Cambridge Default Journal (2025), **11–09**, 1–18

Tecnológico de Monterrey

PROYECT REPORT

Modelos de machine learning para la aprobación de préstamo bancario

Victor Adid Salgado Santana A01710023

Tecnologico de Monterrey, Campus Querétaro, México

Abstract

This project develops a machine learning (logistic regression) model implemented from scratch (without the use of external frameworks) and a scikit learn Random Forest Classifier, with the purpose of classifying credit applications as approved or rejected. The model is trained with relevant data from applicants, such as income, credit history, loan amount, previous loan defaults on file and more data. The goal is to show the process of building a classifier, from data preparation, logistic regression programming: cost function (Binary-Cross-Entropy), gradient descent method, to the evaluation of its performance with metrics such as accuracy, F-1 Score, among others. In addition to its practical application in the banking industry, this project offers a better understanding of how the logistic regression works and a comparison with machine learning models from frameworks like scikit learn

Este proyecto desarrolla un modelo de machine learning (regresión logística) implementado desde cero (sin el uso de frameworks externos) y un Random Forest con scikit learn, con el propósito de clasificar solicitudes de préstamo en aprobadas o rechazadas. El modelo se entrena con datos relevante de los solicitantes, como ingresos, historial crediticio, cantidad de préstamo, préstamos incumplidos anteriormente, entre otros. Con ello se busca documentar el proceso de construcción de un clasificador, desde la preparación de los datos, la programación de la regresión logística: la función de costo (Binary-Cross-Entropy), el método de descenso de gradiente, hasta la evaluación de su desempeño con métricas como accuracy, F-1 Score, entre otras. Además de su aplicación práctica en el ámbito bancario, este proyecto ofrece el funcionamiento interno de la regresión logística para una mejor comprensión y su comparación con modelos ya incorporados en frameworks como scikit learn.

Keywords: Machine Learning, Regresión Logística, Aprobación de préstamo, Clasificación

1. Introducción

Las instituciones financieras enfrentan el desafío de evaluar el riesgo crediticio de sus clientes. Prestar dinero implica un nivel de incertidumbre, y es fundamental que los bancos tengan cierta certeza de que los solicitantes podrán cumplir con sus obligaciones de pago. Entre los criterios más comunes para determinar la aprobación de un préstamo se encuentran: una fuente estable y suficiente de ingresos para poder solventar las cuotas mensuales **BBVA**.

El presente proyecto tiene como objetivo diseñar e implementar un modelo de machine learning para la aprobación de préstamos, específicamente mediante regresión logística. Con el objetivo de automatizar parte del proceso de evaluación, de manera que las instituciones financieras puedan ahorrar tiempo y recursos en el análisis de solicitudes. No obstante, se reconoce que la decisión final siempre debe recaer en un humano, ya que los modelos no pueden ser responsables de la decisón final.

Gracias a la disponibilidad de datos históricos que incluyen información de los solicitantes y el resultado de sus solicitudes, es posible entrenar un modelo capaz de distinguir entre perfiles con alta probabilidad de aprobación y aquellos que probablemente sean rechazados.

2. Marco Teórico

En la clasificación de solicitudes de préstamo, es fundamental contar con un modelo que no solo prediga si un préstamo será aprobado o rechazado, sino que también entregue probabilidades asociadas. Para este propósito, se emplea la regresión logística, un modelo supervisado ampliamente utilizado en problemas de clasificación binaria (2 clases). En este contexto, la regresión logística estima la probabilidad de que un solicitante tenga un préstamo aprobado o rehazado dado su perfil.

2.1 Regresión Logística Machine Learning

2.1.1 Binary Cross-Entropy

Para medir el desempeño de la regresión logistica es necesario una función de costo o "pérdida". Para este modelo se usa la Binary Cross-Entropy o también conocida como Log Loss:

$$Logloss(y, p) = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p_i) + (1 - y_i) log(1 - p_i)) \tag{1}$$

Esta función de pérdida calcula que tan bueno o que tan malo es nuestro modelo. Este problema de clasificación binaria (solo 2 clases) préstamo aprobado o no aprobado tiene 2 posibles valores para la y, la cuál es el valor real: 0 = préstamo No Aprobado,1 = préstamo Aprobado

Cómo este modelo trabaja con probabilidades gracias a la sigmoide, los valores de p están acotados a $p = [0 - 1]$ pues son la probabilidad de que un solicitante tenga un préstamo aprobado o no.

Gracias a los logaritmos, si la probabilidad coincide o se acerca bastante a la real, el error va a tender a 0, pero si la realidad dista completamente de lo predicho, $y = 1, p = 0$. Entonces el error tiende a infinito, pues es como si fuera improbable según nuestro modelo cuando es completamente lo opuesto. Por esa razón en la práctica se le suma un valor muy pequeño $\epsilon \approx 0$ para que en estos tipos de casos (o la situación opuesta) no nos arroje algún error el modelo.

2.1.2 Función Sigmoide

La función sigmoide transforma números continuos reales a valores entre $(0,1)$ **Chen2024**

$$sigmoide(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

Esto es de suma importancia para nuestro modelo de regresión logística, ya que transforma nuestros cálculos en probabilidades.

