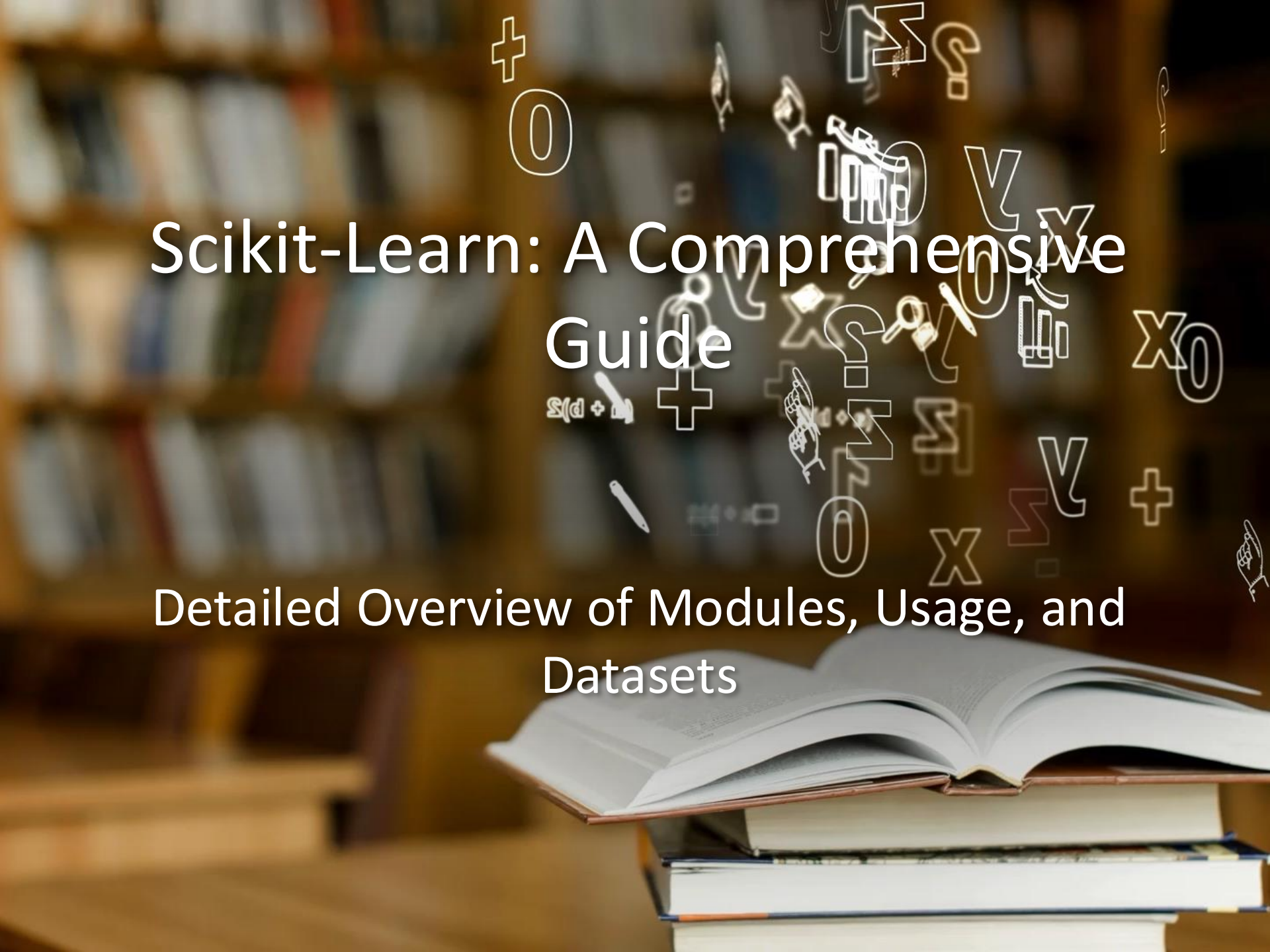

Scikit-Learn: A Comprehensive Guide

Detailed Overview of Modules, Usage, and Datasets

Introduction to Scikit-Learn

- Scikit-learn is a machine learning library in Python built on NumPy, SciPy, and Matplotlib. It provides simple and efficient tools for data mining and data analysis.

Key Features of Scikit-Learn

- Supervised and unsupervised learning algorithms

- Feature extraction and preprocessing

- Model selection and evaluation

- Built-in datasets

Modules in Scikit-Learn

- sklearn.preprocessing: Data preprocessing
- sklearn.model_selection: Model selection
- sklearn.linear_model: Linear models
- sklearn.ensemble: Ensemble methods
- sklearn.svm: Support Vector Machines
- sklearn.tree: Decision Trees
- sklearn.neighbors: Nearest Neighbors
- sklearn.cluster: Clustering
- sklearn.metrics: Metrics and scoring

Preprocessing
(sklearn.preprocessing)

- Tools for preparing data before modeling

- Examples:

- StandardScaler: Standardize features

- MinMaxScaler: Scale features to a range

- OneHotEncoder: Encode categorical variables

Model Selection
(sklearn.model_selection)

- Functions for splitting datasets and evaluating models

- Examples:

- train_test_split: Split datasets into training and test sets

- GridSearchCV: Hyperparameter tuning using cross-validation

Built-in Datasets

- sklearn.datasets provides access to standard datasets

- Examples:

- iris: Iris flower dataset

- digits: Handwritten digits dataset

- boston: Boston house prices (deprecated)

- wine: Wine classification dataset

- diabetes: Diabetes dataset

Example Workflow with Scikit-Learn

1. Load the dataset
2. Preprocess the data (e.g., scaling)
3. Split the dataset (train-test split)
4. Select and train the model
5. Evaluate the model
6. Optimize using hyperparameter tuning