

**CAPSTONE – Final Project Report**

**Project Topic: Disease Prediction Using Machine Learning with Amazon SageMaker**

**Submitted by,**

**Ancy Saji (100925334)**

**Irin Mariya Joseph (100930869)**

**Neethu Puzhakkal (100932992)**

**Introduction**

The project, titled "Disease Prediction System Using Machine Learning with Amazon SageMaker," represents a comprehensive effort to revolutionize disease detection and forecast through the integration of advanced machine learning strategies with the effective infrastructure given by Amazon SageMaker. Its overarching objectives extend past mere technological advancement to include crucial changes in healthcare delivery and patient results.

At its core, this project is driven by the basic need for early disease detection, a critical factor in relieving the effects of different health conditions on people and communities. By leveraging the capabilities of Amazon SageMaker, the project aims to create a versatile and effective infrastructure capable of taking care of fluctuating client requests consistently. This scalability guarantees that the framework can adjust to varying levels of client traffic, subsequently optimizing resource utilization and keeping costs under control.

Past the technical aspects, the user interfaces and APIs designed as part of this activity are meticulously created to be intuitive and simple to explore, guaranteeing that healthcare experts can proficiently interpret disease forecast results and incorporate them into clinical decision-making processes. In addition, the system is planned to be consistently coordinated with existing healthcare platforms, encouraging its adoption and improving interoperability over the healthcare ecosystem.

Central to the project's goals is the deployment of machine learning models in real-time, enabled by Amazon SageMaker's capabilities. This real-time prediction capability holds immense potential in giving timely insights into disease risks and forecasts, permitting proactive interventions and personalized healthcare administration techniques. By persistently checking model performance metrics and data distribution shifts, the framework guarantees the progressing accuracy and reliability of disease predictions, thereby maximizing its clinical utility.

In essence, this project represents a noteworthy progression in healthcare technology, with the potential to revolutionize disease detection and forecast. By tackling the power of machine learning and cloud computing, it seeks to enable healthcare suppliers with actionable experiences and patients with timely interventions, eventually driving to improved health results and enhanced quality of life.

# **Background**

The inspiration behind selecting the "Disease Prediction System Using Machine Learning with Amazon SageMaker" project stems from a combination of thorough preliminary investigate, individual experiences, and a deep seated commitment to addressing a pressing healthcare challenge.

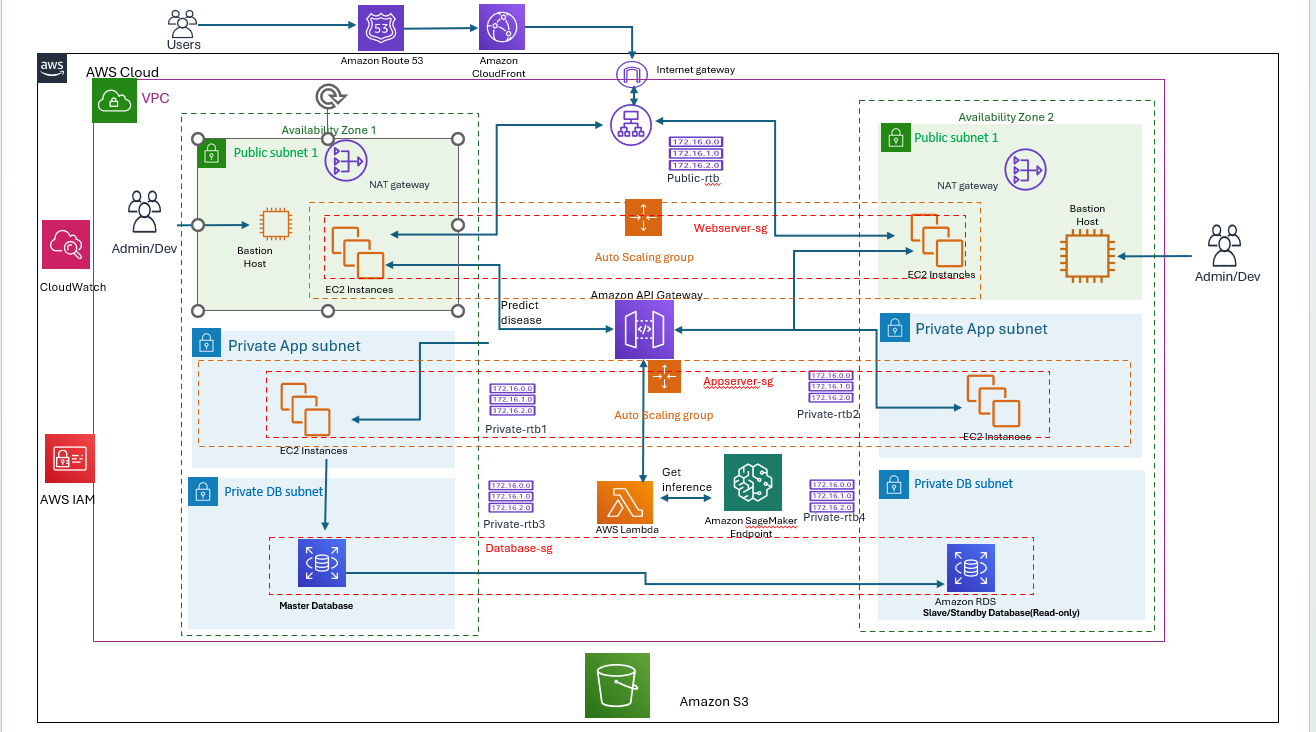
Initial examinations into the scene of healthcare delivery uncovered a glaring gap within the early detection and forecast of diseases, which provoked a more profound exploration into potential solutions. Existing literature and studies, including those by Abdali-Mohammadi et al. (reference 1) and Muchhala et al. (reference 3), highlighted the transformative potential of machine learning in anticipating diseases based on patient information. These discoveries underscored the importance of leveraging progressed technologies to upgrade disease detection accuracy and enable timely interventions.

