Machine Learning Model Explorer (Streamlit Mini Project)

Project Overview

You are required to build an interactive **Machine Learning Model Explorer** using **Streamlit**. The purpose of this project is to help students understand and experiment with various **Supervised** and **Unsupervised** learning algorithms in a simple and visual way.
Users should be able to upload datasets, preprocess them, select models, train and evaluate results — all through an intuitive web dashboard.

Key Python Concepts Used

- **NumPy & Pandas:** Data cleaning, preprocessing, and manipulation
- **Scikit-learn:** Implementation of ML models and evaluation metrics
- **Matplotlib & Seaborn:** Visual representation of results
- **Streamlit:** Building an interactive dashboard interface
- **OOP Concepts:** Understanding models as objects
- **Data Preprocessing:** Encoding, scaling, and handling missing values

Project Requirements

1. Create a **Streamlit web application** with sidebar navigation.
2. Include both **Supervised** and **Unsupervised** learning modules.
3. Allow users to **upload a dataset (CSV format)** and preview it.
4. Handle **missing values, categorical encoding, and feature scaling** automatically.
5. Provide options to select models and **tune hyperparameters** interactively.
6. Implement **training, evaluation, and visualization** for each model.
7. Show clear performance results and visual outputs.

Core Features

- **Dataset Upload:** Accept CSV file input and display sample rows.
- **Automatic Preprocessing:** Handle missing data, encode categorical columns, and scale features.
- **Model Selection:**
  - Choose between **Supervised** and **Unsupervised** algorithms.
  - Configure model parameters via sidebar controls.
- **Training & Testing:** Train models and display evaluation results dynamically.
- **Visualization:**
  - Confusion matrix for classification models
  - Cluster scatter plots for clustering models
- **Performance Metrics:** Show accuracy, classification report, or silhouette score as applicable.

Algorithms Included

Supervised Learning

Algorithm
Decision Tree Classifier
Random Forest Classifier
Support Vector Machine (SVM)

Unsupervised Learning

Algorithm
KMeans
Agglomerative Clustering
DBSCAN

Input Validation Rules

- Dataset must be a valid **CSV file**.
- Target column must be selected for supervised tasks.
- Only **numeric data** is accepted for training models.
- Clear warning messages should appear for missing or invalid inputs.

Expected Streamlit Dashboard

Sidebar Controls:

- Select **Learning Type:** Supervised / Unsupervised
- Upload **Dataset (CSV)**
- Select **Target Column** (for supervised learning)
- Choose **Model Algorithm**
- Adjust **Model Parameters** (e.g., depth, estimators, epsilon)
- Train or Run Model

Main Display Area:

- Dataset Preview
- Model Results (Accuracy, Clusters, etc.)
- Visualizations (Confusion Matrix or Cluster Plot)
- Evaluation Summary