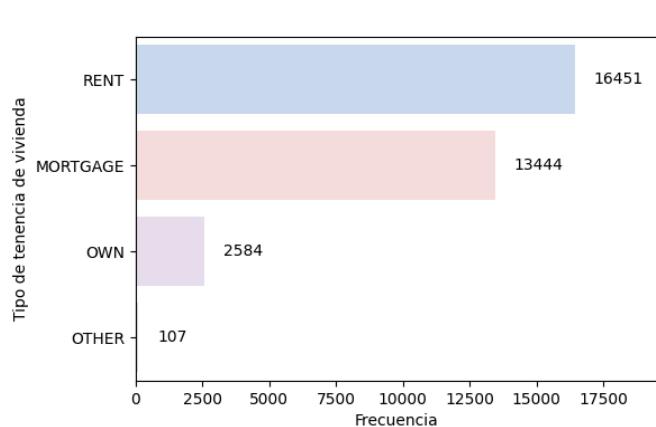


Gráfico 6.1.14 - Frecuencia del Motivo del Préstamo

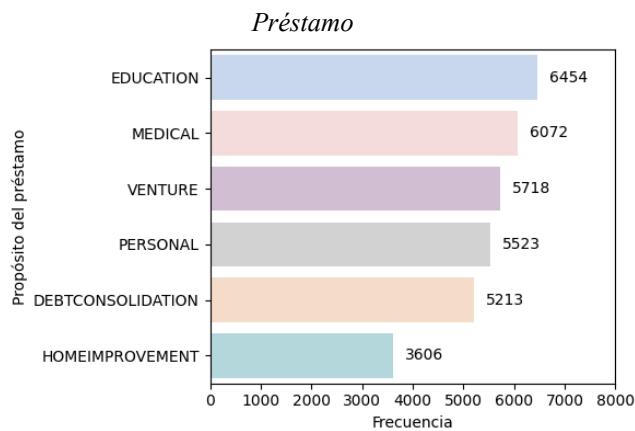
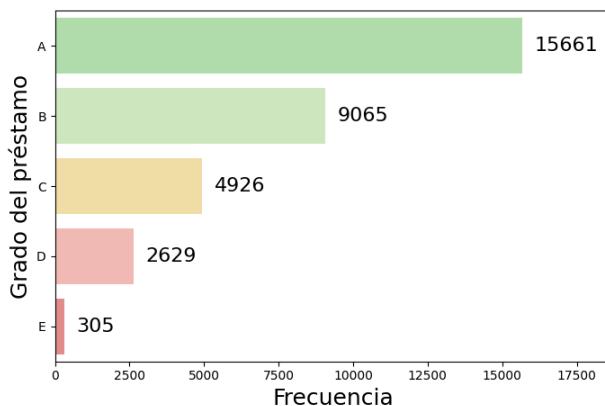
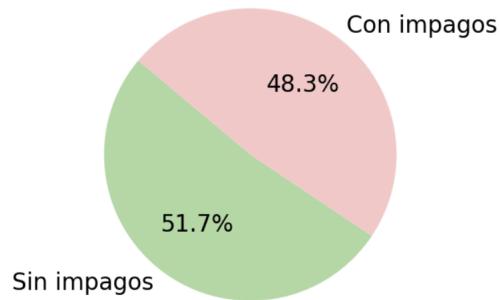


Gráfico 6.1.15 - Frecuencia de Calificación Crediticia



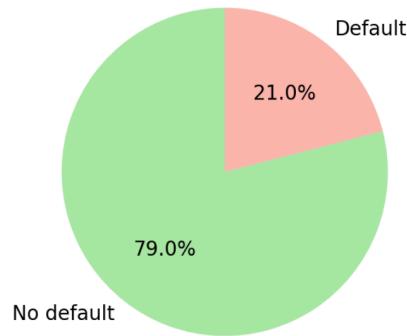
En cuanto al historial de impagos ([Gráfico 6.1.16](#)), se aprecia un equilibrio entre los que no tienen antecedentes de morosidad con un 51,7% y los que sí que lo tienen con un 48,3%.

Gráfico 6.1.16 - Proporción de Clientes con y sin Historial Crediticio



Por último, la variable `current_loan_status`, que actúa como variable objetivo del modelo. refleja si el préstamo terminó en situación de default o fue pagado correctamente (no default). Tal y como muestra el [Gráfico 6.1.17](#), el 21% de los préstamos se encuentran en situación de impago, mientras que el 79% restante han sido cumplidos en tiempo y forma.

Gráfico 6.1.17 - Distribución del Estado del Préstamos (variable objetivo)



Con este análisis de la variable objetivo `current_loan_status` se concluye la fase de exploración univariante del conjunto de datos donde se han caracterizado las principales variables individualmente, identificando valores extremos en edad, ingresos y antigüedad laboral, distribuciones asimétricas en variables numéricas clave, así como la predominancia de determinadas categorías en variables como `home_ownership`, `loan_intent` o `loan_grade`. Estos patrones iniciales ofrecen una primera imagen del perfil de los solicitantes y condicionarán el comportamiento del modelo en fases posteriores.

6.1.5. Análisis Bivariante

Tras examinar cada variable de forma individual en la fase univariante, el análisis bivariante tiene como objetivo explorar las relaciones existentes entre pares de variables, en especial entre las variables explicativas y la variable objetivo `current_loan_status`. Esta etapa permite detectar patrones diferenciales entre clientes que han incurrido en impago y aquellos que han cumplido con sus obligaciones, lo cual resulta fundamental para orientar la selección de variables en la fase de modelado.

Antes de iniciar el análisis bivariante, se ha procedido a filtrar el conjunto de datos con el objetivo de eliminar observaciones claramente erróneas en dos variables clave: la edad del participante y la antigüedad laboral.

En primer lugar, se filtraron aquellas filas en las que la edad del cliente (siempre trabajando con una copia) no se encontraba dentro del rango razonable de 18 a 100 años. Posteriormente, se eliminaron los registros cuya antigüedad laboral superaba los 45 años,

ya que dichos valores no resultaban verosímiles, especialmente cuando se observaban combinaciones poco coherentes con la edad del cliente.

```
# Eliminamos el número de filas fuera del intervalo 18-100
loan_2 = loan_2[(loan_2['customer_age'] >= 18) &
(loan_2['customer_age'] <= 100)]
```

```
# Eliminar filas con antigüedad laboral superior a 45 sin borrar los NA
loan_2 = loan_2[~((loan_2['employment_duration'] > 45) &
(loan_2['employment_duration'].notna()))]
```

De esta manera se filtraron un total de 10 registros los cuales se muestran en la [Tabla 6.1.4](#):

	ID Cliente	Edad	Antigüedad Laboral	Razón de eliminación
0	1	22	123	Antigüedad laboral inválida (123 años)
81	82	144	4	Edad inválida (144)
183	184	144	4	Edad inválida (144)
210	211	21	123	Antigüedad laboral inválida (123 años)
577	576	123	2	Edad inválida (123)
749	748	123	7	Edad inválida (123)
29396	29393	6	4	Edad inválida (6)
29397	29394	8	4	Edad inválida (8)
29398	29395	3	10	Edad inválida (3)
32302	32298	144	12	Edad inválida (144)

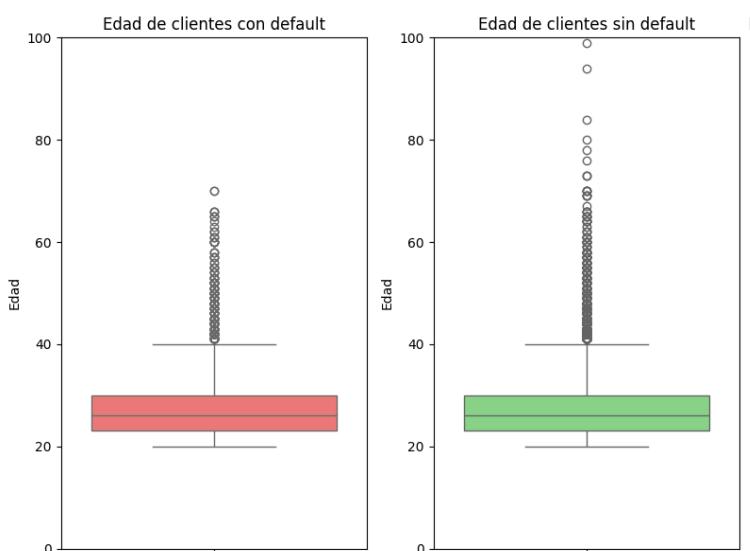
Tabla 6.1.4 - Valores erróneos eliminados

Una vez eliminados estos valores ya podemos proceder a realizar los análisis de las variables cuantitativas:

Edad y Estado del Préstamo

El análisis comparativo de la edad de los solicitantes según su estado de préstamo (*Gráfico 6.1.18*) revela diferencias sutiles pero estadísticamente significativas. En términos generales, los clientes que no incurrieron en impago presentan una edad media de 27,81 años, ligeramente superior a la de los clientes en situación de default, cuya media es de 27,38 años ([Tabla 6.1.5](#)).

Gráfico 6.1.18 - Edad de los clientes según el estado del préstamos



Current_loan_status/Edad	Media	Desviación típica
No default	27.81	6.24
Default	27.38	6.09

Tabla 6.1.5 - Media y desviación típica de la edad por default y no default

Para contrastar si esta diferencia es significativa, se aplicó un test t de Student para muestras independientes. El resultado fue un estadístico t de -5.16 con un p-valor de 2.49e-07 ([Tabla 6.1.5](#)), lo que indica una diferencia significativa al nivel del 1 %. Es decir, aunque la diferencia entre medias es pequeña, no parece atribuible al azar.

	Valor
T-Student ³	-5.162
P-valor	2.487106e-07

Tabla 6.1.6 - T-Student de edad por default y no default

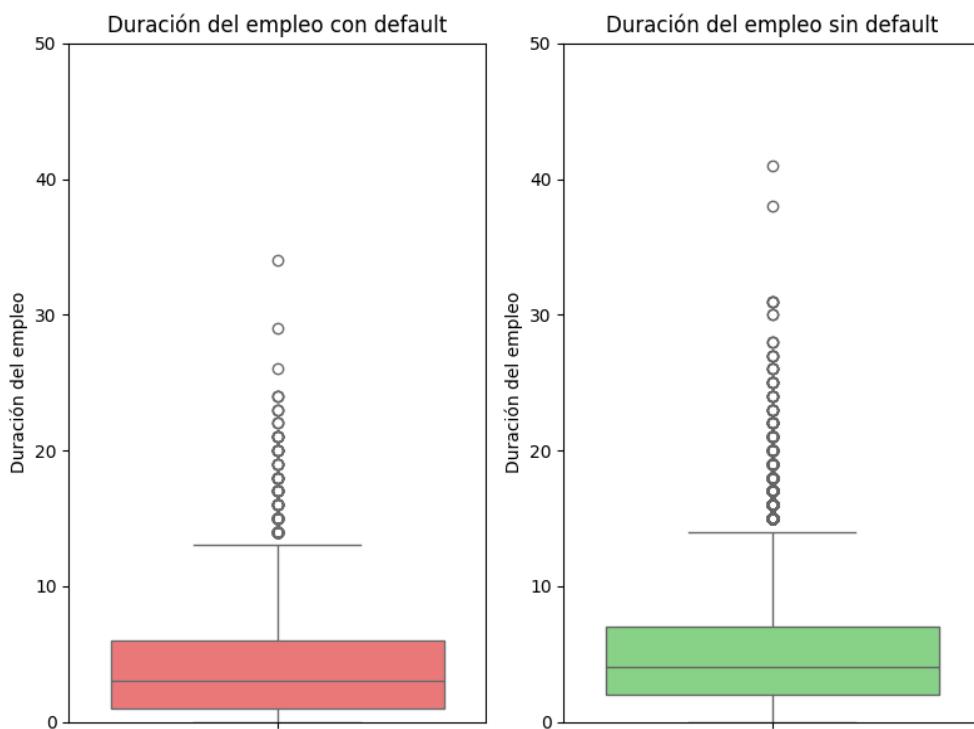
Estos resultados sugieren que la edad puede tener un leve efecto sobre la probabilidad de incumplimiento, siendo los clientes más jóvenes ligeramente más propensos a presentar impagos. Esta relación, aunque débil en términos absolutos, podría tener implicaciones relevantes en combinación con otras variables de riesgo

Duración del empleo y Current_loan_status

En el Gráfico 6.1.19 se representa la duración del empleo de los clientes en función de si han incurrido o no en default. A pesar de que ambas distribuciones presentan una forma similar y se solapan notablemente, se observan ciertas diferencias en las medidas de tendencia central.

³ La prueba t de Student es un contraste de hipótesis que permite determinar si la media de una muestra (o la diferencia entre las medias de dos muestras independientes) difiere significativamente de un valor o de otra media poblacional cuando la varianza poblacional es desconocida y el tamaño muestral es pequeño. Su estadístico sigue la distribución t, cuya forma depende de los grados de libertad de la muestra, y se emplea extensamente para evaluar diferencias de medias bajo supuestos de normalidad.

Gráfico 6.1.19 - Duración del Empleo según el estado del préstamos



Los clientes sin impagos presentan una duración media del empleo de 4,97 años, mientras que aquellos en default registran una media ligeramente inferior, de 4,08 años. La desviación típica en ambos grupos es también similar, lo que indica una variabilidad comparable ([Tabla 6.1.7](#)).

Current_loan_status/Duración del empleo	Media	Desviación típica
No default	4.97	4.07
Default	4.08	3.8

Tabla 6.1.7 - Media y desviación típica de la duración del empleo según default y no default

Aunque la diferencia entre medias parece modesta, el test t de Student devuelve un valor del estadístico de -16,60 ([Tabla 6.1.8](#)), lo que, unido al p-valor muy cerca de 0, indica que la diferencia es estadísticamente significativa. Por tanto, se puede afirmar con un alto grado de confianza que existe una relación entre la duración del empleo y la probabilidad de impago, si bien el efecto no es tan marcado como en otras variables analizadas.

	Valor
T-Student	-16.598
P-valor	4.028466e-61

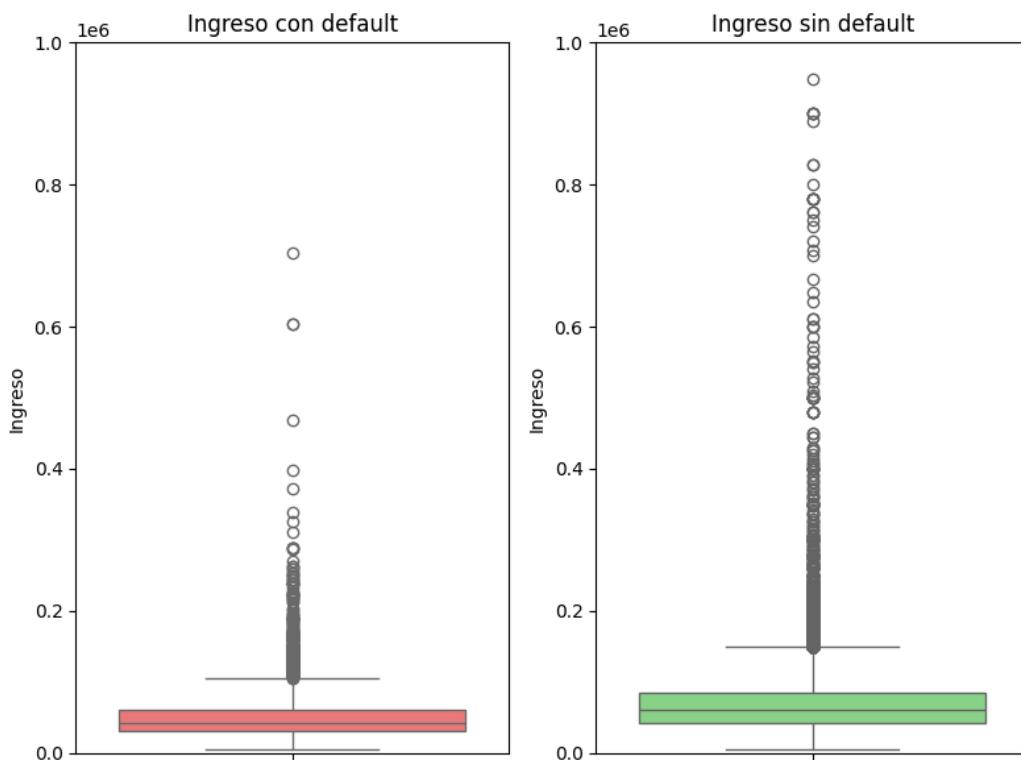
Tabla 6.1.8 - T-Student de la duración del empleo según default y no default

Ingresos y Estado del Préstamo

El [Gráfico 6.1.20](#) muestra la distribución de los ingresos anuales de los clientes según su estado de préstamo (default o no default). A pesar de que ambas distribuciones presentan valores atípicos elevados, se aprecia que los clientes con impagos tienden a concentrarse en tramos de ingreso más bajos. En cambio, los clientes sin impago muestran ingresos más elevados, con mayor dispersión.

Esta diferencia se ve reflejada en la [Tabla 6.1.9](#), donde los clientes sin impagos presentan un ingreso medio de aproximadamente 70.326£, mientras que en el grupo con impago, la media desciende a 49.166£. Las desviaciones típicas, aunque elevadas en ambos casos, también reflejan una mayor variabilidad entre quienes no presentan default.

Gráfico 6.1.20 - Ingresos según el estado del préstamos



Current_loan_status/Ingresos	Media	Desviación típica
No default	70325.89	55498.59
Default	49165.71	34659.89

Tabla 6.1.9 - Media y desviación típica de los ingresos según default y no default

Desde una perspectiva estadística, el valor del estadístico de -20.3 y un p-valor nulo ([Tabla 6.1.10](#)) confirman que la diferencia entre ambos grupos es altamente significativa. Por tanto, los ingresos constituyen una variable claramente asociada al cumplimiento de pago, lo cual sugiere su relevancia en fases posteriores de modelización y predicción de riesgo crediticio.

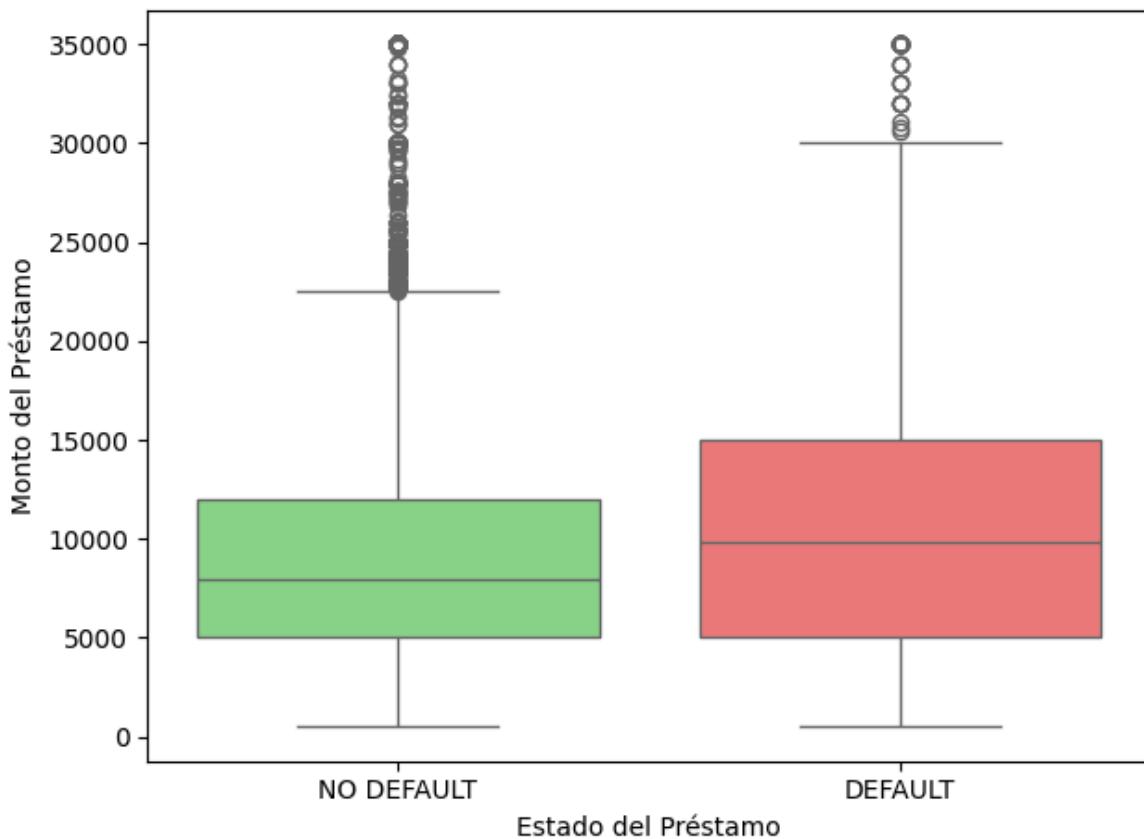
	Valor
T-Student	-203.197
P-valor	0.0

Tabla 6.1.10 - T-Student del Ingreso según default y no default

Monto del Préstamo y Estado del Préstamo

El [Gráfico 6.1.21](#) ilustra cómo varía el monto del préstamo en función del estado del préstamo. Se aprecia que los clientes en default tienden a concentrarse en montos ligeramente más elevados, con una distribución más dispersa y valores más extremos en comparación con quienes no presentan impagos.

