JM0100: Prescriptive Algorithms
Assignment 3: Group project
2024-2025

Instructions

- Use Python packages to solve the tasks. We recommend DEAP for the optimization task.
- Deadline for submitting the solutions is **Friday, 16 May, 18:00**.
- The assignment consists of two parts: a **Prediction Task**, and an **Optimization Task**.
- At the end of the Prediction and Optimization Tasks, you can find some hints. Read everything before you start.
- On the last page, you can find some information on grading.

Deliverables

Submit all solutions electronically through Canvas. Make sure that you submit everything in a single zip file that contains the solutions to the project, described below:

- Submit a pdf file D-XXX.pdf (where XXX is your group number) that contains your answers to the questions.
- In addition, submit the Python or Jupyter notebook file (D-XXX.py or D-XXX.ipynb) that you created for the project.
- Stick to the page limit! You can write *at most 2 pages* for the Prediction Task, and *at most 3 pages* for the Optimization Task.
- Make sure you include a cover page with the team information, i.e., names and student id of the team members.

Name your zip file D-XXX.zip, where XXX is your group number.

Good luck!

Introduction

In this assignment, we will consider a maintenance scheduling problem for airplane engines. Engines are subject to wear and tear over time. A key quantity of interest for machines like engines is called the remaining useful lifetime or RUL for short. In this assignment, you will develop a predictive model that predicts the RUL for engines that are currently in use. Using an engine beyond its RUL is not safe and therefore there are costs associated with this. The only way to ensure that an engine is safe for use is to perform maintenance on it. Given an estimate of the RUL, company X needs to decide on a maintenance schedule so that it can maintain the engines in a timely fashion in order to avoid the aforementioned costs.

You are provided with the following files:

1. `DataTrain.csv`. This is a csv-file that contains data for 100 engines. You will use this data to develop your predictive model. We will refer to this data as *dataset 1*. This dataset contains run-to-failure data for a number of engines.

   Engines in *dataset 1* are considered to start with various degrees of initial wear but are considered healthy at the start of each record. As the number of cycles increases the engines begin to deteriorate until they can no longer function. At this point in time (when the engines can no longer function) the engines are considered unhealthy and cannot perform their intended function. A description of the variables in this dataset is given in Table 1.

   Table 1: Dataset description for *dataset 1*.

Column index	Data field	Type	Description
1	engine_id	Integer	aircraft engine identifier, range [1, 100]
2	cycle	Integer	time, in cycles
3	set1	Double	operational setting 1
4	set2	Double	operational setting 2
5	set3	Double	operational setting 3
6	sensor_val1	Double	sensor measurement 1
7	sensor_val2	Double	sensor measurement 2
...	...		
26	sensor_val21	Double	sensor measurement 21

   The interpretation of the data rows is as follows. For example, the engine with engine id 1, was working for 192 cycles and then it failed. So, (i) at cycle 191 the RUL for engine with engine id 1 was equal to 1; (ii) at cycle 190 the RUL for engine with engine id 1 was equal to 2.

2. `DataSchedule.csv`. This is a csv-file that contains data for 100 engines that are currently in use. We will refer to this data as *dataset 2*. Unlike *dataset 1*, *dataset 2* contains temporal data that terminates some time before a system failure. This dataset contains the same variables as *dataset 1*. The RUL for these engines is not known. Based on your predictive model, you will predict the RUL for these 100 engines.

3. `RUL_consultancy_predictions_A3.csv`. The contents of this file will be explained in the Prediction Task.

1. Prediction Task

In this task, the goal is to develop a predictive model that will predict the remaining useful lifetime RUL of the engines.

1.1 Modeling [7 points]

1. Analyse *data set 1*, using descriptive algorithms (visualization, statistics), to see how you can construct the best features and how to obtain the RUL from this data set. Clearly describe your steps and how you obtained the RUL and features.
2. Use the constructed features from step 1 to develop a Random Forest model.
3. How do you specify your choice of hyperparameters? How do you validate the performance of your model? Which accuracy score did you use and why? Clearly describe every step of your approach. You can use 5-fold cross-validation and a train-test split of $80\% - 20\%$.

