Scaffold ML modeling

Machine Learning Project Notebook – Modeling & Evaluation Guideline (ML Part)

> **Note:** Depending on the problem, some of these steps may differ, merge, or be skipped. This document is **not a form to be filled out**, but a **guideline** to help you check whether you have covered the essential reasoning and reflection steps of a complete ML process.

Scope Clarification

> The sections **Data Understanding** and **Data Cleaning** are evaluated by the **DAIA teacher**.
> My evaluation focus starts **from Feature Engineering onward**, covering **representation issues, modeling, and evaluation** — the *core ML process*.
> While I am partly concerned with how features are represented (encoding, grouping, bias), the main emphasis of this part is **modeling quality**, **comparative analysis**, and **interpretation of results**.
> You are encouraged to include **at least one imbalanced classification problem** to meaningfully discuss **precision, recall**, and related metrics.

Themes in ML Work

Theme	Core Expectation	Typical Student Gap
1. Structure & Planning	Organize the notebook logically; include a plan, track progress, state what was done and learned.	Students dive into code with no structure or reflection.
2. Depth of Understanding	Explain each major decision (encoding, model choice, hyperparameters); show comprehension of what code *means*.	Students run GPT-generated code without understanding.
3. Modeling & Analysis	Build proper ML models, compare alternatives, discuss bias, over/underfitting, and metrics.	Students stop at “working code” or rule-based logic.
4. Experimentation & Evaluation	Perform parameter sweeps, interpret feature importance, test different methods, compare results.	Students run one model once, no systematic comparison.

Theme	Core Expectation	Typical Student Gap
5. Reflection & Communication	Summarize findings, discuss limitations, evaluate personal learning and relevance.	Students deliver outputs, not insights.

Machine Learning Project Notebook Template (ML Part Only)

0. Project Overview & Plan

Purpose: To define the modeling goal and keep track of progress. Helps maintain structure and clarity for both you and the reader.
Learning goal: Understand where you are in the process and what each step contributes to the bigger picture.

Step	Status	What I learned
Data exploration		
Feature engineering		
Modeling		
Evaluation		

---

1. One-Page Synthesis — Fill This First (Required)

Purpose: Forces you to summarize your findings and learning before showing code. This builds narrative thinking and self-awareness.
Learning goal: To distinguish between what the *data* says (objective) and what *you* learned (subjective).

A. Objective insights (what the data/experiments show)

- Target & task (one line): e.g., “Target = `liked_episode` — binary classification.”
- One-line performance baseline (e.g., majority class accuracy = X; chosen model F1 = Y).
- 2–3 useful findings from data/models (features, imbalance, major signal).
- Note on model reliability (overfit? stable CV? train vs. test gap = Z).
- “Decision I make now because of these findings…” (e.g., choose model X, change metric to recall).

B. Subjective reflections (what you learned / what surprised you)

- What surprised you or what you don't understand yet.
- Which design choice might introduce bias (and why).
- If you had 2 more hours you would try…
- How this work helps the stakeholder / why it matters.
- One thing you'd explain differently to a teammate.

> This block must be completed before any model code runs.

---

2. Feature Engineering (Representation Focus)

Purpose: To decide how raw data becomes model-ready.
Learning goal: Understand how feature choices affect bias, interpretability, and model quality.

- Describe the final features fed to the model and the reason for each major transformation.
- Explain why the representation is appropriate for the task (e.g., ordinal vs. categorical).
- If you grouped or encoded values (top-10 + "other"), justify it and note possible bias.
- **Do not perform manual feature selection** based only on correlations; compare against a more extensive model or use algorithmic methods (Lasso, ElasticNet, tree importance).
- Manual feature dropping without comparison = high risk of representation bias.

---

3. Modeling (Algorithms & Training)

Purpose: To connect theory to application — turning features into predictive models.
Learning goal: Understand how different algorithms behave and how to interpret model performance.

- List candidate algorithms and why they fit the task.
- Define train/test split or CV strategy.
- Train at least **two models** and show both train and test metrics.
- Discuss metric choice (precision, recall, F1, RMSE, etc.) and why it fits the goal.
- Provide a concise results table for comparison.
- Use plots for clarity when appropriate.
- Report both train and test metrics. Discuss overfitting or underfitting.
- If you report R^2, also include the residual plot.

---

4. Hyperparameter Exploration

Purpose: To understand model sensitivity and tuning.
Learning goal: Learn how hyperparameters influence generalization, overfitting, and performance.

- Identify key hyperparameters (e.g., `max_depth`, `C`, `learning_rate`) using scikit-learn documentation.
- Run a parameter sweep and visualize how metrics change. Plot both train and test.
- Interpret the effect briefly (≤3 sentences).
- Explain how overfitting or underfitting appears in these curves.

---

5. Evaluation & Over/Underfitting Checks

Purpose: To assess whether the model truly learns the intended patterns.
Learning goal: Distinguish between apparent and genuine model performance.

- Show train vs. test performance and, if possible, learning curves.
- Discuss stability across folds or resamples.
- Interpret precision, recall, and confusion matrix in context (what do false positives mean?).
- If reporting R^2, include a residual plot.
- Explore how sample size affects stability by subsampling the dataset.

---

6. Feature Importance & Bias Checks

Purpose: To interpret what drives the model's decisions.
Learning goal: Understand which features matter and whether they introduce unfairness or bias.

- Quantify which features matter (tree importance, permutation importance, or SHAP).
- For every plot or table, explain what we can conclude from it. Move exploratory plots to an appendix.
- Interpret importance results in context of potential bias.
- If you simplified features (e.g., top-k vs. full encoding), compare models and report metric change.
- Never claim "importance" without quantitative evidence.
- Reflect on fairness implications (if any).
- Try simple counterfactuals: modify one feature (e.g., gender, income, location) and see how predictions change.

---

7. Decision & Next Steps

Purpose: To consolidate learning and communicate takeaways.
Learning goal: Practice translating results into meaningful actions or future experiments.

- State final model choice and why.
- Prioritize next experiments (2–3 items).
- Mention limitations or open questions.

---

8. Reproducibility & Transparency

Purpose: To make your work traceable and credible.
Learning goal: Build habits of scientific integrity and transparency.

- If you used external code (e.g., GPT, StackOverflow), **cite it** and add a one-line check: Did you read the documentation and verify each argument?
- Example note: "I used GPT for X snippet and adapted it by Y; checked doc for parameter Z."

Marking Checklist / Rubric (ML Part)

Criterion	Notes
One-page synthesis (objective + subjective) present	Mandatory
Target + features clearly stated	
Baseline shown and compared	
≥ 2 models compared (train & test/CV metrics)	
Hyperparameter sweep with interpretation	
Over/underfitting check with interpretation	
Feature importance + bias discussion (quantified)	
Final decision + prioritized next steps	
Reproducibility notes & citations of external code	
Narrative present under major outputs (1–3 sentences each)	Mandatory

---

Guidelines on External Code

If you used external snippets or GPT-generated code:

1. Cite the source (URL or “GPT generated”).
2. Add a one-line statement explaining how you checked correctness (e.g., “verified function arguments in docs”).
3. Explain in your own words what the code does.
   Transparency is more important than prohibition.