MIAD

Maestría
en Inteligencia
Analítica de Datos

Contenido

Grupo 13 - Deep Learning

Informe Miniproyecto Predicción de Bancarrota empresarial

1) Exploración de los datos para su entendimiento dentro del contexto organizacional

a) Contexto:

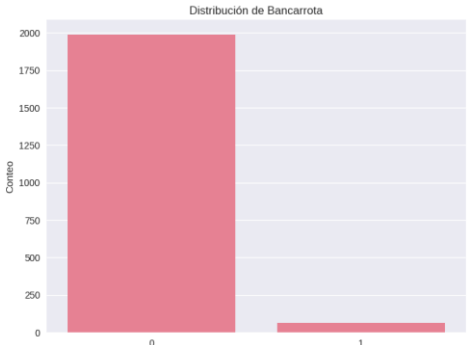
El objetivo de este proyecto es predecir si una empresa se declarará en bancarrota en el año siguiente, utilizando razones financieras derivadas de sus estados contables. Esto tiene un gran valor estratégico para bancos, aseguradoras, fondos de inversión y otros actores del sistema financiero, ya que les permite anticipar riesgos crediticios y ajustar sus decisiones de manera proactiva.

Los datos utilizados provienen de empresas listadas en el **Shanghai Stock Exchange** entre 1999 y 2009. Cada observación representa una empresa en un año determinado y contiene **63 razones financieras**, además de un identificador y una variable binaria (Bankruptcy) que indica si la empresa quebró posteriormente.

b) Estructura general de los datos

- Conjunto de entrenamiento: 2.050 registros y 65 columnas
- No se encontraron valores nulos ni duplicados en la base de datos
- Conjunto de prueba: 500 registros y 64 columnas
- Distribución de clases en la Variable objetivo (Bankruptcy):
  - 0: Empresa saludable – 96.97%
  - 1: Empresa quebrada – 3.02%

Esta proporción revela un desbalance severo de clases, lo cual será un reto importante en el modelamiento.




```
Proporciones de bancarrota:
Bankruptcy
0    0.969756
1    0.030244
Name: proportion, dtype: float64
```

Ilustración 1Distribucion de clases en variable objetivo

c) Estadística descriptiva

Con el objetivo de entender la estructura de los datos y evaluar posibles relaciones entre las variables, inicialmente realizamos un análisis de correlaciones entre todas las variables independientes del conjunto de datos. El propósito de este ejercicio fue identificar relaciones fuertes que pudieran derivar en problemas de multicolinealidad o redundancia de información en etapas posteriores del modelado.

A partir de este análisis, se extraen las 5 correlaciones positivas y las 5 correlaciones negativas más significativas. Entre las correlaciones positivas más fuertes destacan variables como Inventory.and.accounts.receivable.Net.value y Pre.tax.net.Interest.Rate, con correlaciones superiores al 0.99. Por el lado de las correlaciones negativas, observé relaciones importantes como la existente entre Fixed.Assets.to.Assets y Total.debt.Total.net.worth, con coeficientes cercanos a -0.56.

Este primer paso permitió identificar variables altamente relacionadas que, de ser necesario, podrían ser tratadas para evitar redundancias para mejorar la estabilidad de los modelos y eventualmente facilitar la aplicación de técnicas de reducción de dimensionalidad si fuera necesario.

Posteriormente, se realiza un análisis descriptivo de las variables involucradas, observando características generales como promedios, desviaciones estándar y rangos de valores, lo cual brindó un contexto adicional sobre su comportamiento dentro de la muestra.

Tabla 1 Analítica descriptiva top correlaciones.

Estadísticas descriptivas de las variables top correlacionadas:

	Working.Capital.Equity	Total.debt.Total.net.worth	Interest.Coverage.Ratio..Interest.expense.to.EBIT.	Operating.Profit.Rate	Equity.to.Liability	Inventory.Working.Capital
count	2050.000000	2.050000e+03	2050.000000	2050.000000	2050.000000	2050.000000
mean	0.773409	1.063234e+06	0.622023	0.999136	0.087459	0.351283
std	0.046777	3.442681e+07	0.070904	0.000669	0.076315	0.086672
min	0.509761	0.000000e+00	0.000000	0.979597	0.006851	0.213573
25%	0.736877	5.261682e-03	0.567458	0.999021	0.035608	0.279768
50%	0.770761	1.336565e-02	0.617337	0.999183	0.067370	0.339278
75%	0.811126	3.181698e-02	0.677945	0.999359	0.116572	0.416895
max	0.861683	1.256262e+09	0.775476	0.999790	1.000000	0.539227

Al analizar los resultados vemos lo siguiente:

Working.Capital.Equity presentó valores bastante concentrados, con una media de 0.77 y una desviación estándar de apenas 0.046, lo cual indica poca dispersión relativa entre las empresas analizadas.

Total.debt.Total.net.worth mostró una media de aproximadamente 1 millón, pero con una desviación estándar extremadamente alta (más de 34 millones), indicando una distribución fuertemente sesgada con presencia de valores extremos o outliers.

Operating.Profit.Rate fue una de las variables más estables, con una media cercana a 0.999 y una variación mínima.

Otras variables como Quick.Assets.Total.Assets o Fixed.Assets.to.Assets evidenciaron mayor dispersión, lo cual podría influir en la heterogeneidad de los datos.

Tabla 2 Estadística descriptiva top correlaciones con la variable objetivo

Estadísticas descriptivas de las variables más correlacionadas con la bancarrota:

	Debt.ratio..	Current.Liability.to.Current.Assets
count	2050.000000	2050.000000
mean	0.171416	0.064862
std	0.094093	0.051282
min	0.000000	0.000900
25%	0.101543	0.024671
50%	0.156706	0.049630
75%	0.234865	0.093878
max	0.497667	0.318429

	Current.Liability.to.Assets	Liability.Assets.Flag	Revenue.per.person
count	2050.000000	2050.000000	2.050000e+03
mean	0.145461	0.000488	4.319056e+03
std	0.089936	0.022086	1.955506e+05
min	0.000621	0.000000	1.732141e-04
25%	0.075752	0.000000	1.749331e-02
50%	0.128409	0.000000	3.899822e-02
75%	0.200866	0.000000	8.245015e-02
max	0.509076	1.000000	8.853931e+06

	Retained.Earnings.to.Total.Assets	Working.Capital.to.Total.Assets
count	2050.000000	2050.000000
mean	0.944661	0.841137
std	0.024380	0.062363
min	0.663326	0.628993
25%	0.936059	0.799064
50%	0.947768	0.844362
75%	0.958491	0.884139
max	1.000000	0.987732

	ROA.B..before.interest.and.depreciation.after.tax	Tax.rate..A.
count	2050.000000	2050.000000
mean	0.610469	0.149762
std	0.087063	0.167852
min	0.000000	0.000000
25%	0.549875	0.000000
50%	0.610791	0.107021
75%	0.674470	0.258826
max	0.953532	0.966891

	Quick.Assets.Total.Assets
count	2050.000000
mean	0.462103
std	0.211477
min	0.004725
25%	0.295769
50%	0.458141

Posteriormente nos enfocamos en las correlaciones con la variable objetivo para esto, calculamos las correlaciones de todas las variables independientes respecto a Bankruptcy, extrayendo nuevamente las 5 variables con correlaciones positivas más altas y las 5 con correlaciones negativas más fuertes.

Dentro de las variables que presentan correlaciones positivas más importantes con la bancarrota se encuentran Debt.ratio.. y Current.Liability.to.Current.Assets, lo cual tiene sentido conceptual, ya que altos niveles de deuda o de pasivo corriente en proporción a los activos suelen estar asociados a un mayor riesgo de quiebra.

En cuanto a las correlaciones negativas, destacan variables como Retained.Earnings.to.Total.Assets y Working.Capital.to.Total.Assets, indicando que mayores niveles de utilidades retenidas o capital de trabajo en proporción a los activos están asociados a un menor riesgo de bancarrota.

Ya al ver el análisis descriptivo evidenciamos lo siguiente:

- **Debt.ratio.**: presenta una media del 17%, mostrando que en promedio las empresas mantienen un nivel relativamente bajo de endeudamiento sobre el total de activos.
- **Working.Capital.to.Total.Assets**: muestra una media cercana al 84%, lo cual indica una buena proporción de capital de trabajo respecto a activos totales en la mayoría de las observaciones.
- **Retained.Earnings.to.Total.Assets**: tiene valores bastante altos, con una mediana del 94%, reflejando que muchas empresas analizadas han retenido una proporción importante de sus ganancias.

Para entender qué variables son más influyentes, se analizaron visual y estadísticamente algunas razones financieras relevantes:

Tasa de beneficio operativo (Operating.Profit.Rate)

Empresas quebradas tienden a tener menores beneficios operativos. Muestra diferencias sutiles pero consistentes, con menor densidad en la zona alta de rentabilidad para empresas en bancarrota.

