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

Jordi Blasco Lozano

Abstract

Este trabajo aborda la práctica 2 en la asignatura de Fundamentos del Aprendizaje Automático empleando el dataset Zoo de UCI, aplicando seis algoritmos de clasificación: Naive Bayes Gaussiano, MLE Multivariante, Histogram Bayes, Parzen Windows, k-NN Density Bayes y k-NN Rule. Se analiza su rendimiento en una tarea real de clasificación multiclase (7 clases) con características binarias y clases minoritarias. Los resultados revelan que Naive Bayes, Parzen Windows y k-NN Rule logran clasificación perfecta, mientras que métodos más complejos fallan debido a la maldición de la dimensionalidad y la escasez de datos en clases minoritarias.

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, mi conjunto de entrenamiento será de 80 muestras y el de prueba de 21.

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.

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.

3.3. Metodología de Evaluación

Para medir cada modelo utilizaremos las métricas siguientes:

- F1-macro: Promedia el F1-score de todas las clases, tratándolas con igual importancia, dado nuestro dataset desbalanceado será nuestra métrica base.
- Metricas secundarias: Precision, recall para observar como de bueno es cada modelo en falsos positivos y falsos negativos de forma separada.
- Matrices de confusión: Para análisis detallado clase por clase.

4. Modelos Implementados

He implementado y comparado seis clasificadores:

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

5. Resultados

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

Table 1. Resultados en conjunto de prueba (21 muestras, 7 clases)

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

Table 2. Validación cruzada 5-fold en conjunto de entrenamiento

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

5.1. Análisis detallado por modelo

5.1.1. Naive Bayes Gaussiano

Test: Accuracy = 1.0, F1-macro = 1.0

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

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

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

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

Análisis: Naive Bayes logra clasificación perfecta en el conjunto de prueba, clasificando correctamente las

21 muestras. A pesar de la asunción "naive" de independencia entre características, el modelo funciona excepcionalmente bien. Esto sugiere que las características binarias están relativamente poco correlacionadas y que las distribuciones Gaussianas aproximan bien los datos binarios en este caso. La asunción de independencia raramente se cumple en la práctica, pero en este caso no penaliza el rendimiento.

5.1.2. MLE Multivariante (Full Bayesian Gaussian)

Test: Accuracy = 0.7143, F1-macro = 0.4563

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.

Análisis: El modelo multivariante con covarianza completa falla significativamente. Con d características (dimensiones), la matriz de covarianza Σ_y es una matriz $d \times d$ simétrica positiva definida, lo que implica que se deben estimar $\frac{d(d+1)}{2}$ parámetros únicos por clase (136 parámetros con $d = 16$), ya que los elementos fuera de la diagonal principal se duplican. Este número crece cuadráticamente con d, por lo que en espacios de alta dimensión se necesita un gran número de muestras N_y por clase para obtener estimaciones estables. Para clases con 3-6 ejemplos en entrenamiento, esto es imposible, resultando en matrices de covarianza singulares o mal condicionadas porque no hay suficientes datos para capturar las varianzas y covarianzas.

5.1.3. Histogram Bayes

Test: Accuracy = 0.3810, F1-macro = 0.0788

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.

Análisis: El peor rendimiento de todos los modelos, evidenciando la maldición de la dimensionalidad. Con d dimensiones y b bins por dimensión, se necesitan b^d bins. Con 16 características binarias, hay $2^{16} = 65536$ posibles combinaciones (bins). Con solo 101 muestras totales, la mayoría de bins están vacíos, resultando en un colapso total del modelo que predice solo la clase mayoritaria (mamíferos).

5.1.4. Parzen Windows

Test: Accuracy = 1.0, F1-macro = 1.0

Fundamento teórico: Estimador no paramétrico que usa un kernel para suavizar la estimación de densidad. En nuestra implementación, utilizamos un kernel Gaussiano (parámetro kernel='gaussian' en KernelDensity):

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

Donde K_h es el 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. Mediante validación cruzada en el conjunto de entrenamiento, probamos $h \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$ y seleccionamos el valor óptimo $h = 0.1$.

