

Predicción de Clase de Animal mediante Clasificadores Bayesianos, Estimadores No Paramétricos y k-NN

Jordi Blasco Lozano

74527208D

11 de noviembre de 2025

Resumen

Este trabajo aborda la práctica 2 en la asignatura de Fundamentos del Aprendizaje Automático empleando el dataset Zoo de UCI, aplicando los algoritmos requeridos y analizando sus resultados en una tarea real de clasificación con múltiples clases y características binarias. Finalmente se compararán los seis modelos diferentes y explicaremos

1 Introducción y justificación del dataset

He seleccionado el **dataset Zoo** de UCI para esta práctica porque cumple estrictamente con los requerimientos del enunciado y permite aplicar eficazmente todos los métodos estudiados en la asignatura. Las razones principales son:

- **Clasificación multiclase:** El enunciado exige explícitamente abordar problemas con múltiples clases, no solo binarios. El dataset Zoo tiene 7 clases diferentes (mamífero, ave, reptil, pez, anfibio, invertebrado, insecto).
- **Características interpretables:** 16 atributos binarios (presencia/ausencia de características físicas o comportamentales) que facilitan el análisis y la aplicación de estimadores de densidad.
- **Tamaño adecuado:** 101 instancias, suficientes para validación pero pequeñas para observar comportamientos de los métodos no paramétricos con datos limitados.

El objetivo es comparar el rendimiento de clasificadores bayesianos paramétricos versus estimadores no paramétricos de densidad, analizando sus fortalezas y debilidades en clasificación multiclase con datos de tamaño moderado.

2 Dataset y Análisis Exploratorio

El **dataset Zoo** consta de:

- **Tamaño:** 101 instancias (animales)
- **Dimensionalidad:** 16 atributos binarios (0/1) que describen características físicas y comportamentales
- **Clases:** 7 tipos de animales (multiclase):
 - Mamífero (mammal): 41 muestras (40.6 %)
 - Ave (bird): 20 muestras (19.8 %)
 - Reptil (reptile): 5 muestras (5.0 %)
 - Pez (fish): 13 muestras (12.9 %)
 - Anfibio (amphibian): 4 muestras (4.0 %)
 - Invertebrado (invertebrate): 8 muestras (7.9 %)
 - Insecto (insect): 10 muestras (9.9 %)

2.1 Características del dataset

El dataset presenta una distribución de clases moderadamente desbalanceada, con mamíferos dominando (40.6 %) y anfibios siendo la clase minoritaria (4.0 %). Esta distribución es más realista que un desbalance extremo y permite evaluar cómo los diferentes métodos manejan clases con pocas muestras.

Las 16 características binarias incluyen atributos como: tiene pelo, pone huevos, vuela, acuático, depredador, con dientes, columna vertebral, respira, venenoso, con aletas, número de patas, cola, doméstico, tamaño, etc.

2.2 Preprocesamiento

No se requiere normalización porque todas las características son binarias (0 o 1). Los datos están completos sin valores faltantes. El único preprocesamiento necesario es la codificación de las etiquetas de clase a valores numéricos (0-6) para los algoritmos.

3 Metodología: Particionado y Validación

3.1 División Train-Test (80 %-20 %)

He implementado una división estratificada del dataset:

- **Conjunto de entrenamiento:** 80 muestras (79.2 %)
- **Conjunto de prueba:** 21 muestras (20.8 %)

La estratificación garantiza que la proporción de clases se mantenga en ambos conjuntos. Dada la distribución desbalanceada (especialmente las clases minoritarias con 4-5 ejemplos), la estratificación es crucial para asegurar que todas las clases estén representadas en ambos conjuntos.