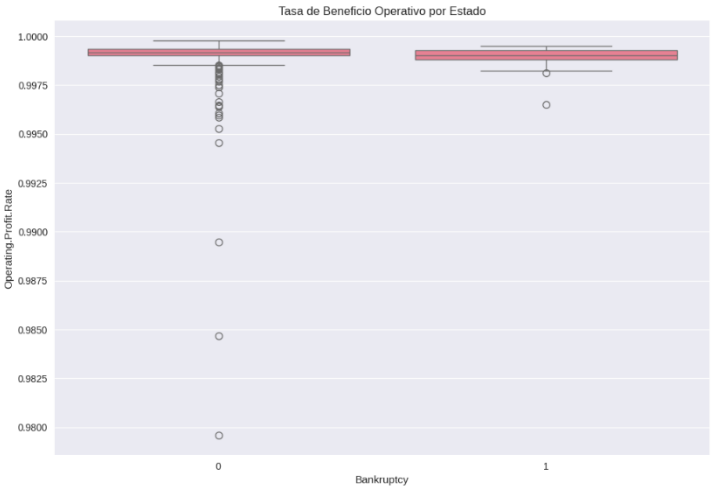


Ilustración 2 Beneficio operativo por estado vs Bankruptcy

Pasivo corriente sobre activos (Current.Liability.to.Assets)

Empresas quebradas presentan mayores proporciones de pasivo corriente. Se observa una clara tendencia hacia valores más altos en el grupo quebrado.

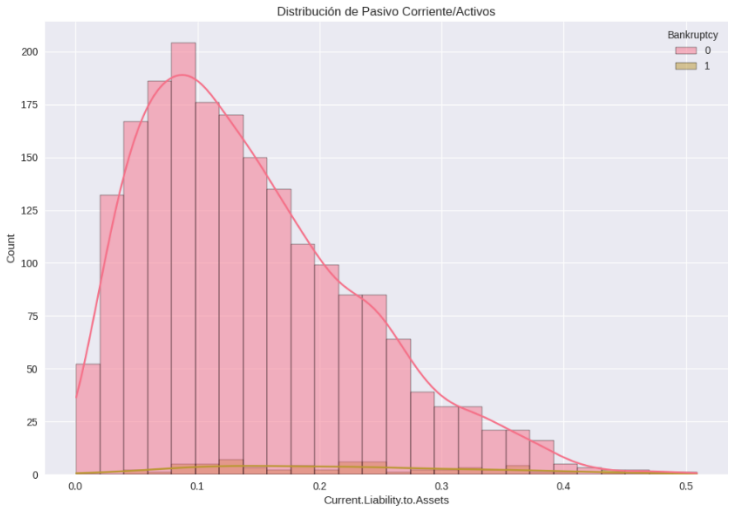


Ilustración 3 Distribucion de pasivo corriente/Activos

Ratio de deuda (Debt.ratio..)

Es una de las variables más discriminantes. Muestra que mpresas quebradas tienen una distribución más extendida hacia altos niveles de endeudamiento.

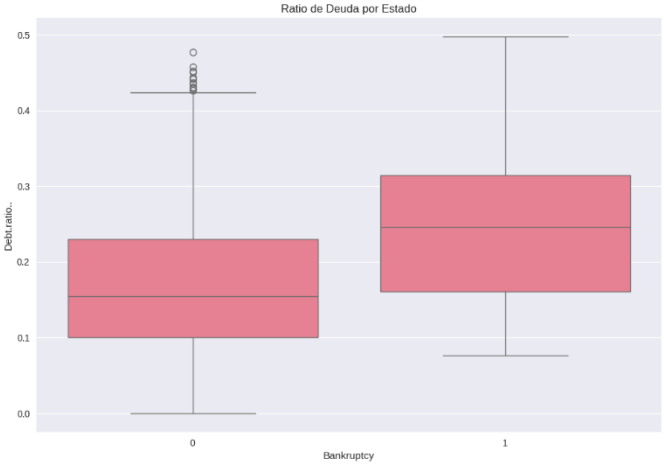


Ilustración 4 Distribución Ratio de deuda por estado

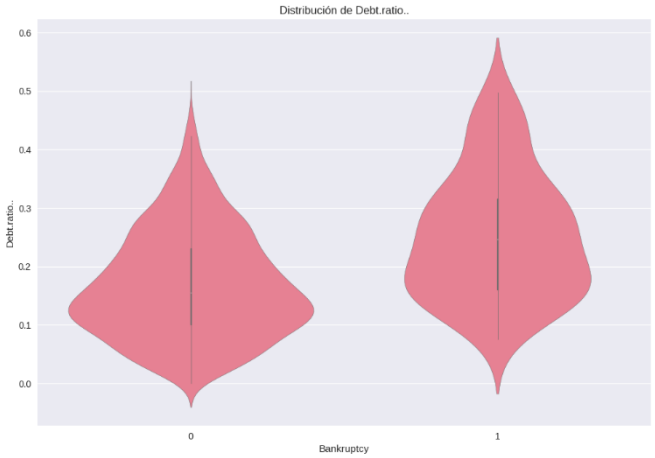


Ilustración 5 Distribución Debt ratio

ROA antes de depreciación e intereses (ROA.B..before...)

Muestra valores visiblemente más bajos en empresas quebradas.

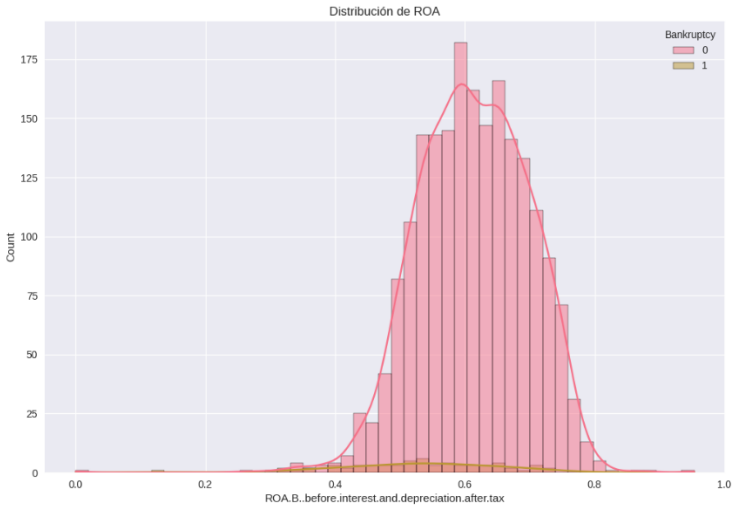


Ilustración 6 Relacion y distribución ROA Bankruptcy

d) Correlaciones y relaciones cruzadas

Se calcularon los coeficientes de correlación de Pearson entre cada variable y la bancarrota y se tomaron el top 10 combinando las 5 más correlacionadas positivamente y las 5 más correlacionadas negativamente. Las variables encontradas tienen sentido en el contexto. Por ejemplo, ROA, da una idea de la eficiencia operativa de la empresa, en cuanto a la capacidad de generar ingresos con sus activos independientemente de la financiación con deuda o capital. El top de correlaciones negativas lo lidera Retained.Earnings.to.Total.Assets lo cual también tiene sentido al ser esta variable usada para

cálculos de predicción de insolvencia, estas variables por tanto pueden ser utilizadas en la solución del problema usándolas en modelos predictivos como se muestra más adelante:

Top correlaciones positivas:

- Debt.ratio.. (0.152)- *Razón de endeudamiento*
- Current.Liability.to.Current.Assets (0.151)- *Pasivo corriente / activo corriente*
- Current.Liability.to.Assets (0.143)- *Pasivo corriente / activos totales*
- Liability.Assets.Flag-*Indicador de pasivo / activos*
- Revenue.per.person-*Ingresos por persona*

Top correlaciones negativas:

- Retained.Earnings.to.Total.Assets (-0.196)- *Utilidades retenidas / activos totales*
- Working.Capital.to.Total.Assets (-0.185)- *Capital de trabajo / activos totales*
- ROA.B..before... (-0.162)- *ROA antes de intereses e impuestos*
- Tax.rate..A.- *Tasa impositiva*
- Quick.Assets.Total.Assets- *Activo líquido / activos totales*.

