



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the implementation details for **Klassify - Multi-Model Classification Explorer**. Klassify is an interactive Streamlit application enabling users to upload datasets, tune hyperparameters, visualize decision boundaries, and compare KNN, SVM, and Decision Tree classifiers using standard evaluation metrics.

1. System Components

Component	Description / Responsibility
UI Layer (Streamlit)	User interaction, file upload, parameter selection, display
Data Handler	Data loading, validation, preprocessing
Model Manager	Model selection, training, prediction, evaluation
Visualization Engine	Plotting decision boundaries, confusion matrix, ROC curves
Metrics Calculator	Computes F1, accuracy, confusion matrix, ROC

2. Class/Interface Overview

Class/Module	Key Methods / Attributes	Relationships
DataHandler	load_data() , preprocess() , validate()	Used by UI, ModelManager
ModelManager	train() , predict() , set_params() , get_model()	Uses scikit-learn models
MetricsCalculator	compute_f1() , compute_roc() , confusion_matrix()	Used by ModelManager
VisualizationEngine	plot_decision_boundary() , plot_confusion_matrix() , plot_roc_curve()	Used by UI

Key Methods Example:

```
class ModelManager:
    def set_params(self, model_type, params): ...
    def train(self, X_train, y_train): ...
    def predict(self, X_test): ...
    def get_model(self): ...
```



3. Data Structure Overview

Data Model	Structure / Fields
Uploaded Dataset	Pandas DataFrame: features (X), labels (y)
Model Parameters	Dict: {param_name: value} per model
Evaluation Results	Dict: { 'f1': float, 'roc': array, 'conf_matrix': array }

4. Algorithms / Logic

Main Flow Pseudocode:

```
# UI Layer
dataset = DataHandler.load_data(uploaded_file)
X_train, X_test, y_train, y_test = DataHandler.preprocess(dataset)
model = ModelManager.set_params(selected_model, user_params)
ModelManager.train(X_train, y_train)
y_pred = ModelManager.predict(X_test)
metrics = MetricsCalculator.compute_all(y_test, y_pred)
VisualizationEngine.plot_decision_boundary(model, X_test, y_test)
VisualizationEngine.plot_confusion_matrix(metrics['conf_matrix'])
VisualizationEngine.plot_roc_curve(metrics['roc'])
```

5. Error Handling

Scenario	Handling Approach
Invalid file format	Show error message, prompt re-upload
Missing/invalid data columns	Display validation error, request correction
Model training failure	Catch exception, display error to user
Visualization errors	Fallback to text message, log error
Unsupported parameter values	Validate input, show warning

End of Document