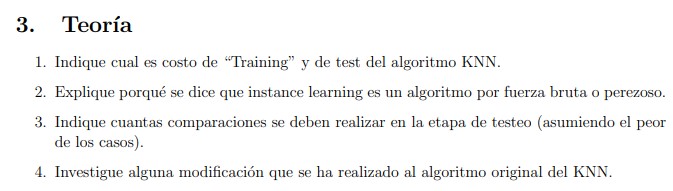
Rafael Vargas Solis 2024004833

Juan Rodríguez Montero 2020426163



1. El algoritmo KNN es un lazy learning. lo que significa que, durante la fase de entrenamiento, no se construye un modelo matemático complejo; en este caso simplemente se almacenan los datos de entrenamiento. El costo de esta fase se limita a guardar los datos, lo que tiene una complejidad de O(n·d), donde *n* representa el número de ejemplos en el conjunto de entrenamiento y *d* el número de sus características.

En la fase de prueba , el algoritmo KNN calcula la distancia entre el punto de prueba y cada uno de los puntos del conjunto de entrenamiento, además debe seleccionar los *k* vecinos más cercanos según la distancia calculada.y asignar la etiqueta del punto de prueba en función de la mayoría de clases de estos vecinos. Calcular las distancias requiere O(n·d) por cada punto de prueba y seleccionar los *k* vecinos más cercanos tiene un costo de O(n),

2. El Instance Learning es un enfoque en el que el algoritmo no construye un modelo explícito durante el entrenamiento, sino que almacena todos los ejemplos del conjunto de datos para usarlos directamente en la fase de predicción. Se habla de perezoso porque pospone todo el trabajo de generalización hasta el momento de la predicción.(solo almacena). y de fuerza bruta por su test , ya que el algoritmo compara el nuevo ejemplo con todos los datos almacenados, esto implica calcular la distancia entre el punto de prueba y cada punto del conjunto de entrenamiento, sin usar un modelo simplificado ni resumido.

3. En KNN o cualquier instance learning, durante la fase de test:

* + *n* = número de ejemplos en el conjunto de entrenamiento
  + *m* = número de ejemplos en el conjunto de prueba (test)

En el peor de los casos el algoritmo debe calcular la distancia con todos los puntos del entrenamiento para cada punto de prueba el número de comparaciones debe de ser n × m

4. KNN híbrido o con selección de vecinos adaptativa

En esta variante del KNN, el número de vecinos k no es fijo, sino que se ajusta dinámicamente según la densidad local de los datos. A diferencia del KNN tradicional, que siempre utiliza un mismo k para todos los puntos de prueba, el KNN adaptativo incrementa k en regiones donde los datos están densamente agrupados y lo disminuye en regiones donde los datos están más dispersos.

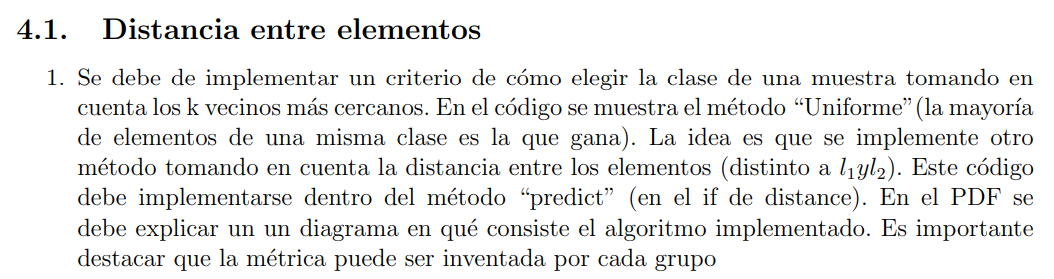
En KNN tradicional, n determina el número total de vecinos disponibles para cada predicción.

En KNN adaptativo, no se usa k fijo para todos los puntos sino:

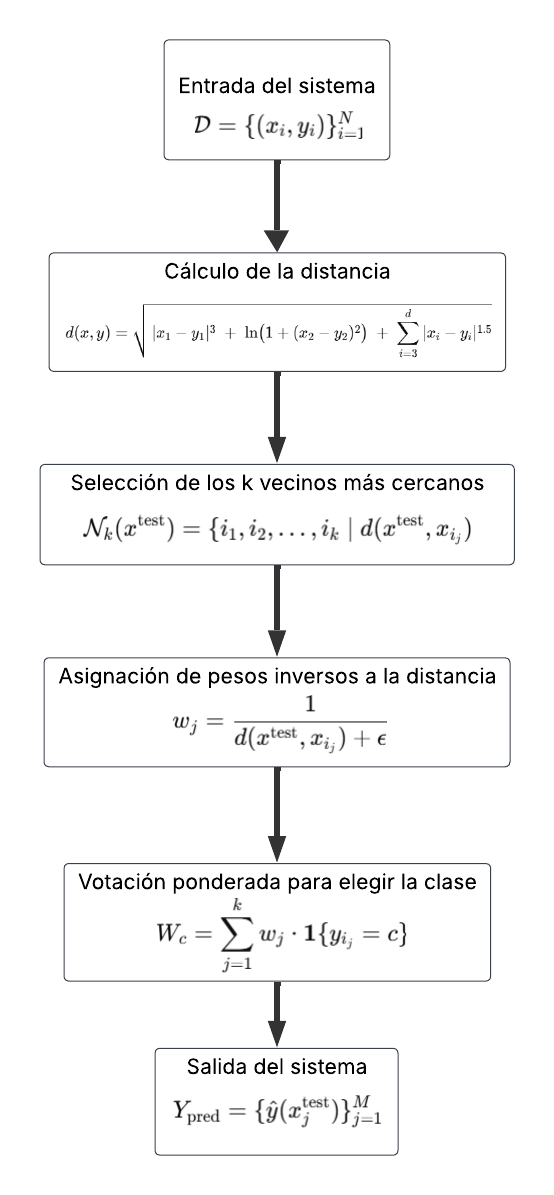
* En zonas densas: se puede aumentar k porque hay muchos vecinos cercanos.
* En zonas dispersas: se puede disminuir k porque hay pocos vecinos cercanos y no queremos incluir puntos lejanos que podrían inducir error.

