Project : Collaborative Sentiment Analysis Pipeline Introduction

Overview

In this project, pairs of students will work together to build a sentiment analysis pipeline using a BERT model. The pipeline is broken into three major components:

- Data Extraction: Load and prepare raw text data.
- Data Processing: Clean and tokenize the text, converting it into the format required for BERT.
- Model Training & Inference: Fine-tune a pretrained BERT model for sentiment classification and create an inference script.

Collaboration will be achieved through Trello Board, Communication app, branching, pull requests, code reviews, and a shared continuous integration (CI) setup. Tasks

Goal Repository Structure

Key Details & Tasks

Trello Board Integration

Each pair of students will create a **Trello board** named **"Sentiment Analysis Project – [Student A & Student B]"**

The board will include the following **lists** representing stages of a real project workflow:

1. **To Do**
    - Contains all assigned tasks that are not yet started.
    - Each task corresponds to a card (e.g., *"Implement Data Loader Function"*).

2. **In Progress**
    - Tasks currently being worked on.
    - Cards moved here must have clear ownership (assign a member).
3. **In Review**
    - Tasks awaiting code review.
    - The reviewer (partner) comments or requests changes before merging.
4. **Done**
    - Completed and validated tasks.

Example Trello Board Structure

List	Example Cards
To Do	- Setup project repo - Define data extraction function - Write data cleaning plan
In Progress	- Implement tokenization logic - Fine-tune model
In Review	PR #3: Add unit tests for preprocessing
Done	- Project setup - GitHub workflow integration

Each **card** should include:

- **Description:** A brief summary of the task.
- **Checklist:** Subtasks or acceptance criteria.
- **Attachments:** Related code snippets, PR links, or test coverage reports.
- **Labels:** Use labels like `backend`, `data`, `model`, `testing`, or `documentation`.

Communication Tools

Students must collaborate using **Microsoft Teams**, **Slack**, or another group communication platform.

Data Extraction (Student 1)

Tasks

- Write functions in data_extraction.py to load the raw data provided.
- Ensure the function handles errors (missing files, wrong formats) gracefully.
- Testing:

- Create unit tests (tests/unit/test_data_extraction.py) that verify the data is loaded correctly, the DataFrame has expected columns, and edge cases are handled.

Data Processing (Student 1 & Student 2)

Tasks

- Implement text cleaning and preprocessing in data_processing.py similar to the Kaggle notebook approach:
  - Remove unnecessary characters, lower-case conversion, and normalization.
  - Tokenize text using the Hugging Face AutoTokenizer (e.g., for "bert-base-uncased").
  - Include splitting of data into training and validation sets.
  -
- Testing:
  - Develop tests in tests/unit/test_data_processing.py to ensure the tokenization produces the expected token IDs, and that text cleaning works as intended.

Model Training & Inference (Student 2 or Joint Effort)

Tasks

- In model.py, load a pretrained BERT model for sequence classification (for example, using AutoModelForSequenceClassification from Hugging Face).
- Fine-tune the model on your sentiment dataset. Use a simple training loop or the Hugging Face Trainer API.
- Create inference.py to allow users to pass in new text and see sentiment predictions.
- Testing:
  - Write tests in tests/unit/test_model.py that at least instantiate the model and run a dummy batch through it to ensure it outputs logits with the expected shape.
  - Test the end-to-end inference process in tests/unit/test_inference.py.

