Naive Bayes Classifier

Profesor Cosijopii García

2025

¿Qué es Naive Bayes?

- Es un clasificador probabilístico basado en el Teorema de Bayes.
- Supone independencia entre las características.
- Muy eficiente para problemas de clasificación con datos categóricos o discretos.

Fórmula:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

Donde:

- $P(C|X)$: Probabilidad posterior de la clase C dada la evidencia X.
- $P(X|C)$: Verosimilitud de X dado C.
- $P(C)$: Probabilidad a priori de la clase C.
- $P(X)$: Probabilidad total de X.

Ejemplo práctico

Dataset:

Clima	Temperatura	Humedad	Jugar Tenis
Sunny	Hot	High	No
Sunny	Mild	High	No
Overcast	Cool	Normal	Yes
Rainy	Cool	Normal	Yes
Sunny	Cool	High	Yes
Overcast	Hot	Normal	Yes
Rainy	Mild	Normal	Yes

Cálculo de probabilidades

Probabilidades a priori:

- $P(\text{Yes}) = \frac{5}{7}$
- $P(\text{No}) = \frac{2}{7}$

Cálculo de probabilidades condicionales:

- $P(\text{Sunny}|\text{Yes}) = \frac{1}{5}$
- $P(\text{Cool}|\text{Yes}) = \frac{3}{5}$
- $P(\text{Normal}|\text{Yes}) = \frac{4}{5}$
- $P(\text{Sunny}|\text{No}) = \frac{2}{2}$
- $P(\text{Cool}|\text{No}) = \frac{0}{2}$
- $P(\text{Normal}|\text{No}) = \frac{0}{2}$

Clasificación final

Dado el ejemplo: $X = (\text{Sunny}, \text{Cool}, \text{Normal})$ **Cálculo de**

$P(\textbf{Yes}|X)$**:**

$$P(\text{Yes}|X) \propto P(\text{Sunny}|\text{Yes}) \cdot P(\text{Cool}|\text{Yes}) \cdot P(\text{Normal}|\text{Yes}) \cdot P(\text{Yes})$$

Sustituyendo:

$$P(\text{Yes}|X) \propto \frac{1}{5} \cdot \frac{3}{5} \cdot \frac{4}{5} \cdot \frac{5}{7} = \frac{12}{175} = 0.068571$$

Cálculo de $P(\textbf{No}|X)$**:**

$$P(\text{No}|X) \propto P(\text{Sunny}|\text{No}) \cdot P(\text{Cool}|\text{No}) \cdot P(\text{Normal}|\text{No}) \cdot P(\text{No})$$

Sustituyendo:

$$P(\text{No}|X) \propto \frac{2}{2} \cdot \frac{0}{2} \cdot \frac{0}{2} \cdot \frac{2}{7} = 0$$

Decisión final

Resultado:

- $P(\text{Yes}|X) > P(\text{No}|X)$
- Por lo tanto, el clasificador predice **Yes**: **Sí jugar al tenis**.

Ventajas de Naive Bayes:

- Rápido y eficiente.
- Funciona bien con datos categóricos.
- Maneja problemas multiclase fácilmente.

Desventajas:

- Supone independencia entre características.
- Puede tener problemas con datos continuos si no se discretizan adecuadamente.

Aprendizaje No Supervisado

- **¿Qué es?**
  - Descubrimiento de patrones en datos sin etiquetas predefinidas.
  - El algoritmo encuentra similitudes y agrupaciones naturales entre los datos.
- **Aplicaciones:**
  - Segmentación de imágenes
  - Recomendación de productos
  - Detección de anomalías
  - Clustering de clientes
  - Análisis de sentimientos
  - Clasificación de documentos

