INFORME RESULTADOS

Kristhian Santiago Palomino Fajardo

RETO DE CREDIT SCORING

Resumen del Proceso

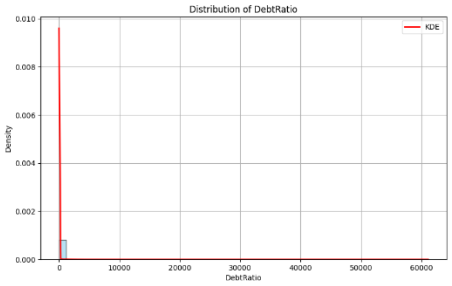
Análisis Exploratorio y Preprocesamiento

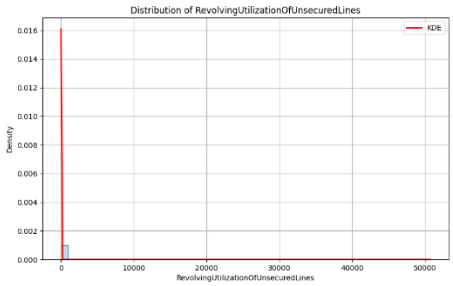
Se utilizó la base cs-training.csv, la cual contiene 150,000 registros de clientes con información crediticia e indicadores de morosidad.

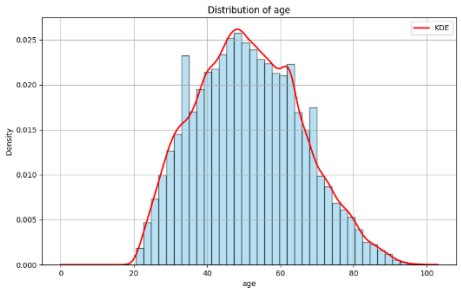
De estos, se eliminaron ~29,731 registros que presentaban `MonthlyIncome` nulo (alrededor de 20% de la muestra). Esta decisión se fundamentó en que dichos clientes, en su mayoría, no tenían un claro historial de ingresos, y la distribución de la variable objetivo (SeriousDlqin2yrs) en esos registros no apuntaba a un grupo especialmente riesgoso. Con ello se evitó introducir un sesgo fuerte mediante una imputación no confiable. Para la columna `NumberOfDependents`, se imputaron valores nulos con 0, dado que su proporción de nulos (~2%) no implicaba un cambio significativo en la distribución y podría asumirse que, en caso de no tener información, lo más seguro era considerarlo "sin dependientes".

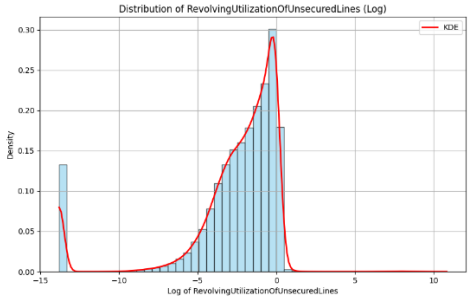
Se generaron las siguientes visualizaciones (todas guardadas en la carpeta *plots*):

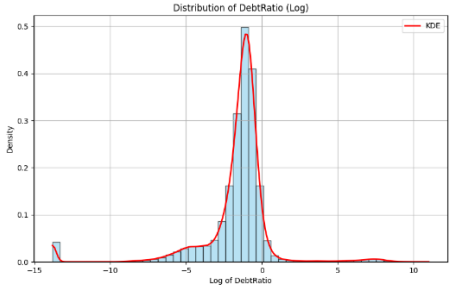
Distribuciones de variables clave: MonthlyIncome, DebtRatio, RevolvingUtilizationOfUnsecuredLines y age (ej. plots/MonthlyIncome.png, plots/MonthlyIncome_log.png, etc.). Allí se apreció la cola larga de variables como MonthlyIncome, justificando la transformación logarítmica.

