Serverless Machine Learning Model Deployment with AWS SageMaker

*A Course Project Report Submitted in partial fulfillment of the course requirements for the award of grades in the subject of*

**CLOUD BASED AIML SPECIALITY**

**(22SDCS07A)**

by

**Name of the student : GUDIPUDI LOHITH KUMAR**

**2210030223**

*Under the esteemed guidance of*

**Ms. P. Sree Lakshmi**

Assistant Professor,

Department of Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**K L Deemed to be UNIVERSITY**

*Aziznagar, Moinabad, Hyderabad,*

*Telangana, Pincode: 500075*

April 2025

**K L Deemed to be UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

*Certificate*

This is Certified that the project entitled **“**Serverless Machine Learning Model Deployment with AWS SageMaker**”** which is a experimental,Simulation& hardware work carried out by GUDIPUDI LOHITH KUMAR 2210030223, in partial fulfillment of the course requirements for the award of grades in the subject of  **CLOUD BASED AIML SPECIALITY**, during the year **2024-2025**. The project has been approved as it satisfies the academic requirements.

**Ms.P.Sree Lakshmi Dr. Arpita Gupta**

**Course Coordinator Head of the Department**

**Ms. P. Sree Lakshmi**

**Course Instructor**

**CONTENTS**

Page No.

1. Introduction 4

2. AWS Services Used as part of the project 5

3. Steps involved in solving project problem statement 6

4. Stepwise Screenshots with brief description 7

5. Learning Outcomes 10

6. Conclusion 10

7. References 11

1. **INTRODUCTION**

*Mini Project Title*

In the era of digital healthcare, the ability to make quick, accurate predictions about a patient's condition is crucial for effective treatment planning and resource management. This project aims to leverage the power of machine learning and cloud computing to address this need by deploying a Serverless Machine Learning Model using AWS SageMaker. The project showcases how modern cloud infrastructure can simplify the deployment and scalability of intelligent healthcare solutions.

The core idea is to develop a predictive model that can process healthcare data—such as demographic details, medical history, hospital admission records, and treatment indicators—and output insights such as disease classification or health condition prediction. Traditional model deployment often requires managing servers, infrastructure scaling, and maintenance, which can be time-consuming and expensive. By adopting a serverless architecture, the project ensures that the machine learning model is deployed in a way that is scalable, cost-effective, secure, and easy to maintain.

The project uses a structured healthcare dataset containing anonymized records of patients. Each record includes vital features like gender, age, blood type, date of admission/discharge, medical conditions, and treatment outcomes. These features are preprocessed and used to train a machine learning model—specifically a Random Forest Classifier, chosen for its accuracy and interpretability in healthcare applications.

After training and evaluation, the model is deployed to AWS SageMaker, where it is exposed as an endpoint capable of real-time inference. To make this endpoint accessible and functional in a serverless environment

**2. AWS Services Used as part of the project**

**1. AWS SageMaker**

AWS SageMaker is the cornerstone of this project. It provides a fully managed platform that streamlines the machine learning workflow—from data preparation to model training, evaluation, and deployment. In this project, SageMaker was used to:

Train the machine learning model (Random Forest Classifier) using a preprocessed healthcare dataset.Host the trained model as a real-time endpoint that can receive input data and return predictions with low latency.

**2. Amazon S3 (Simple Storage Service)**

Amazon S3 acts as the central storage hub for all project assets. It was used to

Store the healthcare dataset, making it accessible for preprocessing and model training.

**3. AWS Identity and Access Management (IAM)**

IAM ensures that all AWS resources interact securely and only with authorized entities. It was used to Create IAM roles and policies that allowed SageMaker to read/write from S3 and invoke other AWS services.Secure access for Lambda functions, ensuring they could call the SageMaker endpoint without exposing credentials.

**4. Amazon CloudWatch**

Amazon CloudWatch plays a key role in monitoring and observability within the deployment.

