Project Report

Early Academic Risk Prediction Tool Using Lightweight Machine Learning

By Benjamin Dillon, Ayman Sharieff : COIL Group

Honor Statement: I have neither given nor received unauthorized assistance on this assignment.

LLM Usage Notice: During the preparation of this assignment, The COIL Group used ChatGPT in the COIL Project Report to aid in report formatting. After using this tool, we have reviewed and edited the content as needed to ensure its accuracy and take full responsibility for the content in relation to grading.

Learning Objectives:

LO1: Build a foundational understanding of machine learning classification methods

Our goal was to learn how classification works in real projects, starting with logistic regression. We wanted to understand how models learn from training data, how coefficients relate to feature importance, and how evaluation metrics like accuracy and confusion matrices reflect model performance.

We met this objective by training a three-class logistic regression model on our synthetic dataset. We interpreted model outputs, analyzed where High and Low risk overlap, and learned why multi-class classification is often harder than binary classification. The hands-on coding clarified how theory translates into real results.

LO2: Strengthen Python skills for data analysis and modeling

We aimed to strengthen our skills in Python's data ecosystem using pandas, NumPy, scikit-learn, matplotlib, seaborn, and ipywidgets.

This objective was fully met through the end-to-end workflow: we explored the Kaggle dataset, generated synthetic features using real statistical distributions, engineered a risk label using conditional logic, visualized data with seaborn, trained our model, and built an interactive widget

interface. This project significantly improved our confidence with practical machine learning coding and working inside Google Colab.

LO3: Learn to design, document, and implement ethical AI practices

A core goal of the project was understanding how to incorporate ethical reasoning into technical decisions. We wanted to practice concepts like data provenance, privacy, fairness, and appropriate use.

We met this objective by generating fully synthetic data based on real distribution patterns, avoiding sensitive attributes like gender or income, evaluating how features influence risk scoring, and documenting the limitations of our model. The project helped us understand how model design choices can unintentionally reinforce bias, and how transparency and careful feature selection can reduce those risks.

Timeline:

Outline how you spent time on your project. Break down the time into specific tasks or milestones. Here is an adjustable schedule to get you started. Actual Details should be 50-100 words each and should compare or reflect on differences from your proposal.

Time	**Task**	**Expected Details from Proposal**	**Actual Details**
Hour 1-2	Research and gather resources	Reviewed existing academic early-warning systems to understand which features best help predict student risk.	We explored the synthetic Kaggle Students Performance dataset in detail to understand realistic scoring distributions. We researched how multi-class logistic regression works,

		We will learn scikit-learn, pandas, and logistic regression documentation to see what kind of project we wanted to develop. We researched ethical concerns with student data and decided to use fully synthetic data. We also identified major academic indicators to include such as grade scores and socioeconomic indicators "lunch status".	especially how the one-vs-rest strategy is implemented in scikit-learn. We also reviewed documentation on pandas and ipywidgets to prepare for later steps. In the first few hours this is how we gathered research info to help us design the beginnings of our model making sure to keep into account the use of sensitive data/ethical concerns.
Hour 3-4	Design the project structure and plan	Planned the overall Google Colab notebook layout and workflow stages. Designed the synthetic dataset schema, selecting features, realistic ranges, and balanced risk categories. Mapped the ML pipeline including data generation, splitting, model training,	We designed the full machine learning workflow including synthetic dataset generation, risk label assignment, preprocessing, model training, evaluation, and the interactive interface. While planning we realized categorizing each student with hyper personal precision wasn't really feasible because too many subtle factors create ambiguity. To fix this we created broader Low, Medium,

		evaluation, and prediction interface.	and High risk categories which made the model way more stable and interpretable. We also structured the notebook to flow from data exploration to visualization to model deployment making sure each step built on the last.
Hour 5-6	Start coding the basic functionalities	Implemented the synthetic data generator using NumPy and pandas to create realistic student records. Built the first logistic regression classifier using scikit-learn. Created a working end-to-end pipeline from data generation to initial predictions. Tested with small datasets to ensure correctness before scaling up.	We started coding the synthetic dataset using real mean and standard deviation values taken from Kaggle's distributions. We created math, reading, and writing scores, included lunch type and test prep as proxies, and engineered an average score feature. We also wrote the conditional logic to create risk labels. After this we trained our first logistic regression model and confirmed that the code successfully ran end-to-end. We identified early issues with: - Class imbalance - Score clipping But fixed them quickly by adjusting the