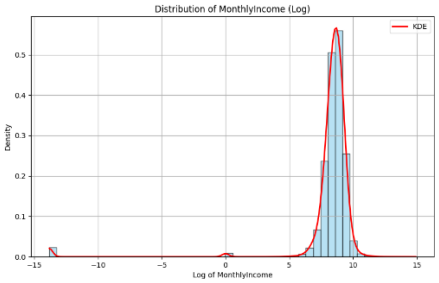




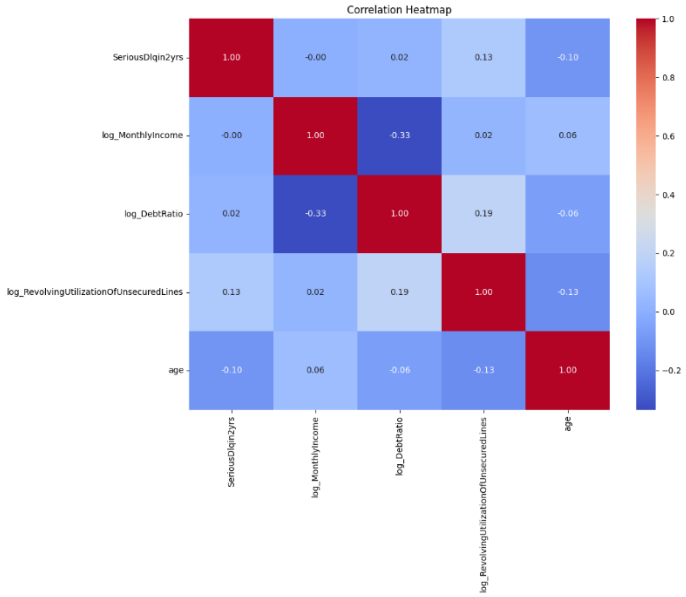








Se generaron matrices de correlación (plots/correlation_heatmap.png), donde se identificaron relaciones bajas entre la mayoría de variables y la variable objetivo SeriousDlqin2yrs, lo que sugiere la necesidad de crear features adicionales.



Feature Engineering

Transformaciones logarítmicas en MonthlyIncome, DebtRatio, y RevolvingUtilizationOfUnsecuredLines para estabilizar la distribución y reducir outliers.

Total_Morosidad: Suma de NumberOfTime30-59Days, NumberOfTime60-89Days y NumberOfTimes90DaysLate.

Income_to_Debt: Relación entre log_MonthlyIncome y log_DebtRatio para capturar de manera más estable la proporcionalidad de ingresos/deuda.

Entrenamiento de Modelos y Visualización de Resultados

División en Entrenamiento y Prueba

Se usó un split 80%-20% con estratificación por SeriousDlqin2yrs. Esto garantiza que la proporción de clientes en default sea representativa en ambos conjuntos.

Escalado

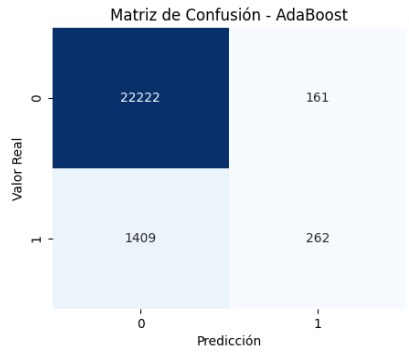
Mediante RobustScaler, se mitigaron outliers manteniendo la robustez necesaria para las variables logarítmicas.

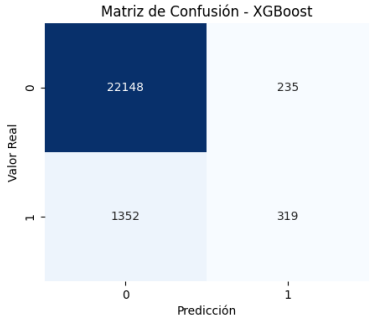
Modelos Entrenados

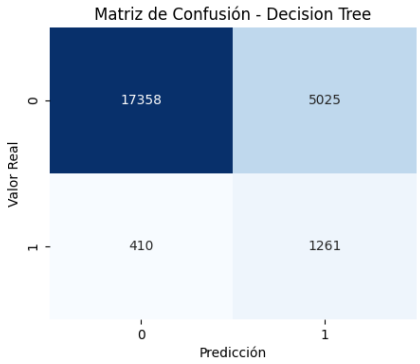
Logistic Regression, Decision Tree, Random Forest, XGBoost y AdaBoost, optimizados con GridSearchCV para maximizar recall, dada la relevancia de identificar clientes de alto riesgo.