**3. Steps involved in solving project problem statement**

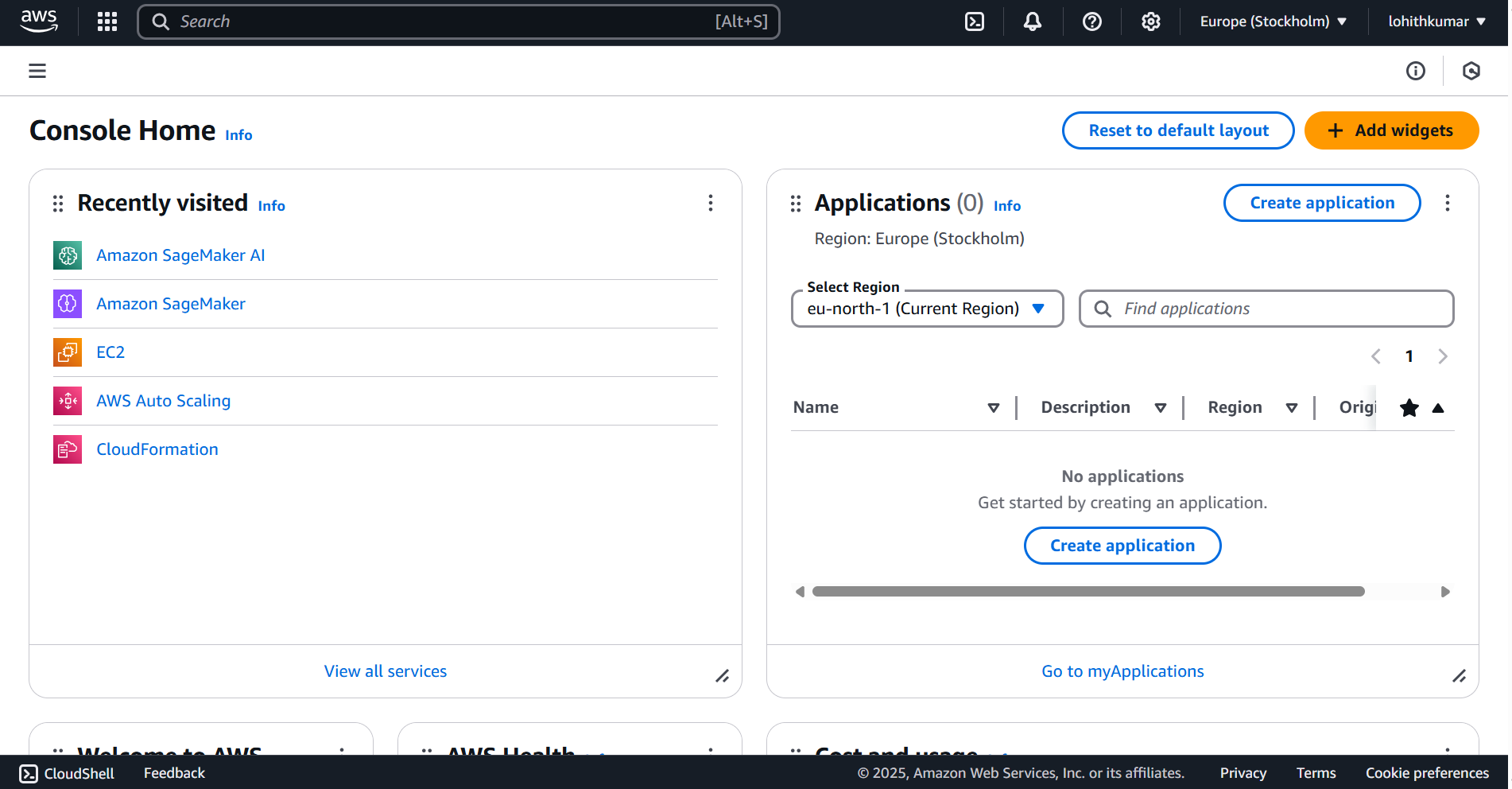
The project began with data loading and preprocessing, where the healthcare dataset was loaded using the Pandas library. Irrelevant columns such as 'Name', 'Doctor', 'Hospital', and 'Room Number' were removed to eliminate noise from the data. To maintain consistency, all column names were converted to lowercase and spaces were replaced with underscores. Additionally, duplicates were dropped, and missing values were handled to ensure data quality and integrity.

Next, in the feature engineering phase, date-related columns like 'date\_of\_admission' and 'discharge\_date' were converted to datetime format to enable time-based computations if needed. Categorical features such as 'gender', 'blood\_type', and 'medical\_condition' were encoded using LabelEncoder to make them suitable for the machine learning model. Numerical features were standardized using StandardScaler to ensure uniform scale and improve model performance.