			thresholds and checking distributions again.
Hour 7-8	Test and debug the initial version	Evaluated the model using accuracy, precision, recall, and confusion matrices. Debugged issues with preprocessing, model training, and misaligned labels. Adjusted the synthetic data balance to improve classification performance.	We evaluated the model with accuracy scores, confusion matrices, and classification reports. We found that High vs Medium categories were often confused so we revised the threshold logic in the risk function to strengthen separation. We tested multiple random seeds, fixed labeling inconsistencies, and confirmed that the model behaved consistently across runs. We also corrected a seaborn heatmap error and verified that the model generalizes well to unseen synthetic data. This debugging stage significantly improved reliability even though it took longer than expected.
Hour 9-10	Refine and add advanced features	Added visualizations for feature importance and model behavior.	We created a clear confusion matrix visualization and improved interpretability by examining model coefficients

		Built a clear and organized ipywidgets interface for interactive predictions. Improved UX with better labels, explanations, and layout. Wrote documentation explaining usage, prediction meaning, and ethical limitations. Polished the notebook with clearer structure and optional stylistic improvements.	to see which features mattered most. We added an ipywidgets interface that lets users adjust three sliders and instantly receive a risk prediction which makes the model feel more interactive. We also drafted an ethical discussion inside the notebook to explain the limitations of predicting academic risk since this kind of model can have real consequences.
Additional		Finished the full project write-up Completed recording of the project demonstration video, showing the interface, workflow, and final model performance.	We spent additional time reorganizing the report and preparing the video demonstration. We cleaned unused code cells, added comments throughout the notebook, and created the short video In the last few hours we mostly focused on making everything

			clear and making sure nothing important got overlooked.

Final Product Description:

i. Minimum Viable Product (MVP):

A Google Colab notebook that generates synthetic academic behavioral data, trains a logistic regression classifier, and uses an interactive widget to predict a student's academic risk as Low, Medium, or High.

ii. Target Product:

A polished Colab notebook with improved UX, feature-importance charts, clearer explanations for predictions, ethical documentation, and well-organized input controls using ipywidgets.

iii. Reach Version:

A Streamlit or Power Apps web application with authentication, saved prediction history, better dashboards, and the option to upload anonymized datasets for testing.

iv. Description of final product including target audience, user story, problem statement, key features, technical details and technologies used.

Our final product is a simple AI-based tool designed to predict a student's early academic risk using basic academic behaviors such as attendance, assignments submitted, study hours, and quiz performance as well as socioeconomic indicators like our "lunch status" var. It uses a logistic regression classifier trained on synthetic data, which makes sure that there are no privacy concerns. The target users include students, educators, and advisors who want early indicators of academic difficulty. The user story is fairly simple the user enters a few academic indicators, receives a "Low / Medium / High" risk prediction, and is shown basic insights about influencing factors. The project is built in Google Colab using Python, pandas, scikit-learn, and interactive

widgets. This MVP demonstrates how lightweight AI can support early intervention while remaining ethical, simple, and accessible.

v. Link to a video demo of product:

https://drive.google.com/file/d/1Da8R82eLFrU8gIX9x7jry3FdpA8LPixv/view

vi. Link to Google Collab work space:

BenjaminDillon-AymanSharieff-COIL.ipynb

All relevant README, Input files, and Code Files are included in the GC.

Consultation and Use of LLMs:

Consultation Description:

We briefly discussed project ideas with classmates, compared dataset options, and shared general debugging advice. We also looked at online documentation for scikit-learn, matplotlib, and ipywidgets. No external code was copied, but we did get conceptual guidance in terms of ideas from online resources on how to structure/format our UI/app as a whole

Use of LLMs:

The use of LLMs, including ChatGPT, Gemini, and Claude, were mostly limited to either idea generation, refining of text products (e.g. proposal/report text formatting, etc), and questions about concepts relating to how we should go about our unique implementation. LLMs were used briefly to help us debug/understand why particular sections of our code were failing/returning nonsense values

Ethical Considerations:

1. Data Provenance & Consent

All data used was synthetic and generated manually in the notebook. No real student data was collected, stored, or shared. No consent was required. Data is not persistent and is recreated on each run. Ownership: fully created by the developer.

2. Privacy & Security Risks

If applied in real scenarios, academic indicators could expose sensitive patterns. Mitigation includes anonymization, removing personal identifiers, restricting access, and not storing outputs. No sensitive attributes (gender, income, etc.) were included.

