Product Requirements and Specification Document

Project Name

ExplainX - Model Interpretability Dashboard

Description

ExplainX is an open-source Streamlit dashboard for visualizing and interpreting machine learning (ML) model predictions using SHAP and LIME. Users can upload models and datasets, view global feature importance, and generate local explanations for individual predictions. The tool is designed for educational and practical use, supporting common ML workflows.

---

1. Goals & Objectives

Goal	Description
Model Interpretability	Enable users to understand ML model predictions
Accessibility	Provide an easy-to-use, web-based interface
Educational Value	Support learning and teaching of model explainability concepts
Open Source	Ensure code is publicly available and well-documented

---

2. Target Users

- Data scientists and ML practitioners
- Students and educators in ML/AI
- Developers seeking model transparency

---

3. Core Features

Feature	Description
Model Upload	Upload pre-trained scikit-learn models (`.pkl` format)
Dataset Upload	Upload datasets in CSV format
Model Summary	Display model type, input features, and basic statistics
Global Feature Importance	Visualize feature importance using SHAP and LIME
Local Explanation	Generate SHAP and LIME explanations for individual predictions
Interactive Selection	Select data points for local explanation via UI
Downloadable Reports	Export explanation results as images or CSV
Documentation & Help	In-app guidance on usage and interpretability concepts

4. Non-Functional Requirements

Requirement	Specification
Performance	Explanations generated within 5 seconds (typical)
Usability	Intuitive UI, minimal configuration
Compatibility	Python 3.8+, Streamlit, scikit-learn models
Security	No data/model storage on server
Open Source	MIT License, public GitHub repository

5. Technical Specifications

5.1 Technology Stack

Component	Technology
Frontend	Streamlit
Backend	Python
ML Libraries	scikit-learn, SHAP, LIME, pandas, numpy

5.2 Data & Model Handling

- **Model Input:** Accepts scikit-learn models serialized with `joblib` or `pickle`
- **Dataset Input:** CSV files; auto-detects feature columns
- **Data Validation:** Checks for feature alignment between model and dataset

5.3 Explanation Methods

- **Global:** SHAP summary plot, LIME feature importance
- **Local:** SHAP force plot, LIME explanation for selected row

5.4 UI/UX

- **Navigation:** Sidebar for uploads and settings
- **Main Area:** Tabs for model summary, global, and local explanations
- **Interactivity:** Data point selection via table or index input

6. User Flow

```
graph TD
    A[Start] --> B[Upload Model]
    B --> C[Upload Dataset]
    C --> D[View Model Summary]
    D --> E[View Global Feature Importance]
    E --> F[Select Data Point]
```

```
    F --> G[View Local Explanation]
    G --> H[Download Report]
```

---

7. Acceptance Criteria

ID	Criteria
AC1	User can upload a scikit-learn model and dataset without errors
AC2	Model summary and feature list are displayed after upload
AC3	Global feature importance plots are generated using SHAP and LIME
AC4	User can select a data point and view local explanations (SHAP, LIME)
AC5	Explanations can be exported as images or CSV
AC6	In-app documentation is accessible and clear

---

8. Out of Scope

- Deep learning model support (e.g., TensorFlow, PyTorch)
- Real-time model training or editing
- Persistent user accounts or data storage

---

9. Milestones & Timeline

Milestone	Target Date
Project Setup	Week 1
Model/Dataset Upload	Week 2
Global Explanations	Week 3
Local Explanations	Week 4
UI/UX Polish & Docs	Week 5
Release & Documentation	Week 6

---

10. Appendix

Example Usage

```
import streamlit as st
import shap
import lime
```

```
import pandas as pd
# Pseudocode for loading model and dataset, generating explanations
```

End of Document