

21 de Septiembre, 2025

Naive Bayes

Integrantes:

Giovanni Orozco

Juan Esteban Lou

Definición general (Juanes)

Naive Bayes es un algoritmo de clasificación supervisada basado en el Teorema de Bayes. Su idea central es calcular la probabilidad de que una instancia pertenezca a una clase determinada, suponiendo independencia entre las características.

Aunque esta suposición de independencia rara vez se cumple en la práctica, el modelo funciona sorprendentemente bien en muchos problemas reales.

La fórmula principal es:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

donde:

- $P(C|X)$: Probabilidad de que la clase sea C dado el conjunto de atributos X.
- $P(X|C)$: Probabilidad de observar los atributos X dado que estamos en la clase C.
- $P(C)$: Probabilidad previa de la clase C.
- $P(X)$: Probabilidad total de observar los atributos X.

Naive Bayes sirve para clasificar datos usando la probabilidad y el teorema de Bayes. Es capaz de predecir a qué clase pertenece algo y combina la probabilidad de que aparezca cierta característica con la probabilidad de que pertenezca a una clase. Se le llama “Naive” o “ingenuo” porque asume que todas las características son independientes entre sí, lo cual lo hace tan poderoso pero a la vez, limitado en algunas ocasiones.

Es muy común ver su aplicación en filtros de texto en correos (spam y no spam), clasificación de texto (reseñas, tema principal de un documento, entre otros), diagnósticos médicos (enfermo o sano) y reconocimiento de sentimientos en redes sociales. Con estas aplicaciones, el modelo puede ser entrenado y aunque asume que siempre las características de las clases son independientes, es poderoso, escalable y muy fácil de realizar.

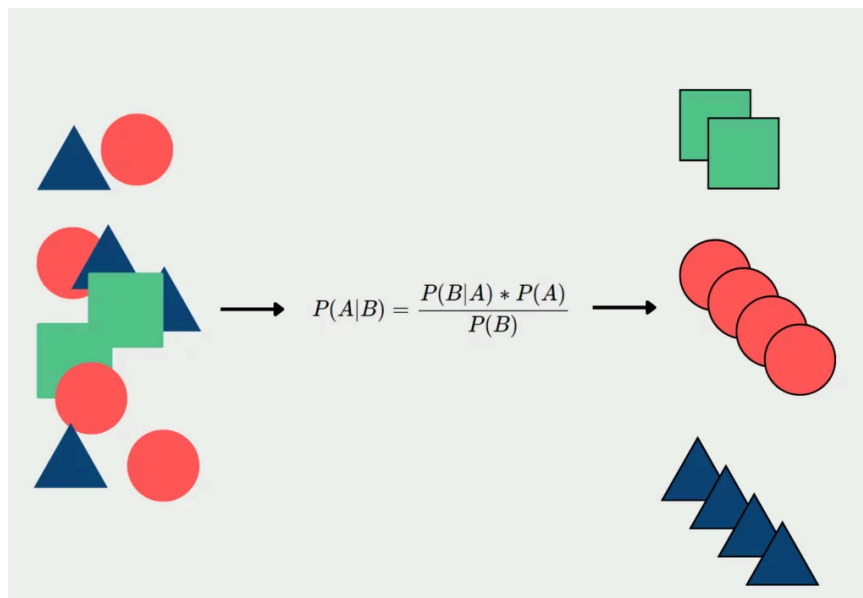
¿Cómo funciona Naive Bayes? (Giovanni)

Naive Bayes funciona como un clasificador que aprende a reconocer patrones en un conjunto de datos y los usa para predecir a qué categoría pertenece un nuevo caso. Primero es necesario entrenar el modelo; se le muestran ejemplos ya clasificados con anterioridad y a partir de ellos, Naive Bayes analiza la frecuencia que aparece cada característica en cada categoría.

Al recibir un nuevo dato, el modelo lo analiza característica por característica. Asume que todas son independientes entre sí y las analiza por separado. Por ejemplo, en correos electrónicos, el cuál es un uso frecuente de Naive Bayes, el modelo observa palabras como “descuento” o “gratis” y calcula en qué categoría es más común, spam o no spam.

Luego de combinar todas las evidencias, elige qué categoría es más probable que esté. El criterio MAP o Máximo A Posteriori compara todas las probabilidades posibles y se queda con el que mayor porcentaje tenga.

En resumen, Naive Bayes toma múltiples pistas, las pondera y clasifica según la opción más probable. A pesar de su sencillez, es sorprendentemente eficaz en problemas reales como filtrado de correo, diagnóstico médico y análisis de texto.



Un “extra” que tiene Naive Bayes es que no importa si al momento de clasificar hay más frecuencia de una clase que de la otra ya que al momento de usar las probabilidades, es mínima la diferencia y no afecta realmente el criterio MAP ni la probabilidad real que el modelo otorga. Sin embargo, es importante evaluar siempre el modelo ya que no debe ser demasiado impreciso ni 100% preciso, ya que eso significa que el modelo está demasiado sesgado y cuando un dato no pertenezca a la clase del modelo de entrenamiento, el modelo dejará de ser fiable.

