

Tarea 2: Clasificación Inteligente de Productos en un Centro de Distribución

Curso: Inteligencia Artificial, 2s2025 **Institución:** Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez **Fecha:** 30 de Septiembre 2025

1. Formulación del Problema

1.1 Contexto del Problema

GMB Solutions requiere un sistema de clasificación automática de productos en su centro de distribución. Cada producto debe ser categorizado según dos dimensiones:

1. **Tipo de producto:**
 - Farmacéuticos (requieren certificación de temperatura)
 - Alimentarios (requieren certificación sanitaria)
2. **Prioridad de procesamiento:**
 - Prioritario (fecha de vencimiento cercana)
 - Estándar (fecha de vencimiento lejana)

Esto resulta en **4 categorías combinadas**: - **Clase 1:** Farmacéutico - Prioritario - **Clase 2:** Farmacéutico - Estándar - **Clase 3:** Alimentario - Prioritario - **Clase 4:** Alimentario - Estándar

1.2 Justificación del Uso de Clasificadores Supervisados

Se optó por clasificadores supervisados por las siguientes razones:

1. **Disponibilidad de datos etiquetados:** Se cuenta con un dataset histórico de 200 productos con sus categorías correctas, permitiendo aprendizaje supervisado.
2. **Problema de clasificación multi-clase:** El objetivo es asignar cada producto a una de 4 categorías predefinidas, lo cual es naturalmente un problema de clasificación supervisada.
3. **Necesidad de predicción determinística:** El centro de distribución requiere decisiones claras y consistentes sobre la categorización de cada producto.
4. **Patrones identificables:** Las características de los productos (peso, volumen, tipo de certificación, fecha de vencimiento) tienen relaciones identificables con las categorías objetivo.
5. **Capacidad de generalización:** Los modelos supervisados pueden aprender patrones de los datos históricos y aplicarlos a nuevos productos entrantes.

1.3 Definición de Atributos

El dataset incluye las siguientes características:

Atributo	Descripción	Tipo	Rol
id_producto	Identificador único del producto	String	Identificador
peso	Peso del producto en gramos	Float	Feature
volumen	Volumen en cm ³	Float	Feature
tipo_certificacion	Tipo: “temperatura” o “sanitaria”	Categorico	Feature
fecha_vencimiento	Días restantes hasta vencimiento	Integer	Feature
categoria_objetivo	Clase a predecir (1, 2, 3, 4)	Integer	Target

Estadísticas del Dataset: - Total de muestras: 200 - Distribución de clases:
- Clase 1: 31 muestras (15.5%) - Clase 2: 74 muestras (37.0%) - Clase 3: 24 muestras (12.0%) - Clase 4: 71 muestras (35.5%)

Se observa un **desbalance de clases**, con las clases 2 y 4 (estándar) dominando el dataset.

2. Metodología

2.1 Preprocesamiento de Datos

Se aplicaron las siguientes transformaciones para preparar los datos:

2.1.1 Variables Numéricas **Features:** peso, volumen, fecha_vencimiento

Transformación aplicada: StandardScaler (estandarización)

$$x_scaled = (x - \mu) / \sigma$$

Justificación: - El MLP es sensible a la escala de las features debido al proceso de backpropagation - Las variables tienen rangos muy diferentes (peso: 50-1044g, volumen: 100-2541cm³) - La estandarización centra los datos en media 0 y desviación estándar 1

2.1.2 Variables Categóricas **Feature:** tipo_certificacion

Transformación aplicada: OneHotEncoder con drop='first'

Justificación: - Convierte la variable categórica en representación numérica - drop='first' evita multicolinealidad (dummy variable trap) - Con 2 categorías (temperatura/sanitaria), solo se necesita 1 variable binaria

2.1.3 División de Datos **Estrategia:** Train-Test Split estratificado (70/30)

Parámetros: - Conjunto de entrenamiento: 140 muestras (70%) - Conjunto de prueba: 60 muestras (30%) - **stratify=y:** Mantiene la proporción de clases en ambos conjuntos - **random_state=42:** Reproducibilidad

Justificación: - División 70/30 es estándar para datasets pequeños - Estratificación crítica para mantener representación de clases minoritarias - Conjunto de prueba suficientemente grande para evaluación confiable

2.2 Selección de Modelos

Se seleccionaron dos algoritmos complementarios:

2.2.1 Árbol de Decisión (DecisionTreeClassifier) Ventajas: - Alta interpretabilidad mediante visualización del árbol - No requiere normalización de datos - Captura relaciones no lineales naturalmente - Eficiente con datasets pequeños