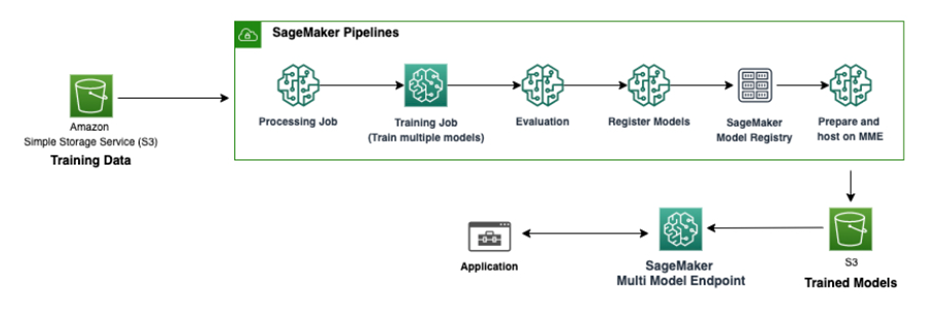
Personal experiences within healthcare settings further reinforced the criticalness of this endeavor. Witnessing firsthand the repercussions of delayed diagnoses and imperfect treatment results underscored the critical need for predictive tools capable of recognizing disease risks at an early stage. These insights, coupled with an enthusiasm for leveraging technology to drive positive change in healthcare, set the choice to pursue this project.

At its core, the problem statement addressed by this project revolves around the insufficiencies of conventional disease detection strategies in capturing subtle early warning signs and foreseeing disease trajectories precisely. Conventional diagnostic approaches often depend on symptomatic presentations, which may not show until diseases have progressed significantly. As a result, openings for preventive mediations and personalized treatment methodologies are frequently missed, driving to adverse health outcomes and increased healthcare costs.

By developing a Disease Prediction System powered by machine learning algorithms and Amazon SageMaker, this project aims to bridge this gap by giving healthcare suppliers with noteworthy insights inferred from patient information. The system's capacity to analyze diverse datasets and identify patterns indicative of illness risks holds immense promise in encouraging early interventions and improving patient results. Eventually, the project seeks to revolutionize disease detection and prognosis, introducing in a modern era of proactive and personalized healthcare delivery.

# **Architecture Design**





# **Gap Analysis**

The transformation of our project design from its initial concept to its current state underscores a journey of substantial enhancements aimed at rectifying shortcomings and elevating the project's overall efficacy and user engagement.

**Limitations of the Previous Design:**

**Single Disease Prediction:** Initially, our project was tailored to predict a single disease based on user input. However, this approach presented limitations, as it failed to cater to the diverse health concerns and conditions prevalent among users. By restricting the system to single disease prediction, we inadvertently narrowed its utility and relevance.

**Inadequate Security and Availability:** When we were first designing, we did not pay enough attention to security and availability. This oversight carried significant risks, such as weakened user confidence, system failures, and data breaches. Without robust security measures and scalable infrastructure, the system's dependability and integrity were at risk.