Distribución en conjunto de prueba:

- Mamífero: 8 muestras
- Ave: 4 muestras
- Reptil: 1 muestra
- Pez: 3 muestras
- Anfibio: 1 muestra
- Invertebrado: 2 muestras
- Insecto: 2 muestras

3.2 Validación Cruzada para Hiperparámetros

Para la optimización de hiperparámetros (bandwidth en Parzen, k en k-NN), utilizo validación cruzada estratificada de 5 folds **únicamente sobre el conjunto de entrenamiento**. Esto evita contaminación de datos (data leakage) y garantiza que las métricas del conjunto de prueba sean imparciales.

Nota importante: Dado que algunas clases tienen muy pocas muestras (3-4 en entrenamiento), scikit-learn genera warnings sobre la validación cruzada 5-fold. Esto es esperado y no afecta significativamente los resultados, pero refleja las limitaciones de trabajar con clases muy minoritarias.

3.3 Metodología de Evaluación

- **Métricas principales:** Accuracy y F1-macro
- **F1-macro:** Promedia el F1-score de todas las clases, tratándolas con igual importancia (crítico para datasets desbalanceados)
- **Matrices de confusión:** Para análisis detallado clase por clase
- **Reporte de clasificación completo:** Precision, recall y F1 por clase

4 Modelos Implementados

He implementado y comparado seis clasificadores siguiendo los contenidos de teoría (T2-T4):

1. **Naive Bayes Gaussiano (T2)**: Clasificador paramétrico con independencia condicional
2. **MLE Multivariante (Full Bayesian Gaussian) (T2)**: Estimación máxima verosimilitud con covarianza completa
3. **Histogram Bayes (T4)**: Estimador no paramétrico basado en histogramas
4. **Parzen Windows (T4)**: Estimador no paramétrico con kernel Gaussiano
5. **k-NN Density Bayes (T4)**: Estimador no paramétrico basado en densidad local
6. **k-NN Rule (T4)**: Regla de los k vecinos más cercanos clásica

4.1 Naive Bayes Gaussiano (T2)

4.1.1 Fundamento teórico

Naive Bayes aplica el teorema de Bayes con la asunción de independencia condicional entre características:

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y)$$

Para cada clase y y característica i , estima $\mu_{y,i}$ y $\sigma_{y,i}^2$ asumiendo distribución Gaussiana:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_{y,i}^2}} \exp\left(-\frac{(x_i - \mu_{y,i})^2}{2\sigma_{y,i}^2}\right)$$

A pesar de que la asunción de independencia raramente se cumple en la práctica, Naive Bayes suele funcionar sorprendentemente bien, especialmente cuando las correlaciones no son excesivamente fuertes.

4.2 MLE Multivariante - Full Bayesian Gaussian (T2)

4.2.1 Fundamento teórico

A diferencia de Naive Bayes, este método **no asume independencia** entre características. Estima la matriz de covarianza completa Σ_y para cada clase:

$$P(\mathbf{x}|y) = \frac{1}{(2\pi)^{d/2} |\Sigma_y|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_y)^T \Sigma_y^{-1} (\mathbf{x} - \boldsymbol{\mu}_y)\right)$$

Donde $\boldsymbol{\mu}_y$ es el vector de medias y Σ_y es la matriz de covarianza $d \times d$ para la clase y .

Limitaciones: Con d características, se deben estimar $d(d+1)/2$ parámetros para cada matriz de covarianza. Esto requiere muchas muestras por clase. Con clases minoritarias (4-5 ejemplos), la matriz puede ser singular o mal condicionada.

4.3 Histogram Bayes (T4)

4.3.1 Fundamento teórico

Estimador no paramétrico que divide el espacio de características en bins (histogramas) y estima la densidad como:

$$\hat{p}(\mathbf{x}|y) = \frac{\text{count}(\mathbf{x} \in \text{bin})}{n_y \cdot V_{\text{bin}}}$$

Donde n_y es el número de ejemplos de la clase y y V_{bin} es el volumen del bin.

