AIIP® Batch 4 Technical Assessment

Deadline: **1900 hrs, 14 October 2024**

Tasks

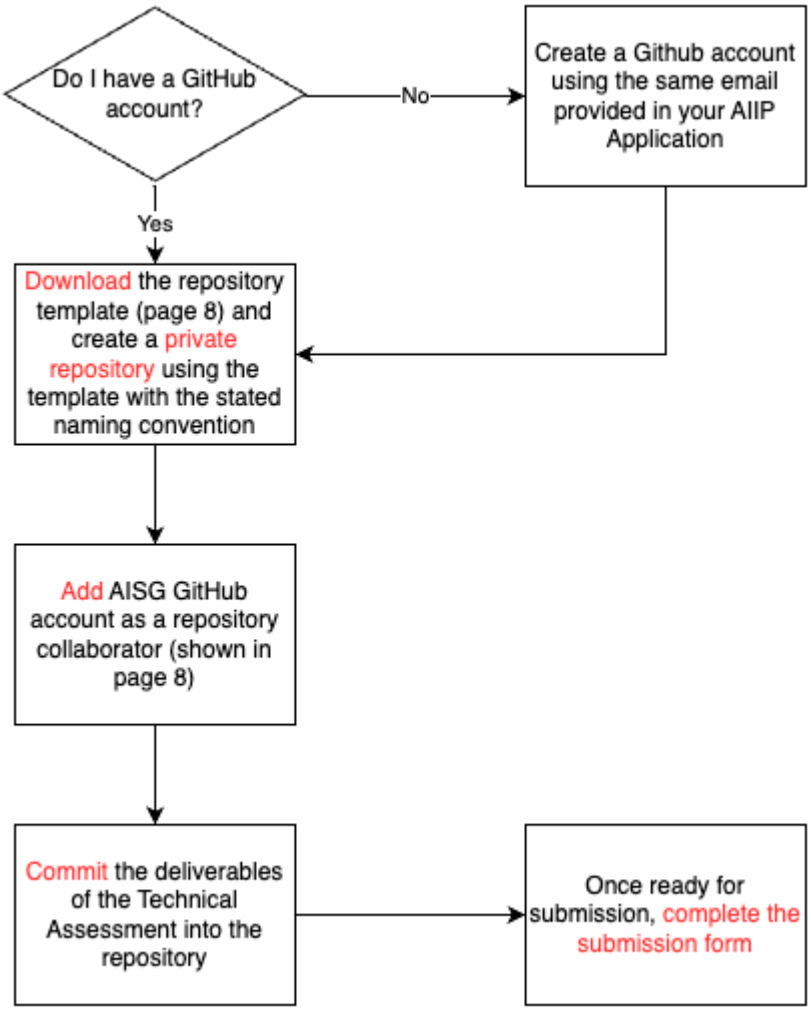
This assessment consists of two parts:

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

AI Singapore
innovation 4.0, 3 Research Link #02-04 Singapore 117602 | *www.aisingapore.org*

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.



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 visualisations 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`**. (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 visualisations that support your findings
6. Organise the notebook so that is it clear and easy to understand

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

1. `.ipynb` missing in the submitted repository
2. `.ipynb` cannot be opened in 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 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 machine learning pipeline 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 visualisation 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 (summarised 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 machine learning pipeline, 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 penalised for any of the following conditions:

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

Maximising solar power generation involves strategically managing operational and manpower costs, particularly by leveraging forecasted weather data to optimise operational planning. During high-efficiency days, which are influenced by favourable weather conditions, increasing battery capacity allows for the storage of excess power generated, ensuring a steady supply even during less optimal periods. Investing in advanced energy storage solutions and grid integration technologies is crucial. Conversely, low-efficiency days provide an ideal window for scheduled maintenance. By taking some panel arrays offline for cleaning and repairs during these periods, downtime is minimised, and overall system performance is maintained. Efficient management of these aspects ensures maximum power generation and reliability of solar power systems.

