**Problem Context**

The goal is to be able to identify engine failure before it happens. For this purpose, you will be provided information about the lifecycle of a few engines along with engine context in terms of sensor readings and operation settings measured at different points through an engine's lifecycle.

**Data**

The dataset consists of sensor readings taken from 27 sensors at 3 different operation settings during the lifecycle of an engine upto the point of failure. Each row in the dataset represents a set of sensor readings for a given engine. At every point, we also know the number of cycles the engine ran from that point before failing. This information is represented in the training data using the following columns:

* engine\_no: unique engine ID
* times\_in\_cycles: the number of cycles from start of engine life at which the current reading is being taken
* op\_setting\_\*: the value of operating setting 1/2/3
* sensor\_\*: value of the reading for sensor 1/2/3/.../27

The dataset is split into train data and test data.

* train dataset: the engine runs until failure, i.e. the last entry in the data for a given engine is that point before which it failed
* test dataset: The engine runs until a certain point, but it didn't necessarily fail in the next cycle.

Things to Note:

* For every engine, the maximum value of times\_in\_cycles is the total number of cycles it ran before failing. For eg, if engine 1 has 10 entries, then it failed on the 11th cycle. This will lead to different number of samples in the data per engine.
* the engine number in train and test are not the same.

Follow the [readme](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/data/README.md) for instructions to setup your data.

**Objective**

The goal of the test is to determine the failure of an engine in advance. This has to be done in 2 ways:

**Part 1: Exploratory Analysis**

You need to identify atleast 3 flags/markers which can help identify an engine failure in advance. These would act as early warning signals. They could be something like "After X cycles, if the value of sensor A, B, C follows this pattern then there is higher than average likelihood that the engine will fail in the next 100 cycles."

You need to:

* List down atleast 10 hypothesis of interest
* Test some of them so that you are able to find atleast 3 flags

You are encouraged to share all the hypothesis which you tested, irrespective of whether it failed or not. It is okay for a hypothesis to fail, that is why it is called a hypothesis. Don't shy away from sharing it, it is part of the process :)

**Part 2: Predictive Modeling**

Once you have identified and validated some hypothesis, you need to convert them into features and train a predictive model. For every engine in the test set, you need to predict whether the engine will fail in the next 100 cycles. To clarify, the test set has more than 100K observations but only 707 unique engine IDs and we expect 1 prediction per ID. For eg, engine no 1 has 186 time cycles in the test set. We need to predict whether the engine will fail somewhere in between the 187th and 206th cycle.

A sample\_submission.csv file is provided as a template, your submission should follow the same format but change the value of probability\_failure column.

**Expected Outcomes**

We expect to have 2 final outcomes:

* Part 1: A notebook showing the analysis done for [part 1 of the objective](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n#part-1-exploratory-analysis)
* Part 2: One or more predictor files which implement [base.py](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/src/base.py). We will use the true labels on the test set to evaluate model performance. You can use the [sample\_predictor.py](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/src/sample_predictor.py) script to get a dummy example of how to extend the base class.

Note:

* We would rerun all your code notebook + script, so make sure the code is clean and environment setup works. There are more instructions to follow about the environment setup below.
* You are encouraged to commit your final submission file in the repo itself just in case there are bugs in the code.

Don't forget to look at the [Getting Started](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n#getting-started) section for some useful starter code.

**Evaluation Criteria**

* Part 1:
  + thought process around generating and evaluating hypothesis.
  + visualisations could be interesting but not necessarily required.
* Part 2:
  + how you convert your learnings from part 1 to create an ML model
  + evaluation setup for judging models
  + code quality while converting the final model from notebook into the predictor script.
  + We do NOT expect you to build the most accurate model. our recommendation is to:
    - build a first model, even if just slightly better than baseline
    - complete the remaining steps to get a submission file
    - if time permits, get into model improvement techniques like hyperparameter tuning

**environment setup**

The test setup is designed to be easily replicable on any machine which clones the repository. To facilitate this, please follow the given steps to setup your local environment correctly.

**python env**

We strongly recommend using a virtualenv with python 3.7.9 installed. The code will be tested using the same python version, you could deviate from this but at your own risk.

During test evaluation, the environment will be setup using a virtual environment created using pyenv as:

pyenv virtualenv 3.7.9 axion-ml-test

It is not mandatory to use pyenv, you can use your favorite python virtual environment manager.

**requirements.txt**

We will use a requirements file for managing dependencies to make sure the code can be run cross-platforms. For this, a standard requirements.txt file will be used. We have already provided some common python packages typically used for similar work. You could simply use one of the following command to setup for virtual environment:

If you're setting up for the first time:

pip install -r requirements.txt

The line above will not remove any additional packages that you might have installed but are not in the requirements.txt. It is recommended to use pip-sync to make sure no dependencies are missing in the requirements file.

pip-sync requirements.txt

You are free to modify (add/remove) the requirements as you see fit. To make sure the exact versions of python packages are installed, we have used [pip-tools](https://pypi.org/project/pip-tools/1.8.0/). You are expected to follow the same regime if you modify the requirements.

If you are not familiar with pip-tools, at a high level there are 2 steps required to add dependencies:

1. add the required packages to requirements.in file; try to be flexible here and not hard-ping the dependency unless absolutely essential
2. run pip-compile requirements.in --output-file=requirements.txt to create a requirements.txt file with hard-pinned versions of packages. pip-tools will make sure that there are no conflicting dependencies, this adds additional confidence on replicating the environment.

**Getting Started**

A few resources have been provided for you to get started:

1. [load\_data.py](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/src/load_data.py) contains helper code to read data from the data dir once copied
2. [base.py](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/src/base.py) provides a base class which your final model should implement
3. [sample\_predictor.py](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/src/sample_predictor.py) a sample implementation which you could use as a reference for your model
4. [test\_predictor.py](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/test_predictor.py) this script can be used to make sure your implementation of base.py is generating output in the correct format. by default, it checks the sample implementation. you are encouraged to use this before making the final prediction.
5. [starter.ipynb](https://github.com/Axion-Ray/ml-take-home-test-anchhabra1305n/blob/main/notebooks/starter.ipynb) a sample notebook with data imported from the data dir. this is where you have all the free reign to do your data analysis.