RESULT REPORT

Kristhian Santiago Palomino Fajardo

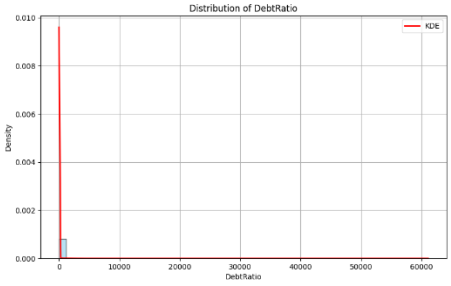
CREDIT SCORING CHALLENGE

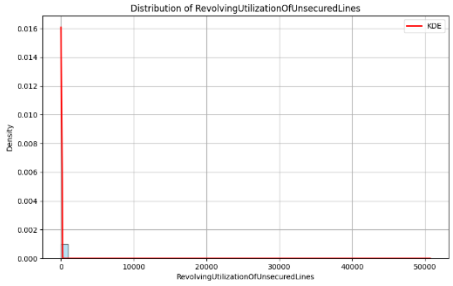
Process Overview

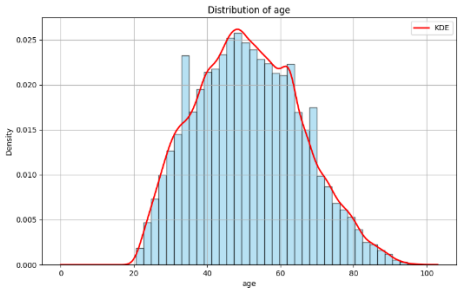
Exploratory Analysis and Preprocessing

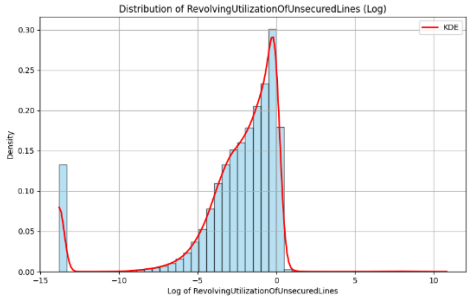
The dataset cs-training.csv was used, containing 150,000 client records with credit information and delinquency indicators. Approximately 29,731 records with missing *MonthlyIncome* were dropped. This decision was based on the observation that most of those clients did not exhibit a clear income history, and their distribution in the target variable (*SeriousDlqin2yrs*) did not indicate a particularly high-risk group. Consequently, reliable imputation could not be performed without introducing bias. For the column *NumberOfDependents*, missing values (around 2% of the sample) were imputed with 0, as it is reasonable to assume that lack of information implies the absence of dependents and this does not significantly alter the distribution.

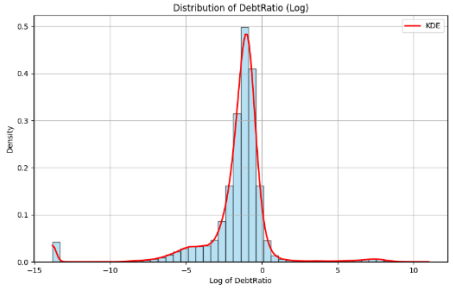
The following visualizations were generated (all saved in the **plots** folder:
Key variable distributions: MonthlyIncome, DebtRatio, RevolvingUtilizationOfUnsecuredLines, and age (e.g., plots/MonthlyIncome.png, plots/MonthlyIncome_log.png, etc.). There, the long tail of variables such as MonthlyIncome was observed, justifying the logarithmic transformation.