Gráfico 6.1.21 - Monto del Préstamo según el estado del préstamos (P99)



Según los datos presentados en la [Tabla 6.1.11](#), el importe medio del préstamo asciende a 11.011€ en los casos de default, frente a 9.421€ en el grupo de no default. Aunque estas cifras podrían parecer próximas, la diferencia resulta estadísticamente significativa. El valor del estadístico T-Student (-66.684) y un p-valor de 0.0 ([Tabla 6.1.12](#)) respaldan esta conclusión, indicando que existe una relación robusta entre el monto solicitado y la probabilidad de impago.

Current_loan_status/Monto del Préstamo	Media	Desviación típica
No default	9421.03	22597.70
Default	11011.09	18301.94

Tabla 6.1.11 - Media y desviación típica del Monto del Préstamo según default y no default

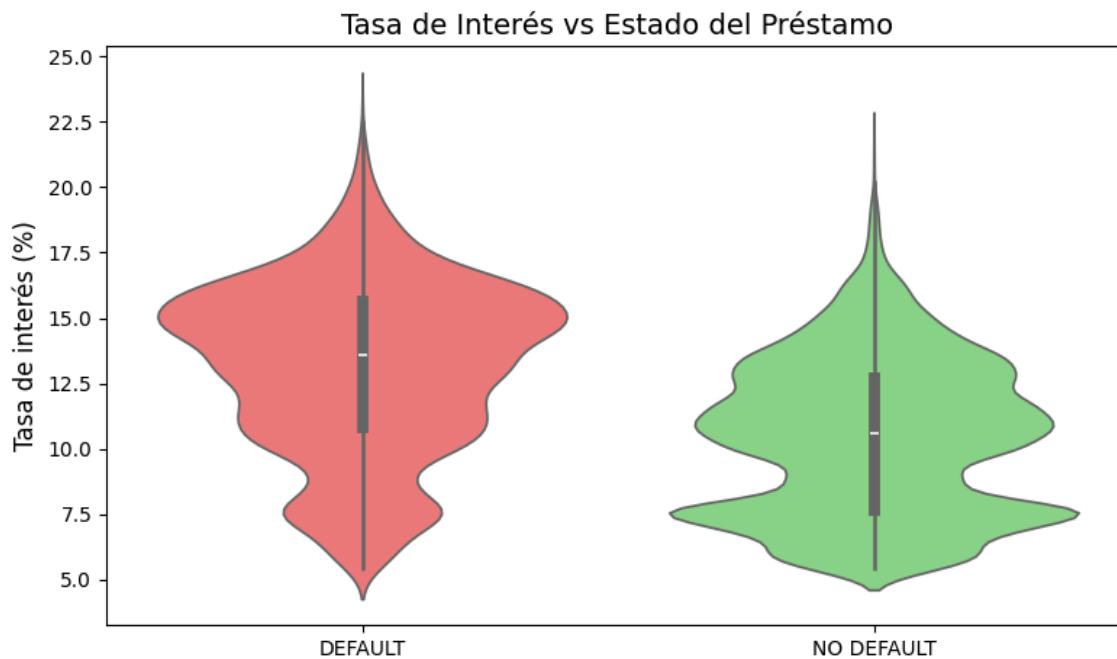
	Valor
T-Student	-66.684
P-valor	0.0

Tabla 6.1.12 - T-Student de la Monto del Préstamo según default y no default

Tasa de Interés y Estado del Préstamo

En el gráfico se observa que los clientes con préstamos en default tienden a tener tasas de interés significativamente más elevadas que aquellos que no han incumplido sus obligaciones. En concreto, la tasa media de interés aplicada a los clientes en default es del 13.13 %, mientras que para los clientes sin default se sitúa en el 10.44 %.

Gráfico 6.1.22 - Monto del Préstamo según el estado del préstamos (P99)



Esta diferencia no solo es visible a nivel gráfico, sino que además ha sido confirmada estadísticamente mediante una prueba T de Student, obteniendo un valor de 222.17 y un p-valor nulo ([Tabla 6.1.14](#)). Esto indica que la diferencia entre ambos grupos es estadísticamente significativa, por lo que podemos afirmar con un alto grado de confianza que el comportamiento de la tasa de interés varía en función del estado del préstamo.

Current_loan_status/Tasa de Interés	Media	Desviación típica
No default	10.44	2.98
Default	13.13	3.30

Tabla 6.1.13- Media y desviación típica de la Tasa de Interés o según default y no default

	Valor
T-Student	222.172
P-valor	0.0

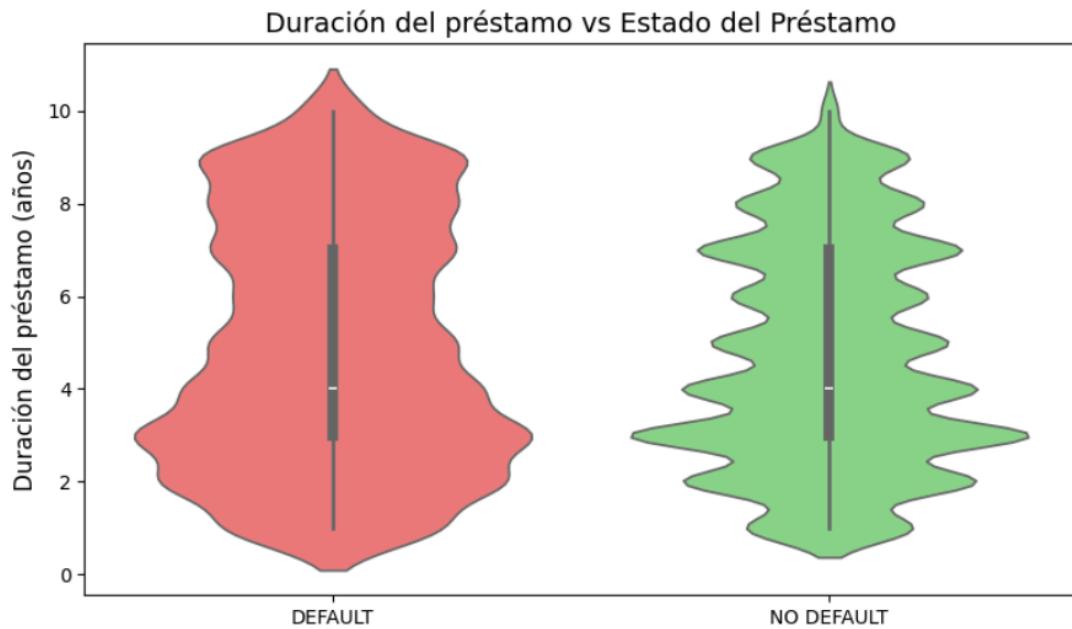
Tabla 6.1.14 - T-Student de la Tasa de Interés según default y no default

Este resultado es coherente con la lógica del sistema financiero: los perfiles con mayor probabilidad de impago suelen recibir condiciones más estrictas, lo que se traduce en tasas de interés más elevadas.

Duración del Préstamo y Estado del Préstamo

En la variable duración del préstamo, también se aprecian diferencias relevantes entre los clientes en default y los que no lo están. Tal como muestra el [Gráfico 6.1.23](#), los clientes con default presentan, en promedio, una duración de préstamo ligeramente mayor (4.94 años) en comparación con los clientes sin default (4.72 años).

Gráfico 6.1.23 - Duración del Préstamo según el estado del préstamos (P99)



Aunque esta diferencia puede parecer sutil a nivel descriptivo, el contraste estadístico realizado mediante una prueba T de Student arroja un valor de 5.995 con un p-valor de 0.0 ([Tabla 6.1.16](#)), lo que confirma que la diferencia entre ambos grupos es estadísticamente significativa.

Current_loan_status/Duración del Préstamo	Media	Desviación típica
No default	4.72	2.42
Default	4.94	2.67

Tabla 6.1.15 - Media y desviación típica de la Tasa de Interés o según default y no default

	Valor
T-Student	5.995
P-valor	2.11e-09

Tabla 6.1.16 - T-Student de la Tasa de Interés según default y no default

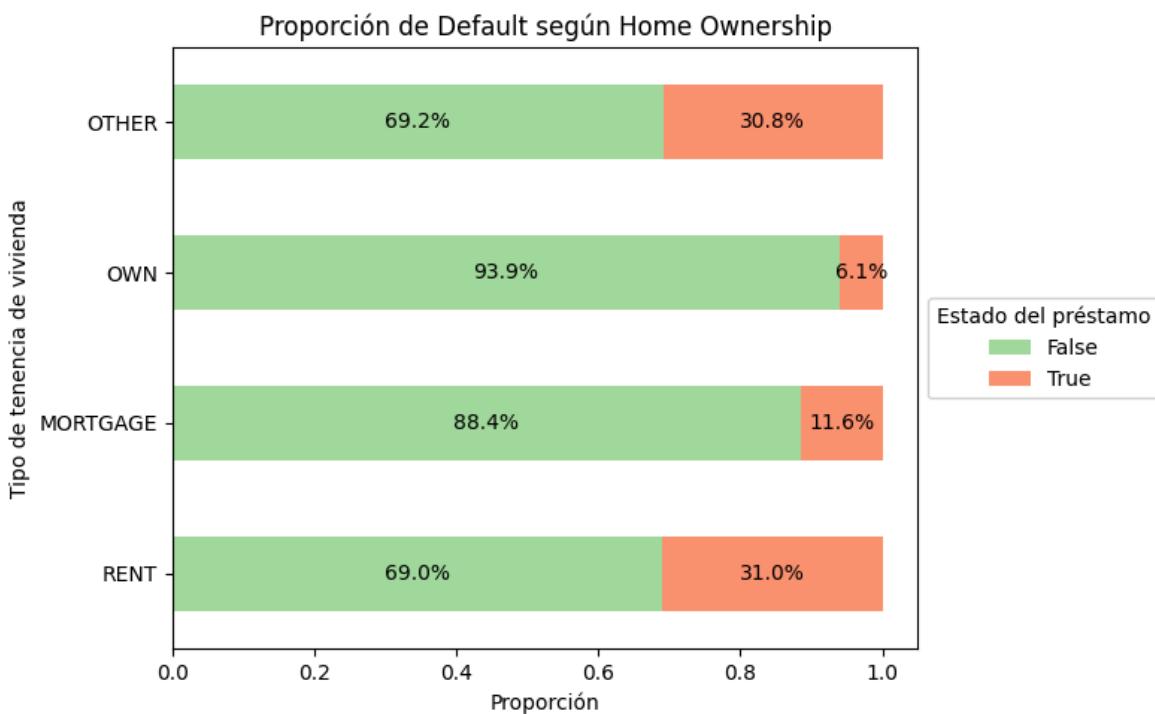
Este resultado sugiere que una mayor duración del préstamo podría estar vinculada a un mayor riesgo de impago, ya sea por el aumento del compromiso financiero a largo plazo o por la acumulación de incertidumbre en el tiempo.

Tras analizar las variables numéricas, se han identificado diferencias significativas entre los clientes en default y no default, lo que sugiere su posible relevancia en fases posteriores de modelado. A continuación, se estudiarán las variables categóricas para evaluar su relación con el estado del préstamo:

Tipo de Vivienda y Estado del Préstamo

El análisis de la variable categórica home_ownership en relación con el estado del préstamo (Current_loan_status) revela una distribución desigual de impagos entre los distintos tipos de tenencia de vivienda. Tal como muestra el [Gráfico 6.1.24](#), los clientes que alquilan (RENT) presentan una tasa de default considerablemente elevada (31%), mientras que aquellos que poseen una hipoteca (MORTGAGE) descienden a un 11,6%, y los propietarios plenos (OWN) exhiben la menor proporción de impago con apenas un 6,1%. Esto sugiere que el tipo de vivienda no es una condición neutral, sino que refleja situaciones financieras diferenciadas que podrían influir en la capacidad de pago.

Gráfico 6.1.24 - Duración del Préstamo según el estado del préstamos (P99)



	No Default	Default
MORTGAGE	11882	1556
OTHER	74	33
OWN	2426	157
RENT	11352	5092

Tabla 6.1.17 - Frecuencia del Tipo de Vivienda según default y no default

Para evaluar si esta asociación es estadísticamente significativa, se realizó una prueba de independencia de Chi-cuadrado. Esta prueba compara las frecuencias observadas con las esperadas bajo la hipótesis nula de que no hay relación entre las variables. En este caso, se obtuvo un estadístico de Chi-cuadrado 2058.5 con 3 grados de libertad.

El p-valor resultante fue cero, lo cual indica que la probabilidad de que estas diferencias se hayan producido por azar es prácticamente nula. Por tanto, se concluye con un nivel de confianza muy alto que existe una relación estadísticamente significativa entre el tipo de vivienda del cliente y la probabilidad de incurrir en impago.

	Valor
Chi-cuadrado	2058.5
Grados de Libertad	3
P-valor	0.0

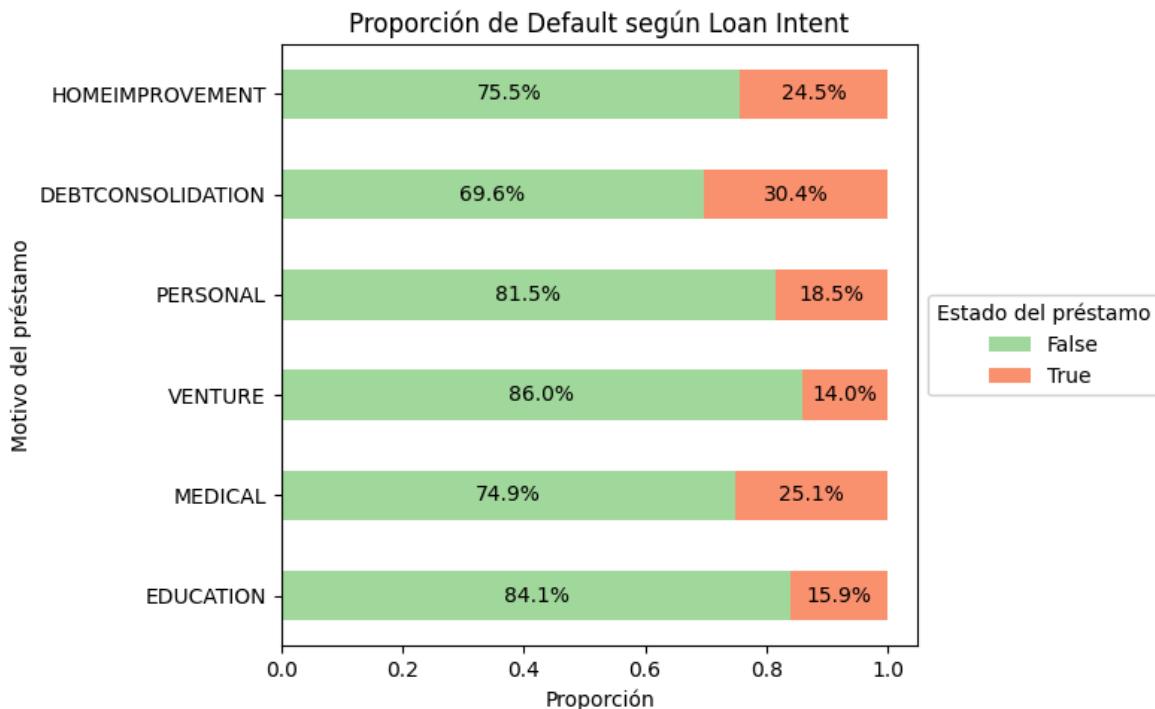
Tabla 6.1.18 - Chi-Cuadrado del Tipo de Vivienda según default y no default

Intención del Préstamo y Estado del Préstamo

En lo que respecta a la variable loan_intent, se aprecian diferencias relevantes en la proporción de impago entre las distintas finalidades del préstamo. Los préstamos destinados a consolidación de deudas (Debt Consolidation) y mejora del hogar (Home Improvement) presentan los porcentajes de impago más elevados, con un 30,4% y un 24,5%

respectivamente. Por el contrario, las finalidades educativas y de inversión reflejan menores tasas de impago, en torno al 14% y 16% ([Gráfico 6.1.25](#)).

Gráfico 6.1.25 - Motivo del Préstamo según el estado del préstamo (P99)



	No Default	Default
DEBTCONSOLIDATION	3629	1583
EDUCATION	5424	1026
HOMEIMPROVEMENT	2724	882
MEDICAL	4548	1523
PERSONAL	4497	1022
VENTURE	4912	802

Tabla 6.1.19 - Frecuencia del Motivo del Préstamo según default y no default

La prueba de chi-cuadrado ($\chi^2 = 652.95$; $gl = 5$; $p < 0,001$) confirma que la relación entre *loan_intent* ([Tabla 6.1.20](#)) y el estado del préstamo es estadísticamente significativa, lo que

indica que el motivo declarado del préstamo puede estar asociado con una mayor o menor probabilidad de impago.

