Comparación de Técnicas de Machine Learning para la Predicción de Bancarrotas en el Sector Bancario

Cristian Marín, Laura Tascón, Samuel Montoya
Ingeniería de Sistemas, Universidad de Antioquia
Medellín, Colombia

***Abstract*—Este estudio compara distintas técnicas de aprendizaje automático enfocados en la predicción de bancarrotas en entidades bancarias. Evaluamos cinco modelos clave: un modelo paramétrico (regresión logística), no paramétrico (k-neighbors classifier), ensambles de árboles de decisión (Random Forest - Gradient Boosting), una red neuronal artificial (MLP) y un maquina de soporte vectorial (SVM). Para el entrenamiento de los modelos se usaron datos históricos de bancos en Taiwán (1999-2009).**

I. Introducción

La Predicción de eventos de bancarrota de las empresas, especialmente en el sector bancario, es un tema de suma importancia para la gestión financiera y toma de decisiones oportunas por parte de los entes regulatorios. Anticipar la insolvencia de los bancos no solo ayuda a extender la vida útil de estas entidades, sino que además previene desestabilizaciones económicas que podrían afectar a toda una población. Aunque los modelos tradicionales, como el análisis discriminante o la regresión logística, han sido útiles, el aprendizaje automático ha traído nuevas herramientas más avanzadas.

Este estudio se enfoca en comparar el rendimiento de diferentes modelos de aprendizaje automático para la predicción de bancarrotas bancarias. Evaluamos desde métodos tradicionales hasta otros más modernos, como los ensambles de árboles y las redes neuronales artificiales.

II. Descripcion del problema

Cuando hablamos de predecir bancarrotas bancarias, nos referimos a estimar la probabilidad de un evento de insolvencia asociado a una entidad bancaria banco dentro de un periodo específico. Dicha tarea de prediccion implica la manipulacion de datos diversos datos financieros, que abarcan desde indicadores contables tradicionales (ratios de endeudamiento, liquidez o rentabilidad) hasta medidas de eficiencia operativa (Sales Per Employee o Fixed Asset Turnover). También variables de flujo de caja (Cash Flow Per Share, Cash Flow to Total Assets) y tasas de crecimiento en diversas áreas, lo que nos permite un análisis detallado y preciso de la situación financiera de las entidades bancarias.

El reto principal de esta problematica es la elección y ajuste del modelo de machine learning para la detectecion de señales o patrones y su posterior uso con nuevos datos. Para lograrlo, es esencial realizar una exploración detallada de sus parámetros y comparar distintos tipos de modelos co el objetivo descubrir las técnicas más potentes para anticipar las quiebras bancarias.

La base de datos empleada en este proyecto contiene información financiera detallada de distintas entidades. Está compuesta por un total de 6819 muestras, cada una representando una instancia o empresa caracterizada mediante múltiples métricas financieras.

El conjunto incluye 96 atributos, de los cuales el primero corresponde a la etiqueta de clase (es decir, la variable a predecir), mientras que los restantes 95 atributos representan variables explicativas que describen el estado financiero de cada empresa a través de diversos indicadores contables, de liquidez, endeudamiento, rentabilidad, eficiencia y crecimiento.

Entre los indicadores más relevantes se encuentran:

- Indicadores de endeudamiento: *Cost of Interest-bearing Debt (X1)*, *Liability to Equity (X35)*, *Interest-bearing Debt/Equity (X8)*.
- Indicadores de liquidez: *Current Ratio (X3)*, *Acid Test (X4)*, *Cash/Current Liability (X17)*.
- Indicadores de rentabilidad: *Return on Total Assets (X51–X53)*, *EPS (X61)*, *Net Income to Total Assets (X66)*.
- Ratios de eficiencia y actividad: *Total Asset Turnover (X71)*, *Inventory Turnover (X74)*.
- Variables binarias: como *X27*, que vale 1 si el pasivo total supera el activo total, y *X69*, que vale 1 si la utilidad neta fue negativa en los dos últimos años.

Para abordar el problema de predicción de quiebra empresarial, se seleccionó el paradigma de aprendizaje supervisado, particularmente bajo un enfoque de clasificación binaria, en virtud de que la variable objetivo (`Bankrupt?`) representa una categoría dicotómica (quiebra vs. no quiebra).

Esta elección metodológica se fundamenta en los siguientes aspectos:

- La existencia de una etiqueta de clase permite el entrenamiento supervisado con estructuras de entrada-salida bien definidas.
- La predicción de la quiebra a partir de métricas financieras cuantitativas es una problemática clásica en la literatura de análisis financiero.
- El conjunto de datos contiene una diversidad de atributos financieros (95 variables independientes) que permiten capturar múltiples dimensiones de la situación económica de las empresas.

Entre los modelos considerados se encuentran:

- Regresión logística, por su interpretabilidad y solidez en problemas lineales.

- Árboles de decisión y Random Forest, por su capacidad para manejar relaciones no lineales y conjuntos de datos complejos.
- Máquinas de vectores de soporte (SVM) y redes neuronales, en contextos que demandan una mayor capacidad de representación.

Dado que es previsible un desbalance en la distribución de clases, se considera el uso de técnicas como el sobremuestreo de la clase minoritaria, submuestreo de la clase mayoritaria, o el ajuste de pesos en los algoritmos (`class_weight=balanced`). Asimismo, se utilizarán métricas adecuadas como el F1-score y el AUC-ROC para evaluar el rendimiento en contextos desbalanceados.

En síntesis, el paradigma seleccionado se ajusta a la naturaleza del problema y al tipo de datos disponibles, permitiendo desarrollar modelos predictivos con potencial de aplicación real en contextos de análisis de riesgo financiero.

III. Estado del arte

Se realizó una búsqueda de estudios previos centrados en **modelos para la predicción de bancarrotas en entidades bancarias**, comparando distintas técnicas estadísticas y machine learning. A continuación, se resumen cuatro artículos clave que abordan el mismo problema planteado en este proyecto.

A. A Comparison of Alternative Bankruptcy Prediction Models — Wu, Gaunt & Gray (2010)

Este estudio compara cinco modelos clásicos de predicción de quiebras: análisis discriminante multivariado (Altman), regresión logística (Ohlson), regresión probit (Zmijewski), modelo hazard (Shumway) y el modelo basado en Black-Scholes-Merton (Hillegeist); utilizando datos de empresas listadas en NYSE y AMEX entre 1980 y 2006. El paradigma de aprendizaje utilizado es supervisado, enfocándose en clasificación binaria. Los autores desarrollan un modelo combinado que integra variables contables, de mercado y estructurales como la diversificación empresarial, lo cual permite obtener un mejor desempeño general. La validación se realizó con conjuntos *in-sample* y *out-of-sample*, evaluando el rendimiento mediante *accuracy* (exactitud), que se define como:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

donde TP y TN representan los verdaderos positivos y negativos, respectivamente. Los resultados muestran que el modelo combinado supera significativamente a los individuales [1].

