Riesgo por incumplimiento de pagos en créditos de vivienda

Cristhian David Tafur Hernández

Monografía para optar al título de Especialista en analítica y ciencia de datos

Tutor
Raúl Ramos Pollán, PhD en Ingeniería Informática por la Universidad de Oporto

Universidad de Antioquia
Facultad de ingeniería, departamento de ingeniería de sistemas UdeA
Especialización en Analítica y ciencia de datos
Medellín, Colombia
2022

Cita	(Tafur Hernández, 2022)
Referencia **Estilo APA 7 (2020)**	Tafur Hernández. (2022). *Riesgo por incumplimiento de pagos en créditos de vivienda* [Monografía]. Universidad de Antioquia, Medellín.

Especialización en analítica y ciencia de datos, Cohorte **3**.

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes

Decano/Director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego Botía Valderrama

TABLA DE CONTENIDOS

1. Resumen ejecutivo **4**

2. Descripción del problema **5**

2.1 Problema de negocio 5

2.2 Aproximación desde la analítica de datos 5

2.3 Origen de los datos 6

2.4 Métricas de desempeño 6

3. Datos **8**

3.1 Datos originales 8

3.2 Datasets 10

3.3 Descriptiva 11

4. Proceso de analítica **12**

4.1 Pipeline principal 12

4.2 Preprocesamiento 13

4.3 Modelos 13

4.4 Métricas 14

5. Metodología **14**

5.1 Baseline 14

5.2 Validación 16

5.3 Iteraciones y evolución 16

5.4 Herramientas 20

6. Resultados **20**

7. Conclusiones **21**

8. Listado de anexos **22**

1. Resumen ejecutivo

El objetivo del presente proyecto es predecir si un solicitante de crédito hipotecario incumplirá el pago de una o más cuotas del potencial crédito usando técnicas de Machine Learning. Se presenta un flujo experimental de distintos procesos y metodologías de analítica de datos para dar solución al problema. El alcance de dicha experimentación llega hasta el uso de los algoritmos de regresión logística y árboles aleatorios de clasificación combinados con preprocesamientos específicos de los datos. La información para la ejecución del proyecto fue suministrada por "Home Credit" por medio de una convocatoria de competencia de predicción realizada en Kaggle en el año 2018.

El problema de Machine Learning en este caso es uno de clasificación de dos clases, donde se busca identificar si un solicitante incumplirá o no el pago de una o más de las cuotas del potencial crédito. Como es usual en estos casos, el problema se caracteriza por tener clases desbalanceadas, donde la mayor parte de la información de la data de entrenamiento es de clientes que no han incumplido sus obligaciones crediticias y una mínima parte de quienes sí lo han hecho.

El proceso experimental en el presente proyecto incluye la evaluación y selección de estrategias de remuestreo para datos desbalanceados, selección de hiperparámetros y opciones de reducción de dimensionalidad. El ejercicio iterativo usa las métricas de validación derivadas de la matriz de confusión para tomar las diferentes decisiones de elección y hacer seguimiento a las mejoras del modelo.

Como resultado, el alcance de las experimentaciones realizadas en el presente proyecto no logran conseguir un modelo con desempeño satisfactorio. El mejor modelo desarrollado obtiene un accuracy de 75% con asertividad asimétrica entre las clases, logrando un F1 socre de 85% para la clase mayoritaria y de 28% para la minoritaria.

2. Descripción del problema

2.1 Problema de negocio

La colocación de créditos es parte de la esencia del negocio de las entidades de intermediación financiera como los bancos. El desarrollo de esta actividad va atada a la gestión adecuada de los riesgos que aseguren la recuperación de los recursos prestados y la rentabilidad del negocio.

En este contexto toman el protagonismo 2 agentes económicos, los bancos y los solicitantes de créditos. La relación entre ellos está basada en flujos de información asimétrica, donde el banco no logra conocer con total certeza si el solicitante tiene la capacidad e intención de cumplir a cabalidad con el pago del crédito demandado. Dado lo anterior, el banco debe hacer uso de la información disponible para intentar identificar con la mayor precisión posible la capacidad e intención de pago de los solicitantes y de acuerdo a ello prestar o no los recursos.

El banco se enfrenta entonces a un problema de elección, donde por un lado debe minimizar el riesgo y asegurar que a los solicitantes a quienes les aprueba el crédito van a cumplir con sus obligaciones, pero a su vez, evitar que una muy alta aversión al riesgo le implique una reducción de clientes (que sí cumplirían) y pierda potenciales beneficios por su intermediación financiera.

2.2 Aproximación desde la analítica de datos

Como se expresó en el análisis del problema de negocio se trata de un reto de gestión de información disponible con el fin de identificar la capacidad e intención de cumplimiento por parte del solicitante. En el presente proyecto el problema de analítica consiste entonces en usar la información disponible para clasificar a los solicitantes en dos clases, con y sin riesgo de incumplimiento de pago.

