Coding Club
IIT Guwahati

Coding Week 25

MACHINE > < LEARNING

TASK RELEASED

{DEADLINE}

23RD MAY

Greetings

Hello there!

We hope you had a fun time going through the resources we shared. Don't worry if you couldn't complete everything — you can always refer back to them whenever needed while working on the tasks.

We've planned two tasks to evaluate your understanding, curiosity, and creativity:

- Task 1 – CampusPulse: Focuses on your ability to explore and analyze real-world student data. Can you look at raw, messy data and turn it into insights and predictions?
- Task 2 – The Rise of the WeatherMind: Tests your ability to implement machine learning systems, use tools, and build intelligent agents that respond like real-world AI assistants.

Good luck, and we hope this turns into an exciting and rewarding learning experience for you!

Setup Environment

This is related to setting up your VS-Code environment to enable you to complete the tasks in Jupyter Notebook, you can skip this if you have already jerry-rigged your Jupyter Notebook development environment.

1. Download VS-Code from here.
2. Install Python from here.
3. Install pip (Python's Package Manager)
    a. Windows: Find instructions here.
    b. Linux: Run in terminal: sudo apt install python3-pip
    c. Mac: Run in terminal:
        i. curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
        ii. python3 get-pip.py
4. Run in terminal:
    a. pip install ipykernel
    b. pip install notebook
5. Install the following VS-Code Extensions:
    a. Python
    b. Jupyter
    c. Jupyter Key Map
    d. Jupyter Notebook Renderers

Task 1 : The CampusPulse Initiative

You've been onboarded to CampusPulse, a strategic data project initiated by the Student Wellness and Experience Board at IIT Guwahati. Your objective is to analyze real, anonymized student data collected from a survey to help uncover the hidden factors that influence student life on campus , from grades and screen time to relationships, stress, and social behaviors.

The board has already hinted at one critical question they care about deeply:
Can we predict, from lifestyle and academic patterns, whether a student is likely to be in a romantic relationship?

But before you dive into predictive modeling, you'll need to clean the data, uncover insights, and identify a few intentionally anonymized features.

Level 1: Variable Identification Protocol 60 Points

As part of their privacy guidelines, three survey questions were anonymized. They've been renamed to Feature_1, Feature_2, and Feature_3. However, the board needs their identities recovered using statistical patterns , without exposing any raw labels.

Objective:

- Use EDA techniques (histograms, scatter plots, correlation heatmaps, etc.) to guess the original meaning of these features.
- Prepare short visual / textual justifications for each guess (e.g., "Feature_A likely represents weight due to its strong correlation with height").

This isn't just a guessing game — it's your first test in data forensics.

Level 2: Data Integrity Audit 30 Points

Survey data is messy by nature. Students may skip questions, input strange values, or leave inconsistent answers. Before CampusPulse can go live, the board expects you to ensure data hygiene.

Objective:

- Identify all features with missing values, propose and apply appropriate imputation strategies for each.
- Justify your choice briefly for each variable with nulls.

Level 3: Exploratory Insight Report 60 Points

CampusPulse isn't just a prediction tool — it's designed to help understand students. That starts with the right questions. You're required to conduct a focused EDA session on the dataset.

Objective:
Ask atleast 5 insightful questions about the student data. For each:

- Build a meaningful plot (bar chart, violin plot, scatterplot, etc.)
- Write a brief interpretation of the insight.

Your goal here is to showcase curiosity, clarity, and visual storytelling.

Level 4: Relationship Prediction Model 80 Points

Your goal is to uncover what patterns in academic, behavioral, or social data might signal a student's likelihood of being in a romantic relationship.

Build and evaluate predictive models using the cleaned dataset, and reflect on their effectiveness.

Objective:

- Apply classification techniques to model relationship likelihood and assess their performance. Think critically about what the models reveal, and what they don't.

This phase is where insight meets action.

Level 5: Model Reasoning & Interpretation 70 Points

The board isn't interested in "just predictions." They want transparency — to understand why your model makes decisions.

Objective:

- For each classifier, visualize decision boundaries using 2D feature pairs (choose any two meaningful features).
- Use SHAP to:
  - Plot global feature importance.
  - Generate local explanations for two students one predicted "Yes" and one "No."
- Interpret the results in plain language: what really drives relationship prediction?

Bonus Level: The Mystery Boundary Match 50 Points

You'll be given five unlabelled decision boundary plots generated from different classifiers. Your task is to identify which algorithm produced each plot, purely from visual clues like linearity, complexity, and shape of the boundary.

Objective:

- For each plot, name the model you believe generated it and explain your reasoning briefly.

This tests your intuition about how different models learn and separate data.