B. Comparative Analysis of Data Mining Methods for Bankruptcy Prediction — Olson, Delen & Meng (2012)

Este trabajo evalúa técnicas de minería de datos para predecir bancarrotas, incluyendo árboles de decisión (CART, C5), redes neuronales (MLP, RBF), máquinas de soporte vectorial (SVM) y regresión logística, bajo un paradigma supervisado. Se aplicó validación cruzada estratificada de 10 *folds*, lo que permite una evaluación más robusta del rendimiento. Se utilizaron métricas como *accuracy*, además de medidas de complejidad del modelo. Cabe resaltar que, para el conjunto de datos manejado, el modelo de árboles de decisión ofreció una precisión de aproximadamente 95%. Además, se exploró la posibilidad de minimizar la cantidad de reglas manteniendo la precisión inicial, mediante el ajuste del parámetro de soporte mínimo [2].

C. An Application of Support Vector Machines in Bankruptcy Prediction Model — Shin, Lee & Kim (2005)

El estudio de Shin et al. (2005) se centró en la clasificación binaria de bancarrota bajo el paradigma supervisado, comparando máquinas de vectores de soporte (SVM) con redes neuronales de retropropagación (BPN). Los autores resaltaron la notable eficiencia de las SVM, debido a su capacidad para generalizar con menos datos de entrenamiento. Se utilizó validación con conjuntos separados de entrenamiento y prueba. La métrica principal fue *accuracy*, y se reportó que las SVM superaron a las BPN en cuanto a rendimiento predictivo, especialmente cuando los datos disponibles eran escasos [3].

D. Machine Learning Models and Bankruptcy Prediction — Barboza, Kimura & Altman (2017)

En esta investigación se evalúan modelos de aprendizaje automático, comparando enfoques modernos (Random Forest, Boosting, Bagging, SVM) con modelos tradicionales (análisis discriminante lineal, regresión logística y redes neuronales artificiales), todos bajo un enfoque supervisado. El conjunto de datos comprende más de 10,000 registros de empresas norteamericanas entre 1985 y 2013, extraídos de Compustat y Salomon Center. Se utilizó validación cruzada y conjuntos de prueba independientes. Las métricas utilizadas incluyen *accuracy*, AUC (Área bajo la curva ROC), y tasas de error tipo I (falsos positivos) y tipo II (falsos negativos). El modelo Random Forest obtuvo una precisión del 87%, superando en un 10% a los modelos convencionales [4].

IV. Entrenamiento y evaluacion de los modelos

La metodología de entrenamiento y evaluación de los modelos de Machine Learning que se ha usado en el presente, garantiza la robustez de los resultados y contempla los problemas relacionados con un conjunto de datos desbalanceado, como es la encontrada dentro del conjunto de datos el cual presenta una asimetría en la distribución de la clase objetivo ”Bankrupt?”: el 96.77% de las instancias corresponden a la clase 0 (No Bancarrota), mientras que solamente el 3.23% pertenecen a la clase 1 (Bancarrota). La estrategia aplicada para el entrenamiento de los modelos supliendo las necesidades son los siguientes:

A. Configuracion experimental

Inicialmente el conjunto de datos fue dividido en dos subconjuntos: un conjunto de entrenamiento y validación (*train-validation set*) y un conjunto de prueba (*test set*). Esta división se realizó para garantizar que la evaluación

final del rendimiento del modelo se llevara a cabo sobre datos no vistos en ninguna etapa del entrenamiento, ademas, se utilizó validación cruzada estratificada (Stratified K-Fold Cross Validation) para la etapa de entrenamiento y ajuste de hiperparámetros. Teniendo en cuenta que el problema es un desbalance asociado a la clase 1 (Bancarrota), la estratificación era necesaria para que cada *fold* de la validación cruzada tuviera la misma proporción de clases que el conjunto de datos original para que el modelo no presentara algun sesgo durante el entrenamiento de los modelos.

B. Pipeline de Preprocesamiento y Técnicas de Remuestreo

Se diseñó un pipeline con la librería 'imblearn' ('ImbPipeline') con el objetivo de aplicar la misma secuencia de transformaciones y tecnicas de remuestreo para evitar algun evento de data leakage (fuga de datos) de los datos de validación o test hacia el de entrenamiento.

El pipeline se estructuró de la siguiente manera:

1) **Escalado de Características:** A las características numéricas se les aplicó un escalado estándar para normalizarlas de una forma tal que tuvieran media cero y desviación típica uno.
2) **Balanceo de Clases:** Se recurre a la técnica combinada SMOTE-Tomek Link para reducir el desbalace de clases asosciado a la clase 1. Por un lado, SMOTE (Synthetic Minority Over-sampling Tecnique) va a generar datos sintéticos a partir del conjunto de la clase con menor numero de elementos, y por el otro lado, Tomek Links se va a encargar de la eliminacion de aquellos pares de instancias de clases contrarias que son los vecinos más cercanos. Este abordaje permite garantizar que el modelo se someta a un conjunto de datos balanceado en el entrenamiento de cada una de las folds, y al mismo tiempo, sin generar las muestras sintéticas en el testeo o la validación.
3) **Clasificador:** El modelo de Machine Learning final (ej., Random Forest, MLP, SVM) se colocó como el último paso del pipeline.

Es importante mencionar que las técnicas de remuestreo y escalado se ejecutan únicamente para los datos de entrenamiento dentro de cada fold de la validación cruzada y se aplican a los datos de validación.

C. Optimización de Hiperparámetros

La búsqueda de los hiperaprametros para cada uno de los modelos se llevó a cabo utilizando el Grid Search al mismo tiempo que Cross-Valiation, técnica que recorre una rejilla predefinida de combinaciones de hiperparámetros. Los rangos de parámetros evaluados para cada modelo se detallan en la Tabla I.

La métrica de evaluación que se utilizó para seleccionar los mejores hiperparámetros durante esta fase de la resolución del problema fue el F1-Score debido a su capacidad de solventar situaciones relacionedos a problemas de clasificación con desequilibrio de clases, ya que tiene en cuenta tanto la precisión como el recall de la clase con menor numero de elementos.

D. Evaluación Final del Modelo

Una vez escogido el modelo más óptimo, se llevó a cabo el entrenamiento del modelo utilizando el pipeline completo con ajustes los hiperparámetros con mejor rendimiento con respecto a la metrica usada y se evaluó el rendimiento del modelo con el conjunto de prueba (*test set*), y cuyas métricas que se consideraron para la evaluación fueron las siguientes:

- **Accuracy:** Es la métrica más intuitiva y mide la proporción de predicciones correctas sobre el total de predicciones.

$$\text{Accuracy} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}$$

  Aunque la Accuracy nos proporciona una visión de conjunto del rendimiento del modelo pero para un conjunto de datos de clases desbalanceadas se puede convertir en una métrica un poco fiable. Un modelo podría alcanzar una alta exactitud simplemente prediciendo la clase mayoritaria, por lo anterior no usa como métrica principal para la optimización y selección de modelos.

