Evaluación del Impacto de la Discretización Local en Clasificadores basados en redes bayesianas

Trabajo de Investigación

Diciembre 2025

Resumen

Este trabajo presenta una evaluación experimental exhaustiva del impacto de la discretización local iterativa en tres Clasificadores basados en redes bayesianas: TAN (Tree Augmented Naive Bayes), KDB (K-Dependence Bayesian Classifier) y AODE (Averaged One-Dependence Estimators). Se comparan las versiones con discretización tradicional (a priori) frente a las versiones con discretización local (TANLd, KDBLd, AODELd) utilizando 27 datasets de referencia. Los resultados muestran que la discretización local mejora en el 17.4 % de los casos evaluados, con mejoras de hasta +1.37 % en datasets específicos como mfeat-fourier. Se identifican las condiciones y datasets donde la discretización local es más beneficiosa, así como una comparativa detallada entre MDLP y métodos de discretización no supervisada.

Índice

1. Introducción

La discretización de variables continuas es un paso fundamental en el preprocesamiento de datos para Clasificadores basados en redes bayesianas, los cuales asumen que las variables predictoras son categóricas. El enfoque tradicional, conocido como discretización a priori o global, aplica el mismo esquema de discretización a todas las instancias del conjunto de datos antes del entrenamiento del clasificador.

La discretización local propone un enfoque alternativo donde el proceso de discretización considera las relaciones estructurales del clasificador, específicamente las dependencias entre variables padres e hijas en la red bayesiana. Este enfoque iterativo busca refinar los intervalos de discretización basándose en la estructura aprendida del clasificador.

1.1. Objetivos del Estudio

Los objetivos principales de este estudio son:

1. Evaluar el impacto de la discretización local en la accuracy de Clasificadores basados en redes bayesianas (TAN, KDB, AODE).
2. Identificar los escenarios y datasets donde la discretización local proporciona beneficios.
3. Comparar el rendimiento de diferentes algoritmos de discretización: MDLP (supervisada), igual amplitud e igual frecuencia (no supervisadas), y PKI.
4. Analizar el efecto del número máximo de puntos de corte (3, 4, 5, ilimitado) en el rendimiento.
5. Estudiar el coste computacional adicional de la discretización local.

1.2. Clasificadores Evaluados

Se evaluaron seis configuraciones de clasificadores:

- **TAN** (Tree Augmented Naive Bayes): Extiende Naive Bayes permitiendo dependencias entre atributos mediante un árbol de máxima verosimilitud.
- **KDB** (K-Dependence Bayesian Classifier, k=2): Permite hasta k dependencias padre por cada atributo.
- **AODE** (Averaged One-Dependence Estimators): Promedia múltiples modelos one-dependence.
- **TANLd, KDBLd, AODELd**: Versiones de los anteriores con discretización local iterativa.

2. Metodología Experimental

2.1. Algoritmos de Discretización

Se evaluaron los siguientes métodos de discretización:

2.1.1. Discretización Supervisada

MDLP (Minimum Description Length Principle): Método de Fayyad-Irani que utiliza el criterio de ganancia de información con penalización MDL para determinar puntos de corte óptimos. Se evaluaron configuraciones con límite de 3, 4, 5 puntos de corte máximos, así como sin límite.

2.1.2. Discretización No Supervisada

- **Igual amplitud (bin-u)**: Divide el rango de valores en k intervalos de igual tamaño.
- **Igual frecuencia (bin-q)**: Divide los datos en k intervalos con aproximadamente el mismo número de instancias.
- **PKI**: Variante de igual frecuencia donde $k = \sqrt{n}$ (pkisqrt) o $k = \ln(n)$ (pkilog), siendo n el número de características.

2.2. Proceso de Discretización Local

La discretización local implementa un proceso iterativo que:

1. Construye el clasificador inicial con discretización tradicional.
2. Re-discretiza cada variable considerando sus padres y la clase.
3. Reconstruye el modelo con la nueva discretización.
4. Repite hasta convergencia (máximo 10 o 100 iteraciones según configuración).

2.3. Configuración Experimental

- **Validación**: 5-fold cross-validation estratificada
- **Repeticiones**: 3 semillas aleatorias por experimento
- **Métrica**: Accuracy (exactitud)
- **Datasets**: 27 conjuntos de datos de referencia
- **Configuraciones de puntos de corte**: 3, 4, 5, ilimitado

2.4. Datasets Utilizados

Se utilizaron 27 datasets del repositorio UCI con características diversas, desde pequeños conjuntos como Iris (150 muestras) hasta grandes bases como Adult (45,222 muestras). Los datasets cubren diferentes dominios: médico, financiero, reconocimiento de patrones, entre otros.

3. Resultados con 10 Iteraciones

En esta sección se presentan los resultados de los experimentos con un máximo de 10 iteraciones para la discretización local.

3.1. Análisis Global de Resultados

De las 324 comparaciones realizadas (27 datasets $\times$ 3 clasificadores $\times$ 4 configuraciones de cortes):

- **Casos positivos** (discretización local mejora): 56 casos (17.3 %)
- **Casos negativos** (discretización tradicional mejor): 268 casos (82.7 %)

3.2. Análisis de Casos Positivos

La Figura 1 muestra el análisis detallado de los casos donde la discretización local mejora el rendimiento.

