Software Requirements Specification (SRS) for Ethical AI Validator

Muhammad Abdullah

August 02, 2025

1 Introduction

1.1 Purpose

The Ethical AI Validator is a Python pip package aimed at auditing AI models for bias, fairness, and compliance with regulations (e.g., GDPR, AI Act). This SRS defines the requirements to guide development, ensuring the library supports AI engineers and organizations in building ethically sound models.

1.2 Scope

The library targets AI practitioners and organizations deploying production-ready models, focusing on bias detection, fairness metrics, and compliance reporting. It will be open-source, hosted on GitHub (whis-19), and distributed via PyPI, with initial support for common datasets and future expansion to real-time monitoring.

1.3 Definitions, Acronyms, and Abbreviations

- **AI**: Artificial Intelligence
- **GDPR**: General Data Protection Regulation
- **FR**: Functional Requirement
- **NFR**: Non-Functional Requirement
- **MVP**: Minimum Viable Product
- **SHAP**: SHapley Additive exPlanations

1.4 References

- Fairlearn Documentation: https://fairlearn.org/
- GDPR Guidelines: https://gdpr.eu/
- PyPI Packaging Guide: https://packaging.python.org/

1.5 Overview

This SRS includes overall description, system features, external interface requirements, non-functional requirements, documentation, and a roadmap.

2 Overall Description

2.1 User Needs

- AI developers need tools to detect and mitigate bias in models.
- Organizations require compliance reports for regulatory adherence.
- Teams seek real-time ethical monitoring for deployed systems.

2.2 Assumptions and Dependencies

- Users have Python 3.8+ and Scikit-learn installed.
- Access to model predictions and training data is available.
- Development relies on GitHub and PyPI for distribution.

2.3 System Context

The library operates as a Python module, analyzing model outputs and datasets, and generates reports for stakeholders.

3 System Features (Functional Requirements)

makecell

3.1 Functional Requirements Table

ID	Feature	Description	Input	Output
FR-001	Bias Detection	Identify bias in model predictions (e.g., gender, race).	Model predictions, true labels	Bias report (e.g., disparity)
FR-002	Fairness Metrics	Calculate fairness metrics (e.g., equal opportunity).	Predictions, protected attributes	Metric scores (dictionary)
FR-003	Compliance Audit	Generate report for GDPR/AI Act compliance.	Model metadata, audit criteria	PDF/HTML compliance report
FR-004	Real-Time Monitoring	Monitor bias in real-time for deployed models.	Streaming predictions	Dashboard update or alert
FR-005	Mitigation Suggestions	Suggest bias mitigation techniques (e.g., reweighting).	Bias report	List of mitigation strategies

ID	Feature	Description	Input	Output
FR-006	Unit Testing	Test all functions with `pytest` for reliability.	Test cases	Pass/fail results (80%+ code coverage)

4 External Interface Requirements

4.1 User Interfaces

- **CLI**: No direct CLI; imported in Python scripts.
- **Example Usage**:

```
from ethical_ai import Validator
validator = Validator(model, data)
report = validator.audit_bias()
```

4.2 Hardware Interfaces

Runs on standard PCs with Python; no special hardware required.

4.3 Software Interfaces

- **Python**: 3.8, 3.9, 3.10, 3.11
- **Dependencies**: `scikit-learn (>=1.0)`, `pandas (>=1.3)`, `reportlab` (for PDF)
- **External APIs**: None (standalone)

4.4 Communications Interfaces

File I/O for report generation (e.g., PDF output).

5 Other Non-Functional Requirements

5.1 Performance Requirements

- Process datasets up to 10,000 samples in under 5 seconds.
- Real-time monitoring handles 100 predictions per minute.

5.2 Security Requirements

- Protect sensitive data (e.g., protected attributes) with anonymization options.
- Ensure compliance report data is non-exportable without consent.

5.3 Quality Attributes

- **Reliability**: 98% accuracy in bias detection on test datasets.
- **Maintainability**: Modular code with clear documentation.
- **Usability**: Intuitive API with examples in README.

5.4 Constraints

- MVP limited to static analysis and basic compliance.
- No support for deep learning models in initial release.

6 Documentation

6.1 User Documentation

- **README.md**: Installation, usage examples, and API reference.
- **Docstrings**: Function details with parameters and returns.

6.2 Developer Documentation

- **Setup Guide**: Contribution, testing, and packaging instructions.
- **Test Suite**: In `tests/` with `pytest`.

7 Appendix

7.1 Glossary

- **Bias**: Systematic error in model predictions affecting certain groups.
- **Fairness**: Equitable treatment across protected attributes.

7.2 Timeline

- **Total Duration**: 6 weeks (Aug 02 - Sep 12, 2025)
- **Estimated Hours**: 35–50 hours

makecell longtable booktabs

7.3 Project Roadmap

Phase	**Timeline**	**Milestones / Deliverables**
Phase 1: Planning	Week 1–2	• Requirement gathering with stakeholders • Define objectives and constraints • Literature review
Phase 2: Design	Week 3–4	• System design and architecture diagrams • UI wireframes • Select libraries and frameworks
Phase 3: Development	Week 5–10	• Core module development • Integrate bias detection and fairness metrics • Test with sample datasets
Phase 4: Testing	Week 11–12	• Unit testing using `pytest` • Evaluation of accuracy and fairness • Refinement based on feedback
Phase 5: Documentation	Week 13	• Write user and developer documentation • Create usage examples
Phase 6: Deployment	Week 14	• Deploy on test server • Final bug fixes • Project presentation/demo