2.1.3 Función de Hipótesis

La regresión Logística como modelo de machine learning busca optimizar los pesos w y los bias b, los cuáles se pueden apreciar en la función de hipótesis, la cuál es una ecuación lineal metida en la función de sigmoide, lo que termina en el cálculo de una probabilidad.

$$hipotesis(h) = \frac{1}{1 + e^{-h}} \tag{3}$$

Dónde:

$$h = w_1 \cdot x_1 + ... + w_n \cdot x_n + b \tag{4}$$

2.2 Gradiente Descendente

Para la optimización de dichos pesos de la función de hipótesis, se utiliza el método de gradiente descendente. El cuál con el cálculo del gradiente de la función de costo busca llegar a la dirección en la que determinados pesos y bias producen el menor error, es decir minimizar la función de costo $J(w, b)$. Con varias iteraciones y con ayuda de un Learning Rate α (dice que tanto se avanza), gradiente descendente es utilizado para moverse en dirección del mínimo, con el fin w y b sean óptimos locales idealmente.

$$w = w - \alpha \frac{\partial J}{\partial w} \tag{5}$$

$$b = b - \alpha \frac{\partial J}{\partial b} \tag{6}$$

2.3 Estandarización por Z-score:

Para procesar los datos antes de ingresarlos al modelo de regresión logística, la estandarización es un proceso importante. Esta transformación centra los datos con media 0 y desviaciones estándar de 1. La estandarización asegura que todas las variables numéricas estén en la misma escala. Esto es importante porque en algoritmos basados en gradiente, como la regresión logística, variables con escalas muy distintas pueden sesgar el proceso de optimización. Ahora los datos dirán que tan cerca o que tan lejos estás de la media $\mu = 0$ en términos de desviaciones estándar (σ)

$$z = \frac{x - \mu}{\sigma} \tag{7}$$

2.4 Winsorización

Para reducir el impacto de valores extremos en una distribución, la winsorización convierte dichos valores en otros valores no tan atípicos. Hay distintas formas de calcular el reemplazo o valor a cambiar. En este proyecto se utilizan límites gaussianos, teniendo así:
$right\_tail = \mu + 3\sigma$
$left\_tail = \mu - 3\sigma$
Esto significa que los valores a más de 3 distribuciones estándar van a ser reemplazados por los límites de las colas, resultando en una distribución entre $[-3\sigma, +3\sigma]$

2.5 SMOTE

Esta es una técnica de sobremuestreo de la clase minoritaria generando muestras sintéticas para el desbalanceo de datos. Para esta técnica, se seleccionan k vecinos, aleatoriamente se elige uno de estos vecinos (X_k), se calcula la diferencia entre uno de los vecinos y una muestra de la clase minoritaria (X). Por medio de una variable aleatoria w entre$[0, 1]$, las muestras sintéticas se calculan de la forma:

$$Z = X_0 + w(X_k - X_0)$$

2.6 Oversampling

Para lidiar con desbalanceo es común recurrir a la técnica de oversampling, la cuál consiste en aumentar el número de registros (observaciones) de la clase minoritaria de modo que igualen a la clase mayoritaria.

2.7 Pruebas de chi-cuadrada

La prueba de chi-cuadrada de independencia se utiliza para analizar si dos variables categóricas están relacionadas o si son independientes.

Se parte de una **tabla de contingencia** con frecuencias observadas O_{ij}, donde:

- i recorre las filas (categorías de la primera variable).
- j recorre las columnas (categorías de la segunda variable).

Formulando las hipótesis:

- H_0 : Las variables son independientes.
- H_1 : Las variables tienen una relación (dependencia).

El estadístico de chi cuadrada se calcula como:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Donde:

- O_{ij} = frecuencia observada en la celda (i, j).
- E_{ij} = frecuencia esperada bajo H_0, dada por:

$$E_{ij} = \frac{(\text{total fila } i)(\text{total columna } j)}{N}$$

con N = total de observaciones.

El valor p se obtiene comparando el estadístico χ^2 con la distribución χ^2:

$$p = P(\chi^2 \geq \chi^2_{observado})$$

Si $p < \alpha$ (normalmente $\alpha = 0.05$), se rechaza H_0 con un $1 - \alpha$ de nivel de significancia y se concluye que las variables están relacionadas.

2.8 Encoding

Para la integración de variables categóricas al dataset, se realiza un "one-hot-encoding", el cuál para las diferentes categorías de una columna se convierten en columnas, se coloca 0 si no pertenecen y 1 si es la categoría correspondiente, se pasa de long a wide format. De esta manera se logran "codificar" en 0's y 1's las distintas opciones de una columna categórica

2.9 Random Forest

Los Random Forest son un algoritmo de machine learning que combinan el resultado de múltiples árboles de decisión para llegar a un resultado. Funciona para problemas de clasificación y regresión. **IBM.**

3. Exploración y limpieza de los datos

El dataset de este proyecto se obtuvo de: **Kaggle**, contiene 45,000 registros y 14 columnas las cuáles se listan a continuación:

3.1 Limpieza de los datos

El dataset no contiene datos nulos, por lo que no se realizo algún método para lidiar con datos faltantes, más adelante se habla acerca del tratamiento de outliers.

Table 1. Variables del dataset y descripción.