- **Precision:** mide la proporción de clases positivas que fueron realmente correctas.

$$\text{Precision} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

  La capacidad de detección de bancarrotas es de suma importancia en este problema debido a la presencia "falsos negativos" (no detectar una bancarrota real) que en terminos financieros y economicos generaría efectos muy desfavorables. Es mejor identificar la mayoría de las bancarrotas reales, aunque esto incremente la cantidad de falsos positivos.

- **F1-Score:** Métrica principal para problemas desbalanceados, representa la media armónica entre precisión y recall.

$$\text{F1-Score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

  Debido a que el F1-Score se estila como métrica principal para finalizar la optimización de hiperparámetros y la comparativa de modelos, al ser la media armónica de la Precisión y del Recall no hay favoritismo por alguna medida.

- **ROC AUC:** la metrica Receiver Operating Characteristic Area Under the Curve (ROC AUC) mide la capacidad del clasificador para distinguir correctamente entre las clases, asignando mayores probabilidades a las instancias positivas reales que a las instancias negativas. Se usará como metrica de calidad del clasificador.

V. Análisis de Resultados del Entrenamiento de Modelos

Esta sección presenta el análisis y la comparación de seis modelos de clasificación entrenados y optimizados para predecir la bancarrota. Se evalúa su rendimiento en métricas

TABLE I
MODELOS EVALUADOS Y SUS HIPERPARÁMETROS PARA OPTIMIZACIÓN

Nombre del Modelo	Clase del Modelo	Parámetros a probar
Logistic Regression	`LogisticRegression`	`C: [0.01, 0.1, 1], penalty: [l1, l2], solver: liblinear`
K-Nearest Neighbors	`KNeighborsClassifier`	`n neighbors: [3, 5, 7, 9], weights: [uniform, distance], metric: [euclidean, manhattan]`
Random Forest	`RandomForestClassifier`	`n estimators: [10, 20, 50], max features: [sqrt, log2], max depth: [10, 20, 30], min samples split: [2, 5]`
MLP Classifier	`MLPClassifier`	`hidden layer sizes: [(64,), (64, 32)], activation: [relu, tanh], solver: adam, alpha: [0.0001, 0.001], learning rate init: [0.001, 0.01]`
Gradient Boosting	`GradientBoostingClassifier`	`n estimators: [10, 50], learning rate: [0.01, 0.1], max depth: [3, 5], subsample: [0.8, 1.0]`
Support Vector Machine	`SVC`	`C: [0.1, 1], kernel: rbf, gamma: scale`

clave para identificar los modelos más prometedores para la siguiente fase de selección de características, según lo estipulado en la sección 5.1 [**?**] imagen.

A. *Análisis de Resultados del Modelo: Regresión Logística*

Mejores hiperparámetros: `C=0.1`, `penalty=l1` y `solver=liblinear`. El F1-Score en entrenamiento (0.3356 ± 0.014) y validación (0.2902 ± 0.036) mostró un leve sobreajuste. En prueba, el F1-Score fue 0.319, con un excelente ROC AUC = 0.9361. Destacó por su alto Recall (0.8409) y baja Precisión (0.1968). Su desempeño se muestran en las Figuras 1, 2 y 3.

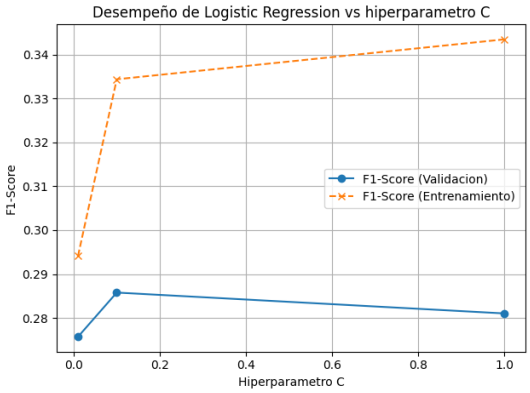

Fig. 1. Desempeño de F1-Score en entrenamiento y validación para Regresión Logística en función del hiperparámetro C.

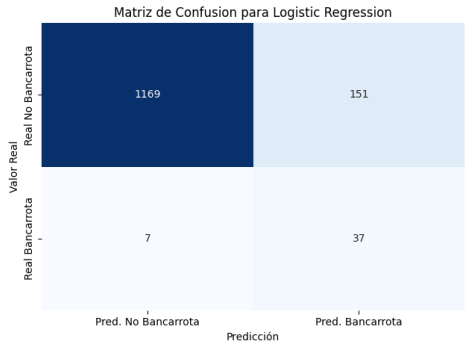

Fig. 2. Matriz de Confusión del modelo de Regresión Logística en el conjunto de prueba.

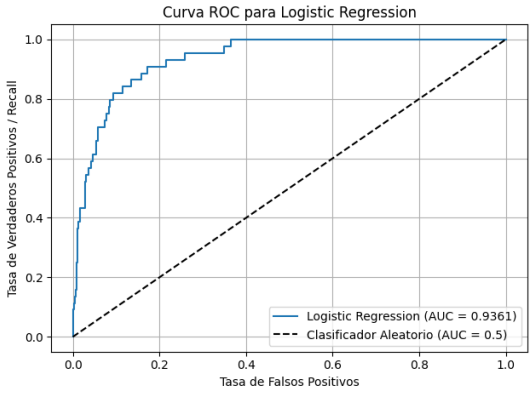

Fig. 3. Curva ROC del modelo de Regresión Logística en el conjunto de prueba.

B. *Análisis de Resultados del Modelo: K-Nearest Neighbors*

Mejores hiperparámetros: `metric=manhattan` y `n_neighbors=3`. Se observó un sobreajuste considerable con F1-Score entrenamiento: 0.587 ± 0.026 vs. validación: 0.312 ± 0.029 El F1-Score en prueba fue de 0.3616, con un ROC AUC de 0.8481. Su desempeño se muestran en las Figuras 4, 5 y 6.

C. *Análisis de Resultados del Modelo: Random Forest*

Mejores hiperparámetros: `max_depth=30`, `max_features=sqrt`, `min_samples_split=5` y `n_estimators=20`. Presentó alto sobreajuste con F1-Score entrenamiento: 0.9806 ± 0.011 vs. validación: 0.4206 ± 0.043. Aún así, obtuvo el F1-Score más alto (0.4727) y un excelente ROC AUC de 0.9389 en prueba. Evidenciando la Precisión (0.3939) sobre el Recall (0.5909). Su desempeño se muestran en las Figuras 7, 8 y 9.

