

## 3. Deep Learning tools

### 3.5 Scikit-learn

Manel Martínez-Ramón  
Meenu Ajith  
Aswathy Rajendra Kurup

- Scikit-learn is a free machine learning library in Python developed by David Cournapeau as a Google Summer of Code project in 2007.
- It has various features for pre-processing, model selection, classification, clustering, regression and dimensionality reduction.
- It is built on top of NumPy, SciPy and Matplotlib.

- The API has three interfaces that do most of the ML tasks. It also has pre-built algorithms.

**Estimator:** It uses the *fit()* method to train the machine learning model. All regression, classification or unsupervised tasks use the estimator interface.

**Predictor** It uses the *predict()* method to make predictions on test features. A single *fit\_predict()* method can be used.

**Transformer** The *transform()* method provides a library of transformations for data preprocessing, dimensionality reduction, feature extraction, and feature selection. The *fit\_transform()* models and transforms the training data simultaneously.

# Datasets

- Toy examples, real data and data generators can be used with the library.
- Here are some examples:

```
1 #LOADING THE TOY DATSET
2 from sklearn import datasets
3 data = datasets.load_wine() #Load the wine dataset.
4 #LOADING REAL-WORLD DATASET
5 from sklearn.datasets import fetch_california_housing
6 house_data = fetch_california_housing() # A housing dataset.
7
8 #LOADING GENERATED DATASET
9 from sklearn.datasets import make_blobs
10 #Develop isotropic Gaussian blobs for clustering.
11 X, y = make_blobs(n_samples=60, centers=3, n_features=3,
                    random_state=0)
```

# Making blobs

```
1 from sklearn.datasets import make_blobs
2 X, y = make_blobs(n_samples=200, centers=3, n_features=2,
   random_state=0)
```

```
1 string=['*r','+k','ob']
2 for j in range(3):
3     ind = np.where(y==j)
4     plt.plot(X[ind,0],X[ind,1],
   string[j])
```

```
5 plt.show()
```

Module 2 (Python tools)/pic

Methods	Functions
Standardization	StandardScaler()-Zero mean, unit var. MinMaxScaler()- Between $a$ and $b$ .
Normalization	Normalizer()-unit norm.
Imputing values	SimpleImputer()- Fills up missing values Mean, most frequent, median, constant.
Polynomial Features	PolynomialFeatures()- Adds complexity by generating polynomial features.
Categorical Features	OneHotEncoder()- Categorical encoding. OrdinalEncoding()- Encodes unique cats.
Numerical Features	KBinsDiscretizer()- Real to discrete bins. Binarizer()- Thresholds numerical features.
Custom Transformers	FunctionTransformers()- Accepts an existing function and uses it to transform the data.

Table: Data preprocessing methods and functions

# Preprocessing

- A preprocessing example that does data imputation

```
1 #IMPUTING VALUES
2 import numpy as np
3 from sklearn.impute import SimpleImputer
4 arr3 = np.array([[np.nan, 2, 8, np.nan], [6, np.nan, np.nan,
      12], [7, 6, 4, np.nan]]) #define an array.
5 print("Original array:",arr3)
6 im = SimpleImputer(missing_values=np.nan, strategy='median') #
      define the preprocessing module.
7 arr_im = im.fit_transform(arr3) #fills the missing data.
8 print("Array after imputing values:",arr_im)
```

Original array: [[nan 2. 8. nan]

[ 6. nan nan 12.]

[ 7. 6. 4. nan]]

Array after imputing values: [[ 6.5 2. 8. 12. ]

[ 6. 4. 6. 12. ]

[ 7. 6. 4. 12. ]]

# Feature selection and machine learning

- **Feature selection:** Scikit-learn provides several feature selection algorithms. The most widely used methods are Recursive Feature Elimination (RFE) and SelectKBest.
- **Learning models:**
  - **Supervised:** The most commonly used supervised learning method includes linear models such as Linear regression, Logistic regression, Ridge regression, Lasso regression, Decision trees, Naive Bayes Classifier, Support Vector Machines, Random Forests, and others;
  - **Unsupervised:** Isomap, t-SNE, K-Means, Gaussian Mixture Models.
- Examples are provided in separate Jupyter notebooks.