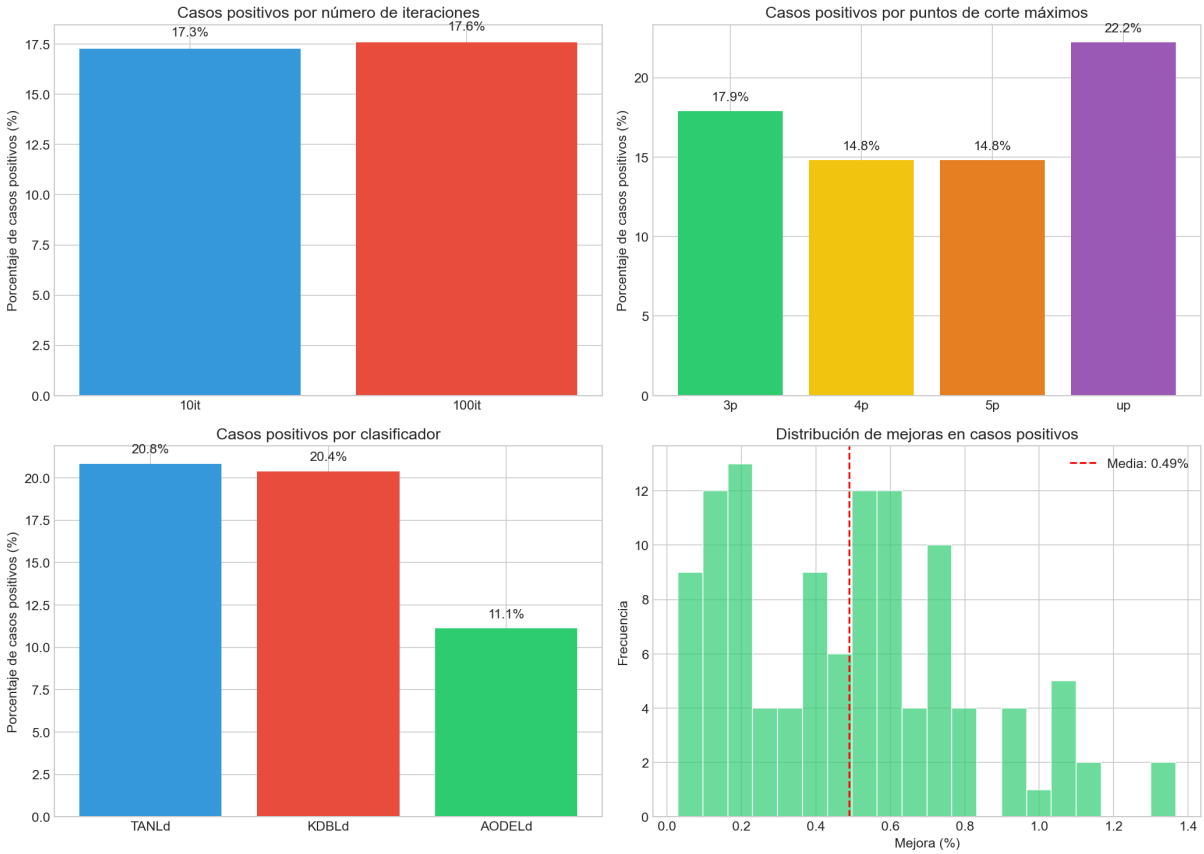

Figura 1: Análisis de casos positivos de discretización local por configuración

3.2.1. Por Configuración de Puntos de Corte

- **3 puntos**: 17.3 % de casos positivos
- **4 puntos**: 14.8 % de casos positivos
- **5 puntos**: 14.8 % de casos positivos
- **Ilimitado**: **22.2 %** de casos positivos (mejor configuración)

La configuración con puntos de corte ilimitados presenta el mayor porcentaje de casos donde la discretización local mejora.

3.2.2. Por Clasificador Base

- **TANLd**: 20.4 % de casos positivos
- **KDBLd**: 20.4 % de casos positivos
- **AODELd**: 11.1 % de casos positivos

Los clasificadores TAN y KDB se benefician más de la discretización local que AODE.

3.3. Datasets más Beneficiados

La Tabla 1 presenta los datasets donde la discretización local produce las mayores mejoras.

Cuadro 1: Top 10 datasets más beneficiados por la discretización local

Dataset	Mejora máx.	Modelo	Configuración
mfeat-fourier	+1.37 %	KDBLd	4p
mfeat-morphological	+1.13 %	KDBLd	4p
diabetes	+0.95 %	KDBLd	up
adult	+0.90 %	KDBLd	up
wine	+0.75 %	TANLd	3p
heart-statlog	+0.74 %	TANLd	up
waveform-5000	+0.71 %	TANLd	up
breast-w	+0.64 %	KDBLd	4p
page-blocks	+0.64 %	AODELd	3p
glass	+0.62 %	KDBLd	4p

La Figura 2 muestra gráficamente los datasets más beneficiados.

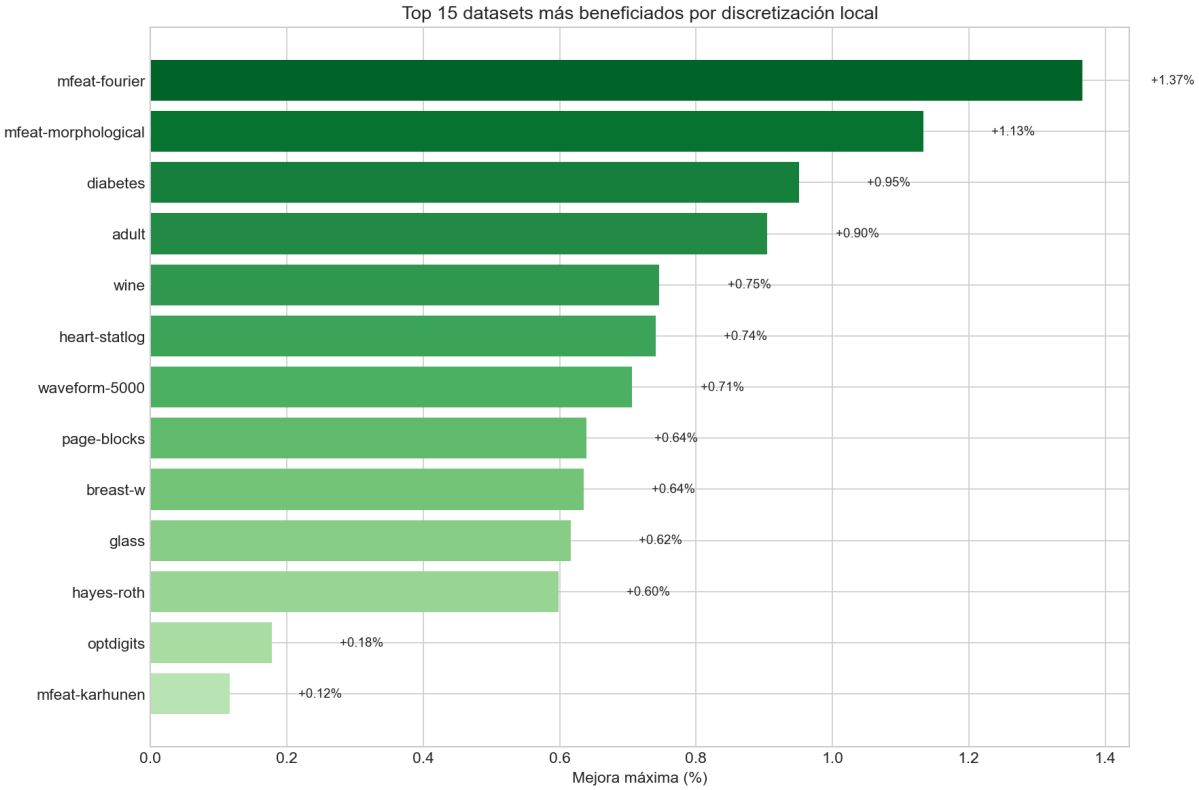

Figura 2: Top 15 datasets más beneficiados por discretización local

3.4. Mejores Casos Individuales

Los mejores casos de discretización local son:

1. **mfeat-fourier** (KDBLd, 4p): 0.7922 → 0.8058 (+1.37 %)
2. **mfeat-morphological** (KDBLd, 4p): 0.6662 → 0.6775 (+1.13 %)

3. **mfeat-fourier** (KDBLd, up): 0.7860 → 0.7968 (+1.08 %)
4. **mfeat-fourier** (AODELd, 3p): 0.7675 → 0.7782 (+1.07 %)
5. **diabetes** (KDBLd, up): 0.7426 → 0.7521 (+0.95 %)

3.5. Casos Negativos y Riesgo

Aunque existen mejoras puntuales, la discretización local puede degradar significativamente el rendimiento en algunos escenarios. En 10 iteraciones, la mediana de la mejora (Local vs mejor base) es de aproximadamente -1.11 puntos porcentuales, con un cuartil inferior alrededor de -4.30 pp, lo que indica que la mayoría de configuraciones no se benefician del enfoque local.

Además, se observan casos de degradación extrema, especialmente en datasets grandes y con muchas clases (por ejemplo, *letter*). La Tabla 2 muestra algunos de los casos más desfavorables.

Cuadro 2: Casos más desfavorables de discretización local (10 iteraciones)

Dataset	**Cortes**	**Modelo**	**Mejor base**	**Acc. base**	**Acc. local**	**Mejora**
letter	up	KDBLd	KDB	0.8797	0.3608	-51.90 %
optdigits	up	KDBLd	KDB-pkilog	0.9485	0.4811	-46.74 %
pendigits	up	KDBLd	KDB-pkilog	0.9717	0.5455	-42.62 %
letter	3p	KDBLd	KDB	0.7376	0.3934	-34.41 %
letter	3p	AODELd	AODE-bin3q	0.6928	0.3564	-33.63 %

4. Estudio Ampliado: 100 Iteraciones

Se realizó un estudio complementario aumentando el límite máximo de iteraciones a 100 para evaluar si la convergencia requiere más ciclos de refinamiento.

4.1. Comparativa 10 vs 100 Iteraciones

- **10 iteraciones**: 56/324 casos positivos (17.3 %)
- **100 iteraciones**: 57/324 casos positivos (17.6 %)

Los resultados con 100 iteraciones muestran patrones prácticamente idénticos a los obtenidos con 10 iteraciones, lo que indica que:

1. La convergencia se alcanza típicamente antes de las 10 iteraciones.
2. El aumento de iteraciones no mejora significativamente el rendimiento.
3. El coste computacional adicional no se justifica con mejoras en accuracy.

5. Análisis Comparativo

5.1. Discretización Supervisada vs No Supervisada

La Figura 3 presenta la comparación entre métodos de discretización supervisada y no supervisada.

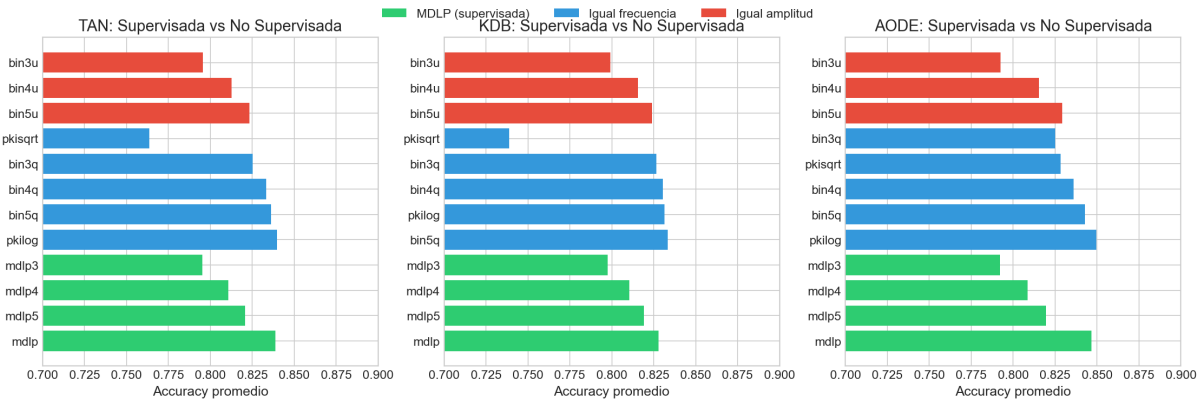

Figura 3: Comparación de accuracy: discretización supervisada vs no supervisada

5.2. Mapa de Calor de Mejoras

La Figura 4 muestra el mapa de calor de mejoras por dataset y modelo.

