R13306 Senior ML Engineer, Commercial Delivery, Genmab – Technical Assessment

---

Assessment Overview & Instructions:

You have been tasked with building and deploying a machine learning model for a commercial application within the healthcare sector. This assessment will evaluate your proficiency across several key areas, including AI/ML development, cloud infrastructure, and MLOps, specifically within AWS environments.

You'll work with various AWS cloud services, integrate cutting-edge AI technologies, and build scalable, production-ready systems. The tasks require you to demonstrate practical expertise in:

- Developing machine learning models.
- Implementing cloud-based infrastructure.
- Ensuring MLOps and deployment scalability.

Please ensure your work is:

- **Complete and reproducible**: Your solutions should be easy to execute and run as intended.
- **Well-documented**: Provide clear explanations of your approach and steps taken.

The assessment is expected to take **5 to 6 hours**. You can work at your own pace and complete it according to your availability.

Once you've completed the assessment, you'll be given an opportunity to present your work in a **30-minute live session** where you will walk us through your solution. This session will be scheduled once your submission is reviewed and determined to meet the expected quality and standards for the role.

Please share your results in a tangible form, such as:

- A **code repository** (GitHub or GitLab) containing your Python code, Infrastructure as Code (IaC) scripts, or any other relevant work.
- Ensure that your code is structured, well-commented, and easy to understand.

If you have any questions or need clarification on any part of the requirements, please don't hesitate to reach out as soon as possible. [End of instructions].

Assessment Structure:

Task 1: Model Development & Deployment
Objective:
Build and deploy a simple AI model for a customer segmentation use case using AWS SageMaker.

Steps:

1. **Dataset**: Use a publicly available dataset (e.g., customer data from Kaggle or UCI Machine Learning Repository or the one attached here as “customer_segmentation_data.csv”) that contains features like demographic information and historical purchases.
2. **Model**: Choose an appropriate machine learning algorithm (e.g., K-means clustering or logistic regression) to develop a model that segments customers into different groups based on their behavior or demographic features.
3. **Environment**: Set up an AWS SageMaker notebook instance and train your model using the dataset. Implement necessary pre-processing and feature engineering steps.
4. **Deployment**: Deploy the model using SageMaker Endpoints to serve predictions. Ensure it is optimized for low-latency inference.
5. **Documentation**: Provide a brief documentation on the steps followed and how you optimized the model for deployment.

Deliverables:

- A working AWS SageMaker endpoint URL for model predictions.
- Python code for training and deployment.
- Documentation explaining your approach.

Task 2: Infrastructure as Code

Objective:

Automate the infrastructure setup for your model deployment using AWS CloudFormation or Terraform.

Steps:

1. Write an Infrastructure as Code (IaC) script that:
    - Creates an AWS SageMaker endpoint.
    - Sets up an S3 bucket for data storage and model artifact storage.
    - Configures IAM roles for access to SageMaker, S3, and other necessary services.
2. Your IaC script should be able to provision and tear down the environment with minimal manual intervention.

Deliverables:

- Terraform or CloudFormation script for setting up the environment.
- A README file explaining how to deploy and destroy the infrastructure.

Task 3: Generative AI Integration

Objective:

Integrate a Generative AI model (e.g., a Hugging Face transformer model) for generating insights from unstructured text data.

Steps:

1. Choose a pre-trained Generative AI model from Hugging Face or AWS Bedrock (e.g., GPT-2, GPT-3, or other relevant NLP models).
2. Fine-tune the model (if necessary) on a small domain-specific dataset (e.g., sales or market insights).
3. Build a simple API using AWS Lambda that generates text-based insights (e.g., summary, prediction, recommendation) from a provided input text.
4. Ensure the solution is optimized for low-latency, cost-efficient inference.

Deliverables:

- Lambda function or API endpoint to generate insights from unstructured text data.
- Documentation explaining how to fine-tune and deploy the Generative AI model.