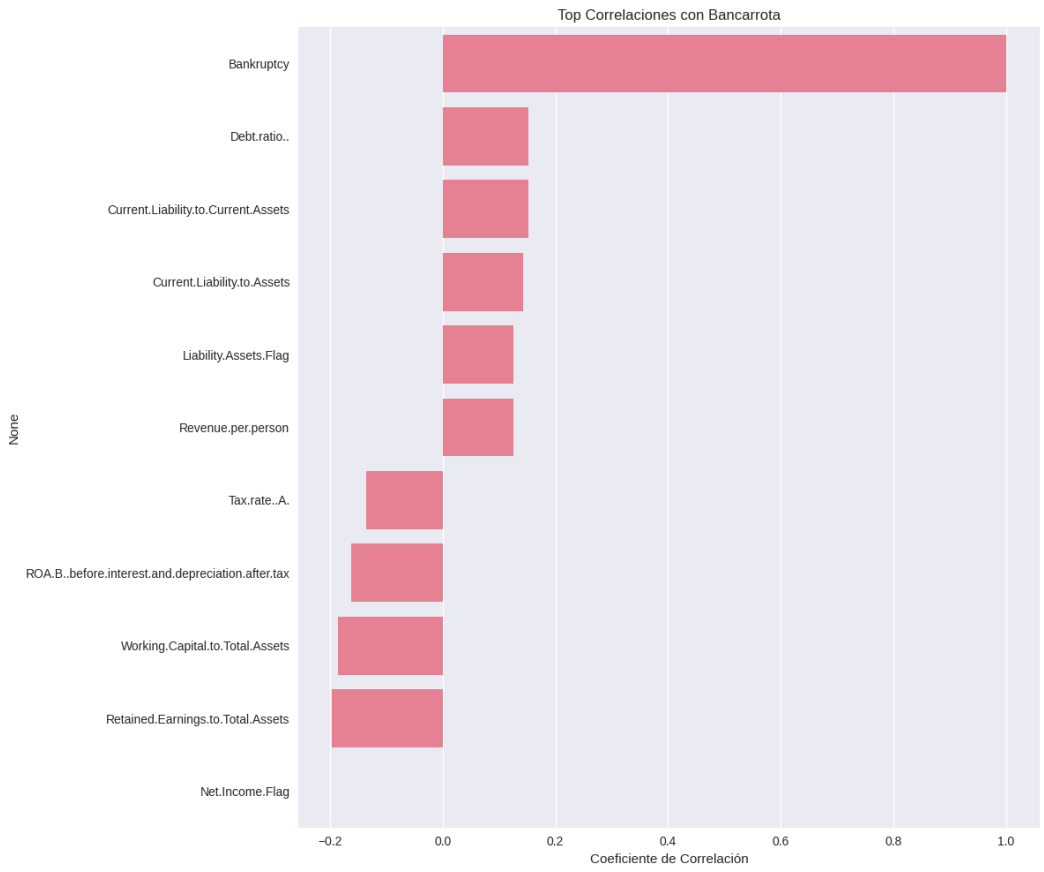


Ilustración 7 Top correlaciones con bancarrota

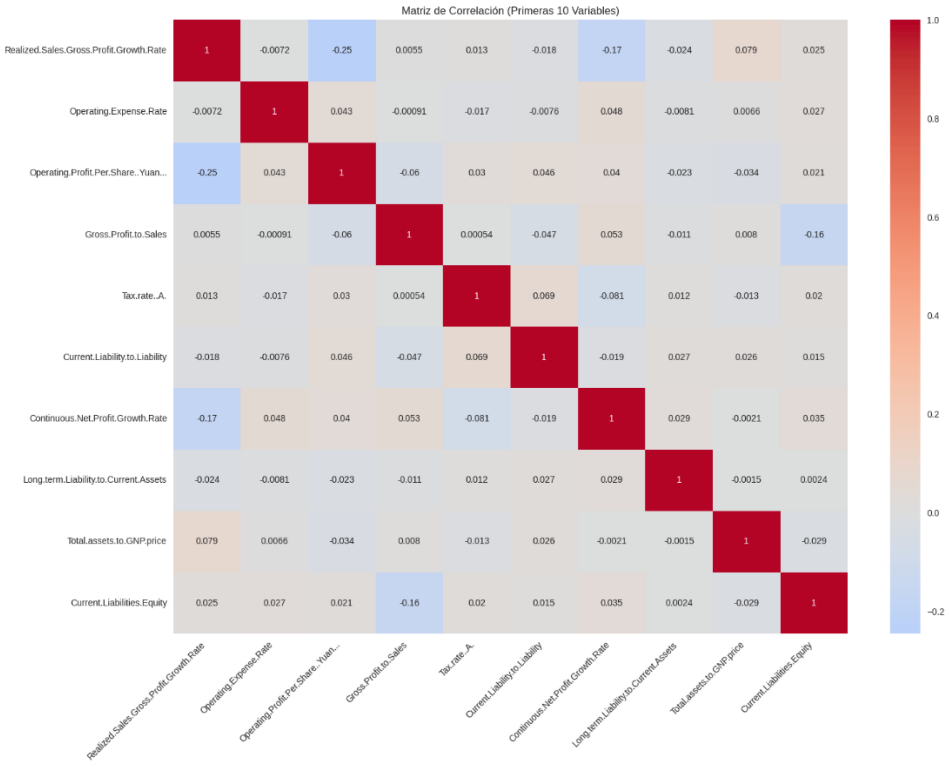


Ilustración 8 Top correlaciones

En esta sección complementamos el análisis descriptivo realizado anteriormente, incorporando la visualización de la matriz de correlaciones entre las variables numéricas. Por temas de espacio, en esta ocasión se presentan únicamente las diez variables más relevantes. Sin embargo, ya contamos con claridad sobre cuáles son las variables que presentan mayores niveles de correlación, y que serán consideradas más adelante para el análisis de multicolinealidad dentro del proceso de modelado.

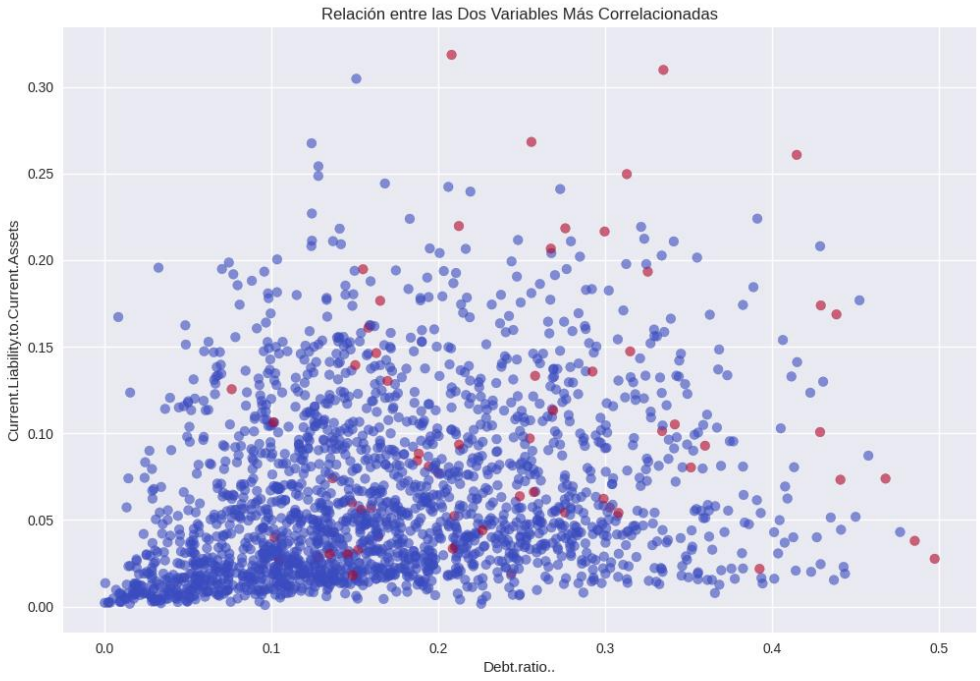


Ilustración 9 Distribución de los casos de bancarrota en función de los indicadores financieros.

Durante el análisis de la relación entre las variables Debt.ratio.. y Current.Liability.to.Current.Assets, se elaboró un gráfico de dispersión para visualizar cómo se distribuyen los casos de bancarrota en función de estos indicadores financieros.

Al observar el gráfico, se identifica cierta tendencia a que los casos de quiebra (representados en rojo) se presenten con mayor frecuencia en empresas que tienen niveles más elevados de deuda y de pasivo corriente respecto a sus activos. Sin embargo, esta concentración no es del todo clara ni exclusiva en la zona superior derecha del gráfico.

De hecho, se evidencia que varios casos de bancarrota también aparecen en empresas con valores moderados e incluso bajos de estos indicadores, lo que indica que aunque las variables analizadas ofrecen información útil, no serían suficientes de manera aislada para construir un modelo predictivo sólido.

A partir de este análisis, concluimos que será necesario complementar la selección de variables incorporando otros factores adicionales, de manera que se pueda mejorar la capacidad de predicción del modelo frente al riesgo de bancarrota.

e) Importancia de variables (Mutual Information)

Se usó mutual information para identificar relaciones no lineales entre las variables y el objetivo.

Variable	Puntuación MI
Tax.rate..A.	0.0216
Working.Capital.to.Total.Assets	0.0199
Retained.Earnings.to.Total.Assets	0.0161
ROA.B..before...	0.0137

Conclusiones analíticas de la exploración

- Las razones financieras asociadas a liquidez y endeudamiento presentan diferencias claras entre empresas quebradas y no quebradas.
- Variables como **Debt Ratio**, **Working Capital**, **Retained Earnings** y **ROA** son especialmente informativas.
- La variable objetivo está severamente desbalanceada, lo que sugiere la necesidad de técnicas de compensación en el modelado.
- El top 10 de variables más correlacionadas con la variable objetivo tiene lógica en el contexto del negocio.

2) Preparación de los datos para poder utilizarlos como entrada para modelos predictivos

Antes de entrenar los modelos de redes neuronales, fue necesario transformar y balancear los datos financieros para garantizar que:

- La distribución de variables no afecte el aprendizaje de los modelos.
- Se reduzca el impacto de outliers, frecuentes en datos financieros.
- Se compense el desbalance de clases, ya que menos del 4% de las empresas del conjunto original se declararon en bancarrota.