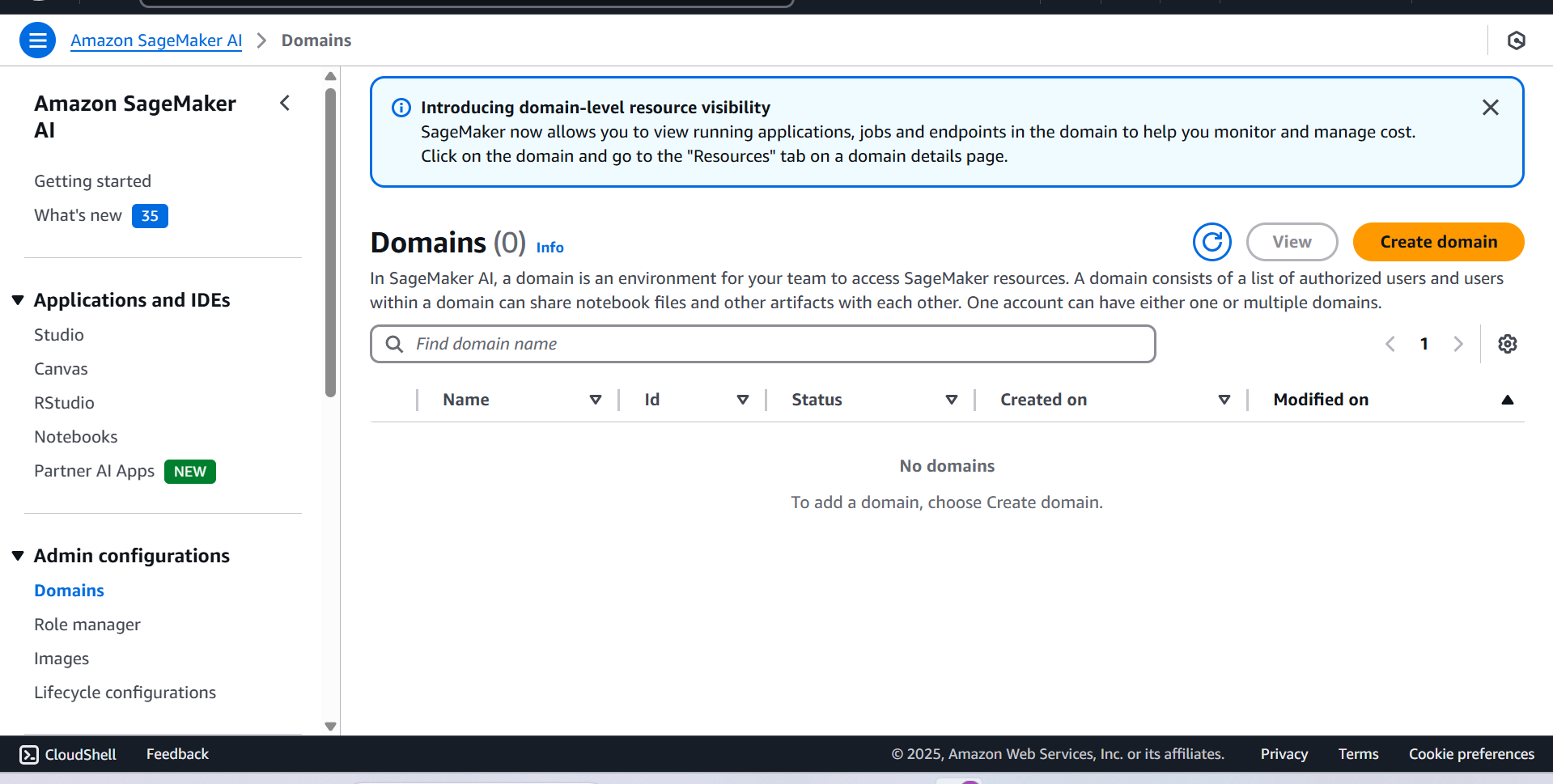
For model building and evaluation, the cleaned dataset was split into training and testing sets. A RandomForestClassifier was trained on the training data due to its robustness and efficiency in handling structured datasets. To further improve the model’s performance, GridSearchCV was employed for hyperparameter tuning. The model was then evaluated using various metrics including accuracy, confusion matrix, and classification report, which provided insights into how well the model was performing on unseen data.

Finally, the model was prepared for deployment using AWS SageMaker. The trained model was serialized using joblib and the artifact was uploaded to an Amazon S3 bucket. A SageMaker endpoint was created to serve the model for real-time inference. To build a fully serverless architecture, an AWS Lambda function was configured to invoke the SageMaker endpoint upon receiving prediction requests. This Lambda function was then integrated with Amazon API Gateway, which enabled the model to be accessed securely over HTTP, completing the end-to-end deployment pipeline.

