Analítica de datos para predicciones en aprobaciones de tarjetas de crédito

María Isabel Martínez Rendón
Santiago Felipe Rosales Guerrero

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor
Javier Fernando Botia Valderrama, Doctor (PhD)

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2023

Cita	Martínez Rendón y Rosales Guerrero [1]
Referencia Estilo IEEE (2023)	[1] M. I. Martínez Rendón y S. F. Rosales Guerrero, "Analítica de datos para predicciones en aprobaciones de tarjetas de crédito", Trabajo de grado especialización, Especialización en Analítica y Ciencia de Datos, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2023.

Especialización en Analítica y Ciencia de Datos, Cohorte IV.

Centro de Investigación Ambientales y de Ingeniería (CIA)

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

PhD	Philosophiae Doctor
ROS	Random Over Sampler
SMOTE	Synthetic Minority Oversampling Technique
SMOTEN	Synthetic Minority Over-sampling Technique for Nominal
SMOTENC	Synthetic Minority Over-sampling Technique for Nominal and Continuous
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
PSO	Particle Swarm Optimization
AUC	Area Under the Curve
ADASYN	Oversample using Adaptive Synthetic
GBDT	Gradient Boosted Decision Trees
SVM	Support vector machine
KNN	K-nearest neighbor**s**
ROC	Receiver operating characteristic
XGBoost	Extreme Gradient Boosting

RESUMEN

Uno de los objetivos de los bancos es ofrecer tarjetas de crédito a clientes que tengan un buen comportamiento de pago. Con esto en mente, el objetivo de este documento es indicar cómo a través de la información personal e historial de pagos, de los clientes existentes en el banco, se puede catalogar a un posible cliente como apto o no apto para la aprobación de un cupo de tarjeta de crédito.

Además de la adecuación y limpieza de los datos, es necesario implementar metodologías para hacer un tratamiento al desbalance de las etiquetas de la base de datos, ya que por lo general los bancos tienen muchos clientes con buen comportamiento en sus pagos y muy pocos que incumplen. Luego, se realiza una competencia de modelos teniendo como línea base la regresión logística, los modelos implementados fueron: *Máquina de Soporte Vectorial, Árboles de Decisión, AdaBoost y Árboles Aleatorios*.

El mejor resultado se obtuvo con el método de sobremuestreo *SMOTENC* y con el modelo *Árboles Aleatorios* con un *Accuracy* de 98.8%, *Balance accuracy score* de 81.2% y *Log Loss* de 0.035.

Palabras clave **— Tarjeta de crédito, bancos, modelos de clasificación, aprendizaje automático, desbalance de datos.**

ABSTRACT

One of the objectives of banks is to offer credit cards to customers with good payment behavior. With this in mind, the objective of this document is to indicate how, through the personal information and payment history of the bank's existing customers, a potential customer can be categorized as eligible or ineligible for credit card quota approval.

In addition to the adequacy and cleanliness of the data, it is necessary to implement methodologies to treat the imbalance of the database labels, since banks usually have many customers with good payment behavior and very few who default. Then, a model competition is carried out having logistic regression as a baseline, the models implemented were: *Support Vector Machine, Decision Trees, AdaBoost* and *Random Forest.*

The best result was obtained with the *SMOTENC* oversampling method and with the *Random Forest* model with an *Accuracy* of 98.8%, *Balance accuracy score* of 81.2% and *Log Loss* of 0.035.

***Keywords* - Credit card, banks, classification models, machine learning, data imbalance.**

I. INTRODUCCIÓN

Con el objetivo de minimizar las pérdidas, los bancos han optado por analizar información de sus clientes y así tomar mejores decisiones. En este documento, se busca analizar el comportamiento financiero de los clientes de un banco para clasificar a nuevos usuarios como aptos o no aptos para recibir un cupo de una tarjeta de crédito; partiendo de dos bases de datos donde la primera contiene información personal de los clientes con 438.557 registros o muestras y la segunda, el historial de pagos con 1.048.575 de registros o muestras [1].

Se buscará el mejor modelo capaz de clasificar si un cliente es apto o no para recibir un cupo de tarjeta de crédito. Para determinar si el modelo puede ser usado como herramienta para la clasificación de clientes en el sector financiero, la métrica de negocio es que el *accuracy* del modelo debe superar un umbral por encima del 75%.

Teniendo en cuenta lo anterior, es necesario realizar la exploración de las bases de datos y a partir de la información contenida en ellas, determinar cómo se puede catalogar a un cliente como apto o no apto. Para abordar el tema de riesgo de no pago, se puede partir del análisis vintage o análisis de cohortes, usado en este tipo de problemas, este análisis permite determinar que un cliente no apto es aquel que no paga en un periodo superior a los 90 días y uno apto es el que paga dentro de este periodo [2]. Además, la exploración de las bases de datos permite realizar una buena adecuación de las mismas, eliminando valores atípicos, dar tratamiento a datos faltantes y revisar que los datos sean coherentes; verificando que los clientes sean mayores de edad, que la cantidad de miembros de familia y cantidad de hijos estén relacionadas, que se hayan diligenciado los campos correctamente, etc.

Lo segundo a tener en cuenta, es que por lo general los datos bancarios tienen un desbalance considerable, ya que lo normal es tener más clientes con buen comportamiento en sus pagos y muy pocos que incumplen. Para subsanar este problema se realizarán diferentes pruebas con métodos de sobremuestreo de la información y generar datos sintéticos que permitan entrenar los modelos, los métodos son: ***Random Over Sampler (ROS)***, ***SMOTE***, ***SMOTEN***, ***KMeansSMOTE***, ***BoderlineSmote*** y ***SMOTENC***. La línea base para determinar qué método de sobremuestreo es adecuado, se realiza con un modelo de *Regresión Logística* y se elige el de mejor puntuación.

Una vez elegido el mejor método de sobremuestreo, se implementa una competencia de modelos entre los cuales están: ***Máquina de Soporte Vectorial***, ***Árboles de Decisión***, ***AdaBoost*** y ***Árboles Aleatorios***. Para cada uno de los modelos se realiza la búsqueda de sus mejores hiperparámetros, teniendo en cuenta que el entrenamiento se hace con los datos sobremuestreados y en sus pruebas se usan los datos originales (desbalanceados). Finalmente, se escoge el modelo con los mejores resultados.

El documento está organizado con la siguiente estructura: un resumen conceptual del proyecto, la introducción que aborda el tema principal del trabajo, la metodología aplicada para la solución, los diferentes resultados obtenidos con sus respectivos análisis y por último las conclusiones.

II. ESTADO DEL ARTE

Uno de los estudios investiga el uso del análisis predictivo para mejorar la estrategia de gestión de riesgos de las entidades financieras a la hora de clasificar a los solicitantes de tarjetas de crédito. El estudio utiliza un conjunto de datos reales de un banco de Taiwán y construye cuatro algoritmos de minería de datos para predecir los deudores de tarjetas de crédito: *Árboles de decisión, Regresión logística, Árboles Aleatorios* y *Redes neuronales*, siendo el modelo de red neuronal prealimentada (*Feedforward Neural Network*) el que mejores resultados obtiene, con una precisión predictiva del 82% [3].

En otro trabajo se analiza la importancia de identificar el riesgo de crédito para evitar pérdidas de ingresos y amenazas a la rentabilidad. Explica que los errores en el análisis del crédito pueden provocar riesgos crediticios, como la pérdida de clientes y la incertidumbre sobre el reembolso de los préstamos. El documento sugiere que las técnicas de clasificación de minería de datos pueden utilizarse para determinar el riesgo crediticio y describe la minería de datos como un proceso computacional que utiliza métodos como la inteligencia artificial, el aprendizaje automático y la estadística para expresar patrones de datos. El estudio descubrió que el uso de un método híbrido que combinaba la selección de características *PCA* (*Principal Component Analysis*) y *PSO* (*Particle Swarm Optimization)* con un algoritmo de red neuronal daba como resultado la mayor precisión, del 82,67%. El estudio también descubrió que la adición de la optimización *PSO* al algoritmo de red neuronal mejoraba el valor *AUC* (*Area Under the Curve*) de 0,706 a 0,749. Los resultados sugieren que las técnicas de minería de datos pueden ser útiles para el análisis y la predicción del riesgo crediticio [4].

