**Nombre del Algoritmo:** Statistical classification

**Descripción del Problema Computacional para la cual el algoritmo es una solución.**

**Delimitación del Problema:**

El problema computacional que aborda el algoritmo de clasificación estadística consiste en asignar una etiqueta o categoría a un conjunto de datos basado en sus características. Este problema ocurre en entornos donde existen datos categóricos o discretos que deben ser predichos a partir de patrones presentes en las entradas, como clasificación de correos electrónicos, reconocimiento de imágenes o diagnóstico médico.

**Comprensión del Problema:**

La clasificación estadística se basa en aprender un modelo a partir de datos etiquetados (entrenamiento) y utilizar dicho modelo para predecir etiquetas desconocidas (prueba). Esto involucra:

* Modelado: Encontrar una función 𝑓(𝑥) que relacione las entradas x con las etiquetas y.
* Generalización: Garantizar que 𝑓(𝑥) funcione bien no solo en los datos conocidos, sino en datos nuevos.

**Clasificación del Problema:**

El problema pertenece a la categoría de aprendizaje supervisado en el ámbito del aprendizaje automático, donde los algoritmos se entrenan con datos etiquetados para hacer predicciones sobre nuevos datos. Es un caso de problema de decisión.

**Viabilidad Computacional:**

La viabilidad depende de:

* La cantidad y calidad de los datos.
* La complejidad del modelo estadístico utilizado.
* Los recursos computacionales disponibles (capacidad de almacenamiento y velocidad de procesamiento).

**Contexto del Problema:**

Este tipo de problema se encuentra en diversas áreas como:

* Marketing (segmentación de clientes).
* Medicina (diagnóstico basado en síntomas).
* Seguridad informática (detección de intrusos).
* Industria (predicción de fallas)

**Análisis de Datos y Estructura del Problema:**

Los datos suelen tener la forma de un conjunto de características𝑋(vector de entrada) asociadas a etiquetas 𝑌(salida). La estructura depende de:

* Dimensionalidad: Número de características (variables independientes).
* Tamaño: Número de observaciones.
* Tipología: Datos categóricos, numéricos o mixtos.

**Evaluación de Complejidad:**

La complejidad computacional varía según el modelo de clasificación. Por ejemplo:

* Clasificación Bayesiana: 𝑂(𝑛⋅𝑘), donde 𝑛 es el tamaño del conjunto de datos y 𝑘 es el número de clases.

**Análisis del Contexto del Problema para la cual el algoritmo es una solución**.

**Contexto general del Problema:**

En términos prácticos, el problema de clasificación estadística aborda la necesidad de tomar decisiones basadas en datos observados. Ejemplo: predecir si un cliente comprará un producto basándose en su historial de compras.

**Identificación de los conjuntos y estructuras de datos:**

1. Datos de entrada (𝑋):
   * Variables independientes o características.
   * Ejemplo: Edad, ingreso, ubicación geográfica.
2. Datos de Salida (𝑌):
   * Etiquetas discretas.
   * Ejemplo: "Comprará" o "No comprará".

**Formalización del problema:**

Dado un conjunto de datos 𝐷= {(𝑥1, 𝑦1), (𝑥2, 𝑦2), ..., (𝑥𝑛, 𝑦𝑛)}, encontrar una función 𝑓: 𝑋→𝑌 que minimice el error de clasificación.

**Análisis de la Dinámica del Problema:**

El problema incluye dos fases:

* Entrenamiento: Crear un modelo estadístico basado en los datos etiquetados.
* Predicción: Aplicar el modelo entrenado para clasificar nuevos datos.

**Evaluación de la Complejidad del Contexto:**

La complejidad práctica está influenciada por:

* El tamaño de los datos (𝑛).
* El número de dimensiones (𝑝).
* La técnica estadística utilizada (regresión logística, naive Bayes, etc.).

**Identificación de los casos límites y excepciones:**

**Casos límite:**

* Datos con alta Dimensionalidad y pocos ejemplos (problema de sobreajuste).
* Datos desequilibrados (cuando una clase domina significativamente).

**Excepciones:**

* Datos ruidosos o incompletos.
* Relación no lineal entre las características y las etiquetas.

**Interacción con otros sistemas:**

El algoritmo de clasificación puede interactuar con:

* Sistemas de bases de datos para extraer datos de entrada.
* Herramientas de visualización para presentar resultados.
* Algoritmos de selección de características para optimizar entradas.

**Especificación Formal:**

**Definición de las entradas y salidas:**

* Entradas:
  + Conjunto de datos de entrenamiento: 𝐷= {(𝑥1, 𝑦1), ..., (𝑥𝑛, 𝑦𝑛)}.
  + Datos de prueba: 𝑋𝑡𝑒𝑠𝑡.
* Salidas:
  + Etiquetas predichas 𝑌^𝑡𝑒𝑠𝑡 para 𝑋𝑡𝑒𝑠𝑡.