	Valor
Chi-cuadrado	652.95
Grados de Libertad	5
P-valor	0.0

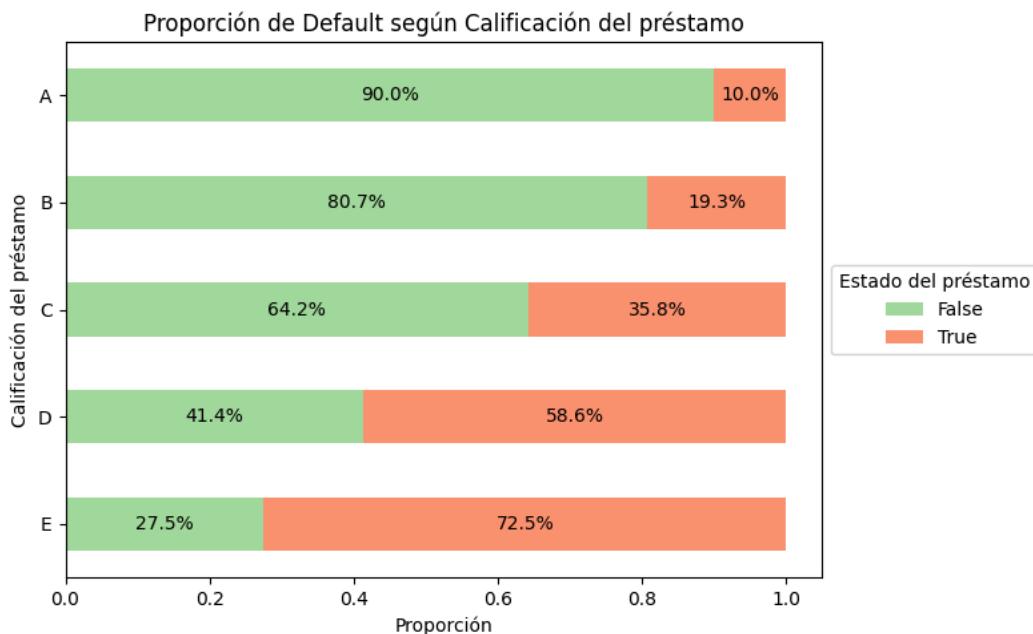
Tabla 6.1.20 - Chi-Cuadrado del Tipo de Vivienda según default y no default

Calificación del Crédito con el Estado del Préstamo

El análisis entre la variable loan_grade y current_loan_status revela una fuerte asociación como era de esperar. Esto lo observamos visualmente en el gráfico de barras apiladas donde se aprecia una fuerte asociación: a medida que la calificación crediticia empeora (de A a E), la proporción de clientes en situación de default aumenta de forma progresiva ([Gráfico 6.1.26](#)):

- Mientras que solo el 10% de los clientes con calificación “A” incurren en impago, este porcentaje asciende hasta el 72.5% en los préstamos con calificación E.
- Esta tendencia creciente sugiere que la calificación asignada a cada préstamo está alineada con el riesgo real de incumplimiento, lo que valida su valor como indicador de solvencia.

Gráfico 6.1.26 - Calificación del Préstamo según el estado del préstamo (P99)



Current_loan_status	No Default	Default
A	14088	1566
B	7311	1751
C	3163	1760
D	1088	1540
E	84	221

Tabla 6.1.21 - Frecuencia de la Calificación del Préstamo según default y no default

Desde un enfoque estadístico, el test de chi-cuadrado respalda esta relación con un valor de $\chi^2 = 4529.30$, con 4 grados de libertad y un p-valor < 0.001 ([Tabla 6.1.22](#)), lo que confirma la dependencia entre ambas variables con un alto nivel de confianza.

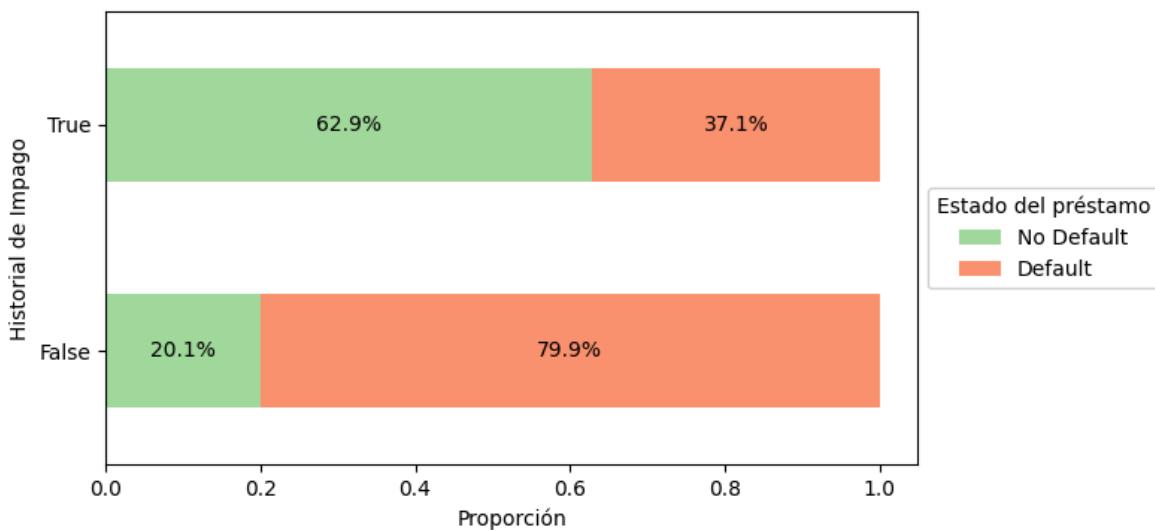
	Valor
Chi-cuadrado	4529.305
Grados de Libertad	4
P-valor	0.0

Tabla 6.1.22 - Chi-Cuadrado de la Calificación del Préstamo según default y no default

Historial Crediticio con el Estado del Préstamo

La variable historical_default, que indica si un cliente ha tenido impagos previos, muestra una relación muy clara con el estado actual del préstamo (current_loan_status). Tal y como se refleja en el gráfico, el 79.9% de los clientes sin historial previo de impago se encuentran actualmente en situación de default, mientras que solo el 37.1% de aquellos con antecedentes de impago han vuelto a caer en morosidad. Esta diferencia es notable y sugiere que, contra la intuición, los clientes sin historial de impago previo podrían presentar un mayor riesgo de impago en el presente. Siendo además estadísticamente significativa esta relación ([Tabla 6.1.23](#))

Gráfico 6.1.27 - Proporción de Default según el historial de impago



	Valor
Chi-cuadrado	2225.653
Grados de Libertad	1
P-valor	0.0

Tabla 6.1.23- Chi-Cuadrado del historial de impago según default y no default

6.2. Análisis de los valores nulos

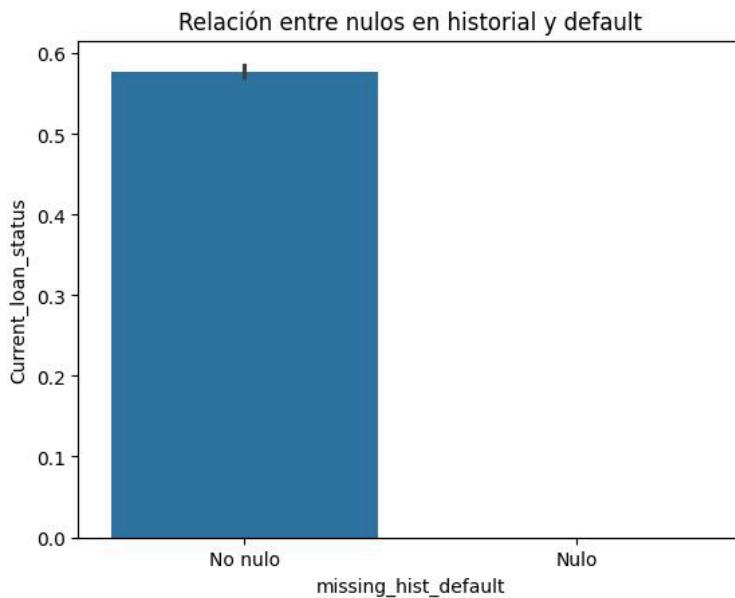
Tras hacer todos los análisis descritos, observamos que el 63.6% de los valores de historical_default eran valores nulos lo que equivale a más de veinte mil registros sin información sobre impagos previos.

Debido a la magnitud, nos resultó prioritario estudiar más en profundidad estos datos faltantes preguntándonos si su distribución era aleatoria o si seguía algún patrón sistemático. Es decir, si su ausencia estaba correlacionada con ciertas características de los clientes, de los préstamos o era completamente al azar.

A partir de esto, la hipótesis que nos planteamos es, que la falta de valores en historical_default, no es completamente aleatoria.

Para contrastar esta hipótesis, se llevó a cabo un análisis exploratorio enfocado en esta variable y su relación con otras. Se compararon las distribuciones de varias variables clave entre dos grupos de datos: por un lado, los registros con valor conocido en historical_default; por otro, los registros donde este valor está ausente.

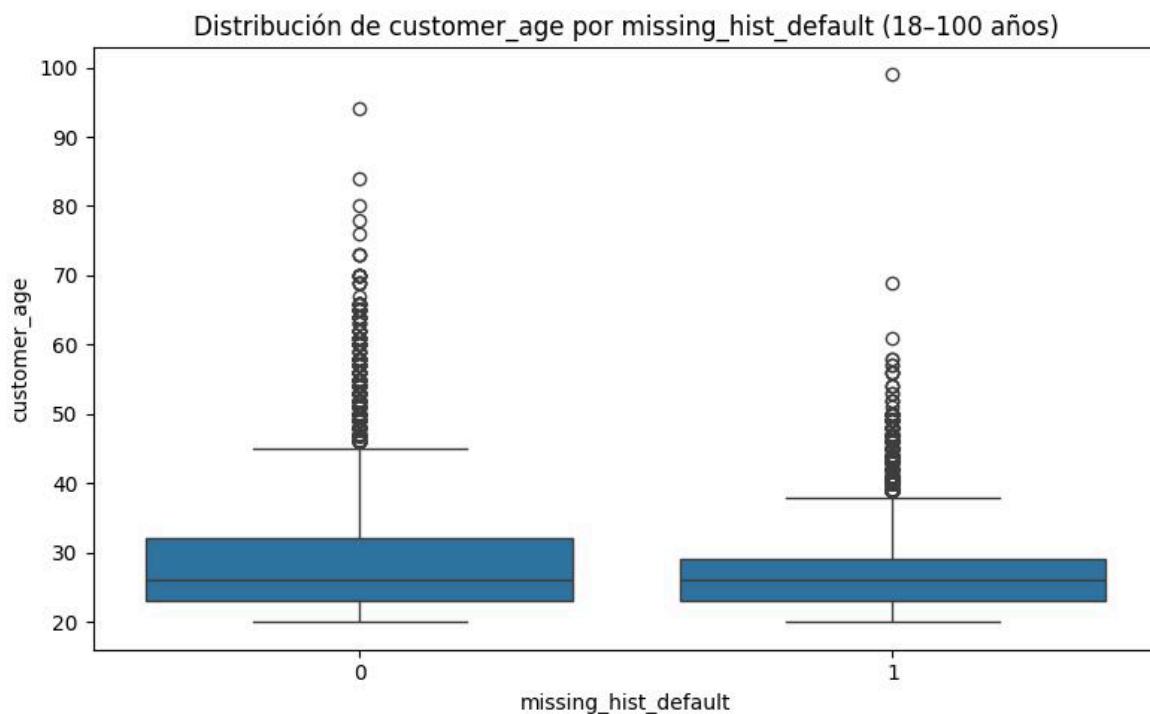
Gráfico 6.2.1 - Relación entre nulos con historial_default



En primer lugar, comparamos historical_default con nuestra variable target como se observa el el [Gráfico 6.2.1](#) en donde nos percatamos de que ninguno de los valores nulos presentaba default en nuestra variable objetivo, sustentando de esta manera nuestra hipótesis inicial de que los valores faltantes no son aleatorios y nos pueden estar dando información de estos perfiles de cliente por lo que se decidió indagar más.

Seguimos con la relación de la variable con customer_age ([Gráfico 6.2.2](#)) utilizando un boxplot para verlo de manera visual. Esto nos muestra que los clientes con valor nulo presentan una mediana inferior y un rango intercuartílico mucho más estrecho que el grupo con dato conocido. Además, se observa que la población con valor nulo tiende a ser más joven en promedio que la población con dato conocido.

Gráfico 6.2.2 - Distribución de customer_age por missing_hist_default

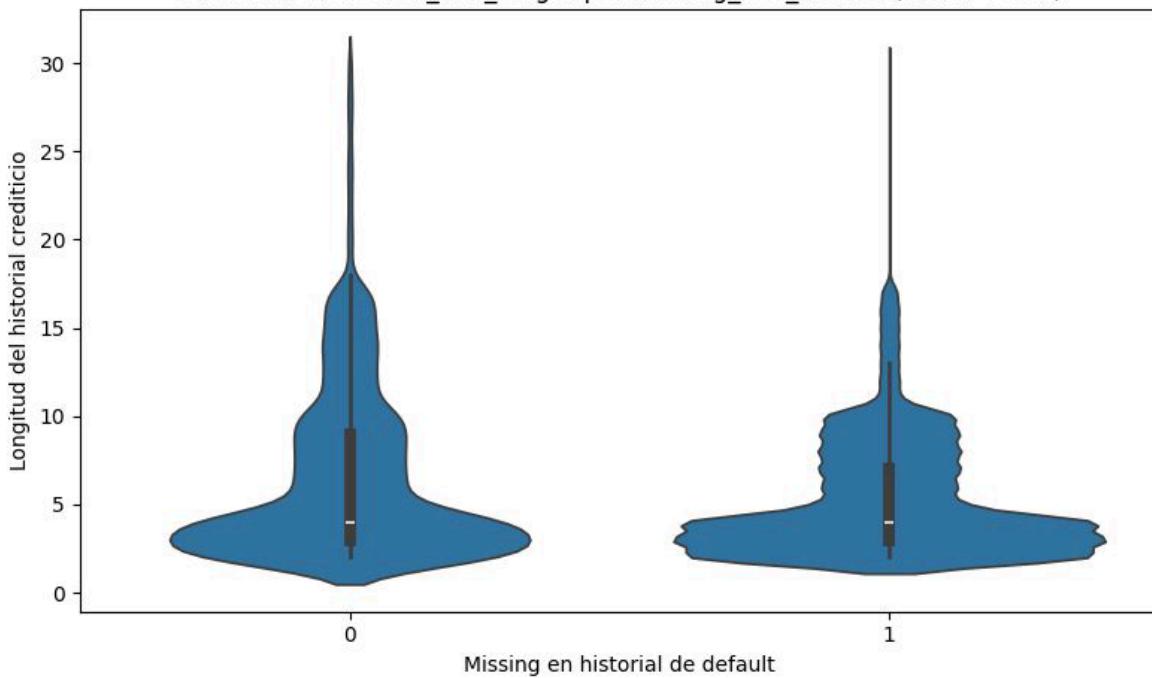


A continuación, realizamos un diagrama violín para comparar la distribución de cred_hist_length con la variable creada missing_hist_default. El [Gráfico 6.2.3](#) nos revela que cuando no hay datos de historical default, la densidad se concentra en el rango de 2 a 6 años, con una mediana inferior y colas más cortas que el grupo que si que tiene datos sobre el historial.

Por el contrario, los clientes con datos en la variable de historical_default muestran una dispersión amplia que se extiende hasta los 30 años de trayectoria. En conjunto, la figura confirma que la falta de historical_default se asocia a historiales crediticios breves, indicativos de clientes recién incorporados al sistema o con poca profundidad documental.

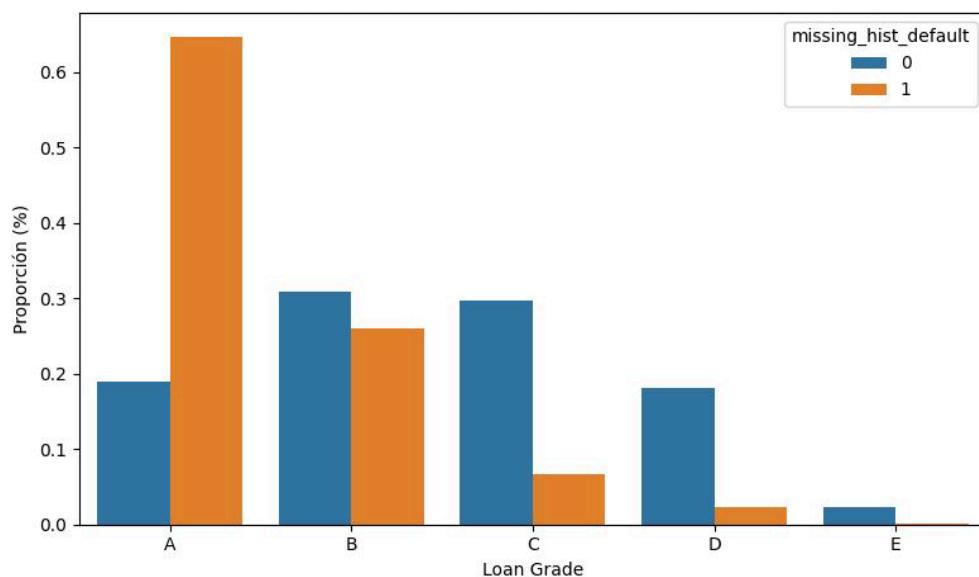
Gráfico 6.2.3 - Distribución de cred_hist_length por missing_hist_default

Distribución de cred_hist_length por missing_hist_default (Violin + Box)



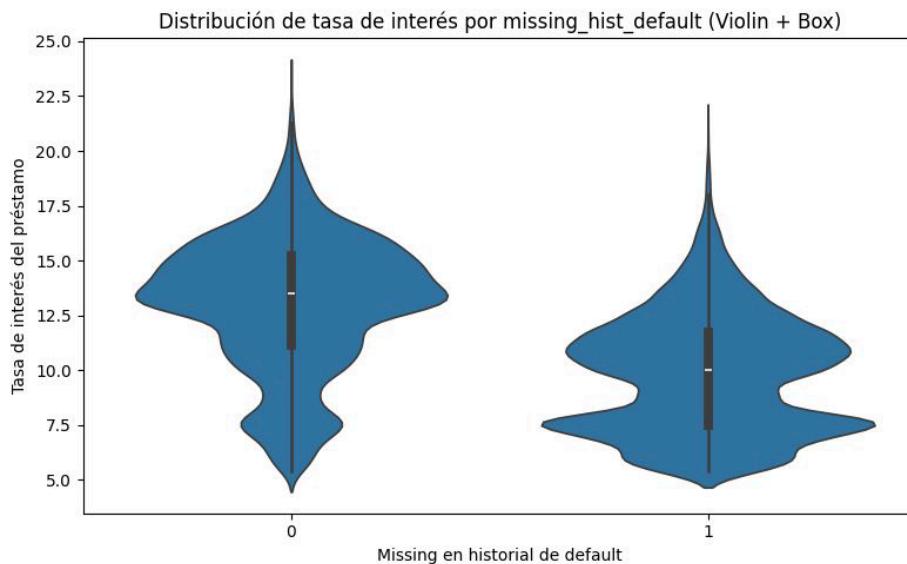
A continuación, nos resultó interesante analizar la relación de la ausencia de historial con atributos directamente vinculados a la evaluación de riesgo y al precio del préstamo: la calificación (loan_grade) y la tasa de interés (loan_int_rate). Por un lado, realizamos un gráfico de barras donde observamos como evolucionaba la calificación del préstamo: cuando $\text{missing_hist_default} = 1$ ([Gráfico 6.2.4](#)), es decir, no hay registros, cerca del 65% de los datos préstamos se encuadran en grado A, mientras que el mismo grado apenas alcanza un 20% entre los registros con historial disponible. A la inversa, los grados intermedios y bajos (B, C, D, E) se concentran mayoritariamente en el grupo con historial ($\text{missing_hist_default} = 0$). Este contraste indica que la ausencia de historical_default no penaliza la calificación crediticia; por el contrario, la entidad otorga sistemáticamente la mejor nota a un segmento sin datos de historial crediticio.