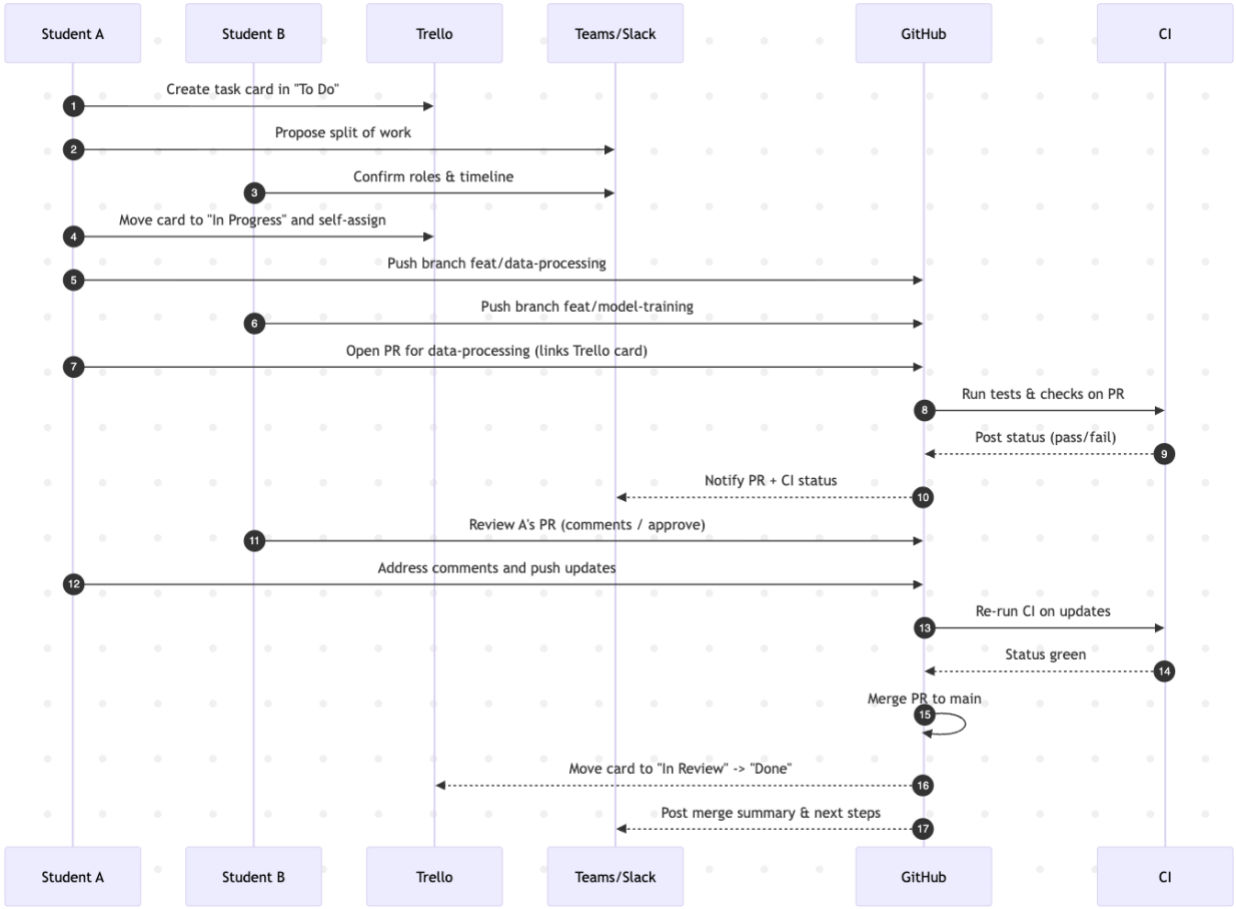
Collaboration

The project involves sequential tasks but can still allow parallel collaboration as follows (example):

Phase	Main Task	Lead Student	What the Other Student Can Do in Parallel
1. Data Extraction	Load & validate data	Student 1	Review dataset structure, draft preprocessing plan
2. Data Processing	Cleaning, tokenization	Both	Work on tokenizer setup in parallel, one cleans data, other implements tokenization
3. Model Training	Fine-tune BERT	Student 2	Student 1 prepares model evaluation metrics & test scripts

4. Inference Script	Build inference pipeline	Student 2	Student 1 writes documentation and runs test inference cases
5. Testing & Report	Final verification	Both	Each reviews the other's test cases and prepares report sections

Example of collaboration using a sequence diagram



Git Workflow

Branching

- Student A works on feature-data-extraction and part of feature-data-processing.
- Student B focuses on feature-model-training (and possibly an inference branch).

Pull Requests

- Each feature branch must be merged into main/master via a pull request.
- Code reviews are mandatory: each student must review and comment on their partner's PR.

Commit messages

- Use descriptive messages that clearly reference the task (e.g., "Implement CSV data loader for sentiment data", "Add tokenization function using AutoTokenizer")

Deliverables

- **Public GitHub Repository URL -** The repository should reflect the structure above with evidence of collaboration (branching, PRs, commit history).
- **Documentation -** A comprehensive README.md that includes setup instructions, usage examples, and a brief description of each component.
- **Project Report -** Overview of:
  - the chosen approach (inspired by the Kaggle notebook),
  - division of labor,
  - Trello board screenshots,
  - Github screenshots of commits and PRs
  - challenges faced,
  - and future improvements.

⚠ Do not forget to add your names.

Resources

Sentiment Analysis using BERT	https://www.kaggle.com/code/prakharrathi25/sentiment-analysis-using-bert
Github CheatSheet	https://education.github.com/git-cheat-sheet-education.pdf
Github Hello World (step-by-step guide)	https://docs.github.com/en/get-started/start-your-journey/hello-world
Github PRs	https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/about-pull-request-reviews
Trello Introduction	https://trello.com/guide/trello-101#what-is-the-board-menu