UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍA
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Seminario de Solución de Problemas de Sistemas Basados en Conocimiento

Práctica No. 9

K-Means vs K-NN

Nombre: Hurtado González Edgar Arturo
Código: 212597894

Introducción

El modelo k-means es un algoritmo de agrupamiento (clustering) utilizado para dividir un conjunto de datos en grupos o clústeres basados en características similares. Funciona de la siguiente manera:

1. *Inicialización:* Comienza seleccionando al azar 'k' centroides (puntos iniciales) en el espacio de características.
2. *Asignación:* Luego, asigna cada punto de datos al centroide más cercano, creando 'k' grupos.
3. *Actualización:* Calcula los nuevos centroides de cada grupo tomando la media de todos los puntos asignados a ese grupo.
4. *Reasignación:* Repite los pasos 2 y 3 hasta que los centroides ya no cambien significativamente o hasta que se alcance un criterio de detención predefinido (como un número máximo de iteraciones).

El valor de 'k' se determina antes de ejecutar el algoritmo y afecta directamente la cantidad de clústeres resultantes. En la práctica, encontrar el número óptimo de clústeres ('k') puede ser un desafío y a menudo se recurre a métodos como el método del codo o el coeficiente de silueta para ayudar en esa elección.

El k-means es efectivo en muchos escenarios, pero puede verse afectado por la inicialización aleatoria de centroides, lo que puede llevar a diferentes soluciones. Además, puede tener dificultades con grupos de formas y tamaños irregulares o con datos de alta dimensionalidad.

Es un algoritmo popular debido a su simplicidad y eficiencia computacional, lo que lo hace útil en una amplia gama de aplicaciones, como segmentación de clientes, análisis de datos y compresión de imágenes, entre otros.

El algoritmo k-means es uno de los métodos más populares en el campo del aprendizaje no supervisado. Aquí hay algunos puntos adicionales para comprender mejor este algoritmo:

Selección inicial de centroides: La elección inicial de los centroides puede afectar la convergencia y la calidad de los clústeres. A veces, una mala inicialización puede llevar a soluciones subóptimas o a quedar atrapado en mínimos locales. Estrategias como la inicialización k-means++ intentan abordar este problema seleccionando centroides iniciales de manera más inteligente para mejorar la calidad de los clústeres.

1. *Sensibilidad a valores atípicos:* El algoritmo k-means puede ser sensible a valores atípicos (outliers). Los valores extremos pueden afectar significativamente la posición y el tamaño de los clústeres. A veces, se preprocesan los datos para mitigar este efecto, por ejemplo, utilizando técnicas de escalamiento o transformación de datos.
2. *Evaluación de la calidad de los clústeres:* Determinar el número óptimo de clústeres ('k') es un desafío. Métodos como el método del codo, que traza la varianza explicada en función del número de clústeres, o el coeficiente de silueta, que mide la cohesión y separación de los clústeres, son comunes para evaluar la calidad de la agrupación.
3. *Eficiencia computacional:* El k-means es eficiente y escalable, lo que lo hace útil para conjuntos de datos grandes. Sin embargo, su rendimiento puede degradarse con conjuntos de datos de alta dimensionalidad, debido al efecto de la maldición de la dimensionalidad.
4. *Limitaciones y variantes:* Existen variantes del k-means que abordan algunas de sus limitaciones, como el MiniBatch K-Means, que utiliza lotes de datos para mejorar la eficiencia en conjuntos de datos grandes, o el K-Means++ para mejorar la inicialización de centroides.
5. *Usos en diversas aplicaciones:* El k-means se aplica ampliamente en campos como minería de datos, procesamiento de imágenes, análisis de redes sociales, bioinformática, entre otros, para descubrir patrones, segmentar datos y reducir la dimensionalidad.

El algoritmo K-Nearest Neighbors (KNN) es un método de aprendizaje supervisado utilizado para problemas de clasificación y regresión. Funciona según el principio de que los puntos de datos similares tienden a estar en la misma clase o tener valores similares.

