Description:

(1) Find all kinds of online resources for learning about Machine Learning and Generative Artificial Intelligence. Articles, courses, videos, papers, blog posts, diagrams, podcasts, etc. It's a vast world. The list should attempt to summarize and categorize things, perhaps with an eye towards prerequisites and follow-ons.

(2) Train some kind of model to be able to "interview" their students, learning what they know and don't know. Also train the model to know about all the resources discovered in #1. Then have the model make recommendations and also help the student read, absorb, and understand each item, with an eye to the overall progress of the student.

Goal/Deliverable: The desired deliverable of this project would work as a smart tutor would and be able to guide a student and surface the next best possible course of action / module / material that the student should know. We envision achieving this using a "smart list", perhaps using mind-map software, or enabled by keyword (or RAGI) search, or recommending several pathways through the material.

Notes: As is, this project does not fit into the Unsupervised Machine Learning (UML) class. But there could be some adjustments that could make it fit.

1. **Knowledge discovery and clustering**: The goal here is to accomplish #1 above and create clusters of resources that represent distinct subdomains
    - Idea: Follow similar guidance from spec (#1) and collect all necessary information.
    - Use clustering techniques to group similar resources or topics based on features like (note Retrieval-Augmented Generation Interfaces (RAGI) as recommended in the write up are semi-supervised in nature), so if we tune this project to focus on unsupervised techniques then, a RAG system can be implemented using:
        - Text embeddings (e.g., BERT, SentenceTransformers).
        - Word vectors (e.g., Word2Vec, GloVe).
        - Topic modeling (e.g., LDA).
2. **Student Profiling**: This would cover spec #2.
    - Use unsupervised methods to group students based on their learning behaviors or quiz results. Techniques could include:
        - Clustering based on quiz performance or knowledge graph embeddings.
        - Using dimensionality reduction (e.g., PCA, t-SNE) to visualize learning trajectories.
3. **Student Resource Discovery with Topic Modeling**:
    - Apply topic modeling (e.g., LDA or Non-negative Matrix Factorization) to identify key topics across the corpus of resources.

    - Students could then:
        - Map their own learning gaps to these topics.
        - Recommend pathways based on clusters or topic coherence.
4. **Progress Analysis with Anomaly Detection**:
    - Use anomaly detection techniques (e.g., Isolation Forest) to identify unusual learning patterns or areas where a student struggles significantly compared to the cluster.

Project 1: Knowledge Clustering

Objective

Develop a system to group similar resources (e.g., articles, videos, or blog posts) into clusters based on their content. The goal is to discover meaningful groupings that can guide students in selecting resources by topic or difficulty level.

Tasks

1. Collect content consists of high-quality educational courses and materials about ML and GenAI.
    - **Example:** Identify ML/AI lectures on YouTube (e.g., Stanford Machine Learning Lectures) and generate transcripts. Then, extract and structure metadata from those transcripts.
    - **Example:** Locate ML/AI courses available online (e.g., Fast.ai), scrape the course content, and generate metadata.
    - **Example:** Find curated lists of high-quality ML/AI books, textbooks, or PDFs (e.g., Coursera's ML book recommendations), and extract detailed metadata for each.
    - **Example:** Compile lists of influential ML/AI research papers (e.g., Top ML Papers on GeeksForGeeks), retrieve full-text versions (where available, such as on Arxiv), and generate metadata.
2. Represent the data using unsupervised embeddings:
    - A major requirement here is to utilize Vector-database / RAG (Retrieval-Augmented Generation) approaches for storing and retrieving resources. You are expected to explore:
        i. Use **TF-IDF**, **Word2Vec**, or **Sentence-BERT** to generate vector representations.
        ii. Students are encouraged to experiment with different embedding models, including popular LLM-based embedding generators.
3. Perform clustering:
    - Experiment with algorithms like **k-means**, **DBSCAN**, or **hierarchical clustering**.

   - Students are encouraged to research more effective clustering algorithms: The idea is to treat this project as a research project and find out different techniques that better suit this type of project.

4. Visualize the clusters:
   - Use **t-SNE** or **UMAP** to create 2D visualizations of the clusters.
   - Note - these tools are dimensionality reduction techniques, the thought here is that the data will be intrinsically high dimensional and dimensionality reduction will be required.
   - Experiment with generating human-readable tags for clusters or features, as well as visualizing these clusters
5. Evaluate cluster coherence:
   - Students are expected to use metrics like Davies-Bouldin index,Gap Statistics Calinski-Harabasz index, Calabsz **silhouette score** or manual inspection to assess quality.
   - See a nice paper here: http://datamining.rutgers.edu/publication/internalmeasures.pdf
   -

Deliverables

- A through 6 - 8 page report will be required at the end of the semester
    - Summarizing the problem
    - Data collection techniques, data cleaning, transformations, normalization and standardization techniques used etc
    - A discussion of embedding techniques utilised
    - A discussion of clustering techniques
    - The methodology utilised in carrying out the research
    - Results of the research including visualization of clusters
    - Discussion on which clustering algorithm worked best and why - note for this to happen, students are encouraged to carry out statistical significance test on the raw results obtained from experimentation
- A final student presentation will be carried out on April 23rd