Gráfico 6.2.4 - Proporción de loan_grade por missing_hist_default



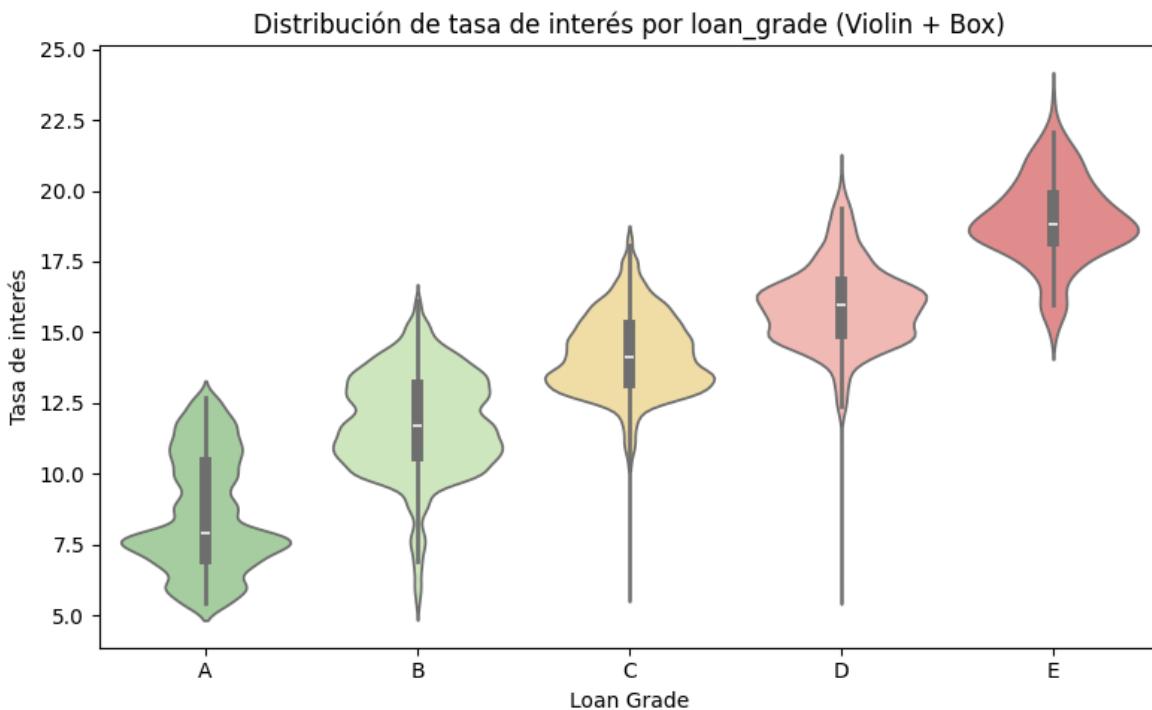
Para ver la distribución de la tasa de interés, consideramos oportuno utilizar otro violin plot ([Gráfico 6.2.5](#)) donde se observa que la distribución de `loan_int_rate` para el grupo sin historial exhibe una mediana sensiblemente inferior y una dispersión más compacta que la del grupo con historial. En concreto, el centro de masa de la densidad se sitúa varios puntos por debajo, reflejando condiciones financieras más favorables.

Gráfico 6.2.5 - Distribución de tasa de interés por missing_hist_default



Para finalizar este apartado, se elaboró un gráfico para examinar la posible relación entre la tasa de interés (loan_int_rate) y la calificación del préstamo (loan_grade). Observamos que, a medida que descendemos en el loan_grade, la mediana de la tasa de interés se eleva de forma progresiva y la dispersión de la distribución se incrementa ([Gráfico 6.2.6](#)). En los grados A y B la densidad aparece concentrada en un rango estrecho, y a partir del grado C, tanto la mediana como el rango intercuartílico se amplían, y en C, D y E surgen colas más prolongadas hacia valores más altos. Este patrón refleja que los préstamos con calificaciones inferiores no solo soportan tasas medias superiores, sino que además presentan una mayor heterogeneidad. Esta relación, nos resultó relevante para realizar la imputación de los valores faltantes de la tasa de interés en función de la calificación crediticia la cual explicaremos en mayor profundidad más adelante.

Gráfico 6.2.6 - Violin plot de la relación de tasa de interés y calificación crediticia



La conclusión que podemos extraer de este análisis es que la ausencia de información en historical_default no se distribuye al azar; identifica de forma consistente a un colectivo más joven y con historiales crediticios breves.

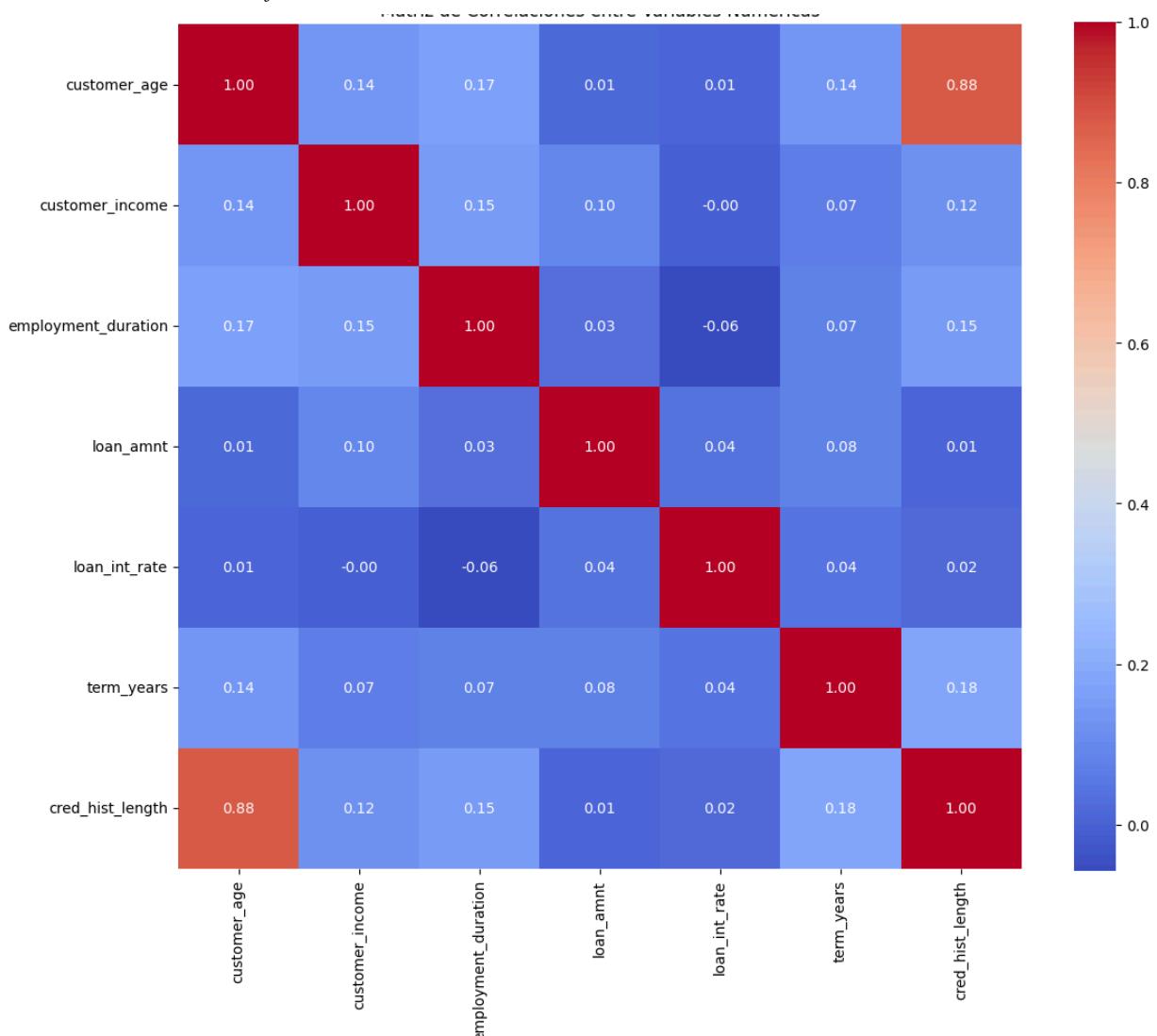
Para reflejar adecuadamente esta circunstancia en los modelos predictivos, hemos considerado conservar los registros afectados y codificar los datos que faltan como una categoría específica.

6.3. Matriz de Correlaciones

Antes de avanzar con el modelado predictivo, se realizó un análisis exhaustivo de las relaciones entre las variables del conjunto de datos. Para ello, se aplicaron tres enfoques complementarios que permiten explorar la estructura de dependencias desde distintas perspectivas: una matriz de correlación para variables numéricas, un análisis para variables categóricas y una matriz combinada que incluye ambos tipos de variables.

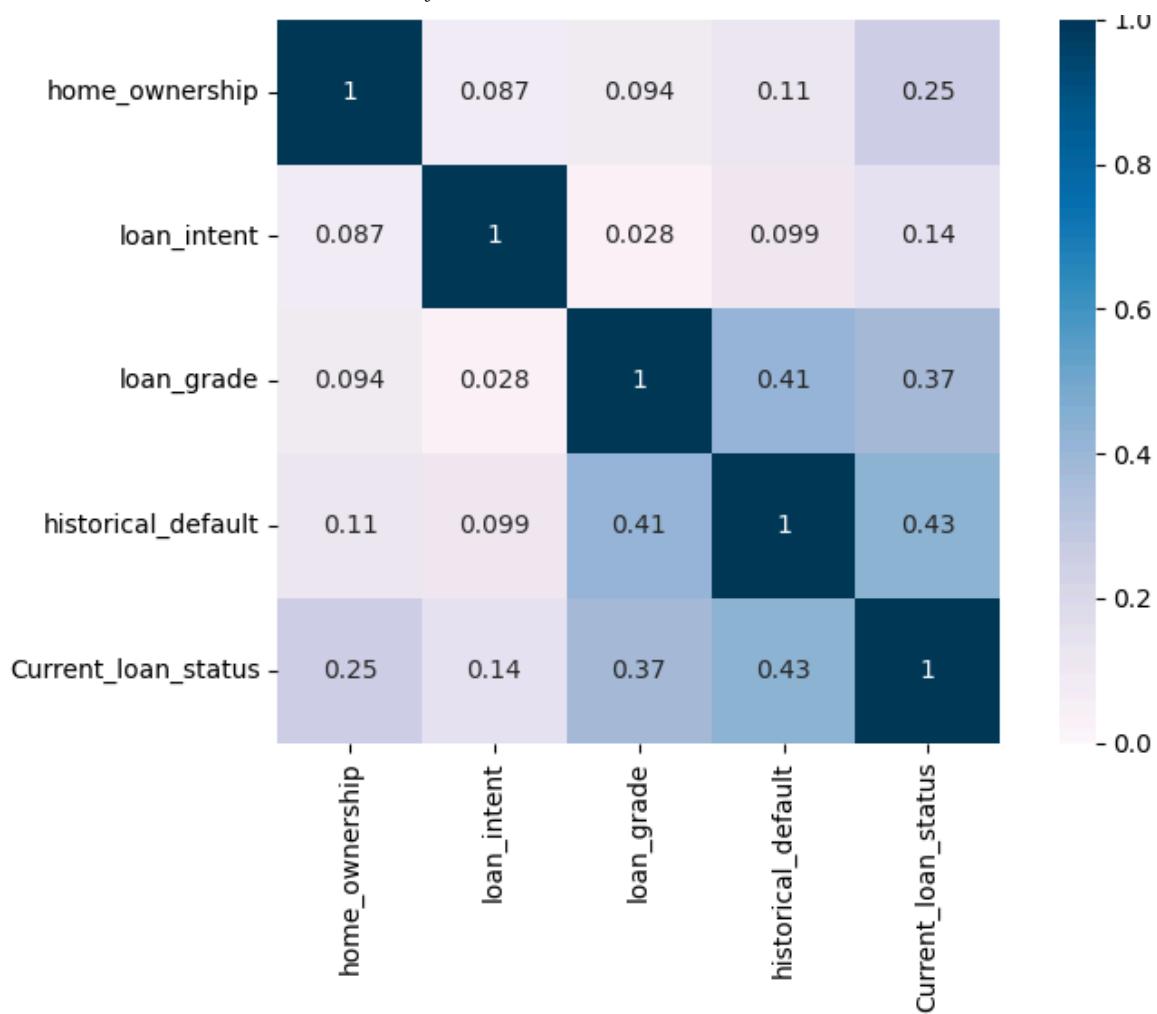
En primer lugar, se empleó la matriz de correlación de Pearson ([Gráfico 6.3.1](#)) para las variables numéricas: customer_age, customer_income, employment_duration, loan_amnt, loan_int_rate, term_years y cred_hist_length. Este análisis permite detectar relaciones lineales y posibles redundancias entre predictores cuantitativos, lo cual es fundamental especialmente en modelos sensibles a la multicolinealidad.

Gráfico 6.3.1 - Matriz de Correlaciones de variables numéricas



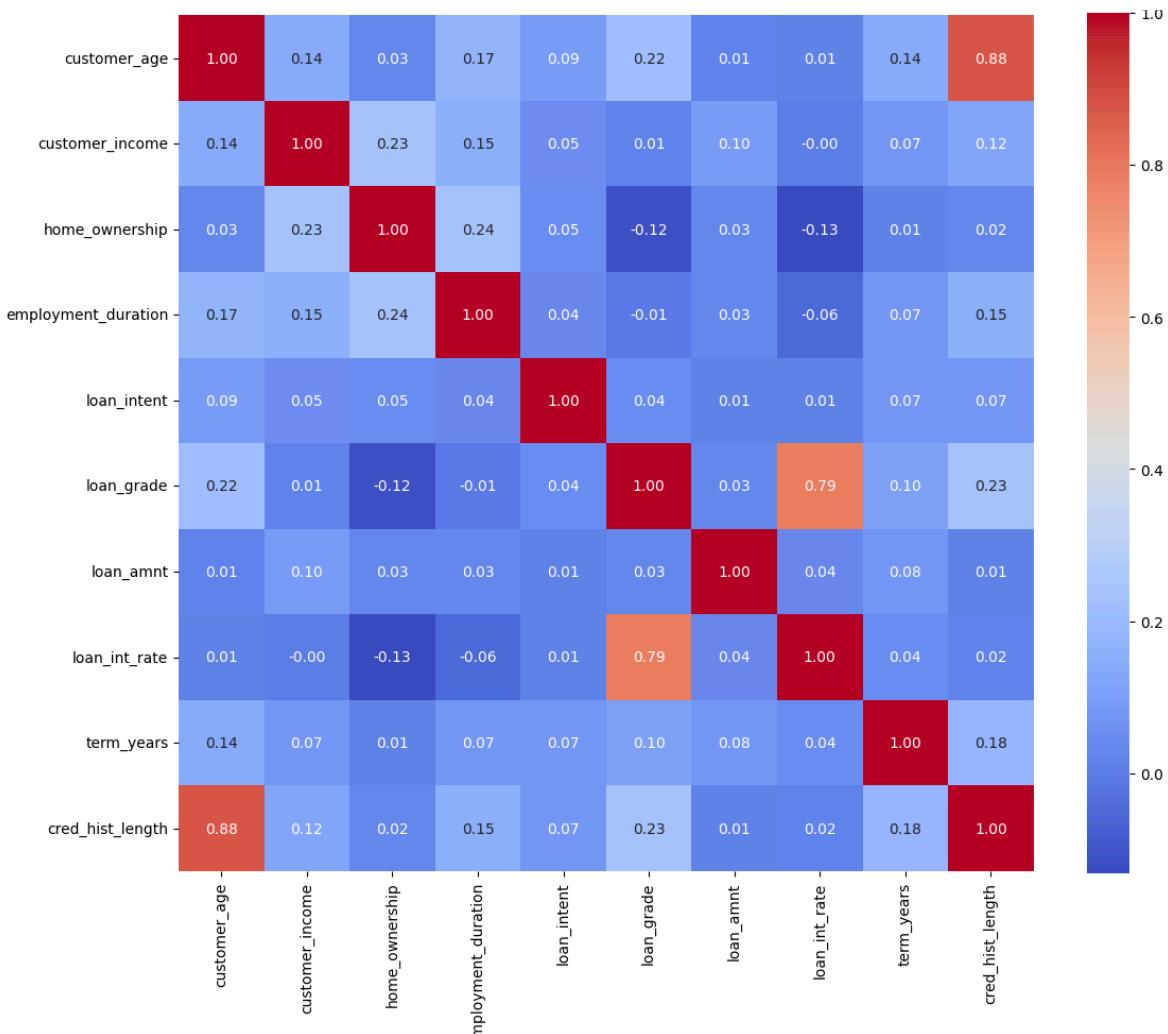
Posteriormente, se analizó la relación entre variables categóricas mediante el coeficiente V de Cramer ([Gráfico 6.3.2](#)), adecuado para identificar asociaciones entre variables cualitativas sin necesidad de asumir orden o linealidad. Esta matriz ofrece una visión más precisa sobre el grado de asociación entre variables como home_ownership, loan_intent, loan_grade e historical_default, incluyendo su relación con la variable objetivo (Current_loan_status).

Gráfico 6.3.2 - Matriz V de Cramer



Por último, se construyó una matriz de correlaciones combinada ([Gráfico 6.3.3](#)), integrando tanto variables numéricas como categóricas (estas últimas transformadas mediante codificación One-Hot). Esta estrategia permite observar interacciones entre todas las variables en un solo análisis, lo que resulta especialmente útil para modelos no lineales como Random Forest.

Gráfico 6.3.3 - Matriz de Correlaciones de variables numéricas y categorías convertidas



En conjunto, los resultados obtenidos no evidencian problemas graves de multicolinealidad ni redundancias críticas, por lo que se consideró adecuado mantener todas las variables como candidatas iniciales para el modelado y la selección posterior de atributos.

6.4. Limpieza y tratamiento de datos

Eliminación de registros duplicados

Se eliminaron los seis registros que aparecían de forma idéntica en el dataset mediante `drop_duplicates()`, garantizando que cada cliente y cada préstamo figuran solo una vez y evitando sesgos derivados de observaciones repetidas.

Eliminación de variables irrelevantes

Se descartaron las variables `customer_id`, ya que cumplía una función de identificación de los registros y `missing_hist_default`, creada exclusivamente para el EDA. Con ello, se simplifica el dataset, reteniendo únicamente las variables útiles para el modelado.

Eliminación de registros sin target o importe

Para garantizar que todos los ejemplos sean utilizables en el training, se eliminaron los cinco registros que tenían valores faltantes en las variables `Current_loan_status` (la variable target) o `loan_amnt`. Estos cinco casos representan menos del 0,02% del total de observaciones, por lo que su eliminación no afecta la representatividad del conjunto de datos. Al descartar estos registros, nos aseguramos de que cada instancia incluya tanto el target a predecir como variables clave, evitando la necesidad de imputaciones arbitrarias en variables esenciales.

Imputación de la tasa de interés por mediana de grupo

Con el fin de evitar cualquier uso de los datos de prueba durante el preprocesamiento, dejamos esta variable con los valores nulos, para tratarla posteriormente en la receta de los modelos con R. En la tabla que se muestra a continuación se muestra la distribución de los valores nulos.

Variable	Cantidad de nulos
<code>customer_id</code>	3
<code>customer_age</code>	0
<code>customer_income</code>	0
<code>home_ownership</code>	0
<code>employment_duration</code>	895
<code>loan_intent</code>	0
<code>loan_grade</code>	0
<code>loan_amnt</code>	1
<code>loan_int_rate</code>	3.116
<code>term_years</code>	0

historical_default	20.737
cred_hist_length	0
current_loan_status	4

Tabla 6.4.1 - Distribución de los Valores Nulos

Eliminación de registros con employment_duration nulo

Sólo el 2,5% de los registros contaba con un valor nulo en la variable de *employment_duration*, por lo que su eliminación no afecta la representatividad del conjunto. El EDA no evidenció una correlación importante de estos valores nulos con la edad, los ingresos, la tasa de interés o el propósito del préstamo. Aunque se observa una ligera mayor proporción de defaults, no resulta suficiente para justificar una imputación. Además, imputar una variable con poca capacidad explicativa habría añadido complejidad y posible variabilidad innecesaria al modelo. Por ello, filtramos el dataset para retener únicamente las filas con *employment_duration* presente, garantizando un flujo de trabajo más sencillo y fiable.

Filtrado de valores extremos en edad y duración de empleo

Se definieron rangos para la edad (18–100 años) y la experiencia laboral (hasta 50 años) como se mencionó en el apartado anterior. De esta forma se eliminaron diez registros con edades fuera de ese rango y dos con más de 45 años de antigüedad laboral. El objetivo fue garantizar que el modelo trabaja sobre datos con rangos realistas.