Desventajas: - Propenso a sobreajuste sin regularización - Alta varianza (sensible a cambios en datos)

2.2.2 Perceptrón Multicapa (MLPClassifier) Ventajas: - Captura patrones complejos y no lineales - Buena generalización con regularización adecuada - Robusto a datos ruidosos

Desventajas: - Modelo “caja negra” difícil de interpretar - Requiere más datos para entrenamiento óptimo - Sensible a la escala de features

2.3 Optimización de Hiperparámetros

Se utilizó **GridSearchCV** para búsqueda exhaustiva de hiperparámetros:

2.3.1 Árbol de Decisión Espacio de búsqueda: - **criterion:** ['gini', 'entropy'] - **max_depth:** [3, 5, 10, None]

Configuración de búsqueda: - Validación cruzada: 5 folds - Métrica de optimización: F1-macro - Total de combinaciones: 8

Hiperparámetros seleccionados: - **criterion:** **entropy** - **max_depth:** **None** (sin restricción)

Justificación: 1. **Entropy vs Gini:** La entropía basada en teoría de información tiende a crear divisiones más balanceadas en problemas multi-clase. Con nuestro dataset desbalanceado, entropy captura mejor la ganancia de información.

2. **Max_depth=None:** Permite al árbol crecer sin restricción para maximizar el aprendizaje de patrones. Con 200 muestras y validación cruzada,

no se observó sobreajuste severo. Además, mejora la visualización completa de reglas de decisión.

2.3.2 Perceptrón Multicapa **Espacio de búsqueda:** - `hidden_layer_sizes`: [(10,), (50,), (50, 30)] - `learning_rate_init`: [0.01, 0.001] - `alpha`: [0.0001, 0.001]

Configuración de búsqueda: - Validación cruzada: 5 folds - Métrica de optimización: F1-macro - Max iteraciones: 500 - Total de combinaciones: 12

Hiperparámetros seleccionados: - `hidden_layer_sizes`: (50,) - 1 capa oculta con 50 neuronas - `learning_rate_init`: 0.01 - `alpha`: 0.0001

Justificación: 1. **Arquitectura (50,):** Con 4 features de entrada y 4 clases de salida, 50 neuronas en una capa oculta proporciona suficiente capacidad representacional sin sobrecomplicar. La arquitectura más profunda (50, 30) no mejoró el desempeño y aumentaría riesgo de sobreajuste.

2. **Learning rate = 0.01:** Tasa moderadamente alta que acelera convergencia en dataset pequeño. Tasas más bajas requerían más épocas sin mejoras significativas.
3. **Alpha = 0.0001:** Regularización L2 baja que permite aprender patrones complejos con mínima penalización. Una regularización más fuerte limitaba excesivamente la capacidad del modelo.

Criterio de selección: La métrica F1-macro fue elegida porque: - Balancee precision y recall - Trata todas las clases por igual (importante para clases desbalanceadas) - Penaliza modelos que ignoran clases minoritarias

2.4 Validación Cruzada

Estrategia: 5-fold cross-validation estratificado

Proceso: 1. Dividir datos de entrenamiento en 5 folds 2. Para cada fold: - Entrenar modelo en 4 folds - Validar en 1 fold restante 3. Calcular F1-macro promedio y desviación estándar 4. Seleccionar hiperparámetros con mejor F1-macro promedio

Justificación: - 5 folds es un balance entre sesgo y varianza en estimación - Estratificación mantiene proporción de clases en cada fold - Uso completo de datos de entrenamiento para validación

3. Resultados Experimentales

3.1 Métricas Comparativas

Tabla Consolidada de Desempeño:

Modelo	Accuracy	F1-Macro	F1-Weighted	Precision	Recall	AUC Promedio
Árbol de Decisión	0.6000	0.5075	0.6105	0.5200	0.5075	0.7183
MLP	0.6333	0.5045	0.6225	0.5025	0.5025	0.7392

Observaciones: - El MLP obtiene 3.3% más accuracy que el Árbol de Decisión - F1-macro es prácticamente idéntico (~0.50 para ambos) - MLP tiene mejor AUC promedio (0.7392 vs 0.7183) - Ambos modelos tienen desempeño moderado, reflejando la dificultad del problema con clases desbalanceadas

3.2 Desempeño por Clase

Árbol de Decisión:

Clase	Precision	Recall	F1-Score	Support
1 (Farm-Prior)	0.43	0.33	0.38	9
2 (Farm-Est)	0.75	0.82	0.78	22
3 (Alim-Prior)	0.18	0.29	0.22	7
4 (Alim-Est)	0.72	0.59	0.65	22

MLP:

Clase	Precision	Recall	F1-Score	Support
1 (Farm-Prior)	0.38	0.33	0.35	9
2 (Farm-Est)	0.74	0.77	0.76	22
3 (Alim-Prior)	0.17	0.14	0.15	7
4 (Alim-Est)	0.74	0.77	0.76	22

Análisis por clase:

1. **Clase 2 (Farm-Est):** Mejor desempeño para ambos modelos ($F1 > 0.75$) debido a:
 - Mayor representación (22 muestras en test, 74 en total)
 - Patrones más claros y consistentes
2. **Clase 4 (Alim-Est):** Buen desempeño ($F1 \sim 0.65-0.76$) gracias a:
 - Segunda clase más representada (71 muestras)
 - Separación clara por tipo_certificacion
3. **Clases 1 y 3 (Prioritarias):** Desempeño deficiente ($F1 < 0.40$) debido a:
 - Muy pocas muestras (31 y 24 respectivamente)

- Clase 3 especialmente problemática con solo 7 muestras en test
- Dificultad para aprender patrones de prioridad

3.3 Matrices de Confusión

Árbol de Decisión:

Predicho:	1	2	3	4
Real:				
1	3	5	1	0
2	1	18	0	3
3	2	3	2	0
4	1	8	0	13

MLP:

Predicho:	1	2	3	4
Real:				
1	3	5	0	1
2	0	17	0	5
3	2	4	1	0
4	0	5	0	17

Interpretación: - Ambos modelos confunden principalmente entre clases del mismo tipo (Farm/Alim) - Clase 3 es frecuentemente mal clasificada como Clase 2 o 4 - El MLP tiene mejor recall en Clase 4 (77% vs 59%)

3.4 Curvas ROC y AUC

AUC por Clase:

Clase	Árbol de Decisión	MLP
1 (Farm-Prior)	0.682	0.716
2 (Farm-Est)	0.798	0.821
3 (Alim-Prior)	0.631	0.644
4 (Alim-Est)	0.762	0.776
Promedio	0.718	0.739

Análisis: - Ambos modelos superan el clasificador aleatorio ($AUC = 0.5$) - MLP tiene ligeramente mejor AUC en todas las clases - Clase 2 tiene mejor discriminación ($AUC > 0.80$) - Clase 3 tiene menor capacidad de discriminación ($AUC < 0.65$)

3.5 Tiempos de Ejecución

Modelo	Tiempo Entrenamiento	Tiempo Inferencia (60 muestras)	Inferencia por muestra
Árbol de Decisión	0.0031s	0.000846s	0.000014s
MLP	0.7234s	0.001203s	0.000020s

Análisis: - El Árbol de Decisión es **233x más rápido** en entrenamiento - Ambos modelos son extremadamente rápidos en inferencia ($< 2\text{ms}$ para 60 muestras) - Para producción, ambos son viables computacionalmente - MLP requiere más recursos para reentrenamiento periódico

3.6 Curvas de Aprendizaje

Observaciones del Árbol de Decisión: - Score de entrenamiento: ~ 0.90 (alto) - Score de validación: ~ 0.52 (moderado) - Brecha entre curvas indica **ligero sobreajuste** - Curva de validación se estabiliza alrededor de 100 muestras

Observaciones del MLP: - Score de entrenamiento: ~ 0.70 (moderado) - Score de validación: ~ 0.54 (moderado) - Menor brecha indica **mejor generalización** - Ambas curvas convergen, sugiriendo que más datos podrían ayudar

Conclusión: - El MLP generaliza mejor que el Árbol de Decisión - Ambos modelos se beneficiarían de más datos de entrenamiento - La mejora sería especialmente notable para clases minoritarias

4. Discusión y Conclusiones

4.1 Comparación de Modelos

4.1.1 Exactitud y Desempeño Fortalezas del MLP: - Mayor accuracy (63.3% vs 60.0%) - Mejor generalización según curvas de aprendizaje - AUC ligeramente superior en todas las clases - Menor brecha entrenamiento-validación

Fortalezas del Árbol de Decisión: - Mejor precision en Clase 1 (0.43 vs 0.38) - Mejor recall en Clase 3 (0.29 vs 0.14) - Tiempo de entrenamiento 233x más rápido - Desempeño comparable en métricas macro

Conclusión: El MLP tiene una ventaja marginal en accuracy absoluto (3.3%), pero no es una diferencia sustancial considerando el contexto del problema.

