IMPLEMENTACIÓN DEL ALGORITMO DE CLASIFICACIÓN K-NN

Actividad 1

Maestro: Pedro Tecuanhuehue - vera
Alumno: Luis Francisco Matlalcuatzi Gonzalez
Fecha: 8 de febrero de 2024

Benemerita Universidad Autonoma de Puebla
Facultad en Ciencias de la Computacion
Mineria de datos

Índice

1. Descripcion del Algoritmo

El algoritmo de *K vecinos más cercanos* (KNN) es como un "vecindario" donde cada punto de datos tiene "vecinos" cercanos. Este algoritmo clasifica o predice a qué grupo pertenece un nuevo punto de datos basándose en qué grupo son sus vecinos más cercanos.
Imagina que tienes una serie de puntos en un mapa, algunos son rojos y otros azules. Quieres saber a qué grupo pertenece un nuevo punto. KNN mira a los K puntos más cercanos al nuevo punto y ve qué color tienen. Luego, el nuevo punto "vota" por el color que más ve entre sus vecinos cercanos, y así se decide a qué grupo pertenece.
El número de vecinos (K) es importante porque determina qué tan precisa será la predicción. Si eliges un número grande de vecinos, el modelo puede ser más preciso, pero también más costoso computacionalmente. Si eliges un número pequeño de vecinos, el modelo puede ser más rápido pero menos preciso.

2. Metodologia

1. **Conversión de Datos Nominales a Numéricos**:
   - Utilizaremos la base de datos GOLF y realizaremos la conversión de los datos nominales a datos numéricos.
2. **División de la Base de Datos**:
   - Dividiremos la base de datos en dos subconjuntos: conjunto de entrenamiento y conjunto de prueba.
   - El conjunto de entrenamiento tendrá un tamaño de 10 elementos, mientras que el conjunto de prueba tendrá un tamaño de 4 elementos.
   - La selección de elementos se realizará de forma aleatoria.
3. **Clasificación para $K = 3$, $K = 5$ y $K = 7$**:
   - Utilizaremos el algoritmo KNN con diferentes valores de K (3, 5 y 7) para clasificar uno de los 4 elementos del conjunto de prueba.
   - Realizaremos el mismo procedimiento para cada valor de K.
4. **Cálculo del Porcentaje de Eficiencia**:
   - Después de clasificar los elementos del conjunto de prueba para cada valor de K, calcularemos el porcentaje de eficiencia de cada clasificación.
5. **Herramientas Utilizadas**:
   - Para llevar a cabo este estudio, utilizaremos los siguientes módulos:
     - *NumPy*: para operaciones numéricas.
     - *Pandas*: para manipulación y análisis de datos.
     - *heapq*: para realizar operaciones de clasificación.
   - Además, convertiremos la base de datos a un archivo CSV para facilitar su manipulación y análisis.
6. **Etapas**:

- **Carga y Preprocesamiento de Datos**:
  - Se utiliza la biblioteca Pandas para cargar los datos desde un archivo CSV denominado `golf.csv`.
  - La columna `Ünnamed: 7"`, si está presente, se elimina para evitar posibles conflictos en el análisis.
- **Manipulación de Datos con NumPy**:
  - Los datos cargados se convierten a un formato NumPy array para facilitar su manipulación y cálculos numéricos posteriores.
  - Se mezclan aleatoriamente los datos para evitar sesgos en el análisis y garantizar la aleatoriedad en la selección de datos de entrenamiento y prueba.
- **División de Datos**:
  - Se dividen los datos en dos conjuntos: un conjunto de entrenamiento y un conjunto de prueba.
  - El conjunto de entrenamiento contiene 10 instancias, mientras que el conjunto de prueba contiene 4 instancias.
  - Se elimina la columna de etiquetas de clase del conjunto de entrenamiento y prueba, ya que solo se necesita para la clasificación.
- **Aplicación del Algoritmo KNN**:
  - Se solicita al usuario el valor de k que determinará la cantidad de vecinos más cercanos a considerar en el proceso de clasificación.
  - Para cada instancia en el conjunto de prueba, se calculan las distancias euclidianas respecto a todas las instancias del conjunto de entrenamiento.
  - Se identifican los k vecinos más cercanos a partir de las distancias calculadas utilizando la estructura de datos `heapq`.
  - Se determina la clase de la instancia de prueba mediante un voto mayoritario entre las clases de los vecinos más cercanos.
- **Resultados y Conclusiones**:
  - Se muestra el resultado final de la clasificación para cada instancia del conjunto de prueba, indicando si se clasifica como `"Yes"` o `"No"` según el voto mayoritario.

- El proceso se repite para diferentes valores de k para evaluar la sensibilidad del modelo a este hiperparámetro.

BASE DE DATOS GOLF:

- La base de datos GOLF es una base de datos de ejemplo.

3. Entrenamiento del Modelo

En esta sección, se detalla el proceso de entrenamiento del modelo utilizando el algoritmo de los K vecinos más cercanos (KNN). Se muestran y comparan los resultados obtenidos con diferentes valores de k, y se sugiere cuál es el mejor valor de k para este problema en función de los resultados.