Compleitud de valores nulos en historical_default para conservar información

Los valores ausentes en la variable *historical_default* se sustituyeron por la etiqueta "No Register", de modo que este campo refleja tres estados posibles: los clientes con incumplimientos previos (True), los que no los han tenido (False) y aquellos sin registro disponible (No Register). Así preservamos la información implícita en la ausencia de valor, permitiendo al modelo diferenciar entre falta de historial y un historial negativo o positivo real.

6.5. Modelos

6.5.1. Librerías en R

En la tabla que se muestra a continuación, mostramos las librerías utilizadas a lo largo de todo el análisis realizado en R:

Librerías	Función principal	Uso
tidymodels	Construir modelos de machine learning	Preprocesamiento, splitting de datos
readr	Importar (leer) archivos	Cargar el archivo csv con los datos
glmnet	Ajustar modelos lineales penalizados	Implementar modelos lineales penalizados
parsnip	Especificar modelos de forma unificada y coherente, independientemente del engine	Ajustar todos los modelos de la misma manera
kknn	Ajustar modelos de Vecinos cercanos (KNN)	Utilizar el modelo de KNN
randomForest	Ajustar modelos de Random Forest	Utilizar el modelo de Random Forest
baguette	Crear modelos de tipo bagging	Utilizar el modelo de Bagging
doParallel	Ejecutar código en paralelo utilizando múltiples núcleos de tu procesador	Ejecutar modelos no lineales más rápido
fastshap	Para calcular los SHAP values	Se utilizó para realizar la interpretabilidad de los modelos
dplyr	Manipulación de datos	Filtrar, unir tablas, transformar columnas, etc.
ggplot2	Visualización de datos	Creación de gráficos
tidyr	Transformar datos	Limpieza de base de datos
tibble	Crear tibble	Transformar data frames

viridis	Escalas de color	Crear gradientes de color
---------	------------------	---------------------------

Tabla 6.5.1- Librerías R

6.5.2. Recodificación de variables

Antes de realizar ningún análisis, realizamos un ajuste mínimo de formatos y tipos de datos que garantiza que el análisis descriptivo se calcule sobre valores coherentes.

Al cargar los datos después de la limpieza en Python fue necesario realizar ajustes a los tipos de datos para poderlos utilizar de manera adecuada en los modelos. A continuación se comparte la tabla con el resumen de los cambios realizados:

Columna	Tipo original	Tipo corregido	Transformación realizada	
0	customer_age	numeric	numeric (sin cambios)	Sin cambios
1	customer_income	numeric	numeric (sin cambios)	Sin cambios
2	home_ownership	character	factor	Conversión a factor
3	employment_duration	numeric	numeric (sin cambios)	Sin cambios
4	loan_intent	character	factor	Conversión a factor
5	loan_grade	character	ordered factor	Conversión a factor ordenado
6	loan_amnt	numeric	numeric (sin cambios)	Sin cambios
7	loan_int_rate	numeric	numeric (sin cambios)	Sin cambios
8	term_years	numeric	numeric (sin cambios)	Sin cambios
9	historical_default	character	factor	Conversión a factor
10	cred_hist_length	numeric	numeric (sin cambios)	Sin cambios
11	Current_loan_status	logical	factor	Conversión a factor binario

Tabla 6.5.2 - Ajustes de tipo aplicados a las variables del Dataset en R

6.5.3. División de datos

Antes de iniciar el preprocesamiento de los datos para la implementación de los modelos, se realizó una división del conjunto de datos en entrenamiento y prueba, con una proporción del 70 % y 30 %, respectivamente. Esta división se llevó a cabo de forma estratificada sobre la variable objetivo, con el fin de mantener la proporción de cada categoría (True/False), dado que dicha variable se encuentra desbalanceada.

6.5.4. Métodos de Preprocesamiento para modelos

En la tabla a continuación, observamos los pasos necesarios de preprocesamiento para los modelos a utilizar en la cual ✓ indica que el método es requerido para el modelo y ✗ indica que no es requerido. El símbolo ○ significa que el modelo se podría beneficiar por la técnica pero esta no es requerida.

Modelo	dummy	zv	impute	decorrelate	normalize	transform
logistic_reg	✓	✓	✓	✓	✗	○
nearest_neighb or	✓	✓	✓	○	✓	✓
bag_tree	✗	✗	✗	○ ¹	✗	✗
svm_*	✓	✓	✓	✓	✓	✓
rand_forest	✗	○	✓ ²	○ ¹	✗	✗
mars	✓	✗	✓	○	✗	○

Tabla 6.5.3 - Preprocesamiento modelos

Notas de la tabla:

1. Eliminar la correlación entre los predictores puede que no ayude a mejorar el rendimiento del modelo. Sin embargo, tener menos predictores correlacionados puede mejorar la estimación de la importancia de las variables. Básicamente, la selección de predictores altamente correlacionados es casi aleatoria.
2. El preprocesamiento necesario para estos modelos depende de la implementación. En particular:
 - Teóricamente, cualquier modelo basado en árboles no requiere imputación de valores faltantes. Sin embargo, muchas implementaciones de ensamblados de árboles sí requieren imputación (Kuhn & Slige, 2023).

A continuación, se describen las recetas de preprocesamiento utilizadas para cada modelo. En el caso de los modelos lineales penalizados, vecinos más cercanos y máquinas de vectores de soporte (SVM), se comienza calculando la mediana de la tasa de interés agrupada por calificación crediticia, utilizando únicamente el conjunto de entrenamiento, con el fin de evitar cualquier uso de los datos de prueba durante el preprocesamiento. Se unen estas medianas a nuestros conjuntos de datos tanto de prueba como entrenamiento para realizar la imputación de la variable en la recta.

Esta decisión de imputar los valores faltantes de la variable de tasa de interés mediante la mediana por calificación crediticia se tomó debido a que la variable presenta una distribución asimétrica con valores atípicos, lo que hace que la mediana sea una medida más robusta del centro que la media, evitando así posibles sesgos. Consideramos esta estrategia adecuada con base en la relación observada entre ambas variables durante el análisis exploratorio de datos (EDA), y la utilizamos para imputar los 895 valores faltantes de la tasa de interés.

```
#Hacemos una tabla con las medianas por calificación de crédito:  
loan_medians <- loan_train %>%  
  group_by(loan_grade) %>%  
  summarise(loan_int_rate_group_median = median(loan_int_rate, na.rm  
= TRUE))  
  
#Hacemos un left join de esta tabla a los datos de train y test  
tomando la información de train también para el test para evitar  
data leakage:  
loan_train <- left_join(loan_train, loan_medians, by = "loan_grade")  
loan_test <- left_join(loan_test, loan_medians, by = "loan_grade")
```

Después de este paso, continuamos con la receta de preprocesamiento utilizada para estos modelos en los cuales se realizan los siguientes pasos:

1. Se excluyen features con varianza casi nula.
2. Se imputaron los valores faltantes de la tasa de interés usando la mediana dentro de cada calificación crediticia.
3. Otorgamos un número a cada nivel de las variables ordinales.
4. Se crearon variables dummy para las variables nominales, estableciendo el parámetro one_hot = FALSE con el fin de evitar colinealidad y generar únicamente $n - 1$ columnas por variable categórica.

5. Nuevamente se excluyen features con varianza casi nula en caso de que exista alguna en las variables dummies creadas.
6. Se realiza la transformación Yeo Johnson a los features numéricos, se utiliza en lugar de Box Cox al contar con variables con valores cero. Esta técnica se utiliza para transformar las variables numéricas con el objetivo de que se acerquen a una distribución normal (gaussiana).
7. Los datos numéricos son normalizados con el objetivo de llevar todas las variables a una misma escala. Esto es especialmente importante cuando los predictores se encuentran en distintas unidades de medida.

```
rec1_loan <- recipe( Current_loan_status ~ . ,
                      data = loan_train) |>
  # Excluimos features con varianza casi nula:
  step_nzv(all_predictors()) |>
  #Imputamos las tasas utilizando la mediana de cada calificación de crédito y eliminamos la tabla de medianas que unimos a la base:
  step_mutate(loan_int_rate = ifelse(is.na(loan_int_rate),
                                      loan_int_rate_group_median,
                                      loan_int_rate)) |>
  step_rm(loan_int_rate_group_median) |>
  #Se otorga un número (1,2,3,4..) a cada nivel de las variables ordinales:
  step_ordinalscore(all_ordered_predictors()) |>
  #Creamos las variables dummy:
  step_dummy(all_nominal_predictors(), one_hot = FALSE) |>
  #Se eliminan las features con varianza casi nula en caso de que hubiera alguna dummy:
  step_nzv(all_predictors()) |>
  #Se realiza la transformación Yeo Johnson a los features numéricos:
  step_YeoJohnson(all_numeric_predictors()) |>
  #Se normalizan los datos numéricos dado que estos están en distintas unidades:
  step_normalize(all_numeric_predictors())
```

Si bien observamos que en la tabla de preprocessamiento ([Tabla 6.5.3](#)) requerido, compartida anteriormente, las regresiones logísticas (Ridge, Lasso y Elastic-Net) no requieren que se normalicen las variables esto habla solo de modelos no penalizados al ser nuestros tres modelos penalizados si se requiere de este paso. El parámetro λ se distribuye entre variables según su escala por lo que las variables más grandes reciben más penalización si no están estandarizadas.

Adicional, se menciona que no se realiza la limpieza de variables altamente correlacionadas dado que como observamos en el apartado de EDA nuestras variables no presentan correlaciones altas, por encima de 0,90, siendo las más relevantes las siguientes:

- cred_hist_length y customer_age: 0,88
- loan_grade y loan_int_rate: 0,79

En el caso de la receta de preprocessamiento para el modelo de bagging, dado que este tipo de modelo no requiere técnicas específicas y solo se sugiere que podría beneficiarse de la eliminación de variables altamente correlacionadas, se optó por no incluir pasos adicionales en la receta, siguiendo el criterio mencionado en el párrafo previo. Únicamente se imputan las observaciones faltantes de la variable de tasa de interés manteniéndonos consistentes con el resto de recetas.

```
rec_bag <- recipe(Current_loan_status ~ ., data = loan_train) |>
  # Imputar las tasas utilizando la mediana por calificación
  crediticia
  step_mutate(loan_int_rate = ifelse(
    is.na(loan_int_rate),
    loan_int_rate_group_median,
    loan_int_rate)) |>
  step_rm(loan_int_rate_group_median))
```

En la receta utilizada para Random Forest, como se mencionó anteriormente, los modelos basados en árboles no requieren imputación de valores faltantes desde el punto de vista teórico. Sin embargo, el motor utilizado (randomForest) no admite valores nulos, por lo que

se requiere aplicar una técnica de imputación. Asimismo, se excluyen las variables con varianza casi nula, ya que no aportan información relevante al modelo.

```
rec_rf <- recipe(Current_loan_status ~ . ,
                   data = loan_train
# Excluye features con varianza casi nula:
step_nzv(all_predictors()) |>
#Imputamos las tasas utilizando la mediana de cada calificación de crédito:
step_mutate(loan_int_rate = ifelse(is.na(loan_int_rate),
                                    loan_int_rate_group_median,
                                    loan_int_rate)) |>
step_rm(loan_int_rate_group_median))
```

Por último, en la receta utilizada para el modelo MARS, y de acuerdo con lo indicado en la tabla de preprocesamiento, se imputan los valores faltantes de la variable de tasa de interés, se crean variables dummy para las variables nominales, se asignan valores numéricos a las variables ordinales y se aplica la transformación de Yeo-Johnson a las variables numéricas.

```
rec_mars <- recipe(Current_loan_status ~ . ,
                     data = loan_train) |>
# Imputamos los datos faltantes de la variable de la tasa de interés:
step_mutate(loan_int_rate = ifelse(is.na(loan_int_rate),
                                    loan_int_rate_group_median,
                                    loan_int_rate)) |>
step_rm(loan_int_rate_group_median)) |>
#Otorgamos un número (1,2,3,4..) a cada nivel de las variables ordinales:
step_ordinalscore(all_ordered_predictors()) |>
#Creamos las variables dummy:
step_dummy(all_nominal_predictors(), one_hot = FALSE) |>
#Se realiza la transformación Yeo Johnson a los features numéricos:
step_YeoJohnson(all_numeric_predictors())
```

6.5.5. Validación Cruzada

Para asegurar una evaluación robusta y replicable del desempeño de los modelos, se crearon dos conjuntos de validación cruzada utilizando 5 particiones y 2 repeticiones en cada uno, estableciendo distintas semillas para cada uno de ellos. Esta elección del número de particiones y repeticiones es con el propósito de poder comparar los modelos de manera consistente manteniendo un tiempo computacional razonable.

El primer conjunto de validación cruzada se utilizó para el ajuste de hiperparámetros en cada uno de los modelos. Lo cual permite encontrar la mejor combinación de parámetros sin utilizar información del conjunto de prueba.

El segundo conjunto de validación cruzada se empleó para realizar el assessment final del desempeño de cada modelo ya ajustado. Esta evaluación independiente permite comparar el rendimiento entre los distintos algoritmos bajo las mismas condiciones, asegurando una comparación justa y evitando el sesgo por sobreajuste durante la selección de parámetros.

6.5.6. Modelos Lineales Penalizados

Se realizaron los modelos lineales penalizados Ridge, Lasso y Elastic-Net en el cuál se seleccionó el siguiente grid para realizar el tuneo y seleccionar el mejor valor.

Modelo	Hiperparámetros	Método grid	Tamaño grid
Ridge	penalty (λ)		100
Lasso	penalty (λ)	grid_max_entropy()	100
Elastic Net	penalty (λ), mixture(α)		300

Tabla 6.5.4- Grid Modelos Lineales Penalizados

Recordemos que el hiperparámetro de penalty controla que tanto se penalizan los coeficientes de las variables independientes mientras que en el caso de mixture del modelo Elastic Net, al tratarse de una combinación de los modelos Ridge ($\alpha=0$) y Lasso ($\alpha=0$),

controla la mezcla de estos dos algoritmos. En cuanto al método utilizado para seleccionar el grid lo consideramos adecuado dado que este genera las combinaciones de los hiperparámetros de manera que estos se distribuyan maximizando la diversidad (entropía) para de esta manera abarcar de mejor manera el espacio de búsqueda.

El tamaño del grid seleccionado para Lasso y Ridge, dado que estos solo cuentan con un hiperparámetro, es de 100 el cual consideramos suficiente para explorar los valores posibles de dicha variable. Esto en conjunto con el método del grid que nos maximiza la cobertura de combinaciones sin tener puntos redundantes.

En el caso del grid para Elastic Net, se decide tomar un tamaño más grande (300) dado que este modelo cuenta con dos hiperparámetros por lo que se requiere de un número mayor de combinaciones para cubrir de manera adecuada las interacciones entre ambos parámetros.

Partiendo de este grid para los tres modelos se construyó el workflow con el modelo a utilizar y la receta de preprocesamiento previamente mencionada. Se evalúan los resultados de los modelos utilizando el primer conjunto de validación cruzada repetida y se recolectan las métricas de cada una de las combinaciones realizadas.

Después, se observan los mejores resultados de las métricas roc_auc y f_meas para evaluar cuál es el mejor valor a seleccionar de los hiperparámetros. Dado nuestro caso de estudio en donde es más perjudicial para una entidad financiera la detección errónea de nuestro evento positivo (Default) por las repercusiones financieras que esto puede implicar vs la detección errónea de los eventos falsos consideramos más apropiado centrarnos en la métrica de f_meas el cuál une las métricas de sensibilidad y especificidad para evaluar que tan bien detecta el modelo los eventos positivos.

Con base en esta métrica, se seleccionó la mejor combinación de hiperparámetros, y se finalizó el workflow con el modelo correspondiente. Luego, se filtraron las métricas para reportar el desempeño del modelo con los valores óptimos encontrados.

Modelo	Penalty	mixture
Ridge	1,15e-10	
Lasso	1,01e-10	
Elastic Net	3,17e-4	0,1453

Tabla 6.5.5- Hiperparámetros Modelos Lineales Penalizados

Como podemos observar los resultados del ajuste de hiperparámetros para los modelos penalizados muestran que tanto Ridge como Lasso seleccionaron valores muy bajos de penalización (penalty), lo que indica que los datos no requerían una regularización fuerte y que el modelo se comporta prácticamente como una regresión logística sin penalización significativa. En el caso del modelo Elastic Net, también se seleccionó un valor bajo de penalty, junto con un valor de mixture cercano a cero (0,1453), lo que se traduce a un comportamiento más cercano al de Ridge que al de Lasso, con una ligera tendencia a la selección de variables. En conjunto, estos resultados sugieren que los datos están bien preparados y no hay predictores irrelevantes o redundantes, por lo que una regularización más agresiva no aporta mejoras significativas en el desempeño de los modelos.

Finalmente, se utilizó el segundo conjunto de validación cruzada repetida para realizar un assessment independiente. Las métricas obtenidas en este segundo conjunto fueron empleadas para comparar el desempeño entre los distintos modelos, evitando así el sesgo por sobreajuste derivado del proceso de ajuste de hiperparámetros.

A continuación se comparte la comparativa de los resultados obtenidos con los modelos lineales penalizados:

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
Ridge	0,967	0,765	0,976	0,858	0,809	0,185	0,925
Lasso	0,956	0,828	0,978	0,833	0,830	0,207	0,930
Elastic Net	0,956	0,827	0,978	0,833	0,830	0,207	0,929

Tabla 6.5.6 - Resultados Modelos Lineales Penalizados

Los resultados presentados en la tabla anterior muestran resultados competitivos entre los tres modelos penalizados evaluados: Ridge, Lasso y Elastic Net. En general, Lasso y Elastic Net obtienen los mejores valores en las métricas más relevantes del caso, como f_meas (0,830), roc_auc (0,978) las cuales engloban a las métricas de precisión, sensibilidad y especificidad, y acuracidad (0,930 y 0,956 respectivamente), superando a Ridge.

La métrica f_meas, que equilibra precisión y sensibilidad, es particularmente relevante en este contexto dado que el objetivo del modelo es identificar correctamente los casos positivos (defaults), minimizando los falsos negativos. En ese sentido, tanto Lasso como Elastic Net resultan más efectivos al lograr una mejor capacidad de detección sin sacrificar precisión.

Si bien Lasso y Elastic Net tienen un desempeño casi idéntico, se podría optar por Lasso como modelo seleccionado siguiendo el criterio de la Navaja de Ockham debido a su simplicidad: al forzar coeficientes exactamente a cero, facilita la interpretación del modelo y la identificación de variables más relevantes. Por lo tanto, considerando tanto el rendimiento como la interpretabilidad, Lasso es la mejor alternativa entre los modelos lineales penalizados evaluados.

6.5.7. Modelos No Lineales

Se realizaron los modelos no lineales para los cuales se seleccionaron los siguientes grids para realizar el tuneo y seleccionar el valor más óptimo:

Modelo	Hiperparámetros	Método grid	Tamaño grid
KNN	neighbors (8,9,10,11,12) dist_power (1,2)		$5 \times 2 = 10$
Bagging	cost_complexity log-scale: 1e-10 a 1e-1)	expand.grid	$10 \times 4 \times 3 = 120$

	tree_depth (10,15,20,25) min_n (5,10,15)		
SVM	cost, rbf_sigma: 10 valores cada uno, distribuidos logarítmicamente con value_seq()		$10 \times 10 = \mathbf{100}$
Random Forest	mtry (5,10,15,25) trees (500,1000) min_n (10,50)		$4 \times 2 \times 2 = \mathbf{16}$
MARS	num_terms (de 10 a 80, salto de 5) prod_degree (1,2,3)		$15 \times 3 = \mathbf{45}$

Tabla 6.5.7 - Grid Modelos No Lineales

Para todos los modelos el método de grid utilizado es el de expand grid con el que se generan todas las combinaciones posibles de los hiperparámetros. A continuación, se comparte una explicación detallada de la selección de valores para cada modelo:

- K-Nearest Neighbors (KNN): se seleccionaron valores de neighbors entre 8 y 12, junto con dos valores posibles para dist_power (1 y 2) recordando que neighbors especifica el número de puntos considerados para clasificar una observación y dist_power es la manera de medir las distancias (1 = Manhattan que es suma las diferencias absolutas entre cada coordenada, 2 = Euclídea que mide la distancia en línea recta entre dos puntos). Se genera un grid de 10 combinaciones permitiendo evaluar el efecto de distintos tamaños de vecindario y métricas de distancia.
- Bagging: el grid incluyó 10 valores logarítmicamente espaciados para cost_complexity que controla el grado de poda del árbol penalizando la complejidad

para evitar sobreajuste en el modelo , combinados con tree_depth (10, 15, 20, 25) hablando de la profundidad máxima del árbol y min_n (5, 10, 15) el cuál establece el número mínimo de observaciones requeridas para dividir un nodo, resultando en 120 combinaciones.