Algunos aspectos clave sobre KNN:

1. *Funcionamiento básico:* Para clasificar un nuevo punto de datos, KNN busca los 'k' puntos de datos más cercanos en función de una medida de distancia (como la distancia euclidiana) desde el punto de consulta. Luego, toma la clase más común entre esos vecinos (en clasificación) o calcula un valor promedio (en regresión) para predecir la clase o el valor del nuevo punto.
2. *Elección del valor de 'k':* La elección de 'k' es crucial en KNN. Valores más pequeños de 'k' pueden llevar a decisiones más sensibles al ruido, mientras que valores más grandes pueden suavizar la frontera de decisión, lo que puede ocasionar la inclusión de puntos de otras clases. En la práctica, se utiliza validación cruzada u otras técnicas para encontrar el valor óptimo de 'k'.
3. *Impacto de la métrica de distancia:* La métrica de distancia utilizada para calcular la cercanía entre los puntos puede variar según el tipo de datos y el dominio del problema. Además de la distancia euclidiana, se pueden usar otras medidas, como la distancia de Manhattan o la distancia de Minkowski.
4. *No paramétrico y sin entrenamiento:* KNN es un algoritmo no paramétrico, lo que significa que no hace suposiciones explícitas sobre la distribución de los datos. Además, no requiere un proceso de entrenamiento costoso, ya que simplemente almacena los datos de entrenamiento para realizar las predicciones.
5. *Sensibilidad a la escala y dimensionalidad:* KNN puede ser sensible a la escala de las características, por lo que a menudo se realiza la normalización de datos antes de aplicar el algoritmo. Además, su rendimiento puede degradarse en conjuntos de datos de alta dimensionalidad debido a la "maldición de la dimensionalidad".
6. *Aplicaciones:* KNN se utiliza en una amplia gama de aplicaciones, como sistemas de recomendación, reconocimiento de patrones, diagnóstico médico, procesamiento de imágenes y más, debido a su simplicidad y facilidad de implementación.
7. *Distancia y métricas:* La medida de distancia más comúnmente utilizada es la distancia euclidiana. Sin embargo, en función del tipo de datos y el problema, se pueden usar otras métricas de distancia, como la distancia de Manhattan, la distancia de Minkowski (que generaliza las dos anteriores) o incluso medidas de similitud coseno para datos vectoriales.
8. *No paramétrico y almacenamiento de datos:* KNN es no paramétrico, lo que significa que no hace suposiciones explícitas sobre la distribución de los datos. Una de sus características es que almacena todos los datos de entrenamiento en memoria, lo que puede requerir mucho espacio, especialmente con grandes conjuntos de datos.
9. *Preprocesamiento de datos:* La escala y la normalización de las características pueden ser importantes para KNN, ya que las diferencias en la magnitud de las características pueden afectar la medida de distancia. Por lo tanto, es común estandarizar o escalar los datos antes de aplicar el algoritmo.
10. *Eficiencia computacional y dimensionalidad:* KNN puede ser costoso computacionalmente, especialmente en grandes conjuntos de datos, ya que necesita calcular las distancias entre el nuevo punto y todos los puntos de entrenamiento. Además, su rendimiento puede degradarse en conjuntos de datos de alta dimensionalidad debido a la dispersión de los datos en un espacio de alta dimensión.

KNN es una herramienta poderosa y fácil de implementar en una variedad de situaciones, pero su rendimiento depende de la elección de 'k', la métrica de distancia, la escala de las características y la dimensionalidad de los datos. Considerar estos factores es esencial para utilizar KNN de manera efectiva y comprender sus limitaciones.

