Assignment-1
Semantic Search on Movie Plots
Assignment Instructions

Due Date: August 26, 2025, 11:59 PM IST
Total Points: 25

1 Objective

Build a semantic search engine for movie plots using SentenceTransformers (`all-MiniLM-L6-v2`). You will complete a Jupyter notebook, host it in a GitHub repository, verify it with unit tests, and submit it via Google Classroom.

2 Deliverables

- A GitHub repository containing your completed notebook and required files.
- The repository URL submitted via Google Classroom.

3 Resources Provided

- Template code and dataset: https://github.com/srinidhi151/AI-Systems-Development--IIIT-blob/main/Assignment-1/movie_search.py
- Template GitHub repository: https://github.com/srinidhi151/AI-Systems-Development--III blob/main/Assignment-1
- Unit tests: Included in the template repository (`tests/test_movie_search.py`)

4 Step-by-Step Instructions

4.1 Set Up Your Environment

1. **Install Prerequisites**:
   - Ensure Python 3.9+ is installed (https://www.python.org).
   - Install Git (https://git-scm.com) for version control.
   - Install Jupyter Notebook: `pip install notebook`.
2. **Create a GitHub Account** (if you dont have one):
   - Go to https://github.com, sign up, and verify your email.
   - Add a professional profile picture and bio (e.g., "Student learning data science").
3. **Fork or Create a Repository**:

- Option 1: Fork the template repository: https://github.com/your-username/movie-search-template. Click **Fork**.
- Option 2: Create a new repository:
  - Click + > **New repository** on GitHub.
  - Name it `movie-search-assignment`.
  - Set to **Public** (or private with instructor access: add `[instructor-username]` as a collaborator).
  - Initialize with a `README.md` and select the **Python** `.gitignore` template.

4. **Clone the Repository Locally**:
   - Open a terminal.
   - Navigate: `cd path/to/your/folder`.
   - Clone: `git clone https://github.com/your-username/movie-search-assignment.git`.
   - Enter folder: `cd movie-search-assignment`.
5. **Set Up a Virtual Environment**:
   - Create: `python -m venv venv`.
   - Activate: `venv\Scripts\activate` (Windows) or `source venv/bin/activate` (macOS/Linux).
   - Install dependencies: `pip install -r requirements.txt`.

4.2 Complete the Assignment

1. **Download Files**:
   - Download the files from the template repository (https://github.com/srinidhi151/AI-Systems-Development--IIIT-Naya-Raipur/blob/main/Assignment-1).
   - Copy them to your repository folder.
2. **Follow the Notebook Instructions**:
   - Complete the five sections:
     (a) Install and import libraries (`sentence-transformers`, `pandas`, `scikit-learn`).
     (b) Load `movies.csv` into a pandas DataFrame.
     (c) Create embeddings using `all-MiniLM-L6-v2`.
     (d) Implement `search_movies(query, top_n)` to return a DataFrame with top `top_n` movies based on cosine similarity.
     (e) Test with query `'spy thriller in Paris'`.

- Add comments to explain your code.

3. **Update README.md**:
   - Edit `README.md` to include:
     - Project overview (e.g., "This is my solution for the semantic search assignment").
     - Setup instructions: How to install dependencies and run the notebook.
     - Testing instructions: How to run unit tests.
     - Usage example: `search_movies('spy thriller in Paris')`.
   - Example:

```
# Movie Semantic Search Assignment

This repository contains my solution for the semantic search on movie plots ass

## Setup
1. Clone: `git clone https://github.com/your-username/movie-search-assignment.g
2. Create virtual environment: `python -m venv venv` and activate it.
3. Install dependencies: `pip install -r requirements.txt`
4. Run notebook: `jupyter notebook movie_search_solution.ipynb`

## Testing
Run: `python -m unittest tests/test_movie_search.py -v`

## Usage
Test the function: `search_movies('spy thriller in Paris')`
```

4.3 Verify with Unit Tests

1. **Ensure Tests Are Included**: Use `tests/test_movie_search.py` from the template.
2. **Run Tests Locally**: Run `python -m unittest tests/test_movie_search.py -v`.
3. **Fix Errors**: Check output for failures and debug (e.g., missing `movies.csv`, incorrect DataFrame columns).

4.4 Commit and Push to GitHub

1. **Stage Changes**: `git add .`
2. **Commit**: `git commit -m "Completed semantic search assignment with passing tests"`.
3. **Push**: `git push origin main`.

4. **Check GitHub Actions**: In the repositorys **Actions** tab, confirm tests passed (green checkmark).

4.5 Submit to Google Classroom

1. **Copy Repository URL**: From https://github.com/your-username/movie-search-assignment
2. **Submit**:
   - Open the assignment in Google Classroom.
   - Click **Turn in** > **Add** > **Link**.
   - Paste the URL and add a note (e.g., "All 4 unit tests passed locally").
   - Click **Turn in**.
3. **Double-Check**: Ensure the repository is public or Im added as a collaborator.

5 Grading Rubric (25 Points)

Criteria	Points	Description
Unit Tests Passing	15	3.75 points per passing test (4 tests: output format, top_n, similarity, relevance)
Code Quality	5	Clean, commented code in notebook and `movie_search.py`; proper error handling
Documentation & GitHub Submission	5	Clear `README.md` and notebook markdown explaining your solution; correct repository setup and URL submission

6 Tips for Success

- **Test Early**: Run unit tests after each code change.
- **Comment Code**: Explain steps (e.g., "Encoding query with MiniLM model").
- **Troubleshooting**:
  - Git errors: Use `git status` or ask for help.
  - Test failures: Read error messages; compare with prompts.
  - Dependencies: Ensure `requirements.txt` is correct.
- **Backup**: Save and commit regularly (use GitHub Codespaces if needed).
- **Ask for Help**: Post in Google Classroom or email [instructor email].

7 Expected Repository Structure

```
movie-search-assignment/
 tests/test_movie_search.py        # Unit tests
 movie_search.py                   # Python module
 movies.csv                        # Dataset
 requirements.txt                  # Dependencies
 README.md                         # Instructions
 .gitignore                        # Ignore unnecessary files
```

8 Submission Details

Deadline: August 26, 2025, 11:59 PM IST. Late submissions may incur a 10% per day penalty.
Questions? Post in Google Classroom or email [instructor email]. Good luck, and have fun building your search engine! This project will enhance your GitHub portfolio.