Columna	Descripción	Tipo
person age	Edad de la persona	Float
person gender	Género de la persona	Categórica
person education	Nivel de educación	Categórica
person income	Ingreso anual	Float
person emp exp	Años de experiencia laboral	Integer
person home ownership	Estatus dueño de una casa	Categórica
loan amnt	Monto del préstamo solicitado	Float
loan intent	Propósito del préstamo	Categórica
loan int rate	Tasa de interés del préstamo	Float
loan percent income	Préstamo como porcentaje del ingreso	Float
cb person cred hist length	Historial crediticio en años	Float
credit score	Credit score de la persona	Integer
previous loan defaults on file	Préstamos incumplidos anteriormente	Categórica
loan status	Préstamo aprobado (1=aprobado, 0=rechazado)	Integer

3.2 Exploración de los datos

A continuación se muestran gráficas que muestran la distribución de las variables numéricas del dataset por tipo de solicitud.

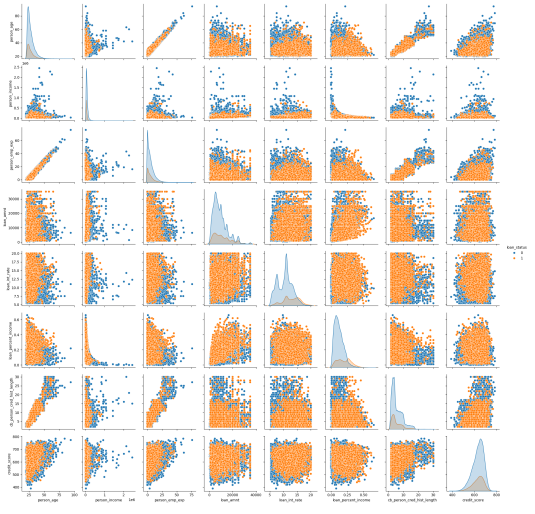

Figure 1. Pair Plot de las variables numéricas por estatus de préstamo

En primer instancia, estas gráficas nos dan un indicio de un desbalanceo de datos, dónde posiblemente hay un mayor número de solicitudes rechazadas, debido a que las distribuciones de frecuencia presentan valores más altos en comparación a las solicitudes rechazadas. Se observan posiblen relaciones lineales entre:

- Edad (person age), años de experiencia laboral (person emp exp), años de historial crediticio(cb person cred hist length)

Para las primeras 3 variables, edad, ingreso y experiencia laboral se observa un comportamiento altamente sesgado a la derecha, de aquí se infiere que la mayoría de los solicitantes son relativamente jóvenes y hay pocos solicitantes con una edad muy avanzada. Para la variable de credit score se observa un comportamiento de sesgo a la izquierda. Para las variables de tasa de interés del préstamo y porcentaje del préstamo en ingreso, se puede observar como la distribución de los solicitantes con préstamos aprobados y rechazados cambia un poco, para estas variables la distribución de aprobados ya no está completamente contenida en la distribución de rechazados.

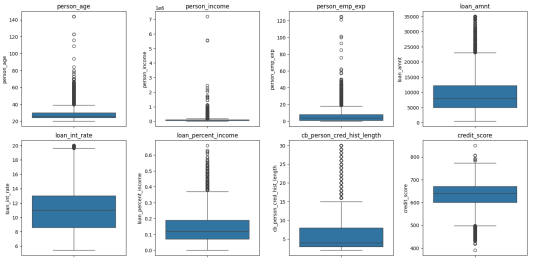

Figure 2. Boxplot de las variables numéricas

Con los diagrama de caja se observa la distribución de las variables, donde se puede apreciar el rango intercuartílico, la mediana y es sencillo detectar la presencia de datos extremos. Como lo es en el caso de la edad, dónde se ven edades inusuales de 120-140 años, valores atípicos muy distantes en los ingresos, personas con experiencia laboral de más de 100 años; para las demás variables se puede observar un comportamiento en el que hay datos con valores muy elevados,es interesante notar como valores de 0 en el loan_percent_income están dentro del rango intercuartílico, si bien, no se muestran como outliers, resulta extraño que el préstamo como porcentaje del ingreso sea 0. Por último para credit score hay outliers con solicitantes con un puntaje muy por debajo del usual y solicitantes con un score muy por encima de lo típico.

A continuación se muestran los histogramas de frecuencia de las variables categóricas por estatus de la solicitud.

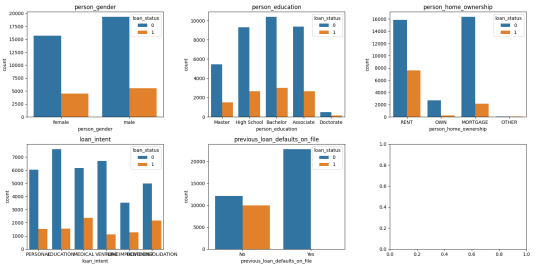

Figure 3. Histograma de frecuencias de las variables categóricas por estatus

Se observa que la distribución para género y educación es similar para créditos aprobados y rechazados, para la variable "previous loan defaults on file" distingue claramente a las solicitudes aprobadas de las rechazadas, esto significa que los préstamos incumplidos anteriormente son un factor determinante para rechazar las solicitudes

4. Preparación de los datos

Para el preprocesamiento y preparación de los datos, se seleccionan todas las variables. Como se implementa una regresión logística, se les aplica un **one hot encoding** a las variables categóricas para poderlas incluir en el modelo (anteriormente solo se consideraban variables numéricas). Es importante analizar las variables que se incluyen en el modelo final debido a que se deben de seguir los principios de IA responsable de equidad e inclusión. **MSFT**
El dataset numérico está compuesto por: 'person age', 'person income', 'person emp exp', 'loan amnt','loan int rate', 'loan percent income', 'cb person cred hist length', 'credit score', 'loan status'. Mientras que las variables categóricas están dadas por: 'person gender', 'person education', 'person home ownership','loan intent', 'previous loan defaults on file'.