D. *Análisis de Resultados del Modelo: MLP Classifier*

Mejores hiperparámetros: `activation=relu`, `alpha=0.0001`, `hidden_layer_sizes=(64,)` y `learning_rate_init=0.01`. Presentó alto sobre ajuste con F1-Score entrenamiento: 1.0 ± 0.0 vs. validación: 0.3543 ± 0.036. El F1-Score en prueba fue 0.3953, con el ROC AUC más bajo (0.8289). Su alta Precisión (0.4048) se compensó

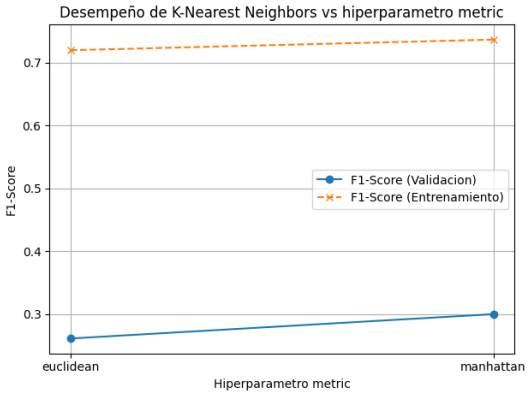

Fig. 4. Desempeño de F1-Score en entrenamiento y validación para K-Nearest Neighbors en función del hiperparámetro metric.

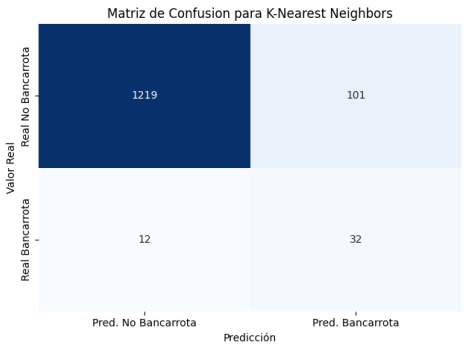

Fig. 5. Matriz de Confusión del modelo K-Nearest Neighbors en el conjunto de prueba.

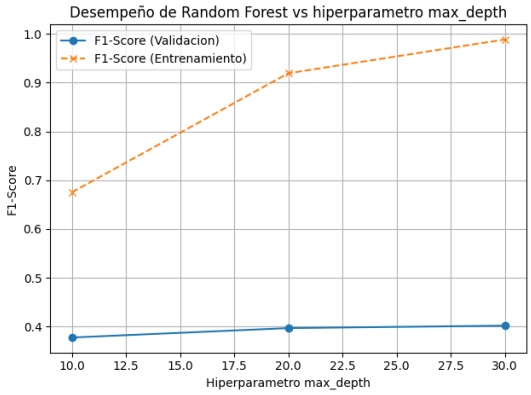

Fig. 7. Desempeño de F1-Score en entrenamiento y validación para Random Forest en función del hiperparámetro max_depth.

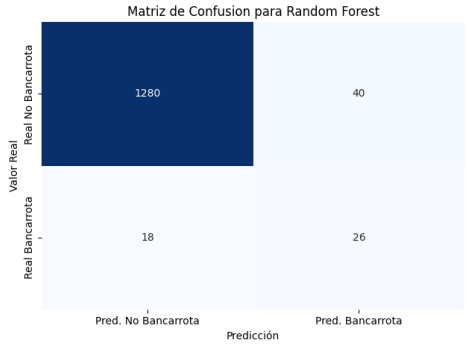

Fig. 8. Matriz de Confusión del modelo Random Forest en el conjunto de prueba.

con un bajo Recall (0.3864). Su desempeño se detalla en las Figuras 10, 11 y 12.

E. Análisis de Resultados del Modelo: Gradient Boosting

Mejores hiperparámetros: `learning_rate=0.1`, `max_depth=5`, `n_estimators=50` y `subsample=0.8`. Se observó sobreajuste moderado con F1-Score entrenamiento: 0.7633 ± 0.017 vs. validación: 0.3875 ± 0.030. En prueba, obtuvo el ROC AUC más alto (0.9454) y un F1-Score competitivo de 0.4341. Su comportamiento se detalla en las Figuras 13, 14 y 15.

F. Análisis de Resultados del Modelo: Support Vector Machine (SVM)

Mejores hiperparámetros: `C=0.1` y `kernel=linear`. Sus F1-Scores en entrenamiento (0.1116 ± 0.065) y validación (0.1075 ± 0.071) fueron extremadamente bajos e inestables. En prueba, obtuvo el F1-Score más bajo (0.09799) y una Precisión extremadamente baja (0.05186), a pesar de un Re-

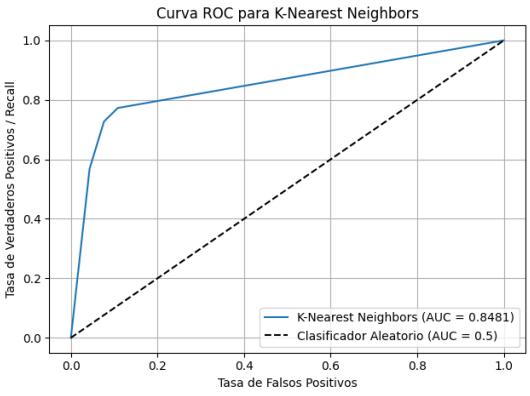

Fig. 6. Curva ROC del modelo K-Nearest Neighbors en el conjunto de prueba.

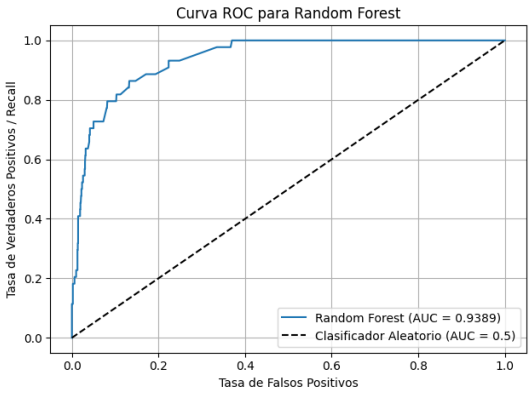

Fig. 9. Curva ROC del modelo Random Forest en el conjunto de prueba.

