1. Main Points

The primary objective of the coursework is to demonstrate the knowledge you have gathered. Focus on choosing a problem that is sufficiently challenging to showcase understanding but not overly trivial. Avoid simplistic tasks, such as a plug-in classifier, that may not adequately demonstrate your knowledge.

2. Dataset Selection

- Start by identifying a labeled open-source dataset.
- Minimize or eliminate time spent on labeling; the emphasis is on models rather than data engineering.
- No strict preferences for datasets; feel free to use any open-source dataset, including those provided in the module - always provide link or access to the dataset.

3. Problem Definition

- Clearly define the problem you intend to address and explain why it is interesting.
- Set realistic objectives, considering the limited time available.

4. Model Selection

- Choose an open-source model suitable for the identified problem.
- Provide a rationale for why the selected model is a good fit.
- Don’t just use the model out of the box - come-up with additions, provide the reasoning for the changes.

5. Accuracy Metrics

- Define the accuracy metrics you plan to use.
- Explain why these metrics are appropriate for the specific problem.

6. Collaboration and Approval

- Prior to starting the coursework, discuss your plan with the instructor to receive feedback.

7. Pair Work Considerations

- If working in pairs, each student must select a different section for work.
- Grades will differ based on individual contributions.

8. Reporting Guidelines

- Present important details clearly and readably.
- Enumerate tables, equations, figures, etc., and refer to them correctly in the text.
- Discuss key concepts rather than restating or rewording published papers.

9. Conciseness and Focus

Exclude discussions on basic and widely-known concepts, such as ConvNet and MLP layers, learning rate, weight decay, batch, etc. Avoid in-depth discussions on older models like (Fast) R-CNN. However make sure you reference them and specify why they are relevant to YOUR project.

- Succinctly discuss basic classifiers like ResNet, focusing on their specific methodology.
- Present mathematical ideas, such as optimizer functionality and loss functions, succinctly with equations and explanations.

10. Code Quality

- Ensure all code submitted is meaningful.
- Models must train effectively, and results should reflect the intended metrics. Your grade will suffer if your model was not able to learn anything.

11. Models that are too basic

- Avoid the use of overly basic models like MLE and NNLM from Weeks 2 and 3.

12. Evaluation Criteria

- Hyperparameters and model discussions are essential, but code quality and meaningful results take precedence.
- Avoid extreme values in metrics (e.g., 100% or 0%) that may indicate bugs; strive for realistic performance.