Análisis: Parzen windows con kernel Gaussiano logra clasificación perfecta con $h = 0.1$ óptimo. A diferencia de histogramas, Parzen produce estimaciones continuas que funcionan bien incluso con datos relativamente escasos. El suavizado del kernel Gaussiano compensa la falta de muestras, evitando las discontinuidades artificiales de los bins. En validación cruzada, obtiene el mejor F1-macro (0.8648), superior incluso a Naive Bayes (0.8505), confirmando que el bandwidth optimizado generaliza bien.

5.1.5. k-NN Density Bayes

Test: Accuracy = 0.4762, F1-macro = 0.5714

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.

Análisis: Rendimiento pobre en test con k=11 óptimo. Con k=11 en un dataset de 80 muestras de entrenamiento, los 11 vecinos más cercanos diluyen la señal local, resultando en estimación de densidad por volumen local inestable. Clases con distribuciones espaciales dispersas sufren especialmente. Comparado con Parzen, que fija el ancho de banda globalmente, la estrategia de volumen adaptativo local de k-NN Density resulta inferior en este caso.

5.1.6. k-NN Rule

Test: Accuracy = 1.0, F1-macro = 1.0

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

$$\hat{y} = \arg\max_{c} \sum_{i \in N_k(\mathbf{x})} \mathbb{1}(y_i = c)$$

Donde $N_k(\mathbf{x})$ son los índices de los k vecinos más cercanos a $\mathbf{x}$. A diferencia de k-NN Density, este método cuenta votos directamente sin estimar densidades.

Análisis: El método clásico de k-NN logra clasificación perfecta con k=1 (vecino más cercano). Esto indica que las clases están bien separadas en el espacio de características y que los ejemplos de test tienen vecinos cercanos de su misma clase en entrenamiento. Con datos bien separados y sin ruido significativo, el vecino más cercano es suficiente y óptimo. Valores mayores de k diluirían innecesariamente la señal. La simplicidad del método (votación directa vs estimación de densidad) resulta ventajosa.

6. Discusión y Conclusiones

6.1. Comparación general de enfoques

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

Modelos paramétricos:

- Naive Bayes: Perfecto (1.0/1.0). La asunción de independencia no penaliza en este dataset. Pocas suposiciones (solo medias y varianzas), robusto con pocas muestras.
- MLE Full: Falla con clases minoritarias (0.71/0.46). Más parámetros ≠ mejor modelo con datos limitados.

Modelos no paramétricos:

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

6.2. Impacto de clases minoritarias

El dataset Zoo tiene clases con solo 3-4 ejemplos en entrenamiento. Naive Bayes maneja bien clases minoritarias estimando solo 16 parámetros/clase, mientras que MLE Full necesita estimar 136 parámetros/clase (matriz 16×16) y falla. Parzen y k-NN Rule no estiman parámetros por clase, usan similitud directa y funcionan bien si ejemplos están separados.

6.3. Métricas: Accuracy vs F1-macro

En dataset multiclase con distribución desbalanceada (40% mamíferos vs 4% anfibios), el F1-macro es esencial. Histogram Bayes tiene 38% accuracy pero solo 7.9% F1-macro, revelando que predice solo mamíferos. Los tres modelos perfectos (Naive Bayes, Parzen, k-NN Rule) tienen Accuracy = F1-macro = 1.0, confirmando clasificación genuinamente balanceada.

6.4. Conclusiones finales

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: No maneja clases con 3-6 ejemplos (matrices singulares)
3. Histogram colapsa: Maldición dimensionalidad (2^{16} bins, 101 muestras)
4. k-NN Density mediocre: Estimación de volumen local inestable

Lecciones clave:

- La asunción de independencia de Naive Bayes no siempre penaliza
- Modelos complejos necesitan datos: MLE Full requiere $O(d^2)$ muestras/clase

- Histogramas sufren curse of dimensionality; Parzen lo mitiga
- k-NN simple supera a k-NN density: menos pasos, menos errores

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.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning education by demonstrating the practical application of fundamental classification algorithms on real-world datasets. There are no negative societal consequences anticipated from this educational work.

References