A blue and white background

Description automatically generated

AI Automation Developer   
Assessment

2025-02-24

**Objective:**

The goal of this assessment is to evaluate your ability to build a machine learning model and an AI agent in Python. You will also be required to document your process, explaining the steps taken, design choices, and the reasoning behind your approach.

**Instructions:**

1. Complete both **Part A** and **Part B** of the assessment.
2. Write clean, well-structured, and well-commented Python code.
3. Provide a written explanation of your approach for each part.
4. Submit your code and documentation in a single file or notebook.

**Part A: Build a Machine Learning Model Using Case Management System Data**

**Task:**

Develop a machine learning model to predict case outcomes based on data from a **case management system**. The dataset should be structured with at least **three related tables**.

**Dataset Requirements:**

If you do not have access to a real case management system, generate a synthetic dataset that mimics real-world case management data. The dataset should include:

1. **Cases Table**: Contains case details (case ID, case type, status, assignee, priority, creation date, resolution time, description, notes, outcome)
2. **Clients Table**: Contains client details (client ID, name, age, risk level, previous cases, etc.).

**Requirements:**

* **Data Preprocessing**: Merge/join tables appropriately, handle missing values, normalize/scale features, and encode categorical variables if needed.
* **Model Selection**: Train a predictive model (e.g., decision tree model) to predict **case resolution status** (e.g., resolved, pending, escalated).
* **Evaluation**: Assess model performance using accuracy, precision, recall, and F1-score.
* **Feature Importance Analysis**: Identify the most influential factors affecting case resolution.

**Bonus Points:**

* Use SQL or Pandas to extract and preprocess the data efficiently.
* Implement a **time-based** train-test split (e.g., train on older cases, test on newer ones).
* Deploy the model as an API using Flask or FastAPI.

**Part B: Develop an AI Agent for Case Insights**

**Task:**

Create an **AI-powered assistant** that provides useful insights into the case management system data. The assistant should be able to analyse case trends, identify key risk factors, and generate summary statistics.

**Requirements:**

* The AI agent should be implemented as a function or class that interacts with the case management data.
* The assistant should be capable of answering **key business questions**, such as:
  + What are the most common types of cases?
  + What factors contribute most to case resolution time?
  + Which assignees have the highest case resolution rates?
  + Are there patterns in case escalations based on client demographics or case types?
* Use **NLP techniques** to analyse notes or case descriptions.
* Provide a written explanation of the approach and implementation.

**Bonus Points:**

* Implement a chatbot interface (e.g., using LangChain or OpenAI's API) that allows users to query the case management system in natural language.

**Submission Requirements:**

* Submit all Python scripts or a Jupyter Notebook (.ipynb) containing your code.
* Include a **README or Markdown** file with explanations of your approach and steps taken.
* Ensure your code is well-commented and modular.