High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **Klassify - Multi-Model Classification Explorer**. Klassify is an interactive Streamlit application enabling users to upload datasets, explore and compare KNN, SVM, and Decision Tree classifiers, tune hyperparameters, visualize decision boundaries, and evaluate models using standard metrics.

---

1. System Architecture Overview

Architecture Description:
Klassify is a modular, client-server web application built with Streamlit. The frontend provides an interactive UI for data upload, parameter tuning, and visualization. The backend handles data processing, model training, evaluation, and visualization generation.

Main System Modules and Roles:

Module	Role
User Interface (UI)	Streamlit-based UI for user interaction, data upload, and visualization
Data Handler	Loads, validates, and preprocesses user datasets
Model Manager	Initializes, trains, and manages KNN, SVM, and Decision Tree models
Hyperparameter Tuner	Provides UI controls and logic for model hyperparameter adjustment
Evaluation Engine	Computes confusion matrix, ROC, F1 scores, and other metrics
Visualization Engine	Generates plots for decision boundaries and evaluation metrics

---

2. Component Interactions

Sequence Step	Interaction Description
1	User uploads dataset via UI
2	Data Handler loads and preprocesses data
3	User selects classifier and tunes hyperparameters via UI
4	Model Manager trains selected model with current parameters
5	Evaluation Engine computes metrics on test data
6	Visualization Engine generates and displays decision boundaries and metric plots in UI

---

3. Data Flow Overview

Data Source	Flow Direction	Destination	Description
User Dataset Upload	UI → Data Handler	Data Handler	User uploads CSV; data is loaded and cleaned
Preprocessed Data	Data Handler → Models	Model Manager	Data split for training/testing
Model Output	Model Manager → Eval	Evaluation Engine	Predictions and model state
Metrics/Plots	Eval/Vis Engine → UI	User Interface	Results and visualizations displayed

---

4. Technology Stack

Layer/Function	Technology/Framework
Frontend/UI	Streamlit
Machine Learning	scikit-learn
Data Processing	pandas, numpy
Visualization	matplotlib, seaborn, Streamlit built-ins
Environment	Python 3.x

---

5. Scalability & Reliability

- **Scalability:** Designed for single-user, interactive educational use; can be containerized for deployment. For larger datasets or concurrent users, consider backend task queues or horizontal scaling.
- **Reliability:** Input validation and error handling ensure robust operation. Stateless design allows for easy restarts and updates.
- **Security:** User data is processed in-memory and not persisted; recommend deploying in secure, isolated environments.

End of Document