AIAP® Batch 21 Technical Assessment

Deadline: **1900 hrs, 17th Nov 2025**

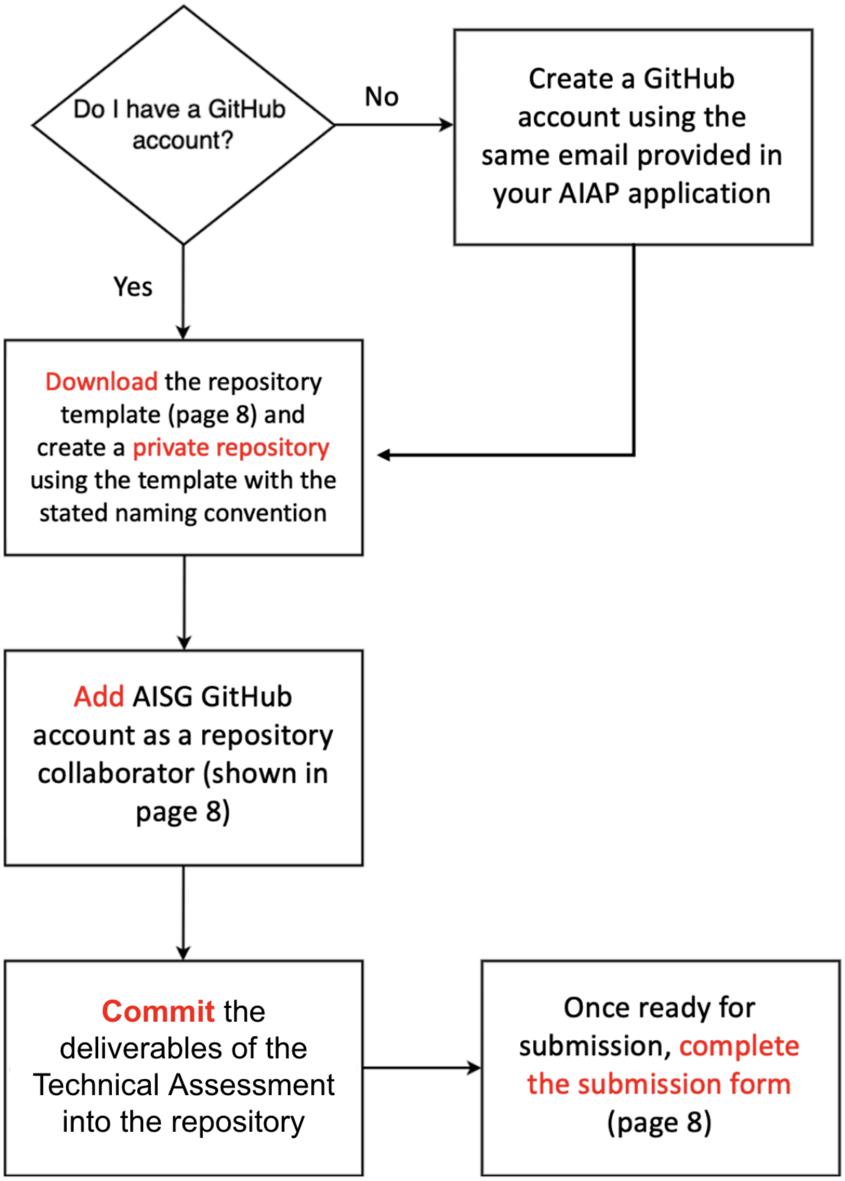
Tasks

This assessment consists of two parts:

1. **Exploratory Data Analysis in Jupyter Notebook**
2. **End-to-end Machine Learning Pipeline in Python Scripts (`.py`)**

Technical Assessment Overview

There are two parts to the Technical Assessment: Exploratory Data Analysis and End-to-end Machine Learning Pipeline. You are to attempt both parts and submit the deliverables by uploading them to your own **private** GitHub repository. The following flowchart outlines the major steps for the Technical Assessment. Details will be provided in the subsequent sections of this document.


Do I have a GitHub account?
No
Create a GitHub account using the same email provided in your AIAP application
Yes
Download the repository template (page 8) and create a private repository using the template with the stated naming convention
Add AISG GitHub account as a repository collaborator (shown in page 8)
Commit the deliverables of the Technical Assessment into the repository
Once ready for submission, complete the submission form (page 8)

Task 1 - Exploratory Data Analysis (EDA)

Using the dataset specified in the **Dataset** section at page 6, conduct an EDA and create an interactive notebook (.ipynb file) in **Python** that can be used as a presentation to explain the findings of your analysis. It should contain appropriate visualizations and explanations to assist readers in understanding how these elaborations are arrived at and their implications.

