AI Labs Case Study Assessment for Senior AI Engineer Candidates

Summary

This document discusses details of the technical assignment tasked to a candidate as a prerequisite task for AI engineering qualification.

Author: AI Labs

Last updated: 2026-01-07

Overview

In testing Senior AI Engineer candidates, Stratpoint Labs want to assess their capability to:

- Apply theories and technologies that they are already experts in or familiar with to solve practical problems.
- Independently study, learn, and employ novel concepts to design AI solutions.

In order to assess these, each candidate must take a programming exam that tests their Python programming, software development, and solution architecture design abilities while also challenging their AI theory, application, and solutioning capabilities.

The exam has the following specifications:

- Must be developed in the Python programming language using Machine Learning and AI frameworks, models, and tools supported in Python
- The exam consists of 2 sub-exams with each focusing on a specific use-case and testing very specific AI capabilities
- The exam will be take-home and must be accomplished in 1 week, with the option of extending to 2 weeks if needed. The deliverables are:
    - Working code uploaded to an online repository such as Github
    - A short slide deck detailing the work done, results, and any discussions
    - Documentation on any supporting material that the candidate used to solve the problems
- LLMs, copilots, assistants, and any other AI tool use is allowed, but please document how and where you used them
- Solutions will be presented to the hiring manager(s) at the end of the 2 weeks via scheduled video-call

- 1 hour will be allocated to the entire presentation allocated as follows:
  - Exam 1 Presentation: 10 minutes
  - Exam 2 Presentation: 10 minutes
  - Combined Q&A: 40 minutes
- Each presentation is expected to have the following:
  - Problem statement/needs addressed
  - General approach and solution/system diagram
  - Assumptions and limitations
  - Methodology
    - Data preparation
    - Tools and frameworks
    - Algorithms and models
    - System flow
  - Results, discussions, and recommendations

Challenge 1 - Predicting Solar Output (DataScience, LLMs)

Background: The first exam is about figuring out what the likely solar power yield will be for a given area experiencing certain weather conditions. This is an extremely important question as, unlike fossil fuel or nuclear power, solar power is highly susceptible to environmental conditions and fluctuating output affects the power stability of the regions and communities served by solar power farms. It is thus important to know what the expected yield is from a farm in order to be able to prepare and deploy mitigation strategies if needed.

Tasks:

- Prepare **tabular solar power output data** (using Rasterio/GDAL) from Australia's daily solar data sourced from *Global Solar Atlas.*
  - Image manipulation on the solar TIFF files needs to be performed to extract the solar values per city included in the weather dataset
  - For a given city, town, or area, use the average of the solar values for the area covered by the city to estimate solar output per unit area.
- Create a **predictive model** that uses the Kaggle Australia Daily Weather Data as regressors to predict the probable solar output.
  - The key challenge is predicting the solar power output for a given time period and area given the expected weather.

    - For simplicity's sake, the candidate will only be *required* to use only 1 year from the dataset as training and testing data as historical yearly solar data is not available. Assume that the extracted solar data matches the chosen year of the weather data. The candidate should specify which year they used as the basis.
    - Alternatively, the candidate can choose to use an aggregate form of the weather data as their regressors.
- Use an **LLM-powered agent** to answer solar output-weather based questions such as *"what is my expected yield tomorrow given the following probable weather conditions ..."* or "*If I set up a solar farm in this city what's my expected daily output for a farm of size...*". These can be answered with the model or using basic statistical techniques.
    - The candidate should come up with *at least* 5 questions that the agent can answer with a high degree of accuracy, supported by the candidate's own analysis of the question.
    - As the expected weather is the key regressor for prediction, the candidate is free to address how the questions with missing requirements are handled. The candidate should explicitly state in their presentation how this is handled. Some examples include:
        - Explicitly asking the user for the expected weather.
        - Assume that the user is talking within the context of the *last year* of the weather data and use that weather data as input to the model. Of course this means that the last year should not have been used to train the model.
        - Let the LLM infer the *expected weather* for that time period (assume that the user is not talking about any year with historical weather data)
    - During the presentation, the reviewing panel can ask the candidate to prompt the agent for answers to other questions.

Challenge 2 - Extracting Data From Receipts (Computer Vision, LLMs)

Background: Business use official receipts and invoices in numerous processes including accounting, reimbursement, supply-and-demand modeling, etc. However, manually processing these invoices is time consuming and tedious, thus it would be advantageous to automate at least the extraction of data from these documents.

Tasks:

- Create an **OCR (optical character recognition) pipeline** and perform **OCR** on the documents found in the SROIE V2 Dataset
- Create a small application using **Gradio** and allows a user to upload files for OCR and displays the retrieved data using a question-and-answer process:
    - The candidate can use any OCR model and/or any multimodal LLM in their application, as well as any other information rectification steps.
    - The candidate should also use an LLM to aid in the following:
        - Rectification of malformed entities (e.g. *sollor* will be corrected to *seller*)
        - Completion of compound entities with missing components (e.g. complex vendor names that are missing components will be completed)
        - Any other form of information correction. All such processes should be noted by the candidate during their presentation.
- The extracted texts in the dataset should be used to evaluate the performance of the agent in terms of:
    - Number of entities extracted
    - Correctness of extracted entities
- The candidate is expected to present the performance of the OCR pipeline using the *train* and *test* components of the dataset
    - Even though the dataset has explicit *train* and *test* partitions, the candidates WILL NOT be expected to train their own OCR model
    - Instead, the candidate will compare their extracted entities with the *ground truth* from the combined *train* and *test* data
    - The candidate should note the performance of their pipeline from the following standpoints:
        - Raw OCR
        - Improved OCR using multimodal LLM
        - Raw OCR + entity analysis LLM
        - Improved OCR using multimodal LLM + entity analysis LLM