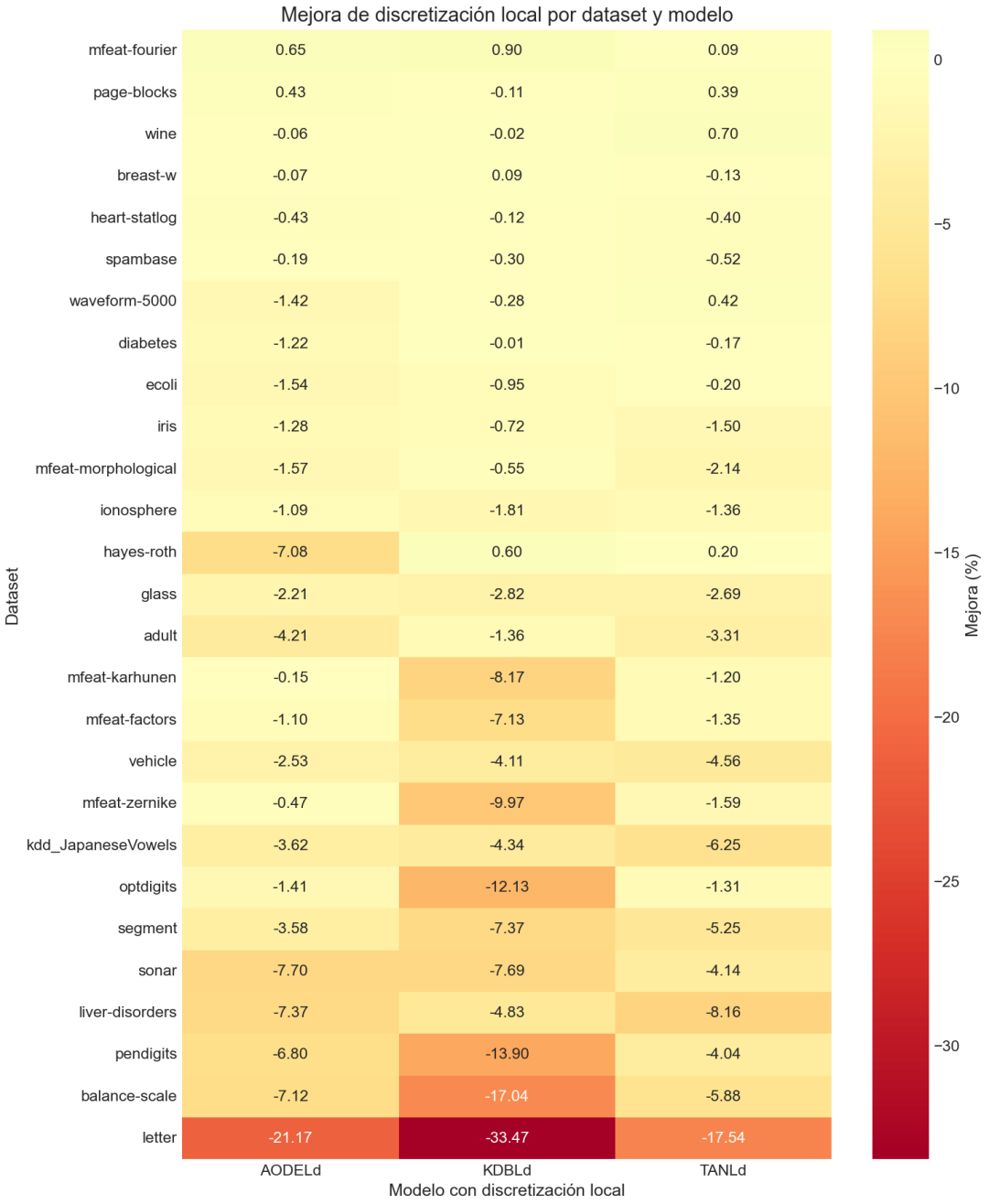

Figura 4: Mapa de calor: mejora porcentual de discretización local por dataset

5.3. Resumen Estadístico Global

La Figura 5 presenta un resumen estadístico completo.

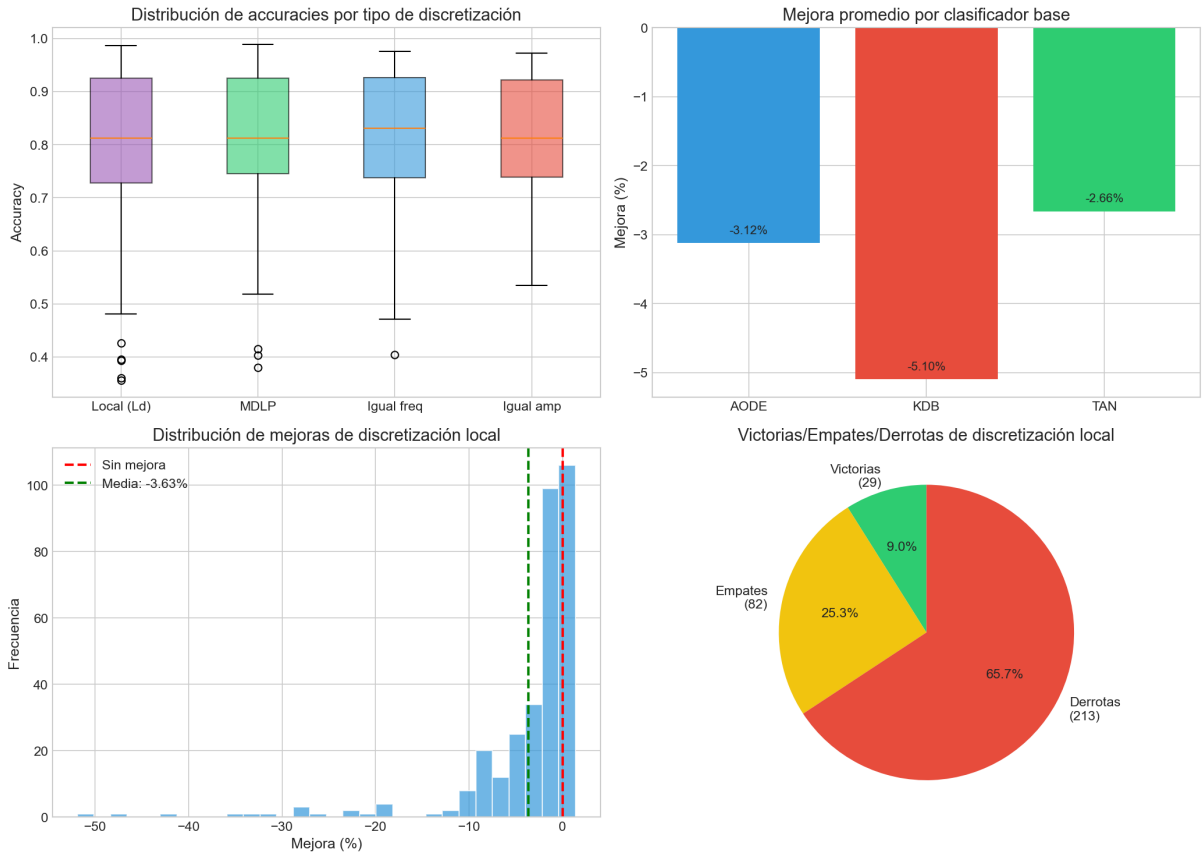

Figura 5: Resumen estadístico global de la discretización local

6. Análisis de Tiempos de Ejecución

6.1. Comparación de Tiempos: 10 vs 100 Iteraciones

La Figura 6 presenta la comparación de tiempos de entrenamiento entre las configuraciones de 10 y 100 iteraciones.