Deliverable

1. Jupyter Notebook in **Python**: a \`.ipynb\` file named **\`eda.ipynb\`**. (do adhere to the naming requirement)

Evaluation

In the submitted notebook, you are required to

1. Outline the steps taken in the EDA process
2. Explain the purpose of each step
3. Explain the conclusions drawn from each step
4. Explain the interpretation of the various statistics generated and how they impact your analysis
5. Generate clear, meaningful, and understandable visualizations that support your findings
6. Organize the notebook so that is it clear and easy to understand

Please note that your submission will be heavily penalized for any of the following conditions:

1. \`.ipynb\` missing in the submitted repository
2. \`.ipynb\` cannot be opened on Jupyter Notebook
3. Explanations missing or unclear in the submitted Jupyter Notebook

Task 2: End-to-end Machine Learning Pipeline

Design and create a machine learning pipeline (MLP) in Python scripts (`.py` files) that will ingest and process the entailed dataset, subsequently, feeding it into the machine learning algorithm(s) of your choice.

Do not develop your MLP in an interactive notebook.

The pipeline should be easily configurable to enable easy experimentation of different algorithms and parameters as well as ways of processing data. You can consider the usage of a config file, environment variables, or command line parameters.

Within the pipeline, data (provided in the Dataset section, Page 6) must be fetched/imported using SQLite, or any similar packages.

Deliverables

1. A folder named `**src**` containing Python modules/classes in `.py` format.
2. An executable bash script `**run.sh**` at the base folder of your submission to run the aforementioned modules/classes/scripts. DO NOT install your dependencies in the `run.sh`; this will be taken care of automatically when we assess the assignment if you have created your `requirements.txt` correctly.
3. A `**requirements.txt**` file in the base folder of your submission.
4. A `**README.md**` file that sufficiently explains the pipeline design and its usage. You are required to explain the thought process behind your submitted pipeline in the README. The README is expected to contain the following:
    a. Full name (as in NRIC) and email address (stated in your application form).
    b. Overview of the submitted folder and the folder structure.
    c. Instructions for executing the pipeline and modifying any parameters.
    d. Description of logical steps/flow of the pipeline. If you find it useful, please feel free to include suitable visualization aids (eg, flow charts) within the README.
    e. Overview of key findings from the EDA conducted in Task 1 and the choices made in the pipeline based on these findings, particularly any feature engineering. Please keep the details of the EDA in the `.ipynb`. The information in the `README.md` should be a quick summary of the details from `.ipynb`.
    f. Describe how the features in the dataset are processed (summarized in a table).
    g. Explanation of your choice of models for each machine learning task.
    h. Evaluation of the models developed. Any metrics used in the evaluation should also be explained.
    i. Other considerations for deploying the models developed.

Evaluation

The submitted MLP, including the `README.md`, will be used to assess your understanding of machine learning models/algorithms as well as your ability to design and develop a machine learning pipeline. Specifically, you will be assessed on

1. Appropriate data preprocessing and feature engineering
2. Appropriate use and optimization of algorithms/models
3. Appropriate explanation for the choice of algorithms/models
4. Appropriate use of evaluation metrics
5. Appropriate explanation for the choice of evaluation metrics
6. Understanding of the different components in the machine learning pipeline

In your submitted Python scripts (`.py` files), you will be assessed on the quality of your code in terms of reusability, readability, and self-explanatory.

Please note that your submission will be penalized for any of the following conditions:

1. Incorrect format for `requirements.txt`
2. `run.sh` fails upon execution
3. Poorly structured `README.md`
4. Disorganized code that fails to make use of functions and/or classes for reusability
5. MLP not submitted in Python scripts (`.py` files), including MLP built using Jupyter Notebooks.

Note for Windows users

DO NOT submit a Windows batch (`*.bat`) script in replacement of the bash script. Use either ‘Windows Subsystem for Linux (WSL)’ or ‘Git Bash’/’cygwin’ for the creation of the bash script.

Problem Statement

Objectives

As a new hire to the AI Health Analytics team at ElderGuard Analytics, you have been entrusted with a critical task: predicting potential health emergencies or concerning situations for elderly residents living independently based on environmental sensor data and indoor air quality monitoring. Your goal is to help the company develop a non-invasive early warning system that can identify when elderly individuals may be experiencing distress, medical episodes, or unsafe living conditions.

ElderGuard Analytics has collected extensive data from smart home monitoring systems deployed in elderly residents' homes, including information about indoor air quality (CO2 Sensors, CO Sensor, Metal Oxide Sensors), environmental conditions (temperature, humidity), HVAC system operations, and lighting patterns throughout different data collection sessions.