Limpieza y separación de variables

- Se eliminaron las columnas no predictivas ID(identificador) y Bankruptcy(Variable objetivo)
- Se separaron los conjuntos:
  - X_train: características predictoras del conjunto de entrenamiento
  - y_train: variable objetivo (Bankruptcy)
  - X_test: características predictoras del conjunto de prueba para Kaggle
- No se presentaban valores nulos en las variables numéricas, por lo cual no fue necesario imputar datos faltantes.

```
Dimensiones del conjunto de entrenamiento: (2050, 65)
Dimensiones del conjunto de prueba: (500, 64)
```

2. Separación de Características y Objetivo

```
# Separar características y objetivo
X_train = train_data.drop(['ID', 'Bankruptcy'], axis=1)
y_train = train_data['Bankruptcy']
X_test = test_data.drop('ID', axis=1)

print("\nDimensiones después de la separación:")
print(f"X_train: {X_train.shape}")
print(f"y_train: {y_train.shape}")
print(f"X_test: {X_test.shape}")
```

```
Dimensiones después de la separación:
X_train: (2050, 63)
y_train: (2050,)
X_test: (500, 63)
```

Ilustración 10 Código implementado en la separación modelo1

a. Normalización de variables

Se identificaron 63 variables numéricas como predictores.
Para reducir la sensibilidad del modelo 1 a valores extremos (**outliers**), se aplicó una normalización robusta utilizando RobustScaler como se ve a continuación.

3. Identificación de Variables Numéricas

In [5]:
```
# Identificar variables numéricas
numeric_features = X_train.select_dtypes(include=[np.number]).columns.tolist()
print(f"\nNúmero de variables numéricas: {len(numeric_features)}")
print("\nVariables numéricas:")
print(numeric_features)
```

```
Número de variables numéricas: 63

Variables numéricas:
['Realized.Sales.Gross.Profit.Growth.Rate', 'Operating.Expense.Rate', 'Operating.Profit.Per.Share..Yuan...', 'Gross.Profit.to.Sales', 'Tax.rate..A.', 'Current.Liability.to.Liability', 'Continuous.Net.Profit.Growth.Rate', 'Long.term.Liability.to.Current.Assets', 'Total.assets.to.GNP.price', 'Current.Liabilities.Equity', 'CFO.to.Assets', 'Current.Liability.to.Current.Assets', 'Persistent.EPS.in.the.Last.Four.Seasons', 'Cash.Total.Assets', 'Inventory.Working.Capital', 'Net.Value.Per.Share..B.', 'Current.Assets.Total.Assets', 'Net.Worth.Turnover.Rate..times.', 'Interest.Coverage.Ratio..Interest.expense.to.EBIT.', 'Equity.to.Liability', 'Operating.profit.Paid.in.capital', 'No.credit.Interval', 'Interest.bearing.debt.interest.rate', 'Net.Income.to.Stockholder.s.Equity', 'Quick.Ratio', 'Cash.Flow.to.Sales', 'Equity.to.Long.term.Liability', 'Working.Capital.Equity', 'After.tax.net.Interest.Rate', 'Current.Liability.to.Assets', 'Net.Value.Per.Share..C.', 'Revenue.per.person', 'Borrowing.dependency', 'Operating.Profit.Rate', 'Long.term.fund.suitability.ratio..A.', 'Pre.tax.net.Interest.Rate', 'After.tax.Net.Profit.Growth.Rate', 'Operating.profit.per.person', 'Realized.Sales.Gross.Margin', 'Cash.Current.Liability', 'Current.Liability.to.Equity', 'Total.expense.Assets', 'Current.Asset.Turnover.Rate', 'Fixed.Assets.Turnover.Frequency', 'ROA.B..before.interest.and.depreciation.after.tax', 'Quick.Asset.Turnover.Rate', 'Debt.ratio..', 'Retained.Earnings.to.Total.Assets', 'Total.debt.Total.net.worth', 'Fixed.Assets.to.Assets', 'Total.Asset.Growth.Rate', 'Inventory.and.accounts.receivable.Net.value', 'Net.Income.Flag', 'Per.Share.Net.profit.before.tax..Yuan...', 'Continuous.interest.rate..after.tax.', 'Liability.Assets.Flag', 'Working.Capital.to.Total.Assets', 'Degree.of.Financial.Leverage..DFL.', 'Operating.Gross.Margin', 'Contingent.liabilities.Net.worth', 'Operating.Profit.Growth.Rate', 'Cash.Flow.to.Liability', 'Quick.Assets.Total.Assets']
```

4. Creación del Pipeline de Preprocesamiento

In [6]:
```
# Crear pipeline de preprocesamiento
preprocessor = ColumnTransformer(
    transformers=[
        ('num', RobustScaler(), numeric_features)
    ],
    remainder='passthrough'
)

# Crear pipeline completo con SMOTE
pipeline = ImbPipeline([
    ('preprocessor', preprocessor),
    ('smote', SMOTE(random_state=42))
])
```

Ilustración 11 Código normalizacion y pipeline para modelo 1.

Se usó Standard scaler para probar en los modelos 2 y 3 que se describirán más adelante

```
from sklearn.preprocessing import StandardScaler

variables_seleccionadas = vars_positivas + vars_negativas

# Separar variables predictoras y objetivo en entrenamiento
X_train = bancarrota[variables_seleccionadas]
y_train = bancarrota['Bankruptcy']

# Estandarizar
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)

# Preparar también el test set para predicciones de Kaggle
X_test = bancarrota_test[variables_seleccionadas]
X_test_scaled = scaler.transform(X_test)
```

Ilustración 12 Código preprocesamiento modelos 2 y 3

Este paso permitió homogenizar las escalas sin verse afectado por valores atípicos extremos, que son comunes en razones financieras.

b. Balanceo de clases

Dado que el conjunto original contenía solo un 3.02% de empresas en bancarrota, se implementó la técnica de SMOTE que genera nuevas observaciones sintéticas de la clase minoritaria para el modelo 1.

4. Creación del Pipeline de Preprocesamiento

In [6]:
```
# Crear pipeline de preprocesamiento
preprocessor = ColumnTransformer(
    transformers=[
        ('num', RobustScaler(), numeric_features)
    ],
    remainder='passthrough'
)

# Crear pipeline completo con SMOTE
pipeline = ImbPipeline([
    ('preprocessor', preprocessor),
    ('smote', SMOTE(random_state=42))
])
```

5. Aplicación del Pipeline

In [7]:
```
# Aplicar pipeline
print("\nAplicando pipeline de preprocesamiento...")
X_train_processed, y_train_processed = pipeline.fit_resample(X_train, y_train)

# Mostrar dimensiones después del procesamiento
print(f"\nDimensiones después del procesamiento:")
print(f"X_train_processed: {X_train_processed.shape}")
print(f"y_train_processed: {y_train_processed.shape}")

# Procesar conjunto de prueba
X_test_processed = pipeline.named_steps['preprocessor'].transform(X_test)
print(f"X_test_processed: {X_test_processed.shape}")
```

```
Aplicando pipeline de preprocesamiento...

Dimensiones después del procesamiento:
X_train_processed: (3976, 63)
y_train_processed: (3976,)
X_test_processed: (500, 63)
```

Ilustración 13 Pipeline de preprocesamiento modelo 1

Resultados balanceo en el Modelo 1:

- Datos originales: 2.050 observaciones
- Datos después de SMOTE: 3.976 observaciones
- Distribución final:
  - 50% clase 0 (no bancarrota)
  - 50% clase 1 (bancarrota)
- Para los modelos 2 y 3 se implementó RandomOverSampler(random_state=42)

Con este ajuste en el balance de clases, ya contamos con una base de datos adecuada para entrenar modelos predictivos de manera más equilibrada. El balance logrado mediante SMOTE en el modelo 1, así como el uso de RandomOverSampler en los modelos 2 y 3, permite reducir el sesgo que podría generar la desproporción original entre clases.
A partir de este punto, se procede con la fase de entrenamiento y evaluación de los modelos con condiciones más equilibradas.

c. Procesamiento del conjunto de prueba

El conjunto X_test fue transformado únicamente mediante el escalamiento (RobustScaler) ya entrenado sobre X_train, sin aplicar balanceo (ya que no es un conjunto supervisado).
Esto asegura que los datos de prueba queden en la misma escala sin introducir información futura (leakage).