3.1. Tabla de Datos

La tabla de datos utilizada para el entrenamiento del modelo se muestra a continuación:

Cuadro 1: Datos

sunny	overcast	rainy	temperature	humidity	windy	play
1	0	0	85	85	1	no
1	0	0	80	90	0	no
0	1	0	83	86	1	yes
0	0	1	70	96	1	yes
0	0	1	68	80	1	yes
0	0	1	65	70	0	no
0	1	0	64	65	0	yes
1	0	0	72	95	1	no
1	0	0	69	70	1	yes
0	0	1	75	80	1	yes
1	0	0	75	70	0	yes
0	1	0	72	90	0	yes
0	1	0	81	75	1	yes
0	0	1	71	91	0	no

3.2. Datos aleatorios

En la tabla 2 se muestran los datos en un orden aleatorio pero solo moviendo las filas de la tabla 1.

Cuadro 2: Datos aleatorios

sunny	overcast	rainy	temperature	humidity	windy	play
0	1	0	72	90	0	'yes'
1	0	0	69	70	1	'yes'
0	0	1	65	70	0	'no'
0	1	0	64	65	0	'yes'
0	1	0	83	86	1	'yes'
0	0	1	75	80	1	'yes'
0	1	0	81	75	1	'yes'
0	0	1	70	96	1	'yes'
1	0	0	75	70	0	'yes'
1	0	0	85	85	1	'no'
1	0	0	80	90	0	'no'
1	0	0	72	95	1	'no'
0	0	1	68	80	1	'yes'
0	0	1	71	91	0	'no'

3.3. Datos de Entrenamiento

En la tabla 3 se muestran los datos utilizados para el entrenamiento del modelo.

Cuadro 3: Datos de entrenamiento

sunny	overcast	rainy	temperature	humidity	windy
0	1	0	72	90	0
1	0	0	69	70	1
0	0	1	65	70	0
0	1	0	64	65	0
0	1	0	83	86	1
0	0	1	75	80	1
0	1	0	81	75	1
0	0	1	70	96	1
1	0	0	75	70	0
1	0	0	85	85	1

4. Clasificación de Datos Desconocidos

En esta sección se presentan los resultados obtenidos al clasificar datos desconocidos utilizando el modelo entrenado con el algoritmo de los K vecinos más cercanos (KNN). Se muestran los datos de prueba y las predicciones realizadas por el modelo, así como cualquier métrica de evaluación relevante.

4.1. Datos de Prueba

Cuadro 4: Datos de prueba

sunny	overcast	rainy	temperature	humidity	windy
0	1	0	72	90	0
0	1	0	83	86	1
0	0	1	68	80	1
0	0	1	71	91	0

4.2. 3 menores valores

Cuadro 5: 3 menores valores

Valor
7.000
10.149
10.488

Cuadro 6: 3 menores valores y resultados correspondientes

Dato 1	**Dato 2**	**Dato 3**	**Dato 4**	**Dato 5**	**Dato 6**	**Resultado**
0	0	1	75	80	1	'yes'
1	0	0	69	70	1	'yes'
0	0	1	65	70	0	'no'

4.3. 5 menores valores

Cuadro 7: 5 menores valores

Valor
7.000
10.149
10.488
12.329
14.000

Cuadro 8: Resultados

Resultado
$[0, 0, 1, 75, 80, 1,' yes']$
$[1, 0, 0, 69, 70, 1,' yes']$
$[0, 0, 1, 65, 70, 0,' no']$
$[1, 0, 0, 75, 70, 0,' yes']$
$[0, 1, 0, 81, 75, 1,' yes']$

4.4. 7 menores valores

Cuadro 9: 7 menores valores

Valor
7.000
10.149
10.488
12.329
14.000
15.588
15.620

Cuadro 10: Resultados

Resultado
$[0, 0, 1, 75, 80, 1, 'yes']$
$[1, 0, 0, 69, 70, 1, 'yes']$
$[0, 0, 1, 65, 70, 0, 'no']$
$[1, 0, 0, 75, 70, 0, 'yes']$
$[0, 1, 0, 81, 75, 1, 'yes']$
$[1, 0, 0, 72, 95, 1, 'no']$
$[0, 1, 0, 64, 65, 0, 'yes']$

5. Apendice

5.1. Codigo Fuente

El código fuente de este proyecto se encuentra disponible en el siguiente repositorio de GitHub: https://github.com/t4dokiary/knn

5.2. Distancia Euclidiana

La distancia euclidiana es un número positivo que indica la separación que tienen dos puntos en un espacio donde se cumplen los axiomas y teoremas de la geometría de Euclides. La distancia entre dos puntos A y B de un espacio euclidiano es la longitud del vector AB perteneciente a la única recta que pasa por dichos puntos. Se define la distancia euclidiana d(A,B) entre los puntos A y B, ubicados sobre una recta, como la raíz cuadrada del cuadrado de las diferencias de sus coordenadas X, esta definición garantiza que: la distancia entre dos puntos sea siempre una cantidad positiva. Y que la distancia entre A y B sea igual a la distancia entre B y A.