Esto en términos de analítica de datos se traduce en un problema de clasificación de 2 clases. Existen diversas alternativas para tratar estos algoritmos con respuestas categóricas, donde se modela la información y se asigna una clase a cada una de las muestras/ sujetos.

2.3 Origen de los datos

Los datos son suministrados por una institución financiera llamada "Home Credit" por medio de una convocatoria de competencia de predicción realizada en Kaggle[1] en el año 2018. Los datos están compuestos por tablas con información a detalle de los diferentes clientes de la entidad y etiquetados de forma binaria de acuerdo a si han incumplido o no con una o más cuotas del crédito que les fue otorgado.

2.4 Métricas de desempeño

2.4.1 Métricas de machine learning

Al tratarse de un problema de clasificación, se recurre al uso de la matriz de confusión y las métricas derivadas de ellas para evaluar el desempeño del modelo desarrollado. En términos generales se evalúa lo siguiente:

A. Solicitantes clasificados como "Sin riesgo de incumplimiento" y que cumplen con la totalidad de sus cuotas.

B. Solicitantes clasificados como "Sin riesgo de incumplimiento" y que incumplen con una o más de sus cuotas.

C. Solicitantes clasificados como "Con riesgo de incumplimiento" y que incumplen con una o más de sus cuotas.

D. Solicitantes clasificados como "Con riesgo de incumplimiento" y que cumplen con la totalidad de sus cuotas.

Desde una perspectiva técnica se obtienen métricas como la precisión, recall (sensibilidad) y F1 score de cada clase, así como el accuracy general del modelo. La precisión muestra el porcentaje de acierto tomando como base la cantidad de muestras predichas como de la respectiva clase, mientras que el recall toma como base la cantidad de muestras que en realidad son de esa clase. El F1 score es una métrica resumen de la precisión y el recall, mientras que el accuracy mide el

[1] Kaggle es la comunidad en línea de científicos de datos más grande del mundo

porcentaje de acierto general del modelo (las muestras clasificadas con acierto sobre el total de muestras).

Dicho lo anterior, la precisión mide que tan confiable es la respuesta del modelo cuando clasifica una muestra como de la respectiva clase, por otro lado, el recall mide que tan bien el modelo puede detectar la clase. Como son métricas distintas, el F1 score busca hacer un resumen de ambas y arrojar un único valor de evaluación para la clase.

2.4.2 Métricas de negocio

Desde una perspectiva del negocio el modelo se puede evaluar en dos vías; primero, en que tan bueno es gestionando el riesgo al ver el porcentaje de acierto cuando clasifica a clientes como quienes incumpliran y NO incumpliran y en realidad son clientes con dicho comportamiento (A y C del listado de 2.4.1). Segundo, una evaluación de aversión negativa al riesgo que vendría definida por los clientes clasificados como quienes incumpliran con sus pagos y en realidad no lo hacen (D del listado de 2.4.1).

Las mediciones anteriores pueden llevarse fácilmente a valores monetarios para tener una magnitud de impacto clara para el negocio. La mala gestión del riesgo implica una cantidad n1 de clientes a quienes se les presta dinero e incumplen con sus pagos, lo que genera reducciones en los ingresos esperados por intermediación y gastos adicionales en la activación de los distintos mecanismos de cobranza de la institución. En cuanto la aversión negativa al riesgo implica una cantidad n2 de clientes a quienes no se les presta dinero pero que en realidad hubiesen cumplido con la totalidad de su obligación, por ende el banco pierde esos ingresos de intermediación.

En resumen del presente apartado, las métricas de desempeño del modelo permiten aterrizar el impacto del mismo en el negocio y definir bajo qué circunstancias este generaría la confianza de ser llevado a producción. Un desempeño deficiente del modelo de clasificación implica costos económicos para la institución financiera. Dado lo anterior, es de esperar que para que un modelo pase a ser usado debe tener errores máximo del 10%.

3. Datos

3.1 Datos originales

La información para la ejecución del proyecto fue suministrada por "Home Credit" por medio de una convocatoria de competencia de predicción realizada en Kaggle.com en el año 2018. Los datos para este proyecto están distribuidos en un total de 7 tablas. En primer lugar, está la tabla principal con la muestra de los créditos a analizar para la implementación del modelo predictivo; las demás tablas traen información extra de los titulares de estos créditos con relación a su comportamiento en pagos con otros préstamos dentro de la misma institución financiera u otras instituciones. La Tabla 1 y Figura 1 hacen un resumen técnico y representación gráfica de los datos del proyecto. El acceso a los mismos está disponible por medio de la plataforma Kaggle.com[2].