Desarrollo

El código realiza una serie de operaciones para el análisis y predicción de diabetes utilizando dos algoritmos: K-Means y k-NN (k-Nearest Neighbors). La lógica general del código es la siguiente:

1. *Preprocesamiento de datos:*
   - Se carga un conjunto de datos sobre diabetes desde un archivo CSV.
   - Se eliminan los valores vacíos.
   - Se aplica codificación de etiquetas a las características categóricas ('gender' y 'smoking_history').
   - Las características se dividen en independientes (X) y dependientes (Y), y se estandarizan las características independientes.

2. *División de datos:*
   - Se dividen los datos estandarizados en conjuntos de entrenamiento y prueba utilizando una función que realiza una mezcla aleatoria y luego particiona los datos.

3. *Visualización de datos:*
   - Se visualiza la distribución de los datos tridimensionales relacionados con la diabetes utilizando colores distintos para diferenciar la presencia y ausencia de la enfermedad.

4. *Implementación de K-Means:*
   - Se implementa el algoritmo K-Means para crear múltiples modelos (50 en este caso).
   - Se evalúan estos modelos y se selecciona el mejor basado en la precisión en el conjunto de prueba.

5. *Visualización de resultados de K-Means:*
   - Se muestran los datos de prueba junto con los centroides obtenidos por el mejor modelo K-Means seleccionado.

6. *Implementación de K-NN:*
   - Se define y utiliza una implementación propia del algoritmo k-NN para hacer predicciones en los datos de prueba.
   - Se evalúa la precisión del modelo K-NN.

7. *Visualización de resultados de K-NN:*
   - Se muestra una visualización tridimensional de los datos de prueba coloreados según las predicciones hechas por k-NN.

Conclusión

En conclusión, el k-means es un algoritmo poderoso y versátil, pero su desempeño depende en gran medida de la elección de 'k', la inicialización y la naturaleza de los datos. Comprender sus limitaciones y explorar variantes puede ser fundamental para su aplicación efectiva.

Ventajas:

- Su tiempo de procesamiento es rápido.
- Apenas requiere optimización.

Desventajas:

- Es algo más complejo de programar.
- Su precisión no es tan buena y concisa conforme a KNN.
- La precisión varía mucho dependiendo del azar de los centroides iniciales.

K-NN es un algoritmo versátil y fácil de entender que puede ser útil en muchas situaciones, especialmente en conjuntos de datos con estructuras claras y bien definidas. Sin embargo, su rendimiento puede verse afectado por la elección de 'k' y por la naturaleza de los datos, por lo que es importante comprender sus limitaciones y explorar su aplicación con cautela.

Ventajas:

- Resulta más preciso a comparación de K Means
- Es más sencillo de programar.

Desventajas:

- Es muy lento de entrenar y para realizar las predicciones por lo que tuve que optimizar usando lo máximo posible usando numba y vectorización.

El código carga datos, realiza preprocesamiento, divide los datos, implementa K-Means y K-NN para predecir la diabetes, evalúa la precisión de los modelos y visualiza los resultados tridimensionales.

