**Problem Statement**

Cancer Mortality & Incidence Rates is a dataset that contains information on the mortality and incidence rates of cancer in different populations. Mortality rates refer to the number of deaths due to cancer in a given population, while incidence rates refer to the number of new cancer cases in a given population. This dataset may include information on the type of cancer, the population being studied (e.g., by age, gender, race/ethnicity), and the time period being analysed. It may be used to analyse trends in cancer rates over time and to identify factors that may contribute to differences in cancer rates between different populations. Understanding cancer mortality and incidence rates can be important for public health efforts to prevent and treat cancer. It can help policymakers, healthcare providers, and researchers identify high-risk groups and allocate resources to address the burden of cancer in different populations.

**Brainstorming and prioritizing**

1. **Data Collection and Preparation:** Gather the dataset containing patient information and health conditions. Preprocess the data by handling missing values, converting categorical variables to numerical format, and performing feature scaling if necessary.
2. **Exploratory Data Analysis (EDA):** Perform data visualisation techniques to gain insights into the dataset. Generate visualisations such as histograms, bar plots, correlation matrices, or scatter plots to understand the relationships between variables.
3. **Machine Learning Model Development:** Select a suitable classification algorithm, such as Logistic Regression, Decision Trees, Random Forest, or Support Vector Machines (SVM), based on your problem requirements. Split the dataset into training and testing sets for model evaluation. Train the selected model using the training data. Tune hyperparameters if necessary to improve model performance.
4. **Model Evaluation:** Evaluate the trained model using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, or ROC curve. Assess the model's performance using techniques like cross-validation to ensure generalizability.
5. **Flask Application Development:** Build a web application using Flask to provide a user interface for interacting with the model. Create routes and views for handling user requests and responses. Implement form validation and data preprocessing for user inputs. Utilize the trained model to make predictions based on user inputs. Return the predictions to the user through the web interface. Deployment and Integration: Deploy the Flask application to a web server or a cloud platform such as Heroku or AWS. Integrate the Flask application with a database if necessary to store and retrieve patient information. Ensure proper security measures such as authentication and authorization for user access. Perform testing and debugging to ensure the application functions correctly.