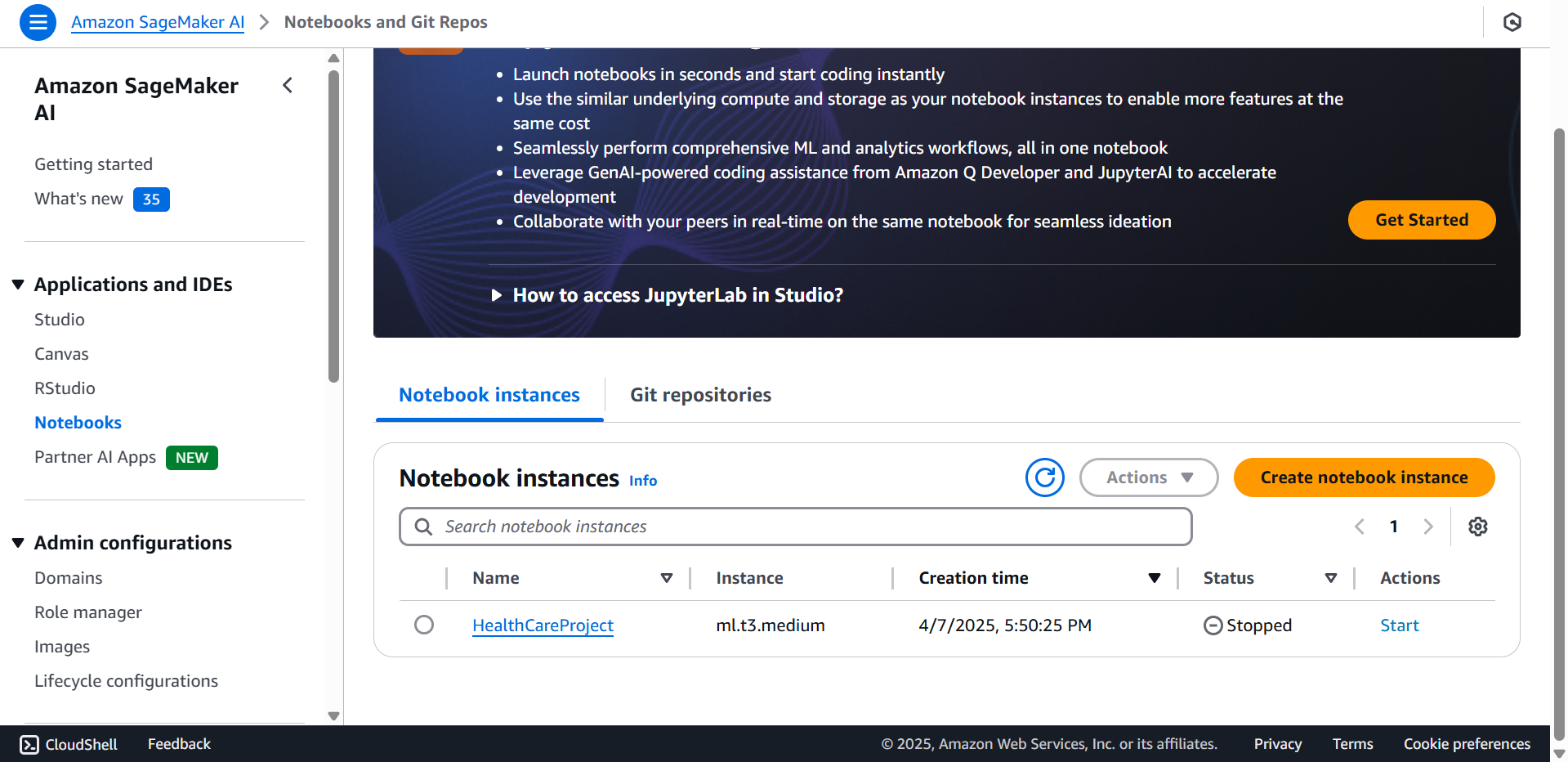
**4. Stepwise Screenshots with brief description**

