**Introducción**

La migraña es una condición médica que afecta aproximadamente al 12% de la población adulta a nivel mundial. Los ataques de migraña causan dolor e incapacidad significativos y representan una carga económica importante en términos de ausentismo laboral y costos médicos directos e indirectos.

Existen varias opciones farmacológicas para el tratamiento agudo de la migraña, incluyendo ergóticos como la cafeína/ergotamina y los triptanes como el sumatriptán. Realizar una evaluación económica comparativa de estas alternativas terapéuticas permite entender mejor los costos, beneficios y trade-offs involucrados en la toma de decisiones.

En este contexto, Evans et al. (1997) desarrollaron un modelo de árbol de decisión para comparar los costos del tratamiento de migraña con sumatriptán versus cafeína/ergotamina en Canadá. El presente informe tiene como objetivo replicar este modelo utilizando la librería networkx de Python, y complementar el análisis determinístico original con simulaciones de Monte Carlo para capturar la incertidumbre en las probabilidades.

El informe se estructura en una sección de desarrollo, donde se explica el código implementado, una sección de justificación de las decisiones de modelado y análisis, y finalmente las conclusiones del estudio realizado. Los resultados permitirán entender mejor las fortalezas y limitaciones de cada estrategia terapéutica en términos económicos.

**Desarrollo**

* Se importan las librerías necesarias: networkx para la generación del grafo, matplotlib para la visualización y random para la simulación.
* Se crea un grafo dirigido G utilizando DiGraph() de networkx.
* Se agregan los nodos uno a uno con nombres significativos utilizando add\_node(). Este enfoque facilita modificar la estructura del árbol fácilmente.
* Se conectan los nodos mediante aristas con add\_edge(), replicando las decisiones y probabilidades del árbol original.
* Se definen posiciones de nodos en un diccionario «pos» para una disposición clara del diagrama. Se utilizan coordenadas cartesianas.
* Se crean diccionarios de costos, probabilities\_sumatriptan y probabilities\_caffeine para parametrizar los valores en las aristas.
* Se grafica el árbol resultante con parámetros de visualización adecuados para mayor legibilidad.
* Se dibujan las etiquetas de costos, probabilidades Sumatriptan y probabilidades Caffeine utilizando las funciones de networkx.
* Explicar con más detalle la lógica de la función deterministic\_analysis():
  + Recibe el diccionario node\_probabilities con las probabilidades en cada nodo.
  + Utiliza una variable current\_node para recorrer el árbol de forma iterativa. Comienza en el nodo raíz "Migraine Attack".
  + En cada nodo, elige una rama de forma aleatoria según las probabilidades en node\_probabilities.
  + Va acumulando el costo en la variable total\_cost a medida que avanza en el árbol.
  + Utiliza una semilla fija (random.seed()) para reproducibilidad.
  + Retorna el nodo final alcanzado y el costo total acumulado.
* Explicar con más detalle la lógica de la simulación de Monte Carlo (monte\_carlo\_simulation\_uniform()):
  + Recibe node\_probabilities y el número de simulaciones a realizar num\_simulations.
  + Utiliza un diccionario results para ir contando las frecuencias de cada nodo final.
  + Por cada simulación, avanza en el árbol de forma aleatoria con probabilidades uniformes hasta llegar a un nodo final.
  + Incrementa el contador en results para ese nodo.
  + Calcula las probabilidades dividiendo las frecuencias sobre el total de simulaciones.
  + Retorna el diccionario de probabilidades estimadas.

**Justificación**

* El uso de la librería networkx permite construir el árbol de decisión de forma más sencilla, flexible y ordenada en contraste con un enfoque manual.
* La parametrización de costos y probabilidades en diccionarios separados facilita la legibilidad, mantenimiento y modificación del modelo.
* La visualización del árbol replicado ayuda a validar que se está implementando correctamente la estructura y parámetros publicados.
* El análisis determinístico es útil para entender el funcionamiento básico del modelo antes de introducir incertidumbre.
* La simulación de Monte Carlo permite propagar la incertidumbre de forma realista al no tener distribuciones conocidas de los parámetros.

**Conclusiones**

* El código Python desarrollado replica de forma precisa el árbol de decisión del estudio, utilizando buenas prácticas de programación.
* Se logra implementar tanto un análisis determinístico como uno probabilístico sobre el modelo.
* El enfoque modular y la parametrización facilitan expandir el análisis en el futuro.
* La simulación de Monte Carlo es una técnica poderosa para los modelos probabilísticos en evaluación de tecnologías sanitarias.
* El uso de librerías especializadas como networkx simplifica la construcción y análisis de modelos como árboles de decisión.

**Bibliografía**

1. Briggs, A. H., Claxton, K., & Sculpher, M. J. (2006). Decision Modelling for Health Economic Evaluation. In Oxford Handbooks in Health Economic Evaluation. Oxford University Press.
2. Evans, R. W., Boan, J. A., Evans, J. L., & Shuaib, A. (1997). Economic evaluation of oral sumatriptan compared with oral caffeine/ergotamine for migraine. Pharmacoeconomics, 12(5), 565–577. <https://doi.org/10.2165/00019053-199712050-00007>
3. Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Proceedings of the 7th Python in Science Conference (SciPy2008) (pp. 11–15). <https://doi.org/10.25080/majora-7b98e3ed-00a>
4. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>