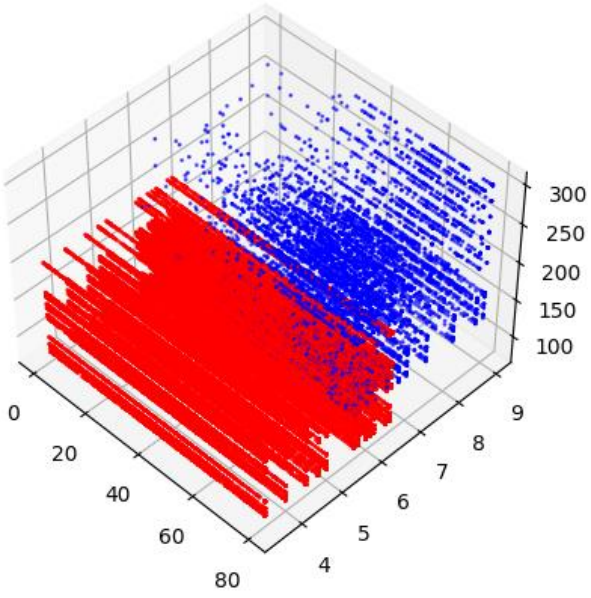
Resultados

Entrenamiento (Score de K-Means): 0.8818125

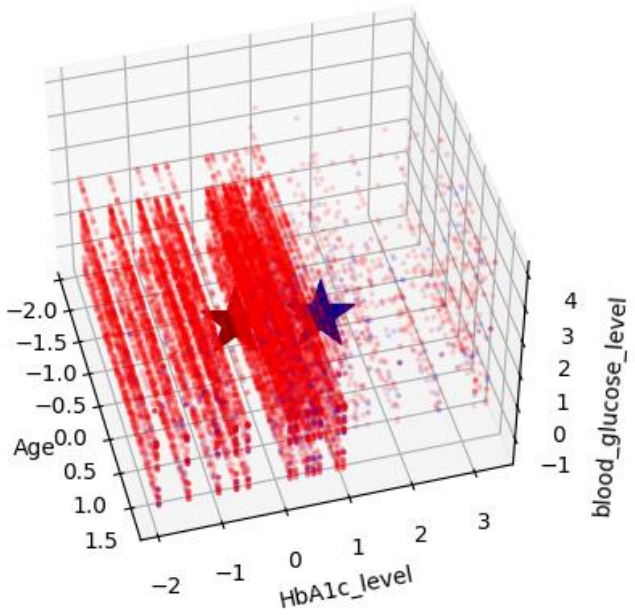
Prueba (Score de K-Means): 0.88655

Score (K-NN): 0.9109

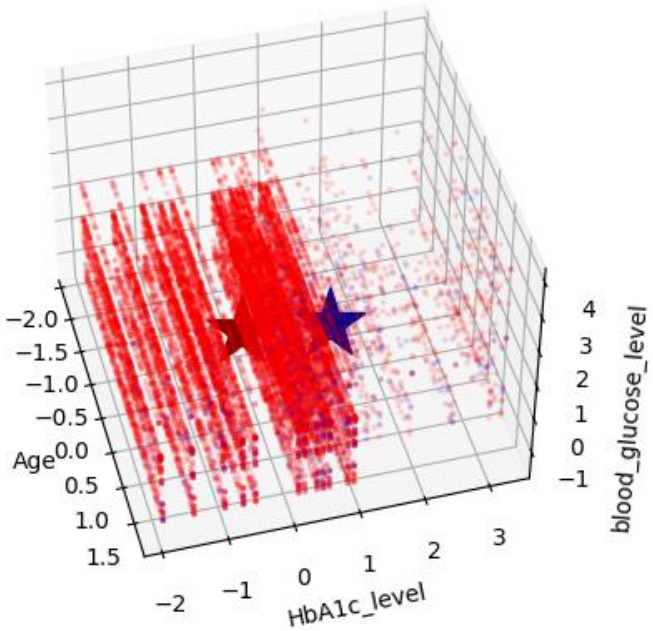
Distribución de los datos con y sin Diabetes



Se grafican los valores de las predicciones y los centroides – K-Means



Se grafican los valores de las predicciones y los centroides – K-NN



Referencias:

colaboradores de Wikipedia. (2023a, septiembre 11). K Vecinos más próximos. https://es.wikipedia.org/wiki/K_vecinos_m%C3%A1s_pr%C3%B3ximos

Na, & Na. (2020, 15 julio). Algoritmo K-Nearest Neighbor | Aprende Machine Learning. Aprende Machine Learning. https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/

Ramírez, L. (2023, 5 enero). Algoritmo K-Means: ¿Qué es y cómo funciona? Thinking for Innovation. https://www.iebschool.com/blog/algoritmo-k-means-que-es-y-como-funciona-big-data/

Gonzalez, L. (2021, 8 septiembre). Algoritmo KMEANS – teoría. Aprende IA. https://aprendeia.com/algoritmo-kmeans-clustering-machine-learning/