Deliverables

There will be 4 main deliverables for each team:

- One deliverable at the end of each block consisting of:
    - Report summarizing the experiments and results for the block.
    - Code used to:
        - Generate, analyze, visualize, and pre-process the data.
        - Train and fine tune models.
        - Run a model on a data set to generate predictions.
        - Analyze the results.
    - Best model that will be used by the TAs for the blind test set evaluation.
- Presentation at the end of the semester summarizing the experiments, results, and observations for a project.
    - Note: each team working in the last block of a semester (i.e. block #3) on a particular project will prepare a presentation for the company interested in this project. The presentation will summarize the work done across blocks 1 to 3 for the project.

Deadlines

- Each team needs to provide the deliverable (report + code + best model) corresponding to a block at the latest on Friday 11:59pm of the last week of the block.
- Any block deliverable that is provided past Friday 11:59pm of the last week of a block will automatically get 0% for the peer evaluation.
- Any block deliverable that is provided past Tuesday 11:59pm following the last week of a block will automatically get 0% for the UdeM evaluation.
- Peer evaluation must be completed by Monday 11:59pm following the last week of a block.

Peer evaluation

- The code provided by a team will be evaluated by at least 2 other teams.

Block 1 evaluation

10% Code review [5% of averaged peer evaluation + 5% UdeM]

Code quality (peer evaluation + UdeM evaluation)	8
Coherent and modular code/file organization (e.g. data processing, model definition, model training, model inference are in different files/modules; no code duplication)	
Code respects the PEP8 standard	
Comments are relevant (see article)	
Proper management of input arguments in the training script (see argparse, python fire, configparser)	
Proper utilization of GitHub (e.g. branching, relevant commits and messages, usage of pull request)	
Meaningful variable and function names	
Executable scripts with a “main” function (see article)	
Reproducible experiments (e.g. seed)	

12% Report evaluation [UdeM]

Introduction	2
Introduction to the project	
Brief introduction to the methods that will be used in the report	
Methodology	6
Description of the algorithms and the experiments (including hyperparameter fine tuning (if appropriate), etc.)	
Data description and data selection (train/valid/test, number of samples, shape/structure of data points)	
Results and discussion	6
Presentation of results (tables, figures, etc.)	
Discussion of results	
Conclusion	2
Recommendation for next steps	
Summary of project state (what was done, what needs to be done)	
Quality of the report	2
Report format (title with team member names, clear sections, flow between sections, figures and tables titled, axes titled, etc.)	
Report is short and to the point (5-7 pages including references, font size 11)	

3% Model performance evaluation on blind test set [UdeM]

- If the best model provided by a team crashes or provides results that are statistically worse than those of the baseline model provided by the TAs, the team gets 0%.
- Otherwise, if the best model provided by a team is statistically equivalent to the baseline model, the team gets 1%.
- Otherwise, if the best model provided by a team is statistically better than the baseline model:
    - The team gets 3% if the model is the best performing one or is statistically equivalent to the best performing model provided by another team.
    - Otherwise, the team gets 2%.

Block 2 evaluation

10% Code review [5% of averaged peer evaluation and 5% UdeM]

Code quality (peer evaluation + UdeM evaluation)	8
Coherent and modular code/file organization (e.g. data processing, model definition, model training, model inference are in different files/modules; no code duplication)	
Code respects the PEP8 standard	
Comments are relevant (see article)	
Proper management of input arguments in the training script (see argparse, python fire, configparser)	
Proper utilization of GitHub (e.g. branching, relevant commits and messages, usage of pull request)	
Meaningful variable and function names	
Executable scripts with a “main” function (see article)	
Reproducible experiments (e.g. seed)	

12% Report evaluation [UdeM]

Introduction	2
Introduction to the project	
Brief introduction to the methods that will be used in the report	
Methodology	6
Description of the algorithms and the experiments (including a description of the approaches used to fine tune the hyperparameters, select the best “model” using checkpointing, etc.)	
Data description and data selection (train/valid/test, number of samples, shape/structure of data points)	
Results and discussion	6
Presentation of results (tables, figures, etc.). Note that this should include: ● A comparison with results from the previous block. ● Figures showing the loss value across epochs/checkpoints and models (using tensorboard).	
Discussion of results	
Conclusion	2
Recommendation for next steps	
Summary of project state (what was done, what needs to be done)	
Quality of the report	2
Report format (title with team member names, clear sections, flow between sections, figures and tables titled, axes titled, etc.)	
Report is short and to the point (5-7 pages including references, font size 11)	

3% Model performance evaluation on blind test set [UdeM]

- If the best model provided by a team crashes or provides results that are statistically worse than those of the baseline model provided by the TAs, the team gets 0%.
- Otherwise, if the best model provided by a team is statistically equivalent to the baseline model, the team gets 1%.
- Otherwise, if the best model provided by a team is statistically better than the baseline model:
    - The team gets 3% if the model is the best performing one or is statistically equivalent to the best performing model provided by another team.
    - Otherwise, the team gets 2%.

Block 3 evaluation

10% Code review [5% of averaged peer evaluation and 5% UdeM]

Code quality (peer evaluation + UdeM evaluation)	8
Coherent and modular code/file organization (e.g. data processing, model definition, model training, model inference are in different files/modules; no code duplication)	
Code respects the PEP8 standard	
Comments are relevant (see article)	
Proper management of input arguments in the training script (see argparse, python fire, configparser)	
Proper utilization of GitHub (e.g. branching, relevant commits and messages, usage of pull request)	
Meaningful variable and function names	
Executable scripts with a “main” function (see article)	
Reproducible experiments (e.g. seed)	

12% Report evaluation [UdeM]

Introduction	2
Introduction to the project	
Brief introduction to the methods that will be used in the report	
Methodology	6
Description of the algorithms and the experiments (including a description of the approaches used to fine tune the hyperparameters, select the best "model" using checkpointing, etc.)	
Data description and data selection (train/valid/test, number of samples, shape/structure of data points)	
Results and discussion	6
Presentation of results (tables, figures, etc.). Note that this should include: ● A comparison with results from the previous block. ● Figures showing the loss value across epochs/checkpoints and models (using tensorboard).	
Discussion of results	
Conclusion	2
Recommendation for next steps	
Summary of project state (what was done, what needs to be done)	
Quality of the report	2
Report format (title with team member names, clear sections, flow between sections, figures and tables titled, axes titled, etc.)	
Report is short and to the point (5-7 pages including references, font size 11)	

3% Model performance evaluation on blind test set [UdeM]

- If the best model provided by a team crashes or provides results that are statistically worse than those of the baseline model provided by the TAs, the team gets 0%.
- Otherwise, if the best model provided by a team is statistically equivalent to the baseline model, the team gets 1%.

- Otherwise, if the best model provided by a team is statistically better than the baseline model:
    - The team gets 3% if the model is the best performing one or is statistically equivalent to the best performing model provided by another team.
    - Otherwise, the team gets 2%.

Global evaluation

25% for final presentation in front of companies (15 min presentation + 5 min questions)

Content of the presentation	5
Description of the project	
Description of the solutions adopted	
Presentation of the achievements	
Identification of major problems	
Synthesis of findings and recommendations	
Format of the presentation	3
The presentation is clear and structured	
Figures and tables are adequate to present the results	
Respect of time	
Questions period	1
The answers to the questions are precise and clear	