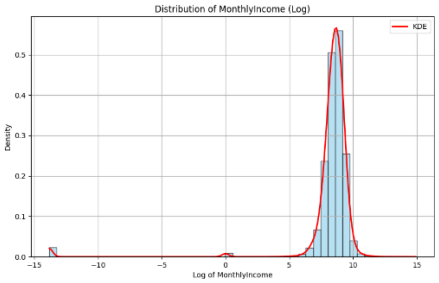




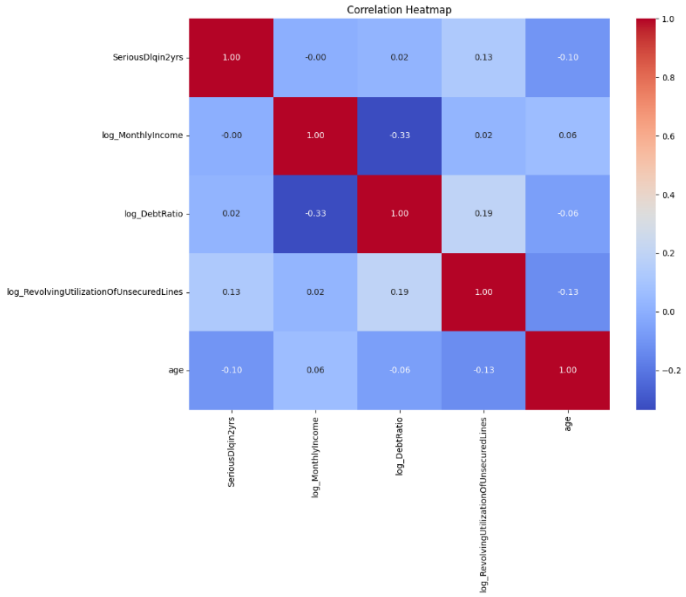








Correlation matrices were generated (plots/correlation_heatmap.png), where low correlations were identified between most variables and the target variable SeriousDlqin2yrs, suggesting the need to create additional features



Feature Engineering

Logarithmic transformations on MonthlyIncome, DebtRatio, and RevolvingUtilizationOfUnsecuredLines to stabilize the distribution and reduce outliers.

Total_Morosidad: Sum of NumberOfTime30-59Days, NumberOfTime60-89Days, and NumberOfTimes90DaysLate.

Income_to_Debt: Ratio between log_MonthlyIncome and log_DebtRatio to capture income/debt proportionality more stably.

Entrenamiento de Modelos y Visualización de Resultados

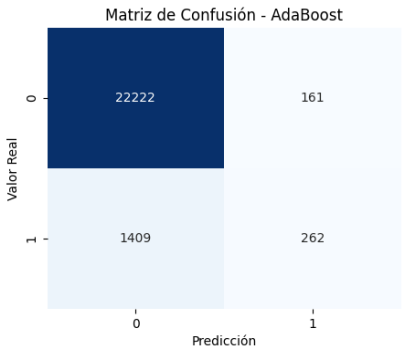
Train-Test Split

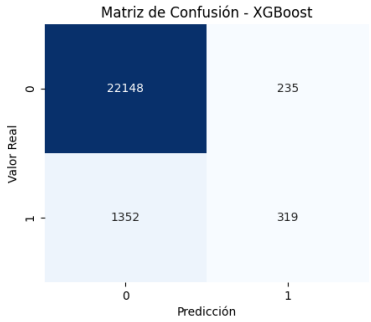
An 80%-20% split was used with stratification by SeriousDlqin2yrs. This ensures that the proportion of defaulting clients is representative in both sets.

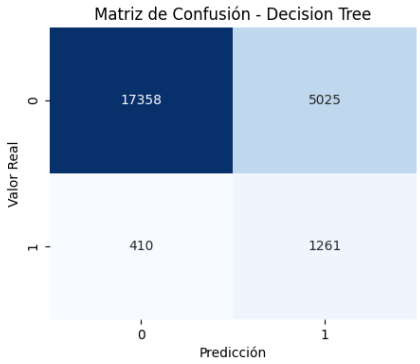
Scaling
Outliers were mitigated using RobustScaler, maintaining the necessary robustness for the logarithmic variables.

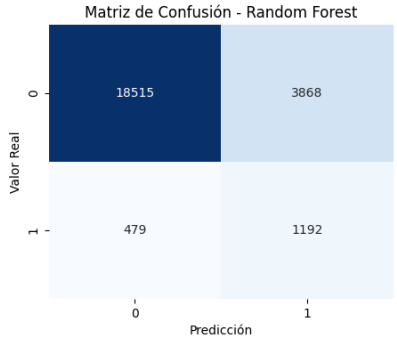
Trained Models
Logistic Regression, Decision Tree, Random Forest, XGBoost, and AdaBoost were optimized with GridSearchCV to maximize recall, given the importance of identifying high-risk clients.