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import RandomOverSampler
variables_seleccionadas = vars_positivas + vars_negativas

# Separar variables predictoras y objetivo en entrenamiento
X = bancarrota[variables_seleccionadas]
y = bancarrota['Bankruptcy']
# División en entrenamiento y validación
X_entrenamiento, X_validacion, y_entrenamiento, y_validacion = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Estandarizar
scaler = StandardScaler()
X_entrenamiento_scaled = scaler.fit_transform(X_entrenamiento)
X_validacion_scaled = scaler.transform(X_validacion)

# Preparar también el test set para predicciones de Kaggle
X_test = bancarrota_test[variables_seleccionadas]
X_test_scaled = scaler.transform(X_test)
#  Aplicar oversampling
ros = RandomOverSampler(random_state=42)
X_resampled_k, y_resampled_k = ros.fit_resample(X_entrenamiento_scaled, y_entrenamiento)
```

Ilustración 14 preprocesamiento adicional conjunto de prueba modelos 2 y 3

d. Almacenamiento del resultado

Para el modelo 1 con el fin de facilitar la reutilización en etapas posteriores del pipeline, los datos procesados fueron almacenados como arrays como se ve a continuación:

6. Guardado de Datos Procesados

In [8]:
```
# Guardar datos procesados
print("\nGuardando datos procesados...")
os.makedirs('archivos', exist_ok=True)

np.save('archivos/X_train_processed.npy', X_train_processed)
np.save('archivos/y_train_processed.npy', y_train_processed)
np.save('archivos/X_test_processed.npy', X_test_processed)

print("\nDatos procesados guardados exitosamente:")
print("- X_train_processed.npy")
print("- y_train_processed.npy")
print("- X_test_processed.npy")
```

```
Guardando datos procesados...

Datos procesados guardados exitosamente:
- X_train_processed.npy
- y_train_processed.npy
- X_test_processed.npy
```

7. Verificación de Balance de Clases

In [9]:
```
# Verificar balance de clases
print("\nDistribución de clases después del balanceo:")
print(pd.Series(y_train_processed).value_counts(normalize=True))
```