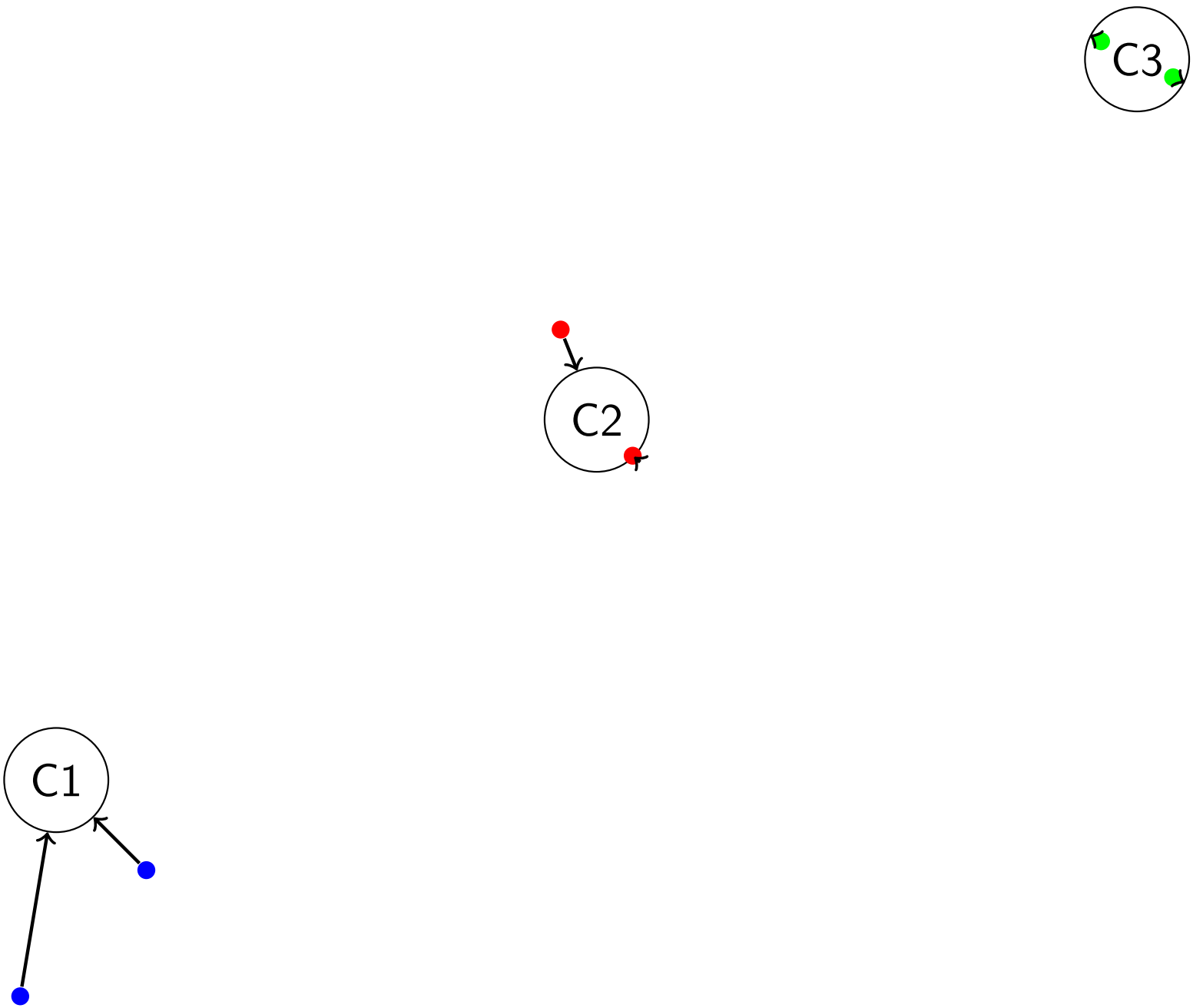
K-means: Agrupando Datos

- **Objetivo:**
  - Dividir un conjunto de datos en K grupos (clusters) con características similares.
  - Minimizar la distancia entre los puntos de datos y los centroides de cada cluster.
- **Proceso:**
  1. Inicializar K centroides aleatorios.
  2. Asignar cada punto de datos al cluster más cercano según la distancia al centroide.
  3. **Recalcular los centroides:**
     - Calcular el centroide de cada cluster tomando el promedio de las coordenadas de los puntos dentro del cluster.
     - Si tenemos un conjunto de puntos $(x_1, y_1), (x_2, y_2), \ldots, (x_{n_k}, y_{n_k})$ en un cluster C_k, el nuevo centroide es (versión para 2 dimensiones):