Esta adaptación mejora la precisión de clasificación, ya que evita que puntos alejados influyan indebidamente en zonas dispersas y permite un análisis más robusto en zonas densas del espacio de características.Con un n más grande, la adaptabilidad de k se vuelve más efectiva, ya que hay más vecinos para ajustar dinámicamente.

Parte Práctica



El algoritmo K-Nearest Neighbors (KNN) implementado consiste en un clasificador supervisado que asigna la clase a una muestra de prueba en función de sus k vecinos más cercanos dentro del conjunto de entrenamiento, considerando una métrica inventada de distancia.



El flujo de procesamiento es el siguiente:

1. Entrada del sistema:
   1. Un conjunto de entrenamiento , donde:
      1. representa un vector de características de dimensión
      2. es la clase a la que pertenece el vector
   2. Un conjunto de prueba
   3. Un parámetro , que indica el número de vecinos más cercanos a considerar
2. Calculo de la distancia inventada (nueva)

Para cada muestra de prueba , se calcula su distancia con respecto a cada muestra de entrenamient .

La métrica inventada se definió como:

Cada termino cumple un papel distinto en cómo se mide la "distancia".

1. Primer atributo: cúbico

Efecto: Este término hace que el primer atributo domine si hay diferencias grandes, ya que su contribución crece más rápido que en una distancia euclidiana.

1. Segundo atributo: logarítmico

Efecto: No importa cuánto crezca la diferencia, el logaritmo asegura que su impacto sea controlado. Esto protege al algoritmo de "outliers" en el segundo atributo.

1. Tercer atributo: potencia 1.5

Efecto: Aporta sensibilidad intermedia en los atributos restantes. No permite que diferencias pequeñas pasen desapercibidas, pero tampoco sobrecarga el cálculo con diferencias enormes.

1. Suma de términos

Cada término aporta algo diferente:

* → Sensibilidad fuerte a diferencias grandes en el primer atributo.
* → Control de diferencias en el segundo atributo.
* → Balance en los demás atributos.

1. Efecto de la raíz cuadrada

Efecto: Gracias a la raíz, aunque un término como d1d\_1d1​ pueda ser muy grande, el resultado final no crece de forma descontrolada, manteniendo la métrica usable en clasificación.

1. Selección de los k vecinos más cercanos

Se ordenan las distancias y se seleccionan los k vectores de entrenamiento con menor valor:

es de los más pequeños

1. Asignacion de pesos inversos a la distancia

Para evitar que todos los vecinos tengan la misma influencia, se asigna a cada vecino un **peso proporcional al inverso de su distancia**:

donde ϵ es un valor muy pequeño que evita divisiones por cero.

Esto implica que los vecinos más cercanos tienen mayor influencia en la clasificación que los más alejados.

1. Votación ponderada para elegir la clase

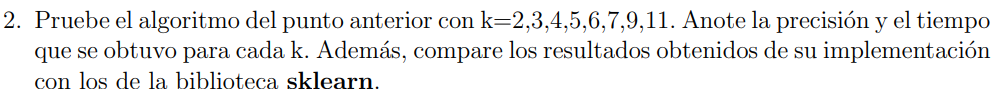
Se acumula el peso de cada vecino en función de su clase. Para cada clase :

donde 1{⋅} es la función indicadora (vale 1 si el vecino pertenece a la clase ccc, y 0 en caso contrario).

Finalmente, la clase predicha para es aquella con mayor peso acumulado:

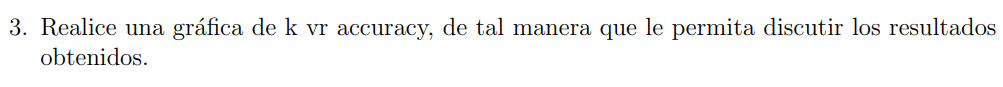
1. Salida del sistema

Se genera un vector con las clases predichas para todas las muestras de prueba:



Calendario

El contenido generado por IA puede ser incorrecto.



Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.