Project evaluation

1) Interesting research question (max 5 points)

2) Technical soundness (max 10 points)

3) Writing clarity report (max 6 points)

4) Code organisation (max 10 points)

1) Interesting research question

What we look for

- **Novelty** or **real-world relevance** in the problem statement
- **Connection** to **broader applications** (e.g., healthcare, finance)
- **Potential** for **future exploration** or **extensions**

Example

- A project that explores how Transformers can classify sentiment in scientific abstracts, filling a gap in existing literature
- This approach could be expanded to automatically generate a domain-specific summary

2) Technical soundness

What we look for

- **Proper data exploration** (e.g., checking data distribution, handling missing values
- Sound **choice** and **justification** of **models** (FNN, CNN, Transformer, etc)
- **Hyperparameter tuning** (learning rate, batch size, network depth, etc)
- Clear **experimental design** and **metrics**

Example

- Explore the dataset’s distribution, identify skewed features, and use a model with a carefully tuned learning rate schedule
- Document each experimental run (varying filter sizes, dropout rates) and justify why certain configurations worked best

3) Writing clarity of the report

What we look for

- **Well-structured report** with **clear problem statement** and **objectives**
- **Logical flow** from **method** to **results** to **discussion**
- **Explanatory figures** and **tables**

Example

- A report that 1) starts with a short abstract, 2) introduces the research question, 3) describes the methods in detail (with figures/workflows), 4) presents results using tables and figures, and 5) concludes with a discussion of findings and limitations

4) Code organisation

What we look for

- **Readable**, **well-documented code (**functions, modules, docstrings)
- **Clear folder structure** (e.g., “src,” “notebooks,” “data”)
- **Easy** to **replicate environment** (requirements file or environment.yml)

Example

- A project repository that contains separate folders for scripts, notebooks, and data
- Each function is documented, and a README explains how to install dependencies and run the experiments
- Version control commits have clear messages.