Tabla 1. Descripción técnica de los datos del proyecto

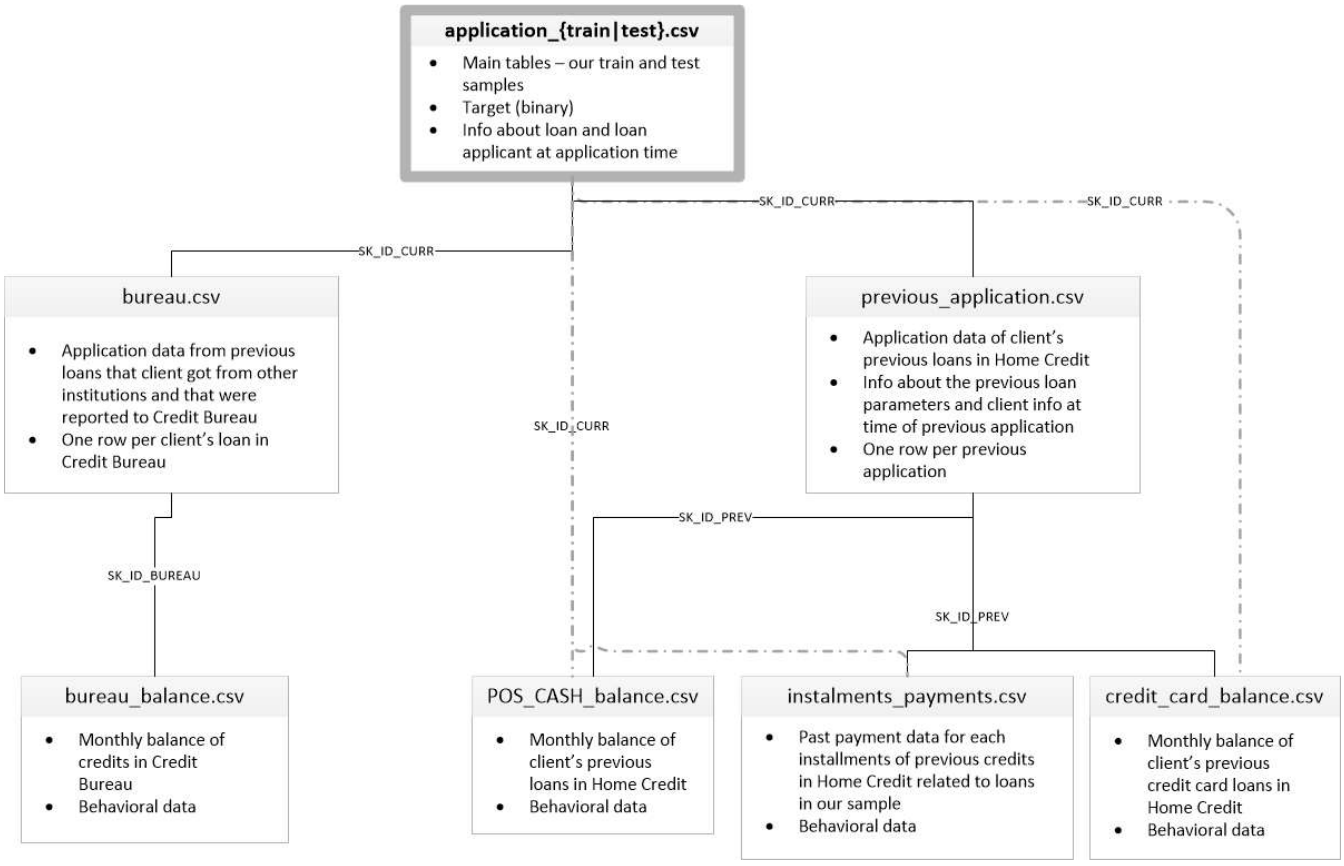
Fichero	Descripción	Número de columnas	Número de muestras	Formato	Tamaño (KB)
application_train	Tabla con datos estáticos para los créditos a usar para el modelo. Una fila representa un préstamo en la muestra	122	307.511	csv	162.240
application_test	Tabla con datos estáticos de créditos para evaluar el modelo . Una fila representa un préstamo en la muestra (Sin clasificación	121	48.744	csv	25.945
bureau	Tabla con todos los créditos anteriores del cliente proporcionados por otras instituciones financieras que se informaron al Buró de Crédito (para los clientes que tienen un préstamo en la muestra). Para cada préstamo de la muestra, hay tantas filas como créditos tenía el cliente en el Buró de crédito antes de la fecha de solicitud.	17	1.716.428	csv	166.032
bureau_balance	Tabla con una fila para cada mes del historial de cada crédito anterior reportado al Burú de Crédito, es decir, la tabla tiene (# préstamos en la muestra * # de créditos	3	27.299.925	csv	366.790