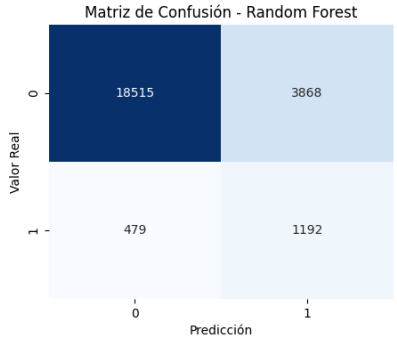
Resultados y Métricas

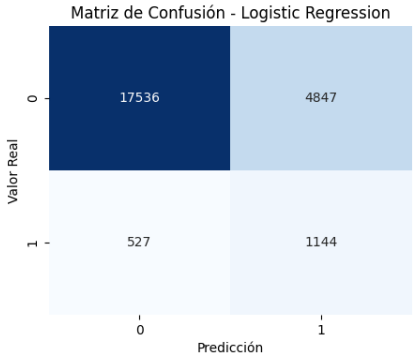
A partir de las matrices de confusión (p.ej., plots/confusion_matrix_Decision_Tree.png), se calcularon Accuracy, ROC AUC y Recall.











Según la matriz de métricas (abajo), Decision Tree y Random Forest presentaron un recall superior al 70%, mientras que XGBoost y AdaBoost superaron el 93% de Accuracy, pero con un recall muy bajo (~20%).

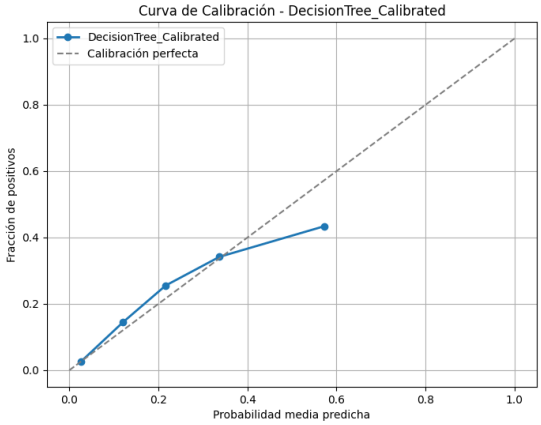
	Accuracy	ROC AUC	Recall
Logistic Regression	0.776	0.802	0.684
Decision Tree	0.774	0.839	0.754
Random Forest	0.819	0.844	0.713
XGBoost	0.934	0.849	0.19
AdaBoost	0.934	0.846	0.156

El Decision Tree mostró un buen equilibrio, y la transformación logarítmica ayudó a mejorar la discriminación en variables con outliers.

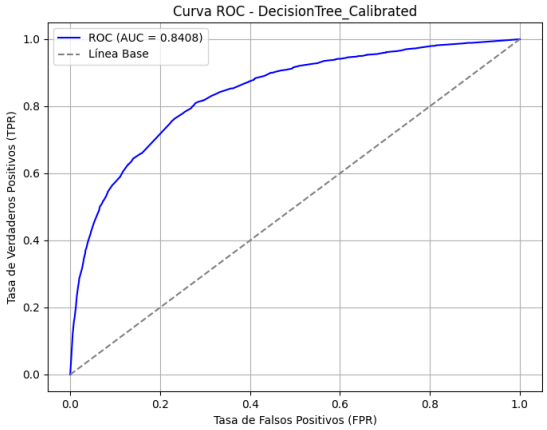
Calibración

Con el fin de asignar probabilidades más confiables, se calibró el **Decision Tree** con método sigmoide, generando:

Curva de Calibración (plots/calibration_curve_Decision_Tree_Calibrated.png).