Otra propuesta es un modelo para la predicción de la morosidad de créditos utilizando varios conjuntos de datos relacionados con el crédito (bases de datos de clientes de Taiwán, el sur de Alemania y Bélgica). El modelo emplea técnicas de remuestreo a nivel de datos para superar el problema del desbalanceo de datos, tales como: *SMOTE* (*Synthetic Minority Over-sampling Technique*), *ADASYN* (*Adaptive Synthetic Sampling*), *K-means SMOTE* (*K-means Synthetic Minority Over-sampling Technique*); siendo el último el mejor método. Aplica varios modelos de aprendizaje automático entre ellos *Árbol de Decisión de Gradiente Aumentado* (*GBDT*), *Regresión Logística, Árboles Aleatorios, Máquina de Vectores de Soporte* (*SVM*) y *K-Nearest Neighbors* (*KNN*) para obtener resultados eficientes, donde el mejor fue *GBDT*. Los resultados de

los conjuntos de datos desbalanceados mostraron una precisión del 66,9% en el conjunto de datos de créditos a clientes de Taiwán, del 70,7% en el conjunto de datos de créditos a clientes del sur de Alemania y del 65% en el conjunto de datos de créditos a clientes de Bélgica. Por el contrario, los resultados obtenidos con los métodos propuestos mejoraron significativamente la precisión. 89% para el conjunto de datos de créditos de clientes de Taiwán, 84,6% en el conjunto de datos de créditos de clientes del sur de Alemania y 87,1% en el conjunto de datos de créditos de clientes de Bélgica. Los resultados evidenciaron que el rendimiento de los clasificadores fue mejor en el conjunto de datos balanceado y también se observó que el rendimiento de las técnicas de sobremuestreo de datos es mejor que el de las técnicas de submuestreo [5].

En otro trabajo, se analiza el aumento del uso de las tarjetas de crédito debido al crecimiento de Internet y al incremento de los fraudes con tarjetas de crédito y de los morosos. Como consecuencia, las entidades emisoras de tarjetas de crédito son cada vez más cautelosas a la hora de aprobarlas a los clientes. Se construye un modelo de deep learning en el cual se experimentó con diferentes configuraciones de *Redes Neuronales Profundas*, incluyendo redes de 2, 3, 5 y 7 capas ocultas con 3, 5, 7, 16, 32 y 64 neuronas con el fin de apoyar las decisiones de aprobación de tarjetas de crédito y compara su rendimiento con los algoritmos tradicionales de aprendizaje automático como *Regresión Logística* y *Máquina de Vectores de Soporte*. El resultado del modelo de *Deep Learning* fue: *Accuracy* de 87.1%, *Precision* 87.9% y *Recall* de 89.2% [6].

En otro artículo se analiza la importancia de la aprobación de créditos, especialmente para tarjetas de crédito, en la economía moderna. Destaca el reto al que se enfrentan los prestamistas a la hora de predecir si un consumidor representa un riesgo crediticio aceptable y si se le debe conceder un crédito. El artículo utiliza modelos de aprendizaje automático para predecir si una solicitud de tarjeta de crédito será aprobada o no. Se utilizaron tres algoritmos: *Gradient Boosting Classifier, Support Vector Classifier y Adaboost Classifier*. El rendimiento de estos algoritmos se evaluó utilizando medidas de precisión como *F1-score, Precision* y *Recall*. La mayor precisión, del 90%, se obtuvo con el clasificador *Gradient Boosting*. El artículo también incluye un análisis exploratorio de datos para identificar las variables más importantes para la aprobación de tarjetas de crédito [7].

III. OBJETIVOS

A. Objetivo general

Desarrollar un modelo de predicción para aprobación de tarjetas de crédito, utilizando técnicas de aprendizaje automático, con el fin de mejorar la precisión y eficiencia en el proceso de entrega de créditos por entidades bancarias.

B. Objetivos específicos

- Procesar los datos obtenidos por la plataforma *Kaggle*, con el fin de limpiar y detectar anomalías que puedan afectar el rendimiento del modelo de predicción.
- Implementar diferentes técnicas de sobremuestreo de datos para mitigar el problema de desbalance con el que cuenta la base de datos original.
- Generar diferentes modelos de aprendizaje automático, como *Regresión Logística, Árboles de decisión, Máquinas de soporte vectorial, Árboles Aleatorios*, entre otros, para determinar cuál tiene el mejor rendimiento en términos de precisión, sensibilidad y pérdida.
- Evaluar los modelos de clasificación para elegir el mejor modelo para la aprobación o no de la tarjeta de crédito en los clientes.

IV. DATOS

Las bases de datos para el desarrollo de este documento, se obtienen del conjunto de datos *Credit Card Approval Prediction* , proporcionado por la plataforma *Kaggle* [1].

A. Base de datos "application_record"

Contiene la información personal de los solicitantes y cuenta con 438.557 registros, a continuación, la descripción del conjunto de datos.

TABLA I
CARACTERÍSTICAS BASE DE DATOS APPLICATION_RECORD

Nombre	Tipo	Descripción
ID	integer	Número de cliente
CODE_GENDER	object	Género
FLAG_OWN_CAR	object	Tiene carro
FLAG_OWN_REALTY	object	Tiene Propiedad
CNT_CHILDREN	integer	Número de hijos
AMT_INCOME_TOTAL	float	Ingresos anuales
NAME_INCOME_TYPE	object	Categoría de ingresos
NAME_EDUCATION_TYPE	object	Nivel de educación
NAME_FAMILY_STATUS	object	Estado civil
NAME_HOUSING_TYPE	object	Forma de Vivir
DAYS_BIRTH	integer	Edad en días
DAYS_EMPLOYED	integer	Fecha inicio del empleo
FLAG_MOBIL	integer	Tiene teléfono celular
FLAG_WORK_PHONE	integer	Tiene teléfono de trabajo
FLAG_PHONE	integer	Tiene teléfono
FLAG_EMAIL	integer	Tiene correo electrónico

OCCUPATION_TYPE	object	Ocupación
CTN_FAM_MENBERS	float	Miembros de la familia

Nota: descripción y tipo de datos de las características de la base de datos.

B. Base de datos "credit_record"

Registra los comportamientos de los usuarios de la tarjeta de crédito y cuenta con 1.048.575 registros, a continuación, la descripción del conjunto de datos.

TABLA II

CARACTERÍSTICAS BASE DE DATOS CREDIT_RECORD

Nombre	**Tipo**	**Descripción**
ID	integer	Número de cliente
MONTHS_BALANCE	integer	Mes de registro. El mes de los datos extraídos es el punto de partida, 0 es el mes actual, -1 es el mes anterior, y así sucesivamente
STATUS	object	Estado. 0: 1-29 días de atraso, 1: 30-59 días de atraso, 2: 60-89 días de atraso, 3: 90-119 días de atraso, 4: 120-149 días de atraso, 5: Deudas atrasadas o incobrables, canceladas por más de 150 días, C: cancelado ese mes, X: Sin préstamo en el mes

Nota: descripción y tipo de datos de las características de la base de datos.

V. METODOLOGÍA

Para abordar el problema descrito en este documento, se define la siguiente metodología (Fig. 1).

