# AIG 130 – Lab 4 – Group 5

## Group Members:

* Masoud Masoori
* Aliyyah Jackhan
* Mohammed Aadil Suhail Shaikh
* Jonathan Chacko Pattasseril

Table of Contents

[Introduction: 1](#_Toc193036323)

[AWS SageMaker 2](#_Toc193036324)

[Overview & Data Preparation 2](#_Toc193036325)

[Model Training & Deployment 2](#_Toc193036326)

[Google AutoML 3](#_Toc193036327)

[Overview & Data Preparation 3](#_Toc193036328)

[Model Training & Deployment 3](#_Toc193036329)

[Azure ML Studio 4](#_Toc193036330)

[Overview & Data Preparation 4](#_Toc193036331)

[Model Training & Deployment 4](#_Toc193036332)

[Comparative Analysis 5](#_Toc193036333)

## 

## Introduction

**Application:** Symptom-to-Treatment Prediction

This report explores how three cloud services—AWS SageMaker, Google AutoML, and Azure ML Studio—can be used to implement an end-to-end machine learning pipeline for predicting treatments based on user-reported symptoms.

## AWS SageMaker

### Overview & Data Preparation

#### Overview:

* Leverage AWS SageMaker for symptom-to-treatment prediction.
* Store data in Amazon S3 for scalable processing.
* Deploy the model as an API for real-time recommendations.

#### Data Preparation:

* Collect structured health datasets and preprocess them.
* Format data for machine learning tasks.
* Upload the cleaned datasets to AWS S3 for seamless SageMaker access.

### Model Training & Deployment

#### Model Training:

* Use SageMaker AutoPilot for automated feature selection.
* Build custom models using frameworks like TensorFlow or PyTorch.
* Benefit from distributed training to handle large datasets.

#### Model Deployment:

* Deploy models as real-time APIs using SageMaker Endpoints.
* Integrate with AWS Lambda and API Gateway for secure connectivity.
* Monitor performance through CloudWatch and SageMaker Model Monitor.

## Google AutoML

### Overview & Data Preparation

#### Overview:

* Employ Google AutoML to predict treatments from symptoms.
* Utilize diverse data sources including public health data, medical journals, and crowdsourced inputs.
* Deploy models as APIs for real-time prediction services.

#### Data Preparation:

* Gather labeled symptom-to-treatment datasets.
* Format the data into clean, structured CSV files.
* Store datasets in Google Cloud Storage for easy access by AutoML.

### Model Training & Deployment

#### Model Training:

* Choose between AutoML Tables (for structured data) or AutoML NLP (for free-text).
* Upload data to Google Cloud Storage and create models within Vertex AI.
* Define symptoms as inputs and treatments as outputs while AutoML automates feature analysis and optimization.

#### Model Deployment:

* Select the best-performing model for deployment.
* Deploy the model as a real-time API via Google Cloud AI Platform.
* Generate an endpoint that can be integrated with apps or chatbots.

## Azure ML Studio

### Overview & Data Preparation

#### Overview:

* Use Azure ML Studio’s no-code/low-code platform to simplify symptom-to-treatment prediction.
* Offer an end-to-end pipeline that covers data ingestion, training, and deployment.
* Ensure built-in security and HIPAA compliance for healthcare applications.

#### Data Preparation:

* Collect and preprocess symptom-to-treatment datasets.
* Upload data to Azure Blob Storage.
* Normalize and label data to ensure quality input for the model.

### Model Training & Deployment

#### Model Training:

* Apply AutoML within Azure ML Studio to select features and optimize a classification model.
* Train the model using both training and testing sets to validate performance.

#### Model Deployment & Monitoring:

* Deploy the trained model as a real-time REST API using Azure Managed Endpoints.
* Monitor model performance through Azure ML Pipelines and set up retraining triggers for data drift.
* **Strengths:** Easy-to-use, scalable, and secure with integrated HIPAA compliance.
* **Limitations:** Cloud-based pricing can be costly and the platform offers less flexibility for deep customizations.

## Comparative Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **AWS SageMaker** | **Google AutoML** | **Azure ML Studio** |
| **Ease of Use** | Requires some ML and cloud expertise. Extensive setup but powerful customization. | Extremely user-friendly with automated model training. Requires minimal ML knowledge. | Drag-and-drop interface makes it beginner-friendly. Some advanced features require coding. |
| **Flexibility** | High flexibility with support for custom models, various ML frameworks (TensorFlow, PyTorch, etc.), and deep customization. | Limited flexibility as it focuses on AutoML. Works best for standard ML tasks but lacks deep customization. | Moderate flexibility—offers no-code and low-code options but also supports Python and R for advanced users. |
| **Scalability** | Highly scalable, integrates well with other AWS services (Lambda, S3, EC2). Supports large-scale enterprise applications. | Scales well but is more suitable for small to medium-scale applications. Best for quick model deployment. | Good scalability with integration into Azure services. Suitable for enterprises but slightly less seamless than AWS. |