Problema de la maldición de la dimensionalidad: Con d dimensiones y b bins por dimensión, se necesitan b^d bins. Con $d = 16$, incluso con $b = 2$ (binario), tenemos $2^{16} = 65536$ posibles combinaciones. Con solo 101 muestras totales, la mayoría de bins estarán vacíos, resultando en estimaciones muy pobres.

4.4 Parzen Windows (T4)

4.4.1 Fundamento teórico

Estimador no paramétrico que usa un kernel (típicamente Gaussiano) para suavizar la estimación de densidad:

$$\hat{p}(\mathbf{x}|y) = \frac{1}{n_y} \sum_{i=1}^{n_y} K_h(\mathbf{x} - \mathbf{x}_i^{(y)})$$

Donde K_h es un kernel Gaussiano con ancho de banda (bandwidth) h :

$$K_h(\mathbf{u}) = \frac{1}{(2\pi h^2)^{d/2}} \exp\left(-\frac{\|\mathbf{u}\|^2}{2h^2}\right)$$

El parámetro h controla el suavizado: valores pequeños capturan detalles locales pero son sensibles al ruido, valores grandes producen estimaciones más suaves pero pueden perder estructura.

Ventaja sobre histogramas: Parzen windows produce estimaciones continuas y suaves, evitando discontinuidades artificiales de los bins.

4.5 k-NN Density Bayes (T4)

4.5.1 Fundamento teórico

Estimador de densidad basado en la distancia al k -ésimo vecino más cercano:

$$\hat{p}(\mathbf{x}|y) = \frac{k}{n_y \cdot V_k(\mathbf{x})}$$

Donde $V_k(\mathbf{x})$ es el volumen de la esfera que contiene los k vecinos más cercanos de \mathbf{x} en la clase y .

Este método adapta el volumen localmente: en regiones densas usa volúmenes pequeños, en regiones dispersas usa volúmenes grandes.

4.6 k-NN Rule (T4)

4.6.1 Fundamento teórico

El método más simple de k-NN: asigna la clase mayoritaria entre los k vecinos más cercanos:

$$\hat{y} = \arg \max_c \sum_{i \in N_k(\mathbf{x})} \mathbb{I}(y_i = c)$$

Donde $N_k(\mathbf{x})$ son los índices de los k vecinos más cercanos a \mathbf{x} .

Diferencia con k-NN Density: k-NN Rule cuenta votos directamente, mientras que k-NN Density estima probabilidades vía densidades locales y usa Bayes para clasificar.

5 Resultados

He evaluado los seis modelos en el conjunto de prueba de 21 muestras. La Tabla 1 muestra los resultados principales.

5.1 Análisis detallado por modelo

5.1.1 1. Naive Bayes Gaussiano - Rendimiento perfecto

Resultados en test: Accuracy = 1.0, F1-macro = 1.0

Naive Bayes logra clasificación perfecta en el conjunto de prueba, clasificando correctamente las 21 muestras. La matriz de confusión muestra ceros fuera de la diagonal principal.

Interpretación: A pesar de la asunción "naive" de independencia entre características, el modelo funciona excepcionalmente bien. Esto sugiere que:

Modelo	Accuracy	F1-macro
Naive Bayes	1.0000	1.0000
MLE Full	0.7143	0.4563
Histogram Bayes	0.3810	0.0788
Parzen Bayes (h=0.1)	1.0000	1.0000
k-NN Density (k=11)	0.4762	0.5714
k-NN Rule (k=1)	1.0000	1.0000

Cuadro 1: Resultados en conjunto de prueba (21 muestras, 7 clases)

Modelo	F1-macro CV (mean)	F1-macro CV (std)
Naive Bayes	0.8505	0.1357
MLE Full	0.5329	0.1021
Histogram Bayes	0.2474	0.1277
Parzen Bayes (h=0.1)	0.8648	—
k-NN Density (k=11)	0.5664	—
k-NN Rule (k=1)	0.8267	—

Cuadro 2: Validación cruzada 5-fold en conjunto de entrenamiento (80 muestras)

- Las características binarias están relativamente poco correlacionadas
- La separación entre clases es clara en el espacio de características
- Las distribuciones Gaussianas aproximan bien los datos binarios en este caso

CV: F1-macro = 0.8505 ± 0.1357 . La desviación estándar moderada refleja variabilidad en algunos folds debido a clases minoritarias, pero el rendimiento en test confirma la robustez del modelo.

