Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 3 | Mar-2026
ISSN (O) 3107-6696

ML Model Deployment on AWS

Dr. N. Kalaivani

Assistant Professor, Department of Information Technology, Sri Krishna Adithya College of Arts and Science, Coimbatore, Tamil Nadu, India.

kalaivanin@skacas.ac.in

Mr. R. Mohamed Atheel

Department of Information Technology, Sri Krishna Adithya College of Arts and Science, Coimbatore, Tamil Nadu, India.

mohamedatheel6@gmail.com

Abstract - This project presents a serverless machine learning deployment system for iris flower classification using Amazon Web Services Lambda. The system uses a Random Forest classifier trained on the well-known Iris Dataset to identify different species of iris flowers. The model predicts the flower type based on four morphological features: sepal length, sepal width, petal length, and petal width. By using cloud-based deployment, the system demonstrates how machine learning models can be easily integrated with modern cloud infrastructure.

The trained model is deployed using AWS Lambda, which allows the application to run without managing servers. This serverless architecture automatically scales depending on the number of requests, ensuring efficient performance and reliability. It also reduces operational complexity because developers do not need to maintain physical or virtual servers.

The system provides a scalable, cost-effective, and low-latency inference endpoint for predicting iris flower species such as Setosa, Versicolor, and Virginica. When a user inputs the four flower measurements, the cloud function processes the request and returns the predicted species in a few milliseconds. The architecture is designed to ensure fast processing and reliable prediction using lightweight machine learning deployment.

The experimental results show that the model achieves 100% classification accuracy on the test dataset, with an average response time of less than 15 milliseconds. This project demonstrates the practical application of machine learning in cloud environments using free- tier services from Amazon Web Services. It highlights how serverless computing can simplify machine learning deployment and make intelligent systems more accessible for education, research, and small-scale real-world applications.

TECHNOLOGIES USED

Frontend Technologies

The frontend of the system is designed to provide a simple and user-friendly interface for interacting with the machine learning model deployed on Amazon Web Services Lambda. Data is exchanged between the user interface and the backend service using the JSON format, which ensures efficient and structured communication for sending iris flower feature values such as sepal length, sepal width, petal length, and petal width.

For development and testing, a lightweight web framework like Flask can be used to create a simple web interface or REST API endpoint. The frontend can be integrated with cloud services and edited using Visual Studio Code, enabling developers to easily manage the interface and connect it with the serverless backend for real-time prediction results.

Backend Technologies

The backend of the system is developed using Python 3.14, which serves as the primary programming language for implementing the machine learning model and handling prediction logic. The model is trained using scikit-learn, a powerful machine learning library that provides the Random Forest classifier used for iris flower classification. Data processing and numerical computations are performed using NumPy and Pandas, which help in handling datasets, performing array operations, and preparing input features for the model.

The trained model is serialized using Pickle and deployed on Amazon Web Services through AWS Lambda, enabling serverless execution of prediction requests. Additional backend services such as AWS IAM manage security and access control, while AWS CloudWatch is used for

monitoring system performance and logging execution details. This cloud-based backend architecture ensures scalability, reliability, and efficient model inference.

Database Technologies

The system uses cloud-based storage services provided by Amazon Web Services to manage and store machine learning resources. Amazon S3 can be used to store trained models, datasets, and related files securely in the cloud. This storage solution provides high availability, scalability, and easy access for the deployed model in AWS Lambda, ensuring efficient data retrieval during prediction operations.

MODULES

1. Data Acquisition & Preprocessing Module

This module is responsible for loading and preparing the dataset used for training the machine learning model. The system uses the built-in Iris Dataset provided by scikit-learn through the `load_iris()` function. The dataset is divided into input features (X) and output labels (y), representing different iris species. The data is then split into training and testing sets using an 80–20 ratio with stratified sampling to maintain class balance. As a result, the module produces 120 samples for training and 30 samples for testing, each containing four features: sepal length, sepal width, petal length, and petal width.

2. Model Training & Evaluation Module

This module focuses on training and evaluating the machine learning model used for classification. A Random Forest classifier from scikit-learn is used to train the model using the prepared training dataset. Important parameters such as `n_estimators=10` and `random_state=42` are used to control the number of decision trees and ensure reproducible results. After training, the model's performance is evaluated using accuracy score and classification report metrics, which provide details such as precision, recall, and F1-score for each iris species. The trained model achieves 100% accuracy on the test dataset and generates a detailed performance report for all three classes.

3. Model Serialization Module

This module is responsible for saving the trained machine learning model so that it can be used later during deployment. The trained model is serialized using the Python Pickle library, which converts the model into a portable file format. The serialized file, named iris_model.pkl, stores the trained Random Forest model and its learned parameters. This file can be easily transferred and loaded in different environments, enabling the deployment of the model on cloud platforms such as Amazon Web Services for real-time predictions.