By leveraging this data, you will develop predictive models that can enhance care coordination, identify patterns of concern, and provide valuable insights to family members and healthcare providers. The success of your predictions will directly impact the company's ability to provide proactive care monitoring and maintain residents' independence longer.

Specifically, your task is to build and evaluate prediction models, and also identify their respective key features of the dataset that categorise the `Activity` level of elderly residents. Your analysis should also include evaluation of which features contribute most significantly to activity level prediction.

In your submission, you are expected to build and evaluate **at least three suitable models** for this task and justify your choices based on the dataset provided.

Dataset

The dataset contains environmental sensor readings and smart home monitoring data collected from elderly residents' homes by the CareGuard Technologies monitoring systems. It includes features such as temperature, humidity, gas concentrations (CO2 Sensors, CO Sensor, Metal Oxide Sensors), HVAC operations, lighting conditions, and activity classifications across various data collection sessions.

Important Note: The dataset may contain synthetic or contaminated data. Therefore, you should state clearly any assumptions or justifications that you make in processing the data, including handling of outliers, missing values, and data quality issues.

You can query the datasets using the following URL:
https://techassessment.blob.core.windows.net/aiap21-assessment-data/gas_monitoring.db

Instructions for querying the database

The dataset can be accessed through the `gas_monitoring.db`. You may find either of the following packages, `SQLite` or `SQLAlchemy`, useful for accessing this database.

You should place the `gas_monitoring.db` file in a `data` folder. Your machine learning pipeline should retrieve the dataset using the relative path `data/gas_monitoring.db`.

DO NOT upload the `gas_monitoring.db` onto your GitHub repository.

List of Attributes

Attribute	**Description**
Time of Day	Time period when measurements were recorded.
Temperature	Indoor air temperature measured in degrees Celsius by environmental sensors installed in the resident's living space.
Humidity	Relative humidity percentage of indoor air indicating moisture levels.
CO2_InfraredSensor	Carbon dioxide concentration readings from an infrared sensor.
CO2_ElectroChemicalSensor	Carbon dioxide concentration readings from an electrochemical sensor.
MetalOxideSensor_Unit1	Metal oxide sensor reading for detecting various gases and volatile organic compounds.
MetalOxideSensor_Unit2	Metal oxide sensor reading for detecting various gases and volatile organic compounds.
MetalOxideSensor_Unit3	Metal oxide sensor reading for detecting various gases and volatile organic compounds.
MetalOxideSensor_Unit4	Metal oxide sensor reading for detecting various gases and volatile organic compounds.
CO_GasSensor	Carbon monoxide concentration levels.
Session ID	Unique identifier for each monitoring session.
HVAC Operation Mode	Current heating, ventilation, and air conditioning system operating mode.
Ambient Light Level	Indoor lighting conditions.
Activity Level	Classified resident activity status based on environmental indicators.

Submission Format

Create a GitHub account using the **same** email provided in your AIAP application form. Download the repository template from:
https://techassessment.blob.core.windows.net/aiap21-assessment-data/aiap21-NAME-NRIC.zip

The downloaded repository template contains a hidden folder: \`.github\`. The \`.github\` folder contains scripts to execute your end-to-end machine learning pipeline using GitHub Actions. Specifically, it will first install the required dependencies using your \`requirements.txt\` and subsequently, execute your bash script (\`run.sh\`). You can manually trigger the pipeline under Actions in your repository.

Using the downloaded template, create a **private** repository using the following naming convention:

aiap21-<full name (as in NRIC) separated by dashes>-<last 4 characters of NRIC>

For example, \`*aiap21-john-lim-der-hui-321A*\`. Candidates who do not adhere strictly to this naming convention may be penalized for their code submission.
Ensure your intended code submission is in the **main** branch of your remote Github repository.

Add the following account as a collaborator in your private repository:

- Username: **AISG-AIAP**
- Email: **aiap-internal@aisingapore.org**

Your repository is to have the following structure:

~~~
```
├──── .github
├──── src
│     └──── (python files constituting the end-to-end ML pipeline in .py format)
├──── README.md
├──── eda.ipynb
├──── requirements.txt
└──── run.sh
```
~~~

We encourage you to adhere to Git best practices. Once your repository is ready for submission, complete the following form at https://forms.gle/ZubmvhhF29azDaLY7

NOTE: During the assessment period, you are still allowed to make changes to your repository if you submit the Google form before the deadline.
After the deadline, do **not** make any further changes to your code repository. Candidates who do not adhere strictly to this instruction can be penalized for their code submission.