Project: Bias Detection and Mitigation in AI Training Data

Phase 5: Deployment Process

 Objective

To deploy the bias detection and mitigation pipeline as a standalone, reproducible application that can be integrated into production workflows or used for batch processing of training data.

 Deployment Steps

1. **Prepare the Environment**
   - Set up a Python virtual environment:

     python -m venv venv

     source venv/bin/activate (or venv\Scripts\activate on Windows)
   - Install dependencies:

     pip install -r requirements.txt

2. **Structure the Application**
   - `main.py`: Contains the pipeline for bias detection and mitigation.
   - `bias_detection/`: Module with `BiasDetector` class.
   - `bias_mitigation/`: Module with `BiasMitigator` class.

- `data/`: Folder to store datasets.

- `reports/`: Output folder for bias and mitigation reports.

3. **Model Integration (Optional)**

   - If using an ML model, wrap pre- and post-processing steps in the same pipeline.

   - Include model prediction after mitigation (if needed).

4. **Run the Pipeline**

   - Execute main.py:

   python main.py

   - Outputs:

     - Bias detection report

     - Mitigation summary

     - Logs/statistics for review

5. **Testing and Validation**

   - Use Phase 4 test cases.

   - Ensure reproducibility and accuracy of outputs.

   - Monitor correlations and distribution changes.

6. **Packaging for Reuse**

   - Create a CLI or API wrapper if desired.

   - Package modules using setuptools for installation.

7. **Optional: Containerization**

   - Use Docker for consistent deployment:

   docker build -t bias-pipeline .

```
docker run bias-pipeline
```

8. **Optional: Cloud Integration**

   - Deploy on platforms like AWS Lambda, Google Cloud Functions, or Azure Functions for serverless workflows.

   - Schedule jobs with cloud-based schedulers for automated periodic bias detection.

 Notes

- Logs and reports should be saved with timestamps.

- Ensure compliance with data privacy and fairness auditing standards.

Status: Ready for integration with user-facing tools or pipelines.