4.1 Análisis de correlaciones

Para nuestras variables de tipo numérico, se hace un análisis de correlación con el objetivo de eliminar variables altamente correlacionadas $r > 0.8$.

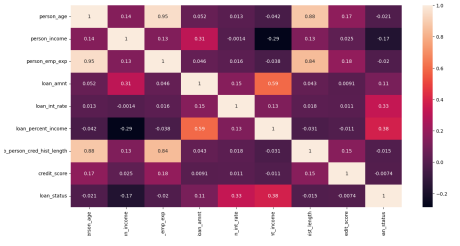

Figure 4. Mapa de correlaciones de las variables numéricas

Con el análisis se puede observar que 'person emp exp' y 'cb person cred hist length' esta muy relacionado con la 'person age' $r > 0.8$. Esto es lógico, debido a que típicamente, las personas con mayor edad son las que tienen más años de experiencia laboral y más años de historial crediticio.

4.2 Análisis de independencia de las variables categóricas

Table 2. Prueba Chi-cuadrada entre variables categóricas y *loan_status*

Variable	Categoría	0	1	Conclusión
2***person_gender**	Female	15674	4485	2*p = 0.914 : No relación significativa
	Male	19326	5515	
5***person_education**	Associate	9378	2650	5*p = 0.733 : No relación significativa
	Bachelor	10381	3018	
	Doctorate	479	142	
	High School	9301	2671	
	Master	5461	1519	
4***person_home_ownership**	MORTGAGE	16345	2144	4*p = 0.000 : Relación significativa
	OTHER	78	39	
	OWN	2729	222	
	RENT	15848	7595	
6***loan_intent**	DEBTCONSOLIDATION	4982	2163	6*p ≈ 2.17e-194 : Relación significativa
	EDUCATION	7601	1552	
	HOMEIMPROVEMENT	3525	1258	
	MEDICAL	6170	2378	
	PERSONAL	6031	1521	
	VENTURE	6691	1128	
2***previous_loan_defaults_on_file**	No	12142	10000	2*p = 0.000 : Relación significativa
	Yes	22858	0	

Con lo observado podemos inferir que las variables "person home ownership", "previous loan defaults on file" y "loan intent" tienen una relación significativa con loan-status, pues tienen un p-value < 0.05

4.3 Datos atípicos

Cómo criterio para tratar a los datos atípicos se eliminarán los registros con valores fuera del comportamiento natural de cada variable.

- **Solicitudes con personas de más de 100 años**
- **Solicitudes con personas con más de 80 años de experiencia laboral**
- **Solicitudes con ingresos elevados desproporcionables**
- **Solicitudes con un préstamo que les represente un 0% de su ingreso**

Se eliminaron registros con edades mayores a 100 años y experiencias laborales mayores a 80 años, al considerarse poco realistas y posiblemente atribuibles a errores de captura. De igual manera, se eliminaron ingresos extremadamente altos detectados como valores atípicos y aislados ($> 5,000,000$) fuera de rango según boxplots. También, se eliminaron los registros que contaban con un préstamo solicitado que equivalía al 0% de sus ingresos. Por último, se eliminaron los registros que tenían un

valor de 'OTHER' en la variable person home ownership debido a la escasa cantidad de registros con esa categoría.

4.4 Selección de variables

Con el análisis de correlaciones se opta por eliminar las variables 'person emp exp' y 'cb person 'cred hist length', debido a que están altamente correlacionadas con 'person age'. Por otro lado, por el análisis de las variables categóricas, se opta por eliminar las variables 'person gender' y 'person education', debido a que las distribuciones eran muy similares entre las clases y no tenían una relación significativa (con la variable loan status). Con esta selección de variables y el tratamiento de datos atípicos, se obtiene un dataset con 44969 registros y 10 variables: 'person age', 'person income', 'person home ownership', 'loan amnt','loan intent', 'loan int rate', 'loan percent income', 'credit score','previous loan defaults on file', 'loan status'. Cómo algunas de estas variables son de tipo categóricas, al momento de hacer el one-hot-encoding el dataset resulta en 19 columnas y los mismos 44969 registros. Con el objetivo de eliminar una posible multicolinealidad con las variables categóricas al momento de estar encoded y tener una base de referencia, se eliminan las columnas: 'previous loan defaults on file Yes', 'person home ownership OTHER' y 'loan intent PERSONAL'.