```
Distribución de clases después del balanceo:
Bankruptcy
0    0.5
1    0.5
Name: proportion, dtype: float64
```

Ilustración 15 Continuación de almacenamiento modelo 1

3) Análisis Preliminar de Selección de Modelos Relevantes

El proyecto tiene como objetivo predecir la probabilidad de que una empresa se declare en bancarrota en el siguiente año, utilizando para ello un conjunto de 63 indicadores financieros extraídos de reportes públicos.
El problema se enmarca dentro de una tarea de clasificación binaria supervisada, pero presenta desafíos importantes: un desbalance marcado entre clases, ya que solo el 3% de las observaciones corresponden a casos positivos de quiebra, y la presencia de múltiples variables numéricas con correlaciones moderadas pero relaciones no estrictamente lineales.

El análisis exploratorio inicial confirmó que los patrones de bancarrota no son evidentes a partir de simples correlaciones lineales, sugiriendo la necesidad de aplicar modelos capaces de capturar interacciones más complejas entre las variables.
En este contexto, las redes neuronales multicapa (MLP y Keras) se presentan como una alternativa adecuada para modelar la probabilidad de bancarrota de forma más efectiva.

En línea con este enfoque, se evaluaron modelos de redes neuronales multicapa (MLP y Keras) ya que estos tienen gran utilidad en un problema como este donde se desean realizar predicciones de posible bancarrota, este tipo de modelos son adecuados para la clasificación usando una función sigmoid en la capa de salida, con una adecuada configuración se pueden usar para capturar las relaciones entre variables financieras que pueden ser no lineales y complejas, como se vió en la baja correlación encontrada en el análisis exploratorio inicial. Al capturar esta relación se puede llegar a una clasificación útil en estos casos.

En nuestro caso de estudio se tienen clases desbalanceadas, por lo cual es útil tener una métrica como el AUC para la evaluación de los modelos.

Se probaron MLP con skealearn usando RandomOverSampler para corregir el desbalanceo de las clases lo cual es muy útil en este caso donde las clases están diferenciadas en frecuencia por ser atípica la clase donde las empresas se van a bancarrota que es una minoría como se espera y se vio también en el análisis exploratorio preliminar.

Regularización y early stopping permiten mejorar la escalabilidad y la generalización.

Lo anterior también representa ventajas frente a modelos tradicionales como los lineales o incluso árboles de decisión como Random Forest, ya que las redes neuronales pueden capturar relaciones no lineales más complejas y adaptarse mejor a patrones ocultos en los datos financieros, especialmente en contextos con clases desbalanceadas.

4) Desarrollo y Calibración de Modelos

Entre los modelos relevantes se seleccionaron los 3 mejores para presentar a continuación, se muestra su configuración, el modelo 1 fue el inicialmente planteado, con el modelo 2 se encontraron mejores resultados para la métrica seleccionada y se ajustaron los parámetros buscando una mejora hasta obtener finalmente el modelo 3

Modelo	Descripción general de la configuración/Arquitectura	Pretratamiento usado adicional al escalado y división en entrenamiento y prueba
Modelo 1: MLP Classifier de sklearn	Capas ocultas: 2 (100, 50) Ambas con Activación: ReLU Optimización: Adam Iteraciones: max_iter=1000 Control de overfitting: early_stopping=True	Se dividieron los datos de train en 80% entrenamiento y 20% validacion para el modelo

Modelo 2: Red Neuronal con TensorFlow/Keras	Dos capas ocultas: una con 64 neuronas y otra con 32, ambas con activación ReLU. Se añadió Dropout del 30% tras cada capa para evitar sobreajuste. La capa de salida tiene una única neurona con activación sigmoide. Optimización con Adam y función de pérdida binary_crossentropy. Se monitorea el AUC como métrica principal. 100 épocas con early stopping (paciencia=5), 20% de los datos de entrenamiento como validación interna.	Selección 10 variables de mayor correlación con el objetivo. Se dividieron los datos de train en 80% entrenamiento y 20% validación para el modelo
