**Descripción general del proyecto**

Nuestro proyecto consiste en desarrollar una aplicación que analice un dataset relacionado a <enfermedad>, utilizando diferentes estructuras de datos como listas, árboles y grafos. La aplicación contará con una interfaz gráfica, que permitirá cargar el dataset y visualizar diferentes variables presentes en él, además de la posibilidad de seleccionar las variables relevantes.

Se implementa un algoritmo de búsqueda utilizando las diferentes estructuras de datos para así acceder a los datos de manera eficiente. Finalmente, la aplicación mostrará las recomendaciones para el tratamiento, brindando al usuario opciones recomendadas según las características y antecedentes médicos.

El proyecto se desarrollara utilizando el lenguaje de programación python y diversas bibliotecas para la interfaz gráfica. Se tomará atención al rendimiento y eficiencia de los algoritmos de búsqueda y análisis, así como que la interfaz gráfica sea una experiencia fluida.

**Fundamentación de la solución al problema planteado**

Para desarrollar la aplicación que analice un dataset de alguna enfermedad y genere recomendaciones de tratamiento, se puede utilizar Python como lenguaje de programación y diversas bibliotecas como pandas, numpy y tkinter para la interfaz gráfica. A continuación, te proporcionaré una estructura básica del programa, donde se utilizan listas, árboles y grafos para comparar su rendimiento en tiempo y memoria:

1. Carga del Dataset:

- Implementa una interfaz gráfica utilizando la biblioteca tkinter para permitir al usuario cargar el dataset en un formato compatible, como CSV o Excel.

- Utiliza la biblioteca pandas para leer el archivo y almacenar los datos en un dataframe.

2. Análisis de Datos:

- Muestra en la interfaz gráfica las diferentes variables presentes en el dataset y permite al usuario seleccionar las variables relevantes para el análisis.

- Realiza el preprocesamiento de los datos, como la limpieza de valores nulos, transformación de variables categóricas y normalización si es necesario.

3. Implementación de Estructuras de Datos:

- Utiliza listas para almacenar los datos del dataset de forma lineal.

- Implementa un árbol para organizar los datos de manera jerárquica, si existe una relación entre ellos.

- Crea un grafo para representar las conexiones y relaciones entre los diferentes elementos del dataset, como por ejemplo, los síntomas, tratamientos y resultados.

4. Algoritmos de Búsqueda y Análisis:

- Implementa diferentes algoritmos de búsqueda, como búsqueda lineal, búsqueda binaria o búsqueda por índice, para comparar su rendimiento en tiempo y memoria.

- Desarrolla algoritmos de análisis de datos, como clustering o regresión, para identificar patrones y predecir el tratamiento más adecuado para cada paciente.

5. Generación de Recomendaciones:

- Utiliza los resultados del análisis de datos para generar recomendaciones de tratamiento basadas en las características y antecedentes médicos de cada paciente.

- Presenta las recomendaciones al usuario a través de la interfaz gráfica.

Recuerda que este es solo un esquema básico, y dependiendo de la complejidad y tamaño del dataset, así como de los algoritmos de búsqueda y análisis utilizados, podrías necesitar ajustar y optimizar el código para obtener un rendimiento óptimo.

**Analisis de requerimiento**

**Tecnologías a utilizar**

* **Lenguaje de programación:** Python
* **Bibliotecas:** Pandas

Scikit-learn

Tensorflow

tkinter

* **Hojas de cálculos:** Microsoft Excel, hoja de cálculo de google
* **APIS?**

**Ventajas y desventajas python**

Se utilizara python para la realización del datasets, ya que cuenta con una sintaxis sencilla y legible. También cuenta con una amplia cantidad de bibliotecas y frameworks. Integración con otras tecnologías y lenguajes como SQL para interactuar con la base de datos. Si bien no es el lenguaje de programación más rápido muchas de sus bibliotecas de análisis de datos están escritas en lenguajes de programación más rápidos como C o C++.

Si bien este lenguaje fue la opción que como grupo escogimos no podemos no considerar las desventajas de este como por ejemplo: la seguridad de tipos, ya que al ser de tipado débil, se pueden cambiar el tipo de dato de una variable sin hacer conversiones, por lo que puede producir errores. Otra desventaja vendría a ser el gran consumo de memoria y la velocidad de depuración no es la más rápida pero como dije anteriormente se compensa con las bibliotecas.

Bibliografia

<https://mappinggis.com/2019/07/lenguajes-de-programacion-para-realizar-ciencia-de-datos/>

<https://keepcoding.io/blog/que-son-datasets/>