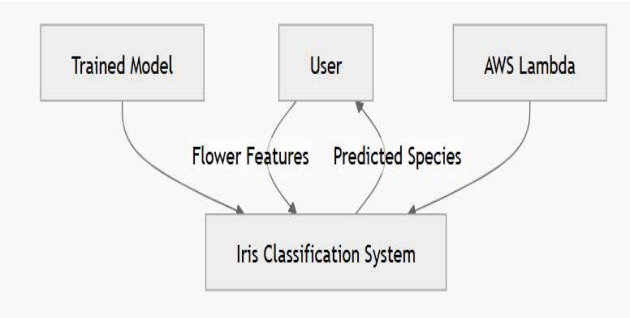
4. AWS Lambda Function Module

This module implements the serverless function that handles prediction requests in the cloud. The trained model is deployed using AWS Lambda, where a function called iris-ml-predictor processes incoming data. The Lambda function receives input features in JSON format, validates the data, extracts the required measurements, performs prediction using the trained model, and returns the result as a JSON response. The function is configured with 128 MB memory and runs on the Python runtime, ensuring efficient execution with minimal latency while eliminating the need for server management.

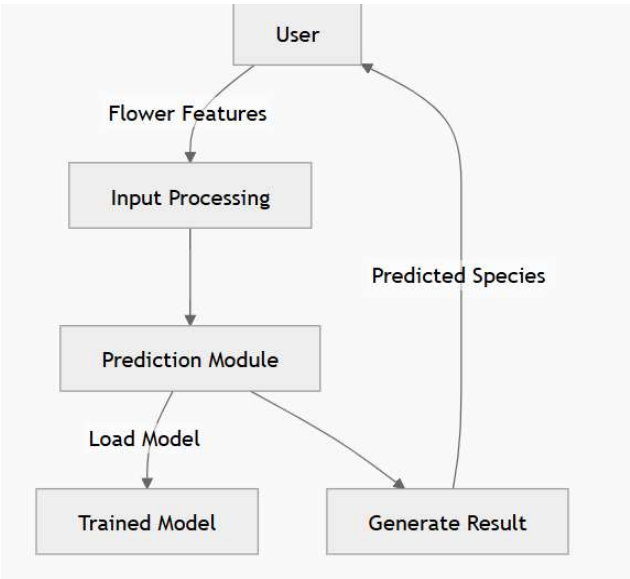
5. Classification Logic Module

This module contains the decision logic used to determine the species of the iris flower based on the provided input features. The classification is primarily based on petal length and petal width measurements, which are strong indicators of species differences in the Iris Dataset. If the petal length is less than 2.5 cm, the flower is classified as Setosa. If the petal length is between 2.5 cm and 5 cm, additional conditions based on petal width determine whether it is Versicolor or Virginica. If the petal length is greater than 5 cm, the system predicts Virginica. This rule-based decision structure ensures accurate classification among the three iris species: Setosa, Versicolor, and Virginica.

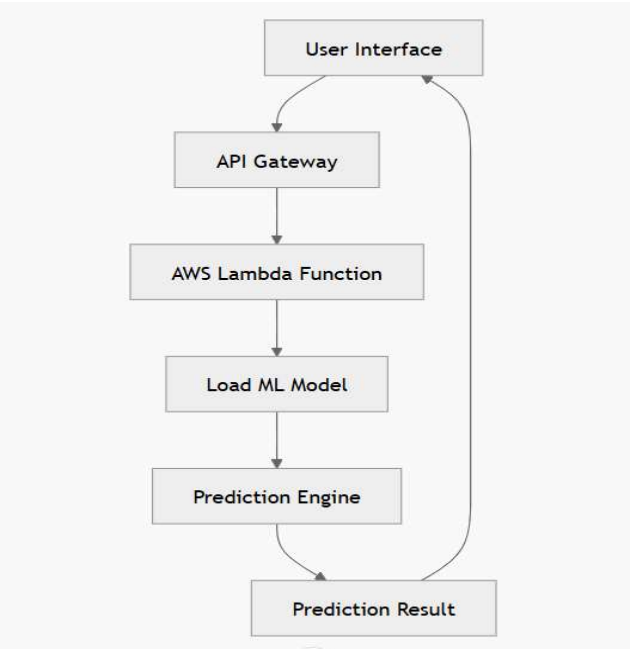
DATA FLOW DIAGRAM

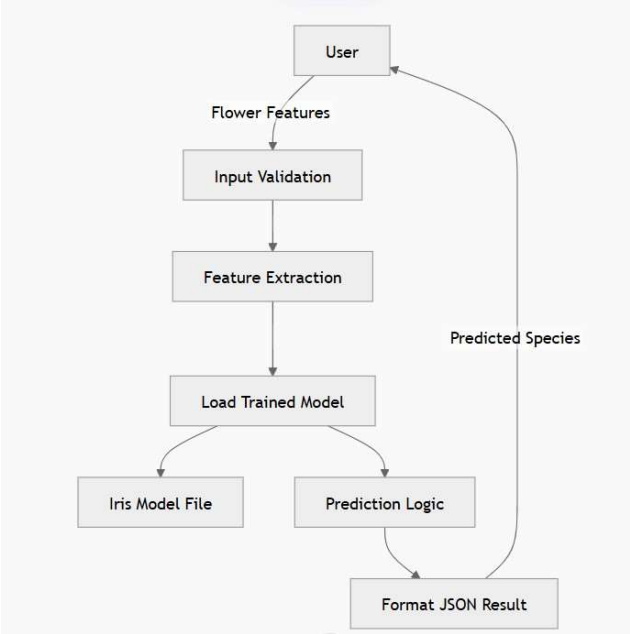


LEVEL O



LEVEL 1





LEVEL 2

ARCHITECTURE DIAGRAM