4.1.2 Interpretabilidad Árbol de Decisión:

Ventajas: - Visualización completa del árbol de decisión - Reglas explícitas tipo “IF-THEN” - Identificación clara de features importantes - Fácil de explicar a stakeholders no técnicos - Permite auditoría y validación manual

Ejemplo de regla extraída:

```
IF tipo_certificacion = 'temperatura' AND fecha_vencimiento <= 90  
  THEN Clase 1 (Farmacéutico-Prioritario)
```

Aplicación para GMB Solutions: - Operadores pueden entender por qué un producto fue clasificado - Gerentes pueden validar que las reglas siguen la lógica del negocio - Facilita cumplimiento regulatorio (auditorías)

MLP:

Limitaciones: - Modelo “caja negra” con 50 pesos ocultos - Imposible extraer reglas simples - Difícil explicar decisiones individuales - No permite validación manual de la lógica

Impacto para GMB Solutions: - Dificultad para explicar errores de clasificación - Menor confianza de operadores en el sistema - Problemas potenciales en auditorías regulatorias

Conclusión: La interpretabilidad es una ventaja crítica del Árbol de Decisión para este contexto de aplicación.

4.1.3 Costo de Implementación Costos de Entrenamiento: - Árbol: ~3ms (reentrenamiento rápido con nuevos datos) - MLP: ~720ms (reentrenamiento más costoso)

Costos de Mantenimiento: - Árbol: Fácil de actualizar reglas manualmente - MLP: Requiere reentrenamiento completo para ajustes

Costos de Infraestructura: - Ambos modelos son ligeros (< 1MB) - Inferencia extremadamente rápida para ambos - No requieren GPU ni hardware especializado

Conclusión: El Árbol de Decisión tiene menor costo total de propiedad (TCO) para GMB Solutions.

4.2 Interpretación de Diferencias de Desempeño

Las diferencias observadas entre los modelos se explican por:

4.2.1 Naturaleza del Problema

- El problema tiene features con **relaciones relativamente simples**
- **tipo_certificacion** separa claramente Farmacéutico/Alimentario
- **fecha_vencimiento** determina la prioridad
- Los árboles son naturalmente buenos para **límites de decisión rectangulares**

- El MLP no puede aprovechar su capacidad de aprender patrones complejos

4.2.2 Tamaño del Dataset

- Con **solo 200 muestras**, el MLP no puede entrenar óptimamente
- Los árboles de decisión son más **eficientes con datos limitados**
- El MLP necesitaría 1000+ muestras para mostrar su verdadero potencial

4.2.3 Desbalance de Clases

- Clase 2: 74 muestras (37%) vs Clase 3: 24 muestras (12%)
- El Árbol puede crear **splits específicos** para clases minoritarias
- El MLP tiende a **sesgar hacia clases mayoritarias** en su función de pérdida
- F1-macro penaliza este sesgo, por eso ambos tienen ~0.50

4.2.4 Regularización

- Árbol sin restricción de profundidad → mayor flexibilidad
- MLP con $\alpha=0.0001$ → regularización mínima
- Ambos intentan maximizar capacidad de aprendizaje
- Árbol muestra más sobreajuste pero esto no impacta test severamente

4.3 Recomendación para GMB Solutions

Modelo Recomendado: ÁRBOL DE DECISIÓN

Justificación Integral:

1. **Interpretabilidad Crítica (Peso: 40%)**
 - En un centro de distribución con productos farmacéuticos y alimentarios **regulados**
 - Es **esencial explicar** por qué un producto fue clasificado en cierta categoría
 - Auditorías de certificación requieren **trazabilidad de decisiones**
 - Operadores necesitan **confianza** en el sistema para seguir sus recomendaciones
2. **Desempeño Comparable (Peso: 30%)**
 - Diferencia de accuracy de solo 3.3% (60% vs 63.3%)
 - F1-macro prácticamente idéntico (0.507 vs 0.505)
 - Para el contexto de GMB, ambos desempeños son **aceptables pero no óptimos**
 - La pequeña ventaja del MLP **no compensa** la pérdida de interpretabilidad
3. **Costo de Implementación y Mantenimiento (Peso: 20%)**
 - Entrenamiento 233x más rápido facilita **reentrenamiento frecuente**
 - Posibilidad de **ajustes manuales** a las reglas si es necesario