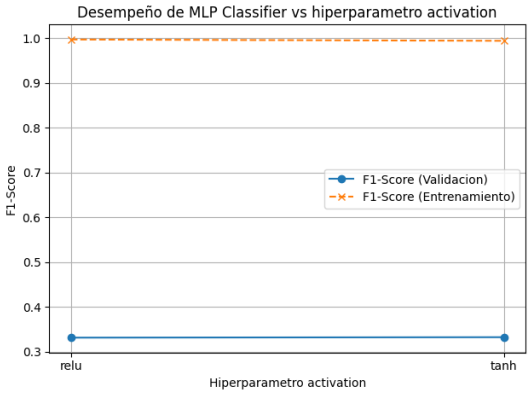

Fig. 10. Desempeño de F1-Score en entrenamiento y validación para MLP Classifier en función del hiperparámetro activation.

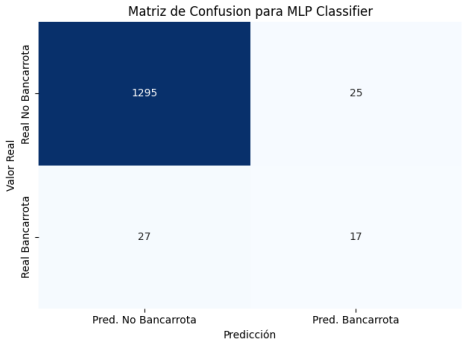

Fig. 11. Matriz de Confusión del modelo MLP Classifier en el conjunto de prueba.

call muy alto (0.8864). Su Accuracy general fue muy baja (0.4736). Su comportamiento se detalla en las Figuras 16, 17 y 18.

G. Comparación Global de Modelos

La Tabla II y la Figura 19 resumen el desempeño de todos los clasificadores en el conjunto de prueba.

1) Discusión Comparativa y Modelos Top:

- **F1-Score:** El Random Forest (0.472727) consiguió el F1-Score más alto, seguido del Gradient Boosting (0.434109), lo que indica que lograron el mejor compromiso entre el recall y la precisión.
- **ROC AUC:** En este caso, el modelo Gradient Boosting (0.945351) se llevó el AUC más alto, mostrando la mejor capacidad discriminativa, seguido por Random Forest (0.938895) y Logistic Regression (0.936054).
- **Precisión vs. Recall:** Logistic Regression (0.840909) y SVM (0.886364) lograron un Recall muy alto, detectando la mayoría de las bancarrotas. Sin embargo, SVM presentó una Precisión extremadamente baja (0.051862),

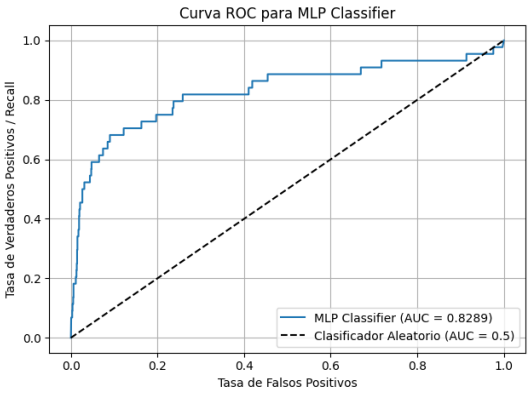

Fig. 12. Curva ROC del modelo MLP Classifier en el conjunto de prueba.

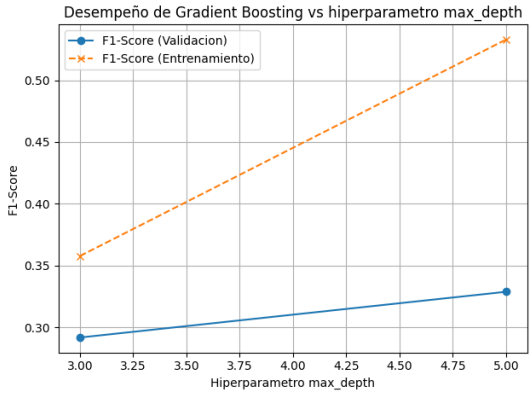

Fig. 13. Desempeño de F1-Score en entrenamiento y validación para Gradient Boosting en función del hiperparámetro max_depth.

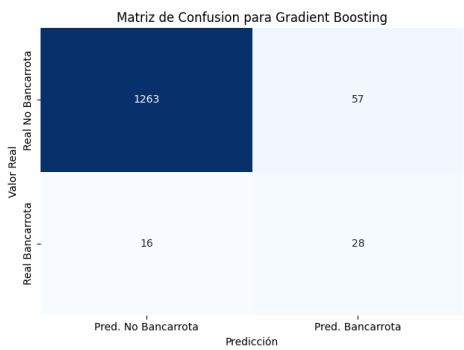

Fig. 14. Matriz de Confusión del modelo Gradient Boosting en el conjunto de prueba.

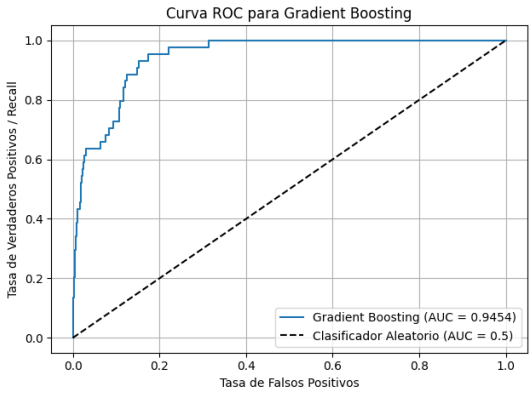

Fig. 15. Curva ROC del modelo Gradient Boosting en el conjunto de prueba.

TABLE II
RESUMEN DE MÉTRICAS DE DESEMPEÑO EN EL CONJUNTO DE PRUEBA PARA TODOS LOS MODELOS

Modelo	Accuracy	Precision	Recall	F1-Score	ROC AUC
Random Forest	0.957478	0.393939	0.590909	0.472727	0.938895
Gradient Boosting	0.946481	0.329412	0.636364	0.434109	0.945351
MLP Classifier	0.961877	0.404762	0.386364	0.395349	0.828934
K-Nearest Neighbors	0.917155	0.240602	0.727273	0.361582	0.848063
Logistic Regression	0.884164	0.196809	0.840909	0.318966	0.936054
Support Vector Machine	0.473607	0.051862	0.886364	0.097990	0.803426

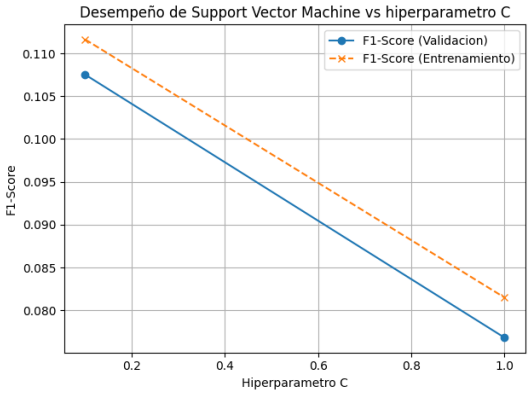

Fig. 16. Desempeño de F1-Score en entrenamiento y validación para Support Vector Machine en función del hiperparámetro C.