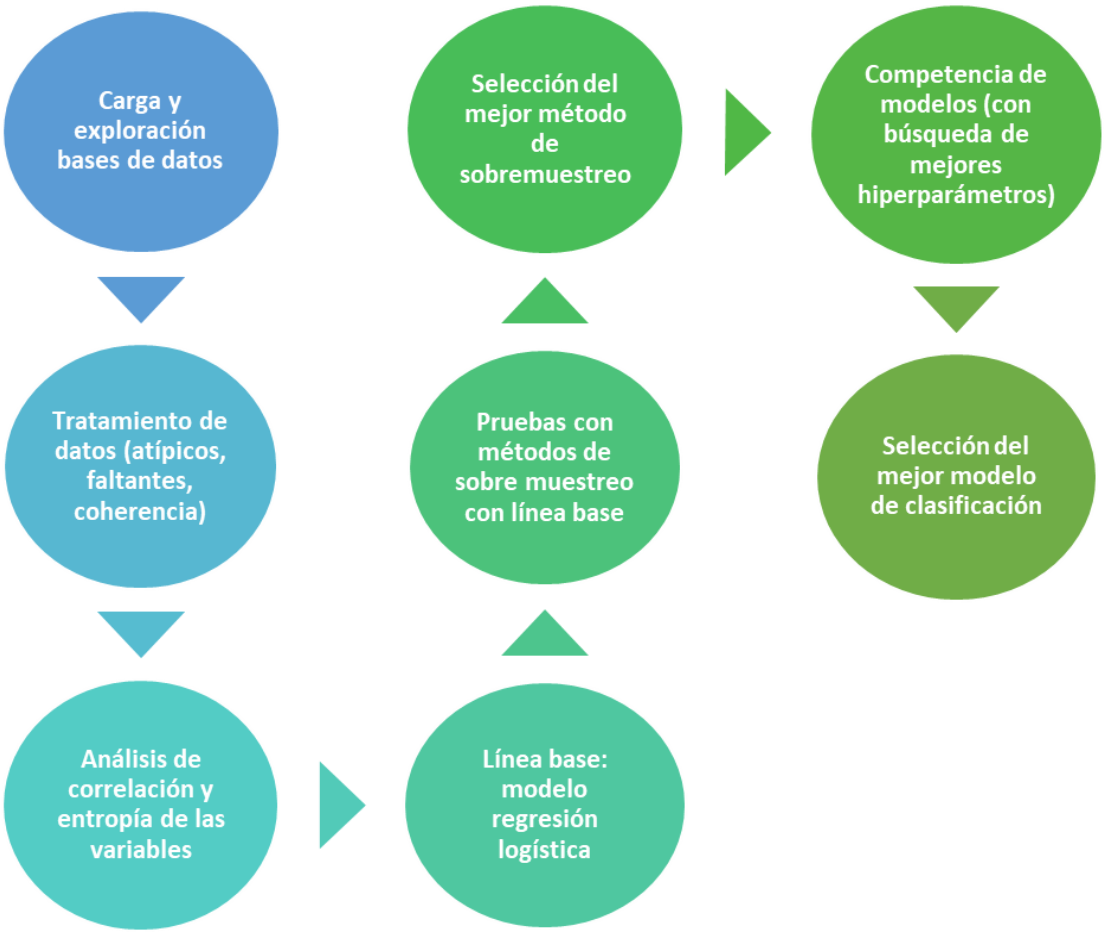


Fig. 1. Flujo de metodología

Nota: pipeline del proceso de selección del mejor modelo de clasificación.

A. Carga y exploración de bases de datos

Los datos se cargan y se almacenan en un par de objetos Data Frame, correspondientes a credit_record.csv y application_record.csv respectivamente. Una vez cargada la base de datos, se procede a analizar la cantidad de registros, búsqueda de registros duplicados y

faltantes, se realiza una exploración de caracteres especiales y se analiza el balance de las categorías en cada una de las variables para ambos DataFrame.

B. Tratamiento de datos

Del paso anterior se obtuvo que la variable *FLAG_MOBILE* del conjunto de datos *application_record,* tiene el valor 1 para todos los registros, por lo que se decide eliminarla dado que no hay varianza en dicha característica.

La característica OCCUPATION_TYPE tiene 134.203 registros faltantes pero dado que se asume que es una característica que aporta información al modelo, se imputan estos registros usando la moda (Laborers).

Se transforma el DataFrame *credit_record* para obtener la variable de salida, de manera que para los clientes que en su historial crediticio lleven una deuda superior a 60 días, se les asignará la etiqueta de NO_APTO (1) y para cualquier otro caso APTO (0).

Se codifican las variables categóricas, usando el método *getDummies*[1] de la librería Pandas, que utiliza la codificación *One-Hot*. Se identifican y eliminan 1.820 datos atípicos, utilizando el algoritmo *Local Outlier Factor*[2] *(LOF)*. Finalmente, se realiza una fusión de los dos DataFrame tipo "*inner*" obteniendo así un nuevo dataset de 36.457 registros.

C. Análisis de correlación y entropía de las variables

Se implementa una matriz de correlación sobre los datos, para identificar características que puedan ser redundantes y debido a la alta correlación obtenida entre las variables "*CNT_FAM_MEMBERS*" y "*CNT_CHILDREN*" (0.88) y entre las variables "*NAME_INCOME_TYPE_Pensione*r" y "*DAYS_EMPLOYED*" (0.99) se decide eliminar las características "*CNT_CHILDREN*" y "*NAME_INCOME_TYPE_Pensioner*".

Se realiza un análisis de la entropía relativa sobre las variables numéricas y una selección de características por información mutua pero no se obtiene eliminar ninguna otra característica, dado a que no se evidencia información mutua alta ante las variables.

[1] https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

D. Línea base

Como línea base se entrena un modelo de *Regresión Logística*[3] con el 80% de los datos originales (con desbalance) y se prueba el modelo con el 20% de los datos restantes. Para validar el rendimiento del modelo, se calculan las métricas de *Accuracy*, *Balance accuracy score, Log Loss, Recall* y *F1-score*.

E. Pruebas de métodos de sobremuestreo

Se realizan pruebas sobre un modelo de *Regresión Logística* configurado con los hiperparámetros de la línea base y entrenado con los datos tratados con cada uno de los métodos de sobremuestreo, *Random Over Sampler*[4], *SMOTE*[5], *SMOTEN*[6], *KMeansSMOTE*[7], *BoderlineSmote*[8] *y SMOTENC*[9]. Se obtienen las métricas de rendimiento calculadas en la línea base, para cada uno de los modelos.

F. Selección de método de sobremuestreo

Basado en las métricas de rendimiento calculadas en el paso anterior se elige el método de sobremuestreo con mejores resultados. El método con mejores métricas para la base de datos es *SMOTENC***.**

G. Competencia de modelos

Con los datos sobremuestreados con el método *SMOTENC*, se realiza la búsqueda de hiperparámetros para los modelos de *Máquina de Soporte Vectorial*[10]*, Árboles de decisión*[11]*,*

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[4] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html
[5] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
[6] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTEN.html
[7] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.KMeansSMOTE.html
[8] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.BorderlineSMOTE.html
[9] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTENC.html
[10] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[11] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

AdaBoost[12] *y Árboles Aleatorios*[13]. Se entrena cada uno de los modelos configurado con sus mejores hiperparámetros y con la base de datos balanceada; se prueba el modelo con los datos originales (desbalanceados) para obtener las métricas de rendimiento calculadas para la línea base.

H. Selección del modelo de clasificación

Del paso anterior se analiza el modelo con mejores resultados en sus métricas y se evalúa si el modelo cumple con la métrica definida por el negocio (*accuracy* superior al 75%). El modelo con mejor resultado, entrenado con los datos balanceados, es *Árboles Aleatorios*. Finalmente, es guardado con extensión .joblib[14] para usarlo en futuras predicciones de etiquetas.

[12] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
[13] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[14] https://joblib.readthedocs.io/en/latest/generated/joblib.dump.html

VI. RESULTADOS Y ANÁLISIS

A. Línea base

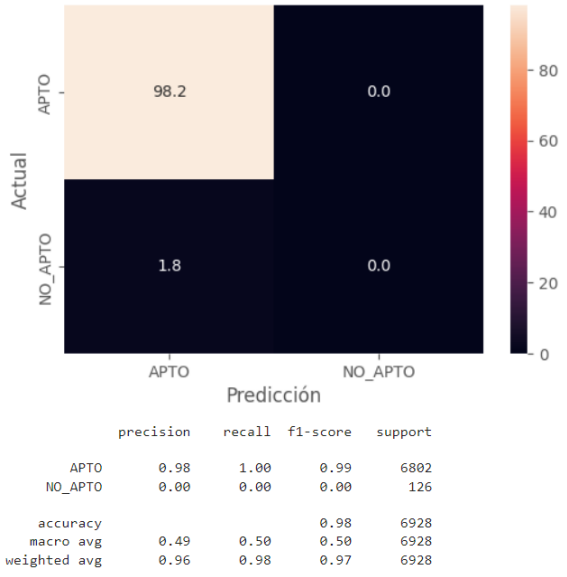
Como primera iteración se realizó la búsqueda de los mejores hiperparámetros para una regresión logística, entrenada con el 80% de los datos originales y probada con el 20% restante. Los resultados de este modelo se pueden visualizar en la siguiente tabla y en las Fig.2 y Fig.3.