Curva ROC para el modelo calibrado (plots/roc_curve_Decision_Tree_Calibrated.png).

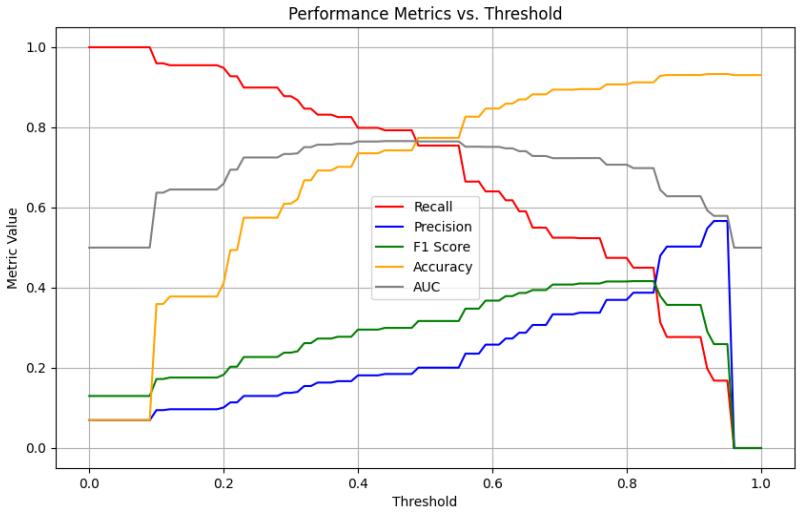


Efecto del Cutoff o Threshold en la Predicción de Default

Se evaluó la sensibilidad de ambos modelos (no calibrado vs. calibrado) a lo largo de múltiples thresholds (0.0 a 1.0):

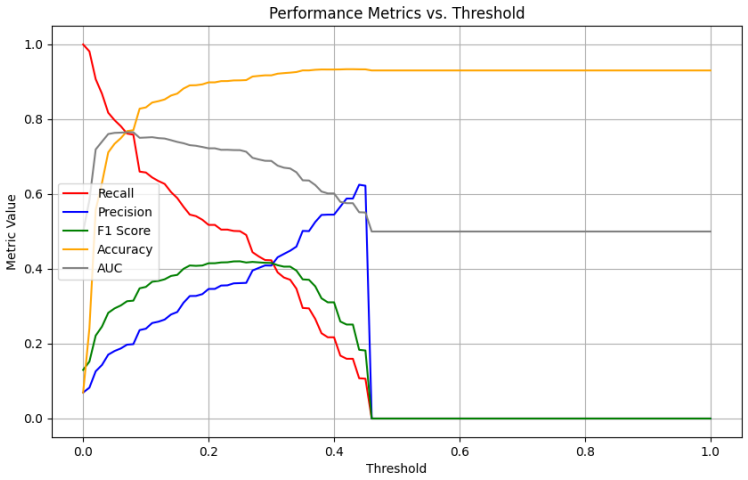
Curvas de Métricas (plots/threshold_metrics.png):

Modelo no calibrado:



Al disminuir el threshold por debajo de 0.5, el recall aumenta (capturando más clientes en riesgo), pero la tasa de falsos positivos sube de manera notable.

Modelo calibrado



A niveles de threshold < 0.5, si bien sube el recall, también aumenta considerablemente la proporción de clientes clasificados como default sin serlo.

Con threshold > 0.5, la precisión mejora a costa de una caída en el recall

El umbral de 0.5 en el modelo no calibrado es el que mejor balancea la detección de defaulters (recall ~75%) frente a los falsos positivos, evitando sobrecastigar a clientes que podrían pagar.

Pese a la calibración, el modelo no evidenció un recall superior al no calibrado, por lo que la recomendación final es mantener threshold = 0.5.

Conclusiones del Credit Scoring

Transformaciones logarítmicas: Altamente eficaces para variables sesgadas, logrando mejorar la discriminación de los modelos.

