50.038 Computational Data Science
Week 4 (Blended Learning)
Academic Success Predictor (EDM)

See also

- `README.md` (full instructions + rubric)
- `data_dictionary.md` (column definitions)

Scenario

You are the teaching team's lead data scientists. Using a (synthetic) pre-course survey dataset, you will build a model to predict student outcomes and extract cohort insights that could inform supportive interventions.

Files (Week 4 folder)

- `student_success_survey.csv`
- `edm_template.ipynb`

Task 1 — Predict student success (choose ONE framing)

A) Regression

- Target: `final_course_score` (0–100)
- Evaluate: MSE and R^2

B) Classification

- You define a binary label (e.g., "Distinction" if score $\geq T$)
- Evaluate: confusion matrix + precision/recall/F_1

Requirement

- Implement the predictive model as a single linear layer using PyTorch: `nn.Linear`

Task 2 — Cohort analysis (EDA)

Before modeling, justify feature choices using visuals. Minimum EDA questions to answer:

1. Does grit correlate with planned weekly study hours?
2. Is CGPA roughly linearly related to `final_course_score`?
3. Are there noticeable differences by academic pillar?

Task 3 — Feature engineering (≥ 2 engineered features)

Create at least two engineered features. Suggested examples:

- `avg_grit`: average grit items with reverse-coding where appropriate
- `tech_readiness`: diagnostic correctness + tool experience
- `time_budget`: `hours_per_week_planned` − `commute_minutes_daily`/60

Task 4 — Dimensionality reduction (PCA)

Use PCA on your preprocessed design matrix and discuss:

- whether students cluster
- what PCA does *not* tell you

Task 5 — "At-Risk Intervention Alert" (agentic add-on)

Build a function that identifies at-risk students and outputs a supportive recommendation.

Discussion prompt: Why might you prioritize high Recall (even if Precision drops) for an at-risk alert system?

Logistics (Lab Sprint)

- Work with your project team during lab.
- Submit a fully executed, well-documented `edm_template.ipynb` by end of lab.

Submission checklist

- Notebook runs end-to-end without errors
- Clear plots + short written interpretation
- ≥ 2 engineered features + justification
- PyTorch `nn.Linear` + training loop

- Proper evaluation metrics
- Intervention alert + ethical reflection