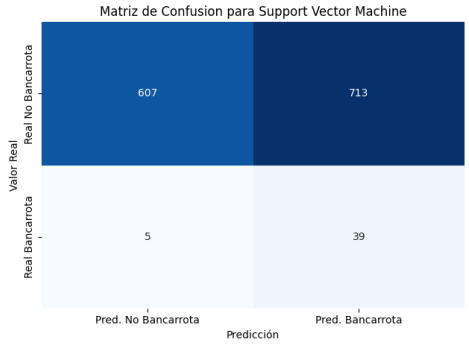

Fig. 17. Matriz de Confusión del modelo Support Vector Machine en el conjunto de prueba.

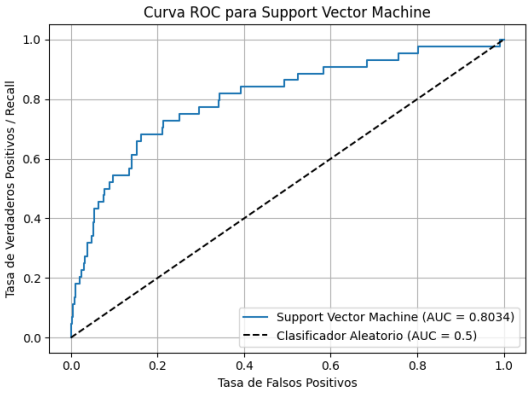

Fig. 18. Curva ROC del modelo Support Vector Machine en el conjunto de prueba.

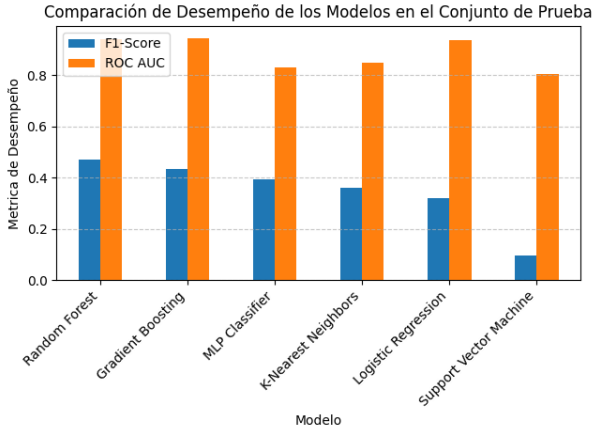

Fig. 19. Comparación de Desempeño de F1-Score y ROC AUC de los Modelos en el Conjunto de Prueba.

haciéndolo inviable. El MLP Classifier ha tenido la precisión más alta (0.404762), pero un bajo del Recall (0.386364). Por otro lado, Random Forest y Gradient Boosting nos han permitido tener una buena compensación.

- **Sobreajuste:** Random Forest, MLP Classifier y Gradient Boosting mostraron un sobreajuste importante sugiriendo la necesidad de mayor regularización para mejorar la generalización, por otro lado, el modelo SVM también fue extremadamente inestable.

Tomando el F1-Score como métrica principal para balancear Precisión y Recall en un escenario de clases desequilibradas, vemos que los 2 mejores modelos predictivos para la selección son:

- **Random Forest:** Por tener aún mejor F1-Score y un perfecto ROC AUC.
- **Gradient Boosting:** Por su ROC AUC excepcional y un F1-Score muy competitivo.

VI. REDUCCIÓN DE DIMENSIONALIDAD

La reducción de la dimensionalidad es fundamental para atenuar el problemas relacionados con los problemas de alta dimensionalidad (curse of dimensionality) e incrementar la eficiencia. Se aplicaron tres métodos: correlación de Pearson, analisis de varianza y seleccion de caracteristicas.

A. Análisis de Correlación de Pearson

Este análisis identifica características con correlación lineal muy baja (< 0.01) con respecto a la variable objetivo,

mencionando una poca influencia en la predicción. Las características candidatas a ser eliminadas por su baja correlación se contemplan en la Tabla III.

TABLE III
CARACTERÍSTICAS CANDIDATAS A ELIMINAR POR CORRELACIÓN DE PEARSON (< 0.01)

Característica	Correlación de Pearson
After-tax net Interest Rate	-0.009647
Continuous interest rate (after tax)	-0.009364
Pre-tax net Interest Rate	-0.009313
Accounts Receivable Turnover	-0.008084
No-credit Interval	-0.007949
Total income/Total expense	-0.006761
Average Collection Days	-0.006636
Allocation rate per person	0.005860
Long-term Liability to Current Assets	0.005183
Inventory/Current Liability	0.005037
Interest Expense Ratio	0.004718
Interest Coverage Ratio (Interest expense to EBIT)	-0.004492
Quick Assets/Current Liability	-0.004274
Working capitcal Turnover Rate	-0.004192
Revenue Per Share (Yuan ¥)	-0.004038
Cash Flow to Sales	-0.003902
Continuous Net Profit Growth Rate	-0.003852
Inventory/Working Capital	-0.003241
Net Value Growth Rate	-0.002472
Revenue per person	-0.002472
Inventory Turnover Rate (times)	-0.002436
Operating Profit Rate	-0.000146

B. Análisis de Varianza

Este análisis identifica características con varianza extremadamente baja (< 0.001), las cuales aportan poca información al modelo y pueden ser eliminadas. Las características candidatas a ser eliminadas por su baja varianza se contemplan en la Tabla IV.

C. Selección Secuencial de Características (Sequential Feature Selection - SFS)

SFS busca el mejor subconjunto de características añadiendo/eliminando iterativamente características dentro de cada modelo apartir de la metrica F1-Score como criterio de seleecion. Resultados de SFS:

- Características iniciales: 95
- Características seleccionadas: 44
- Reducción: 53.68%

D. Reentrenamiento de Modelos con Características Reducidas

Los modelos Random Forest y Gradient Boosting, los mejores identificados previamente, fueron reentrenados con el subconjunto de 44 características de SFS. Se observa que para las metricas de seleccion F1-Score se mantuvo constante mientras que la medición ROC AUC disminuyó de forma más notable. Los resultados son contemplado en la TablaV.