- SVM: para los hiperparámetros cost y rbf_sigma, se generaron 10 valores logarítmicamente espaciados para cada uno, generando un grid de 100 combinaciones. De esta manera aseguramos una buena cobertura del espacio a tunear. Recordemos que el costo controla la tolerancia del modelo a errores mientras que rbf_sigma controla cuánto influye cada punto de entrenamiento en la predicción.
- Random Forest: se exploraron combinaciones entre mtry (5, 10, 15, 25), trees (500, 1000) y min_n (10, 50), dando como resultado 16 combinaciones. El tamaño reducido del grid es suficiente debido a la robustez de este modelo y la baja sensibilidad de sus hiperparámetros a pequeños cambios para encontrar una buena solución sin gastar tiempo y recursos innecesarios. Se destaca que mtry establece el número de predictores considerados en cada división de árbol, trees es número total de árboles y min_n establece el número mínimo de observaciones por nodo terminal.
- MARS (Multivariate Adaptive Regression Splines): se utilizó un grid de 15 valores para num_terms (de 10 a 80 con saltos de 5) y 3 valores para prod_degree (1, 2, 3), generando 45 combinaciones. Esta selección permite capturar modelos desde simples hasta complejos mediante num_terms que determina el número máximo de funciones base, evaluando tanto términos lineales como interacciones regulados mediante el parámetro de prod_degree.

En todos los casos, los grids fueron diseñados para maximizar la eficiencia exploratoria, buscando un balance entre distintas combinaciones y tiempo computacional.

A continuación se comparten los hiperparámetros seleccionados para cada modelo en base a los mejores resultados obtenidos para la métrica más relevantes en nuestro caso de uso (f_{meas}):

Modelo	Hiperparámetros
KNN	neighbors = 11 dist_power = 1
Bagging	cost_complexity = 1e-6 tree_depth = 25 min_n = 15
SVM	cost = 10.1 rbf_sigma = 0,0774
Random Forest	mtry = 5 trees = 1.000 min_n = 10
MARS	num_terms = 30 prod_degree = 3

Tabla 6.5.8 - Hiperparámetros Modelos No Lineales

La selección de los hiperparámetros óptimos para cada modelo se realizó en función del mejor desempeño obtenido según la métrica f_{meas} , priorizando la capacidad de los modelos para detectar correctamente los casos positivos (defaults) sin comprometer la precisión. A continuación, se detallan los valores elegidos y su interpretación para cada caso:

- KNN seleccionó neighbors = 11 y dist_power = 1, lo que indica que el modelo logra su mejor desempeño utilizando una métrica de distancia Manhattan y un vecindario moderadamente amplio. Esto sugiere que un número ligeramente mayor de vecinos ayuda a suavizar las predicciones, lo que mejora la generalización sin perder sensibilidad.

- En el modelo de Bagging, el mejor resultado se obtuvo con un `cost_complexity` muy bajo (`1e-6`), `tree_depth = 25` y `min_n = 15`. Esta selección favorece a árboles relativamente profundos y poco podados, lo cual es coherente con el enfoque de Bagging que se beneficia de modelos base complejos combinados por agregación para reducir la varianza sin sobreajuste.
- Para el modelo SVM, los valores óptimos fueron `cost = 10.1` y `rbf_sigma = 0,0774`. Esta combinación indica que el modelo penaliza fuertemente los errores de clasificación (costo alto), al mismo tiempo que conserva una moderada flexibilidad en el kernel. Esto permite un ajuste a la frontera de decisión sin sobreajuste, maximizando la capacidad de discriminación.
- En Random Forest, se seleccionó `mtry = 5`, `trees = 1.000` y `min_n = 10`. Esta selección indica que el mejor desempeño se logra cuando se explora un subconjunto pequeño de predictores por división, con un gran número de árboles y un tamaño mínimo moderado de nodos terminales. Este ajuste favorece la diversidad entre árboles y permite capturar patrones complejos con buena estabilidad.
- Por último, MARS alcanzó su mejor `f_meas` con `num_terms = 30` y `prod_degree = 3`, lo que indica que el modelo se beneficia de un número moderado de términos con interacciones de hasta tercer orden. Esta selección permite modelar relaciones no lineales sin introducir sobreajuste.

En conjunto, estas combinaciones muestran que cada modelo requiere un equilibrio distinto entre flexibilidad y generalización para maximizar su capacidad de predicción. La selección cuidadosa de estos hiperparámetros es clave para lograr los mejores resultados y que estos fueran comparables entre modelos.

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
KNN	0,983	0,822	0,984	0,928	0,872	0,184	0,950
Bagging	0,989	0,886	0,992	0,956	0,919	0,193	0,968
SVM	0,987	0,886	0,983	0,946	0,915	0,195	0,966
Random Forest	0,991	0,895	0,993	0,965	0,929	0,193	0,971
MARS	0,976	0,844	0,986	0,901	0,871	0,195	0,948

Tabla 6.5.9 - Resultados Modelos No Lineales

Los resultados de la tabla comparativa muestran que el modelo de Random Forest presenta el mejor rendimiento general entre los modelos no lineales evaluados. Este modelo obtuvo los valores más altos en todas las métricas clave: la mayor sensibilidad (0,895), que refleja su capacidad de identificar correctamente los casos positivos (defaults), así como la mayor precisión (0,965) y f-measure (0,929), que muestran un excelente equilibrio entre exactitud y cobertura. Además, alcanzó el mejor valor de AUC (0,993), indicando una buena capacidad de discriminación entre clases, y la mayor acuracidad con un valor de 0,971.

Si bien Bagging y SVM también mostraron métricas competitivas, Random Forest los superó en todas las métricas para este problema, incluyendo especificidad, sensibilidad y f-measure, lo que lo convierte en el mejor modelo dentro de los evaluados. Por tanto, considerando el objetivo del caso de estudio y el balance entre precisión, sensibilidad y capacidad general de predicción, Random Forest se posiciona como el modelo más adecuado.

6.5.8. Comparativa Modelo Lineal Seleccionado vs Modelo No Lineal

Por último, realizamos la comparativa de las métricas obtenidas con el mejor modelo Lineal y no Lineal para realizar la selección de nuestro modelo final:

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
Lasso	0,956	0,828	0,978	0,833	0,830	0,207	0,930
Random Forest	0,991	0,895	0,993	0,965	0,929	0,193	0,971

Tabla 6.5.10- Resultados Modelos No Lineal vs Lineal Penalizado

La comparación entre los modelos Lasso y Random Forest revela diferencias claras en cuanto a su capacidad predictiva. Si bien Lasso presenta métricas aceptables y destaca por su simplicidad e interpretabilidad, el modelo de Random Forest supera a Lasso en todas las métricas, especialmente en aquellas más relevantes para este caso de estudio.

En particular, Random Forest alcanza un valor de f-measure de 0,929, significativamente superior al 0,830 de Lasso, lo que indica una mayor capacidad para equilibrar precisión y sensibilidad. Además, obtiene una mayor sensibilidad (0,895 vs. 0,828), lo cual es especialmente importante en contextos donde no identificar correctamente un caso positivo (como un default) puede tener consecuencias costosas. Adicionalmente, cuenta con una mejor área bajo la curva ROC (0,993 vs. 0,978), lo que refleja una mejor capacidad del modelo para discriminar entre clases en todos los umbrales de decisión posibles.

Aunque Lasso tiene la ventaja de facilitar la interpretabilidad del modelo al reducir la cantidad de variables utilizadas, esta ventaja es superada por el excelente desempeño predictivo de Random Forest adicional a que parte del presente trabajo se basa en realizar la interpretabilidad del modelo mediante técnicas como SHAP. Por tanto, considerando tanto el objetivo del análisis como las métricas evaluadas, Random Forest es el modelo más adecuado para ser implementado, ya que ofrece una mayor capacidad de detección de casos relevantes sin sacrificar precisión ni estabilidad.

Posterior a seleccionar Random Forest como el mejor modelo durante el proceso de validación cruzada, se procedió a realizar el ajuste final utilizando la totalidad del conjunto de entrenamiento y evaluándose posteriormente sobre el conjunto de prueba. Los resultados obtenidos reflejan un desempeño sobresaliente del modelo, alcanzando una acuracidad

(accuracy) del 97,5% y un valor F1 (f_meas) de 93,8%, lo que indica un muy buen equilibrio entre precisión y sensibilidad en la identificación de casos positivos (defaults).

En particular, la sensibilidad (sens) de 91% demuestra la capacidad del modelo para detectar correctamente la mayoría de los casos de incumplimiento, mientras que la especificidad (spec) del 99,2% indica una excelente capacidad para reconocer correctamente los casos negativos. Además, la métrica roc_auc de 99,4% confirma que el modelo tiene un excelente poder discriminativo entre las clases.

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
Random Forest	0,992	0,910	0,994	0,968	0,938	0,195	0,975

Tabla 6.5.11- Resultados Random Forest Conjunto de prueba

6.5.9. Interpretabilidad del modelo seleccionado

Con el objetivo de comprender mejor el comportamiento del modelo Random Forest y justificar sus predicciones, se implementó la técnica de interpretabilidad SHAP (SHapley Additive exPlanations). Esta herramienta permite descomponer la predicción realizada para una observación individual en la contribución que cada una de las variables tuvo en dicha predicción, proporcionando una visión transparente del proceso de decisión del modelo.

En los gráficos generados, se analizaron las contribuciones de las características para cada una de las observaciones de nuestra base de datos, comparando su influencia tanto en la predicción de la clase TRUE (default) como en la de FALSE (no default). Los valores del eje x representan la magnitud de la contribución de cada variable a la predicción final. Es importante mencionar que en el caso de las variables numéricas estas se normalizaron con el propósito de que se encontraran en las mismas unidades y se facilitara su entendimiento mediante el gradiente de color.

Gráfico 6.5.1 - SHAP Summary Plot Variables Categóricas

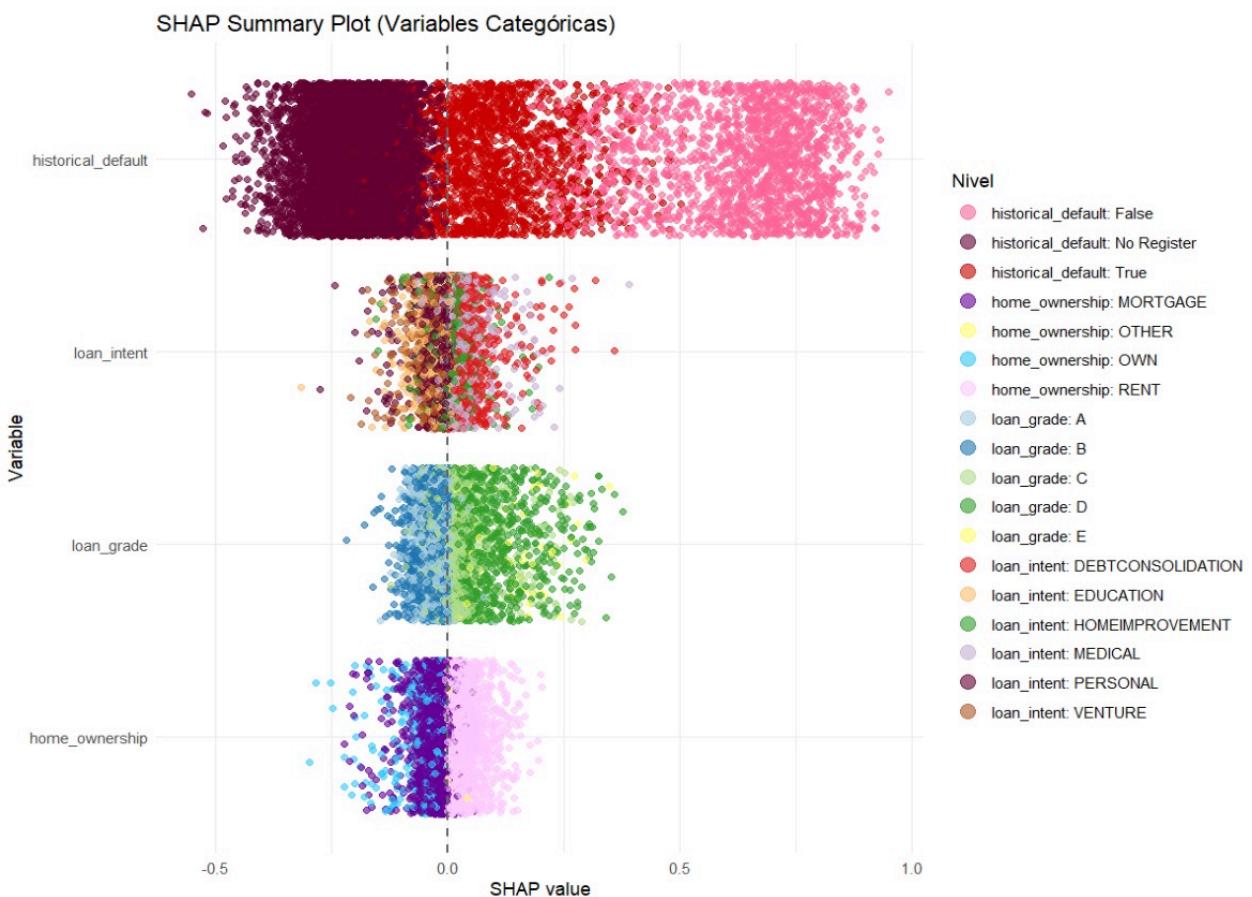
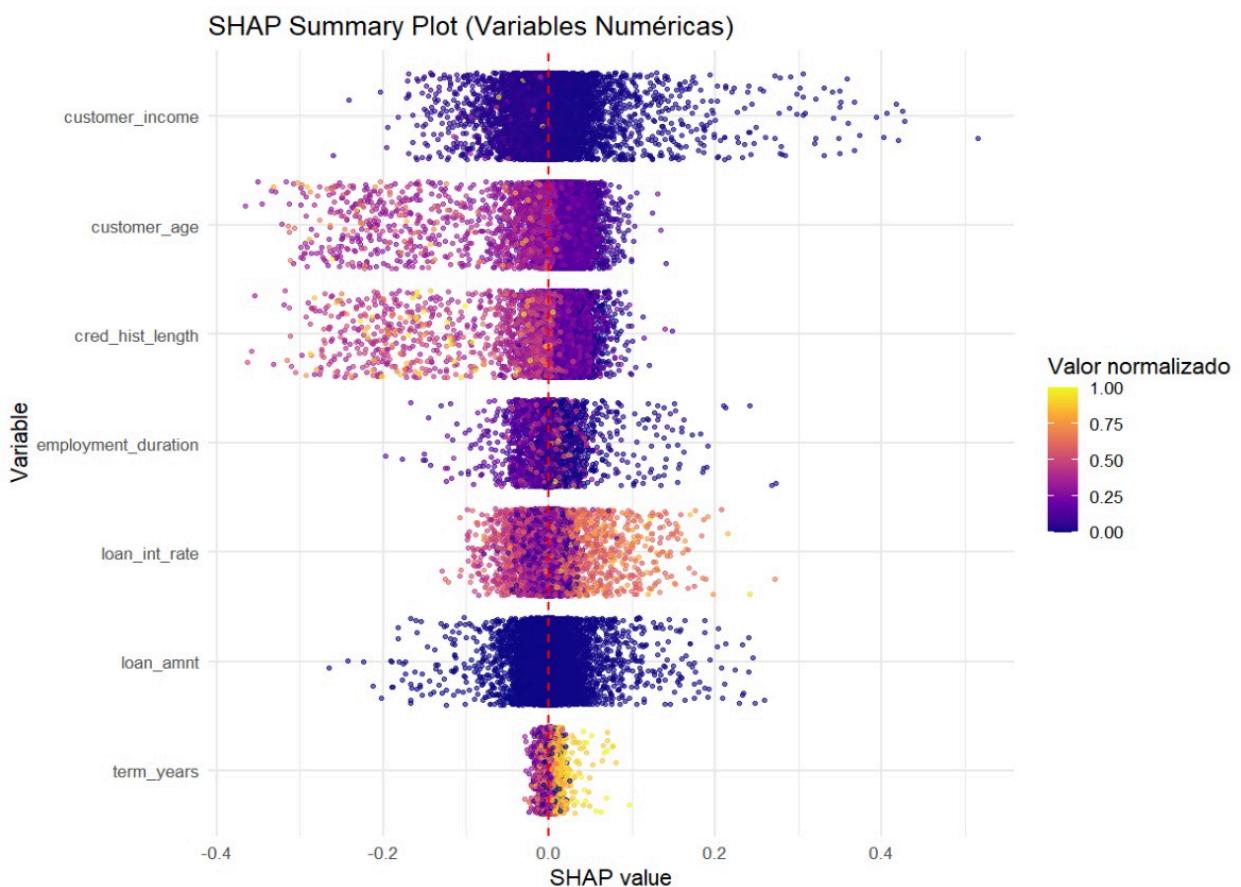
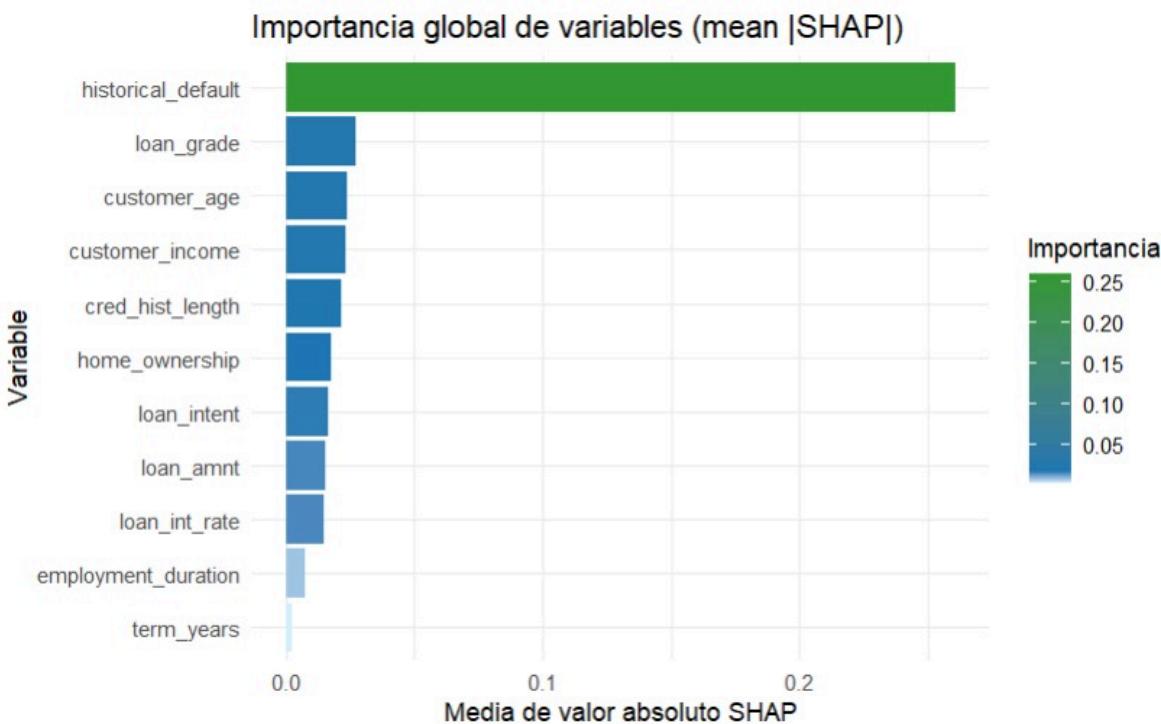


Gráfico 6.5.2 - SHAP Summary Plot Variables Numéricas



Al descomponer su impacto por categoría, se observa que los valores "True" y "No Register" en historical_default contribuyen de manera importante a predecir default (valores SHAP > 0), mientras que los valores faltantes (No Register) se asocian fuertemente a la predicción de no default (valores SHAP < 0). Este patrón podría indicar un posible sesgo del modelo dado que si hay muchos valores faltantes en esta variable y estos se correlacionan con la clase negativa, el modelo podría estar sobreajustado.