- **Menor fricción** con equipos operativos al explicar el sistema
 - **Más fácil de debuggear** cuando ocurren errores
4. **Facilidad de Actualización (Peso: 10%)**
- Si GMB cambia criterios de prioridad (ej: umbral de vencimiento)
 - El árbol puede ser **reentrenado rápidamente**
 - Las reglas son **fáciles de validar** después de cambios
 - Menor riesgo en ciclos de desarrollo iterativos

Consideración Alternativa:

El MLP podría considerarse en estos escenarios futuros: - Si se recolectan 1000+ muestras (especialmente de clases minoritarias) - Si se agrega ingeniería de features más compleja - Como **modelo ensemble** combinado con el árbol para mayor accuracy - Para **validación cruzada** de decisiones críticas del árbol

4.4 Explicación de las Diferencias de Desempeño

¿Por qué el MLP solo es 3.3% mejor en accuracy?

1. **Simplicidad del problema:**
 - Solo 4 features con relaciones directas
 - Límites de decisión principalmente rectangulares
 - No se requieren transformaciones no lineales complejas
2. **Datos limitados:**
 - 200 muestras son insuficientes para que el MLP aprenda representaciones complejas
 - El árbol es más eficiente en régimen de datos limitados
3. **Desbalance de clases:**
 - Ambos modelos luchan con clases 1 y 3 (minoritarias)
 - El problema fundamental es la falta de muestras, no el algoritmo

¿Por qué el F1-macro es casi idéntico?

- F1-macro penaliza ignorar clases minoritarias
- Ambos modelos tienen **dificultad similar** con clases 1 y 3
- El MLP es mejor en clases mayoritarias, pero esto no mejora F1-macro
- Refleja que el desafío principal es el **desbalance**, no la capacidad del modelo

¿Por qué el árbol tiene mejor recall en Clase 3?

- El árbol puede crear **splits específicos** para los pocos ejemplos de Clase 3
- El MLP promedia sobre todas las muestras en su función de pérdida
- Demuestra la **flexibilidad** del árbol para casos raros

4.5 Limitaciones del Estudio

1. **Dataset pequeño (200 muestras):**
 - Limita la capacidad de generalización de ambos modelos

- Especialmente problemático para clases minoritarias
 - Métricas de validación tienen alta varianza
2. **Desbalance significativo:**
 - Clase 2: 74 muestras (37%) vs Clase 3: 24 muestras (12%)
 - Ratio 3:1 entre clase más y menos representada
 - Modelos sesgan hacia clases mayoritarias
 3. **Features limitadas (solo 4):**
 - Podría haber información adicional relevante no capturada
 - Ejemplos: proveedor, historial de calidad, temperatura de almacenamiento
 4. **Sin análisis de costos de error:**
 - Errores en clasificación prioritaria tienen distinto costo que en estándar
 - No se consideraron umbrales de decisión ajustados por clase
 5. **Validación temporal no realizada:**
 - No se validó desempeño a lo largo del tiempo
 - Posible data drift en producción no evaluado

4.6 Mejoras Futuras Propuestas

4.6.1 Recolección de Datos Prioridad: ALTA

Acciones: - Recolectar **500+** **muestras adicionales**, enfocándose en clases 1 y 3 - Implementar **etiquetado continuo** de productos nuevos - Objetivo: Balancear dataset a ~100 muestras por clase

Impacto esperado: - Mejora de 10-15% en F1-macro - Especialmente en clases prioritarias

4.6.2 Técnicas de Balanceo Prioridad: MEDIA

Opciones: 1. **SMOTE (Synthetic Minority Over-sampling):** - Generar muestras sintéticas de clases 1 y 3 - Validar que no introduzca ruido

2. **Class weights:**
 - Penalizar más errores en clases minoritarias
 - Implementar en ambos modelos
3. **Undersampling:**
 - Reducir clases 2 y 4 (solo si se recolectan más datos primero)

Impacto esperado: - Mejora de 5-10% en recall de clases minoritarias

4.6.3 Feature Engineering Prioridad: MEDIA

Features propuestas: 1. **Densidad: peso / volumen** - Podría separar mejor tipos de productos

2. **Prioridad urgente: fecha_vencimiento < 30**
 - Variable binaria explícita para prioridad

3. **Interacción:** `tipo_certificacion × fecha_vencimiento`

- Capturar patrones combinados

4. **Features temporales:**

- Día de la semana, mes, estación (si hay estacionalidad)