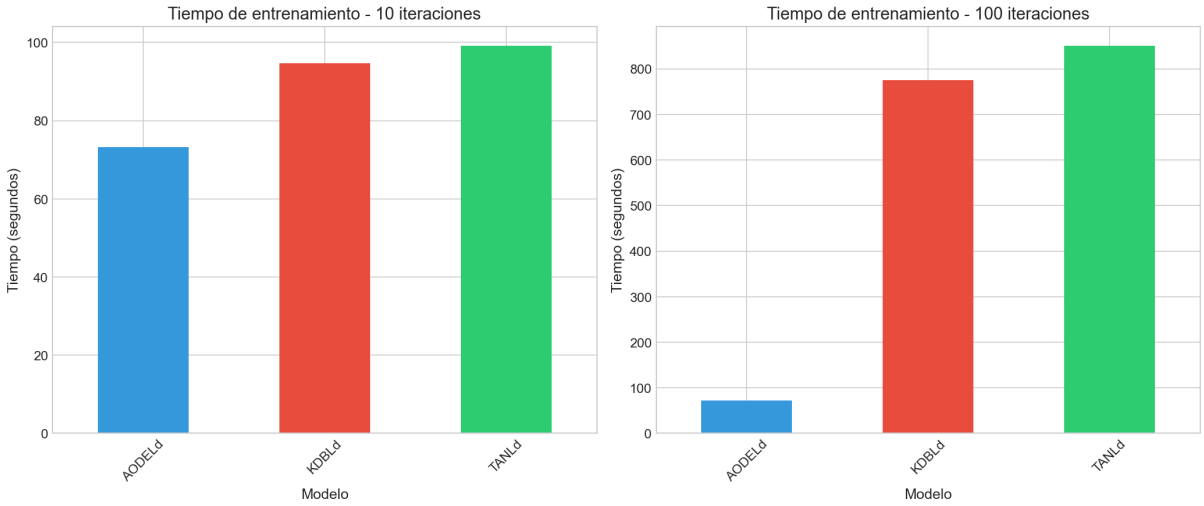

Figura 6: Comparación de tiempos de entrenamiento: 10 vs 100 iteraciones

6.2. Tiempos por Configuración de Puntos de Corte

La Figura 7 muestra cómo varía el tiempo de entrenamiento según la configuración de puntos de corte máximos.

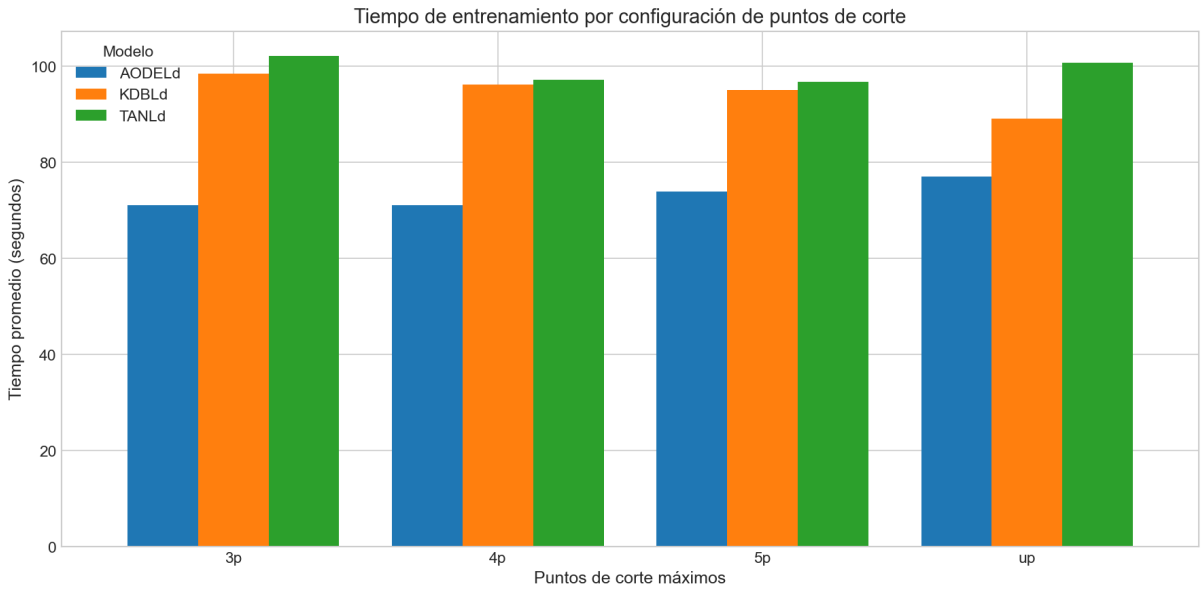

Figura 7: Tiempos de entrenamiento por configuración de puntos de corte

6.3. Datasets más Costosos

La Figura 8 identifica los datasets que requieren mayor tiempo de procesamiento.