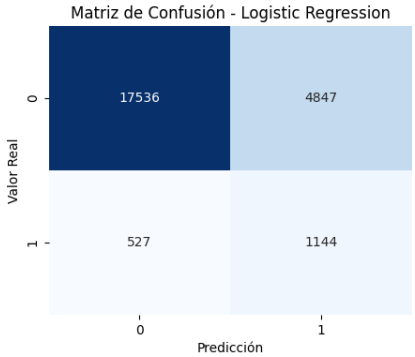
Results and Metrics
Based on the confusion matrices (e.g., plots/confusion_matrix_Decision_Tree.png), Accuracy, ROC AUC, and Recall were calculated.











According to the metrics matrix (below), Decision Tree and Random Forest presented a recall higher than 70%, whereas XGBoost and AdaBoost exceeded 93% in Accuracy but had very low recall (~20%).

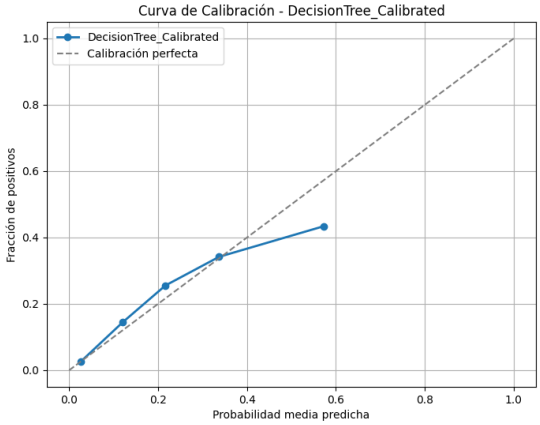
	Accuracy	ROC AUC	Recall
Logistic Regression	0.776	0.802	0.684
Decision Tree	0.774	0.839	0.754
Random Forest	0.819	0.844	0.713
XGBoost	0.934	0.849	0.19
AdaBoost	0.934	0.846	0.156

The Decision Tree showed a good balance, and the logarithmic transformation helped improve discrimination in variables with outliers.

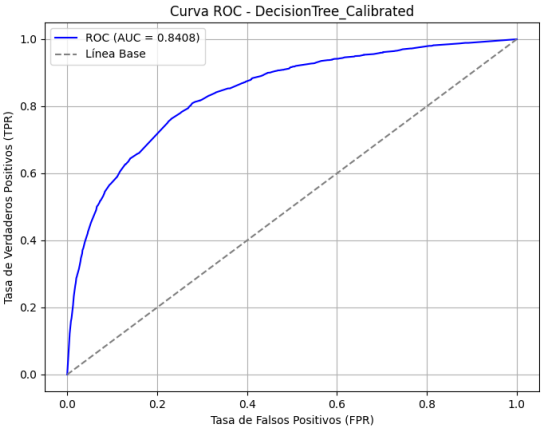
Calibración

In order to assign more reliable probabilities, the Decision Tree was calibrated using the sigmoid method, generating:

A Calibration Curve(plots/calibration_curve_Decision_Tree_Calibrated.png).



An ROC Curve for the calibrated model (plots/roc_curve_Decision_Tree_Calibrated.png).