Ejemplo conceptual (Giovanni)

El problema es predecir si un correo electrónico es spam o no. Los datos que nos ofrecen es que en los correos de spam, la palabra “gratis” es muy frecuente. Por otro lado, en los de no spam, la palabra “Tarjeta” lo es.

Si se recibe un nuevo correo diciendo “tarjeta gratis”, el modelo realizará teorema de Bayes según el dataset que se le envió. Suponiendo que la probabilidad de spam es de 50% y de no spam es 50% y las probabilidades condicionales son las siguientes:

Probabilidad de que salga “gratis” dado que es spam es alta, 0.8;

Probabilidad de que salga “tarjeta” dado que es spam es baja, 0.1;

Probabilidad de que salga “gratis” dado que no es spam es baja, 0.2;

Probabilidad de que salga “tarjeta” dado que no es spam es alta, 0.7;

Al aplicar el teorema de Bayes, la probabilidad de que sea spam es de 0.04, a diferencia que sea no spam es de 0.07, por lo que el correo calificaría el texto como no spam bajo esas condiciones y probabilidades.

Casos de uso (Juanes)

♦ Ejemplo 1: Clasificación de correos

- Palabras clave como “gratis”, “promoción”, “compra ahora” tienen alta probabilidad de aparecer en correos spam.
- El modelo analiza la frecuencia de estas palabras y calcula la probabilidad de que un nuevo correo sea spam.

♦ Ejemplo 2: Diagnóstico médico

- A partir de síntomas (fiebre, tos, dolor muscular), se calcula la probabilidad de que un paciente tenga gripe u otra enfermedad.
- Aunque los síntomas pueden estar relacionados, Naive Bayes los considera independientes y aun así clasifica bien.

♦ Ejemplo 3: Clasificación de sentimientos en reseñas

- Si un texto contiene palabras como “excelente”, “maravilloso” → más probable que sea positivo.
- Si contiene “horrible”, “malo” → más probable que sea negativo.

Ventajas y desventajas (Juanes)

Ventajas

- Simple y rápido: fácil de implementar y computacionalmente eficiente.
- Funciona bien con datos pequeños: no necesita grandes cantidades de entrenamiento.
- Escalable: se adapta bien a conjuntos de datos grandes y de muchas dimensiones.
- Buen desempeño en texto: muy usado en clasificación de correos, sentimientos, documentos, etc.

Desventajas

- Suposición de independencia: en la práctica, las variables suelen estar correlacionadas, lo que puede reducir la precisión.
- Siempre asume independencia total entre las probabilidades, lo cual no siempre es correcto; en esos casos es necesario usar otro modelo.
- No captura relaciones complejas entre atributos.
- Probabilidades cero: si una palabra/atributo nunca aparece en una clase, su probabilidad se vuelve 0.
- Menos preciso que modelos más complejos cuando los datos tienen muchas dependencias.

Conclusiones (Giovanni)

- Simple y efectivo

Aunque Naive Bayes es un modelo “ingenuo” porque asume independencia entre todas sus características, logra clasificaciones demasiado precisas, tanto para datasets pequeños como grandes.

- Generalización

Al entrenar con ejemplos previamente etiquetados, aprende patrones estadísticos y puede predecir correctamente casos nuevos. No necesita de reglas complicadas para lograrlo, ya que puede ser desde enfermedades hasta sentimientos de texto, como más ejemplos.

- Velocidad y escalabilidad

Naive Bayes es ligero y rápido; capaz de procesar miles de registros en segundos. Esta característica lo hace ideal para aplicaciones en tiempo real que buscan una respuesta rápida y mayormente asertiva.

- Importancia del preprocesamiento

Transformar datos a vectores o usar técnicas de Machine Learning permiten mantener proporciones que mejoran la fiabilidad y la consistencia del modelo.

- Potencial para decisiones automatizadas

El modelo permite automatizar predicciones críticas basadas en datos. Las decisiones son más rápidas y no tienen sesgos, los datos no mienten.

- Limitaciones

Al suponer que existe siempre independencia entre sus características, el fallo está muy presente. Sin embargo, incluso con la limitación, los resultados siguen siendo sólidos y muestran la potencia de la probabilidad como herramienta de predicción.

Referencias (Giovanni)

Berrar, D. (2019). Bayes' Theorem and Naive Bayes Classifier.

Lakshmipathi, N. (2020). Sentiment Analysis of IMDB Movie Reviews. Extraído de Kaggle.
<https://www.kaggle.com/code/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews>

Lowd, D. y Domingos, P. (2005). Naive Bayes Models for probability estimation.
<https://dl.acm.org/doi/abs/10.1145/1102351.1102418>

Murphy, K. (2006). Naive Bayes classifiers.
[https://datajobs.com/data-science-repo/Naive-Bayes-\[Kevin-Murphy\].pdf](https://datajobs.com/data-science-repo/Naive-Bayes-[Kevin-Murphy].pdf)