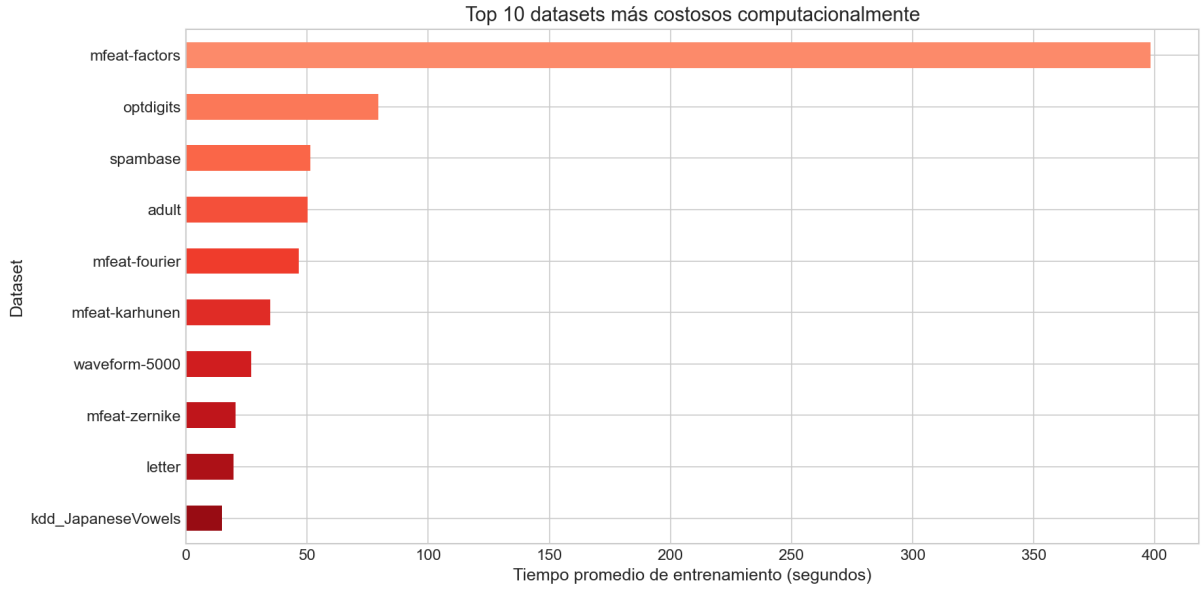

Figura 8: Top 10 datasets más costosos computacionalmente

6.4. Overhead de la Discretización Local

La Figura 9 presenta el overhead introducido por la discretización local respecto a la discretización tradicional.

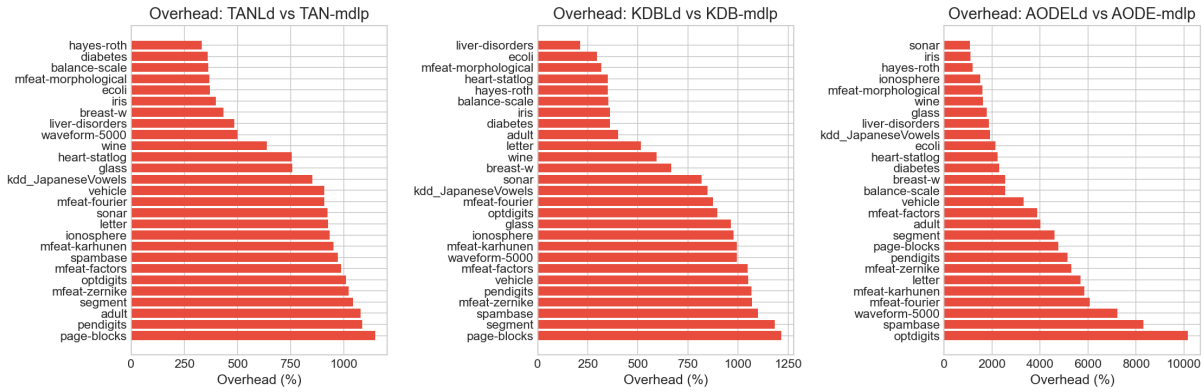

Figura 9: Overhead de discretización local vs tradicional

6.5. Análisis de Escalabilidad

La Figura 10 muestra cómo escala el tiempo de entrenamiento con el tamaño del dataset.

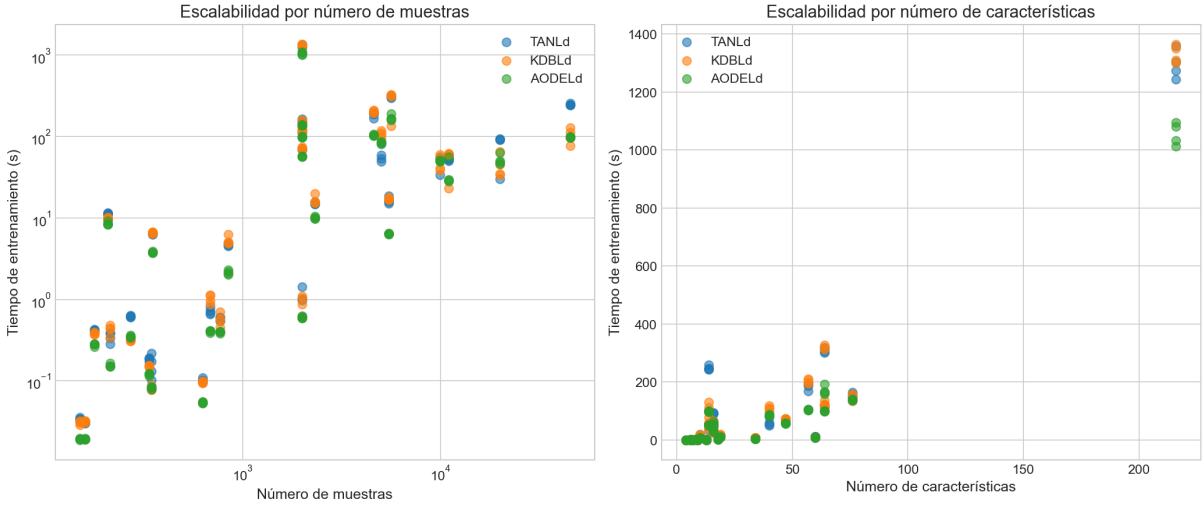

Figura 10: Análisis de escalabilidad: tiempos vs tamaño del dataset

7. Análisis Comparativo del Algoritmo de Discretización MDLP

Esta sección presenta un análisis detallado del rendimiento del algoritmo MDLP (supervisado) comparado con los métodos de discretización no supervisada.

7.1. MDLP vs Igual Frecuencia e Igual Amplitud

La Figura 11 muestra la comparativa de accuracy entre MDLP y los métodos no supervisados.