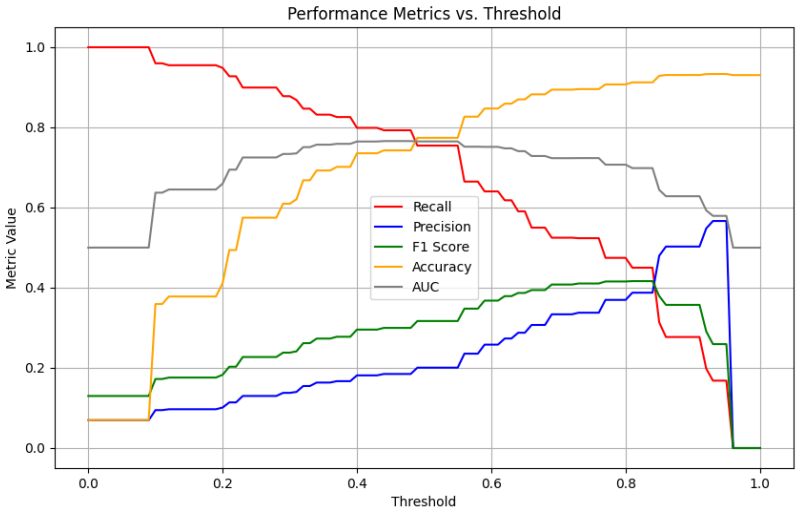


Effect of the Cutoff (Threshold) on Default Prediction

The sensitivity of both models (non-calibrated vs. calibrated) was evaluated across multiple thresholds (0.0 to 1.0):

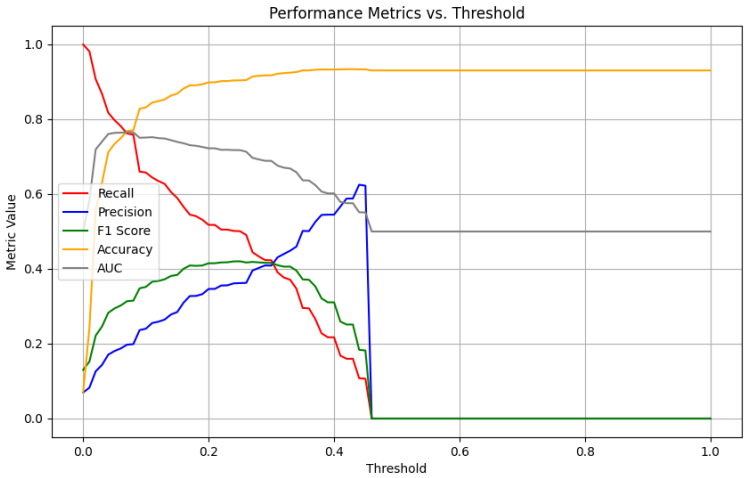
Metric Curves (plots/threshold_metrics.png):

Non-calibrated model



When the threshold is decreased below 0.5, recall increases (capturing more at-risk clients), but the false positive rate rises notably.

Calibrated model



At thresholds < 0.5, although recall increases, there is also a significant rise in the proportion of clients classified as default when they are not.

With thresholds > 0.5, precision improves at the expense of a drop in recall.

The 0.5 threshold for the non-calibrated model best balances the detection of defaulters (recall ~75%) against false positives, thereby avoiding over-penalizing clients who might pay. Despite calibration, the model did not show a higher recall than the non-calibrated one, so the final recommendation is to maintain threshold = 0.5.

Conclusions for Credit Scoring

Logarithmic Transformations: Highly effective for skewed variables, improving the discrimination of the models.

Trade-off Recall vs. Precision: A threshold < 0.5 increases recall but also false positives; a threshold > 0.5 reduces false positives at the expense of recall.

Calibrated Decision Tree: Provides greater interpretability in the probability of default, but does not significantly increase the detection rate (recall).

Threshold Selection: The threshold = 0.5 for the non-calibrated model offers the best balance; the company could adjust it according to its risk appetite, considering the cost of each type of error.

Questionnaire Responses

How to build the target variable:
Based on the payment history, classify as default (1) those who exceed 90 days of missed payments; non-default (0) otherwise.

Relevant Metrics:
Recall to maximize detection of high-risk clients.
ROC AUC as a global indicator of discrimination.
Precision if the goal is to limit false positives.

Effect of the Threshold:
A low threshold means higher recall but more false positives.
A high threshold means fewer false positives, but recall is reduced (risk of underestimating defaulters).

Additional Variables (for AB InBev):
Transaction volumes and purchase frequency (transactional data).
Geographic location.
Seasonal factors (peaks in consumption during festive periods).

CLIENT SEGMENTATION CHALLENGE

Process Overview

Data Preparation and Inspection

The same dataset used in the Credit Scoring challenge was utilized, but with the original variables (MonthlyIncome, DebtRatio, etc.) for direct segmentation.

