**Project documentation**

**Proposed solution**

The proposed solution for the cancer mortality and incidence rates classification project involves several steps. Firstly, a dataset containing patient information and health conditions is obtained. The dataset is then preprocessed to handle missing values, encode categorical variables, and scale numerical features if necessary. Exploratory data analysis techniques are applied to gain insights from the dataset through visualisations and summary statistics. A suitable classification algorithm, such as Logistic Regression, Decision Trees, Random Forest, or SVM, is chosen to develop the machine learning model. The dataset is split into training and testing sets, and the model is trained using the training data. The model's performance is evaluated using appropriate metrics on the testing data. Finally, a Flask-based web application is developed to create a user interface where users can input their information and receive predictions on cancer mortality and incidence rates based on the trained model.

**Solution Architecture**

The Solution Architecture is divided into several components , as outlined below:

* **Dataset and Data Preprocessing:**
  + Obtain a dataset that includes patient information and health-related features.
  + Clean the dataset by handling missing values and transforming categorical variables into numerical representations.
  + Scale numerical features if needed.
* **Exploratory Data Analysis (EDA):**
  + Explore the dataset to gain insights and identify patterns and relationships.
  + Create visualisations and summary statistics to better understand the data.
* **Machine Learning Model Development:**
  + Choose an appropriate classification algorithm like Logistic Regression, Decision Trees, Random Forest, or SVM.
  + Split the dataset into training and testing sets to train and evaluate the model.
  + Train the selected model using the training data and adjust its parameters if necessary.
* **Flask Web Application Development:**
  + Develop a web application using Flask to provide a user-friendly interface for the cancer mortality and incidence rates classification system.
  + Design and implement routes and views to handle user requests and responses.
  + Apply form validation and data preprocessing techniques to handle user inputs effectively.
  + Utilize the trained machine learning model to make predictions based on user inputs.
  + Display the predictions to the user through the web interface.

**Stakeholder and Customer Requirements**

1. **End Users:**

Users should be able to access the web application easily and receive accurate predictions regarding cancer mortality and incidence rates based on their inputs.

1. **Healthcare Professionals:**

The system should provide reliable and interpretable predictions to assist healthcare professionals in decision-making and patient management.

1. **Data Privacy and Security:**

Ensuring the confidentiality and integrity of patient data by implementing appropriate security measures.

1. **Scalability and Performance:**

Designing the system to handle a potentially large number of users and providing efficient and responsive predictions.

1. **User-Friendly Interface:**

Creating an intuitive and visually appealing web interface for ease of use.

**Requirement Analysis**

* **Dataset:** A dataset containing patient information and health conditions is required for training and testing the machine learning model.
* **Machine Learning Algorithms:** Selection and implementation of appropriate classification algorithms such as Logistic Regression, Decision Trees, Random Forest, or SVM.
* **Data Preprocessing:** Handling missing values, encoding categorical variables, and scaling numerical features if necessary.
* **Exploratory Data Analysis:** Generating visualisations and summary statistics to understand the dataset and identify patterns.
* **Model Evaluation:** Using suitable evaluation metrics to assess the performance of the trained model.
* **Flask Web Framework:** Developing a web application using Flask to create the user interface and handle user requests and responses.
* **Python:** Implementing the project using the Python programming language.

**Open Source Frameworks**

**Machine Learning Algorithms:** classification algorithms such as Logistic Regression, Decision Trees, Random Forest, or SVM.

**Data Preprocessing:** Scikit-learn: Scikit-learn is a widely used machine learning library in Python. It provides a range of classification algorithms, data preprocessing tools, and evaluation metrics, making it suitable for model development and evaluation.

**Exploratory Data Analysis:** NumPy is a fundamental library for scientific computing in Python. It provides efficient numerical operations and data structures, which are essential for working with arrays and manipulating data. Pandas is a data manipulation and analysis library. It offers powerful data structures and data analysis tools, making it useful for data preprocessing, exploratory data analysis, and feature engineering.

**Data Visualization :** Matplotlib and Seaborn: Matplotlib and Seaborn are libraries for creating visualisations in Python. They provide a wide range of plotting functions and customization options, enabling the generation of informative visualisations during the exploratory data analysis phase.