****

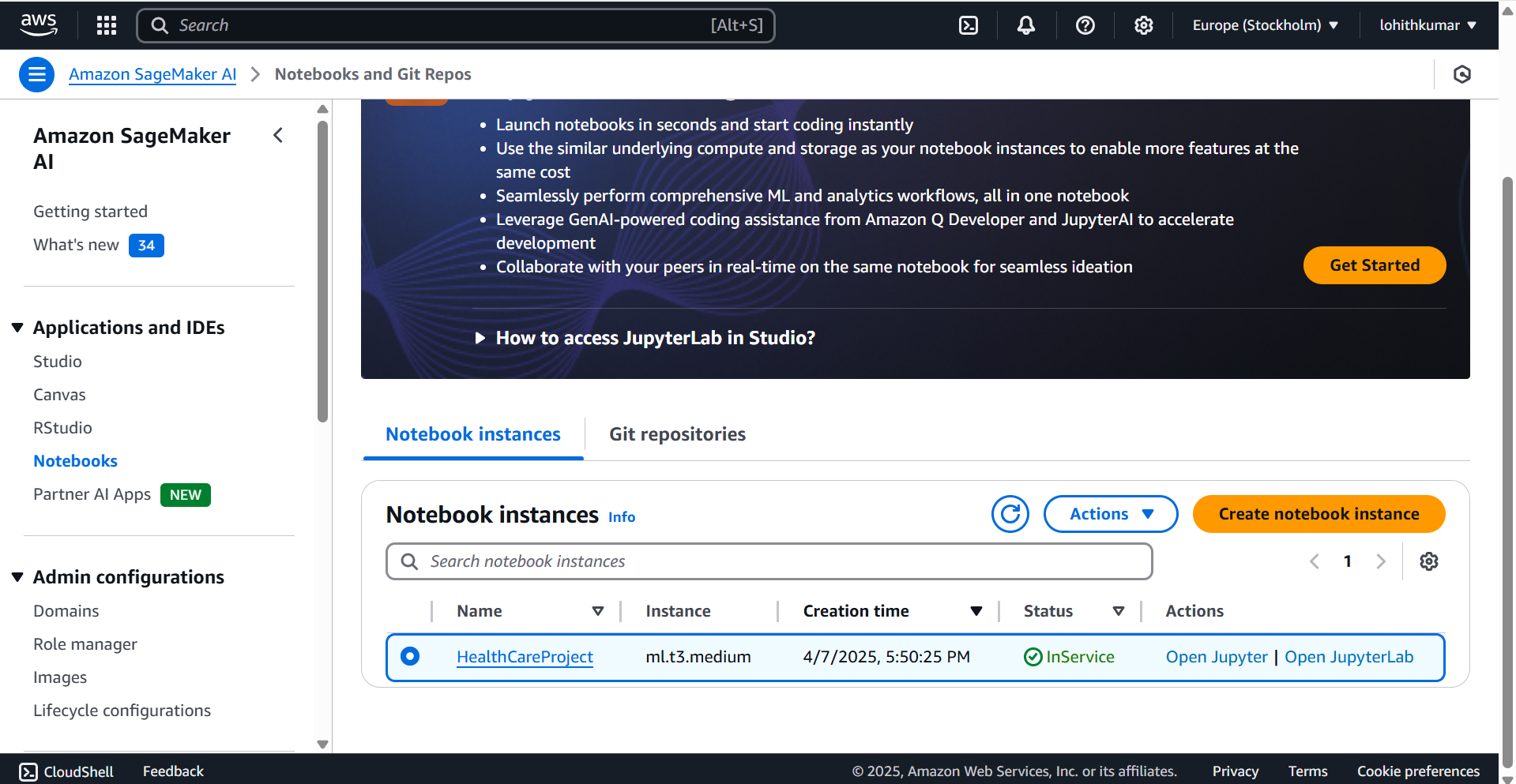
1-Login to the AWS free tier account

****

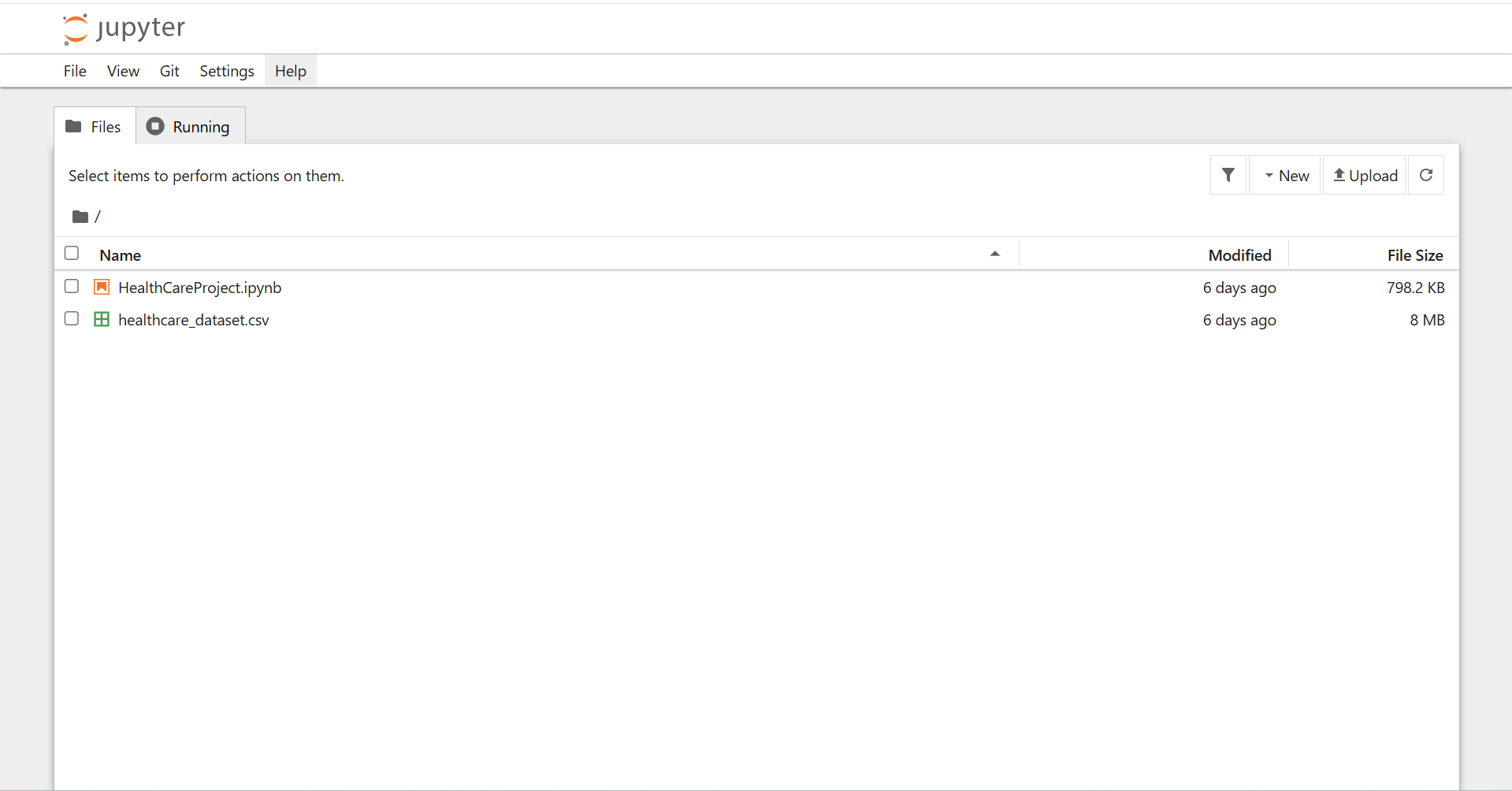