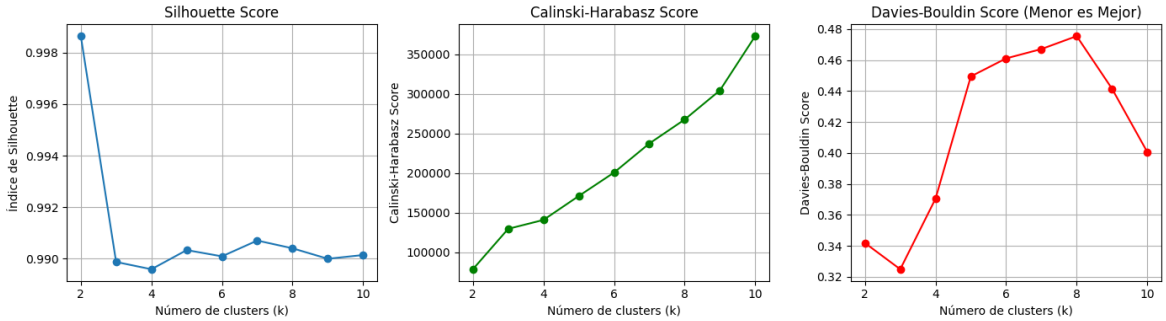
RobustScaler was applied to reduce the impact of extreme values without excessively distorting the data structure.

Algorithms and Results

KMeans was run with various values of k (from 2 to 10), generating metric curves (Silhouette, Calinski-Harabasz, Davies-Bouldin) saved in plots/metric_curves.png.

Boxplots for each variable by cluster were generated (e.g., plots/boxplot_MonthlyIncome_by_cluster.png), showing the distribution of each feature across segments.

DBSCAN was executed to capture potential outliers or groups with arbitrary shapes.