$$\text{Centroide}_k = \left(\frac{1}{n_k} \sum_{i=1}^{n_k} x_i, \frac{1}{n_k} \sum_{i=1}^{n_k} y_i \right)$$

     - Este proceso se repite para cada cluster C_k.
  4. Repetir los pasos 2 y 3 hasta que los centroides no cambien significativamente (convergencia).

Visualización de K-means



Ejercicio: Aplicando K-means

Supongamos que tenemos los siguientes puntos de datos bidimensionales:

$$P = \{(1,2), (2,3), (3,3), (6,5), (7,6), (8,7)\}$$

Queremos aplicar el algoritmo K-means con $K = 2$ clusters.

1. **Paso 1: Inicialización de los centroides.** - Inicializa aleatoriamente los centroides C_1 y C_2. Por ejemplo:

$$C_1 = (1,2), \quad C_2 = (7,6)$$

2. **Paso 2: Asignación de puntos a los clusters.** - Asigna cada punto al centroide más cercano. ¿Qué puntos asignarías a C_1 y cuáles a C_2?
3. **Paso 3: Recalcular los centroides.** - Calcula los nuevos centroides tomando el promedio de las coordenadas de los puntos asignados a cada cluster.
4. **Paso 4: Repetir.** - Repite los pasos de asignación de puntos y recalculación de centroides hasta que los centroides no cambien significativamente.

Ejercicio: Aplicando K-means

Supongamos que tenemos los siguientes puntos de datos bidimensionales:

$$P = \{(1,2), (2,3), (3,3), (6,5), (7,6), (8,7)\}$$

Queremos aplicar el algoritmo K-means con $K = 2$ clusters.

1. **Paso 1: Inicialización de los centroides.** - Inicializa aleatoriamente los centroides C_1 y C_2. Por ejemplo:

$$C_1 = (1,2), \quad C_2 = (7,6)$$

2. **Paso 2: Asignación de puntos a los clusters.** - Asigna cada punto al centroide más cercano. ¿Qué puntos asignarías a C_1 y cuáles a C_2?
3. **Paso 3: Recalcular los centroides.** - Calcula los nuevos centroides tomando el promedio de las coordenadas de los puntos asignados a cada cluster.
4. **Paso 4: Repetir.** - Repite los pasos de asignación de puntos y recalculación de centroides hasta que los centroides no cambien significativamente.

Evaluación de K-means

- **Métricas:**
  - **Suma de los cuadrados de las distancias dentro de los clusters (SSE):** Cuanto menor sea, mejor será la calidad de los clusters.
  - **Índice de silueta:** Mide qué tan bien asignado está un punto de datos a su cluster en comparación con otros clusters.
  - **Coeficiente de correlación de Pearson:** Puede ser utilizado para ver la relación entre las distancias de los puntos dentro de un mismo cluster.

Ventajas y Desventajas

- **Ventajas:**
  - Algoritmo simple y eficiente.
  - Fácil de implementar y entender.
  - Funciona bien cuando los clusters son esféricos y de tamaño similar.
  - Escalable a grandes volúmenes de datos.
- **Desventajas:**
  - Requiere definir el número de clusters (K) de antemano.
  - Sensible a la inicialización de los centroides. Diferentes inicializaciones pueden dar resultados distintos.
  - Asume que los clusters son convexos y de igual tamaño, lo cual no siempre es cierto.
  - No es ideal para datos con clusters de formas no lineales o distribuciones muy distintas.
  - Sensible a los valores atípicos (outliers).