Modelo 3: Red Neuronal con TensorFlow/Keras : con regularización L2	Respecto al modelo 2 que fue la base, Se aumentó el número de neuronas en las capas ocultas (128 y 64) y se aplicó regularización L2, se utilizó un Dropout de 0.2. Se implementó early stopping monitorizando la pérdida en el conjunto de validación (val_loss) con una paciencia de 20 épocas y patiente aumentada, lo cual permite detener el entrenamiento si no hay mejora en la pérdida durante 20 iteraciones consecutivas, evitando así el sobreajuste. Además, se activó la opción restore_best_weights=True, lo que asegura que se conserven los pesos del modelo con mejor desempeño en validación.	Selección 10 variables de mayor correlación con el objetivo. Se dividieron los datos de train en 80% entrenamiento y 20% validación para el modelo

La métrica de validación usada fue el AUC.

Códigos para el desarrollo de los modelos

Modelo 1

print("\nDefiniendo modelo MLP...")

model = MLPClassifier(hidden_layer_sizes=(100, 50), activation='relu', solver='adam', max_iter=1000, random_state=42, early_stopping=True)

Ilustración 16 Códigos de entrenamiento y evaluación Modelo 1

Modelo 2

Ilustración 17 Códigos de entrenamiento y evaluación Modelo 2

Modelo 3

Corresponde a variaciones en los parámetros del modelo 2 que se muestran en la imagen

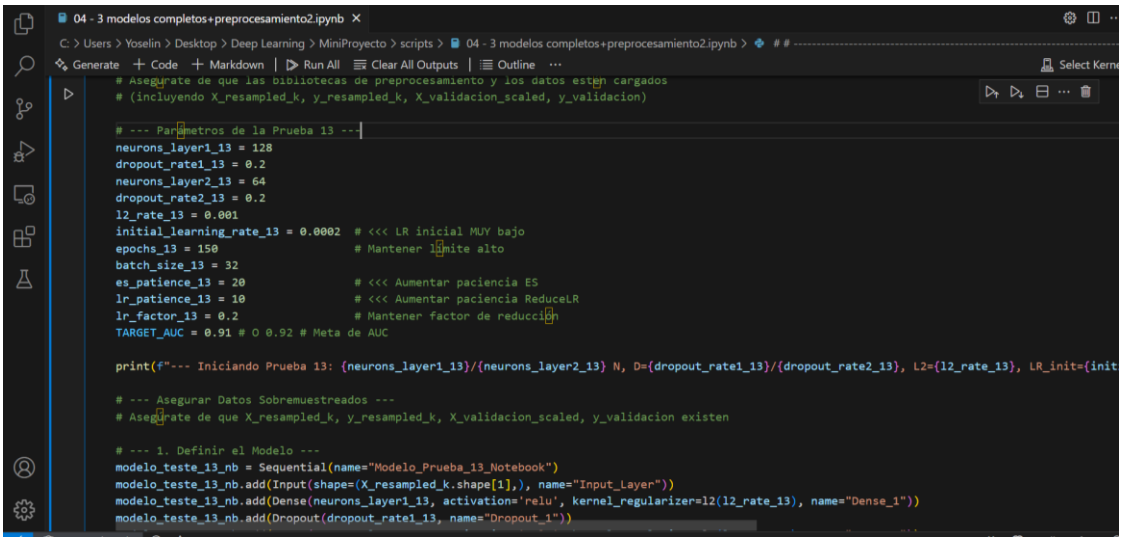


Ilustración 18 Códigos de entrenamiento y evaluación Modelo 3 parte a.

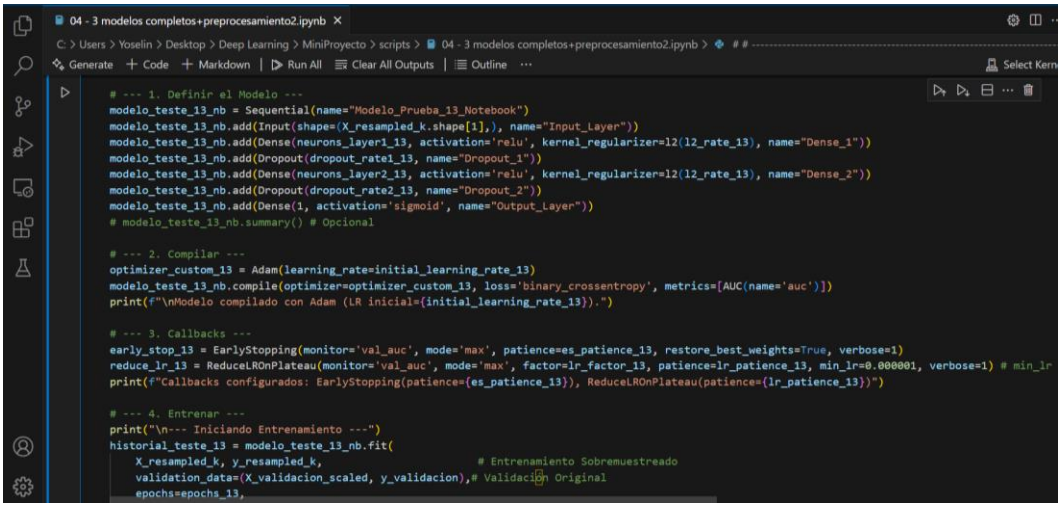
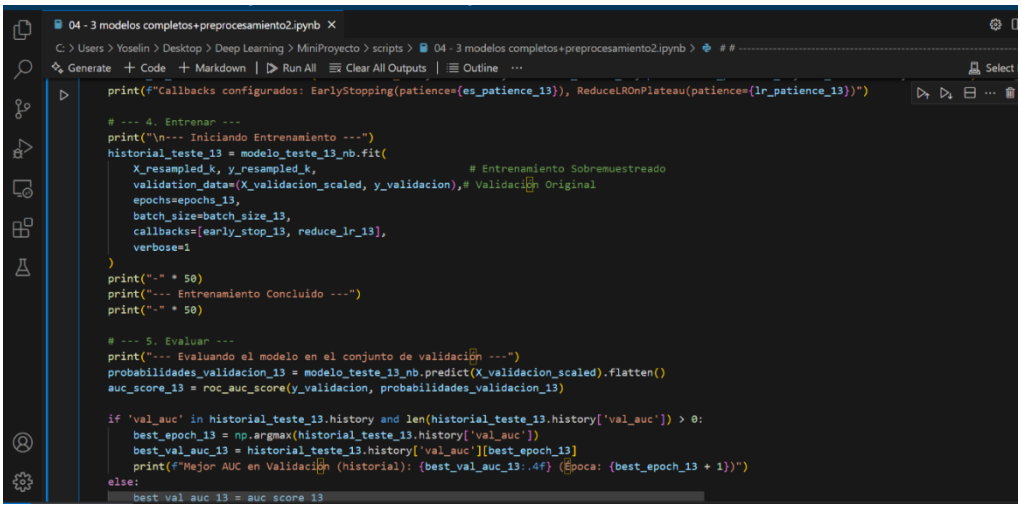


Ilustración 19Códigos de entrenamiento y evaluación Modelo 3 parte b.



Adicionalmente se estableción un objetivo en la métrica de AUC=0.91

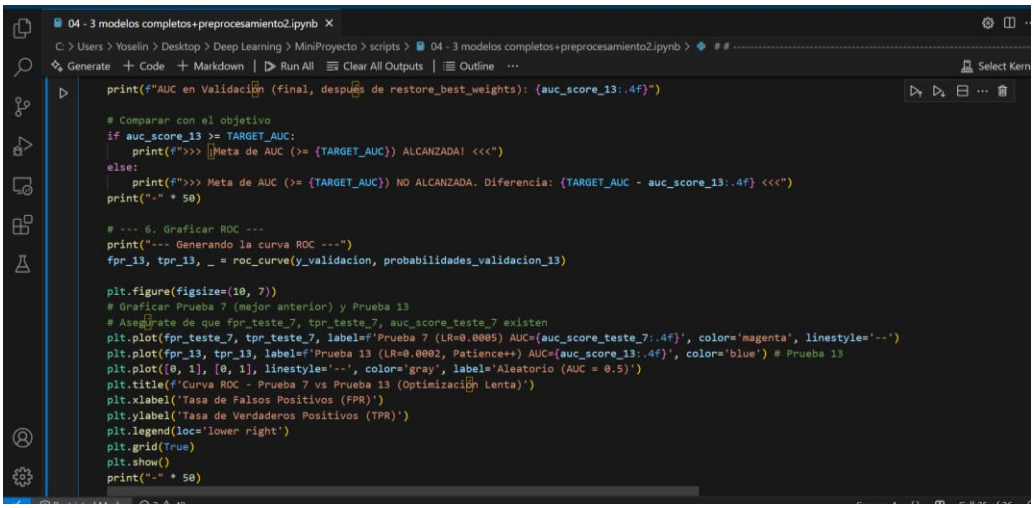


Ilustración 20 Códigos de entrenamiento y evaluación Modelo 3 métrica objetivo

5) Visualización de Resultados

a) Curva ROC para los modelos seleccionados

Curva ROC Modelo 1

La curva ROC muestra un buen ajuste de la muestra un área bajo la curva alta, lo que indica buena discriminación global entre empresas en riesgo y saludables el AUC para este modelo fue de 0.7246.

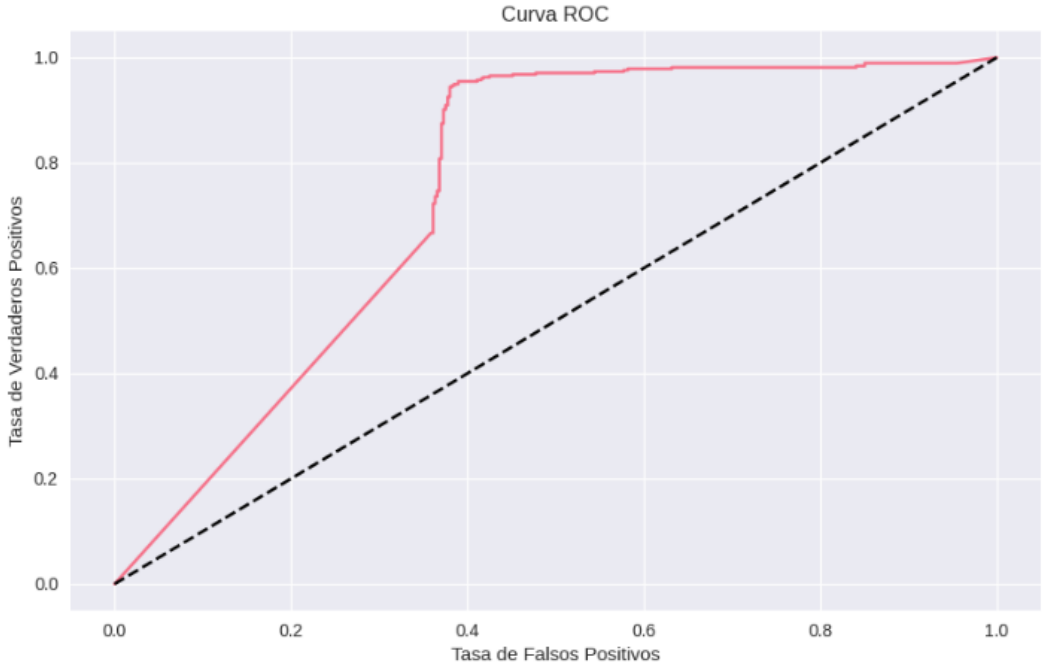


Ilustración 21 Curva ROC y AUC Modelo 1

Curva ROC Modelo 2

La red neuronal también muestra buenos resultados con un AUC de 0.8486 lo que indica que el modelo es adecuado para diferenciar entre clases las empresas saludables y las de Bancarrota. En este aspecto es un mejor modelo que el inicial.

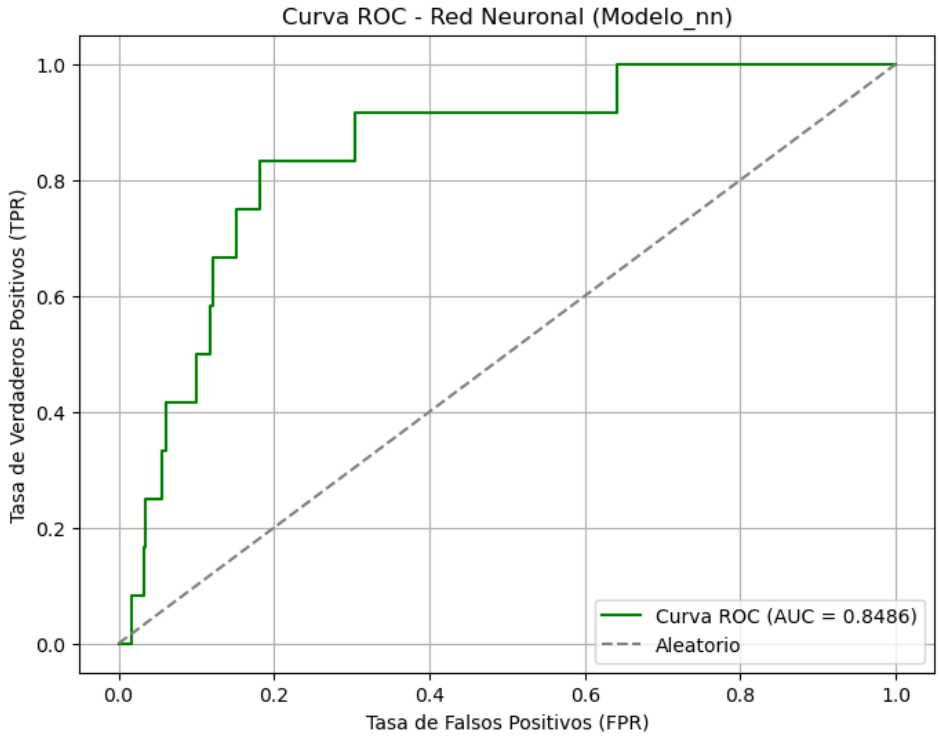


Ilustración 22 Curva ROC y AUC Modelo 2

Curva ROC Modelo 3

La línea azul cubre el área bajo la curva para el modelo 3 que corresponde a la prueba 13, con este se obtuvo un AUC de 0.8867 en los datos de validación, en esta métrica el modelo muestra ser el que mejor diferencia entre las clases de empresas.

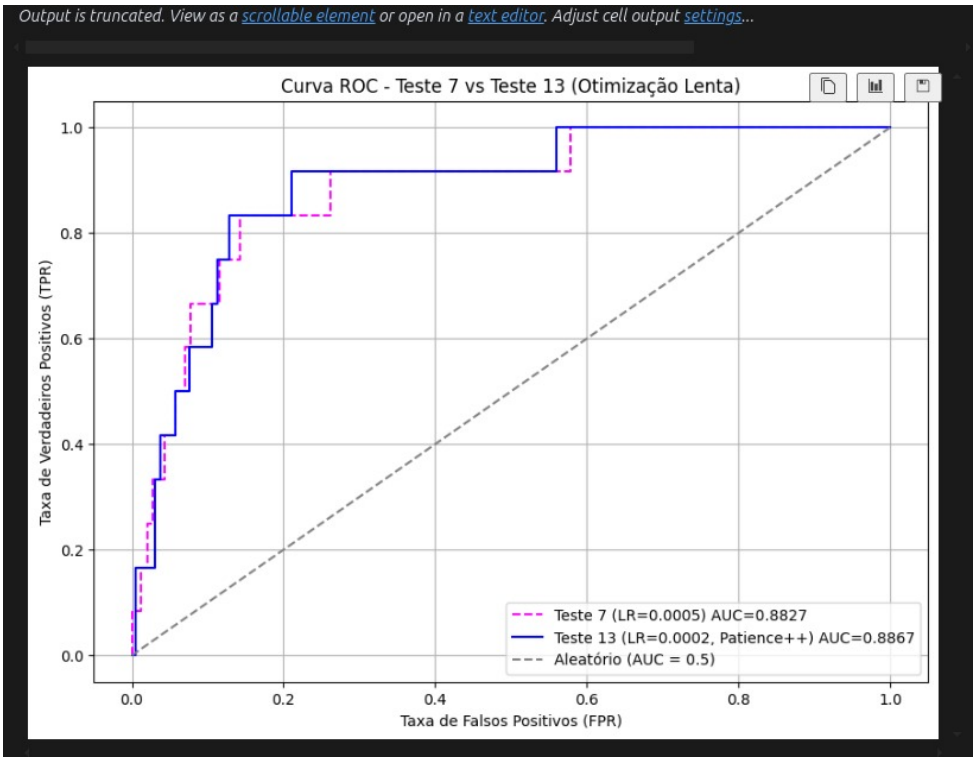


Ilustración 23 Curva ROC y AUC Modelo 3.

b) Comparativo de los AUC para los 3 modelos seleccionados

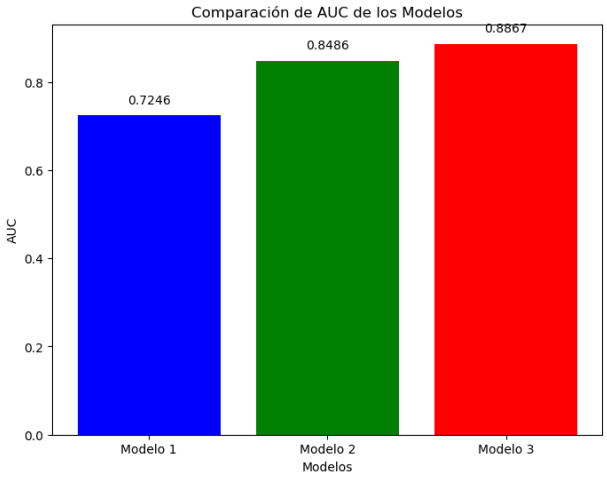


Ilustración 24 Comparación AUC de los modelos.

En la gráfica de comparación de AUC para los datos de validación, obtenidos entre los modelos, podemos ver que los modelos 2 y 3 discriminan mejor entre clases quiebra y no quiebra, sin embargo, un valor de AUC= 0.7246 para el modelo 1 es aceptable también. Por lo tanto se puede decir que los tres modelos seleccionados tienen un buen desempeño para discriminación de clases, y que los procesos de balanceo de clases fueron de utilidad.

c) Resultados de la matriz de confusión para cada modelo

Matriz Modelo 1

En los datos evaluados en la matriz de confusión para el modelo 1 se puede ver que es adecuado para el caso debido a que acierta en la gran mayoría de veces para los datos donde las empresas realmente se declaran en bancarrota 361 frente a los datos donde falla 14 es decir acierta el 96.3% de las veces lo cual es un resultado muy conveniente para el caso, ya que puede ayudar a el objetivo de que bancos y acreedores puedan tomar acciones para proteger sus intereses, y para los demás participantes del mercado obtener primas por riesgo de crédito más justas.

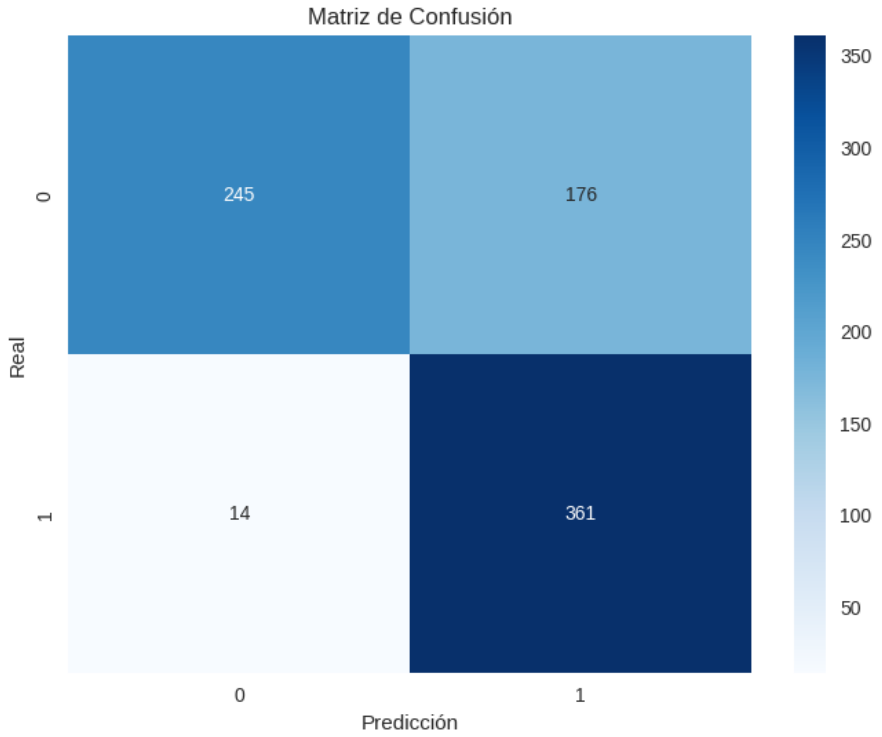


Ilustración 25 Matriz de confusión Modelo 1.

Matriz modelo 2

la efectividad del modelo para la clasificación de mayor interés, que son las empresas que realmente están señaladas como quiebra, vemos que la predicción del modelo no es superior al azar 50%, en el caso de la clase mayoritaria, vemos que el modelo predice correctamente el 89.7% de los casos, este modelo en cuanto a la matriz de confusión no muestra ser más efectivo que el modelo 1 y tampoco ser tan útil para el objetivo del proyecto.

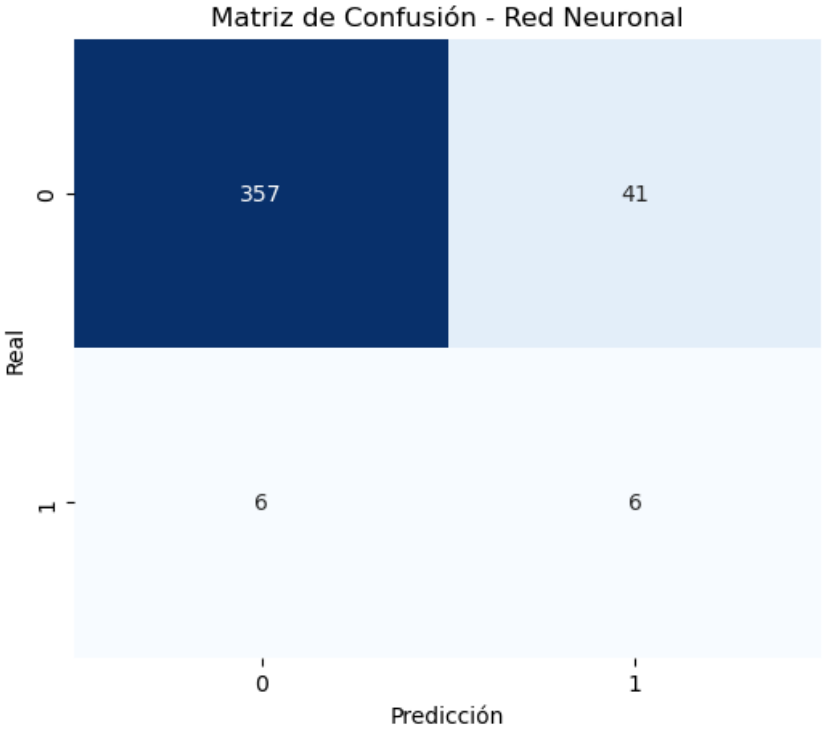