3. Fairness & Bias

ML models may unintentionally disadvantage certain groups. To mitigate this, no demographic data was used, and the model was evaluated to ensure consistent behavior across all synthetic inputs. Features were chosen carefully to avoid reinforcing inequities.

4. Misuse & Safety

Academic risk predictions could be misused to label or punish students. The project explicitly states the tool is supportive, not evaluative. Recommendations emphasize that it cannot replace human judgment.

5. Transparency & Accountability

The model's assumptions, limitations, and failure modes are clearly documented. A small Model Card and Data Sheet are included in the notebook. The developer maintains responsibility for improvements.

6. Cross-Cultural & Accessibility Considerations

In Ecuador, internet access and bandwidth may be limited, so Google Colab ensures accessibility. In the U.S., educational data privacy laws (FERPA) require stronger safeguards. UI language, simplicity, and low computational cost make the tool usable across both contexts.

7. Licensing & Sustainability

All tools and libraries used in this project—such as **pandas, NumPy, matplotlib, and scikit-learn**—are fully open-source and distributed under permissive licenses (BSD, MIT, or similar). No proprietary software or restricted datasets were required at any stage. The model operates efficiently on standard CPU hardware, ensuring low energy consumption and long-term reproducibility.

i. Completeness & Specificity

Our approach to sustainability focuses on four project-specific areas:

1. **Open Datasets**

   The core data comes from the publicly available *Students Performance in Exams* dataset (Kaggle). To avoid licensing concerns and ensure long-term reproducibility, we generated our own **synthetic risk labels**, making the final dataset fully redistributable in our GitHub repository.

2. **Open-Source Code & Pipeline Transparency**

   All code is stored in GitHub under an open license, with clear modular organization (code/, data/, tests/, docs/).

   Anyone can rerun the entire pipeline directly through Google Colab with no installation required.

3. **Lightweight Model = Low Computational Cost**

   Logistic Regression was intentionally selected because it is interpretable, sustainable, and computationally light.

   It trains in under a second and runs on any laptop or institutional computer, requiring no GPUs.

4. **Long-Term Maintainability**

   The pipeline avoids fragile dependencies, uses stable libraries, and loads data directly from GitHub rather than cloud storage. This ensures that future students or instructors can reuse or extend the project without compatibility or storage issues.

ii. Risk Analysis & Mitigation

We identified several realistic risks associated with educational ML systems and implemented concrete mitigations:

1. **Risk: Algorithmic Bias**

   Danger: Demographic variables could cause unfair or discriminatory predictions.

   Mitigation implemented: We removed all demographic attributes (gender, ethnicity, parental education, lunch type) and used only academic features (math, reading, writing scores).

   Evidence: The GitHub dataset and Colab notebook show that only academic numerical columns were used.

2. **Risk: Misinterpretation of Predictions**

   Danger: Users might wrongly assume that the model is authoritative or high-stakes.

   Mitigation implemented: Labels are fully synthetic and deliberately simple (Low/Medium/High).

   We clearly state that the system is a *demonstration*, not a real student assessment tool.

3. **Risk: Privacy Issues**

   Danger: Real student data might expose sensitive information.

   Mitigation implemented: No real institutional data was used; the dataset is anonymized and public, and the risk labels are fully artificial.

4. **Risk: Reproducibility Loss Over Time**

   Danger: Dependencies or storage links might break.

   Mitigation implemented:

   – All data stored as .csv on GitHub

   – Fixed library versions in the README

   – No reliance on Google Drive paths

iii. Cross-Cultural Insight

We compared sustainability and accessibility issues across **Ecuador and the United States**, identifying important contrasts:

- **Hardware & Connectivity Differences**

  Students in Ecuador may rely more on shared or low-spec devices, making lightweight models (like Logistic Regression) essential. U.S. institutions generally have greater access to campus computing resources, but also stricter compliance obligations (FERPA, privacy audits).

- **Open-Source Accessibility**

  Because all tools used are free and open-source, our project can be reused and taught in both countries without licensing barriers. This increases educational inclusiveness, especially in lower-resource environments.

- **Ethical Requirements Differ**

  U.S. institutions often require formal data governance for any ML use in education, while Ecuador faces practical constraints such as device availability and inconsistent internet access.

  Our fully offline-capable and lightweight model addresses both realities.

This analysis ensured our system remained accessible, fair, and ethically aware across both cultural contexts.