TABLA III

RESULTADOS REGRESIÓN LOGÍSTICA - LÍNEA BASE

Hiperparámetros	**Accuracy (%)**	**Balance Accuracy Score (%)**	**Log Loss**
C= 0,001 *class_weight*= None *multi_class*= ovr *solver*= liblinear	98,18	50	0,1021

Nota: resultados del modelo de regresión logística, entrenado con los datos originales.



	precision	recall	f1-score	support
APTO	0.98	1.00	0.99	6802
NO_APTO	0.00	0.00	0.00	126
accuracy			0.98	6928
macro avg	0.49	0.50	0.50	6928
weighted avg	0.96	0.98	0.97	6928

Fig. 2. Matriz de confusión línea base

Nota: porcentaje de predicciones correctas e incorrectas del modelo línea base regresión logística.

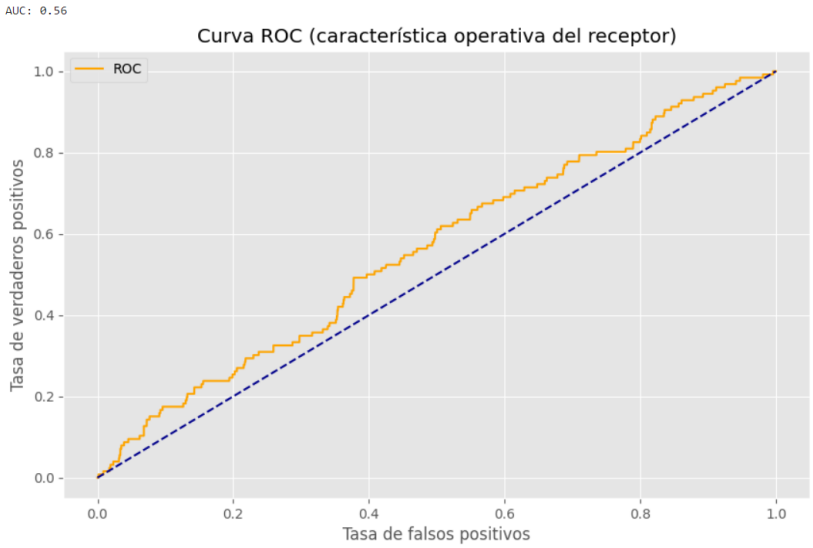


Fig. 3. Gráfica de sensibilidad contra especificidad del modelo línea base

Dado el gran desbalance en las etiquetas de la base de datos (*APTO*: 35841 y *NO_APTO*: 616) se puede evidenciar un sobreajuste del modelo, ya que es muy bueno prediciendo las etiquetas de *APTO* y no es capaz de predecir correctamente los clientes con etiqueta *NO_APTO*.

Para tratar este desbalance, se implementan varios métodos de sobremuestreo de datos y se entrena un modelo de *Regresión Logística* con sus mejores hiperparámetros. El detalle de los resultados se registra en la TABLA IV y en las figuras Fig. 4 hasta Fig. 9.

TABLA IV

RESULTADOS MÉTODOS DE SOBREMUESTREO

Método	**Hiperparámetros**	**Accuracy (%)**	**Balance Accuracy Score (%)**	**Log Loss**
ROS	*C*= 1000 *class_weight*= balanced *multi_class*= multinomial *solver*= newton-cg	59,38	58,4	0,6706
SMOTEN	*C*= 0.1 *class_weight*= None *multi_class*= multinomial *solver*= newton-cg	75,48	51,64	0,4432
KmeansSMOTE	*C*= 10 *class_weight*= balanced *multi_class*= multinomial *solver*= newton-cg	79,04	53,79	0,4857

BorderlineSMOTE	*C*= 100 *class_weight*= None *multi_class*= ovr *solver*= newton-cg	84,8	52,58	0,3972
SMOTENC	*C*= 1 *class_weight*= None *multi_class*= ovr *solver*= newton-cg	86,36	51,43	0,3726
SMOTE	*C*= 1000 *class_weight*= None *multi_class*= multinomial *solver*= newton-cg	85,15	51,24	0,3806

Nota: resultados del modelo de Regresión Logística, entrenado con los datos con los métodos de sobremuestreo y probados con los datos originales.

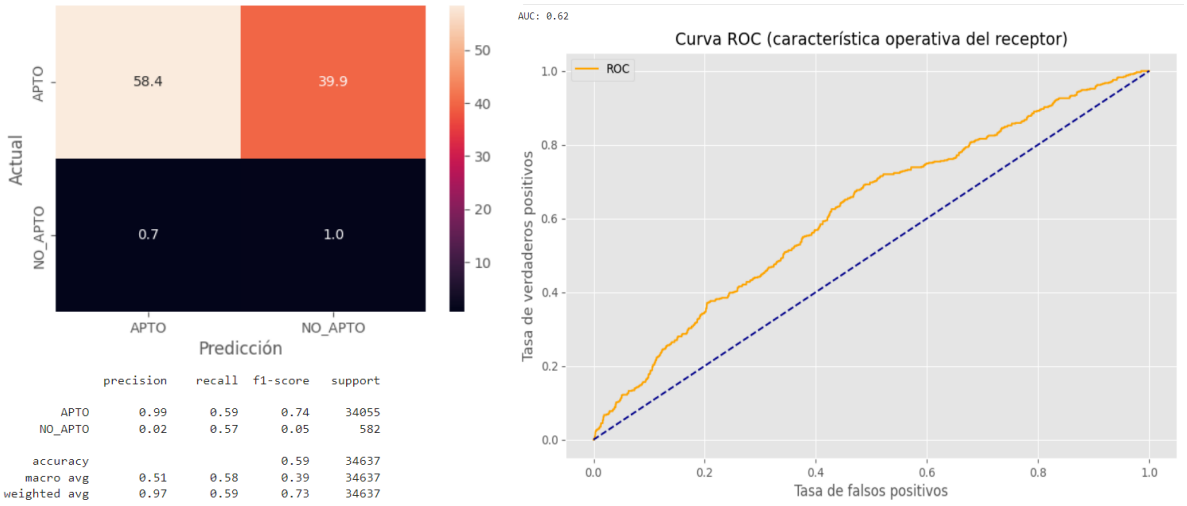
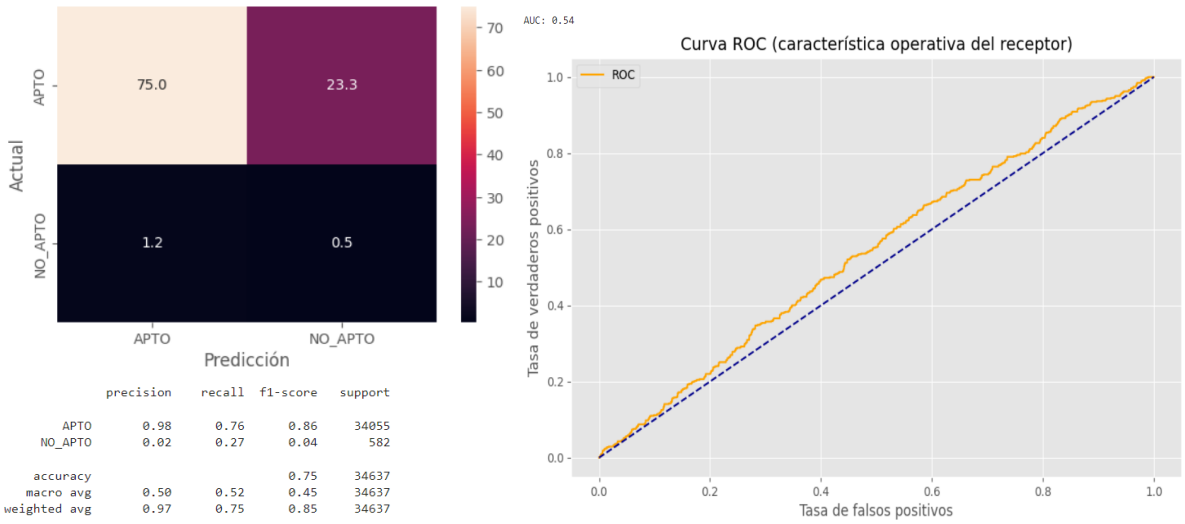