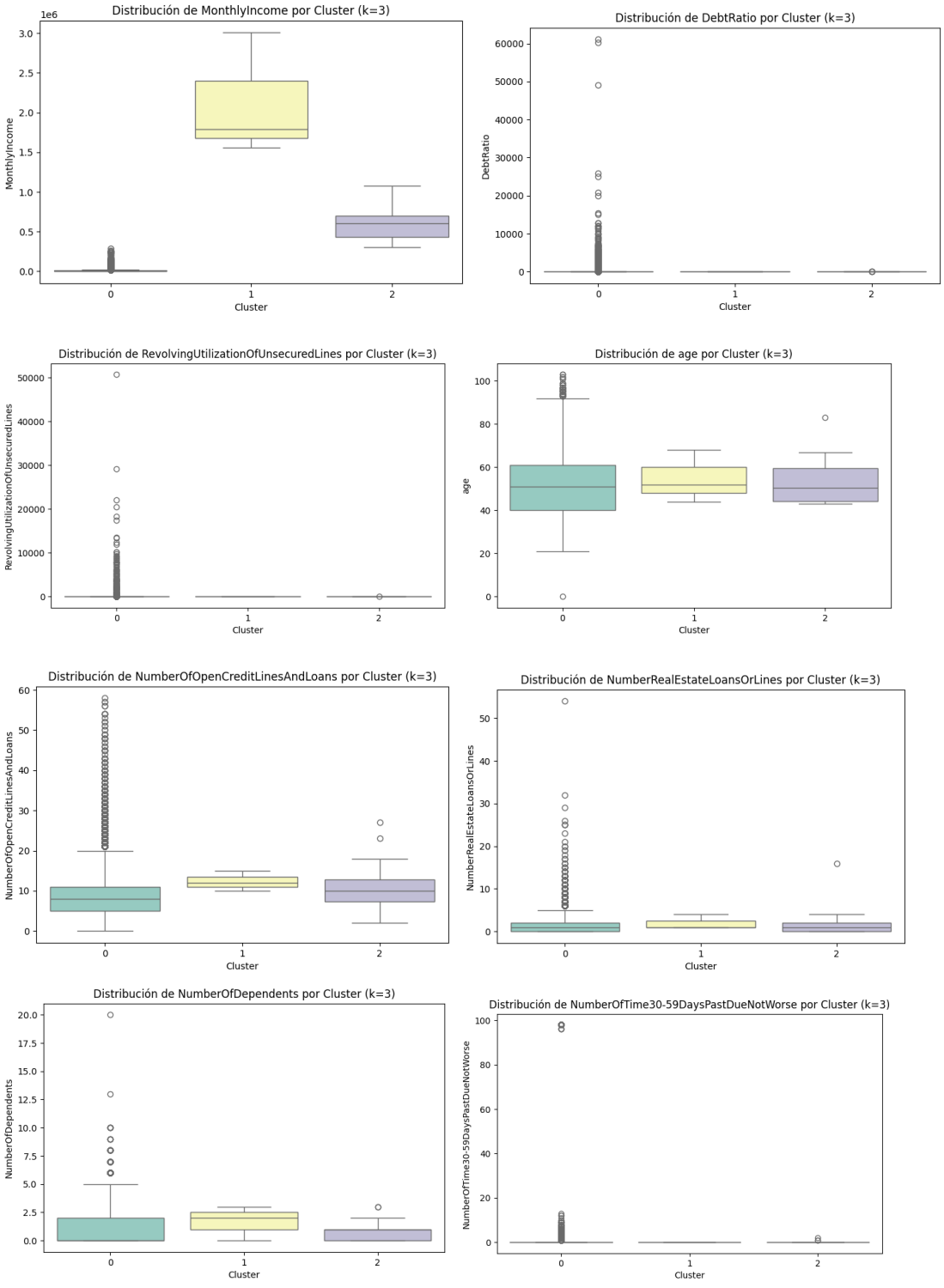
Metrics Analysis and Observations

Metrics Analysis and Observations::

From k=2 to k=10, Silhouette Scores were consistently high (≥0.9899), indicating strong data separability.

The Calinski-Harabasz index increased with k, signaling greater between-cluster dispersion as the number of clusters increases.

The Davies-Bouldin index presented moderate values (~0.32 to 0.47), with slight peaks indicating potential alternative optimal points.



Given the elevated metric values, choosing k requires balancing interpretability with segmentation detail.

Cluster selection

An initial solution with k=3 clusters was defined (as indicated by "KMeans executed with 3 clusters"). However, very high Silhouette values (>0.99) suggest a very high intra-cluster cohesion for various values of k.

The business could choose k=3, k=6, or k=10, depending on the desired granularity. For example, k=3 generates broader, easier-to-manage groups, while k=10 provides more specific subsegments for highly personalized strategies.

Boxplots by Cluster:

Saved as:

- boxplot_MonthlyIncome_by_cluster.png
- boxplot_DebtRatio_by_cluster.png
- boxplot_RevolvingUtilizationOfUnsecuredLines_by_cluster.png
- etc.

These plots reveal notable differences in income, number of dependents, total delinquency, etc., helping to describe each cluster's profile (e.g., a cluster with low income and high delinquency vs. a cluster with multiple credit lines and high payment capacity).**DBSCAN**

The output "DBSCAN executed" confirmed the automatic detection of outliers or atypical points; however, the results were not optimal.

Conclusions for Segmentation

Silhouette >0.99 and increasing Calinski-Harabasz values suggest a strongly separable structure across different k values. Although k=3 was executed as a midpoint solution, the choice of k should balance interpretability and detail.

Interpretación de los 3 Clústeres Formados:

Clúster 0:

Clients with moderate incomes (median MonthlyIncome = 5,400) and relatively high average DebtRatio (indicating some extreme values), with about 8 open credit lines and some mild payment delays. This represents a moderate-risk group that may benefit from targeted monitoring and adjusted financing options.Representa un grupo de riesgo medio, cuyos clientes pueden beneficiarse de programas de seguimiento y opciones de financiamiento ajustadas para minimizar el sobreendeudamiento.

Clúster 1:

Clients with high incomes (median MonthlyIncome ≈ 1,794,060) and extremely low DebtRatio (median ≈ 0.0028), with a higher number of open credit lines (median = 12) and

almost no payment delays. This is a premium, low-risk segment suitable for expanded credit offers and high-value financial products.

Clúster 2:

Clients with moderate to low incomes (median MonthlyIncome ≈ 605,685) and moderately low DebtRatio (median ≈ 0.00385). They typically have around 10 open credit lines and minimal indications of delinquency. This group likely represents an intermediate-risk segment that might require personalized monitoring or restructuring strategies to prevent further deterioration.

