Assessment Task: Machine Learning Engineer

Company: Engage
Role: Machine Learning Engineer
Time: 4-5 hours

Introduction

Engage reimagines the property journey. Designed for home seekers, property owners, and brokers, it brings every stakeholder together on one connected platform powered by intelligent technology. By automating regulations, streamlining data, and enhancing transparency, Engage transforms complexity into clarity - creating a seamless experience built on trust, speed, and shared success.

Objective

Design and implement an AI-powered restaurant discovery solution demonstrating LLM expertise, ML engineering, backend skills, and alignment with Engage's values: bias for action, ownership, innovation, and continuous learning.

Context

This assessment simulates a real-world ML engineering scenario. You'll build an end-to-end AI solution that demonstrates your ability to work with modern LLM frameworks, train production-ready ML models, and design scalable backend systems. We've chosen the restaurant discovery domain to assess your technical skills in a neutral context - this is purely an assessment exercise and not related to Engage's core business.

The Challenge

Build a comprehensive AI-powered restaurant discovery system that combines modern LLM capabilities with traditional ML techniques. You'll create a RAG-based semantic search engine, develop a rating prediction model, and design a production-ready backend architecture. This multi-faceted challenge tests your end-to-end ML engineering skills, from research and experimentation to deployment and monitoring.

Instructions

Part 1: LLM-Powered Restaurant Search (60%)

Task 1.1: RAG System Architecture & Implementation (40%)

Build a working RAG-based restaurant search handling queries like "Find Italian restaurants in downtown Dubai with outdoor seating under AED 200 per person."

- Design production-ready architecture (LangChain/LangGraph, vector DB, embeddings, LLM)
- **Implement data pipeline:** ingest 50 restaurants (JSON), generate embeddings, hybrid search
- **Build RAG system:** semantic retrieval, contextual LLM responses, edge case handling
- Create system diagram documenting technology choices and rationale

Task 1.2: Agentic Workflow Design (20%)

Design multi-agent system using LangGraph:

- **Agent 1:** Query understanding & entity extraction (cuisine, location, price, ambiance)
- **Agent 2:** Restaurant retrieval & filtering
- **Agent 3:** Response generation with personalized recommendations
- Include conditional logic, memory management, multi-turn conversations

Values Reflection: Document how you demonstrated bias for action (build vs. plan trade-offs), ownership (decision validation), and innovation (novel approaches, calculated risks).

Part 2: ML Rating Prediction Model (25%)

Task 2.1: Feature Engineering & Model Implementation (25%)

Build a restaurant rating prediction model using the provided dataset (restaurants, reviews, user data, dining trends).

- **Design feature pipeline:** structured data (cuisine, price, location), unstructured (reviews), time-series (trends)
- Implement model with hyperparameter tuning (XGBoost/Random Forest/Neural Networks)
- **Evaluation:** RMSE, MAE, R^2 with validation strategy
- **Document:** model interpretability, drift handling, retraining strategy, production monitoring

Part 3: Backend & Production (10%)

Task 3.1: API & Deployment Design (10%)

- **REST API endpoints:** restaurant search (RAG), rating prediction (ML), health monitoring
- Include schemas, error handling, rate limiting
- **AWS deployment:** SageMaker (ML), Lambda/ECS (APIs), CI/CD pipeline
- **Monitoring:** model performance, API latency, error detection, scalability plan

Part 4: Strategic Scenarios (5%)

Question 1: Product team wants to launch rating prediction in 2 weeks but accuracy is 75% (target: 85%). The design team suggests a "beta" label. How do you balance urgency vs. quality? What's your recommendation?

Question 2: AI search returns poor results for "romantic date night restaurants." Debug systematically, identify root cause (data/model/product), propose fixes (short & long-term), define success metrics.

Question 3: Describe a recent ML project failure/obstacle. What did you learn and how did you apply it? (Assesses: Learn Eternally, Disagree and Commit)

Deliverable

- **Code Repository:** Python code, README, requirements.txt, unit tests
- **Technical Doc (3-4 pages):** Architecture diagrams, design decisions, performance results, deployment plan
- **Demo Video (5-10 min):** Prototype walkthrough, key features, technical challenges

Evaluation Criteria

Technical Skills (70%)

- **LLM (25%):** RAG implementation, LangChain/LangGraph, prompt engineering, agentic workflows
- **ML (25%):** Feature engineering, model selection, evaluation, production best practices
- **Backend (20%):** API design, architecture, deployment, scalability

Competency Framework (20%)

- **Technical Excellence:** Code quality, system design, innovation
- **Delivery & Execution:** Reliable delivery, project management, quality assurance
- **Vision & Strategy:** Technical vision, research approach

Values & Culture (10%)

- **Bias for Action:** Pragmatic prioritization, shipping mentality
- **Ownership:** Decision accountability, trade-off transparency
- **Innovation:** Experimentation, calculated risk-taking
- **Learn Eternally:** Growth mindset, continuous improvement
- **Outcomes Over Outputs:** Business impact focus

Dataset Provided: restaurants.json (50 listings), reviews.csv (1000 reviews), user_data.csv (user preferences), dining_trends.csv

Submission: GitHub repo link + demo video
Timeline: 4-5 hours over 1 week
Contact: moe.reza@cencorpgroup.com

This assessment evaluates technical depth, product thinking, and cultural alignment with Engage's values-driven engineering culture.