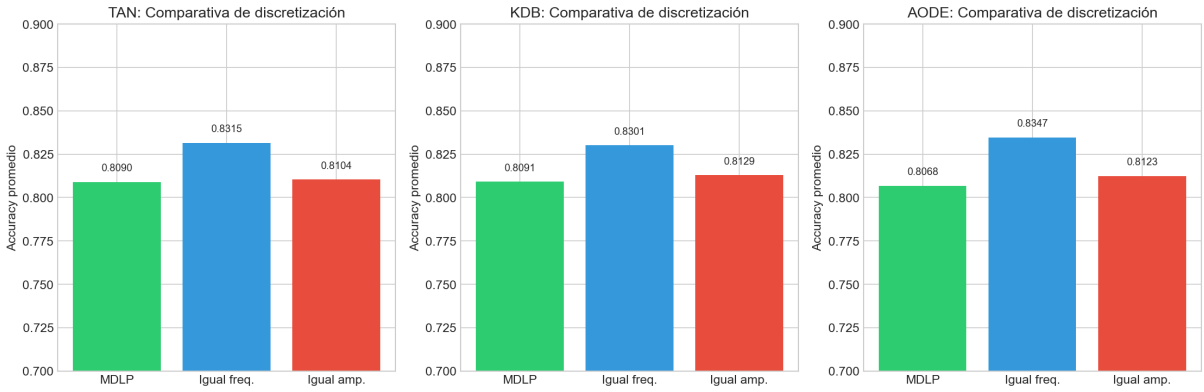

Figura 11: Comparativa MDLP vs discretización no supervisada

7.1.1. Resultados Clave

Los resultados del análisis comparativo revelan un hallazgo importante:

- **MDLP vs Igual Frecuencia**: -2.38 % (MDLP obtiene peor accuracy en promedio)
- **MDLP vs Igual Amplitud**: -0.36 % (MDLP obtiene peor accuracy en promedio)

Esto indica que, contrariamente a lo esperado, el método de discretización supervisada MDLP no supera consistentemente a los métodos no supervisados más simples.

7.2. Tabla Comparativa por Modelo y Configuración

Cuadro 3: Comparativa MDLP vs discretización no supervisada

Modelo	Cortes	MDLP	Igual Freq.	Igual Amp.
TAN	3p	0.7954	0.8251	0.7955
TAN	4p	0.8108	0.8334	0.8126
TAN	5p	0.8208	0.8361	0.8232
KDB	3p	0.7975	0.8265	0.7992
KDB	4p	0.8105	0.8306	0.8155
KDB	5p	0.8193	0.8333	0.8240
AODE	3p	0.7921	0.8251	0.7923
AODE	4p	0.8085	0.8361	0.8154
AODE	5p	0.8197	0.8429	0.8292

7.3. Análisis por Dataset

La Figura 12 muestra la diferencia de accuracy entre MDLP e igual frecuencia por cada dataset.

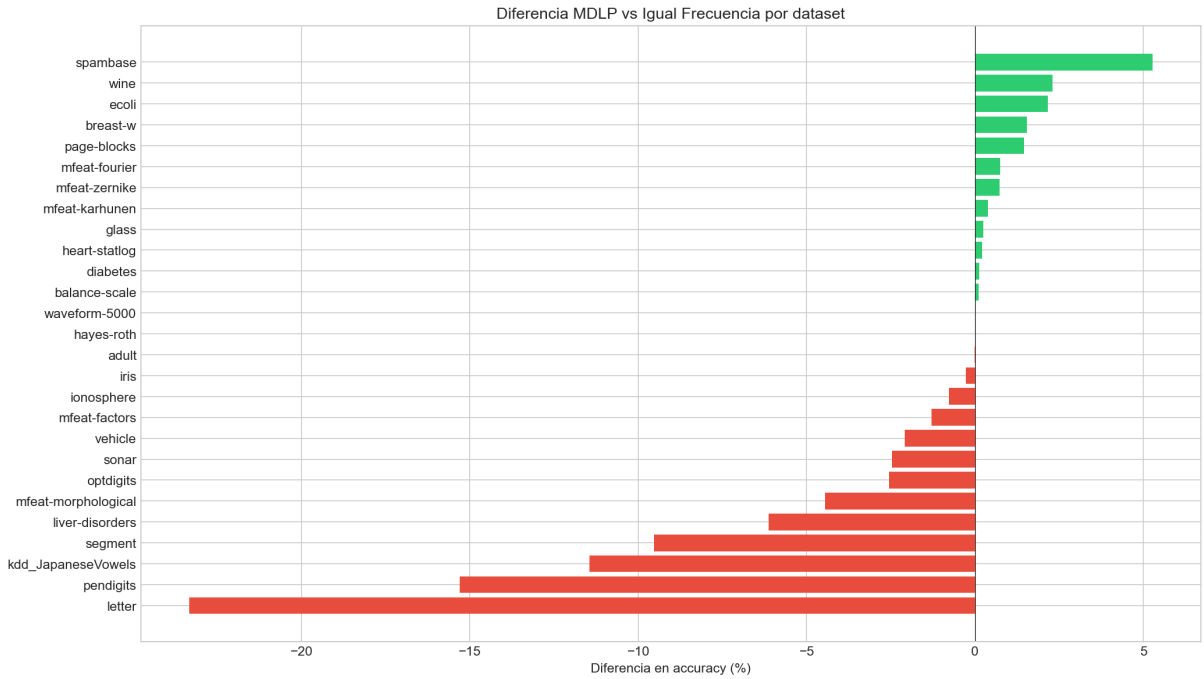

Figura 12: Diferencia MDLP vs Igual Frecuencia por dataset

7.4. MDLP vs PKI

Para la configuración de puntos de corte ilimitados, se comparó MDLP con las variantes PKI:

- **PKI-sqrt**: Utiliza $k = \sqrt{\text{features}}$ intervalos
- **PKI-log**: Utiliza $k = \ln(\text{features})$ intervalos

En los resultados agregados, PKI-log es competitivo y en promedio supera ligeramente a MDLP (del orden de +0.25 puntos porcentuales), pero no de forma consistente en todos los datasets. Por el contrario, PKI-sqrt tiende a degradar el rendimiento de manera marcada (aprox. -6.08 pp de media), siendo claramente inferior a MDLP en la mayoría de casos.