Ilustración 26 Matriz de confusión Modelo 2.

Matriz Modelo 3

Podemos ver que el modelo 3 es muy eficiente en el punto clave del ejercicio clasificando correctamente las empresas de quiebra, de los 12 casos, 11 fueron clasificados correctamente. Para el caso de las observaciones donde realmente no pertenecen a la clase quiebra, el modelo predice correctamente el 68.3%. el porcentaje de casos de quiebra clasificados correctamente en este modelo es de 91,7%.

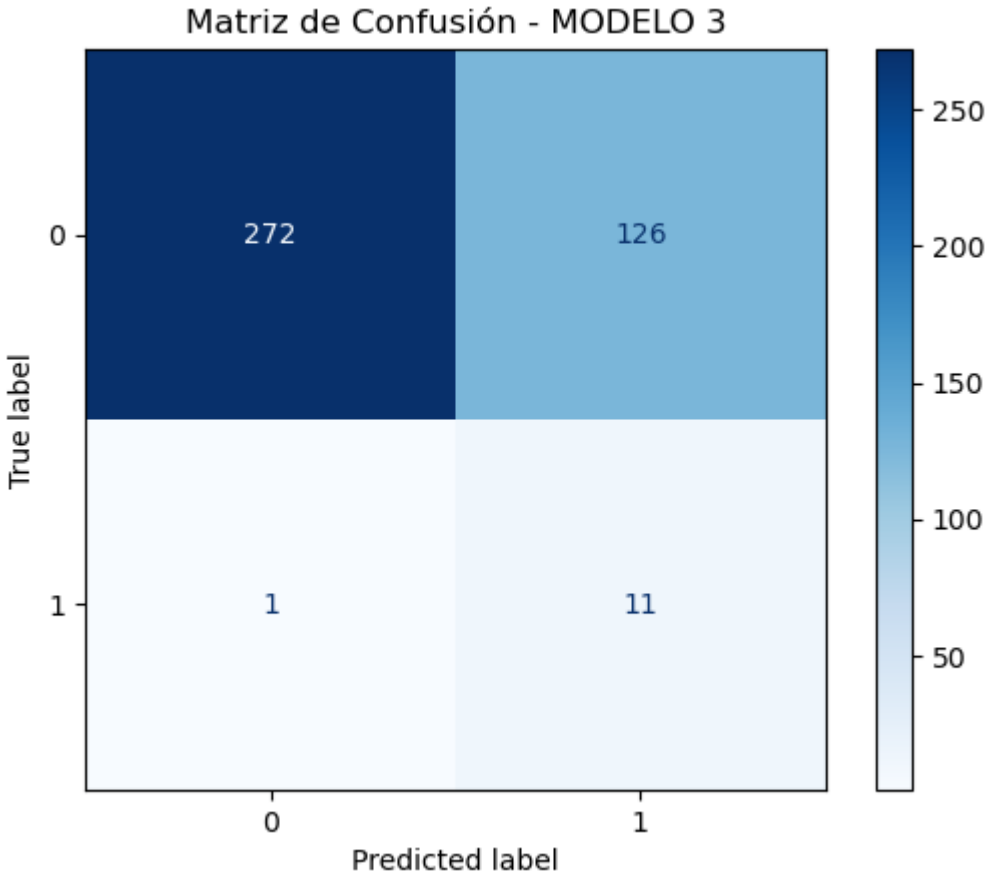


Ilustración 27 Matriz de confusión Modelo 3.

d) Otras métricas de análisis de los modelos

Por último, se compilaron en una tabla los resultados principales para los modelos en los datos de validación en cuanto a la predicción de la clase de mayor interés que era el riesgo de quiebra. Entre los 3 modelos el mas equilibrado es el modelo 1

Tabla 3 Comparativo métricas adicionales de los modelos

Métrica	Modelo 1	Modelo 2	Modelo 3
Precision (clase 1 quiebra)	67.9	12.8	8.03
Recall (clase 1)	96.3	50.0	91.7
F1-score (clase 1)	79.1	20.0	14.8
Accuracy (total)	66.4	88.5	69.02

El modelo 1 es el más preciso y por lo tanto confiable cuando predice una clase de quiebra, lo cual es bueno, los modelos 2 y 3 en este sentido presentan resultados muy bajos. Para tener una mejor visión en este caso podemos centrarnos en el análisis del F1-Score ya que combina precisión y Recall , en este sentido se puede ver que el modelo 1 tienen un valor mucho más alto, vemos que identifica bien las empresas en bancarrota, con una precisión moderada.

Los modelos 2 y 3 predicen bien las quiebras, pero muchas de las predichas no corresponden a empresas en riesgo de quiebra, lo cual puede llevar a confusiones haciéndolos menos confiables para las empresas sanas que pueden ser perjudicadas.

Conclusiones Generales y Recomendaciones

Se logró construir un modelo predictivo lo suficientemente robusto, basado en redes neuronales (MLP Classifier) para anticipar si una empresa se declarará en bancarrota en el año siguiente, utilizando únicamente razones financieras públicas. La predicción temprana de este fenómeno es fundamental para instituciones financieras, aseguradoras, proveedores y entidades reguladoras.

La exploración inicial de los datos da ideas adecuadas para generar modelos mas robustos y confiables, en este caso identificar el desbalanceo de clases fue de gran ayuda para el desarrollo de los modelos incluyendo en el preprocesamiento metodologías de balanceo de clases.

Variables como Debt.ratio.., ROA, y Working.Capital.to.Total.Assets mostraron alta correlación con la variable objetivo lo cual tiene sentido en el contexto del proyecto.

Las curvas ROC y Precision-Recall confirmaron la capacidad del modelo seleccionado para distinguir eficientemente entre empresas en riesgo y saludables.

La matriz de confusión evidenció que los falsos negativos fueron mínimos, aspecto crítico en la gestión de riesgo de bancarrota.

El modelo 1 fue el que presentó un mejor equilibrio entre distinguir entre clases y clasificar correctamente las observaciones, la arquitectura de este fue particularmente diferente a los otros dos modelos que presentaron un desempeño más alto para diferenciar entre clases de acuerdo con los resultados de AUC.

El modelo 1 es mejor en términos de balance entre precisión y recall para la predicción de quiebras por lo tanto podría ser el más adecuado de los 3 en este contexto, ya que minimiza los falsos negativos protegiendo de mejor manera a los acreedores y bancos de empresas que realmente están en quiebra manteniendo además un mejor balance de falsos positivos.

Recomendaciones Finales

- Implementar el modelo como herramienta de monitoreo de riesgo temprano, acompañada de un umbral ajustable basado en sensibilidad deseada.
- Incluir variables macroeconómicas o de mercado para mejorar la capacidad predictiva.
- Considerar modelos ensemble (e.g., XGBoost, LightGBM) para comparación futura.

La implementación de modelos predictivos con redes neuronales en el ámbito financiero no solo es viable, sino altamente efectiva cuando se combinan buenas prácticas de preprocesamiento, balanceo y validación. Este ejercicio demuestra que con un diseño metodológico riguroso es posible anticipar comportamientos críticos en los negocios y tomar decisiones basadas en datos.

Anexos

Enlace al repositorio con análisis asociados:

https://github.com/felipelobatodasilva/MiniProyecto