[2] URL de la competencia de Kaggle: https://www.kaggle.com/c/home-credit-default-risk

	anteriores relativos * # de meses donde se tiene algo de historial observable para los créditos anteriores).				
previous_application	Tabla con todas las solicitudes anteriores de préstamos de Home Credit a clientes que tienen préstamos en la muestra. (solicitudes que no necesariamente implican un crédito aprobado).	37	1.670.214	csv	395.482
POS_CASH_balance	Tabla con una fila para cada mes del historial de cada crédito anterior en Home Credit (crédito al consumo y préstamos en efectivo) relacionado con los préstamos de la muestra, es decir, la tabla tiene (# préstamos en la muestra * # de créditos anteriores relativos * # de meses en el que se algo de historial observable para los créditos anteriores).	8	10.001.358	csv	383.500
installments_payments	Historial de reembolso de los créditos previamente desembolsados en Crédito para la vivienda relacionados con los préstamos de la muestra.	8	13.605.401	csv	706.171
credit_card_balance	Tabla con una fila para cada mes de historial de cada crédito anterior en Home Credit (crédito al consumo y préstamos en efectivo) relacionado con préstamos en la muestra, es decir, la tabla tiene (# préstamos en la muestra * # de tarjetas de crédito anteriores relativas * # de meses en los que tenemos algo de historial observable para la tarjeta de crédito anterior).	23	3.840.312	csv	414.632

Fuente: Elaboración del autor

Figura 1. Estructura de los datos del proyecto



Fuente: Home Credit Default Risk. Kaggle.
https://www.kaggle.com/competitions/home-credit-default-risk/data

3.2 Datasets

La obtención de los dataset de entrenamiento y test para el modelo usan inicialmente solo la tabla principal "application_train". Partiendo de este fichero se hace la exploración inicial de los datos, la transformación de los mismos y la división en "train" y "test" para la primera iteración del modelo. También se procesaron las tablas "bureau" y "previous_application" para ser incluidas como información adicional iteraciones posteriores del proyecto. Los demás datasets que suministra la fuente no fueron usados en la experimentación del presente ejercicio predictivo.

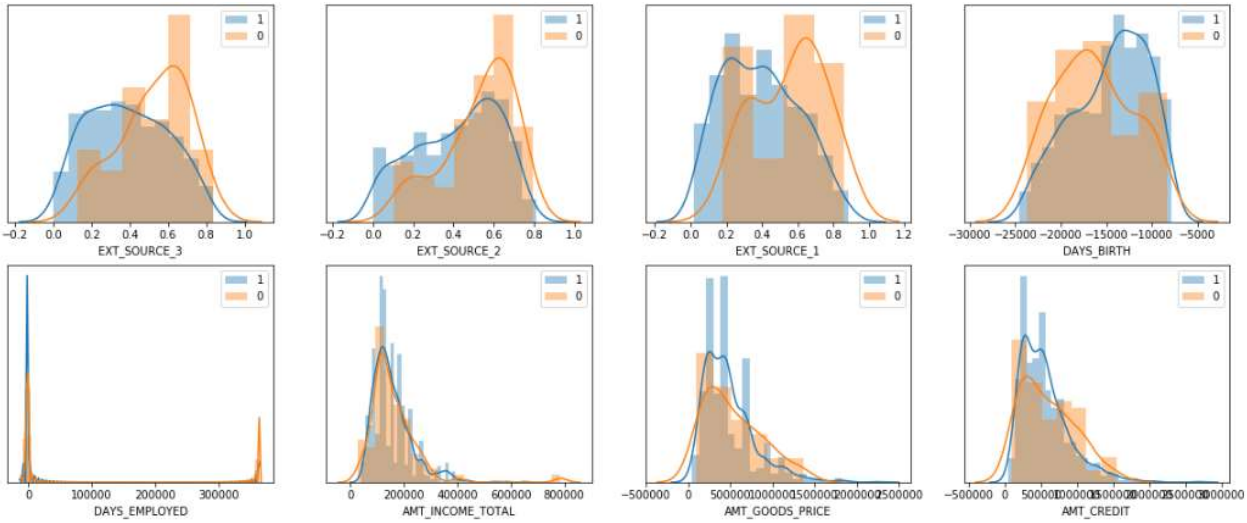
3.3 Descriptiva

El desarrollo del presente proyecto usa en las primeras iteraciones del modelo exclusivamente la tabla Aplication_train la cual es la tabla principal del proyecto, esta contiene información para una muestra de créditos etiquetados con la clasificación objetivo del proyecto e información del titular del crédito al momento de su solicitud. Son datos estáticos para todas las aplicaciones; una fila representa un préstamo en la muestra, es decir, no hay información repetida para ninguno de los créditos de la data.