Con esta selección de variables se utilizan las siguientes 15 columnas para hacer las clasificaciones: 'person age', 'person income', 'loan amnt', 'loan int rate','loan percent income', 'credit score', 'person home ownership MORTGAGE', 'person home ownership OWN', 'person home ownership RENT','loan intent DEBTCONSOLIDATION', 'loan intent EDUCATION','loan intent HOMEIMPROVEMENT', 'loan intent MEDICAL','loan intent VENTURE','previous loan defaults on file No'.
La variable a clasificar corresponde a: 'loan status'

4.5 Estandarización

Por último se procede a estandarizar los datos, la media y desviación estándar de cada columna es almacenada en dado caso que se quieran clasificar nuevos solicitantes.

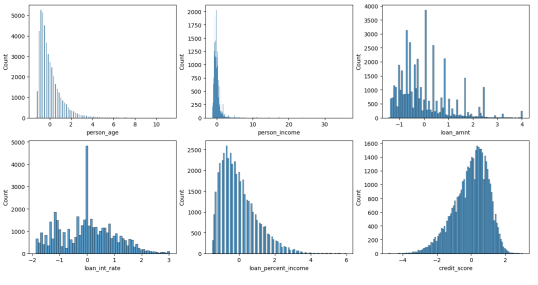

Figure 5. Gráfica de las variables numéricas estandarizadas

4.6 Winsorización

Con las variables numéricas estandarizadas es sencillo notar la gran varianza que hay en algunas variables, para lidiar con los efectos de valores extremos se procede a realizar una winsorización, la cuál acota las distribuciones entre $[3\sigma, +3\sigma]$ reemplazando valores que sobresalen de estos límites a los acotados previamente.

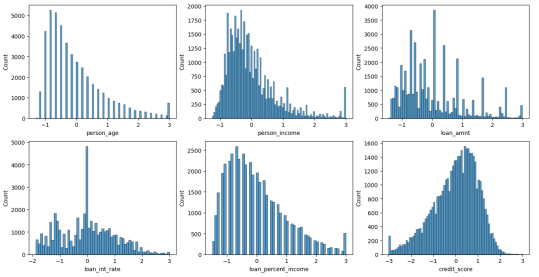

Figure 6. Gráfica de las variables numéricas winsorizadas

4.7 Desbalanceo de clases

Se analiza la cantidad de datos que cuentan con un préstamo aprobado y con un préstamo rechazado.

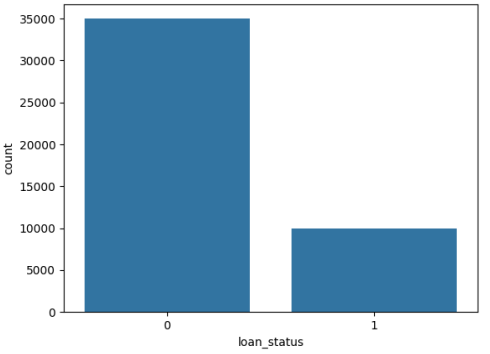

Figure 7. Conteo de registros por clase

Cómo se observa, de los 44,969 registros, 34,970 de ellos pertenecen a personas con préstamos rechazados y el resto de los registros a solicitantes con préstamos aprobados. Esto significa que el 77% de los datos pertenece a la clase 0 y sólo el 22% a la clase 1.
Si se busca tener un modelo balanceado, con un desempeño aceptable en la clasificación de ambas clases, es necesario aplicar alguna técnica de balanceo de datos. Para este caso, se emplean técnicas de oversampling para mantener los datos de la clase mayoritaria al aumentar las muestras de la clase menos representada. En este proyecto se hace la comparación de los modelos sin ninguna técnica de desbalanceo, aplicando SMOTE y aplicando un sobremuestreo aleatorio de la clase minoritaria (préstamos aceptados), con las técnicas de sobremuestreo ambas producen un total de 69940 registros, SMOTE con datos sintéticos y el otro volviendo a tomar registros de las solicitudes aprobadas.

5. Construcción del modelo

Dado a que la implementación de la regresión logística es desde 0 y sin ayuda de frameworks se programan las siguientes funciones:

- **sigmoid(x):** Regresa las probabilidades dada una x que corresponde a la función de hipótesis
- **log_loss(y, y_predicted):** Cálcula el costo de la solución programando la binary cross entrophy
- **logistic_regression(X, y, learning_rate, epochs):** En esta función se hace la implementación de la regresión logística, usando las funciones anteriores y computando el cálculo del gradiente

y la implementación del gradiente descendente para la optimización de los pesos, esta función devuelve los pesos w, el bias b y un arreglo con el historial del costo de la solución *cost_history*

- Adicionalemente se programan las funciones para estandarizar los datos, realizar el split de datos de entrenamiento, validación y testeo, realizar SMOTE u random oversampling y funciones para evaluar el desempeño del modelo.

5.1 Modelo con framework

Se opta por un modelo de random forest para clasificación como método a implementar por medio de la librería sci-kit learn.
Hiperparámetros del modelo Random Forest: 'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'n_estimators': 100, 'n_jobs': None, 'oob_score': False, 'random_state': 42, 'verbose': 0, 'warm_start': False

5.2 Split de los datos

Para este proyecto se había establecido un split de train y test únicamente, ahora se incorpora un set de validación. Lo que resulta en proporciones de de 80% datos de entrenamiento, 10% datos de validación y 10% datos de testeo. Se resamplean aleatoriamente los registros antes de hacer el split.

6. Resultados

En esta sección se presentan los resultados obtenidos tanto para el modelo desde 0 como para el modelo con librerías.