**Frontend Design Simplifications:** The frontend design in its initial iteration lacked sophistication and advanced features. This simplistic approach hindered user interaction and engagement, limiting the system's appeal and usability. Users encountered challenges navigating the interface and lacked essential functionalities to tailor their experience to their specific needs.

**Need for Changes and Improvements:**

**Expansion to Multiple Disease Prediction:** Understanding the need for more precise application we pivoted towards implementing multiple disease prediction capabilities. This strategic shift broadens the system's applicability, allowing users to receive predictions tailored to their unique health profiles and concerns. By supporting a wider range of illnesses, we improve the system's applicability and relevance.

**Enhanced Security and Availability:** In order to address the weaknesses in these areas, we made significant changes to the system's security and availability. By establishing multiple availability zones and fortifying security features like encryption, authentication techniques, and access controls, we lower risks and establish confidence in users regarding data privacy and system dependability.

**Frontend Design Enhancements:** We made significant improvements to the frontend design in order to promote a more understanding and enjoyable user experience.Introducing search functionality and empowering users to manage diseases dynamically enhances usability and customization options. These improvements facilitate seamless navigation and interaction, fostering greater user satisfaction and engagement.

**Achievements with the New Design:**

Expanded Predictive Capabilities: The integration of multiple disease prediction capabilities represents a significant milestone, augmenting the system's predictive accuracy and relevance. Users now benefit from personalized predictions aligned with their specific health conditions, enhancing the system's value proposition and utility.

Increased Security and Reliability: We have greatly increased the system's resilience and reliability by implementing stringent security improvements and infrastructure optimizations.

Improved User Experience: with user friendly UI and improvisation with the addition of search capabilities and dynamic illness management, users can now customize their experience, which increases satisfaction and engagement.

Essentially, the development of our project design represents a dedication to tackling constraints and providing a strong, adaptable, and user-focused outcome. We have advanced the project toward attaining its goals while satisfying the changing needs and expectations of our users through strategic adjustments and enhancements.

# **Implementation**

**Cloud Resources:**

For our project, we leveraged a variety of cloud services and resources to build and deploy our disease prediction system:

Amazon S3 (Simple Storage Service): Used to store project data securely, including the CSV file containing medical datasets.

Amazon SageMaker: Utilized for model training and deployment. We imported a sample Jupyter notebook into SageMaker to train our disease prediction model.

Amazon EC2: Deployed our website on EC2 instances to ensure high availability and scalability. Load balancing and auto-scaling features were implemented across multiple availability zones for enhanced performance.

Amazon Lambda: Integrated with the SageMaker endpoint to process prediction requests triggered by user inputs on the website. The Lambda function executed the prediction logic and returned results to the UI in real-time.

Amazon RDS (Relational Database Service): Employed for managing structured data, such as user information and symptom data. This ensured efficient data organization and retrieval for disease prediction.

Amazon API Gateway: Facilitated communication between the website's frontend UI and the Lambda function, enabling seamless interaction and data exchange.

Application Load balancer and Auto-scaling group: We used these features to make our system highly available and scalable

**Development Stages:**

**Data Collection:**

Data collection for disease identification involved gathering real symptoms of various diseases from credible sources on the internet. To ensure authenticity and reliability, no dummy values were entered during the data collection process. Instead, comprehensive symptom datasets were obtained from reputable sources such as Kaggle.com and various health-related websites.

The dataset obtained consisted of 5000 rows of patient records, each containing information about specific symptoms. These symptoms spanned across 132 different types, covering a wide range of health indicators. Additionally, the dataset categorized these symptoms according to their corresponding diseases, with a total of 40 classes of general diseases identified.

By meticulously curating this dataset from reliable sources, we ensured the accuracy and comprehensiveness of the symptom data, facilitating the development of a robust disease prediction model. This extensive dataset served as the foundation for training the machine learning algorithms to accurately predict diseases based on input symptoms.

Dataset Link: [DISEASE PREDICTION USING MACHINE LEARNING WITH GUI (kaggle.com)](https://www.kaggle.com/datasets/neelima98/disease-prediction-using-machine-learning):

**Data Setup and Model Training:**

We initiated the project by setting up an Amazon S3 bucket and uploading the required CSV file containing medical datasets.

