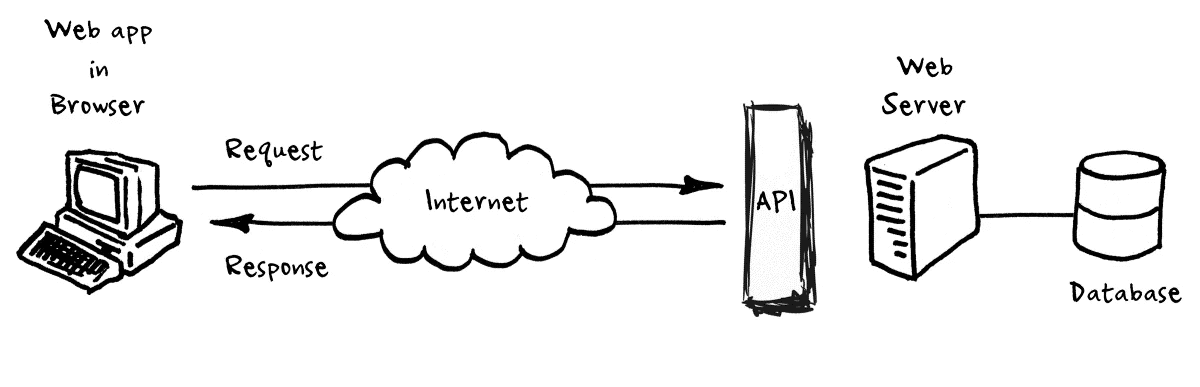
**EMBEDDING MACHINE LEARNING FOR SEPSIS PREDICTION INTO A WEB APPLICATION WITH FASTAPI**

****

**INTRODUCTION**

In the realm of healthcare, early detection and timely intervention can save lives. Sepsis, a life-threatening condition triggered by an infection, requires swift diagnosis and treatment. By embedding a machine learning model for sepsis prediction into a web application, healthcare providers can identify at-risk patients and take proactive measures promptly. In this article, we will explore how FastAPI, a robust web framework, can be utilized to seamlessly integrate a sepsis prediction model into a web application, enabling healthcare professionals to make informed decisions and potentially save lives.

**PROJECT GOAL**

The goal is to develop a web application that can take in relevant patient data and provide predictions on the likelihood of sepsis. By embedding a machine learning model into a FastAPI application, we can automate the prediction process and assist healthcare professionals in identifying potential sepsis cases promptly.

**DATA MODELING/FEATURE SELECTION/ FEATURE ENGINEERING/HYPERPARAMETER TUNING**

For more detailed notebook [Click here](https://github.com/Rose-KamauTheAnalyst/Sepsis-prediction-and-FASTAPI-Deployement)

**FASTAPI**

**1.1. FastAPI: Why Choose FastAPI?**

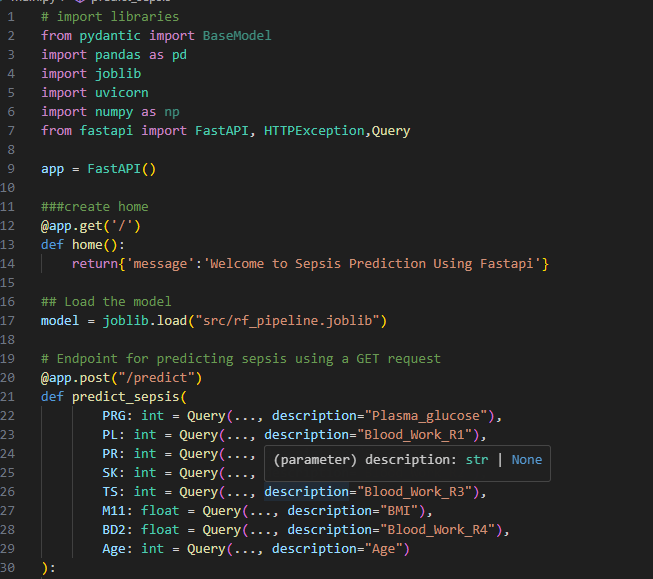
FastAPI, a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints, has gained popularity for its speed and simplicity. In this article, we'll explore how to deploy a Sepsis prediction API using FastAPI and Docker.

FastAPI is rapidly gaining popularity due to its exceptional performance and user-friendly approach. Built on top of Starlette and Pydantic, FastAPI combines high-speed performance with type annotations, resulting in robust web APIs. With asynchronous support and automatic validation of request/response data, FastAPI stands out as an ideal choice for developing scalable and efficient web applications.

**1.2. Creating the API**

Setting Up the FastAPI Application:

The core of our application is a FastAPI script (main.py) that serves as an interface for predicting sepsis. Let's break down the essential components.



[Click here](https://github.com/Rose-KamauTheAnalyst/Sepsis-prediction-and-FASTAPI-Deployement/blob/main/Src/main.py.py) to review this main.py.

A screen shot of a computer program

Description automatically generated

In the above code, we define a FastAPI application, create a home endpoint, and a /predict endpoint to make sepsis predictions based on provided input parameters.

To run the web application [click here](http://localhost:8000/docs)

**CREATING A DOCKERFILE:**

**7.1 What is Docker**

Docker is an open-source platform that allows you to automate the deployment and scaling of applications using containerization. It provides a lightweight and portable environment that encapsulates all the dependencies and configurations required to run an application, making it easier to deploy and manage applications across different environments.

**7.2 Docker Setup and Installation**

Before we proceed with Dockerizing our FastAPI application, ensure that you have Docker installed on your machine. Visit the official Docker website ([https://www.docker.com](https://www.docker.com/)) and follow the instructions to download and install Docker for your specific operating system.

**7.3 The Dockerfile**

To containerize our FastAPI application, we need to create a Dockerfile. The Dockerfile contains instructions for building a Docker image, which is a portable snapshot of our application and its dependencies. Here is the Dockerfile for our FastAPI application:

[Click here](https://github.com/Rose-KamauTheAnalyst/Sepsis-prediction-and-FASTAPI-Deployement/blob/main/Src/Dockerfile) to review this code.

**A screenshot of a computer program

Description automatically generated**

**Building and running the Docker Image:**

To deploy our FastAPI application using Docker, follow these steps:

1. Ensure Docker is installed on your machine.
2. Navigate to the directory containing the Dockerfile and run the following command in the terminal:



1. After the build is complete, run the Docker container:



**Conclusion:**

In this article, we explored the implementation of a Sepsis prediction API using FastAPI and Docker. The combination of FastAPI's simplicity and Docker's containerization provides an efficient way to deploy and scale machine learning applications. Consider extending this example by integrating additional features or deploying the API to cloud platforms for broader accessibility.

**Appreciation**

I highly recommend Azubi Africa for their comprehensive and effective programs. Read More articles about [Azubi Africa here](https://medium.com/@azubiafrica) and take a few minutes to visit this link to learn more about Azubi Africa’s life-changing [programs](https://www.azubiafrica.org/data-analytics?utm_source=medium%2Clinkedin&utm_medium=articles&utm_campaign=DAP+Learners).

Thank you for reading. Feedback will be well appreciated.

For more about this repo get it on my [GitHub](https://github.com/Rose-KamauTheAnalyst)