TABLE IV
CARACTERÍSTICAS CANDIDATAS A ELIMINAR POR VARIANZA (< 0.001)

Característica	Varianza
Operating Gross Margin	0.000291
Realized Sales Gross Margin	0.000290
Operating Profit Rate	0.000184
Pre-tax net Interest Rate	0.000132
After-tax net Interest Rate	0.000110
Non-industry income and expenditure/revenue	0.000135
Continuous interest rate (after tax)	0.000123
Cash flow rate	0.000241
Net Value Per Share (B)	0.000975
Net Value Per Share (A)	0.000979
Net Value Per Share (C)	0.000979
Cash Flow Per Share	0.000291
Operating Profit Per Share (Yuan ¥)	0.000776
Per Share Net profit before tax (Yuan ¥)	0.000976
Realized Sales Gross Profit Growth Rate	0.000005
Operating Profit Growth Rate	0.000138
After-tax Net Profit Growth Rate	0.000220
Regular Net Profit Growth Rate	0.000222
Continuous Net Profit Growth Rate	0.000126
Total Asset Return Growth Rate Ratio	0.000014
Cash Reinvestment %	0.000380
Current Ratio	0.000540
Interest Expense Ratio	0.000126
Long-term fund suitability ratio (A)	0.000781
Borrowing dependency	0.000205
Contingent liabilities/Net worth	0.000185
Operating profit/Paid-in capital	0.000767
Net profit before tax/Paid-in capital	0.000888
Inventory and accounts receivable/Net value	0.000162
Operating profit per person	0.000988
Inventory/Working Capital	0.000121
Working Capital/Equity	0.000154
Current Liabilities/Equity	0.000175
Retained Earnings to Total Assets	0.000683
Total income/Total expense	0.000183
Total expense/Assets	0.000763
Working capitcal Turnover Rate	0.000032
Cash Flow to Sales	0.000024
Current Liability to Equity	0.000175
Equity to Long-term Liability	0.000338
Cash Flow to Liability	0.000889
Cash Flow to Equity	0.000183
Current Liability to Current Assets	0.000864
Liability-Assets Flag	0.000916
No-credit Interval	0.000151
Gross Profit to Sales	0.000291
Net Income to Stockholder's Equity	0.000117
Liability to Equity	0.000191
Degree of Financial Leverage (DFL)	0.000302
Interest Coverage Ratio (Interest expense to EBIT)	0.000205
Net Income Flag	0.000000

TABLE V
RESULTADOS DE RENDIMIENTO DE MODELOS CON SFS

Modelo	Métrica	Valor con SFS	Valor Original
Random Forest	Dimension	(5455, 44)	(5455, 95)
	Mejor F1-Score CV	0.4113	-
	F1-Score en Test	0.4746	0.4727
	ROC AUC en Test	0.9241	0.9389
Gradient Boosting	Mejor F1-Score CV	0.3734	-
	F1-Score en Test	0.4295	0.4341
	ROC AUC en Test	0.9285	0.9454

E. Conclusiones de la Reducción de Dimensionalidad

La reducción de dimensionalidad con SFS alcanzó una reducción del 53.68% de características, y ha sido clave para reducir la complejidad del modelo, por otro lado, el reentrenamiento de los mejores modelos adquiridos (Random Forest y Gradient Boosting) demostraron una ligeramente mejora bajo la metrica F1-Score, que apesar de una leve disminucion en la medida en ROC AUC el resultado conseguido sigue siendo suficiente, demostrando el aporte del SFS en la selección de caracteristicas.

F. Análisis de Componentes Principales (PCA)

PCA se utilizó para llevar a cabo la reducción de la dimensión, lo cual se logra transformando las características originales en una nueva serie de componentes no correlacionados entre ellos, conservando así la mayor varianza posible. Para ello, se eligieron 50 componentes principales donde se conservan el 95% de la varianza explicada, tal como se puede observar en la figura. 20 y una reduccion de dimensionalidad:

- Características iniciales: 95
- Características seleccionadas: 50
- Reducción: 47.37%

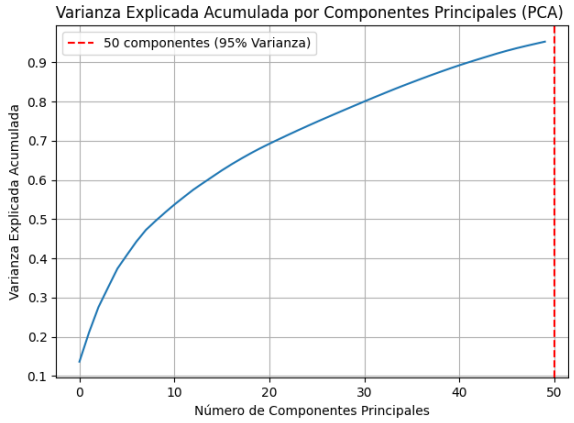

Fig. 20. Varianza Explicada Acumulada por Componentes Principales (PCA)

G. Reentrenamiento de Modelos con Características Reducidas (PCA)

Los modelos Random Forest y Gradient Boosting, los mejores identificados previamente, fueron reentrenados utilizando los datos transformados por PCA.De manera similar, Gradient Boosting también experimentó una disminución en ambas métricas de rendimiento al usar las componentes de PCA. Se observa el rendimiento contemplado por el reentrenamiento de los modelos en la TablaVI.

H. Conclusiones de la Reducción de Dimensionalidad (PCA)

La aplicación de PCA hizo descender la dimensionalidad hasta 50 componentes, logrando explicar un 95% de la varianza. No obstante, este reentrenamiento de los modelos de Random Forest y de Gradient Boosting con los componentes transformados supuso una caída en el F1-Score y el ROC AUC respecto a su rendimiento original. Esto nos advertiría que, para este conjunto de datos y los modelos, la transformación lineal que se da en PCA, a pesar de que es efectiva en reducir la dimensionalidad en función de la varianza, quizás no mantiene la capacidad predictiva de la misma forma que otros métodos de selección de características.

TABLE VI
RESULTADOS DE RENDIMIENTO DE MODELOS CON PCA

Modelo	Métrica	Valor con PCA
Random Forest	Dim. Entrenamiento	(5455, 50)
	Dim. Prueba	(1364, 50)
	Mejor F1-Score CV (PCA)	0.3630
	F1-Score en Test (PCA)	0.3551
	ROC AUC en Test (PCA)	0.9036
Gradient Boosting	Mejor F1-Score CV (PCA)	0.3124
	F1-Score en Test (PCA)	0.3699
	ROC AUC en Test (PCA)	0.9137

VII. CONCLUSIONES

Este estudio exhaustivo abordó la predicción de la bancarrota de empresas, un problema crítico en finanzas. Se exploraron múltiples modelos de clasificación, seleccion de caracteristicas relevantes y técnicas de reducción de dimensionalidad, enfocando la optimización alrededor de la metrica F1-Score debido al desbalance de clases. Los resultados de la fase inicial se resumen en la siguiente tabla:

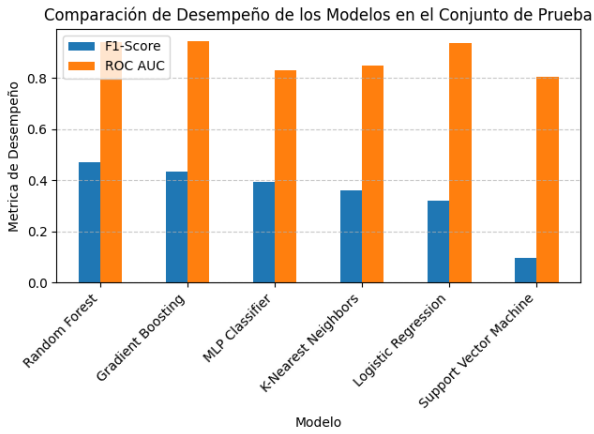

Fig. 21. Comparación de Desempeño de los Modelos en el Conjunto de Prueba (F1-Score y ROC AUC)

Como se observa en la figura 21, Tanto el modelo Random Forest como el Gradient Boosting demostraron ser los que ofrecían un rendimiento general más óptimo frente al conjunto de prueba en F1-Score y ROC AUC, apartir de esto, se seleccionaron para las siguientes fases de optimización.