Final Conclusions

Credit Scoring:

- Log Transformations: Proven to be highly effective in stabilizing skewed distributions and improving model discrimination by reducing the influence of outliers.
- Threshold Analysis:
  The evaluation of different thresholds indicates that a cutoff of **0.5** for the non-calibrated Decision Tree model strikes the best balance between recall and false positives. Although calibration improves probability interpretability, it did not significantly increase recall, reinforcing the selection of a 0.5 threshold.
- Business Impact:
  Employing the calibrated Decision Tree model with a threshold of 0.5 ensures an optimal trade-off in risk detection, thereby minimizing financial losses due to undetected defaulters.

Client Segmentation:

- A segmentation using k=3 clusters is a viable solution, as determined by the elbow method and internal metrics.
- The three clusters exhibit distinct profiles, supporting targeted marketing and risk management strategies:
  - Cluster 0: Moderate risk with average income and moderate credit usage.
  - Cluster 1: Premium, low-risk customers with high income and minimal defaults.
  - Cluster 2: Intermediate-risk clients who may need closer monitoring and possibly tailored credit restructuring.

- Visualizations such as boxplots and PCA projections validate the segmentation quality and provide actionable insights for business strategy.

MLOps Standards:
In my career, I have applied and promoted good MLOps practices. In projects, I have automated real-time data processing pipelines and model deployment in production.

Script Structure in an ML Project with Business Objectives:
I consider it essential that the script structure is modular and clear. I typically include:

Initial Setup: Library imports, definition of parameters, and environmental variables.

Data Ingestion and Processing: Loading, cleaning, and transforming data from various sources.

Exploratory Analysis: Visualization and analysis to understand distribution and detect patterns or anomalies.

Feature Engineering: Creation and selection of relevant variables, applying scaling or normalization when necessary.

Training and Validation: Splitting data into training and test sets, hyperparameter tuning, and cross-validation.

Evaluation and Reporting: Generation of performance metrics, error analysis, and visualization of results to support decision-making.

Deployment and Automation: Integration of the model into CI/CD pipelines that allow for its updating and continuous monitoring in production.

Importance of Integrating an ML Project into the Company's Infrastructure:
It is crucial for Machine Learning projects to be integrated into the company's system framework because it enables the solution to align with internal processes and strategic objectives. This integration facilitates automation and the continuous flow of data, reducing manual intervention and improving operational efficiency, as well as ensuring scalability, maintenance, and continuous monitoring of the model. This results in greater robustness and reliability in decision-making.

Experience with Cloud Architectures and Knowledge of AzureML
Yes, I have experience in cloud computing working with platforms such as Azure and AWS. I have implemented ETL processes, developed CI/CD pipelines, and deployed models in production in cloud environments. My work with Azure has allowed me to become familiar with tools and frameworks such as AzureML, which facilitates automation, scaling, and comprehensive model management.

Familiarity with the Spark Framework and Development in PySpark/Scala:
I have used PySpark to implement both a churn prediction model and a predictive model to optimize the placement of coolers. These projects enabled me to leverage Spark's distributed processing capabilities.

Tabla de Conceptos Relevantes

Concept	Description	Example of a Business Scenario
Recommender Systems	Algorithms that suggest personalized products or services based on user behavior.	Recommending beers or products based on purchase history in distribution chains.
XGBoost	An optimized boosting algorithm that improves accuracy by modeling residual errors.	Predicting default by combining multiple decision trees with regularization.
Cross-Validation	A validation technique to assess model performance by repeatedly partitioning the dataset into training and validation sets.	Robust evaluation during the training of credit scoring models.
Silhouette Test	A metric that measures the cohesion and separation of clusters; higher values indicate well-defined clusters.	Determining the optimal number of clusters for segmenting customers in marketing campaigns.
ROC Curve	A curve that shows the relationship between the true positive rate and false positive rate for various thresholds.	Evaluating a model's ability to differentiate between defaulters and non-defaulters.
AUC Metric	The area under the ROC curve, which quantifies the model's ability to discriminate between classes.	Global model comparison in terms of discriminative capacity in credit scoring.