En la tabla se tiene un "ID" de la muestra, el "Target" que contiene la clasificación, 48 variables categóricas y 72 numéricas. El "Target" trae dos clases (0 y 1), donde "1" representa a los clientes con dificultades de pago, clientes que tuvieron un retraso en el pago de más de "X" días en al menos una de las primeras "Y2 cuotas del préstamo en nuestra muestra y "0" para todos los demás casos. Al evaluar la distribución de las clases se observa que se trata de un problema de clasificación con datos desbalanceados donde el 91,9% de las muestras son de la clase "0" y el 8,1% de la clase "1".

La exploración de datos se hace en dos bloques, uno para las variables numéricas y otro para las categóricas. En el bloque numérico se estimó la correlación entre las características y se aplicó un análisis de la varianza con el fin de evaluar si existe diferencia entre las medias de las variables al agruparlas por cada una de las dos clases del TARGET (0 y 1). – (media de las muestras de la clase 0 vs media de las muestras de la clase 1). De este análisis ANOVA se encuentra que 65 de las 72 variables tienen medias estadísticamente diferentes al agruparse por las clases del TARGET, la figura 2 expone gráficamente unos ejemplos de esta evaluación de diferencia de medias. Para el bloque de variables categóricas se visualizan las clases de cada una.

Figura 2. Ejemplos gráficos de análisis ANOVA de variables numéricas



Fuente: Elaboración del autor

4. Proceso de analítica

4.1 Pipeline principal

La Figura 3 muestra el flujo de trabajo general que se aplicó en el presente proyecto, se muestra de forma general los pasos y experimentaciones realizadas para la obtención del modelo de clasificación buscado.

Figura 3. Flujo de trabajo general del proyecto

Fuente: Elaboración del autor

4.2 Preprocesamiento

El preprocesamiento de los datos para el presente proyecto se aplicó en dos bloques, uno para variables numéricas y otro para las categóricas. Para las variables numéricas se recurre a imputar los valores faltantes con la mediana de cada una de las características. En cuanto a las categóricas, estas se transformaron usando el método One Hot Encoding para todas las clases, incluidas las NaN en cada variable. Luego de tener esta estructura, la información se reescala usando el método MinMaxScaler que transforma los valores en una escala de 0 a 1 partiendo de los valores mínimos y máximos de cada variable. Lo anterior aplica tanto para la tabla principal application, como para las tablas bureau y previous_applications que se usan para ampliar las información de ingesta al modelo.

Dado el problema de desbalance en el target de la información, el preprocesamiento de los datos también incluye pruebas de remuestreo. Se evaluó el subsampling que es la eliminación de muestras de la clase mayoritaria de acuerdo a una evaluación de similitudes entre vecinos más cercanos, también se aplicó la metodología de oversampling que implica la creación de muestras sintéticas de la clase minoritaria y una alternativa combinada de sub y oversampling.

4.3 Modelos

En el modelamiento del proyecto se prueban dos algoritmos de clasificación, uno de regresión logística y otro de bosques aleatorios de clasificación. Inicialmente se utilizan los valores por defecto de los hiperparámetros del modelo con las distintas opciones de remuestreo para solucionar el problema de desbalance de los datos y usando el hiperparámetro dispuesto en cada algoritmo para el tratamiento de este tipo de datos.

Usando estos algoritmos tal y como vienen especificados por defecto y las métricas de validación se evalúa y selecciona la mejor alternativa para tratar los datos desbalanceados. Posteriormente, se testean distintas combinaciones de hiperparámetros para los dos algoritmos de clasificación usados y se elige la mejor combinación de hiperparámetros para cada uno de los algoritmos a partir del valor de accuracy, luego se confrontan los respectivos “mejores modelos” y se selecciona uno de los dos.

Después de tener identificada la mejor forma de hacer frente a los datos desbalanceados y el mejor modelo (algoritmo y sus respectivos hiperparámetros), se hace la estimación base del proyecto la cual suministra las métricas de validación de partida para mejorar el modelo. A este modelo base se le prueban dos opciones de mejora, la primera es la ampliación de la información usando las tablas adicionales, y la segunda es reduciendo la dimensionalidad de los datos por medio de la identificación de las Features más relevantes y la aplicación del método de análisis de componentes principales (PCA).

4.4 Métricas

Tal como se mencionó antes, al tratarse de un problema de clasificación se recurre a la matriz de confusión y las métricas derivadas de ella para testear los modelos. Son los valores de precisión, recall y F1 score de cada clase junto con el accuracy del modelo los que permiten la toma de decisiones y la selección de las distintas alternativas evaluadas. Para obtener dichas métricas se recurre a las herramientas suministradas por sklearn.metrics relacionadas a la matriz de confusión y los indicadores basados en ella.