1.2 Prediction [3 points]

1. Use your predictive model from Task 1.1 to predict the RULs for the engines in *data set 2* which are not known. Please round your predictions to integer values. These integer-valued predictions will be used in the subsequent tasks.
2. After that, compare your predictions with that of an external data set produced by a consultancy company. This data set can be found in `RUL_consultancy_predictions_A3.csv`. The idea is to see whether your predictions are significantly different from the consultancy's. Specify your choice of metrics that measure that difference or any alternative statistical testing techniques that you use, and analyze and discuss any differences.

Tips and Hints Prediction Task

- The cycle is used to calculate the RUL; however, they are not the same.
- Input variables that are important for predicting the RUL will correlate with it.
- Not all scoring functions can be used for this problem. Make sure that you choose the correct scoring function.
- Don't spend too much time on getting an optimal score. We will mainly grade you based on your explanation in your report.

2. Optimization Task

Introduction

In this task, the goal is to develop a maintenance schedule for the engines in *data set 2*. The output of Task 1.2 should be a list that contains a predicted RUL for each engine in *data set 2*. These predicted values will now be used as input in order to allocate teams of workers to perform maintenance.

Maintenance teams

We want to allocate workers to different engines in order to perform maintenance. Workers perform maintenance in teams, and teams can only work on one engine at a time. Also, two teams cannot work on the same engine simultaneously. There are different types of teams: teams of type A and teams of type B. There are G teams in total. In our case, assume that there are $G = 4$ teams: 2 teams of type A and 2 teams of type B. Assume that teams T_1, T_3 are type A, and teams T_2, T_4 are type B.

The engines that need maintenance are numbered with indices from the set $\mathcal{M} = \{1, \ldots, M\}$, with M being the total number of engines. Teams of type A need μ_j^A days to perform maintenance on engine $j \in \mathcal{M}$, and teams of type B need μ_j^B days.

In our case, $M = 100$. The values for $\mu_j^A, j \in \mathcal{M}$ are as follows: $\mu_j^A = 5, j \in \{1, 2, \ldots, 20\}$, (ii) $\mu_j^A = 3, j \in \{21, 22, \ldots, 55\}$, (iii) $\mu_j^A = 4, j \in \{56, 57, \ldots, 80\}$, and (iv) $\mu_j^A = 5, j \in \{81, 82, \ldots, 100\}$. The values for $\mu_j^B, j \in \mathcal{M}$ are as follows: $\mu_j^B = \mu_j^A - 1, j \in \{1, 2, \ldots, 25\}$, (ii) $\mu_j^B = \mu_j^A + 3, j \in \{26, 27, \ldots, 70\}$, and (iii) $\mu_j^B = \mu_j^A + 2, j \in \{71, 72, \ldots, 100\}$.

Timing of events, penalty costs and safety due date

Time is discrete and measured in days. For each engine we associate a *safety due date* with its RUL. Suppose that we are currently at day t_1 and that engine j has a RUL of R_j, then the safety due date is equal to day $t_s = t_1 + R_j - 1$. This means that on day $t_s = t_1 + R_j - 1$ the engine is still working properly, and on the next day it does not work properly anymore.

If an engine is not maintained by its safety due date, then the company incurs a penalty cost. The cost c depends on the number of days the engine remains unmaintained: For every day t after the safety due date t_s, the cost incurred is

$$c = c_j \times (t - t_s)^2$$

for engine $j \in \mathcal{M}$, up to a maximal daily cost of 250 which stays constant for any further unmaintained days. For example, on the third day after the safety due date, a machine j with $c_j = 5$ would incur a cost of $5 \times 3^2 = 45$.

In our case, the values for $c_j, j \in \mathcal{M}$ are as follows: $c_j = 4, j \in \{1, 2, \ldots, 25\}$, (ii) $c_j = 2, j \in \{26, 27, \ldots, 45\}$, (iii) $c_j = 5, j \in \{46, 47, \ldots, 75\}$ and (iv) $c_j = 6, j \in \{76, 77, \ldots, 100\}$.