7.5. Interpretación de Resultados

El rendimiento inferior de MDLP frente a métodos no supervisados puede explicarse por:

1. **Sobreajuste**: MDLP puede generar demasiados puntos de corte adaptados al conjunto de entrenamiento.
2. **Sensibilidad al ruido**: La ganancia de información puede ser engañosa en presencia de ruido.
3. **Simplicidad efectiva**: Métodos más simples como igual frecuencia proporcionan una discretización más robusta y generalizable.

8. Conclusiones

8.1. Hallazgos Principales

1. **La discretización local mejora en casos específicos**: Aunque globalmente la discretización local no supera a los métodos tradicionales, se identificaron 113 casos (17.4 %) donde proporciona mejoras, con incrementos de hasta +1.37 % en accuracy.

2. **Existe riesgo de degradación severa**: Además de los casos positivos, aparecen degradaciones muy pronunciadas en algunos datasets (p.ej. *letter* con KDBLd y cortes ilimitados), por lo que el uso de discretización local debe validarse siempre por dataset/configuración.

3. **Datasets beneficiados**: Los datasets mfeat-fourier, mfeat-morphological, diabetes, adult y wine son los más beneficiados por la discretización local. Estos datasets comparten características como:

   - Número moderado de características
   - Valores continuos con distribuciones complejas
   - Correlaciones entre variables

4. **Configuración óptima**: La configuración con puntos de corte ilimitados (up) presenta el mayor porcentaje de casos positivos (22.2 %), sugiriendo que la flexibilidad en la discretización beneficia al enfoque local.

5. **TANLd y KDBLd concentran las mejoras**: En 10 iteraciones, TANLd y KDBLd presentan el mismo porcentaje de casos positivos (20.4 %). KDBLd alcanza la mayor mejora observada (+1.37 %), pero también muestra los peores casos, lo que sugiere un trade-off entre potencial de mejora y riesgo.

6. **MDLP no supera a igual frecuencia**: Sorprendentemente, el método supervisado MDLP obtiene en promedio 2.38 % menos accuracy que el método no supervisado de igual frecuencia, cuestionando la superioridad asumida de los métodos supervisados.

7. **Las iteraciones adicionales no ayudan**: No hay diferencia significativa entre 10 y 100 iteraciones, indicando convergencia rápida del algoritmo.

8.2. Recomendaciones Prácticas

Basándose en los resultados experimentales, se recomienda:

1. **Usar discretización local selectivamente**: Considerar la discretización local para datasets con características similares a mfeat-fourier, diabetes o adult (múltiples variables numéricas correlacionadas).

2. **Preferir igual frecuencia**: Para casos generales, la discretización de igual frecuencia ofrece el mejor balance entre simplicidad y rendimiento.

3. **Configuración de puntos de corte ilimitados**: Si se opta por discretización local, usar la configuración sin límite de puntos de corte.

4. **Elegir según objetivo (mejora vs robustez)**: KDBLd ofrece el mayor potencial de mejora, pero es más propenso a degradaciones severas; TANLd tiende a ser una alternativa más conservadora. AODELd muestra menos casos positivos.

5. **Limitar iteraciones a 10**: No hay beneficio en aumentar el número de iteraciones más allá de 10.

8.3. Cuándo Usar Discretización Local

Cuadro 4: Guía para uso de discretización local

Escenario	Recomendación	Justificación
Dataset general	No usar	Métodos tradicionales son mejores
Muchas variables numéricas	Evaluar	Potencial beneficio
Correlaciones entre variables	Evaluar	Mayor probabilidad de mejora
Dataset tipo mfeat-*	Sí usar	Beneficio demostrado
Datasets pequeños (¡500)	Evaluar	Puede mejorar, pero con alta varianza
Datasets grandes (¿5000)	Evaluar	Coste computacional alto

8.4. Direcciones Futuras

1. Investigar características específicas de los datasets donde la discretización local funciona mejor.
2. Desarrollar heurísticas para predecir cuándo usar discretización local.
3. Evaluar mecanismos de regularización para reducir el sobreajuste.
4. Estudiar la combinación de discretización local con otros métodos de preprocesamiento.

A. Apéndices

A.1. Detalles Técnicos del Algoritmo de Discretización Local

El algoritmo de discretización local sigue el siguiente pseudocódigo:

```
Algoritmo: Discretización Local Iterativa
Entrada: Dataset D, Clasificador C, max_iteraciones
Salida: Clasificador C' con discretización optimizada

1. D' = discretizar_global(D, MDLP)
2. C = entrenar(C, D')
3. Para i = 1 hasta max_iteraciones:
   4. Para cada variable X en D:
      5. padres = obtener_padres(X, C)
      6. X' = re_discretizar(X, padres, clase)
   7. D'' = aplicar_nueva_discretizacion(D, X')
   8. C' = entrenar(C, D'')
   9. Si convergencia(C, C'):
      10. retornar C'
   11. C = C'
12. retornar C
```

A.2. Análisis de Datasets con Mejores Resultados

Los datasets donde la discretización local muestra mejores resultados comparten ciertas características:

- **mfeat-fourier** (76 características): Dataset de reconocimiento de dígitos con coeficientes de Fourier. La alta dimensionalidad y correlación entre características beneficia la re-discretización local.

- **mfeat-morphological** (6 características): Características morfológicas de dígitos escritos a mano. Distribuciones no estándar de los valores.
- **diabetes** (8 características): Dataset médico con variables correlacionadas relacionadas con factores de riesgo.
- **adult** (14 características, 6 numéricas): Dataset de censo con mezcla de variables categóricas y numéricas.

A.3. Resultados Completos

Las tablas completas de resultados para cada configuración de puntos de corte se incluyen en los archivos:

- `tables/accuracy_comparison_3p.tex`
- `tables/accuracy_comparison_4p.tex`
- `tables/accuracy_comparison_5p.tex`
- `tables/accuracy_comparison_up.tex`
- `tables/top_positive_cases.tex`
- `tables/mdlp_comparison_table.tex`

Los datos procesados están disponibles en:

- `accuracy_summary_10it.csv`
- `improvements_summary.csv`
- `positive_local_cases.csv`
- `mdlp_comparison.csv`

Referencias

1. Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.
2. Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. Machine Learning.
3. Webb, G. I., Boughton, J. R., & Wang, Z. (2005). Not so naive Bayes: Aggregating one-dependence estimators. Machine Learning.
4. Sahami, M. (1996). Learning limited dependence Bayesian classifiers.