As an Artificial Intelligence Engineer at SolaraTech, your primary responsibility is to develop models that classify solar panel efficiency as 'Low', 'Medium', or 'High'. By leveraging historical same day forecasted weather data, you will implement predictive algorithms that identify and learn from patterns associated with varying efficiency levels.

These models will be instrumental in strategically managing the company's operational and manpower costs, optimising energy production on high-efficiency days and scheduling maintenance during low-efficiency periods. Your work will ensure that SolaraTech maximises power output while maintaining cost-effectiveness and reliability.

In your submission, you are expected to evaluate **at least three suitable models** for predicting the day's efficiency based on historical forecasted data.

Dataset

The two dataset provided (weather & air quality) contains information forecasted weather data compiled from various meteorological agencies and air quality data.

Note: There could be synthetic features in the dataset. Therefore, you would need to **state and verify any assumptions that you make**.

You can query the datasets using the following URL:
https://techassessment.blob.core.windows.net/aiip4-assessment-data/weather.db
https://techassessment.blob.core.windows.net/aiip4-assessment-data/air_quality.db

Instructions for setting up SQLite and querying the database

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

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

DO NOT upload the `weather.db` & `air_quality.db` onto your GitHub repository.

List of Attributes (weather.db)

Attribute	Description
data_ref	A unique alpha-numeric entry generated by computer
date	The date when the data was recorded
Wet bulb temperature	Temperature incorporating humidity effect.
Daily Rainfall Total (mm)	Total rainfall recorded in a day, in millimeters.
Highest 30 Min Rainfall (mm)	Maximum rainfall in any 30-minute period of the day, in millimeters.
Highest 60 Min Rainfall (mm)	Maximum rainfall in any 60-minute period of the day, in millimeters.
Highest 120 Min Rainfall (mm)	Maximum rainfall in any 120-minute period of the day, in millimeters.
Min Temperature (deg C)	Minimum temperature of the day, in Celsius.
Maximum Temperature (deg C)	Highest temperature recorded on the day, in Celsius.
Min Wind Speed (km/h)	Minimum wind speed of the day, in km/h.
Max Wind Speed (km/h)	Maximum wind speed recorded on the day, in km/h.
Sunshine Duration (hrs)	Duration of sunshine in hours for the day.
Cloud Cover (%)	Percentage of the sky covered by clouds.
Relative Humidity (%)	Average relative humidity for the day, in percentage.
Air Pressure (hPa)	Atmospheric air pressure, in hectopascal (hPa).
Dew Point Category	Category of dew point temperature conditions.
Wind Direction	General direction from which the wind is coming.
Daily Solar Panel Efficiency	Efficiency rate of solar panels for the day.

List of Attributes (air_quality.db)

Attribute	Description
data_ref	A unique alpha-numeric entry generated by computer
date	The date when the data was recorded
PM25 North	Particulate matter of 2.5 micrometers in the north.
PM25 South	Particulate matter of 2.5 micrometers in the south.
PM25 East	Particulate matter of 2.5 micrometers in the east.
PM25 West	Particulate matter of 2.5 micrometers in the west.
PM25 Central	Particulate matter of 2.5 micrometers in the central area.
PSI North	Pollutant Standards Index in the north.
PSI South	Pollutant Standards Index in the south.
PSI East	Pollutant Standards Index in the east.
PSI West	Pollutant Standards Index in the west.
PSI Central	Pollutant Standards Index in the central area.

Submission Format

Create a GitHub account using the **same** email provided in your AIIP application form.

Download the repository template from:
https://techassessment.blob.core.windows.net/aiip4-assessment-data/aiip4-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:

aiip4-<full name (as in NRIC) separated by dashes>-<last 4 characters of NRIC>

For example, **`*aiip4-john-lim-der-hui-321A*`**

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 submission form at: https://forms.gle/59Kdjv4FRVxHZbq96**

NOTE: During the assessment period, you are still allowed to make changes to your repository after submitting the form.