Product Requirements & Specification Document

Project Name

StackVote - Ensemble Learning Playground

Description

StackVote is an open-source Streamlit application enabling users to experiment with ensemble learning techniques—stacking, blending, and voting classifiers. Users can upload datasets, select and configure base models, tune hyperparameters, and visualize ensemble performance compared to individual models. The app is designed for educational purposes and rapid prototyping.

---

1. Goals & Objectives

Goal	Description
Interactive Ensemble Learning	Allow users to build, configure, and compare ensemble classifiers
Educational Focus	Provide clear visualizations and explanations
Open-Source Accessibility	Ensure code is reusable and extensible

---

2. Target Users

- Students and educators in machine learning
- Data science enthusiasts
- Developers prototyping ensemble models

---

3. Core Features

Feature	Description
Dataset Upload	Upload CSV datasets (tabular, <50MB)
Data Preview	Display sample rows, summary statistics, and class distribution
Model Selection	Choose base models (e.g., Logistic Regression, Decision Tree, SVM, KNN)
Ensemble Methods	Select stacking, blending, or voting
Hyperparameter Tuning	Adjust key parameters for each base model
Training & Evaluation	Train models, display metrics (accuracy, F1, ROC-AUC, confusion matrix)
Visualization	Compare ensemble vs. individual model performance (charts, tables)
Download Results	Export trained model and evaluation metrics

---

4. Functional Requirements

4.1 Dataset Handling

- Accept CSV file upload (tabular, <50MB)
- Parse and validate data (handle missing values, categorical encoding)
- Display data preview and summary

4.2 Model & Ensemble Configuration

- List available base models (scikit-learn classifiers)
- Allow selection of 2-5 base models
- Enable hyperparameter adjustment (via Streamlit widgets)
- Support selection of ensemble method:
  - Stacking (with meta-learner selection)
  - Blending
  - Voting (hard/soft)

4.3 Training & Evaluation

- Split data into train/test (user-configurable ratio)
- Train individual base models and ensemble
- Compute and display:
  - Accuracy, F1-score, ROC-AUC (if applicable)
  - Confusion matrix
  - Cross-validation scores (optional)

4.4 Visualization

- Plot performance comparison (bar/line charts)
- Show confusion matrices
- Display ROC curves (if applicable)

4.5 Export

- Allow download of:
  - Trained ensemble model (pickle)
  - Evaluation metrics (CSV/JSON)

---

5. Non-Functional Requirements

Requirement	Specification
Performance	Responsive UI for datasets up to 50MB
Usability	Intuitive, minimal-click workflow
Compatibility	Python 3.8+, Streamlit, scikit-learn, pandas, numpy
Security	No code execution from uploaded files
Documentation	Clear user guide and code comments

6. Technology Stack

Component	Technology
Frontend	Streamlit
Backend	Python
ML Framework	scikit-learn
Data Handling	pandas, numpy

7. User Flow

```
graph TD
A[Upload Dataset] --> B[Preview & Configure Data]
B --> C[Select Base Models & Tune Params]
C --> D[Choose Ensemble Method]
D --> E[Train & Evaluate]
E --> F[Visualize Results]
F --> G[Download Model/Metrics]
```

8. UI Wireframe (Textual)

- **Sidebar:** Dataset upload, model/ensemble selection, parameter tuning
- **Main Area:** Data preview, training progress, evaluation metrics, visualizations, download buttons

9. Acceptance Criteria

ID	Criteria
AC1	User can upload and preview a CSV dataset
AC2	User can select and configure at least 2 base models
AC3	User can choose stacking, blending, or voting ensemble
AC4	App trains models and displays comparative metrics and visualizations
AC5	User can download trained model and evaluation results

10. Out of Scope

- Regression tasks (classification only)
- Deep learning models
- Real-time data streaming

11. Milestones

Milestone	Description
M1: Data Upload & Preview	Dataset handling and preview
M2: Model Selection	Base model and ensemble config
M3: Training & Evaluation	Model training and metrics display
M4: Visualization	Performance and comparison charts
M5: Export & Documentation	Download features, user guide

---

12. Example Code Snippet

```
from sklearn.ensemble import StackingClassifier, VotingClassifier
from sklearn.model_selection import train_test_split
# Load data, split, fit base models, create ensemble, evaluate
```

---

13. License

- MIT License (open-source)

---

14. References

- scikit-learn Ensemble Methods
- Streamlit Documentation