Trade-off Recall vs. Precisión: Un threshold < 0.5 aumenta recall, pero también falsos positivos; uno > 0.5 reduce falsos positivos, sacrificando recall.

Árbol de Decisión calibrado: Aporta mayor interpretabilidad en la probabilidad de incumplimiento, pero no incrementa sustancialmente la tasa de detección (recall).

Elección de umbral: El threshold = 0.5 para el modelo no calibrado ofrece el mejor equilibrio; la empresa podría ajustarlo según su apetito de riesgo, teniendo presente el costo de cada tipo de error.

Respuestas al Cuestionario

Cómo construir la variable objetivo:

A partir del historial de pagos, clasificar como default (1) a quien exceda los 90 días de impago; no default (0) de lo contrario.

Métricas relevantes:

Recall para maximizar la detección de clientes en riesgo.

ROC AUC como indicador global de discriminación.

Precision si se busca limitar falsos positivos.

Efecto del threshold:

Threshold bajo significa mayor recall, pero más falsos positivos.

Threshold alto significa menos falsos positivos, pero se reduce el recall (riesgo de subestimación de clientes en default).

Variables adicionales (AB InBev):

Volúmenes y frecuencia de compra (transaccionales).

Ubicación geográfica.

Factores estacionales (picos de consumo en periodos festivos).

RETO DE SEGMENTACIÓN DE CLIENTES (NO SUPERVISADO)

Resumen del Proceso

Preparación e Inspección de Datos

Se trabajó con las mismas bases empleadas en el reto de Credit Scoring, pero usando variables originales (MonthlyIncome, DebtRatio, etc.) para una segmentación directa.

Se aplicó RobustScaler para reducir el impacto de valores extremos sin distorsionar excesivamente la estructura de los datos.

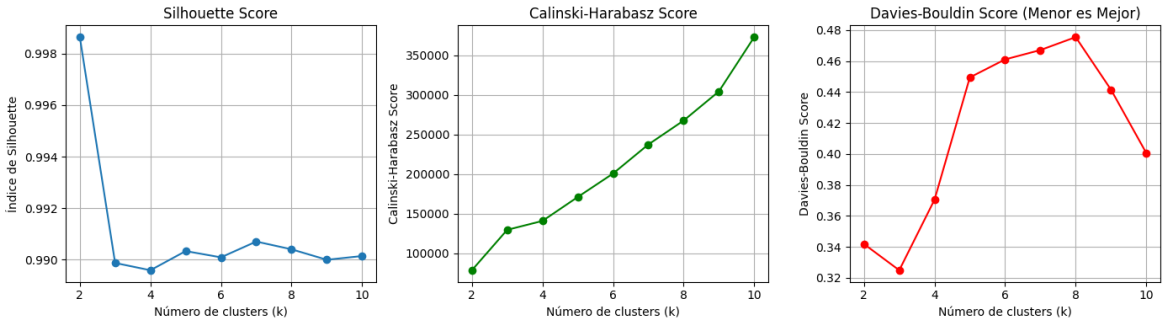
Algoritmos y Resultados

KMeans con diferentes valores de k (2 a 10), generando curvas de métricas (Silhouette, Calinski-Harabasz, Davies-Bouldin) guardadas en *plots/metric_curves.png*.

Boxplots por cada variable y por cluster (e.g., *boxplot_MonthlyIncome_by_cluster.png*), mostrando la distribución de cada feature en cada segmento.

Visualización PCA (*pca_clusters_kmeans.png*) para observar la separación bidimensional de los clusters.