A maintenance schedule is made for a planning horizon of T days, that is, the maintenance schedule pertains to the planning period which consists of days $1, 2, \ldots, T$. In this task, we make the following assumption: If maintenance is performed on an engine during the planning period, then the RUL (after maintenance) of the engine will exceed T. This ensures that engines cannot "fail" twice within a planning horizon (that is, engines cannot have two safety due dates within a planning horizon). Therefore, maintenance on engine $j \in \mathcal{M}$ can be performed at most once during the planning period. The company is only interested in *complete* schedules: If a team is assigned to perform maintenance during the planning period, then the team must be able to complete the maintenance within the planning period.

2.1 Genetic Algorithm [14 points]

First, you are going to develop a Genetic Algorithm to solve the optimization problem. Company X is going to allocate teams to different engines in order to perform maintenance. Assume that we are currently at day $t = 1$ and that company X wants to allocate teams to engines in order to minimize penalty costs for a planning horizon of $T = 30$. In other words, company X wants to allocate teams to engines that have a predicted safety due date of less than 30.

1. Write a Genetic Algorithm that solves the problem for company X.

2. Clearly explain your approach. Answer in particular the following questions: (a) What does an individual look like, representing a complete maintenance schedule? (b) What is your selection strategy (to select parents)? (c) What is your crossover method? (d) What is your mutation method, if you want to use one? If not, why do you not need one? (e) How do you deal with infeasible individuals after crossover and/or mutation? (f) What are your population size, your crossover and mutation probabilities, and any other design choices and hyperparameters you might need?

3. The algorithm should be suitable to solve the problem, and should be clearly described. It should be able to accept the predicted RULs for all engines as input, and output a valid maintenance solution for it. Such a solution should provide: (i) a list of all machines, indicating for any *maintained* machine the type of the team doing the maintenance, start-date, and end-date of the maintenance, as well as the penalty costs incurred by that machine; (ii) the total penalty costs. It is not important here what the concrete cost of such a solution is, it just should be feasible. Make sure your solution format is easy to read and understand.

2.2 Optimization [3 points]

Now, you are going to solve the optimization problem with the Genetic Algorithm you have created in Task 2.1, and the RULs you have computed in the Prediction Task.

1. Run your Genetic Algorithm for a total runtime of max. 5 minutes, using your RUL predictions as input. Report on the best found solution.

2. As explained above, this solution should contain: (i) a list of all machines, indicating for any *maintained* machine the type of the team doing the maintenance, start-date, and end-date of the maintenance, as well as the penalty costs incurred by that machine; (ii) the total penalty costs. Make sure your solution format is easy to read and understand.

3. Run your algorithm 30 times for at most 5 minutes, and plot the average-best-fitness. The actual performance you achieve is not that important, although a notably bad performance or lack of any improvement could of course be an indication of bugs.

4. Discuss your findings.

2.3 Comparison [3 points]

1. Run your Genetic Algorithm on the same problem, but with the RUL predictions of the consultancy company. Run it 30 times for at most 5 minutes, and plot the average-best-fitness.

2. Discuss your findings. Are the results different? If so, can you explain these differences?

Tips and Hints Optimization Task

- Normally, it is already essential to make sure that your Python code is well-written and readable. However, for this assignment, this is even more important. For example, make sure that the names for variables etc. have the same name in your code as they do in your report.

- Again, your implementation and explanation are way more important than the results.

- It is easy to make small mistakes; repeatedly check if your solutions fulfill all constraints, and if your penalty costs are computed accurately.

- If you have trouble finding any feasible solutions, check if you are initializing the population with too many maintenance jobs.

On Grading

- Plagiarism is not allowed; do not submit anything you have not written yourself, or which is not your idea. We will check for plagiarism, and it might lead to zero points and even your exclusion from this course for this year.
- We have to be able to run your Python code, so we will subtract points if you use hardcoded file paths that do not work on other machines.
- We will subtract points if your Python code is not well-written. Well-written means:
  - The program is easily readable and includes enough functions and procedures for the various steps.
  - The program uses meaningful names for the variables used (not just a, b, c, d for example).
  - The program includes sufficient comments to explain all major steps (inside the program, not only using the comment structure in Jupyter).