Gráfico 6.5.3 - SHAP Bar Plot



Esto se refuerza con el gráfico de barras de importancia global , en el que se visualiza la media del valor absoluto de los SHAP values para cada variable. En este [Gráfico 6.5.3](#), historical_default muestra una magnitud media significativamente mayor que el resto de las variables, lo que confirma su dominancia dentro del modelo.

6.5.10. Comparativa modelo sin historical default

Como se mencionó previamente, al analizar la variable historical_default se identificó que la ausencia de datos no era aleatoria ([Gráfico 6.2.1](#)), sino que contenía información relevante: específicamente, ninguno de los registros con valores nulos presentó incumplimiento en la variable objetivo. Esta observación sugirió un posible sesgo en el modelo al incluir dicha variable. Por ello y con los resultados obtenidos mediante SHAP, se optó por entrenar nuevamente los modelos excluyendo historical_default, con el objetivo de comparar los resultados obtenidos y evaluar el impacto que esta variable tenía en nuestros modelos.

Se utilizaron los mismos algoritmos que en el escenario original para realizar las predicciones y seleccionar el modelo más adecuado. De igual manera, se mantuvieron los mismos grids de hiperparámetros previamente definidos ([Tabla 6.5.7](#) y [Tabla 6.5.4](#)) , ya que estos habían sido cuidadosamente diseñados para cubrir un rango representativo y razonable de valores posibles en cada caso. Esta consistencia permite realizar una comparación justa entre ambos escenarios bajo condiciones controladas.

La única excepción fue el modelo K-Nearest Neighbors (KNN), para el cual se decidió ampliar el rango del hiperparámetro neighbors, incorporando los valores 13 y 14. Esta decisión se basó en los resultados obtenidos en el primer escenario, donde los valores cercanos a 12 ofrecieron un buen desempeño, pero no mostraban una clara tendencia decreciente en la métrica de evaluación. Al extender el rango, se buscó confirmar si el modelo podría beneficiarse de un número ligeramente mayor de vecinos, permitiendo una mayor estabilidad en la clasificación sin pérdida significativa de precisión, especialmente en un contexto donde la distribución de clases podía haberse visto alterada tras la eliminación de la variable `historical_default`.

Modelo	Hiperparámetros
Ridge	penalty = 1,1533e-10
Lasso	penalty = 0,0006
Elastic Net	penalty = 5,5660e-5 mixture = 0,8504
KNN	neighbors = 12 dist_power = 1
Bagging	cost_complexity = 1e-7 tree_depth = 20 min_n = 5
SVM	cost = 10,1

	rbf_sigma = 0,0774
Random Forest	mtry = 5 trees = 1.000 min_n = 10
MARS	num_terms = 25 prod_degree = 3

Tabla 6.5.12- Hiperparámetros Modelos Sin Historical Default

La [Tabla 6.5.12](#) muestra los hiperparámetros óptimos seleccionados tras la validación cruzada. Los ajustes observados reflejan que, al eliminar la variable historical_default, algunos modelos requerían mayor regularización, profundidad o complejidad estructural para mantener un buen rendimiento. Esto sugiere que historical_default era una variable con alto poder predictivo.

Estos cambios también evidencian que algunos algoritmos, como Bagging y Elastic Net, son más sensibles a la pérdida de variables altamente predictivas, mientras que otros como SVM y Random Forest muestran mayor estabilidad en sus configuraciones óptimas.

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
Ridge	0,963	0,417	0,852	0,749	0,535	0,116	0,850
Lasso	0,952	0,469	0,853	0,720	0,568	0,135	0,852
Elastic Net	0,952	0,470	0,853	0,719	0,568	0,136	0,852

Tabla 6.5.13 - Resultados Modelos Lineales Penalizados sin historical_default

Los resultados presentados en la [Tabla 6.5.13](#) muestran una disminución notable en el desempeño de los modelos lineales penalizados (Ridge, Lasso y Elastic Net) al excluir la variable historical_default. En particular, se observa una caída significativa en métricas clave como la sensibilidad (sens) y la métrica F1 (f_meas), que son especialmente importantes en este contexto, ya que reflejan la capacidad del modelo para detectar correctamente los incumplimientos.

Dentro de este grupo, el modelo Lasso y el modelo Elastic Net presentan un rendimiento prácticamente idéntico, alcanzando una sensibilidad de 0,470 y una F1 de 0,568. Aunque sin gran diferencia con Ridge, estos dos modelos superan claramente a Ridge en ambas métricas, lo que los posiciona como las mejores opciones dentro de los modelos lineales penalizados, siguiendo nuevamente el criterio de la navaja de Ockham Lasso es la mejor alternativa entre los modelos lineales penalizados evaluados.

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
KNN	0,973	0,513	0,873	0,833	0,635	0,128	0,877
Bagging	0,978	0,708	0,920	0,896	0,791	0,164	0,922
SVM	0,975	0,664	0,901	0,874	0,754	0,160	0,910
Random Forest	0,986	0,712	0,933	0,929	0,806	0,159	0,929
MARS	0,965	0,573	0,882	0,811	0,671	0,147	0,883

Tabla 6.5.14- Resultados Modelos No Lineales

La evaluación comparativa de los modelos no lineales nos muestra nuevamente que Random Forest es el modelo con mejor desempeño global. Este modelo presenta la mayor sensibilidad (0,712), lo que indica su capacidad para identificar correctamente los casos positivos (default), manteniendo además una alta especificidad (0,986) y la mayor precisión (0,929). Adicionalmente, alcanza el mejor valor de F-measure (0,806) y la mayor área bajo la curva ROC (0,933), lo que sugiere un excelente equilibrio entre verdaderos positivos y falsos positivos a través de distintos umbrales.

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
Lasso	0,952	0,469	0,853	0,720	0,568	0,135	0,852
Random Forest	0,986	0,712	0,933	0,929	0,806	0,159	0,929

Tabla 6.5.15- Comparativa Modelos Lineales vs No Lineal sin historical_default

Realizamos la comparativa del modelo lineal contra el no lineal y nuevamente llegamos a la conclusión de que el modelo más óptimo para este caso es Random Forest al superar de manera importante los indicadores obtenidos con Lasso.

El modelo Random Forest, entrenado sin la variable historical_default, logró un desempeño sólido en el conjunto de prueba, alcanzando una precisión de 0,924, una sensibilidad de 0,713 y un f-measure de 0,805. Aunque estas métricas son inferiores respecto al modelo que incluía dicha variable, los resultados siguen siendo competitivos y reflejan una capacidad adecuada para identificar correctamente los casos de incumplimiento sin depender de una variable potencialmente problemática por su elevada proporción de valores nulos y su fuerte asociación con la variable objetivo. Estos resultados sugieren que, incluso sin historical_default, es posible construir un modelo robusto y más alineado con criterios de interpretabilidad y equidad, aunque con cierto sacrificio en rendimiento predictivo.

Modelo	spec	sens	roc_auc	precision	f_meas	detection_prevalence	accuracy
Random Forest	0,985	0,713	0,935	0,924	0,805	0,160	0,928

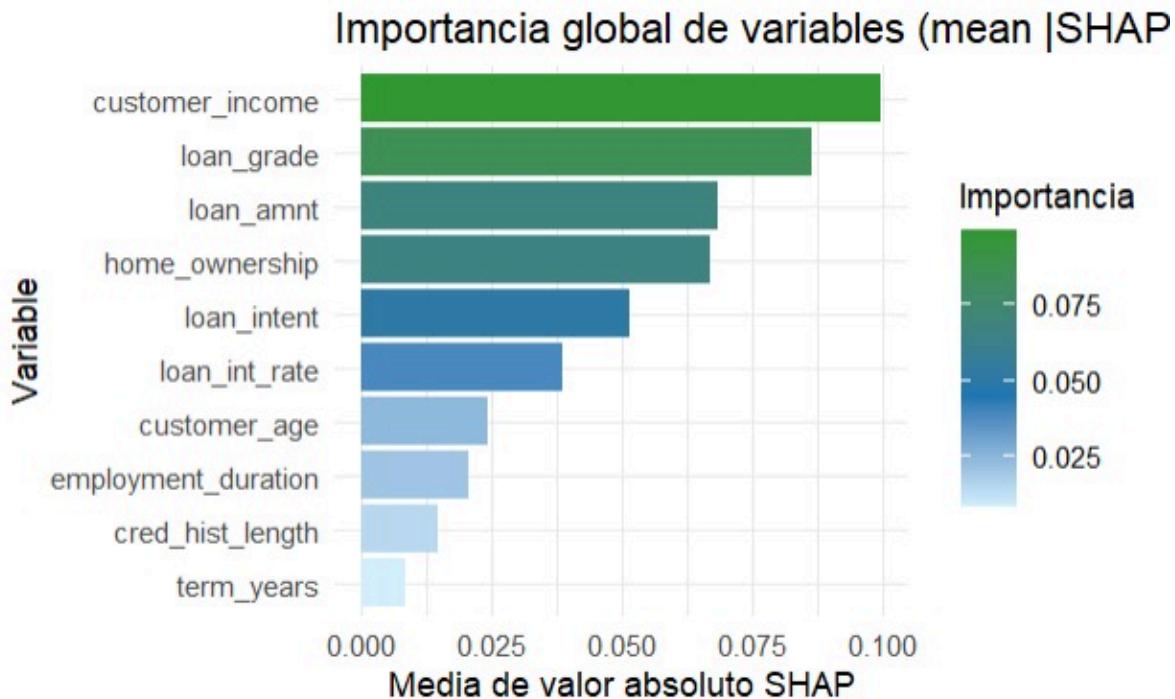
Tabla 6.5.16- Resultados Random Forest Conjunto de prueba sin historical_default

6.5.11. Interpretabilidad del modelo seleccionado sin historical_default

El análisis de interpretabilidad mediante valores SHAP permite identificar con claridad las variables que tienen un mayor impacto en las predicciones del modelo. Según el gráfico de importancia global, las variables con mayor contribución promedio al modelo son

customer_income, loan_grade, loan_amnt y home_ownership lo cual se ve respaldado por los SHAP Summary Plots individuales que se comparten posteriormente.

Gráfico 6.5.4 - SHAP Bar Plot sin historical_default



En el gráfico SHAP ([Gráfico 6.5.5](#)) para variables numéricas, observamos que ingresos más altos tienden a reducir la probabilidad de default (valor negativo), mientras que montos de préstamo más elevados tienen una influencia más mixta pero en general positiva sobre el riesgo de incumplimiento. Esto lo observamos en el [Gráfico 6.5.6](#) donde, al igual que en el EDA, se aplica el percentil 99 para poder visualizar de manera más clara los datos. En este gráfico observamos una gran dispersión de los valores SHAP, lo que nos indica que no podemos asociar una tendencia del monto del préstamo a un valor SHAP únicamente considerando dicha variable. La tasa de interés (loan_int_rate) también tiene un impacto claro: a mayor tasa, mayor contribución al riesgo (SHAP positivo), lo cual es coherente desde una perspectiva financiera.

Gráfico 6.5.5 - SHAP Summary Plot Variables Numéricas sin historical_default

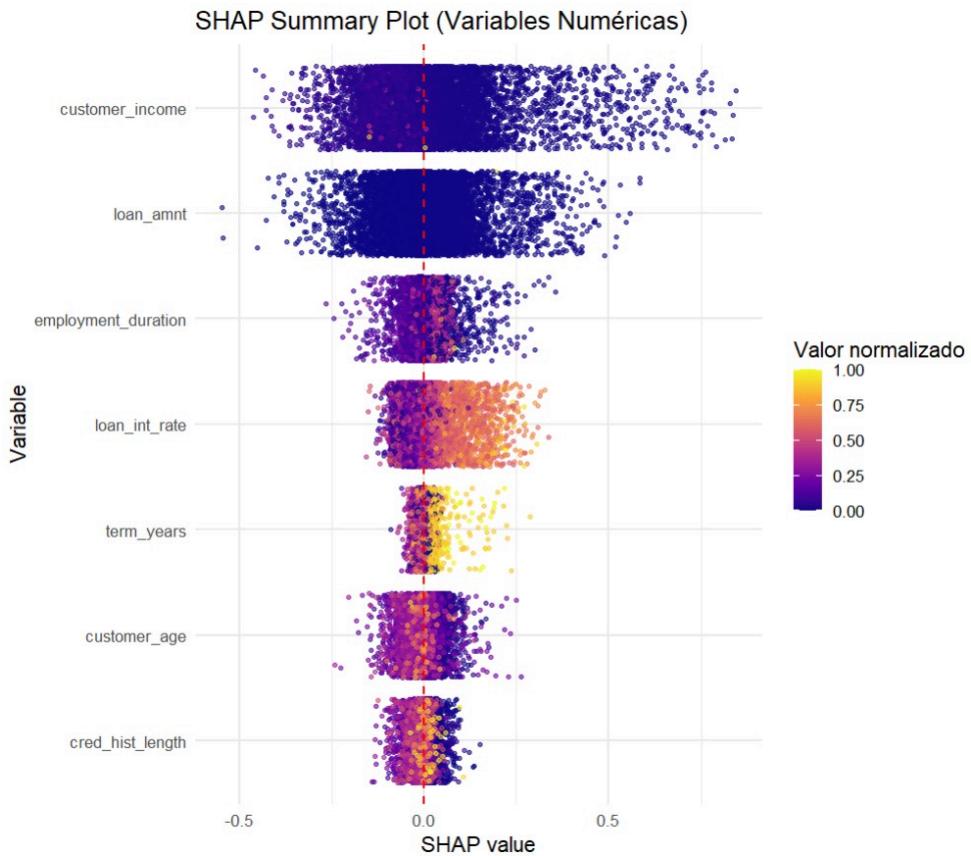
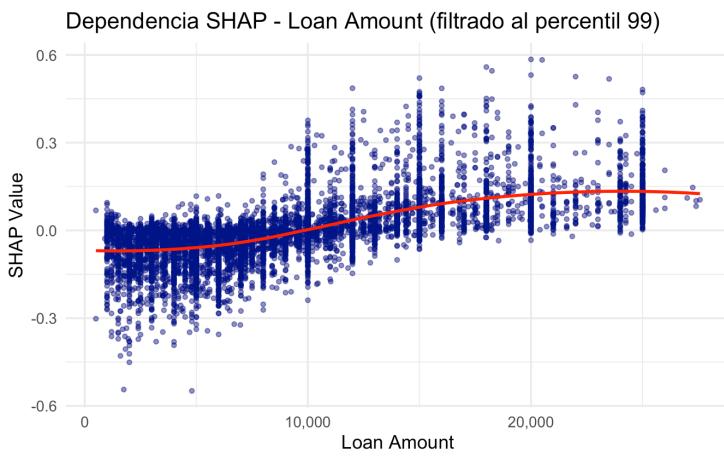


Gráfico 6.5.6 - Dependencia SHAP Loan Amount

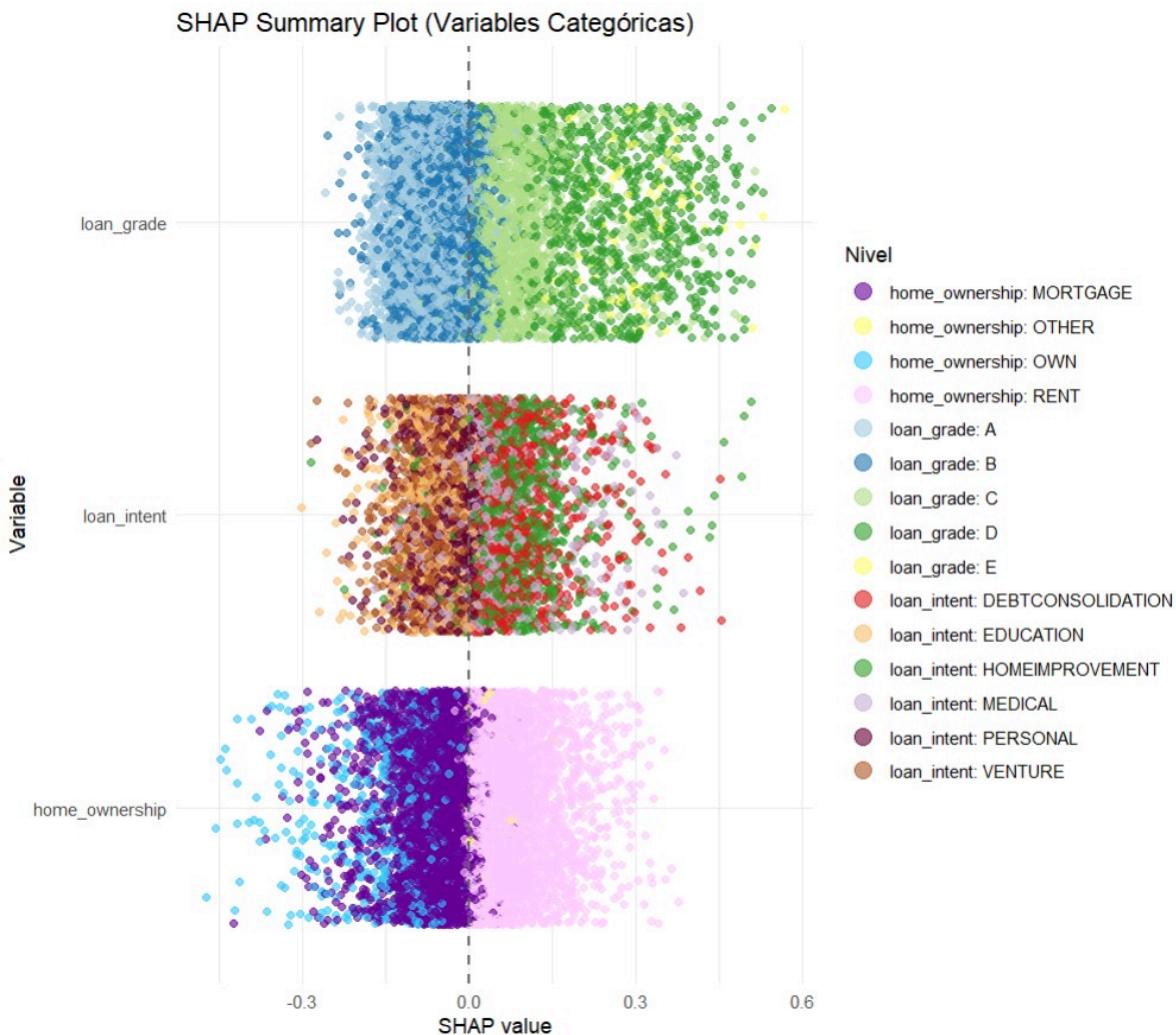


En cuanto a las variables categóricas, el gráfico muestra una clara separación en los efectos de los distintos niveles. Por ejemplo, ciertas categorías dentro de loan_grade (C, D o E) y loan_intent (Debt Consolidation) presentan valores SHAP consistentemente positivos, lo

cual sugiere una asociación con mayor probabilidad de default. En contraste, categorías como Mortgage o Own de la propiedad de vivienda parecen asociarse a menor riesgo, mostrando valores SHAP más cercanos o inferiores a cero.

