Análisis de viabilidad crediticia mediante KD-Tree y KNN

Un enfoque de Aprendizaje Automático para evaluar el riesgo de pago de nuevos solicitantes

Abel E. Borit Guitton, Luis A. Borit Guitton, Betzy J. Yarín Ramírez

Universidad Nacional de San Agustín
Maestría en Ciencias de la Computación
Algoritmos y Estructuras de Datos

Agosto, 2023

Índice

- Introducción
- Problema
- Objetivos
- Implementación del KD-Tree y KNN
- Métricas
- Conclusiones
- Referencias

Introducción

La relevancia de KNN y KD-Tree en diversos contextos:

- Identificación de patrones:
  - Detectan similitudes y estructuras en datos complejos.
- Eficiencia en búsquedas:
  - Optimizan la búsqueda de vecinos cercanos en conjuntos de datos extensos.
- Análisis exploratorio:
  - Facilitan la exploración y comprensión de conjuntos de datos multidimensionales.

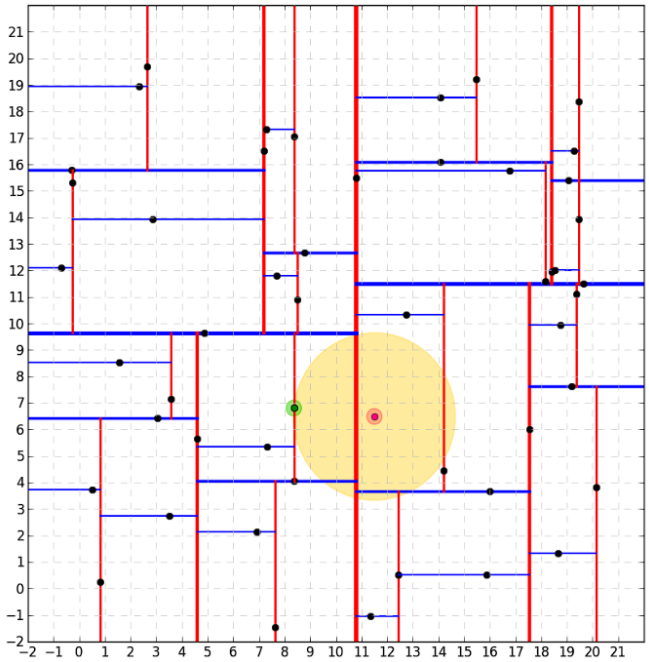


Figure: 1. Kd-Tree y KNN

Problema

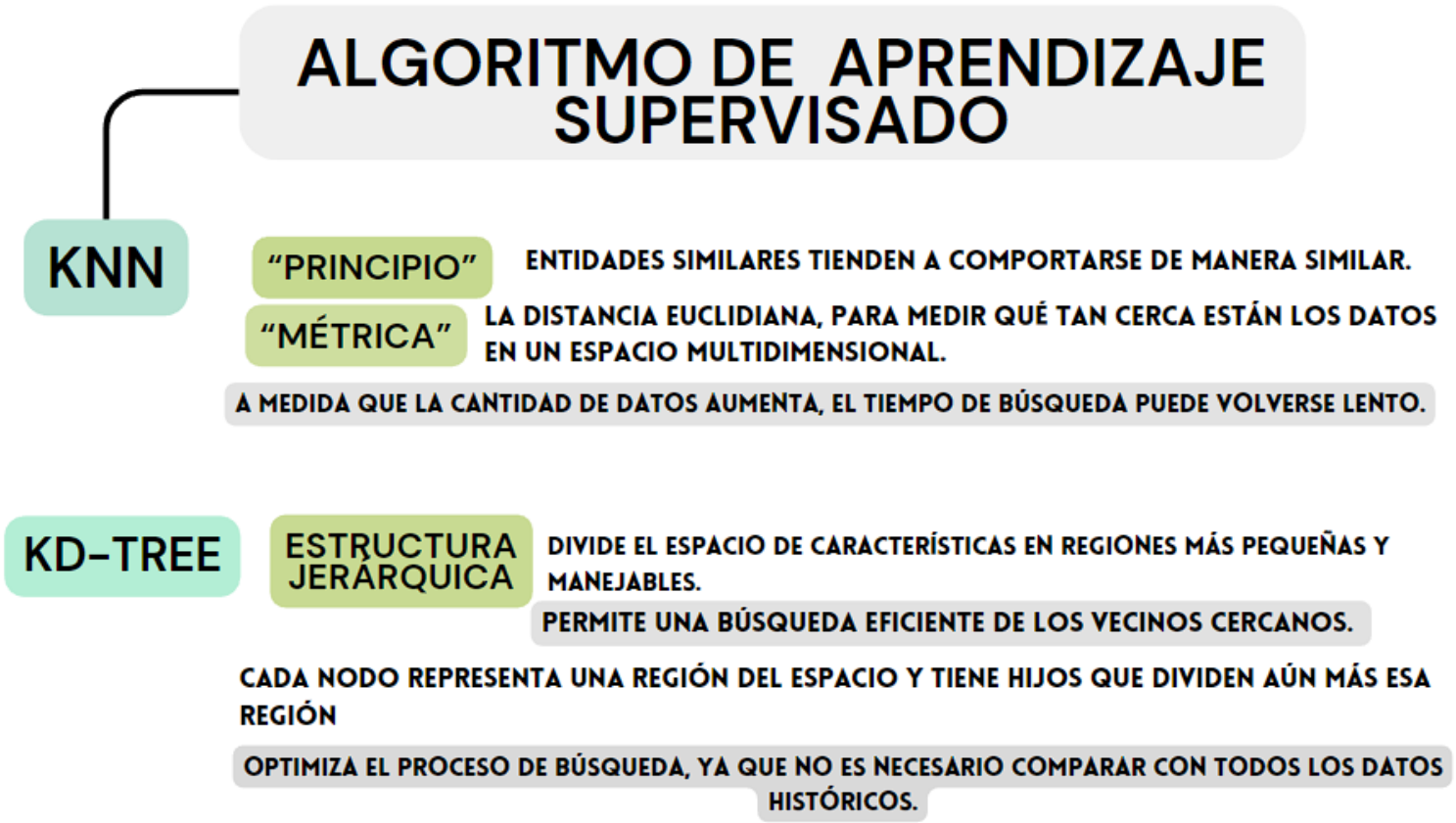
Clasificación de solicitantes de crédito utilizando KD-Tree y K-Nearest Neighbors (KNN)

- La implementación utilizará los datos de edad y monto de crédito de clientes que pagaron y no pagaron préstamos en el pasado. Mediante el algoritmo K-Nearest Neighbors (KNN) y una estructura KD-Tree, clasificará nuevos solicitantes en estas categorías. Se requiere tomar decisiones informadas sobre aprobaciones de crédito basadas en similitudes con el historial de pagos.