2-Search for the AMAZON SAGEMAKER AI

****

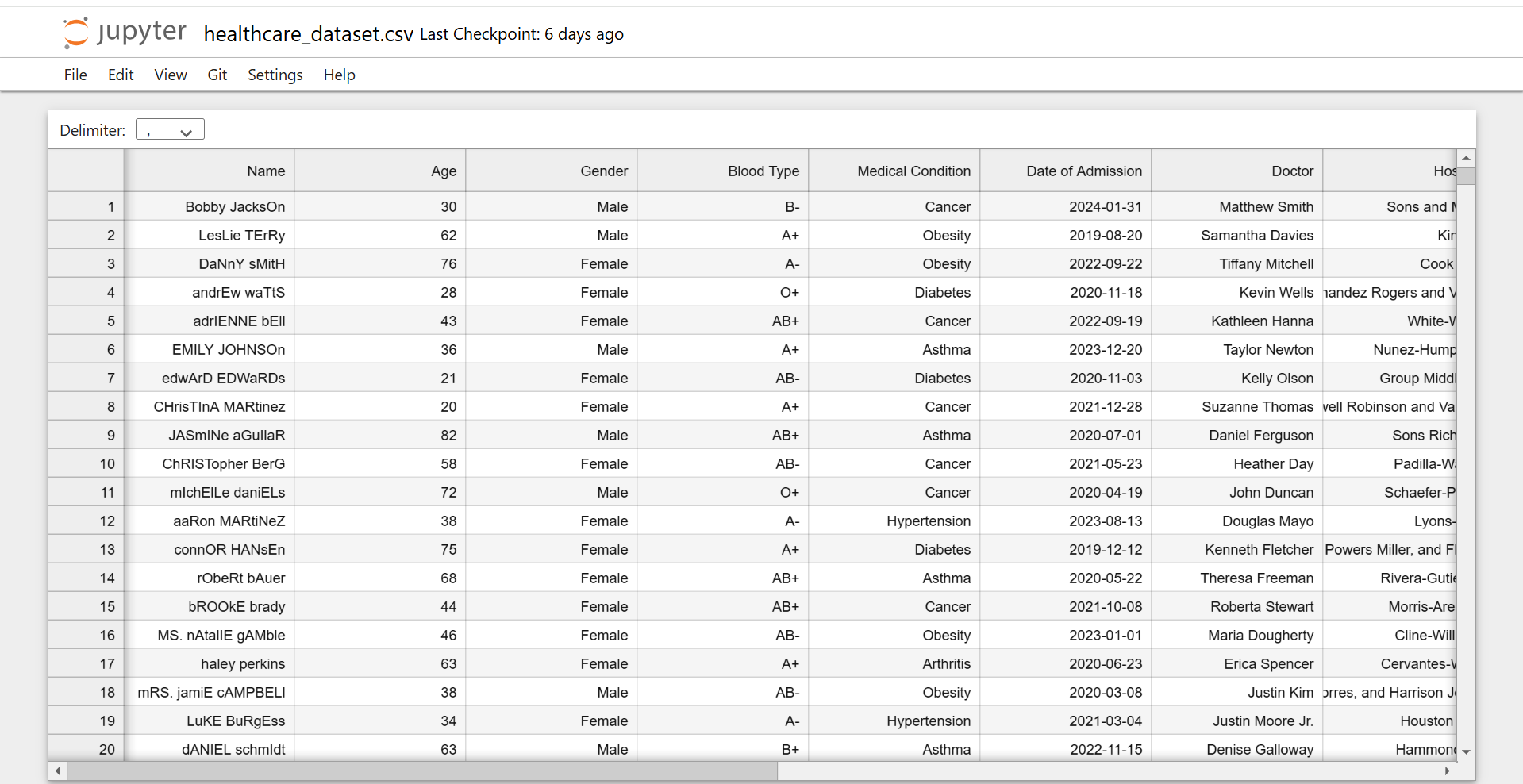
3-Create the notebook using the Sagemaker AI