Fig. 4. Matriz de confusión y ROC del modelo con método ROS

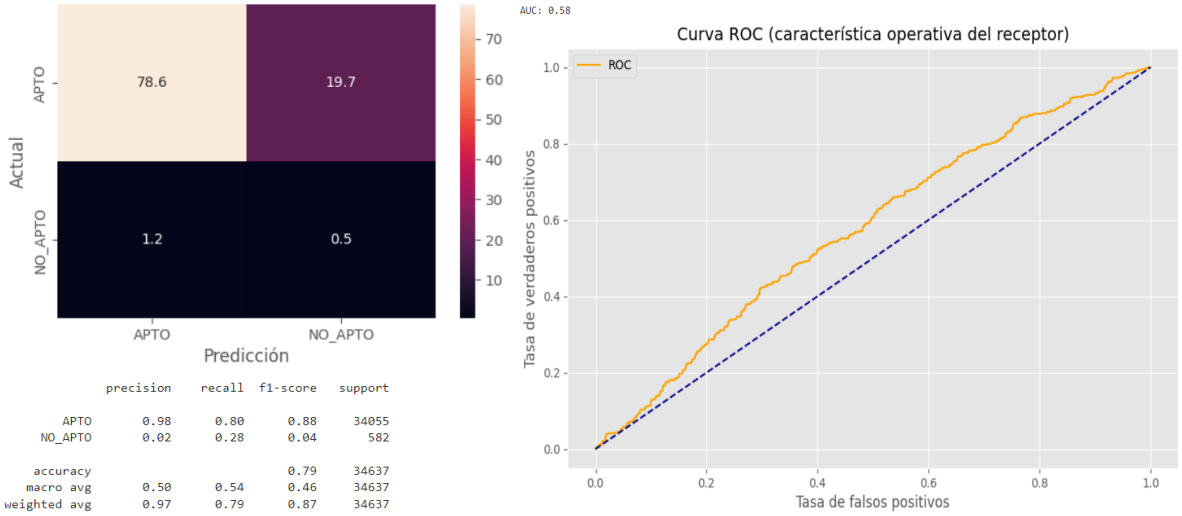
Nota: matriz de confusión y gráfica de sensibilidad contra especificidad del modelo con datos sobremuestreados con el método ROS.



	precision	recall	f1-score	support
APTO	0.98	0.76	0.86	34055
NO_APTO	0.02	0.27	0.04	582
accuracy			0.75	34637
macro avg	0.50	0.52	0.45	34637
weighted avg	0.97	0.75	0.85	34637

Fig. 5. Matriz de confusión y ROC del modelo con método SMOTEN

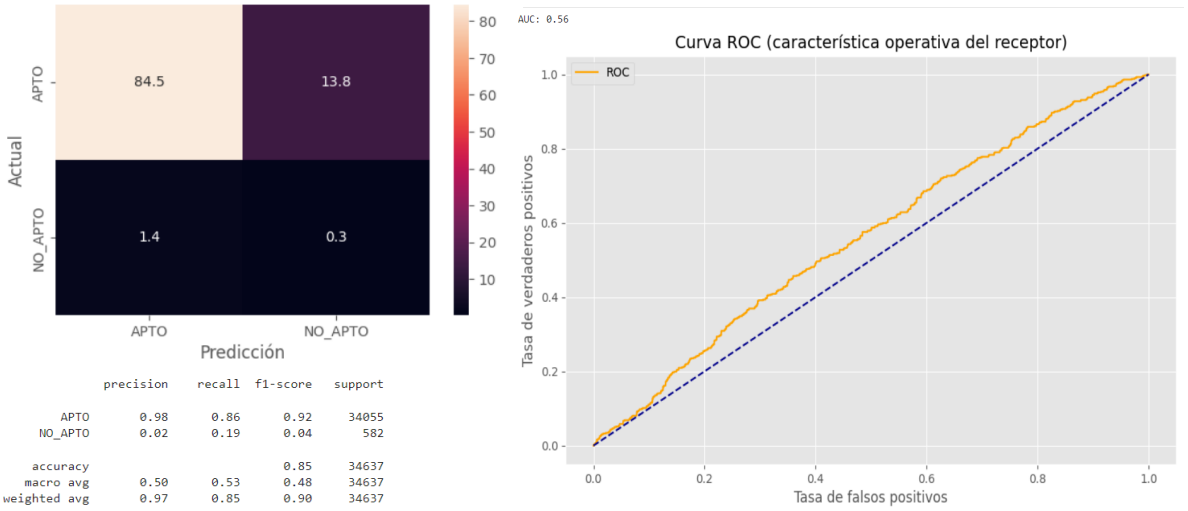
Nota: matriz de confusión y gráfica de sensibilidad contra especificidad del modelo con datos sobremuestreados con el método *SMOTEN*.



	precision	recall	f1-score	support
APTO	0.98	0.80	0.88	34055
NO_APTO	0.02	0.28	0.04	582
accuracy			0.79	34637
macro avg	0.50	0.54	0.46	34637
weighted avg	0.97	0.79	0.87	34637

Fig. 6. Matriz de confusión y ROC del modelo con método KmeansSMOTE

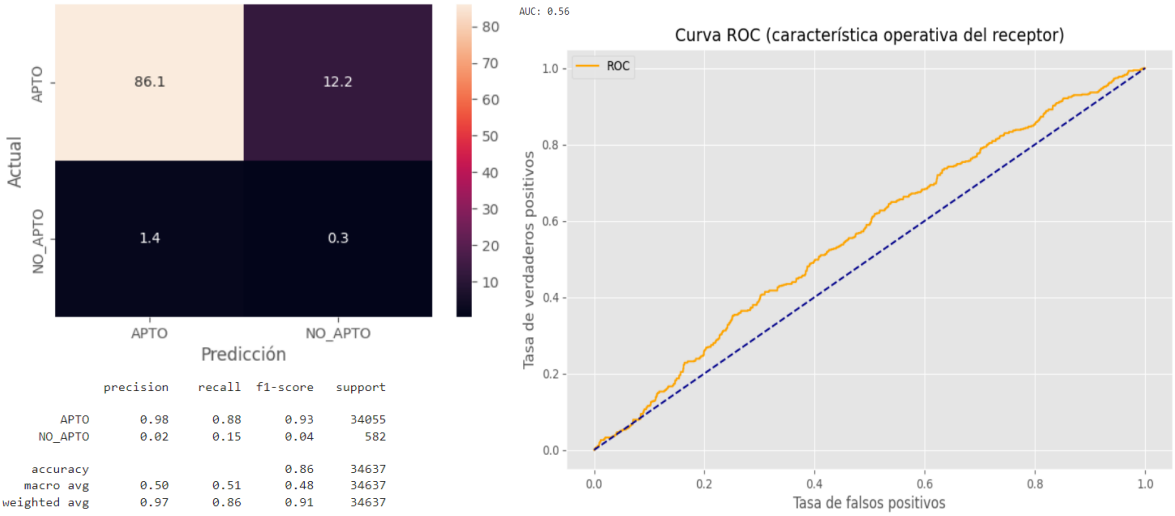
Nota: matriz de confusión y gráfica de sensibilidad contra especificidad del modelo con datos sobremuestreados con el método *KmeansSMOTE*.



	precision	recall	f1-score	support
APTO	0.98	0.86	0.92	34055
NO_APTO	0.02	0.19	0.04	582
accuracy			0.85	34637
macro avg	0.50	0.53	0.48	34637
weighted avg	0.97	0.85	0.90	34637

Fig. 7. Matriz de confusión y ROC del modelo con método BorderlineSMOTE

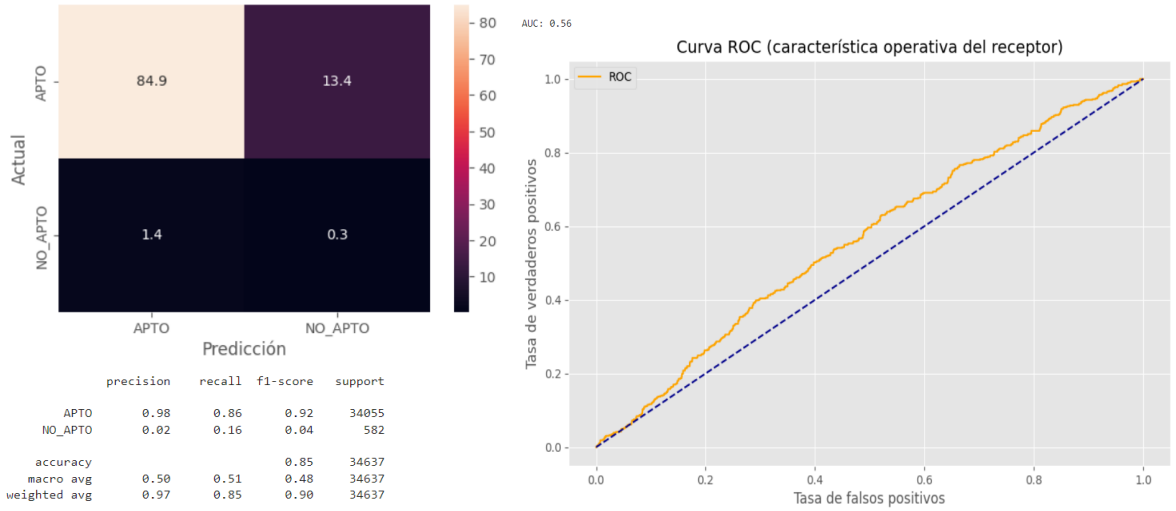
Nota: matriz de confusión y gráfica de sensibilidad contra especificidad del modelo con datos sobremuestreados con el método *BorderlineSMOTE*.