5. Metodología

5.1 Baseline

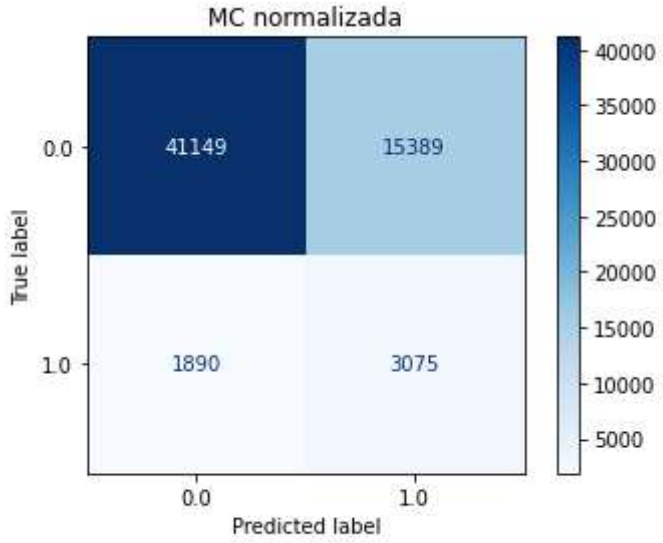
La iteración base del presente proyecto parte del resultado obtenido luego del preprocesamiento de datos, la evaluación y selección de la alternativa para tratar los datos desbalanceados y la confrontación de la regresión logística y el bosque aleatorio de clasificación con los mejores hiperparámetros testeados. Es decir luego de completar los pasos del 1 al 4 del flujo de trabajo presentado en la Figura 3.

La iteración base es de por sí el resultado de distintas experimentaciones previas enfocadas a definir aspectos como los descritos en el párrafo anterior. Respecto al tratamiento de los datos desbalanceados, luego de probar las 3 estrategias de remuestreo (subsampling, oversampling y mixta) y los hiperparámetros de asignación de pesos que traen los algoritmos de clasificación

para este problema, se establece que son estos hiperparámetros de pesos la mejor alternativa para hacerle frente al desbalance.

Luego, se procedió a la aplicación de un grid search a cada uno de los algoritmos en consideración para probar entre un set arbitrario de hiperparámetros los que mejor funcionan; al tener las mejores especificaciones de cada modelo se confrontaron entre ellos para seleccionar el mejor. El fruto de todo el proceso descrito en este apartado es el modelo base y sus resultados se muestran a continuación en la Figura 4.

Figura 4. Resultado del modelo base



	precision	recall	f1-score	support
0.0	0.96	0.73	0.83	56538
1.0	0.17	0.62	0.26	4965
accuracy			0.72	61503
macro avg	0.56	0.67	0.54	61503
weighted avg	0.89	0.72	0.78	61503

Fuente: Elaboración del autor

La Figura 4 muestra el resultado de un bosque aleatorio de clasificación con un total de 200 árboles, una profundidad máxima de 10 ramificaciones y una asignación de pesos a las clases de

“balanced_subsample”, siendo esta la mejor especificación identificada a a la hora de testear los hiperparámetros.

Al evaluar el Accuracy vemos un acierto general de 72%, sin embargo el desempeño es muy desigual entre las clases, donde la clase mayoritaria (0) tiene un F1 score de 83%, y la minoritaria de tan solo un 26%. Es decir que el modelo tiene una dificultad importante para identificar la clase minoritaria.

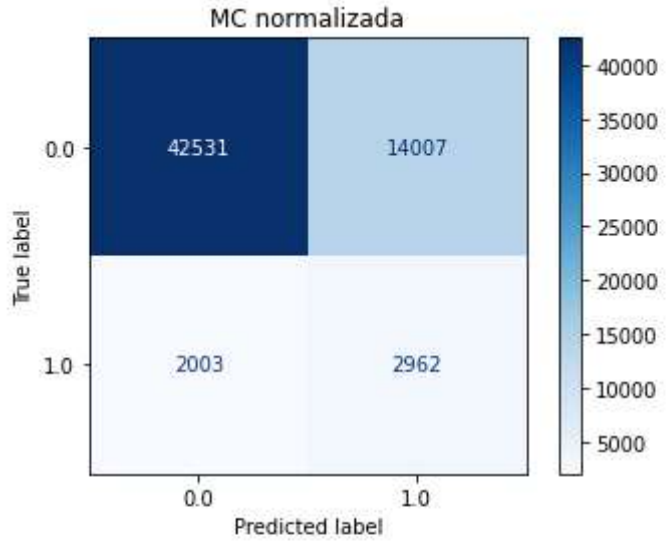
5.2 Validación

La partición de los datos para la validación del modelo fue diseñada con una distribución de 80% de los datos para entrenamiento y 20% para test. Esta partición se realizó de forma estratificada de acuerdo a la distribución de las muestras respecto al Target del proyecto. Adicionalmente, cuando se aplicó el grid search para la selección de los hiperparámetros se usó el método de validación cruzada con un total de 5 Folds.

