Group Assignment: End-to-End Machine Learning Project

Overview: In this group project, you will work in assigned teams of 4-5 to develop and package an end-to-end machine learning system using a dataset you find online. The chosen dataset and project must be of sufficient complexity and size to allow meaningful analysis and model development. Your ML system must be a complete and functional program, and must be packaged as a standalone executable file (e.g. .exe) that can run independently on a Windows machine without requiring manual setup of the Python environment. The emphasis is on a complete solution (from data to deployment) and the quality of your final report and presentation.

Below are the required steps and deliverables, along with a detailed assessment rubric. Follow the timeline and instructions carefully to ensure your project meets all requirements.

1. Dataset and Title Proposal

- Find a Unique Dataset & Project Idea: After groups are formed, each group must locate an interesting dataset online and propose a unique project title. The project should address a clear machine learning problem using this dataset. Ensure the dataset has enough samples/features to be non-trivial and relevant to your idea.
- Submit to Google Sheet: One member of each group must enter the proposed project title and a brief description of the dataset on the Google Sheet via the link provided [**here**] before **30/4/2025 23:59**. Please include a link to the dataset or a short explanation of how the dataset was obtained.
- First-Come, First-Served Approval: Title approval is on a first-come, first-served basis. No two groups may use the same dataset. Check the sheet to avoid duplicates. If another group has already listed your dataset or a very similar project, you will need to choose a different one.
- Instructor Review: The instructor will review each entry on the sheet. You will receive feedback or approval for your proposal. Approval is required before proceeding to the next stage. If your proposal is not approved (e.g. due to dataset overlap or insufficient complexity), refine your idea and update the sheet as instructed.

2. Graphical Abstract Submission

- Purpose: Once your project title and dataset are approved, each group must create a graphical abstract that visually summarizes your proposed ML system. This is a single image or diagram that gives an overview of your project at a glance.

- Content Requirements: The graphical abstract should illustrate your system's architecture and objectives. It must clearly show:
    - The data source (origin of your dataset) and how data flows through your system.
    - The problem being addressed (e.g. classification of images, predictive analytics, etc.), including any specific goal like improving accuracy beyond an existing benchmark or deploying a model in a new context (for example, *improving model accuracy beyond existing benchmarks*).
    - The main components of your ML system architecture (e.g. data preprocessing, model training, evaluation, and deployment interface). Use simple labels/arrows to make the process clear.
- Submission: Upload the graphical abstract via the Google Form provided [**here**] before **7/5/2025 23:59**. Only one group member needs to submit, but make sure the submission includes your group number/name and project title. This deliverable is intended to ensure you have a clear plan.

3. Final Report and Presentation

- Final Report (max 20 pages): Upon completing the project, each group will prepare a comprehensive report documenting your end-to-end ML system. The report should be well-structured and no more than 20 pages, excluding appendices or references. It must cover the following sections:
    - Abstract: A concise summary of the problem, approach, and results.
    - Introduction: Background context and motivation for the project. Include the problem statement here or in a separate section (define what you are trying to solve and why it's important).
    - Problem Statement: Clearly articulate the ML problem you are addressing and your project's objectives. What question are you answering or what target are you predicting?
    - Related Work: Brief review of any existing solutions, models, or literature related to your problem. Cite any relevant research or projects to show how your work is distinct or builds upon prior work.
    - Dataset Exploration: Description of the dataset (source, size, features) and any initial exploration/insights. Include summary statistics or example

instances to illustrate the data's characteristics. Explain why this dataset is suitable for your problem.

  - Data Preprocessing: Outline the preprocessing and cleaning steps. (How did you handle missing values, normalize or encode features, etc.?) If you did feature engineering or dimensionality reduction, describe those as well.

  - Modeling Approach: Explain the machine learning techniques and models you applied. Justify why you chose these models/algorithms for the problem. Include details of model architecture (if applicable, e.g. for neural networks) and how you trained them. If you tried multiple models, discuss them.

  - System Packaging: Describe how you packaged the end-to-end system. For example, is there a user interface (CLI commands, a web app via Streamlit, etc.)? How can someone run or interact with your solution? Detail the system architecture from a deployment perspective (how data flows from input to output in your implemented system).

  - Evaluation: Present the results of your model(s). Include evaluation metrics (accuracy, F1, RMSE, etc. as appropriate) and error analysis. Compare performance of different approaches if you tried multiple. Discuss whether you met your objectives (e.g. did you improve accuracy over a baseline or benchmark?). You should also mention any challenges, surprises, or improvements made during testing.

  - Conclusion: Summarize what was accomplished and any key takeaways. You can also suggest future work or how the system could be improved or extended.

  - References: List any sources, datasets, or external libraries/documentation you cited. Follow a consistent citation format.