TABLE VII
TABLA COMPARATIVA DE DESEMPEÑO (ORIGINAL, SFS, PCA)

Modelo	Método	Reducción (%)	F1-Score	ROC AUC	Precision	Recall	Accuracy
Gradient Boosting	Control	0.00	0.4341	0.9454	0.3294	0.6364	0.9465
	SFS	53.68	0.4295	0.9285	0.3048	0.7273	0.9377
	PCA	47.37	0.3699	0.9137	0.2647	0.6136	0.9326
Random Forest	Control	0.00	0.4727	0.9389	0.3939	0.5909	0.9575
	SFS	53.68	0.4746	0.9241	0.3784	0.6364	0.9545
	PCA	47.37	0.3551	0.9036	0.3016	0.4318	0.9494

Se utilizaron dos métodos de reducción de dimensionalidad: Selección Secuencial de Características (SFS) y PCA (Análisis de componentes principales) para que siguiente a su aplicacion contemplar los efectos de estos métodos de reducción de la dimensionalidad en el rendimiento de los modelos Random Forest y Gradient Boosting. En el caso del PCA, se seleccionaron 50 componentes principales los cuales explican el 95% de la varianza total que se puede observar en la figura20 de Varianza Explicada Acumulada.

A. Comparación General del Impacto de la Reducción de Dimensionalidad

Se presenta una tabla comparativa y gráficos que incluyen los resultados de los modelos sin reducción (control), con SFS y con PCA. Podemos visualizar dichos resultados en la tabla VII.

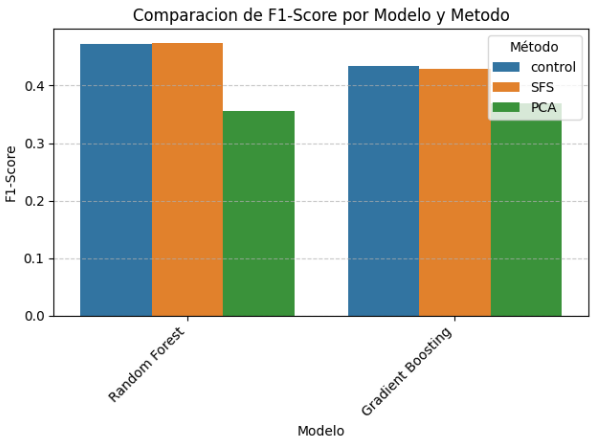

Fig. 22. Comparación de F1-Score por Modelo y Método

Como se observa en la Tabla VII y las Figuras 22 y 23:

- Para el modelo Random Forest, se observó que SFS genera un F1-Score muy ligeramente por encima del modelo control (0.4746 vs 0.4727) con un ROC AUC muy alto (0.9241) mientras que, por otro lado, PCA llevó a una caída más importante en F1-Score (0.3551) y en ROC AUC (0.9036).

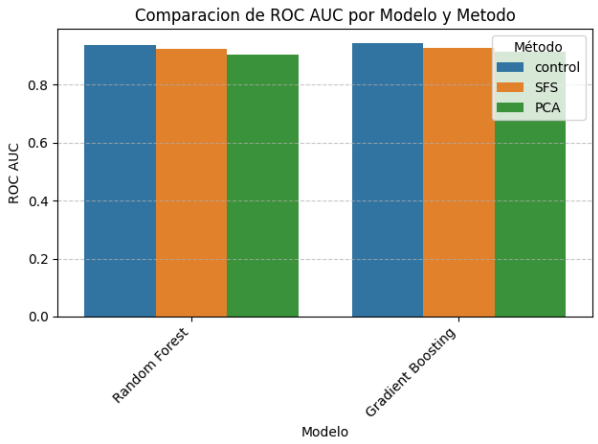

Fig. 23. Comparación de ROC AUC por Modelo y Método

- Para el modelo Gradient Boosting, SFS mantuvo el F1-Score muy similar al del modelo control (0.4295 vs 0.4341), pero tuvo una leve caída en ROC AUC (0.9285 vs 0.9454) aunque PCA, al igual que en el caso de Random Forest también hablaría de un fallo en el rendimiento en cuanto a F1-Score (0.3699) y un ROC AUC (0.9137) que está muy por debajo de los resultados del modelo control.

VIII. CONCLUSIONES GENERALES

A pesar de la reducción significativa lograda con PCA (una reducción de características de 47.36%) manteniendo la alta varianza de la representación de las características con este nuevo espacio de representación, la calidad de los aprendices se redujo con respecto a los modelos originales y los optimizados con SFS, lo que denota que para la clasificación de bancarrota la información predictiva se mantiene mejor conservada cuando se seleccionan las características originales más relevantes de forma directa (SFS) que al transformarlas linealmente en nuevas dimensiones (en PCA).

El modelo Random Forest fue el que tuvo mejor rendimiento global en esta discriminacón para la clasificación de bancarrota. Además, la técnica SFS es la que se considera más poderosa a la hora de conservar y mejorar la propiedad

predictiva de los modelos, alcanzando el mejor resultado en comparación a PCA y con los modelos base. Por su parte, aunque PCA es una herramienta útil para la reducción de la cifra de dimensiones, ha quedando demostrado que la SFS es la más potente en este caso específico y para este conjunto de datos.

REFERENCES

[1] Y. Wu, C. Gaunt, and S. Gray, "A comparison of alternative bankruptcy prediction models," *Accounting & Finance Association of Australia and New Zealand (AFAANZ) Conference*, 2010.

[2] D. L. Olson, D. Delen, and Y. Meng, "Comparative analysis of data mining methods for bankruptcy prediction," *Computers & Operations Research*, vol. 39, no. 2, pp. 291-300, 2012.

[3] K.-S. Shin, T. S. Lee, and H.-J. Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications*, vol. 28, no. 1, pp. 127-135, 2005.

[4] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, pp. 400-417, 2017.