5.1.2 2. MLE Full - Problemas con clases minoritarias

Resultados en test: Accuracy = 0.7143, F1-macro = 0.4563

El modelo multivariante con covarianza completa falla significativamente. La matriz de confusión revela que:

- Clasifica correctamente mamíferos (8/8) y aves (4/4)
- Falla completamente en reptiles (0/1), anfibios (0/1) e invertebrados (0/2)
- Clasificando erróneamente estas clases minoritarias como mamíferos

Causa del fallo: Con solo 16 características, la matriz de covarianza 16×16 requiere estimar 136 parámetros por clase. Para clases con 3-6 ejemplos en entrenamiento, esto es imposible:

- Matrices de covarianza singulares o mal condicionadas
- Warnings de división por cero durante el cálculo de densidades
- El modelo colapsa prediciendo solo las clases mayoritarias

Lección: La estimación multivariante completa **requiere muchas más muestras** que Naive Bayes. Con clases minoritarias, la asunción de independencia de Naive Bayes es una ventaja, no una limitación.

5.1.3 3. Histogram Bayes - Colapso por maldición dimensionalidad

Resultados en test: Accuracy = 0.3810, F1-macro = 0.0788

El peor rendimiento de todos los modelos. La matriz de confusión muestra que predice solo mamíferos:

- Mammal: 8/8 correctos

- Todas las demás clases: 0 % de recall

Explicación: Con 16 características binarias, hay $2^{16} = 65536$ posibles combinaciones (bins). Con solo 101 muestras totales y 41 mamíferos:

- La mayoría de bins están vacíos ($p(\mathbf{x}|y) = 0$)
- Solo los bins con mamíferos tienen densidad estimada
- El modelo predice siempre la clase mayoritaria por defecto

Maldición de la dimensionalidad: Caso de libro de texto sobre por qué los histogramas fallan en alta dimensionalidad, incluso con características binarias.

5.1.4 4. Parzen Windows - Rendimiento perfecto

Resultados en test: Accuracy = 1.0, F1-macro = 1.0 **Hiperparámetro óptimo:** bandwidth $h = 0.1$

Parzen windows logra clasificación perfecta, igualando a Naive Bayes y k-NN Rule. Con $h=0.1$:

- El kernel Gaussiano proporciona suavizado local adecuado
- Evita las discontinuidades artificiales de los histogramas
- Captura la estructura de las clases sin overfitting

CV: F1-macro = 0.8648 (el mejor en validación cruzada), superior incluso a Naive Bayes (0.8505). Esto confirma que el bandwidth optimizado generaliza bien.

Ventaja clave: A diferencia de histogramas, Parzen produce estimaciones continuas que funcionan bien incluso con datos relativamente escasos. El suavizado del kernel compensa la falta de muestras.

5.1.5 5. k-NN Density Bayes - Rendimiento mediocre

Resultados en test: Accuracy = 0.4762, F1-macro = 0.5714 **Hiperparámetro óptimo:** $k = 11$
Rendimiento pobre en test. Análisis de la matriz de confusión:

- Confunde mamíferos (0/8 correctos) clasificándolos como reptiles
- Acierta en aves (4/4), invertebrados (2/2) e insectos (1/2)
- Resultados erráticos y poco fiables

Problema: Con $k=11$ en un dataset de 80 muestras entrenamiento:

- Los 11 vecinos más cercanos diluyen la señal local
- La estimación de densidad por volumen local es inestable
- Clases con distribuciones espaciales dispersas sufren especialmente

Comparación con Parzen: Ambos son estimadores no paramétricos, pero Parzen ($h=0.1$) fija el ancho de banda globalmente mientras k-NN adapta el volumen localmente. En este caso, la estrategia fija de Parzen es superior.

