

CUADRO COMPARATIVO DE ALGORITMOS

Machine Learning
Análisis Técnico Comparativo
Jessica Melani Romero Lora

ÁRBOLES DE DECISIÓN

Tipo de aprendizaje	Supervisado (clasificación y regresión)
Paradigma	Basado en reglas jerárquicas
Complejidad temporal	$O(n \cdot m \cdot \log n)$
Fundamento matemático	Entropía, Índice Gini, Ganancia de información
Funcionamiento	División recursiva del espacio de características mediante selección de mejor atributo en cada nodo usando criterios de pureza
Aplicaciones principales	<ul style="list-style-type: none">• Análisis de riesgo crediticio• Diagnóstico médico• Detección de fraude• Segmentación de clientes
Mejor opción cuando	Necesitas interpretabilidad, datos con relaciones no lineales, mezcla de variables categóricas y numéricas
Supera a otros en	Interpretabilidad y visualización de reglas de decisión
Interpretabilidad	Muy alta
Velocidad entrenamiento	Rápida
Velocidad predicción	Muy rápida
Capacidad con Big Data	Media
Precisión general	Media
Datos necesarios	Medio (cientos a miles)
Resistencia overfitting	Baja (requiere poda)
Normalización de datos	No requerida
Ventajas principales	Fácil interpretación, no requiere preprocesamiento extenso, maneja datos no lineales, captura interacciones automáticamente
Desventajas principales	Propenso a overfitting sin poda, inestable ante pequeños cambios, sesgado con clases desbalanceadas

NAIVE BAYES

Tipo de aprendizaje	Supervisado (clasificación probabilística)
Paradigma	Probabilístico bayesiano
Complejidad temporal	$O(n \cdot m)$
Fundamento matemático	Teorema de Bayes, Probabilidad condicional
Funcionamiento	Cálculo de probabilidades condicionales y a priori de cada clase, asumiendo independencia condicional entre características
Aplicaciones principales	<ul style="list-style-type: none">• Filtrado de spam• Análisis de sentimientos• Clasificación de documentos• Categorización de noticias
Mejor opción cuando	Clasificación de texto y NLP, necesitas predicciones en tiempo real, dataset pequeño, alta dimensionalidad
Supera a otros en	Velocidad con textos y cuando hay pocas muestras de entrenamiento
Interpretabilidad	Alta
Velocidad entrenamiento	Muy rápida
Velocidad predicción	Muy rápida
Capacidad con Big Data	Alta
Precisión general	Media
Datos necesarios	Poco (decenas a cientos)
Resistencia overfitting	Alta
Normalización de datos	No requerida
Ventajas principales	Extremadamente rápido, funciona bien con pocos datos, excelente para clasificación de texto, escalable
Desventajas principales	Supuesto de independencia irreal, sensible a características irrelevantes, no captura interacciones

K-NEAREST NEIGHBORS (KNN)

Tipo de aprendizaje	Supervisado (clasificación y regresión)
Paradigma	Basado en instancias (instance-based)
Complejidad temporal	Entrenamiento: $O(1)$, Predicción: $O(n \cdot m)$
Fundamento matemático	Distancia euclidiana, Manhattan, Minkowski
Funcionamiento	Sin entrenamiento formal, búsqueda de K vecinos más cercanos mediante cálculo de distancias y votación mayoritaria para clasificar
Aplicaciones principales	<ul style="list-style-type: none">• Sistemas de recomendación• Reconocimiento de patrones• Imputación de valores faltantes• Detección de anomalías
Mejor opción cuando	Datasets pequeños a medianos, pocos atributos (baja dimensionalidad), límites de decisión irregulares
Supera a otros en	Simplicidad de implementación y adaptación a nuevos datos sin reentrenamiento
Interpretabilidad	Media
Velocidad entrenamiento	Instantánea
Velocidad predicción	Lenta
Capacidad con Big Data	Muy baja
Precisión general	Media
Datos necesarios	Poco a Medio (depende de dimensionalidad)
Resistencia overfitting	Media
Normalización de datos	Obligatoria
Ventajas principales	Muy simple de implementar, sin fase de entrenamiento, adaptable a nuevos datos, efectivo con datos localizados
Desventajas principales	Lento en predicción con big data, maldición de dimensionalidad, sensible a datos irrelevantes y ruido

SUPPORT VECTOR MACHINES (SVM)

Tipo de aprendizaje	Supervisado (clasificación y regresión)
Paradigma	Basado en margen de separación
Complejidad temporal	$O(n^2 \cdot m)$ a $O(n^3 \cdot m)$
Fundamento matemático	Optimización convexa, Multiplicadores de Lagrange, Funciones kernel
Funcionamiento	Búsqueda del hiperplano óptimo que maximiza el margen entre clases, usando vectores de soporte y kernels para problemas no lineales
Aplicaciones principales	<ul style="list-style-type: none">• Reconocimiento facial• Clasificación de imágenes• Escritura manuscrita• Bioinformática
Mejor opción cuando	Clasificación binaria de alta precisión, alta dimensionalidad, margen de separación claro, datasets medianos
Supera a otros en	Clasificación binaria con alta dimensionalidad y datos con margen claro
Interpretabilidad	Baja
Velocidad entrenamiento	Lenta
Velocidad predicción	Rápida
Capacidad con Big Data	Baja
Precisión general	Alta
Datos necesarios	Medio (cientos a miles)
Resistencia overfitting	Alta
Normalización de datos	Obligatoria
Ventajas principales	Efectivo en alta dimensión, robusto ante overfitting, versátil con kernels, memoria eficiente
Desventajas principales	Costoso computacionalmente, difícil de interpretar, selección de kernel y parámetros compleja

REDES NEURONALES

Tipo de aprendizaje	Supervisado, No supervisado, Reforzamiento
Paradigma	Basado en conexiones neuronales
Complejidad temporal	Variable (depende de arquitectura y profundidad)
Fundamento matemático	Gradiente descendente, Backpropagation, Cálculo diferencial
Funcionamiento	Capas de neuronas interconectadas procesan información mediante funciones de activación, ajustando pesos con backpropagation
Aplicaciones principales	<ul style="list-style-type: none">• Visión por computadora• Procesamiento de lenguaje natural• Reconocimiento de voz• Traducción automática
Mejor opción cuando	Problemas muy complejos y no lineales, grandes volúmenes de datos, datos no estructurados, patrones abstractos
Supera a otros en	Problemas complejos con grandes volúmenes de datos no estructurados (imágenes, audio, texto)
Interpretabilidad	Muy baja
Velocidad entrenamiento	Muy lenta
Velocidad predicción	Rápida
Capacidad con Big Data	Muy alta
Precisión general	Muy alta
Datos necesarios	Mucho (miles a millones)
Resistencia overfitting	Baja (requiere regularización)
Normalización de datos	Recomendada
Ventajas principales	Mejor rendimiento en problemas complejos, aprende representaciones automáticamente, maneja múltiples tipos de datos

Desventajas principales

Requiere grandes cantidades de datos, caja negra (no interpretable), costoso en tiempo y recursos, muchos hiperparámetros