DBSCAN, ejecutado para capturar posibles outliers o grupos de forma arbitraria



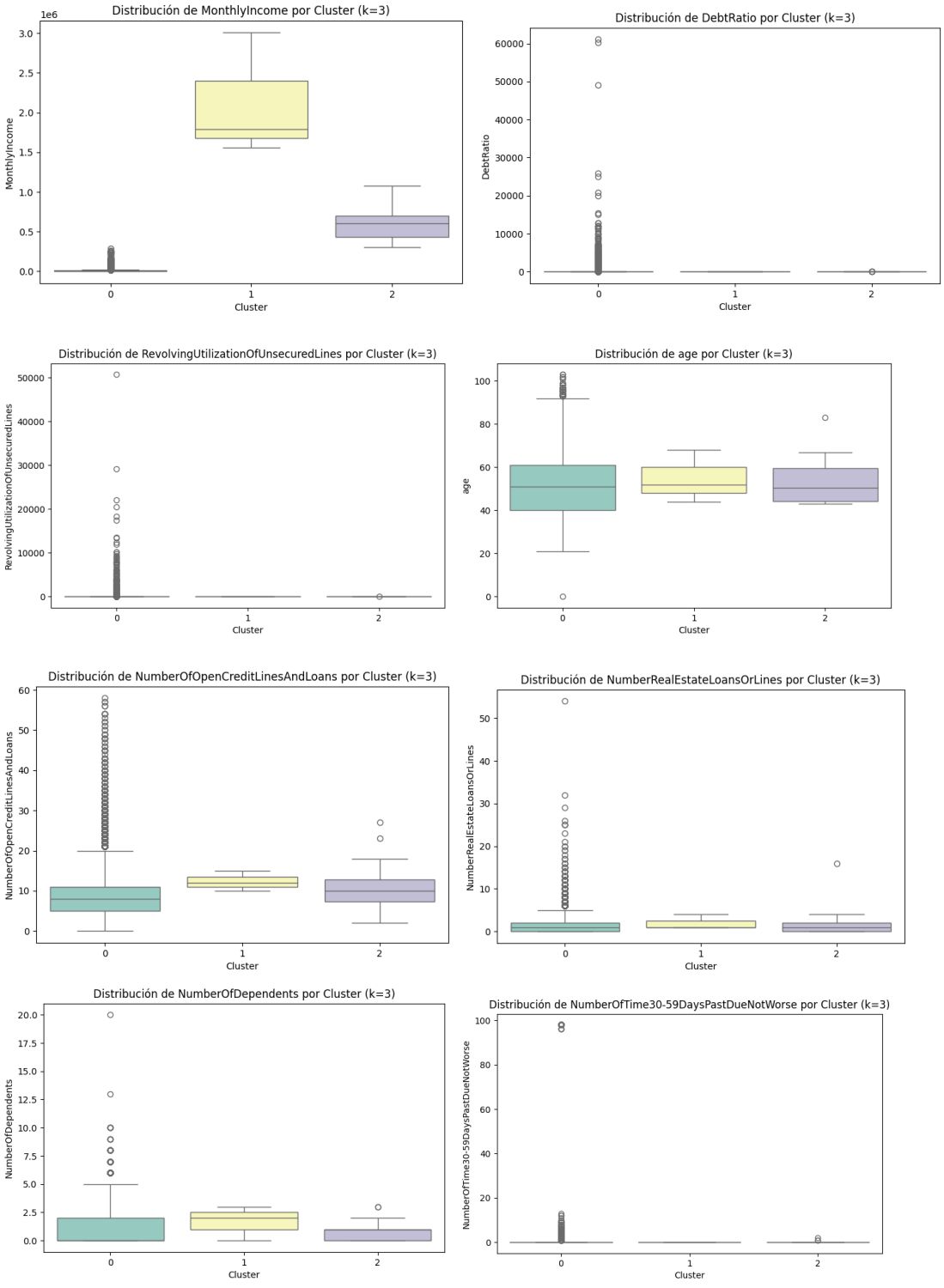
Análisis de Métricas y Observaciones

Curvas de Métricas:

De k=2 a k=10, se obtuvieron Silhouette Scores consistentemente altos (≥0.9899), lo cual sugiere una fuerte separabilidad de los datos.

El índice Calinski-Harabasz incrementa con k, indicando una mayor dispersión inter-cluster conforme crece el número de grupos.

El Davies-Bouldin presenta valores moderados (~0.32 a 0.47), con ligeros picos que apuntan a distintos puntos óptimos potenciales.