	precision	recall	f1-score	support
APTO	0.98	0.88	0.93	34055
NO_APTO	0.02	0.15	0.04	582
accuracy			0.86	34637
macro avg	0.50	0.51	0.48	34637
weighted avg	0.97	0.86	0.91	34637

Fig. 8. Matriz de confusión y ROC del modelo con método SMOTENC

Nota: matriz de confusión y gráfica de sensibilidad contra especificidad del modelo con datos sobremuestreados con el método *SMOTENC*.



	precision	recall	f1-score	support
APTO	0.98	0.86	0.92	34055
NO_APTO	0.02	0.16	0.04	582
accuracy			0.85	34637
macro avg	0.50	0.51	0.48	34637
weighted avg	0.97	0.85	0.90	34637

Fig. 9. Matriz de confusión y ROC del modelo con método SMOTE

Nota: matriz de confusión y gráfica de sensibilidad contra especificidad del modelo con datos sobremuestreados con el método *SMOTE*.

Con base en los resultados anteriores, se seleccionó *SMOTENC* como método de remuestreo porque tiene el mejor porcentaje de *Accuracy* (86.36%) y el valor menor de *Log Loss* (0.3726).

B. Competencia de modelos

Una vez seleccionado el método de remuestreo (SMOTENC), se decide realizar la búsqueda del mejor modelo, para la base de datos, entre los siguientes candidatos: Árboles de decisión, AdaBoost, Máquina de Soporte Vectorial y Árboles Aleatorios. Para cada modelo se realizó la búsqueda de sus mejores hiperparámetros, se entrenó el modelo con todos los datos remuestreados y se probó con todos los datos originales. Los resultados se indican en la siguiente tabla:

TABLA V

RESULTADOS DE COMPETENCIA DE MODELOS

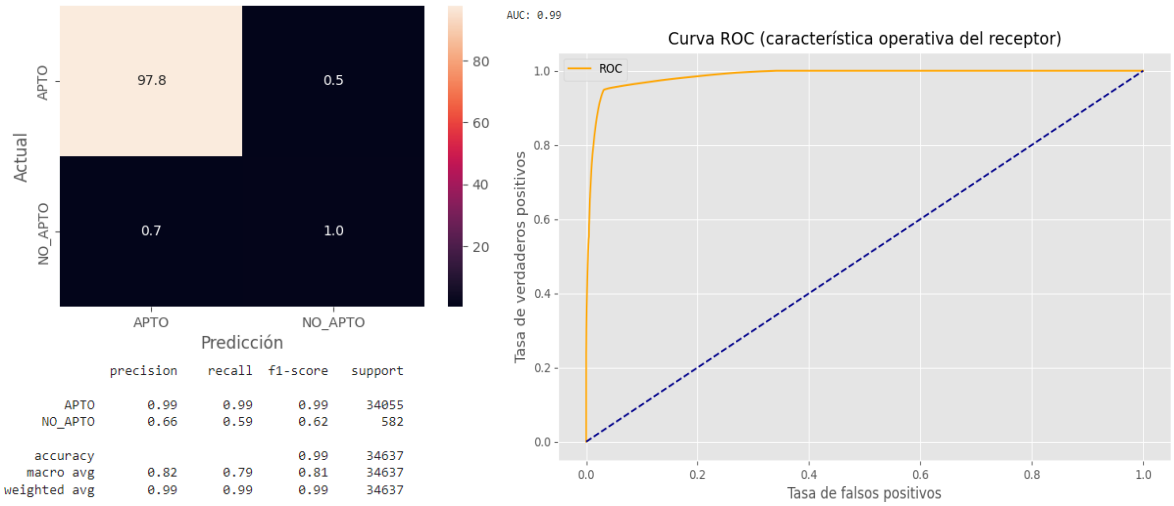
Modelo	**Hiperparámetros**	**Accuracy (%)**	**Balance Accuracy Score (%)**	**Log loss**
Árboles de decisión	*ccp_alpha*: 0.0 *Criterion*: 'gini' *Max_depth*: 41 *Número de nodos terminales*: 1249 *scoring*: 'f1'	98,79	79,2	0,036
Árboles de decisión	*ccp_alpha*: 0.0 *Criterion*: 'gini' *Max_depth*: 3 *Número de nodos terminales*: 8 *scoring*: 'f1'	72,72	53,02	0,6068
Árboles de decisión	*Alpha* : 0.0 *Criterion* : entropy *Max_depth* : 40 *Profundidad del árbol*: 33 *Número de nodos terminales*: 1318 *scoring*: 'accuracy'	98,81	79,97	0,0308
AdaBoost	*learning_rate*: 0.1, *n_estimators*: 200 *scoring*: accuracy	92,72	58,21	0,6512
Máquina de soporte vectorial	*C*: 0.1	87,09	51,3	-
Árboles Aleatorios	*criterion*: 'gini' *max_depth*: None *n_estimators*: 150 *n_jobs*: -1 *class_weight*: balanced	**98,8**	**81,15**	**0,0352**

Nota: resultados de los modelos entrenados con los datos sobremuestreados y probado con los datos originales.

Teniendo en cuenta sólo las métricas de *accuracy*, *balance accuracy score* y *log loss*, es evidente que los modelos implementados obtuvieron mejores resultados que los del modelo línea base siendo el mejor *Árboles Aleatorios*.

Los resultados más bajos se obtuvieron primero con cierta configuración de *árboles de decisión* en la cual se limitó el parámetro '*max_depth*' a valores pequeños (2, 4 y 8) y segundo con *máquina de soporte vectorial*.

A continuación, se indican las gráficas de matriz de confusión y curva *ROC* resultantes de los modelos con sus mejores hiperparámetros, se indica también la importancia de las características de los modelos que tienen el método '*feature_importances*' de *Sklearn*.



	precision	recall	f1-score	support
APTO	0.99	0.99	0.99	34055
NO_APTO	0.66	0.59	0.62	582
accuracy			0.99	34637
macro avg	0.82	0.79	0.81	34637
weighted avg	0.99	0.99	0.99	34637

Fig. 10. Matriz de confusión y ROC de Árbol de decisión (mejores hiperparámetros)

Nota: matriz de confusión y gráfica de sensibilidad contra especificidad de Árbol de decisión (mejores hiperparámetros).