In this stage, we also constructed Python code implementing the random forest classification algorithm with scikit-learn. Our dataset was then divided into training and testing data, ensuring the model's training on a subset while retaining another portion for evaluation. This process facilitated the assessment of the model's accuracy and generalization abilities, enhancing the reliability of our disease prediction system.

Using Amazon SageMaker, we imported a sample Jupyter notebook and trained a machine learning model capable of disease prediction. This involved creating a SageMaker notebook instance, uploading the notebook file, and executing it to train the model.

**Website Deployment and User Interaction:**

Following model training, we deployed our website on Amazon EC2 instances. The website's frontend UI was designed to prompt users to input personal information and select symptoms for disease prediction.

We ensured high availability and security of the website using load balancing and auto-scaling features across multiple availability zones.

**User Input and Disease Prediction:**

Users interacted with the website by providing personal information and selecting symptoms from a list. Selected symptoms were added to a symptom list, and users could remove any accidentally added symptoms.

Upon clicking the "Predict Disease" button, the selected symptoms were formatted into an array and sent to a REST API, triggering a Lambda function.

**Lambda Function and Result Presentation:**

The Lambda function, integrated with the SageMaker endpoint, processed the symptom array and generated disease predictions in real-time.

The prediction results were relayed back to the website's UI using JavaScript code, providing users with instant disease diagnosis outcomes.

This comprehensive cloud-based implementation ensured the efficient functioning of our disease prediction system, offering users a seamless and intuitive experience while facilitating accurate disease predictions.

**Configuration and Deployment:**

**Amazon S3 Bucket Creation:**

We began by setting up an S3 bucket to store our project data securely.

**CSV File Upload:** After creating the S3 bucket, we uploaded the CSV file containing medical datasets to the bucket. This file was essential for training our disease prediction model later in the project.

We used the AWS Management Console to upload the file, ensuring that it was accessible and properly organized within the bucket.

**Amazon SageMaker Model Training:**

Moving on to model training, we utilized Amazon SageMaker to develop and train our disease prediction model. We imported a sample Jupyter notebook into SageMaker to streamline the process.

We created a SageMaker notebook instance and uploaded the notebook file containing the model training code. Then, we executed the notebook to train the model using the medical datasets stored in the S3 bucket.

**Website Deployment on Amazon EC2:**

With the model trained, we focused on deploying our disease prediction system's frontend website on Amazon EC2 instances. This step involved configuring and launching compute instances to host the website. We ensured high availability and scalability of the website by implementing load balancing and auto scaling features across multiple availability zones.

**User Interaction and Prediction Process:**

After deployment of the website, users can access the website and can interact and predict their disease with it by providing personal information and selecting symptoms for disease prediction.

**User Accesses Website**:

Users access the website, where they are prompted to provide personal information such as name, age, and gender.

**Symptom Selection:**

Upon entering personal information, users are presented with a comprehensive list of over 100 symptoms. They have the option to select multiple symptoms relevant to their condition.

**Symptom Management:**

Users can add selected symptoms to a symptom list. Additionally, they have the flexibility to remove any accidentally added symptoms from the list.

**Disease Prediction Request:**

After finalizing their symptom selection, users click the "Predict Disease" button to initiate the disease prediction process.

**Symptom Formatting:**

The selected symptoms are formatted into an array representation using JS code where the presence of each symptom is denoted by '1', and absence by '0'.

**REST API Call:**

The formatted symptom array is sent to a REST API, triggering a Lambda function associated with the SageMaker endpoint.

**Lambda Function Execution:**

The Lambda function processes the incoming symptom array using the trained machine learning model deployed on SageMaker.

**Disease Prediction:**

Based on the processed symptom array, the Lambda function generates predictions for potential diseases.

**Result Presentation:**

The prediction results are relayed back to the website's UI using JavaScript code, enabling users to view instant disease diagnosis outcomes.

**Lambda Function and Result Presentation Flow:**

**Integration with SageMaker Endpoint:** The Lambda function is integrated with the SageMaker endpoint generated during the model training phase.

**Symptom Array Processing:** Upon receiving the symptom array from the REST API call, the Lambda function processes the data to extract relevant features for disease prediction.

**Disease Prediction:** Leveraging the machine learning model deployed on SageMaker, the Lambda function performs disease prediction based on the processed symptom array.

**Result Relay to UI:** The prediction outcomes, representing potential diseases, are relayed back to the website's UI using JavaScript code.

**Real-Time Presentation:** Users receive the disease prediction results instantly on the website's UI, providing them with timely and accurate diagnosis information.