- Recorded Presentation & Demo: Each group must also submit a video presentation with a live demo of the working system:

  - The presentation should be concise and engaging. Aim for a clear explanation that could be around 5–10 minutes long (you do not need to fill 10 minutes if not necessary, but keep it informative and to-the-point).

  - Content: Introduce the problem and dataset briefly, then explain your solution's architecture. Walk through how your system works using the graphical abstract or other visuals. Demonstrate the system in action – for example, show your Streamlit app responding to user input, or run your script/notebook on a sample to produce output. The demo should prove that your end-to-end pipeline functions correctly.

  - Clarity: All group members do not need to speak, but the presentation as a whole must be well-organized and easy to follow. Use slides or screen sharing effectively so that the viewer understands each step of your pipeline. Ensure the audio is clear. Practice to avoid long pauses or technical issues in the recording.
  - Format: The video file will be submitted via the provided Google Form. Check that the video is accessible (permission settings) and playable.

- Submission: Each group must upload their final report (as a PDF) and their presentation video via the Google Form provided [here]. Both the report and video must be submitted by **Week 15, Friday, 23:59**. Late submissions will not be accepted and will result in a score of zero (0) marks.
- Important: Only the final report and presentation will be graded for the project. Steps 1 and 2 (proposal and graphical abstract) are mandatory checkpoints to help you succeed, but their quality is not separately scored. Focus on producing a high-quality final deliverable. Ensure that your code is functional and the system is packaged so that it can be run or reviewed by the instructor if needed (include any necessary instructions in an appendix or in comments).

Assessment Rubric

The final project will be evaluated out of 100 marks, distributed across several criteria. The grading focuses on the quality of your problem formulation, technical implementation, and final outputs (report & presentation). Below is the breakdown of marks and what is expected for each component:

- Problem Definition & Understanding (10 points): Clarity and depth of your problem statement. A top-scoring project has a well-defined ML problem with clear objectives and rationale. The scope is appropriate (not too trivial nor too broad), and you demonstrate understanding of why the problem is important.
- Dataset Selection & Complexity (10 points): Quality and suitability of the chosen dataset. Full marks require a dataset that is sufficiently complex and sizable for meaningful analysis. The data should be relevant to the problem and preferably novel in context. Credit is given for insightful discussion of dataset characteristics and why it fits the project. Poor choices (e.g. a toy dataset that doesn't allow interesting analysis) will score lower.

- Model Development & Methodology (20 points): How well you developed the machine learning solution. This includes choice of algorithms/models, justification for those choices, and how you trained/tuned them. High scores reflect appropriate model selection (e.g. using state-of-the-art methods or thoughtful baselines), proper justification, and possibly creative improvements (such as feature engineering, ensembling, or tuning to boost performance). Simply applying a basic model without justification or experimentation would score low, whereas a thorough exploration of different models and hyperparameters with reasoning will score high.
- System Implementation & Packaging (15 points): The system must be fully implemented and packaged into an executable .exe file. The executable should run the entire ML pipeline from input to output, and demonstrate the final model in action. Instructions for using the .exe (e.g. sample input files, expected outputs, basic usage guide) should be included in the report appendix or submitted alongside the file. Projects that submit only Python scripts or notebooks without packaging into a .exe will not be considered complete.
- Analysis & Evaluation of Results (20 points): Depth of analysis of your model's performance and the insights you draw. High-scoring projects will use appropriate evaluation metrics and provide a thorough interpretation of results (discussing accuracy, errors, comparisons to benchmarks or baseline models, etc.). This includes identifying any limitations (underfitting/overfitting issues) and highlighting interesting findings. Discussing why the model performs as it does and what could be improved demonstrates critical thinking. Minimal or shallow evaluation (only providing raw numbers without discussion) will not score well.
- Report Quality and Structure (15 points): Clarity, organization, and completeness of the written report. A great report is well-formatted and includes all required sections with logical flow. The writing should be clear and concise, free of grammatical errors, and technical content should be explained correctly. Figures/tables (if any) should be labeled and referenced in the text. Full marks go to reports that are easy to read and understand, thoroughly document the project, and stay within the page limit while covering everything important. Missing sections, poor writing, or disorganized structure will reduce the score.
- Presentation & Demo Clarity (10 points): Effectiveness of the video presentation and demo in conveying your project. This includes how well the group explained the problem, approach, and results in a limited time, and how clearly the system demo was shown. A top presentation is well-organized, engaging, and demonstrates the system's functionality clearly (the audience can follow what the system does and

how). Good use of visuals (slides or live demo screen) and speaking clarity are expected. If the presentation is confusing, too long/short without covering key points, or if the demo is missing or not working, points will be deducted.

Total: 100 points

Note: The rubric focuses on the final deliverables. Make sure your report and presentation clearly highlight all aspects of your project, from problem motivation to system demo, to maximize your points in each category. Good luck, and we look forward to seeing your innovative machine learning solutions!