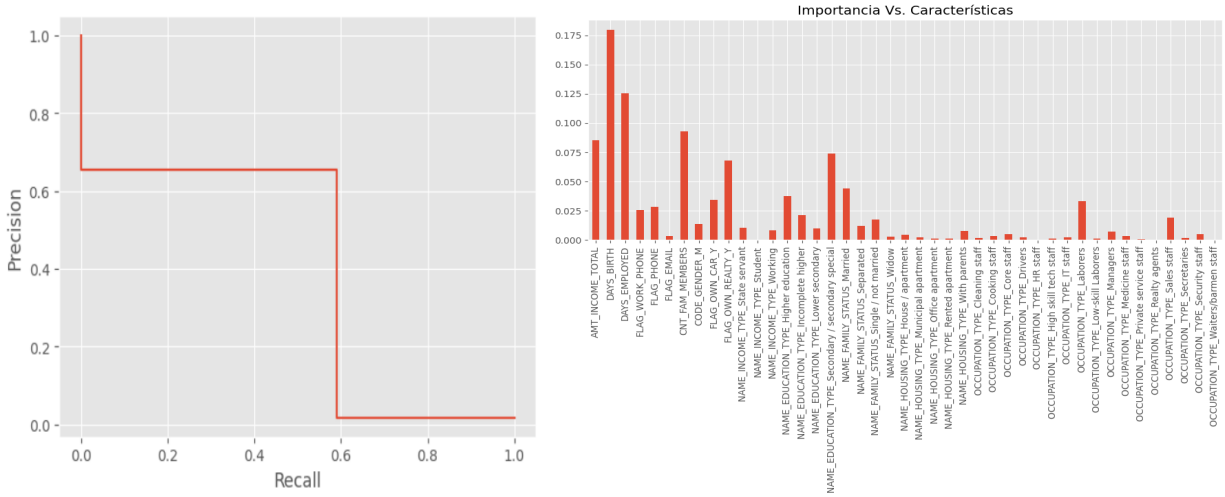
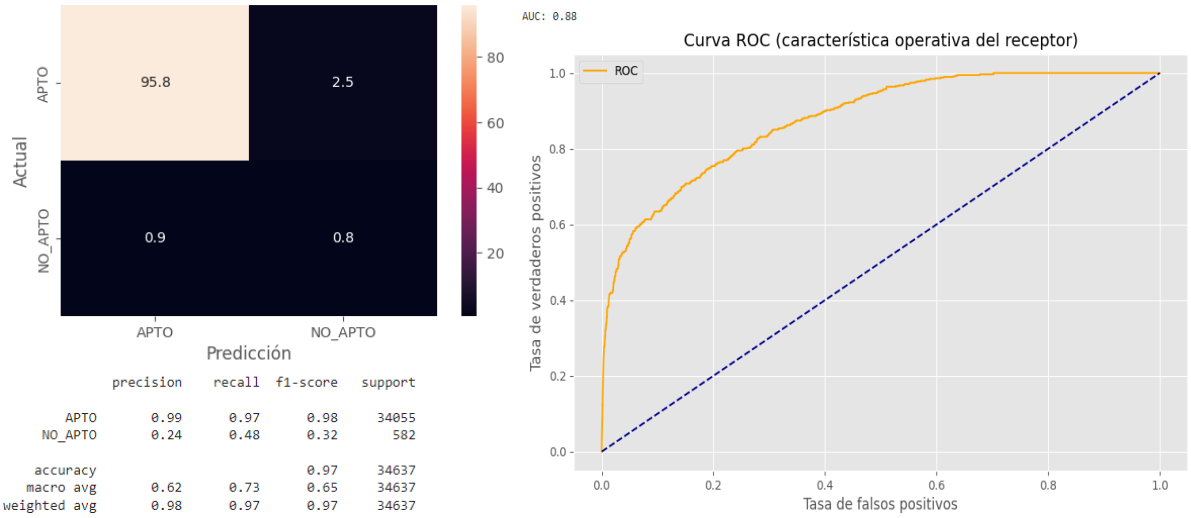


Fig. 11. Precision Recall Curve e importancia de características de Árbol de decisión

Nota: gráfica de precisión y recuperación del modelo e importancia de características de Árbol de decisión.



	precision	recall	f1-score	support
APTO	0.99	0.97	0.98	34055
NO_APTO	0.24	0.48	0.32	582
accuracy			0.97	34637
macro avg	0.62	0.73	0.65	34637
weighted avg	0.98	0.97	0.97	34637

Fig. 12. Matriz de confusión y ROC de AdaBoost

Nota: matriz de confusión y gráfica de sensibilidad contra especificidad de AdaBoost.

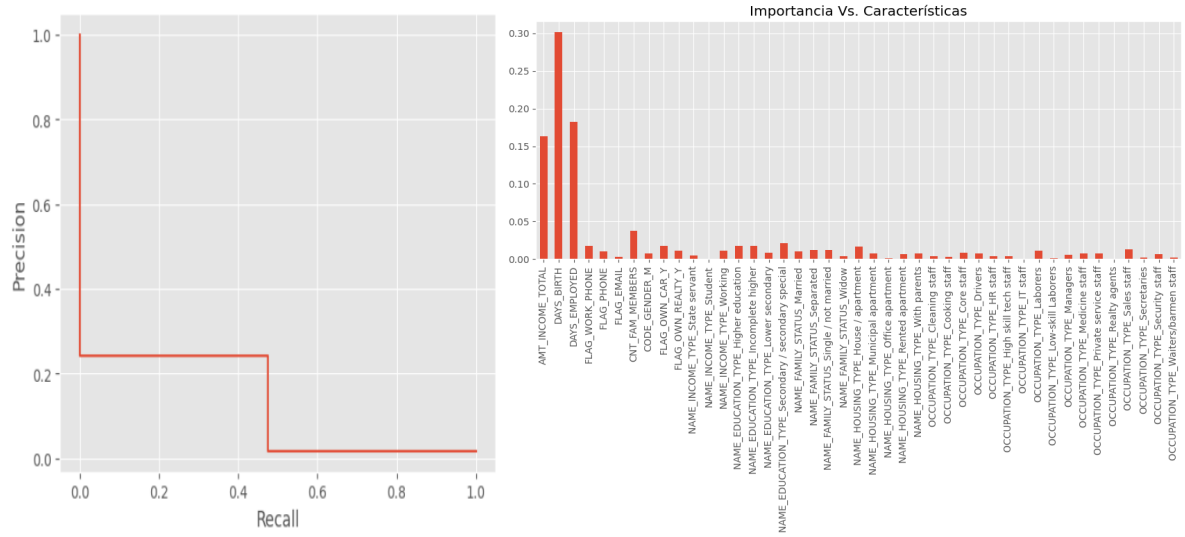
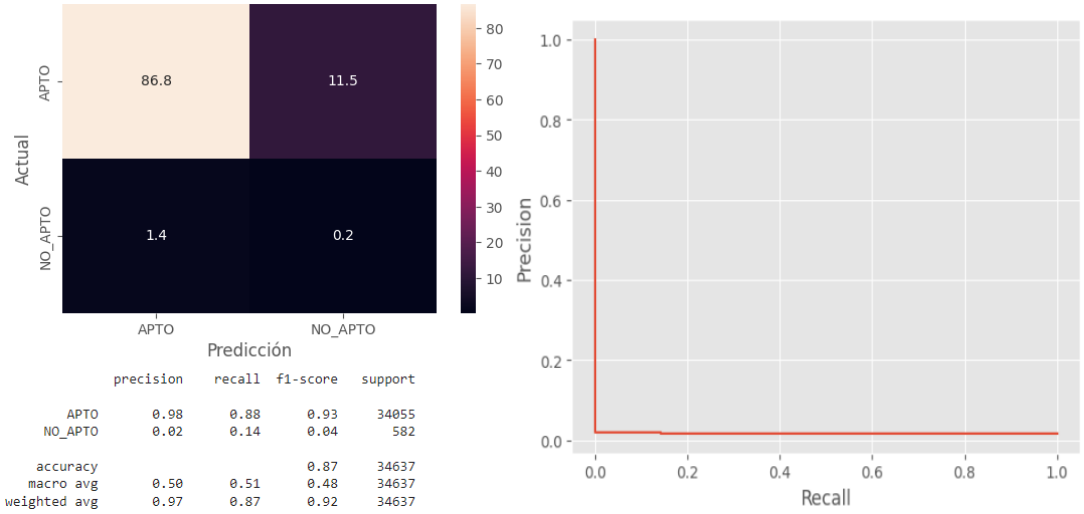


Fig. 13. Precision Recall Curve e importancia de características AdaBoost

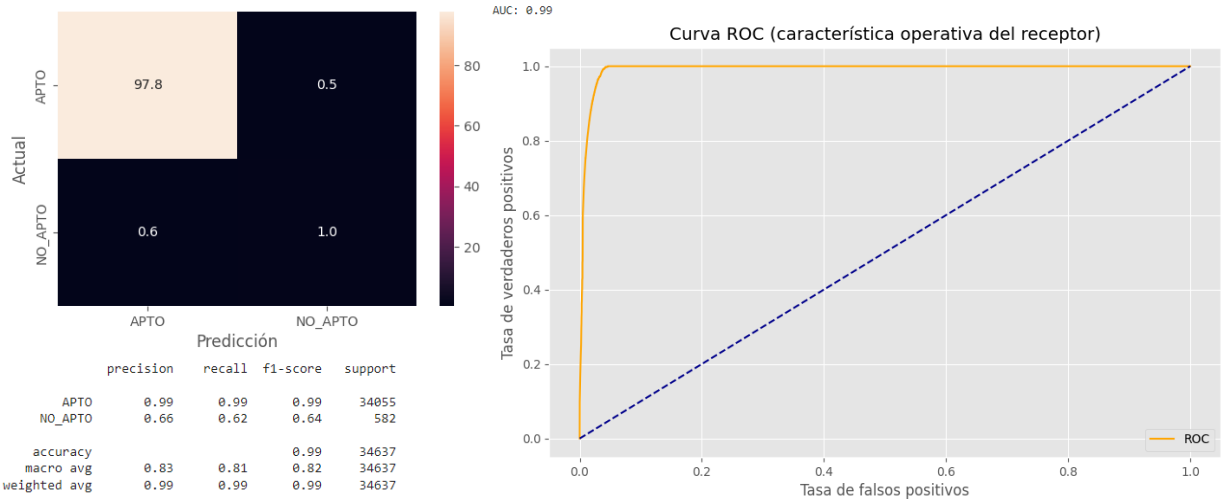
Nota: gráfica de precisión y recuperación del modelo e importancia de características de AdaBoost.