(https://aws.amazon.com/sagemaker)(https://medium.com/)(https://docs.aws.amazon.com/lambda)(https://aws.amazon.com/tutorials/run-serverless-code/)(https://aws.amazon.com/what-is/api/)

**Challenges Encountered and Resolutions:**

**Function Code Issues in Lambda:** During the development of our Lambda function, we encountered issues with the Python code implementation. Specifically, there were errors related to data parsing and handling within the function.

To address this, we thoroughly reviewed and debugged the Python code, identifying and correcting syntax errors, data processing inconsistencies, and runtime issues.

Additionally, we optimized the code for efficiency and performance, ensuring smooth execution and accurate prediction outcomes within the Lambda environment.

**Model Training Errors**: During model training in SageMaker, we encountered errors related to data preprocessing and algorithm selection. We addressed this by refining our data preprocessing techniques and experimenting with different machine learning algorithms until achieving satisfactory results.

**Website Deployment Complexity:** Deploying the website on Amazon EC2 instances presented challenges related to configuration and scaling. We overcame this by referring to AWS documentation, tutorials, and community forums to implement load balancing and auto scaling effectively.

Through these efforts, we successfully resolved technical challenges encountered during the implementation process, ensuring the smooth functioning and reliability of our disease prediction system.

# **Screenshots:**

EC2 instance for website

A screenshot of a computer

Description automatically generated

Website homepage

A close-up of a heart

Description automatically generated

Getting user input

A screenshot of a computer

Description automatically generated

Result

A screenshot of a computer

Description automatically generated

Creation of ML-trained model and Endpoint

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Training job details

A screenshot of a computer

Description automatically generated

Endpoint

A screenshot of a computer

Description automatically generated

Trained ML model

A screenshot of a computer

Description automatically generated

Visual representation of the ML tasks

A screenshot of a computer

Description automatically generated

IAM role for Amazon SageMaker tasks

A screenshot of a computer

Description automatically generated

Lambda function

A screenshot of a computer

Description automatically generated

IAM role for Lambda function(to invoke ML endpoint)

A screenshot of a computer

Description automatically generated

REST API configuration through API Gateway

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Website running through Load Balancer

A close-up of a heart

Description automatically generated

A screenshot of a computer

Description automatically generated

Load Balancer configuration

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Auto-scaling group

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Instances created after the addition of autoscaling policy

A screenshot of a computer

Description automatically generated

# **Learning Outcome**

**Technical Skills Enhancement:**

**Cloud Computing Expertise**: We obtained practical experience with cloud computing by setting up and implementing a variety of AWS services which includes SageMaker, S3, EC2, API and Lambda. This included configuring our disease prediction system's infrastructure,, allocating resources, and maximizing its performance.

**Machine Learning Proficiency:** Training and deploying machine learning models using SageMaker enhanced our understanding of model development, hyperparameter tuning, and real-time inference. We gained insights into data preprocessing, model selection, and evaluation techniques, crucial for accurate disease prediction.

**Web Development Competence:** Implementing user interfaces and integrating backend services improved our web development skills. We learned about frontend design, backend logic implementation, and API integration, contributing to a comprehensive understanding of full-stack development.

**Managerial Skills Enhancement:**

**Project Planning and Management**: Our ability to plan and carry out project tasks as a team improved our project management abilities. To guarantee that milestones are completed on time, we learned how to set goals, distribute resources wisely, and track our progress.

**Risk Assessment and Mitigation:** By recognizing various risks like resource limitations and technical difficulties, we were able to take proactive measures to reduce them. We reduced project interruptions and kept the project moving forward by creating backup plans and adjusting to new situations.

**Algorithm Understanding:** We explored learning techniques by putting the Random Forest classification algorithm into practice and learned how decision trees can be combined to increase predictive accuracy. We learned about the principles behind Random Forest, such as bootstrapping and feature randomness, which enhance model robustness and generalization.

**Interpersonal Skills Development:**

**Communication and Collaboration:** Working in a team fostered effective communication and collaboration skills. We learned to express ideas clearly, actively listen to teammates' perspectives, and resolve conflicts constructively. Regular meetings and updates facilitated alignment and coordination among team members.

**Leadership and Decision-Making:** Assuming responsibility for assignments and overseeing particular project elements enabled us to improve our leadership abilities. We acquired expertise in forming well-informed judgments, assigning tasks, and inspiring group members to meet objectives.