5.3 Iteraciones y evolución

Luego de lo expuesto en el apartado 5.1 que describe el proceso que deriva en la iteración base del proyecto, se evaluó el impacto de dos cambios sobre el modelo; uno en relación a ampliar los datos de ingesta al modelo y el otro al reducir la dimensionalidad de los mismos. La Figura 5 muestra el resultado obtenido por la ampliación de la data.

Figura 5. Resultado del modelo con información ampliada



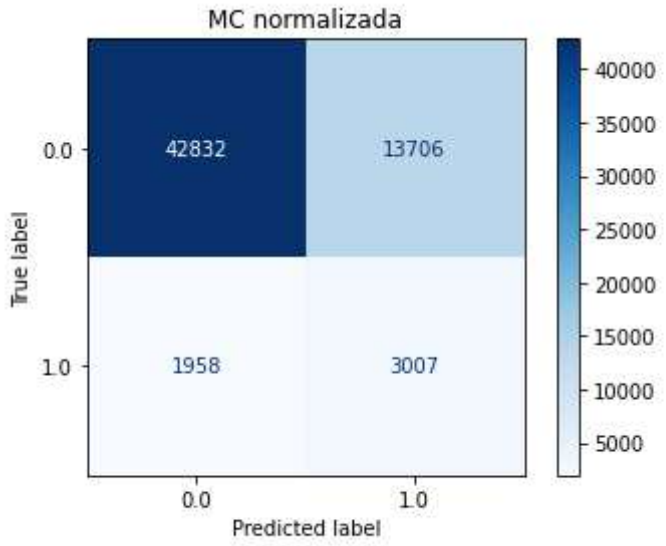
	precision	recall	f1-score	support
0.0	0.96	0.75	0.84	56538
1.0	0.17	0.60	0.27	4965
accuracy			0.74	61503
macro avg	0.56	0.67	0.56	61503
weighted avg	0.89	0.74	0.80	61503

Fuente: Elaboración del autor

En este caso, la ampliación de la data implica pasar de 244 a 441 features al adherir las tablas de bureau y previous_application preprocesadas. Respecto a la iteración base, vemos que el accuracy pasa de 72% a 74%, y que el desempeño desigual por clases continua, donde el F1 score de ambas clases aumenta en un punto porcentual. De lo anterior se puede concluir que ampliar la data no tiene un impacto significativo sobre las métricas de validación del modelo.

Continuando con las iteraciones propuestas, se aprovecha la alternativa que ofrecen los bosques aleatorios para identificar la importancia de las Features en el proceso predictivo. El objetivo entonces es reducir la dimensionalidad de la data manteniendo solo las características que acumulan aproximadamente el 80% de la importancia total; aplicando esto, se pasa de 441 a 64 features. La Figura 6 muestra el resultado de esta iteración.

Figura 6. Resultado del modelo de reducción de dimensionalidad basado en la importancia de las características

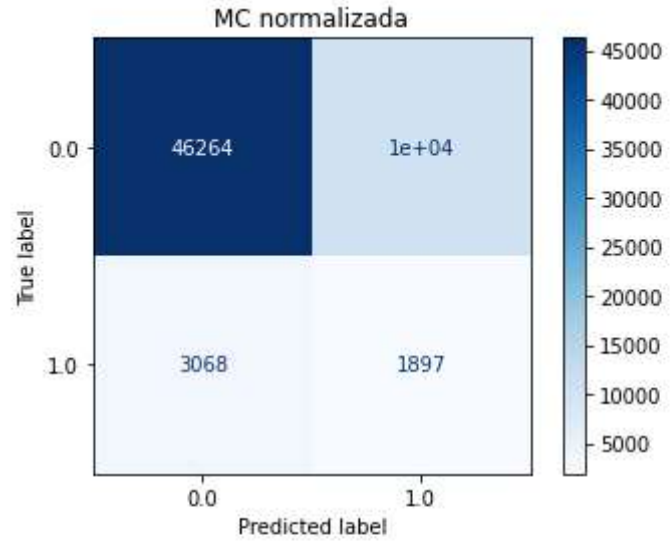


	precision	recall	f1-score	support
0.0	0.96	0.76	0.85	56538
1.0	0.18	0.61	0.28	4965
accuracy			0.75	61503
macro avg	0.57	0.68	0.56	61503
weighted avg	0.89	0.75	0.80	61503

Fuente: Elabolación del autor

Respecto a la iteración previa (la de ampliación de la data), se observa que al reducir la dimensionalidad y dejar solo las 64 características más importantes tanto el accuracy como el F1 score de las dos clases mejoran marginalmente. Adicionalmente, también se prueba el efecto de reducir la dimensionalidad aplicando el método de análisis de componentes principales (PCA) cuyo resultado se muestra en la Figura 7.