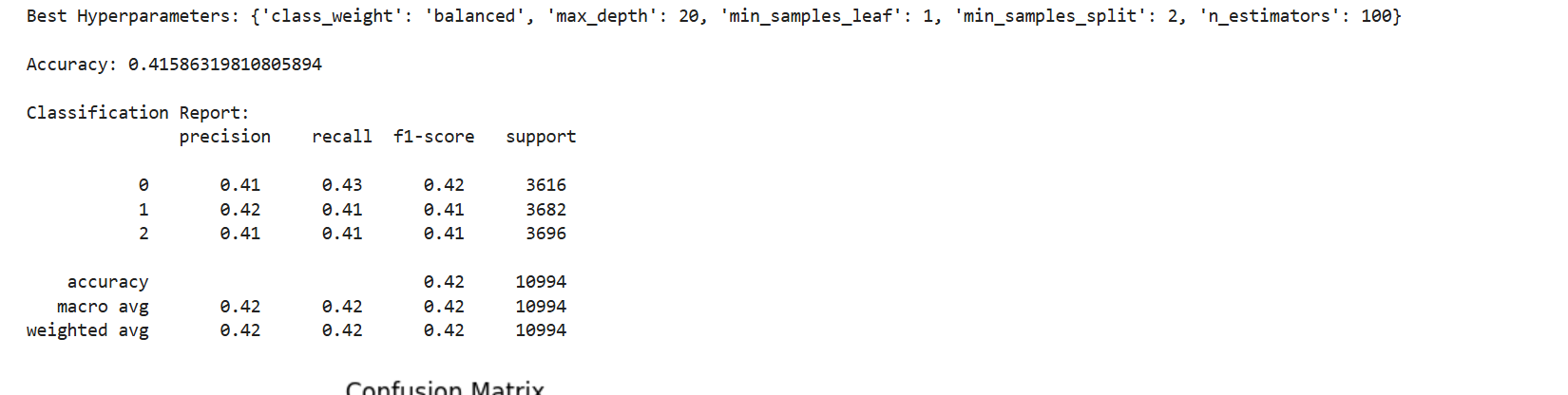
4-Start the Services and open Jupyter notebook