6.1 Modelo sin frameworks

Debido a los distintos experimentos en la ejecución del modelo, se llega a los siguientes hiperparámetros con buena convergencia en pocas épocas para el modelo sin frameworks.
Al correr dicho modelo para nuestros datos de entrenamiento (con desbalance) con un learning rate $\alpha = 0.1$ y *epochs* = 200. Se obtienen los siguientes pesos:

Table 3. Pesos (w) y bias (b) del modelo

Parámetro	Valor
w 1	-0.009
w 2	-0.319
w 3	-0.097
w 4	0.696
w 5	0.715
w 6	-0.062
w 7	-0.767
w 8	-0.318
w 9	-0.211
w 10	-0.003
w 11	-0.374
w 12	-0.066
w 13	-0.113
w 14	-0.403
w 15	0.861
b	-1.3

Al analizar los coeficientes se nota como el modelo de regresión logística asigna una mayor importancia a las variables 4, 5,7 y 15 que corresponden a: la tasa de interés del préstamo, el porcentaje del préstamo de acuerdo al ingreso de la persona, si la persona tiene una hipoteca y si la persona no tiene un historial de préstamos incumplidos . Esto debido a que $|w4|, |w5|, |w7|, |w15|$ los valores absolutos de los pesos de esas variables son lo mayores respecto a los demás parámetros. Esta información está alineada con lo investigado anteriormente acerca de los criterios de aprobación o rechazo de los préstamos, pues al igual que las instituciones bancarias, nuestro modelo se apoya en el porcentaje que le representaría al solicitante adquirir ese préstamo de acuerdo a su ingreso, la tasa de interés del préstamo, si la persona no tiene un historial de préstamos incumplidos la probabilidad de ser aceptado aumenta, mientras si la persona tiene una hipoteca disminuye la probabilidad de aceptación.

6.1.1 Evaluación del modelo

A continuación se muestra el reporte de clasificación del modelo para el train, val y test de los datos, comparando los distintos escenarios: sin balanceo de datos, con SMOTE y con un sobremuestro aleatorio.

Table 4. Resultados de clasificación por técnica y conjunto de datos

Técnica	Clase	Precisión	Recall	F1-score
		Sin balanceo		
Train	0	0.89	0.96	0.92
	1	0.80	0.57	0.67
	Accuracy		0.87	
Val	0	0.89	0.97	0.92
	1	0.83	0.56	0.67
	Accuracy		0.88	
Test	0	0.88	0.96	0.92
	1	0.80	0.55	0.65
	Accuracy		0.87	
		SMOTE		
Train	0	0.88	0.81	0.84
	1	0.82	0.89	0.86
	Accuracy		0.85	
Val	0	0.88	0.81	0.84
	1	0.82	0.89	0.85
	Accuracy		0.85	
Test	0	0.89	0.80	0.84
	1	0.81	0.90	0.85
	Accuracy		0.85	
		Oversampling		
Train	0	0.87	0.80	0.84
	1	0.82	0.88	0.85
	Accuracy		0.85	
Val	0	0.88	0.80	0.84
	1	0.82	0.89	0.86
	Accuracy		0.85	
Test	0	0.88	0.80	0.83
	1	0.80	0.88	0.84
	Accuracy		0.84	

Como se observa en la tabla de resultados, la regresión logística con clases desbalanceadas funciona muy bien para la clase mayoritaria y de forma consistente, pues sus métricas no varían entre los distintos splits (train, val y test). No obstante, este modelo no logra clasificar correctamente la clase minoritaria, lo que indicaría un alto sesgo para esta clase, aunque tiene la accuracy más alta de los 3 escenarios este modelo solo se enfoca en clasificar correctamente a los rechazados, se observa con un recall < 60 para todos los sets que el modelo tiene una alta cantidad de falsos negativos, lo que indicaría que nuestras inferencias son correctas, no puede distinguir correctamente a la clase minoritaria.
Por parte de los datos con SMOTE y Oversampling se observan métricas consistentes (hay una varianza baja) tanto en el train, validation y test, esto significa que no hay un overfitting por parte del modelo, si bien disminuye un poco su accuracy se observa como las evaluaciones para ambas clases son más cercanas, lo que indica un equilibrio entre ambas clases, el modelo tiene un buen desempeño identificando ambas clases (no hay un underfitting), particularmente SMOTE cuenta con métricas ligeramente mejores al sobremuestreo aleatorio.

Se puede ver el desempeño del modelo a través de la matriz de confusión para los datos de testeo obtenidos por la técnica de SMOTE.