	precision	recall	f1-score	support
APTO	0.98	0.88	0.93	34055
NO_APTO	0.02	0.14	0.04	582
accuracy			0.87	34637
macro avg	0.50	0.51	0.48	34637
weighted avg	0.97	0.87	0.92	34637

Fig. 14. Matriz de confusión y Precision Recall Curve de Máquina de Soporte Vectorial

Nota: matriz de confusión y gráfica de precisión y recuperación de *Máquina de Soporte Vectorial*.



	precision	recall	f1-score	support
APTO	0.99	0.99	0.99	34055
NO_APTO	0.66	0.62	0.64	582
accuracy			0.99	34637
macro avg	0.83	0.81	0.82	34637
weighted avg	0.99	0.99	0.99	34637

Fig. 15. Matriz de confusión y ROC de Árboles Aleatorios

Nota: matriz de confusión y gráfica de sensibilidad contra especificidad de Árboles Aleatorios.

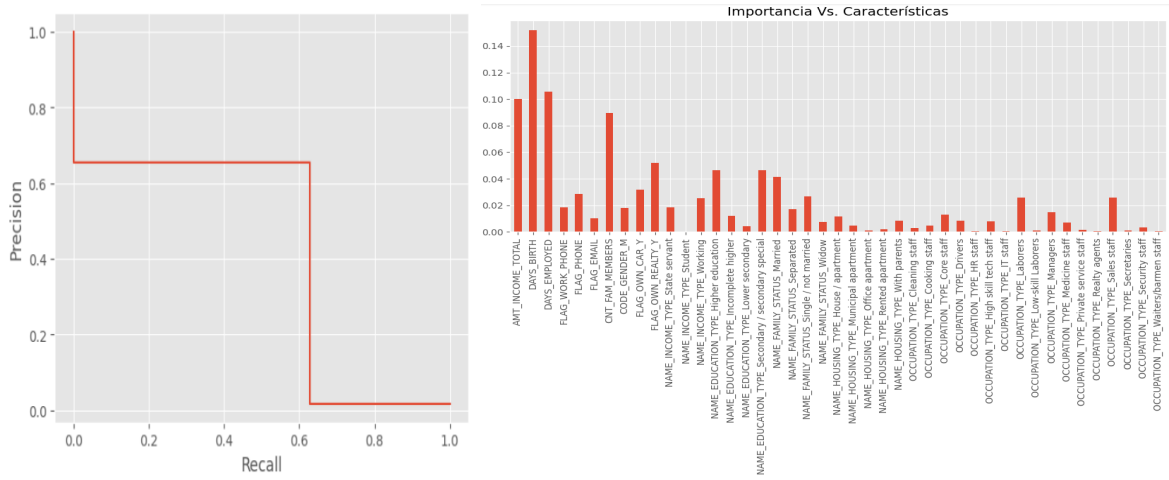


Fig. 16. Precision Recall Curve e importancia de características Árboles Aleatorios

Nota: gráfica de precisión y recuperación del modelo e importancia de características de Árboles Aleatorios.

De las imágenes anteriores, se puede observar que los resultados obtenidos son mejores que los de la línea base. Las gráficas de importancia de características varían dependiendo el modelo utilizado pero se ve una tendencia en tres características principales que son: *DAYS_BIRTH, AMT_INCOME_TOTAL* y *DAYS_EMPLOYED*. El modelo *Árboles Aleatorios* indica una mayor distribución en la importancia de las características. Las métricas de *precision*, *recall* y *F1-score*, indican notable mejoría y presentan un mejor puntaje para los casos de solicitantes aptos sobre los no aptos.

Los mejores resultados se obtuvieron con el modelo ***Árboles Aleatorios*** con un *accuracy* de 98.8%, un *balance accuracy score* de 81.15% y *log loss* de 0.0352. El modelo obtuvo la mejor matriz de confusión, curva *ROC* con 0.99 *AUC* y la mejor gráfica de *precision recall Curve*.

De los resultados anteriores y teniendo en cuenta que la métrica definida por el negocio es que el modelo supere un umbral del 75% en el *accuracy*, se puede afirmar que el modelo cumple con el objetivo y puede ser implementado en un ambiente productivo como un aplicativo web o móvil, para que pueda ser utilizado como herramienta en la clasificación de clientes en el sector financiero.

El código del tratamiento de los datos y los modelos implementados en este documento, puede encontrarse en el notebook alojado en un repositorio de GitHub.

VII. CONCLUSIONES

Revisar los datasets y realizar la exploración de los datos permitió tratar algunos problemas como la falta de datos y valores atípicos, además se pudo realizar un pre procesamiento con el fin de adecuar los datos para los modelos y obtener una variable objetivo.

Tratar el desbalance de las etiquetas de la base de datos fue de gran importancia ya que los primeros resultados indicaron un sobre ajuste en los modelos y las métricas utilizadas no eran las esperadas, la solución fue implementar métodos de sobre muestreo donde el mejor resultado se obtuvo con **SMOTENC**.

Después de implementar diferentes modelos y realizar la búsqueda de los mejores hiperparámetros para cada modelo, se concluye que los modelos basados en árboles de decisión son los que mejores resultados obtuvieron siendo ***Árboles Aleatorios*** el mejor para la base de datos utilizada.

Debido a la naturaleza de los datos y principalmente al desbalance en sus etiquetas, fue necesario implementar otras métricas que permitieran medir los resultados de los modelos, algunas de ellas fueron: *Balance accuracy score* , *Log Loss* y *Precision Recall Curve*.

Se concluye que el modelo ***Árboles Aleatorios*** cumple con los objetivos propuestos por el negocio y se puede utilizar para una implementación en la industria.

VIII. RECOMENDACIONES Y TRABAJOS FUTUROS

Se recomienda realizar una reducción de dimensionalidad con diferentes métodos como Principal Component Analysis (PCA) o Singular Value Decomposition (SVD), con el objetivo de reducir costos computacionales en los entrenamientos de los diferentes modelos.

Similar a la búsqueda de hiperparámetros, se recomienda implementar todos los métodos de sobremuestreo para cada uno de los modelos y seleccionar el que arroje mejores resultados.

Para futuros trabajos, se propone adicionar a la competencia de modelos otros algoritmos como: Gradient Boosting Classifier, Extreme Gradient Boosting (XGBoost) y redes neuronales.

REFERENCIAS

[1] "Credit card approval prediction," Kaggle, 24-Mar-2020. [En línea]. Disponible en: https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction.

[2] D. Bhalla, "CREDIT RISK: VINTAGE ANALYSIS," 09 2019. [En línea]. Disponible en: https://www.listendata.com/2019/09/credit-risk-vintage-analysis.html.

[3] O. J. Leong and M. Jayabalan, "A comparative study on credit card default risk predictive model," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 8, pp. 3591–3595, 2019.

[4] I. Sugiyarto, B. Sudarsono, and U. Fadillah, "Performance comparison of data mining algorithm to predict approval of credit card," *SinkrOn*, vol. 4, no. 1, p. 149, 2019.

[5] T. M. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201173–201198, 2020.

[6] M. G. Kibria y M. Sevkli, "Application of Deep Learning for Credit Card Approval: A Comparison with Two Machine Learning Techniques", International Journal of Machine Learning and Computing, vol. 11, núm. 4, pp. 412-417, 2021.

[7] N. Dalsania, D. Punatar, and D. Kothari, "Credit card approval prediction using classification algorithms," International Journal for Research in Applied Science and Engineering Technology, vol. 10, no. 11, pp. 507–514, 2022.