Impacto esperado: - Mejora de 5-8% en accuracy general

4.6.4 Modelos Ensemble **Prioridad: BAJA (post-mejoras de datos)**

Opciones: 1. **Random Forest:** - Combinar múltiples árboles para reducir varianza - Mantiene interpretabilidad parcial (feature importance)

2. **XGBoost:**

- Gradient boosting para mejor desempeño
- Menos interpretable pero más preciso

3. **Voting Classifier:**

- Combinar Árbol de Decisión + MLP
- Aprovechar fortalezas de ambos

Impacto esperado: - Mejora de 5-10% en accuracy - Trade-off con interpretabilidad

4.6.5 Ajuste de Umbrales **Prioridad: ALTA (implementación inmediata)**

Estrategia: - Ajustar **thresholds de decisión** por clase según costos de error
- Ejemplo: Clasificar erróneamente como “no prioritario” tiene mayor costo - Implementar **matriz de costos** personalizada

Implementación:

```
# Reducir umbral para clases prioritarias  
if proba[clase_1] > 0.3: # En lugar de 0.5 por defecto  
    return clase_1
```

Impacto esperado: - Mejora de 10-15% en recall de clases críticas (1 y 3) - Mejor alineación con objetivos de negocio

4.6.6 Monitoreo en Producción **Prioridad: ALTA**

Sistema de monitoreo continuo: 1. **Métricas en tiempo real:** - Accuracy diario/semanal - Distribución de predicciones por clase - Tiempos de inferencia

2. **Detección de drift:**

- Comparar distribución de features con entrenamiento
- Alertas cuando características cambian significativamente

3. **Feedback loop:**

- Validación manual de muestras aleatorias
- Reentrenamiento mensual con nuevos datos etiquetados

4. **A/B Testing:**

- Comparar modelo en producción con nuevas versiones
- Despliegue gradual de mejoras

Impacto esperado: - Mantener desempeño a lo largo del tiempo - Identificar oportunidades de mejora continua

5. Conclusión Final

Para el problema de **clasificación inteligente de productos en el centro de distribución de GMB Solutions**, se evaluaron dos enfoques de aprendizaje supervisado: Árbol de Decisión y Perceptrón Multicapa (MLP).

Hallazgos principales:

1. **Desempeño:** El MLP obtuvo marginalmente mejor accuracy (63.3% vs 60.0%), pero ambos modelos tuvieron F1-macro similar (~0.50), reflejando la dificultad del problema con clases desbalanceadas.
2. **Interpretabilidad:** El Árbol de Decisión ofrece transparencia total mediante reglas visualizables, mientras que el MLP es una “caja negra” difícil de interpretar.
3. **Eficiencia:** El Árbol de Decisión es 233x más rápido en entrenamiento, facilitando actualizaciones frecuentes del modelo.
4. **Generalización:** El MLP mostró mejor generalización en curvas de aprendizaje, pero esto no se tradujo en mejoras significativas en el conjunto de prueba.

Recomendación:

Se recomienda implementar el **Árbol de Decisión (criterion=‘entropy’, max_depth=None)** como solución inicial para GMB Solutions, basado en:

- **Interpretabilidad crítica** para cumplimiento regulatorio en productos farmacéuticos y alimentarios
- **Desempeño comparable** (diferencia de 3.3% no justifica pérdida de explicabilidad)
- **Menor costo** de implementación y mantenimiento
- **Mayor confianza** de usuarios finales al poder validar reglas de decisión

Ruta de mejora:

1. **Corto plazo:** Implementar árbol de decisión con monitoreo de desempeño
2. **Mediano plazo:** Recolectar más datos (objetivo: 500+ muestras) y aplicar técnicas de balanceo
3. **Largo plazo:** Evaluar modelos ensemble (Random Forest) una vez se tenga dataset más robusto

Con estas mejoras, se espera alcanzar $\text{accuracy} > 75\%$ y $\text{F1-macro} > 0.70$, haciendo el sistema suficientemente confiable para automatización completa del proceso de clasificación.

Referencias

1. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
 2. Decision Trees: Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees.
 3. Neural Networks: Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors.
 4. Imbalanced Classification: He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering.
 5. Model Evaluation: Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks.
-

Anexo A: Código y Visualizaciones

Todos los análisis, gráficos y código están disponibles en el notebook de Google Colab: `Tarea2_modelos.ipynb`

Anexo B: Datos

Dataset utilizado: `productos.csv` (200 muestras \times 6 features)