Figura 7. Resultado del modelo de reducción de dimensionalidad usando PCA



	precision	recall	f1-score	support
0.0	0.94	0.82	0.87	56538
1.0	0.16	0.38	0.22	4965
accuracy			0.78	61503
macro avg	0.55	0.60	0.55	61503
weighted avg	0.87	0.78	0.82	61503

Fuente: Elabolación del autor

En el caso de la iteración con PCA, se decide realizar la ejecución con un total de 90 componentes los cuales representan entre el 85% y 87% de la información total de la varianza. Se observa que si bien el accuracy es mejor respecto a la reducción por importancia de las características, la diferencia en acierto entre las clases aumenta, donde este modelo aprende a identificar mejor la clase mayoritaria pero empieza a fallar más en la minoritaria, lo anterior al analizar las variaciones sobre el F1 score.

5.4 Herramientas

El desarrollo del presente proyecto se realizó utilizando el lenguaje de programación Python y el entorno de desarrollo de Google Colaboratory. La librería principal es la de scikit-learn con sus herramientas o desarrollos especializados en los algoritmos de clasificación como LogisticRegression y RandomForestClassifier, las herramientas de preprocessing y de metrics para la validación del modelo. Adicionalmente se usan librerías base como pandas, numpy y matplotlib, así como librerías de imblearn para aplicar las diferentes técnicas de remuestreo.

6. Resultados

Después de las diferentes iteraciones realizadas sobre el modelo, el mejor resultado obtenido es el expuesto en la Figura 6, en el que se usan solo las características que representan aproximadamente el 80% de la importancia total de las variables en el modelo de bosque aleatorio de clasificación con 200 árboles, una profundidad máxima de 10 ramificaciones y la asignación de pesos a las clases para el desbalance. Si bien el accuracy del modelo con uso del método de PCA es mayor, este pierde eficiencia identificando la clase minoritaria, por ello no se considera el mejor resultado.

Volviendo a la iteración del mejor resultado (Figura 6), se observa que el modelo logra un F1 score de 85% para la clase mayoritaria, pero de tan solo un 28% para la minoritaria. El problema del desbalance no permite que los modelos desarrollados en el presente proyecto logren identificar de forma satisfactoria la clase minoritaria (clientes con riesgo de incumplimiento); adicionalmente, el error de la clase mayoritaria no alcanza a estar por debajo del 10%, por lo que tampoco es un resultado del todo deseado para esta clase.

7. Conclusiones

El alcance de las experimentaciones realizadas en el presente proyecto no logran conseguir un modelo con desempeño satisfactorio. El mejor modelo desarrollado obtiene un accuracy de 75% con asertividad asimétrica entre las clases, logrando un F1 socre de 85% para la clase mayoritaria y de 28% para la minoritaria. Por lo anterior, no es posible concluir que se haya obtenido un desarrollo óptimo para aplicar en el negocio de estudio, sin embargo, se consigue un flujo experimental adecuado que consigue mejoras en el desempeño en sus respectivas iteraciones.

Es posible continuar con la mejora iterativa del modelo de clasificación considerando la inclusión de la información adicional suministrada por la fuente y que no fue usada, distintas alternativas de preprocesamiento de los datos y/o el uso de algoritmos de clasificación más sofisticados; aspectos que no fueron planificados en el alcance del presente proyecto pero que se pueden desarrollar en proyectos futuros.

Partiendo de un criterio arbitrario, se puede considerar que un modelo con desempeño satisfactorio sería aquel que logre una asertividad por encima del 90%, es decir, con errores máximo del 10% (basado en los niveles de significancia máximos que se suelen usar en ejercicios estadísticos); se debe enfatizar que lo deseado es que este nivel de predicción sea para ambas clases, tanto la mayoritaria como la minoritaria, no basta un accuracy general por encima del 90% si el modelo tiene dificultades para identificar alguna de las clases.

8. Listado de anexos

Archivo excel con descripción de cada una de las variables contenidas en los datos usados en el presente proyecto.

- Type_variables.xlsx

Notebooks del proyecto:

- 01. Cargue y exploración de datos
- 02. Gestión de datos desbalanceados y selección de estrategia
- 03. Evaluación de hiperparámetros y selección del mejor modelo
- 04. Preprocesamiento del dataset para ampliar la información de entrenamiento
- 05. Ejecución del mejor modelo con el dataset ampliado
- 06. Reducción de dimensionalidad y prueba de desempeño

Estos anexos pueden encontrarse en el repositorio de GitHub:
https://github.com/cristhiant24/EACD-HomeCreditDefaultRisk