SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

Course Name – Big Data Analytics
III Year / V Semester

Unit 2 – Data Science using Python

Topic – Scikit Learn

Scikit Learn

- Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python.
- It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.
- This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.
- It was originally called scikits.learn and was initially developed by David Cournapeau as a Google summer of code project in 2008.
- Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation),

Scikit Learn

The functionality that scikit-learn provides include:

- Regression, including Linear and Logistic Regression
- Classification, including K-Nearest Neighbors
- Clustering, including K-Means and K-Means++
- Model selection
- Preprocessing, including Min-Max Normalization

Scikit Learn

Features
Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modelling the data.

Some of the most popular groups of models provided by Sklearn are as follows –

- Supervised Learning algorithms – Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

- Unsupervised Learning algorithms – On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

- Clustering – This model is used for grouping unlabeled data.

Scikit Learn

Features

- Cross Validation – It is used to check the accuracy of supervised models on unseen data.
- Dimensionality Reduction – It is used for reducing the number of attributes in data which can be further used for summarisation, visualisation and feature selection.
- Ensemble methods – As name suggest, it is used for combining the predictions of multiple supervised models.
- Feature extraction – It is used to extract the features from data to define the attributes in image and text data.
- Feature selection – It is used to identify useful attributes to create supervised models.

Scikit Learn

Estimator API

- It is one of the main APIs implemented by Scikit-learn. It provides a consistent interface for a wide range of ML applications that's why all machine learning algorithms in Scikit-Learn are implemented via Estimator API. The object that learns from the data (fitting the data) is an estimator. It can be used with any of the algorithms like classification, regression, clustering or even with a transformer, that extracts useful features from raw data.

- For fitting the data, all estimator objects expose a fit method that takes a dataset shown as follows –
- estimator.fit(data)
- Next, all the parameters of an estimator can be set, as follows, when it is instantiated by the corresponding attribute.
- estimator = Estimator (param1=1, param2=2)
- estimator.param1
- The output of the above would be 1.

Scikit Learn

Steps in using Estimator API

Followings are the steps in using the Scikit-Learn estimator API –

Step 1: Choose a class of model

- In this first step, we need to choose a class of model. It can be done by importing the appropriate Estimator class from Scikit-learn.

Step 2: Choose model hyperparameters

- In this step, we need to choose class model hyperparameters. It can be done by instantiating the class with desired values.

Step 3: Arranging the data

- Next, we need to arrange the data into features matrix (X) and target vector(y).

Step 4: Model Fitting

- Now, we need to fit the model to your data. It can be done by calling fit() method of the model instance.

Step 5: Applying the model

•After fitting the model, we can apply it to new data. For supervised learning, use **predict()** method to predict the labels for unknown data. While for unsupervised learning, use **predict()** or **transform()** to infer properties of the data.

THANK YOU