**Definición formal del problema mediante funciones:**

* Función de entrenamiento:

𝑓train (D)→f(x)

* Función de Predicción:

𝑓predict (𝑓, X test) → Y^test

**Relaciones y Restricciones:**

* **Relaciones:** 
  + 𝑌 debe estar relacionado de manera significativa con 𝑋.
* **Restricciones:**
  + Datos numéricos deben ser normalizados.
  + Ausencia de valores nulos en 𝑋 y 𝑌.

**Invariantes y propiedades:**

* El modelo debe ser consistente: Predicciones similares para entradas similares.
* Robustez ante pequeñas variaciones en los datos.

**Control de Flujo:**

1. Preprocesamiento de datos (limpieza y normalización).
2. División del conjunto de datos en entrenamiento y prueba.
3. Entrenamiento del modelo.
4. Validación del modelo.
5. Predicción sobre nuevos datos.

**Pseudocodigo del Algoritmo:**

Algoritmo ClasificacionEstadistica

// Entrenamiento del modelo

Definir datos\_entrenamiento como Matriz[n][m] // Características de entrenamiento

Definir etiquetas\_entrenamiento como Vector[n] // Etiquetas de entrenamiento

Definir modelo como Diccionario // Almacena parámetros del modelo

Funcion EntrenarModelo(datos\_entrenamiento, etiquetas\_entrenamiento)

Definir probabilidad\_clase como Diccionario

Definir probabilidad\_condicional como Diccionario

// Calcular probabilidades de clase

Para cada clase en etiquetas\_entrenamiento

probabilidad\_clase[clase] <- Contar(etiquetas\_entrenamiento == clase) / tamaño(etiquetas\_entrenamiento)

FinPara

// Calcular probabilidades condicionales para cada característica

Para cada característica desde 1 hasta m

Para cada clase en probabilidad\_clase

probabilidad\_condicional[característica][clase] <- CalcularDistribucion(datos\_entrenamiento, etiquetas\_entrenamiento, característica, clase)

FinPara

FinPara

modelo.probabilidad\_clase <- probabilidad\_clase

modelo.probabilidad\_condicional <- probabilidad\_condicional

Retornar modelo

FinFuncion

// Clasificación de nuevos datos

Funcion Clasificar(datos, modelo)

Definir predicciones como Vector[tamaño(datos)]

Para cada instancia en datos

Definir mejor\_clase como ""

Definir mejor\_probabilidad como -Infinito

Para cada clase en modelo.probabilidad\_clase

probabilidad <- modelo.probabilidad\_clase[clase]

Para cada característica desde 1 hasta m

probabilidad <- probabilidad \* modelo.probabilidad\_condicional[característica][clase][instancia[característica]]

FinPara

Si probabilidad > mejor\_probabilidad Entonces

mejor\_probabilidad <- probabilidad

mejor\_clase <- clase

FinSi

FinPara

predicciones[instancia] <- mejor\_clase

FinPara

Retornar predicciones

FinFuncion

// Proceso principal

Escribir "Iniciando el modelo de clasificación estadística..."

modelo <- EntrenarModelo(datos\_entrenamiento, etiquetas\_entrenamiento)

Escribir "Modelo entrenado correctamente."

// Clasificar datos nuevos

Definir datos\_prueba como Matriz[k][m]

predicciones <- Clasificar(datos\_prueba, modelo)

Escribir "Predicciones realizadas: ", predicciones

FinAlgoritmo

**Casos de usos Posible:**

* Diagnóstico Médico:
  + Clasificar enfermedades basándose en los resultados de pruebas médicas. Ejemplo: Determinar si un paciente tiene diabetes basado en niveles de glucosa y otras métricas.
* Filtrado de Correos Electrónicos:
  + Identificar si un correo es spam o no spam utilizando características como frecuencia de palabras clave, remitente, y longitud del mensaje.
* Clasificación de Documentos:
  + Asignar categorías a documentos basándose en las frecuencias de las palabras presentes. Ejemplo: Noticias clasificadas como "deportes", "política" o "tecnología".
* Análisis de Clientes en Finanzas:
  + Determinar si un cliente tiene un perfil de "alto riesgo" o "bajo riesgo" para otorgar créditos, basado en datos demográficos y transacciones previas.
* Reconocimiento de Voz:
  + Clasificar comandos de voz (e.g., "abrir archivo", "cerrar programa") basándose en las características estadísticas del audio.
* Clasificación de Imágenes:
  + Determinar si una imagen pertenece a una clase específica (e.g., "gato", "perro", "paisaje") utilizando un modelo basado en histogramas o patrones de píxeles.