# **Future Scope:**

**Admin Module Enhancement:**

Add extensive user management features to the admin module so that administrators can manage user permissions, access levels, and system configurations. Incorporate content management, analytics, and reporting features as well to give administrators insightful data on user activity and system performance.

**Model Refinement and Expansion:**

By including more diseases and symptoms in the training dataset, the disease prediction model can be improved even further. We can increase the model's accuracy and predictive power by adding a variety of medical data points over time. This will allow users to receive more accurate disease identification and prognosis.

**Integration of Doctor Modules and Appointment Booking:**

Add specific doctor modules to the website so that medical professionals can examine prediction results, access patient records, and offer tailored advice. Improve the platform by adding the ability for patients to make appointments online with physicians based on anticipated illnesses. This will facilitate easy communication and patient care.

**Integration with remote consultation services:**

Include online consultation features in the platform to enable online health services and remote consultations. Make it possible for patients to obtain medical resources, participate in teleconsultations with medical experts, and receive diagnosis and treatment from a distance. This will increase patient convenience and improve accessibility to healthcare services.

**Optimized User Interaction and Accessibility:**

Focus on enhancing the platform’s user-friendliness, navigation, and interface to elevate the user experience. Employ principles of responsive design to ensure that content is readily available across various screens and devices, catering to a diverse user base. Include features for accessibility and support for multiple languages to cater to users with different language preferences and disabilities.

# **Conclusion:**

From the beginning to the implementation of our project, "Disease Prediction System Using Machine Learning with Amazon SageMaker", we have achieved important goals and provided meaningful advancements to healthcare and predictive analytics.

The major accomplishments of our project are as follows:

* We have effectively deployed a machine learning model on Amazon SageMaker for disease forecasting, harnessing the power of advanced algorithms like Random Forest.
* We developed a user friendly online platform for users to input symptoms and get immediate predictions of potential illnesses.
* We created a scalable and efficient system architecture by incorporating cloud-based services such as S3, EC2, Lambda, and API Gateway.
* We have put in place security protocols and industry standards to safeguard patient information and maintain compliance with healthcare laws.

Future improvements for the project have been outlined, such as adding an admin module, expanding the disease prediction model, integrating doctor modules, and incorporating telemedicine services. This shows our commitment to constantly improving and innovating.

The impact of our project extends further than its technical execution. Our goal is to empower people to take proactive steps in managing their health and well-being by providing a reliable and easily accessible disease prediction platform. Our project aids in the overall goal of utilizing technology to enhance healthcare results, increase patient involvement, and encourage early detection and prevention of illnesses.

Reflecting on our experience, we have acquired valuable knowledge about the challenges of creating machine learning applications in actual situations. From the start of generating ideas to facing technical challenges and improving our solution, we have demonstrated resilience, teamwork, and a strong commitment to leveraging technology for beneficial social change. As we conclude this project, we are committed to enhancing our expertise, capabilities, and impact in the field where healthcare and data science meet.

**REFERENCES**

1. *Fardin Abdali-Mohammadi, M. N. (n.d.). https://www.researchgate.net/profile/Maytham-N-Meqdad/publication/344219772\_Development\_of\_an\_IoT-based\_and\_cloud-based\_disease\_prediction\_and\_diagnosis\_system\_for\_healthcare\_using\_machine\_learning\_algorithms/links/5faa202fa6fdcc0624250d47/Development-of-.*
2. *https://aws.amazon.com/sagemaker. (n.d.). https://aws.amazon.com/sagemaker/getting-started/.*
3. *https://aws.amazon.com/tutorials/run-serverless-code/. (n.d.). https://aws.amazon.com/tutorials/run-serverless-code/.*
4. *https://aws.amazon.com/what-is/api/. (n.d.). https://aws.amazon.com/what-is/api/.*
5. *https://docs.aws.amazon.com. (n.d.). https://docs.aws.amazon.com/sagemaker/.*
6. *https://docs.aws.amazon.com/lambda. (n.d.). https://docs.aws.amazon.com/lambda/latest/dg/welcome.html.*
7. *https://medium.com/. (n.d.). https://medium.com/ds3ucsd/aws-sagemaker-introduction-tutorial-7ea2f28bb67.*
8. *Sanket Jayesh Muchhala, H. S. (n.d.). https://www.researchgate.net/publication/356267168\_Disease\_Prediction\_using\_ML.*
9. *Sireesha Muppala, R. D. (n.d.). https://ieeexplore.ieee.org/document/10162810.*