5.1.6 6. k-NN Rule - Rendimiento perfecto

Resultados en test: Accuracy = 1.0, F1-macro = 1.0 **Hiperparámetro óptimo:** $k = 1$

El método clásico de k-NN logra clasificación perfecta con $k=1$ (vecino más cercano). Esto indica:

- Las clases están bien separadas en el espacio de características
- Los ejemplos de test tienen vecinos cercanos de su misma clase en entrenamiento
- La estructura del dataset es relativamente simple

CV: F1-macro = 0.8267, ligeramente inferior a Parzen (0.8648) y Naive Bayes (0.8505) en validación cruzada, pero perfecto en test.

Ventaja de $k=1$: Con datos bien separados y sin ruido significativo, el vecino más cercano es suficiente. Valores mayores de k diluirían innecesariamente la señal.

6 Discusión y Conclusiones

6.1 Comparación general de enfoques

6.1.1 Modelos paramétricos vs no paramétricos

Los resultados revelan patrones claros sobre cuándo cada enfoque es apropiado:

Modelos paramétricos exitosos:

- **Naive Bayes:** Perfecto en test (1.0/1.0). La asunción de independencia, aunque incorrecta teóricamente, no penaliza en este dataset.
- **Ventaja:** Pocas suposiciones (solo medias y varianzas), robusto con pocas muestras.

Modelo paramétrico fallido:

- **MLE Full:** Falla con clases minoritarias (0.71/0.46). Estimación de covarianza completa imposible con 3-6 ejemplos.
- **Lección:** Más parámetros \neq mejor modelo. Con datos limitados, modelos simples ganan.

Modelos no paramétricos exitosos:

- **Parzen:** Perfecto (1.0/1.0) con $h=0.1$ óptimo. Suavizado continuo supera limitaciones de histogramas.
- **k-NN Rule:** Perfecto (1.0/1.0) con $k=1$. Simplicidad efectiva cuando datos están separados.

Modelos no paramétricos fallidos:

- **Histogram:** Colapso total (0.38/0.08). Maldición dimensionalidad en acción.
- **k-NN Density:** Mediocre (0.48/0.57). Estimación de densidad por volumen local inestable.

6.2 Impacto de clases minoritarias

El dataset Zoo tiene clases con solo 3-4 ejemplos en entrenamiento (anfibios, reptiles). Observamos:

- **Naive Bayes:** Maneja bien clases minoritarias. Estima μ y σ^2 por característica (16 parámetros/clase).
- **MLE Full:** Falla con minoritarias. Necesita estimar 136 parámetros/clase (matriz 16×16).
- **Parzen y k-NN Rule:** No estiman parámetros por clase, usan similitud directa. Funcionan bien si ejemplos están separados.
- **Histogram:** Fragmenta los pocos ejemplos en bins vacíos. Imposible estimar densidad.

Principio general: Con clases minoritarias, preferir modelos que requieran **pocos parámetros por clase** o **métodos basados en distancia** sin estimación explícita de densidad.

6.3 Métricas: Accuracy vs F1-macro

En dataset multiclase con distribución desbalanceada (40 % mamíferos vs 4 % anfibios):

- **Histogram Bayes:** Accuracy 38 % pero F1-macro 7.9 %. Predice solo mamíferos.
- **F1-macro:** Promedia rendimiento de todas las clases. Penaliza modelos que ignoran minoritarias.
- **Uso correcto:** F1-macro es métrica principal para datasets desbalanceados multiclase.

Los tres modelos perfectos (Naive Bayes, Parzen, k-NN Rule) tienen Accuracy = F1-macro = 1.0, confirmando clasificación genuinamente balanceada.