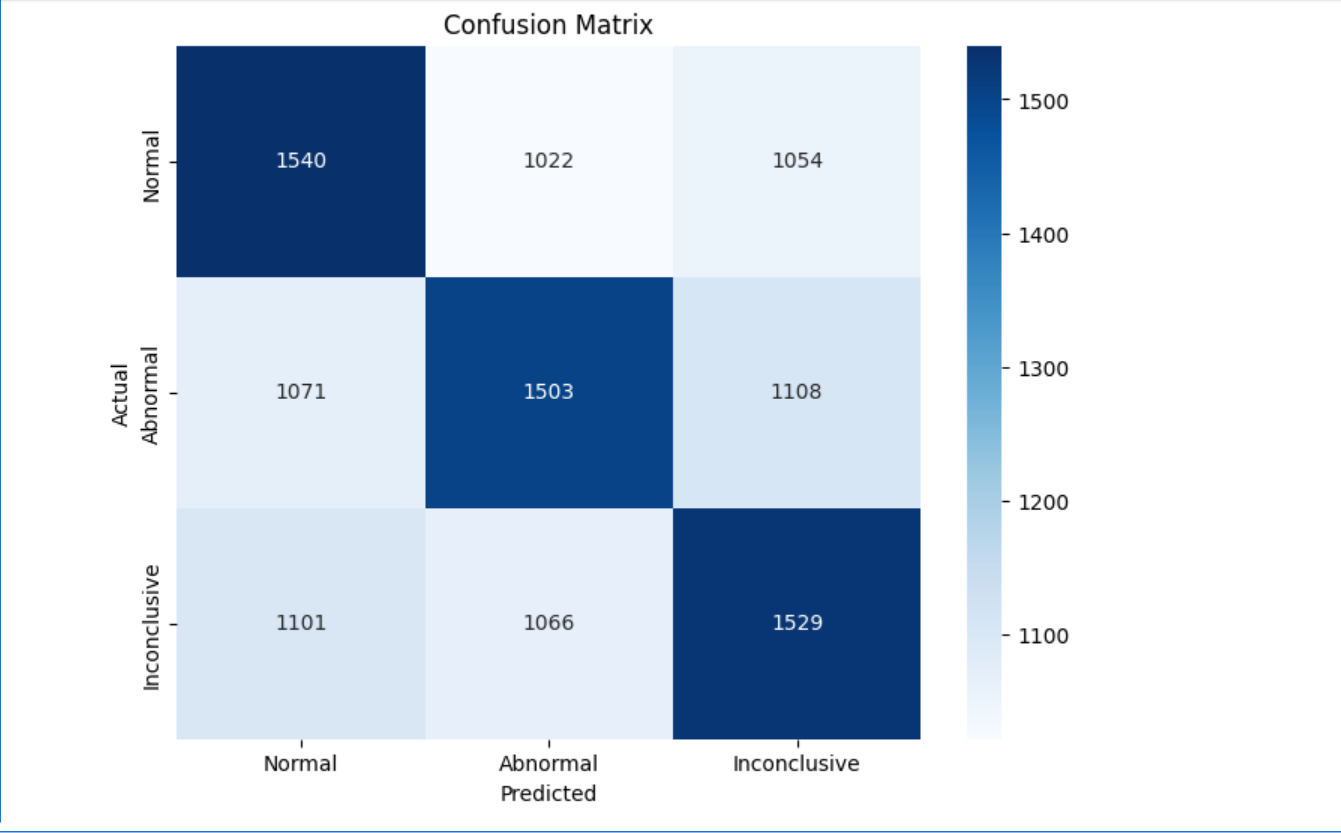


5- Take the csv file and ipynb file



6-Data set in my csv file





7-Final output of my project

**5. Learning Outcomes**

This project provided a comprehensive, hands-on experience in building and deploying a machine learning solution using cloud-based infrastructure. It offered practical exposure to AWS SageMaker, enabling the training, deployment, and management of machine learning models in a fully managed cloud environment. A key takeaway was the understanding of serverless architecture, achieved through the use of AWS Lambda and API Gateway, which allowed for scalable, on-demand inference without the need to manage backend servers. The project also strengthened skills in data preprocessing and feature engineering, particularly in handling categorical and numerical features, managing missing data, and applying appropriate encoding and scaling techniques. Additionally, model optimization techniques such as GridSearchCV were employed to fine-tune hyperparameters and boost model performance. The integration of multiple AWS services—including S3, SageMaker, Lambda, IAM, and CloudWatch—enabled the development of a secure, efficient, and maintainable end-to-end machine learning pipeline. Finally, the use of CloudWatch offered valuable insights into monitoring and debugging deployed ML models, providing visibility into logs and performance metrics. Overall, the project helped solidify concepts in both machine learning and cloud computing, with a strong focus on solving real-world problems in the healthcare domain.

**6. Conclusion**

In conclusion, this project successfully demonstrates the end-to-end workflow of a serverless machine learning model deployment using AWS services. From data cleaning and feature engineering to model training and cloud deployment, each step was carefully executed to build a scalable and reliable prediction system for healthcare applications. The use of AWS SageMaker drastically simplified model training and deployment, while services like Lambda and API Gateway enabled real-time access in a cost-effective, serverless manner. This project not only highlights the technical feasibility of deploying ML in cloud ecosystems but also emphasizes its practical impact in domains like healthcare where fast, reliable predictions are critical. The architecture used here can be easily adapted or scaled for other industries and use cases, making it a powerful blueprint for cloud-based AI solutions.

**7. References**

* AWS SageMaker Documentation
* AWS Lambda Documentation
* Amazon API Gateway
* Amazon S3 Documentation
* Amazon CloudWatch
* Scikit-learn Documentation
* Pandas Library
* Healthcare Dataset