Objetivos

- Implementar KNN y KD-Tree: Optimizar la clasificación de nuevos solicitantes de crédito basados en características de edad y monto de crédito.
- Evaluar la viabilidad crediticia basada en similitudes con historiales de pagos anteriores.
- Mostrar los resultados de la clasificación en un gráfico que destaque la posición de los solicitantes en función de sus características de edad y monto de crédito.

KD-Tree y KNN[1]



[1] George T. Heineman, Gary Pollice, and Stanley Selkow. *Algorithms in a Nutshell*. 2009.

Implementación KD-Tree[2]

Algorithm 2 KD-Tree

function CONSTRUIRKDTREE($puntos, profundidad$)
  if longitud($puntos$) $= 0$ **then**
    return $Nulo$
  end if
  $eje \leftarrow profundidad$ mód longitud($puntos[0]$)
  Ordenar $puntos$ por el valor en el eje-ésimo eje
  $mediana \leftarrow$ longitud($puntos$) dividido 2
  $nodo \leftarrow$ nuevo KDNODE($puntos[mediana], eje$)
  $nodo$.izquierda $\leftarrow$ CONSTRUIRKDTREE($puntos[0 : mediana], profundidad + 1$)
  $nodo$.derecha $\leftarrow$ CONSTRUIRKDTREE($puntos[mediana + 1 : \text{longitud}(puntos)], profundidad + 1$)
  return $nodo$
end function

Figure: 2. Pseudocódigo KD-Tree

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006.