Task 2 : The Rise of the WeatherMind

You've been tasked to be in Project SkyLink, an elite initiative to construct a powerful AI entity the WeatherMind, capable of perceiving the world, interpreting human queries, and delivering precise, contextual responses through tool-enhanced reasoning.

Before you embark on this mission, it's crucial to understand the tools and ideas at the heart of your journey.

What is an LLM?

A Large Language Model (LLM) is a type of artificial intelligence trained on massive text datasets to understand and generate human-like language. It can answer questions, write essays, summarize content, translate languages, and more all using patterns learned from data. However, LLMs by themselves are passive responders. They wait for input and return output without context, without tools, and without autonomy.

What is Agentic AI?

Agentic AI goes one step further. Instead of being reactive, agentic systems are designed to:

- Make decisions
- Use tools and APIs,
- Plan actions over time
- Adapt to dynamic input and Remember Prior Context

In short, agentic AI is goal driven. It doesn't just respond, it acts, using its environment (tools, APIs, data) to achieve tasks intelligently. To learn more about the difference between LLM and Agentic AI click here.

Enter LangGraph

One of the most powerful ways to build such AI agents is using LangGraph, a tool that allows you to define graphs of stateful nodes, routing logic, and tool integration. It lets you go from an LLM that answers to an agent that thinks, chooses, and executes.

Drowning in info and freaking out over a new framework? Chill—documentation is basically your tech BFF. It's got you covered!

Your mission unfolds in four escalating levels. Each level tests your mastery over documentation reading which is very important in the world of coding. The future of intelligent assistants depends on your success.

Level 1: Core Activation 45 Points

Your first objective is simple but critical: boot the core.

- Construct a LangGraph node called chatbot, powered by an LLM (Gemini preferred, though any open-source model may be used).
- Integrate a companion tool: the calculator, capable of performing basic arithmetic and supports basic BODMAS rule.
- Visualize the awakening: Once your LangGraph is wired, render the graph as an image. We want to see the neural pathways—the AI's skeleton—coming to life.

This stage is about grounding WeatherMind, giving it a voice and a brain for math. Once complete, the AI awakens.

Level 2: Senses of the World 30 Points

The AI must now gain awareness of external domains. You'll install new senses:

- Fashion Recommender Tool: Given a query like "What's trending in Tokyo?", it should parse the location and surface real-time fashion trends.
- Weather Extractor Tool: Using a weather API, this tool must fetch and return current weather conditions based on the input query.

These tools extend WeatherMind's grasp beyond basic interaction, now it can perceive the world and respond with data-backed intelligence.

Level 3: Judgement and Memory 100 Points

Now comes the challenge of autonomy.

- Equip the WeatherMind with routing logic, the ability to determine which tool to use based on the intent of a user prompt. Should it calculate? Forecast weather? Recommend a look?
- Enhance the chatbot's capabilities with conversational memory, allowing it to remember and build on prior interactions, enabling flowing dialogue.

At this level, the WeatherMind becomes more than a reactive tool. It becomes an intelligent conversational agent, capable of reasoning and remembering.

Level 4: The Architect's Trial – Multi-Agent Evolution 175 Points

The AI must evolve again, this time into a collaborative intelligence.
You are to design a multi-agent system, more advanced than the typical planner-executor duo.

Your responsibilities:

- Define multiple agents, each with a distinct role, maybe researchers, decision-makers, analysts, it's up to you.
- Expand or redefine the toolset and even the core application (beyond just weather/fashion) to justify the architecture.
- Implement custom routing logic between agents, ensuring they collaborate smoothly, avoid conflict, and can fail gracefully.

Bonus: if everything runs seamlessly, including prompt chaining, agent cooperation, and tool execution, you'll earn top distinction.

For More Resources:

- How to use Gemini API
- How to get Weather API
- Videos on LangGraph
- Not Satisfied with Gemini and want to use other models? Use HuggingFace

Deliverables

You are required to submit a GitHub repository for each task containing:

- A well-structured Jupyter Notebook (.ipynb) with clear markdown headings for each level
- Clean, well-documented code with appropriate markdowns, comments and explanations
- Visualizations with proper labels and captions
- A README.md file briefly describing your approach and structure of the notebook
- A project report in PDF of both Tasks explaining your approach, experimentations and results.
- Remember, It's not just about the final code—we want to see how you got there. Show us your process, your experiments, and how you pushed for the best solution or accuracy you could reach.

Make sure the repository is public or has view access enabled, and include the GitHub link in your final submission form.

Deadline - 23rd May

Submission Link

Join the WhatsApp Group

Have any doubts ?
Contact us

Prakhar
+91 9305415358

Shirshendu Sen Gupta
+91 9674397828