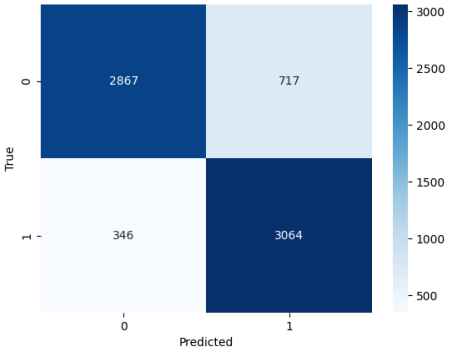

Figure 8. Matriz de confusión SMOTE para testing

Se observa un desempeño general (accuracy) del 85%, apoyados de las tablas mostradas anteriormente y la matriz de confusión, se observa como el modelo tiende a clasificar más solicitudes aprobadas cuando en realidad son rechazadas (Falsos positivos), no obstante logra detectar y clasificar correctamente el 80% de los casos cuándo se tratan de solicitudes rechazadas. Cómo el desempeño no varía drásticamente entre los distintos sets, quiere decir que no hay un overfitting, pues el modelo generaliza satisfactoriamente los sets de validación y de testeo, no obstante, hay un sesgo ligeramente mayor para las solicitudes aprobadas, pues se observa como se clasifican más solicitudes de este tipo, existiendo casi el doble de falsos positivos que de falsos negativos, a pesar de ello, comparado el f1-score de cada clase se tiene una evaluación ≈ 0.84 lo que indica que el modelo generaliza aceptablemente bien.

6.2 Modelo con frameworks

Debido al buen desempeño del modelo desde 0 con SMOTE, se implementa el RandomForest para el dataset generado con esta técnica, para poder comparar el mejor modelo a mano contra un modelo de librerías en las mismas condiciones.
En la siguiente gráfica se observan los pesos asignados a cada variable

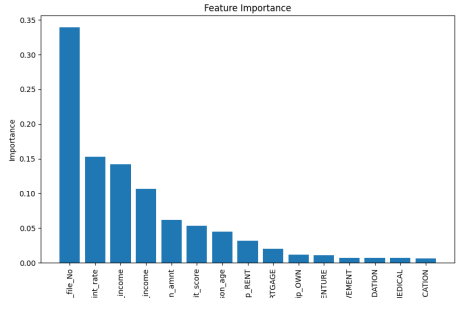

Figure 9. Importancia de las variables en el Random Forest

Similar a la regresión logística desde 0, el Random Forest se apoya principalmente en el porcentaje

que le representaría al solicitante adquirir ese préstamo de acuerdo a su ingreso, la tasa de interés del préstamo, la cantidad de préstamo solicitado y el historial de créditos incumplidos anteriormente. Igualmente, estos resultados son coherentes con las decisiones que toman los bancos para aprobar o rechazar préstamos.

6.2.1 Evaluación del modelo

A continuación se muestra el reporte de clasificación del modelo:

Table 5. Reporte de clasificación para Train, Val y Test

Conjunto / Clase	Precisión	Recall	F1-score
Train			
0	1.00	1.00	1.00
1	1.00	1.00	1.00
Accuracy		1.00	
Val			
0	0.96	0.94	0.95
1	0.94	0.96	0.95
Accuracy		0.95	
Test			
0	0.97	0.93	0.95
1	0.93	0.97	0.95
Accuracy		0.95	

Si bien el modelo fue perfecto en el entrenamiento, esto no supone un overfitting fuerte obligatoriamente, ya que al ver los resultados para los sets de validación y testeo el desempeño es similar entre ellos (indicando una leve varianza), ya que las métricas disminuyeron alrededor de un %5 comparado con el set de entrenamiento. Cómo el desempeño es bueno y el f1-score es igual para ambas clases (0.95) se deduce que el modelo tiene un buen balance y generaliza adecuadamente ambas categorías, indicando un sesgo bajo. Pueden existir indicios de overfitting, pero como las métricas para los sets de validación y testeo se mantienen similares y con un desempeño general alto del 95%, el modelo tiene una buena solidez.

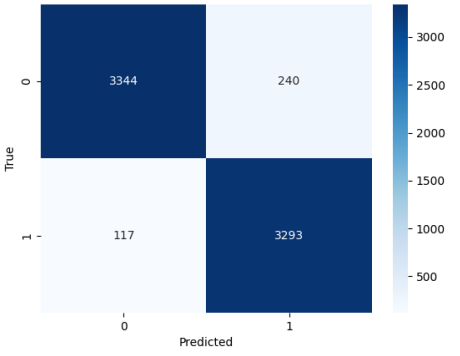

Figure 10. Matriz de confusión de los datos de testeo

Con la matriz de confusión del Random Forest para los datos de testeo se observa el buen desempeño para hacer las clasificaciones, la cantidad de falsos positivos y falsos negativos es mínima, no obstante, se sigue observando que hay una mayor cantidad de falsos positivos (cerca del doble de falsos negativos). Pero de forma general el desempeño es sobresaliente, comparado con los modelos sin framework mejora las clasificaciones aproximadamente en un +10%

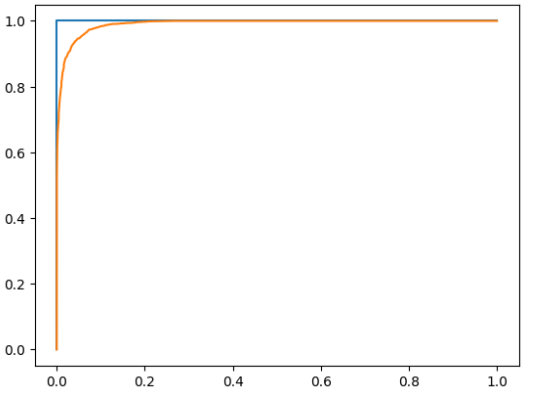

Figure 11. Curva ROC del modelo de Random Forest

En la curva ROC se observa como el clasificador perfecto estaría en la esquina superior a la izquierda, para el modelo generado con Random Forest se observa una curva muy cercana a esta esquina. Si bien el modelo no es un clasificador perfecto, si se acerca mucho a él.