Dado que las métricas son muy elevadas, la elección de k implica un balance entre interpretabilidad y detalle en la segmentación.

Elección de Clusters

Se definió un k=3 como inicial (log con *"KMeans ejecutado con 3 clusters"*). Sin embargo, datos como Silhouette >0.99 sugieren que existe una altísima cohesión intra-cluster para diversos valores de k.

El negocio podría elegir k=3, k=6 o k=10, según la granularidad deseada. Por ejemplo, k=3 genera grupos más amplios y fáciles de gestionar, mientras k=10 ofrece subsegmentos más específicos para estrategias ultra-personalizadas.

Boxplots por Clúster

Guardados como:

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

Revelan diferencias notables en ingresos, número de dependientes, morosidad total, etc. Esto facilita describir el "perfil" de cada clúster (p. ej., clúster con bajo ingreso y alta morosidad, vs. clúster con múltiples líneas de crédito y alta capacidad de pago).

DBSCAN

Se ejecutó *"DBSCAN executed"*, confirmando la detección automática de outliers o puntos atípicos, no se obtuvieron buenos resultados.

Conclusiones de la Segmentación

Silhouette >0.99 y Calinski-Harabasz crecientes sugieren una estructura fuertemente separable en diversos valores de k. k=3 (ejecutado) es un punto de partida viable que brinda un nivel de detalle intermedio.

Interpretación de los 3 Clústeres Formados:

Clúster 0:

Clientes con ingresos moderados (median MonthlyIncome = 5400) y cifras de deuda relativamente altas en términos de media (aunque la mediana de DebtRatio es baja), lo que sugiere la presencia de outliers extremos. Aproximadamente 8 líneas abiertas, con un leve nivel de retrasos.

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:

Clientes con ingresos elevados (median MonthlyIncome ≈ 1,794,060) y una proporción de deuda extremadamente baja (median DebtRatio ≈ 0.0028). Poseen un mayor número de líneas de crédito (median NumberOfOpenCreditLinesAndLoans = 12) y casi no presentan retrasos.

Se trata de un segmento premium o de bajo riesgo, idóneo para ofertas de crédito ampliadas o productos financieros de mayor valor, con alto potencial de fidelización.

Clúster 2:

Clientes con ingresos moderados a bajos (median MonthlyIncome ≈ 605,685) y DebtRatio moderadamente bajos (median ≈ 0.00385). Tienen un número moderado de líneas de crédito y presentan ligeros indicios de morosidad (con valores de retraso casi nulos en la mediana, pero con una media que sugiere la existencia de casos aislados).

Este grupo podría representar un segmento en transición o de riesgo intermedio, donde es fundamental implementar estrategias de reestructuración o asistencia crediticia para evitar un deterioro en el comportamiento financiero.

Conclusiones finales

Credit Scoring:

- Transformaciones logarítmicas: Se ha demostrado que son muy eficaces para estabilizar las distribuciones sesgadas y mejorar la discriminación de los modelos reduciendo la influencia de los valores atípicos.
- Análisis de umbrales: La evaluación de distintos umbrales indica que un umbral de 0,5 para el modelo de árbol de decisión no calibrado logra el mejor equilibrio entre recuerdo y falsos positivos. Aunque la calibración mejora la interpretabilidad de las probabilidades, no aumentó significativamente la recuperación, lo que refuerza la selección de un umbral de 0,5.
- Impacto empresarial: El empleo del modelo de árbol de decisión calibrado con un umbral de 0,5 garantiza un equilibrio óptimo en la detección de riesgos, minimizando así las pérdidas financieras debidas a morosos no detectados.

Client Segmentation:

- Una segmentación utilizando k=3 clusters es una solución viable, tal y como determinan el método del codo y las métricas internas.
- The three clusters exhibit distinct profiles, supporting targeted marketing and risk management strategies:
    - Cluster 0: Riesgo moderado con ingresos medios y uso moderado del crédito.

