Scikit-learn (also known as sklearn) is a **popular open-source machine learning library** in Python. It provides **simple and efficient tools** for data mining, data analysis, and machine learning. Built on **NumPy**, **SciPy**, and **matplotlib**, it supports both **supervised and unsupervised learning**, along with tools for model selection and evaluation.

Step-by-Step Guide to Learn Scikit-learn

Step 1: Prerequisites

Before diving into Scikit-learn, you should be comfortable with:

- □ **Python programming** (variables, loops, functions, classes)
- □ **NumPy** (for arrays and numerical operations)
- □ **Pandas** (for data manipulation)
- □ **Matplotlib / Seaborn** (for visualization)

□ **Resources:**

- Python: https://docs.python.org/3/tutorial/
- NumPy: https://numpy.org/learn/
- Pandas: https://pandas.pydata.org/docs/getting_started/

Step 2: Install Scikit-learn

Install it using pip:

```
pip install scikit-learn
```

Step 3: Learn the Basics of Scikit-learn

Start with understanding the **main components**:

- **Datasets** (loading built-in data like Iris, Boston, etc.)
- **Preprocessing** (scaling, encoding, etc.)
- **Model selection** (train-test split, cross-validation)
- **Estimators** (models like `LinearRegression`, `RandomForestClassifier`)
- **Pipelines** (chaining preprocessing and modeling steps)

Try this official tutorial:-

Scikit-learn Getting Started Guide

Step 4: Hands-on Practice with Real Examples

Example 1: Train a Simple Model

Here's a simple example using scikit-learn to build a model that predicts House Price

 Goal

Example: House Price Prediction in India (Simulated)

We'll Use: -

• scikit-learn for model training (LinearRegression)

• pandas for a custom dataset

• Features relevant to Indian housing

How to Install You can install everything in one line using pip:

pip install scikit-learn numpy pandas

Full code housing.py :-

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Step 1: Simulated dataset (India-based features)
data = {
    'Area': [1200, 1000, 1500, 800, 950, 1350, 1600, 1100, 1400, 1250],
    'BHK': [3, 2, 3, 2, 2, 3, 4, 2, 3, 3],
    'Bathrooms': [2, 1, 2, 1, 2, 2, 3, 1, 2, 2],
    'LocationScore': [8, 6, 9, 5, 7, 8, 10, 6, 9, 7],
    'Age': [5, 10, 3, 12, 8, 4, 2, 9, 3, 6],
    'Price': [75, 60, 90, 50, 58, 85, 100, 65, 92, 80]  # in Lakhs
}
df = pd.DataFrame(data)

# Step 2: Feature matrix and target variable
X = df[['Area', 'BHK', 'Bathrooms', 'LocationScore', 'Age']]
y = df['Price']  # in Lakhs
```

```
# Step 3: Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Step 5: Predict on test set and calculate RMSE
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
print(f"\nModel RMSE: {rmse:.2f} Lakhs")

# Step 6: Get user input for prediction
print("\n Enter the following details to estimate house price (in India):")
try:
    area = float(input("Built-up Area (in sq ft): "))
    bhk = int(input("Number of Bedrooms (BHK): "))
    baths = int(input("Number of Bathrooms: "))
    location_score = float(input("Location Score (1-10): "))
    age = int(input("Age of Property (in years): "))

    input_features = pd.DataFrame([[area, bhk, baths, location_score, age]],
                                  columns=['Area', 'BHK', 'Bathrooms', 'LocationScore', 'Age'])
```

```
    predicted_price = model.predict(input_features)[0]
    print(f"\n Estimated House Price: ₹{predicted_price:.2f} Lakhs")

except ValueError:
    print(" Invalid input. Please enter numeric values.")
```

Step 5: Explore Common Algorithms in Scikit-learn

- **Regression**
  - LinearRegression
  - Ridge, Lasso
- **Classification**
  - LogisticRegression
  - KNeighborsClassifier
  - DecisionTreeClassifier
  - RandomForestClassifier
  - SVM (Support Vector Machines)
- **Clustering**
  - KMeans
  - DBSCAN
- **Dimensionality Reduction**
  - PCA (Principal Component Analysis)

 Learn what each algorithm does and when to use it.

Step 6: Learn Preprocessing Techniques

- `StandardScaler`, `MinMaxScaler` – feature scaling
- `LabelEncoder`, `OneHotEncoder` – encoding categorical variables
- `SimpleImputer` – handling missing data

 Use `Pipeline` and `ColumnTransformer` to combine preprocessing and modeling.

Step 7: Model Evaluation and Tuning

Learn to evaluate model performance using:

- Confusion Matrix
- Accuracy, Precision, Recall, F1 Score
- ROC Curve, AUC

Also explore:

- `GridSearchCV` and `RandomizedSearchCV` for hyperparameter tuning.

Step 8: Projects for Practice

- House Price Prediction (regression)
- Customer Segmentation (clustering)

Step 9: Learn from Resources

- □ Official Documentation
- □ Book: *"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow"* by Aurélien Géron
- □ Courses:
  - Coursera: *Machine Learning with Python*
  - Kaggle: *Intro to Machine Learning*

Step 10: Build Your Portfolio

Start creating GitHub projects using Scikit-learn to show your skills:

- Include notebooks (.ipynb), markdown explanations, visualizations.
- Write blog posts or LinkedIn articles about what you learn.