6.4 Validación de la metodología

La división train-test estratificada (80 %-20 %) con validación cruzada interna para hiperparámetros ha sido fundamental:

- **Evaluación imparcial:** El test set nunca se usa durante entrenamiento u optimización
- **Hiperparámetros óptimos:** GridSearchCV encuentra $h=0.1$ (Parzen), $k=11$ (k-NN Density), $k=1$ (k-NN Rule)
- **Coherencia CV-Test:** Los modelos con mejor F1-macro en CV (Parzen 0.865, NB 0.851) también son perfectos en test
- **Detección de inestabilidad:** k-NN Density con $F1-CV=0.566$ confirma su mediocridad en test (0.571)

6.5 Conclusiones finales

6.5.1 Resultados principales

En el dataset Zoo multiclase con clases minoritarias:

1. **Tres modelos perfectos:** Naive Bayes, Parzen Windows ($h=0.1$) y k-NN Rule ($k=1$) logran clasificación perfecta (Accuracy = F1-macro = 1.0)
2. **MLE Full falla:** Accuracy 71.4 %, F1-macro 45.6 %. No maneja clases con 3-6 ejemplos (matrices singulares)
3. **Histogram colapsa:** Accuracy 38.1 %, F1-macro 7.9 %. Maldición dimensionalidad (2^{16} bins, 101 muestras)
4. **k-NN Density mediocre:** Accuracy 47.6 %, F1-macro 57.1 %. Estimación de volumen local inestable

6.5.2 Lecciones sobre teoría T2-T4

De teoría paramétrica (T2):

- La **asunción de independencia** de Naive Bayes no siempre penaliza. Con características poco correladas, simplifica y robustece.
- **Modelos complejos necesitan datos:** MLE Full con covarianza completa requiere $O(d^2)$ muestras/clase. Con clases minoritarias, Naive Bayes ($O(d)$ parámetros) gana.
- La **parsimonia** (pocos parámetros) es ventaja con datos limitados, no limitación.

De teoría no paramétrica (T4):

- **Histogramas vs Parzen:** Los histogramas sufren severamente curse of dimensionality. Parzen con suavizado continuo lo mitiga.
- **Bandwidth crítico:** $h=0.1$ óptimo en Parzen. Muy pequeño (overfitting) o grande (underfit) fallan.
- **k-NN simple vs k-NN density:** k-NN Rule (votación directa) supera a k-NN Density (estimación explícita). Menos pasos \rightarrow menos fuentes de error.
- **k=1 efectivo:** Cuando datos están separados sin ruido, el vecino más cercano es suficiente y óptimo.

6.5.3 Importancia del F1-macro

El F1-macro ha sido crucial para detectar modelos que ignoran clases minoritarias:

- Histogram: 38.1 % accuracy parece razonable”, pero F1-macro 7.9 % revela que predice solo mamíferos
- MLE Full: 71.4 % accuracy oculta fallo total en 3 clases (0 % recall)
- Los modelos perfectos tienen Accuracy = F1-macro = 1.0, confirmando balance genuino

En clasificación multiclase desbalanceada, **reportar solo accuracy es científicamente inaceptable**.

6.5.4 Conclusión práctica

Para el dataset Zoo:

- **Recomendación principal:** Naive Bayes, Parzen ($h=0.1$) o k-NN Rule ($k=1$). Los tres son perfectos y simples.
- **Evitar:** MLE Full (falla con minoritarias), Histogram (curse of dimensionality), k-NN Density (inestable)
- **Principio general:** En datos pequeños multiclase, preferir modelos **simples con pocas suposiciones** (Naive Bayes) o **basados en distancia sin estimación** (k-NN directo, Parzen).

Esta práctica demuestra que complejidad \neq mejor rendimiento. Los clasificadores más sencillos (Naive Bayes, k-NN $k=1$) igualan o superan a métodos sofisticados cuando los datos tienen estructura simple y están bien separados. La teoría T2-T4 predice exactamente estos comportamientos: los resultados empíricos validan la teoría.