6.3 Comparación de modelos

Table 6. Comparación de Modelos de Regresión Logística con y sin Undersampling

Modelo	Learning rate	Épocas	Tiempo (s)	Accuracy	Precision	Recall	F1 Score
Regresión Logística SMOTE	0.1	200	50.33	0.85	0.85	0.85	0.845
Random Forest SMOTE	-	-	6.63	0.95	0.95	0.95	0.95

6.4 Hardware

Dispositivo empleado para la ejecución de los modelos

Table 7. Especificaciones del sistema utilizado en los experimentos

Característica	Especificación
Procesador	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (2.21 GHz)
RAM	12.0 GB

6.5 Conclusiones

En este proyecto se construyó una regresión logística desde 0 (sin uso de frameworks) y se implementó un Random Forest con scikit learn como modelos de machine learning para clasificar las solicitudes de préstamos en aprobado y rechazado. A lo largo de este proyecto se hicieron tareas de extracción de datos, exploración, limpieza, tratamiento de outliers, transformaciones y desbalanceo de clases. Con el paso del tiempo se fueron incorporando más técnicas para mejorar el desempeño de los modelos, se añadió un set de validación, se profundiza en los valores atípicos, el tratamiento de los outliers por medio de winsorización, se incluyeron las variables categóricas por medio de one hot encoding en el análisis y se determinó su significancia con la variable 'loan status', en lugar de utilizar undersampling se emplean técnicas de sobremuestreo (SMOTE y oversampling aleatorio) para no perder información de la clase mayoritaria. Gracias a estas herramientas se seleccionaron 15 variables para la clasificación del estatus del préstamo. Se logra diagnisticar los modelos considerando la varianza (overfitting) y sesgo (underfitting) por medio de las métricas en cada uno de los sets (train, val y test) y las matrices de confusión para un apoyo visual.

Los resultados muestran una mejora en comparación al primer modelo desarrollado, dónde anteriormente se tenían métricas que rondaban el 75% ahora se tiene alrededor de un 85% para el mejor modelo implementado desde 0 y sin frameworks. Cómo es de esperarse el Random Forest implementado con librerías obtuvo un desempeño excepcional, si bien posee un ligero overfitting, tiene bajo sesgo ya que logra clasificar correctamente ambas clases con un accuracy y f1-score del 95% para los datos de testeo.

Los resultados de los modelos implementados a lo largo del proyecto fueron consistentes entre sí y entre la toma de decisiones que realizan las entidades financieras. Al igual que los bancos, nuestros modelos se apoyan en gran manera en que los solicitantes no tengan préstamos incumplidos anteriormente (previous loan defaults on file), lo que se conoce típicamente como el buró de crédito, lo que le representaría al solicitante adquirir ese préstamo en términos de su ingreso (loan percent income) y la tasa de interés del préstamo (loan int rate).

Con estos modelos se podría ahorrar tiempo y dinero en la aprobación y rechazo de solicitudes, es importante mencionar que aunque los modelos presentan un muy buen desempeño, la última decisión le debería de corresponder a un humano.

7. Bibliography

- Chen, Y., Li, L., Li, W., Guo, Q., Du, Z., & Xu, Z. (2024). Fundamentals of neural networks. AI Computing Systems, 17–51. https://doi.org/10.1016/B978-0-32-395399-3.00008-1
- ¿Qué miran los bancos para apobar o denegar un préstamo? (n.d.). Retrieved August 27, 2025, from https://www.bbva.com/es/salud-financiera/como-aprueban-o-deniegan-un-prestamo-criterio-entidades-financieras/
- Regresión logística en el aprendizaje automático - GeeksforGeeks. (n.d.). Retrieved August 27, 2025, from https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/
- Loan Approval Classification Dataset. (n.d.). Retrieved August 28, 2025, from https://www.kaggle.com/datas approval-classification-data

- Definición de inteligencia artificial responsable - Azure Machine Learning | Microsoft Learn. (n.d.). Retrieved August 28, 2025, from https://learn.microsoft.com/es-es/azure/machine-learning/concept-responsible-ai?view=azureml-api-2
- Cervinski, M. A., Bietenbeck, A., Katayev, A., Loh, T. P., Van Rossum, H. H., & Badrick, T. (2023). Advances in clinical chemistry patient-based real-time quality control (PBRTQC). Advances in Clinical Chemistry, 223–261. https://doi.org/10.1016/bs.acc.2023.08.003
- Como lidiar con el desbalanceo de datos | Alura Cursos Online. (n.d.). Alura. https://www.aluracursos.com/blog/lidiar-con-el-desbalanceo-de-datos
- Cochran, William G. (1952). «The Chi-square Test of Goodness of Fit». The Annals of Mathematical Statistics 23 (3): 315-345
- NV Chawla, KW Bowyer, LOHall, WP Kegelmeyer, "SMOTE: técnica de sobremuestreo de minorías sintéticas", Revista de investigación en inteligencia artificial, 321-357, 2002.
- Ibm. (2025, 27 febrero). Random Forest. IBM Think. https://www.ibm.com/mx-es/think/topics/random-forest

Appendix 1. Github Repository

Github ML model from scratch