En conjunto, estos resultados evidencian que, incluso sin la variable historical_default, el modelo es capaz de identificar patrones relevantes y asignar de manera adecuada la importancia a variables relacionadas con el perfil del cliente, el tipo de préstamo y las condiciones del mismo. Además, la estabilidad y sentido económico de las contribuciones SHAP refuerzan la confianza en la capacidad explicativa del modelo ajustado.

Gráfico 6.5.7 - SHAP Summary Plot Variables Categóricas sin historical_default



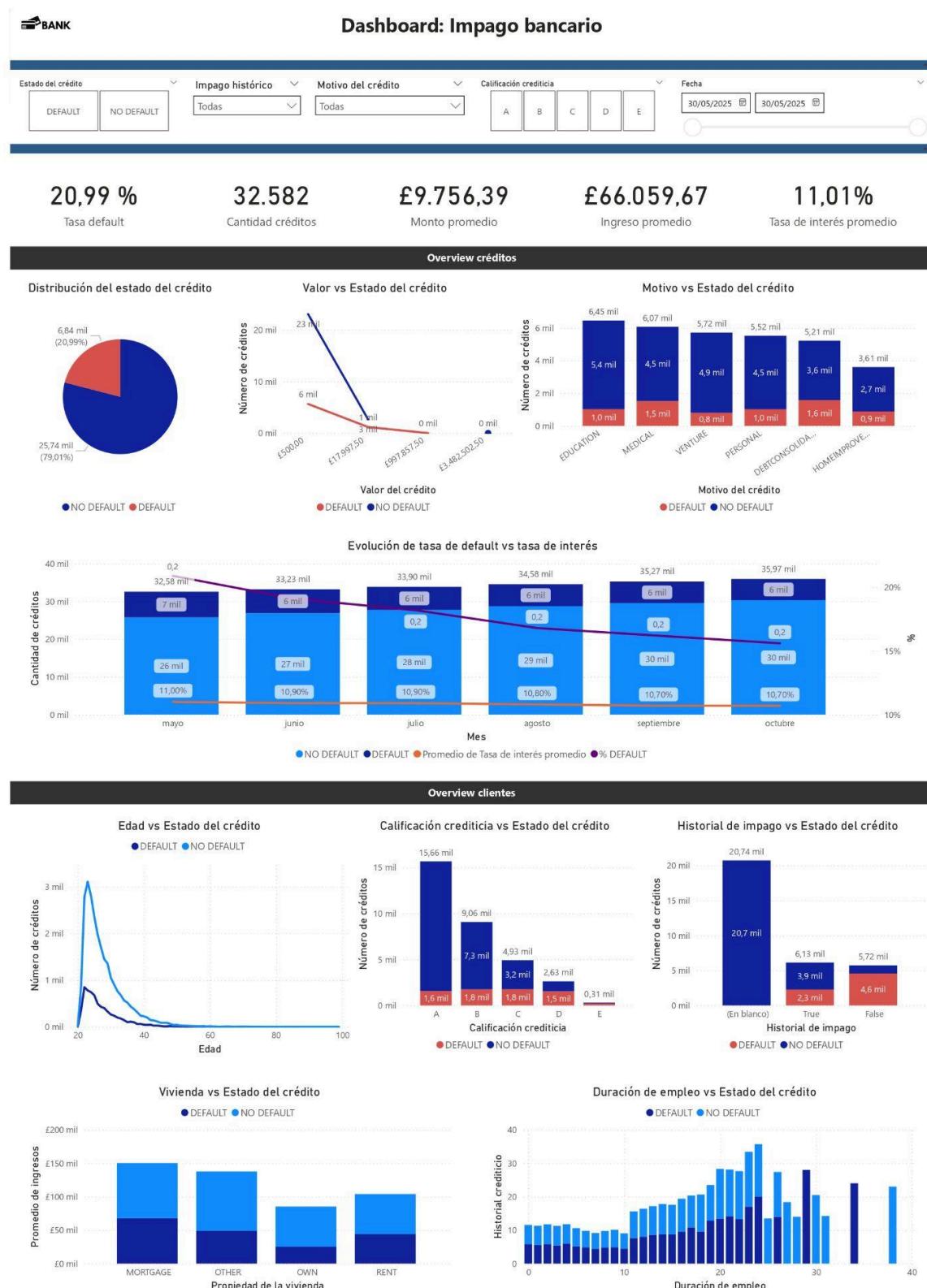
7. MONITOREO Y VISUALIZACIÓN DE DATOS

En el marco de la implementación del modelo predictivo, hemos diseñado dos dashboards complementarios con propósitos estratégicos: uno orientado al negocio y otro dedicado al monitoreo del desempeño del modelo.

Dashboard estratégico (perfil no técnico):

- Visualización de la tasa de default y principales KPIs del portafolio crediticio: cantidad de créditos, monto promedio, ingreso promedio y tasa de interés promedio.
- Distribución del estado del crédito (default/no default) y análisis del valor promedio por estado.
- Segmentación del motivo y la calificación crediticia según el estado del crédito.
- Evolución temporal de la tasa de default y tasa de interés promedio.
- Análisis de clientes según variables clave.
- Gráficos comparativos y series temporales para facilitar la toma de decisiones estratégicas y la identificación de tendencias de riesgo.

Figura 7.1- Dashboard estratégico

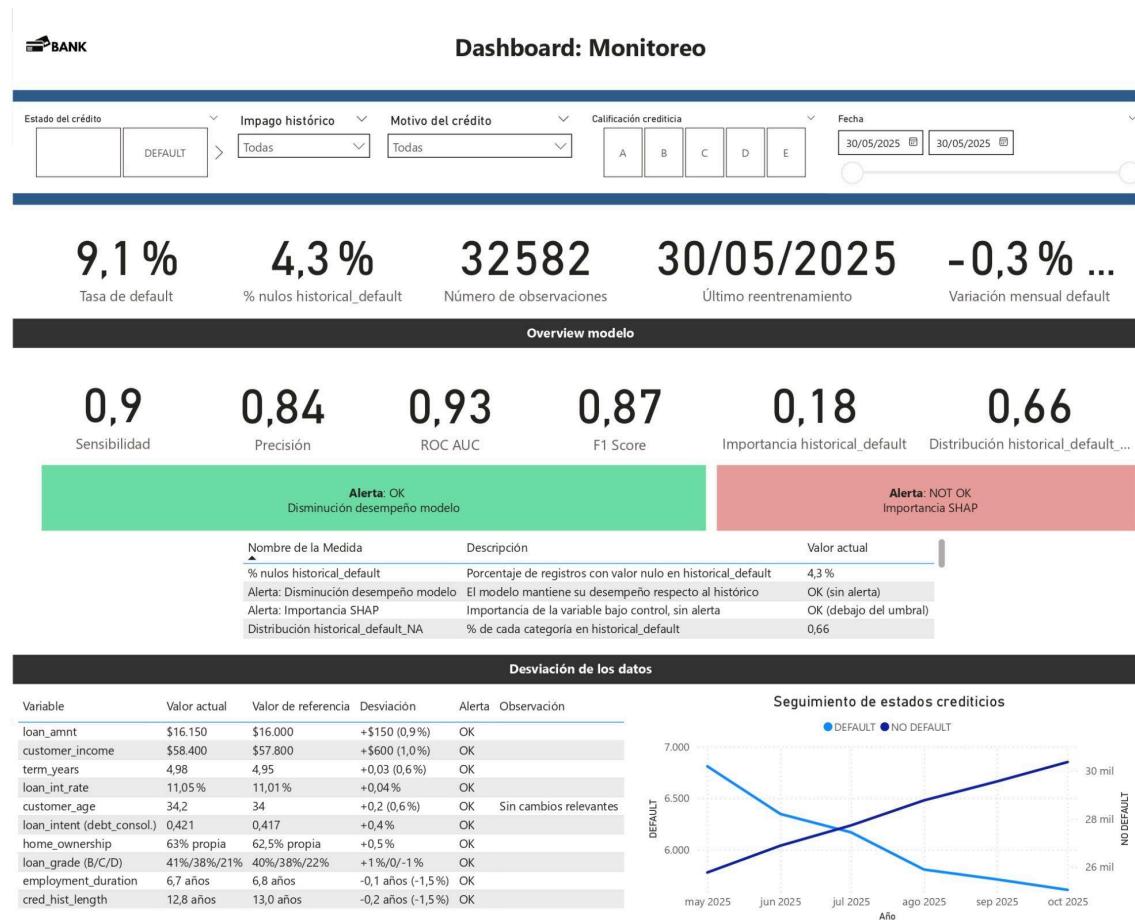


El dashboard de negocio representa una herramienta clave para la gestión operativa y estratégica de la compañía, ya que permite visualizar de forma clara y en tiempo real los indicadores más relevantes para la toma de decisiones. Gracias a su diseño intuitivo y a la selección de KPIs alineados con los objetivos organizacionales, este dashboard facilita el seguimiento de la tasa de default, la distribución de los clientes y la evolución del portafolio crediticio. De esta manera, el área de negocio puede identificar oportunidades de mejora, anticipar tendencias y responder de manera proactiva ante cambios en el entorno o en la calidad de la cartera.

Dashboard técnico (equipo de Data Science):

- Seguimiento de métricas de desempeño del modelo: sensibilidad, precisión, ROC AUC y F1-score.
- Generación de alertas automáticas por degradación del modelo o cambios en la importancia de variables.
- Control de calidad de los datos: porcentaje de nulos y desviaciones respecto a valores de referencia.
- Visualización de la evolución temporal de los estados crediticios.

Figura 7.2- Dashboard técnico



Por su parte, el dashboard de monitoreo del modelo cumple una función fundamental en la gobernanza y sostenibilidad del modelo predictivo. Su principal objetivo es asegurar que el modelo mantenga un desempeño adecuado a lo largo del tiempo, permitiendo detectar posibles desviaciones, alertas o fenómenos de drift que puedan comprometer su efectividad. Además, posibilita un seguimiento continuo de variables críticas, como la integridad y evolución de los datos históricos, garantizando transparencia y trazabilidad en la gestión algorítmica. Esta vigilancia continua refuerza la capacidad de la organización para adaptar el modelo a nuevos contextos, incorporar aprendizajes y mantener la confianza en los resultados generados.

Adicionalmente, la propuesta de diseñar un dashboard específico para el monitoreo de la variable historical_default y el desempeño del modelo evidencia un enfoque responsable y adaptable, esencial en el contexto actual de analítica avanzada y toma de decisiones basadas en datos. Este tipo de herramienta permite observar, de manera continua, cómo evoluciona la proporción de valores nulos en historical_default, identificar posibles cambios en su correlación con el estado de default y alertar sobre eventuales sesgos que puedan surgir con el tiempo. El monitoreo visual y sistemático de esta variable ofrece, además, una valiosa trazabilidad frente a auditorías o requerimientos regulatorios, ya que documenta de forma transparente el uso y la evolución de una variable crítica en el modelo.

Si bien no utilizaremos la variable historical_default en el modelo inicial, realizaremos un seguimiento del comportamiento de la variable en las próximas inyecciones de datos.

En síntesis, el dashboard de monitoreo no solo respalda la transparencia y la responsabilidad, sino que también habilita a la organización a tomar decisiones informadas sobre posibles ajustes, reentrenamientos o futuras inclusiones de variables, de acuerdo con la evidencia recolectada. En definitiva, la existencia de este dashboard de monitoreo no solo facilita el deployment seguro del modelo, sino que garantiza un aprendizaje y adaptación constante ante la evolución de los datos, consolidando así un proceso de mejora continua y alineado con las mejores prácticas del sector. Para ver en detalle los dashboards, consultar el documento adjunto.

8. TRANSPARENCIA Y CUMPLIMIENTO NORMATIVO EN LA TOMA DE DECISIONES AUTOMATIZADAS

Por último, es fundamental realizar una reflexión ética y regulatoria sobre el uso de modelos de machine learning (ML) en la toma de decisiones crediticias. Si bien estos modelos pueden ofrecer un gran poder predictivo, también llevan el riesgo de reproducir o incluso amplificar sesgos presentes en los datos históricos, lo cual podría derivar en decisiones injustas o discriminatorias hacia determinados grupos poblacionales.

Desde una perspectiva regulatoria, el uso de sistemas automatizados en decisiones crediticias debe alinearse con normativas vigentes como el Reglamento General de

Protección de Datos (GDPR). Esta regulación no solo garantiza el derecho a la protección de datos personales, sino que también establece el derecho a recibir explicaciones cuando una decisión ha sido tomada exclusivamente por medios automatizados. En este sentido, la transparencia y la interpretabilidad de los modelos no solo son una buena práctica, sino una obligación ética y legal.

La incorporación de herramientas de interpretabilidad como SHAP es clave para aportar claridad sobre cómo se generan las predicciones del modelo, lo que permite fomentar la confianza en los sistemas automatizados y habilita tanto a usuarios como a instituciones a comprender e, incluso, cuestionar dichas decisiones. Garantizar la equidad, la responsabilidad y la transparencia en el desarrollo y uso de estos modelos es esencial para su implementación ética en el ámbito financiero.

9. CONCLUSIONES

El presente Trabajo de Fin de Máster logra el objetivo de demostrar la utilidad y aplicabilidad de modelos de Machine Learning en la predicción de impago en el sector financiero. A través de la comparación de múltiples algoritmos, tanto lineales como no lineales, se evidenció que, para este dataset, técnicas como Random Forest y SVM superan en precisión a los modelos tradicionales, especialmente cuando se incorpora una rigurosa ingeniería de variables y un tratamiento exhaustivo de los valores nulos.

Entre los hallazgos más relevantes, se destaca que variables como los ingresos, la calificación crediticia y el historial previo de impagos tienen un peso significativo en la probabilidad de default.

En una primera etapa, se construyó un modelo que incluía la totalidad de las variables del dataset. Al aplicar la técnica de interpretabilidad SHAP, se identificó que la variable historical_default destacaba con una relevancia significativamente superior respecto al resto de variables del modelo. Esta situación se corresponde con lo observado durante el análisis exploratorio de datos (EDA), donde detectamos que aproximadamente dos tercios de las observaciones de la variable historical_default presentaban valores nulos. Un análisis más detallado reveló que estos valores nulos no eran aleatorios, sino que corresponden en su

mayoría a casos que no registraban impago en la variable objetivo. Debido a las limitaciones inherentes al dataset, no fue posible determinar con certeza el origen de estos valores nulos. Por ello, y con el fin de evitar posibles sesgos en el modelo, optamos por desarrollar una versión alternativa excluyendo esta variable. Aunque esto implicó una ligera disminución en el desempeño de las métricas, consideramos que representa un enfoque más conservador y robusto, priorizando la transparencia y la integridad del modelo predictivo.

Para continuar, se sugiere realizar el monitoreo de la variable historical_default mediante el dashboard propuesto para analizar si esta variable sigue comportando de la misma manera y posteriormente evaluar la viabilidad del modelo incluyendo dicha variable.

El desarrollo de dashboards interactivos orientados tanto al área técnica como al área de negocio facilitó la traducción de los resultados analíticos en indicadores accionables, agilizando la toma de decisiones y el seguimiento de los riesgos de crédito de manera más proactiva y eficiente. Lo más relevante de la propuesta es implementar un dashboard específico para monitorear la variable historical_default y el desempeño del modelo a lo largo del tiempo. Este enfoque permite:

- Monitorear el comportamiento de los datos, observando si la proporción de valores nulos en historical_default cambia.
- Aportar trazabilidad y transparencia, facilitando la toma de decisiones y mostrando evidencia de un monitoreo consciente y responsable.
- Asegurar la adaptabilidad y gobernanza del modelo, mostrando cómo varía el rendimiento (f1-score, sensibilidad, precisión, ROC AUC) con y sin la variable, y habilitando la implementación de nuevas versiones del modelo según evolucione el entorno.

Entre las principales limitaciones del estudio se encuentra el uso de un dataset público, que, si bien es robusto y representativo, podría no reflejar la totalidad de las causas presentes en un entorno real. Además, la falta de integración de variables externas (por ejemplo, información macroeconómica o comportamental) limita la capacidad predictiva a ciertos perfiles. Futuras líneas de investigación deberían explorar la integración de fuentes de datos alternativas, así como la evaluación del impacto real en los indicadores de negocio.

En definitiva, la modelización predictiva apoyada en el análisis de datos y la inteligencia artificial, representan una herramienta clave para transformar la gestión del riesgo en el sector financiero, contribuyendo a una toma de decisiones más ágil, precisa y fundamentada. Apostar por prácticas de monitoreo y gobernanza, como las aquí planteadas, es fundamental para avanzar hacia una gestión del riesgo más responsable, transparente y adaptativa en el tiempo.

10. BIBLIOGRAFÍA

Adugna, T. D., Ramu, A., & Haldorai, A. (2024). A review of pattern recognition and machine learning. *Journal of Machine and Computing*, , 210–220.

10.53759/7669/jmc202404020

Bishop, C. M. (2006). *Pattern recognition and machine learning* (Springer ed.)

Bussmann, N., Giudici, P., Tanda, A., & Yu, E. P. (2025). Explainable machine learning to predict the cost of capital. *Frontiers in Artificial Intelligence*, volume 8 - 2025

<https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1578190>

Calificación crediticia – qué es y para qué sirve (2015, -12-11). Retrieved Apr 25, 2025, from
<https://economipedia.com/definiciones/calificacion-crediticia-rating.html>

Crédito bancario - concepto y crédito para empresas .

<https://concepto.de/credito-bancario/>

Garantizando la transparencia en modelos de machine learning (2024, -07-10). Retrieved Apr 25, 2025, from
<https://tecnoloworld.net/machine-learning/garantizando-la-transparencia-en-modelos-de-machine-learning/>

Kuhn, M., & Slige, J. (2023). *Tidy modeling with R* (Version 1.0.0 ed.). O'Reilly.

Mancisidor, R. A., Kampffmeyer, M., Aas, K. & Jenssen, R. (2018, June 7). Segment-based credit scoring using latent clusters in the variational autoencoderRetrieved Apr 25, 2025, from <http://arxiv.org/abs/1806.02538>

Mas Elias, J. (2019). Análisis univariante.(PID_00268326)

"Modelos de Machine Learning para la Predicción de Impago en el Sector Financiero" (Meneses Soto, Quintero Estevez, Meiners de Alba)

<https://openaccess.uoc.edu/bitstream/10609/148455/3/AnalisisUnivariante.pdf>

Paccagnini, A., Habib, A. M., Weber, P., Tanda, A., Bussmann, N., Giudici, P., & Yu, P.Explainable machine learning to predict the cost of capital.

Peña, D. (2002). *Ánalisis de datos multivariante*

Q. Xu, Y. Liao, Q. Li, J. Zhang, Z. Song, L. Wang, & X. Yuan. (2024). SHAP-based interpretable models for credit default assessment using machine learning. Paper presented at the 2024 14th International Conference on Software Technology and Engineering (ICSTE), 213–217. 10.1109/ICSTE63875.2024.00044

Tanasuica Zotic, C. (2024). A quantitative analysis of default risk using machine learning and SHAP value interpretation. *Proceedings of the International Conference on Business Excellence*, 18(1), 233–245. 10.2478/picbe-2024-0020

11. ANEXOS

A continuación, se incluye el enlace al repositorio de GitHub que contiene el código desarrollado en R y Python adicional a los dashboards en PowerBI presentados en el presente documento para mayor información.

<https://github.com/anaquintero9/Tfm-credit-default.git>