Implementación KNN[3]

Algorithm 3 K-Nearest Neighbors

```
function BuscarKNN(nodo, punto, k)
    function BúsquedaRecursiva(nodo, punto, k, cercanos)
        if nodo = Nulo then
            return
        end if
        distancia ← norma(punto − nodo.punto)
        if longitud(cercanos) < k then
            Agregar (distancia, nodo.punto) a cercanos
            Ordenar cercanos por la distancia
        else if distancia < cercanos[−1][0] then
            Eliminar el último elemento de cercanos
            Agregar (distancia, nodo.punto) a cercanos
            Ordenar cercanos por la distancia
        end if
        distancia_eje ← punto[nodo.eje] − nodo.punto[nodo.eje]
        nodo_cerca, nodo_lejano ← elegir entre (nodo.izquierda, nodo.derecha) y
(nodo.derecha, nodo.izquierda) según distancia_eje ≤ 0
        BúsquedaRecursiva(nodo_cerca, punto, k, cercanos)
        if |distancia_eje| < cercanos[−1][0] then
            BúsquedaRecursiva(nodo_lejano, punto, k, cercanos)
        end if
    end function
    puntos_cercanos ← []
    BúsquedaRecursiva(nodo, punto, k, puntos_cercanos)
    return [punto para distancia, punto en puntos_cercanos]
end function
```

Figure: 3. Pseudocódigo KNN

Métricas

- Cada una se calcula utilizando diferentes fórmulas. F1-Score depende de las primeras métricas y busca encontrar un equilibrio entre ellas.
- El resultado de las métricas depende de la evaluación del modelo y de los datos utilizados para el entrenamiento y prueba.
- El tener valores iguales en las métricas es un indicio que el modelo está obteniedo resultados consistentes, también se le atribuye a la sencillez del modelo.

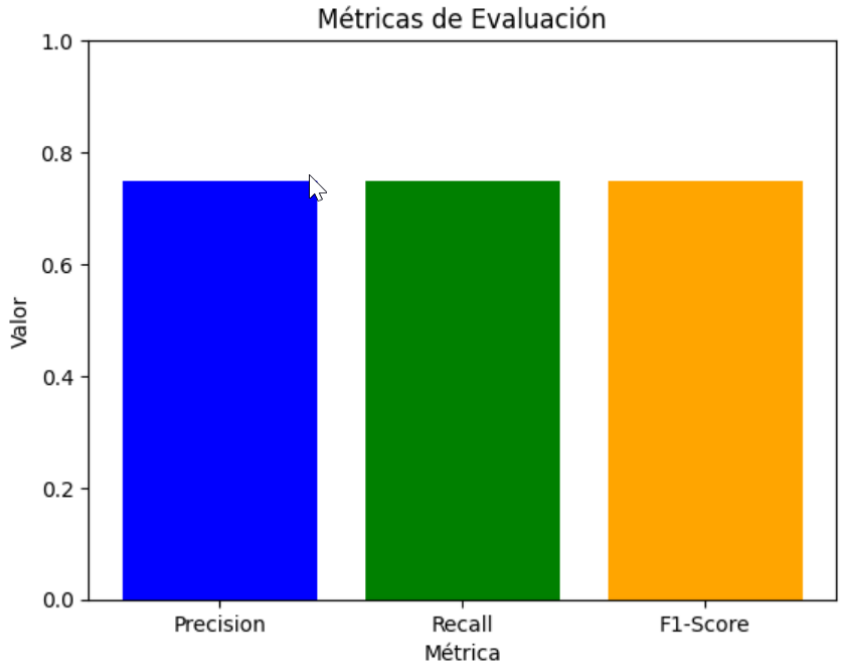


Figure: 4.Desempeño del Modelo

Conclusiones

- La implementación de KNN y KD-Tree para evaluar el riesgo crediticio demuestra ser eficaz para clasificar nuevos solicitantes en base a características como edad y monto de crédito.
- La estructura KD-Tree mejora la eficiencia en la búsqueda de vecinos cercanos, acelerando el proceso de clasificación.
- Los resultados muestran cómo los solicitantes son clasificados en función de la similitud con clientes anteriores, destacando el impacto de los vecinos cercanos en la decisión.
- El valor de k (número de vecinos más cercanos) se establece en 3 en la implementación. Elegir un valor óptimo para k es una decisión importante y puede afectar la precisión del modelo.

Referencias

- Alpaydin, Ethem. *Introduction to Machine Learning*. 2014.
- Bishop, Christopher M. *Pattern Recognition and Machine Learning*. 2006.
- Heineman, George T., Gary Pollice, and Stanley Selkow. *Algorithms in a Nutshell*. 2009.