- Cluster 1: Clientes premium de bajo riesgo con ingresos elevados e impagos mínimos.
- Cluster 2: Clientes de riesgo intermedio que pueden necesitar un seguimiento más estrecho y posiblemente una reestructuración del crédito a medida.

Las visualizaciones, como los gráficos de caja y las proyecciones PCA, validan la calidad de la segmentación y proporcionan información práctica para la estrategia empresarial.

Conceptos y Comprensión

Estándares de MLOps:
En mi trayectoria, he aplicado y promovido buenas prácticas de MLOps. En proyectos he automatizado pipelines de procesamiento de datos en tiempo real y el despliegue de modelos en producción.

Estructura del script en un proyecto de ML con objetivos de negocio:
Considero fundamental que la estructura del script sea modular y clara. Normalmente, incluyo:

Configuración inicial: Importación de librerías, definición de parámetros y variables de entorno.

Ingesta y procesamiento de datos: Carga, limpieza y transformación de datos provenientes de diversas fuentes.

Análisis exploratorio: Visualización y análisis para entender la distribución y detectar patrones o anomalías.

Ingeniería de características: Creación y selección de variables relevantes, aplicando escalado o normalización cuando es necesario.

Entrenamiento y validación: División de los datos en conjuntos de entrenamiento y prueba, ajuste de hiperparámetros y validación cruzada.

Evaluación y reporte: Generación de métricas de rendimiento, análisis de errores y visualización de resultados para la toma de decisiones.

Despliegue y automatización: Integración del modelo en pipelines CI/CD que permitan su actualización y monitoreo en producción.

Importancia de la integración de un proyecto de ML en la infraestructura de la empresa:
Es crucial que los proyectos de Machine Learning se integren en el marco del sistema de la compañía porque permite que la solución se alinee con los procesos internos y objetivos estratégicos, facilita la automatización y el flujo continuo de datos, reduciendo la

intervención manual y mejorando la eficiencia operativa y garantiza la escalabilidad, el mantenimiento y el monitoreo continuo del modelo, lo que resulta en una mayor robustez y confiabilidad en la toma de decisiones.

Experiencia en arquitecturas cloud y conocimiento de AzureML:
Sí, cuento con experiencia en cloud computing trabajando con plataformas como Azure y AWS. He implementado procesos ETL, desarrollado pipelines de CI/CD y desplegado modelos en producción en entornos cloud. Mi trabajo con Azure me ha permitido familiarizarme con herramientas y frameworks como AzureML, lo que facilita la automatización, el escalado y la gestión integral de los modelos.

Familiaridad con el framework Spark y desarrollo en PySpark/Scala:

He utilizado PySpark para implementar tanto el modelo de churn prediction como el modelo predictivo para optimizar la ubicación de coolers. Estos proyectos me permitieron aprovechar las capacidades de procesamiento distribuido de Spark.

Tabla de Conceptos Relevantes

Concepto	Descripción	Ejemplo de Escenario de Negocio
Recommender Systems	Algoritmos que sugieren productos o servicios personalizados basados en el comportamiento del usuario.	Recomendación de cervezas o productos en función del historial de compras en cadenas de distribución.
XGBoost	Algoritmo de boosting optimizado que mejora la precisión mediante el modelado de errores residuales.	Predicción de default mediante la fusión de múltiples árboles de decisión con regularización.
Cross-Validation	Técnica de validación para evaluar la performance del modelo mediante particiones repetidas del dataset.	Validación robusta en el entrenamiento de modelos de scoring crediticio.
Silhouette Test	Indicador que mide la cohesión y separación de clusters; valores más altos indican clusters bien definidos.	Selección del número óptimo de clústeres para segmentar clientes en campañas de marketing.
ROC Curve	Curva que muestra la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos, para diversos umbrales.	Evaluación de la capacidad del modelo para diferenciar entre defaulters y no defaulters.

AUC Metric	Área bajo la curva ROC, que cuantifica la capacidad del modelo para discriminar entre clases.	Comparación global de modelos en términos de capacidad discriminativa en scoring de crédito.