Final Project: Integrating Your ML Pipeline

Due: Sunday, December 7th @ midnight (with 2-hour grace period)

Points: 180 points total

Submission: Group submission by the Team Leader to Gradescope (Parts A + B submitted separately)

- **Part A: Final Code Notebook (.ipynb)** — Well-organized and executable repo of your project code.
- **Part B: Technical Report (.pdf)** — Professional format, 8–10 pages total, including graphics (appendices permitted for additional figures).

Overview

This Final Project brings together all the stages of the Machine Learning workflow you've practiced in Milestones 1 and 2—from problem framing and data exploration, through feature engineering and model selection, to final training and analysis. By the end of this project, your team will have:

- A **clean, reusable notebook** containing your best preprocessing pipeline for your dataset, your fine-tuned final model, and your analytical code, including visualizations. This represents a **repository** of your work on the project, and should provide clear exposition of all your work, so that you could pick it up again in a year without much trouble (and *so that your teaching team can grade it*).
- A **professional report** describing your process, results, and insights; write it for a technical audience (so you don't have to explain what "accuracy" means), but one not necessarily current on all aspects of deep learning (so you might have to motivate and justify your preprocessing steps, your network architecture, and your choice of metrics).

Structure and Deliverables

Whenever possible, use the rubric labels given here, i.e., your notebook should have sections "A.1: Setup and Context", "A.2: Model Selection and Retraining", and so on.

Part A: Final Code Notebook (80 pts)

Your final notebook must present a **complete, clean, and reproducible implementation** of your project pipeline. It should read as a coherent document with sections **A.1**, **A.2**, and **A.3** that show (1) how you prepared your data, (2) how you trained your final model, and (3) how you evaluated it. The notebook must run **end-to-end without errors**, using no hard-coded paths.

A.1: Setup and Context (10 pts)

Provide a brief, clear overview of the project context and the work leading to this final model.

Required Elements

- **Summary of Milestones 1 and 2:**
  A short Markdown overview describing major steps you took previously, including:
  - Data cleaning and preprocessing
  - Feature engineering or augmentation strategies
  - Early model experiments in Milestone 2 and what you learned from them
- **Final preprocessing pipeline:**
  Include all finalized code for:
  - Loading the dataset
  - Cleaning steps (text normalization, image transforms, etc.)
  - Preprocessing (tokenization, resizing, normalization, augmentation, etc.)
    This is your opportunity to refine or simplify any parts of your earlier pipeline based on what you learned in Milestone 2.

Your notebook should begin with this summary to orient the reader.

A.2: Model Selection and Retraining (25 pts)

You will choose **one final model** based on your Milestone 2 validation results and retrain it cleanly in this notebook.

Required Elements

- **Final architecture and justification (Markdown + code):**
  - Clearly state what architecture you selected (CNN, pretrained encoder, TF-IDF + dense, etc.).
  - Briefly explain the method and metrics used to select your final model (a fuller explanation will be provided in the Project Report)
- **Hyperparameter definitions:**
  Clearly document the values for:

    - Learning rate and LR schedulers
    - Batch size
    - Number of epochs
    - Dropout rates
    - Weight decay / L2 regularization (if any)
    - Anything else…
- **Optimization strategy:**
  Show the optimizer configuration and describe why it is appropriate (e.g., Adam for stability, SGD + momentum, etc.).
- **Callbacks:**
  Include any callbacks used during training (early stopping, checkpointing, LR scheduler).
  Briefly explain how these helped prevent overfitting or improve training stability.
- **Retraining from scratch:**
  Run the full training loop using:
    - The entire training set
    - A held-out validation split
    - A separate test set reserved for final evaluation

All of this should be implemented cleanly and commented in the notebook.

A.3: Evaluation and Visualization (25 pts)

This section should illustrate how you measured, evaluated, and visualized your model's performance.

Required Elements

- **Explanation of evaluation strategy:**
  In a Markdown cell, describe:
    - How the validation split was used
    - How the test set was kept separate
    - What metrics are appropriate for your dataset:
        - Food101: accuracy, per-class accuracy, top-k accuracy, etc.
        - HuffPost: macro-F1, weighted-F1, accuracy, etc.
- **Required plots and printouts:**
    - **Training vs. validation loss curves**
    - **Training vs. validation accuracy (or other metric)**
    - **Confusion matrix** on the test set
    - **Any dataset-specific evaluations** (per-class metrics tables, F1 score tables, etc.)
- **Final test metrics:**
  Include a clean printout of the final test performance (accuracy/per-class metrics/etc.).

Plots must be readable, properly labeled, and integrated into the notebook narrative.

A.4: Code Readability and Documentation – General Requirements (20 pts)

(This next section is not a separate graded section of the notebook, but an indication of how it will be graded.)

Overall, your notebook should be polished, well-structured, and easy for an LF to follow.

Required Elements

- **Professional notebook header:**
  A Markdown cell at the top including:
  - Title of the project
  - Names of team members
  - Date and course
  - One-sentence description of the notebook's purpose
- **Clear section organization:**
  The notebook must follow the structure of Part A (A.1, A.2, A.3, no need for A.4), with headings and logical subsections.
- **Markdown documentation:**
  Provide Markdown explanations before major code cells describing:
  - What the code does
  - How it fits into the modeling workflow
- **Readable, commented code:**
  - Every major block of code should have comments
  - Functions should be named clearly
  - Magic numbers should be avoided or explained (make them constants at top of cell)
  - Code should be modular when appropriate
- **Reproducibility:**
  - The notebook must run end-to-end without errors
  - No hard-coded absolute file paths
  - Any random seeds should be set for reproducibility

Part B: Technical Report PDF (100 pts)

Your technical report in PDF form should present a clear, reproducible, and well-reasoned account of your final modeling pipeline. This section is worth **100 points**, divided among the four components below. The paper should be at least 8-10 pages, including graphics. Integrate text, formatted lists, tables, and graphics to effectively communicate. Your goal is to demonstrate professional documentation, thoughtful analysis, and evidence-based conclusions.

Your report should start with a **title page** containing:

- Project Title
- Team Member Names
- Course & Semester
- Date Submitted
- Dataset Used (optional but helpful)

After the title page, please **follow the organization given here**, using section titles exactly as listed below (B.1, B.2, B.3, etc.), and numbered subsections (e.g., B.1.1, B.1.2) as appropriate.

B.1: Introduction and Problem Definition (20 pts)

Your introduction must clearly frame the project and the dataset.

Required Elements

1. **Problem description and motivation:**
   Describe the classification problem you are solving and why it matters. You may invent a **plausible business or organizational context** that would make this model valuable (e.g., food image sorting for a restaurant app, topic routing for news feeds).
2. **Project goals in both business and data-science terms:**
   - Business goal: What real-world outcome are you supporting?
   - Data-science goal: What quantity or behavior is the model trying to predict, and with what constraints?
3. **Dataset description:**
   Provide a concise summary including:
   - Source of the dataset
   - Number of classes and their distribution
   - Key input modalities (images for Food101; text headlines for HuffPost)
   - Summary of features used
   - Any challenges: imbalance (HuffPost), noise (Food101), mislabeled data, class overlap, etc.

A brief EDA summary (plots or tables) should appear here or in an appendix.

B.2: Methodology (25 pts)

This section should explain **exactly how you built, trained, and validated your final model**, with enough detail for someone else to reproduce your work.

Required Elements

1. **Preprocessing Pipeline:**
   Describe how inputs were prepared. Examples:
   - *Food101:* resizing, normalization, augmentation steps, class sampling strategy
   - *HuffPost:* tokenization, vectorization (TF-IDF, embeddings), handling rare words
     Include any relevant statistics or visualizations (frequency distributions, augmentation examples, vocabulary size, etc.).
2. **Model architecture and justification:**
   Describe the network or classifier you chose as your **final model**, and explain:
   - Why it is appropriate for this dataset
   - How your earlier Milestone 2 experiments informed your choice
   - Key architectural components (e.g., CNN blocks, dropout layers, embeddings, pretrained backbone)
3. **Training strategy:**
   Specify:
   - Loss function (cross-entropy variants)
   - Optimizer, learning rate schedule, regularization
   - Important hyperparameters (batch size, number of epochs, etc.)
   - Callbacks (early stopping, checkpoints, LR reduction)
4. **Performance validation:**
   Describe how you evaluated performance:
   - Training/validation split
   - Test set procedure
   - Any stratification or class balancing
   - Measures taken to avoid data leakage

B.3: Results and Evaluation (25 pts)

This section presents and interprets the performance of your single final model**.**

Required Elements

1. **Quantitative results (tables and plots):**
   You must include:
   - Training, validation, and test accuracy
   - Training and validation loss curves
   - Confusion matrix for test predictions
   - **Dataset-specific metrics:**
     - *Food101:* (e.g., per-class accuracy or top-k accuracy)
     - *HuffPost:* (e.g., macro-F1, weighted-F1, and a class-frequency table)
2. **Interpretation of results:**
   Explain what the metrics reveal about:
   - Which classes are easiest/hardest
   - Patterns in the confusion matrix
   - Evidence of overfitting/underfitting
   - Impact of image quality (Food101) or imbalance (HuffPost)
3. **Summary of final model performance:**
   Instead of comparing multiple models, provide a concise explanation of:
   - Why this model represents your "best" solution
   - What its performance indicates about the difficulty of the task
   - Any surprising behaviors or strengths/weaknesses

Clarity and correctness of interpretation are part of the grade.

B.4: Discussion and Reflection (25 pts)

This final section evaluates your ability to think critically about your model, your data, and the overall project.

Required Elements

1. **Limitations and sources of error:**
   Describe realistic limitations such as:
   - *Food101:* noisy images, high similarity among classes, lighting variation
   - *HuffPost:* extreme imbalance, ambiguous category boundaries, short/noisy text
     Also mention model-related limitations (capacity, representation limits, sensitivity to hyperparameters).
2. **Potential improvements:**
   Propose at least **three** concrete improvements. Be specific. Examples:
   - Stronger augmentation (Food101)
   - Class-weighted or focal loss (HuffPost)
   - More expressive architecture (CNN backbone, transformer encoder)

    - Better tokenization or embeddings
    - Additional data cleaning or balancing techniques
3. **Reflection on what you learned:**
   Provide a thoughtful reflection on:
    - What mattered most in achieving good performance
    - What you discovered about tuning, optimization, or generalization
    - What you would do differently if you repeated the project
    - Any insights about dataset difficulty and model behavior

This should read as a concise, meaningful reflection—not a list of steps.

B.5: AI Use Disclosure (5 pts)

Include a short section describing any use of AI tools (e.g., ChatGPT, Copilot) in accordance with CDS policy. Where did you find it helpful? For what tasks? Where was it not as useful? What advice would you give to other students taking this module about using AI tools?