teleuniv
Innovative Interactive Immersive

# Python: Scikitlearn

# Session - 5

Previous sessions:

Data Types, Collections

Control Statements, Operators

This Session:

ScikitLearn

# Introduction

Python has a rich and healthy ecosystem of various libraries for data analysis.

But one of them stands out as the best and most effective library. No points for guessing, it is **Scikit-Learn,**

Scikit-learn was initially developed by David Cournapeau as a Google summer of code project in 2007. In the same year, Matthieu Brucher joined the project. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel of INRIA got involved with the project and made the first public release, February the 1st 2010. Since then, several new contributions have been made to the project.

# Introduction

Machine Learning library

Designed to inter-operate with Numpy and SciPy

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

The library is built upon the SciPy (Scientific Python)

Extensions or modules for SciPy care conventionally named SciKits.
As such, the module provides learning algorithms and is named scikit-learn.

# What are the features

**Clustering**: for grouping unlabeled data such as KMeans.

**Cross Validation**: for estimating the performance of supervised models on unseen data

**Datasets**: for test datasets and for generating datasets with specific properties for investigating model behavior.

**Dimensionality Reduction**: for reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis

# What are the features

**Ensemble methods**: for combining the predictions of multiple supervised models.

**Feature extraction:** for defining attributes in image and text data.

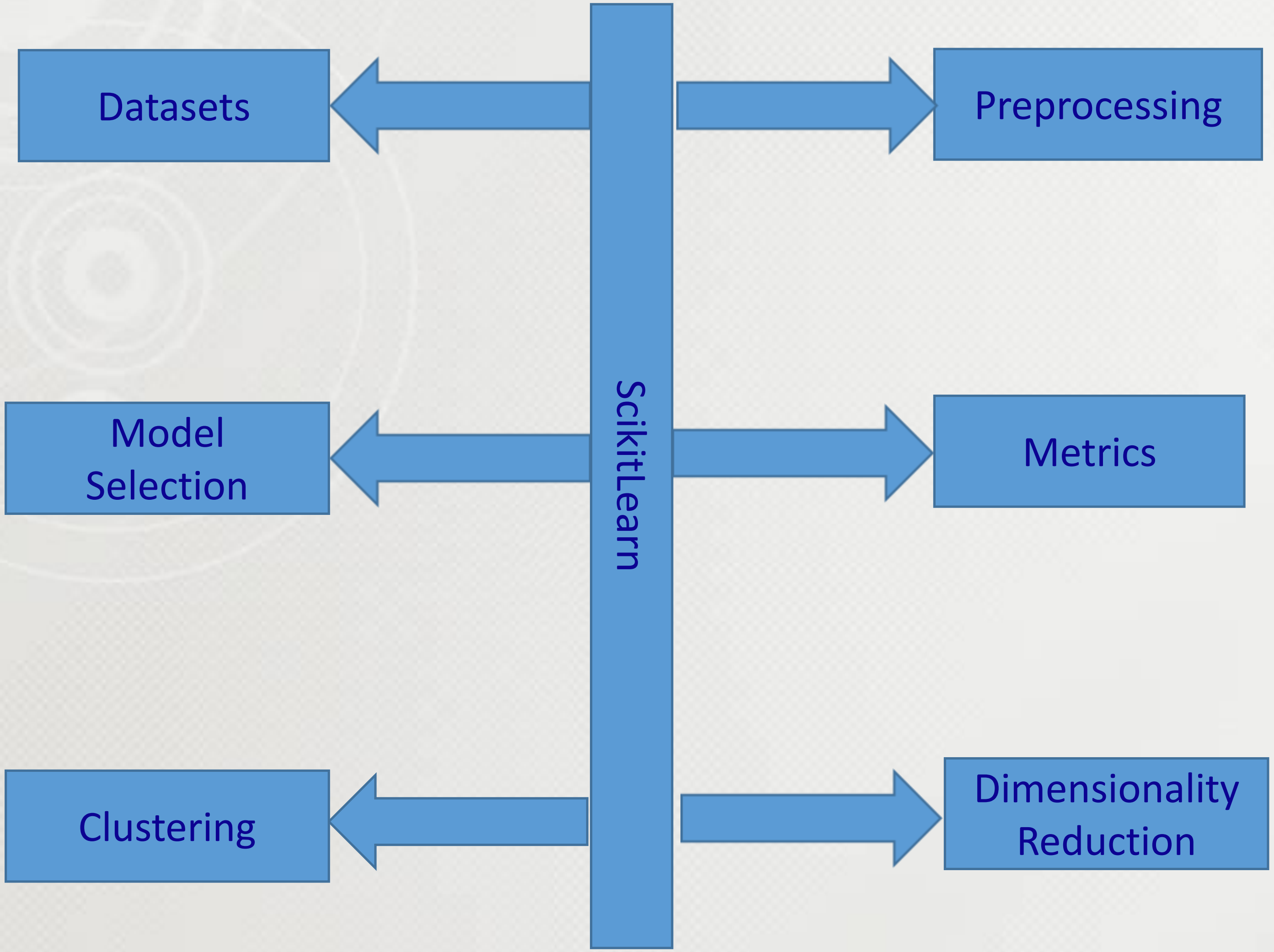
**Feature selection:** for identifying meaningful attributes from which to create supervised models.

**Parameter Tuning:** for getting the most out of supervised models.

**Manifold Learning:** For summarizing and depicting complex multi-dimensional data.

**Supervised Models:** a vast array not limited to generalized linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

# ScikitLearn



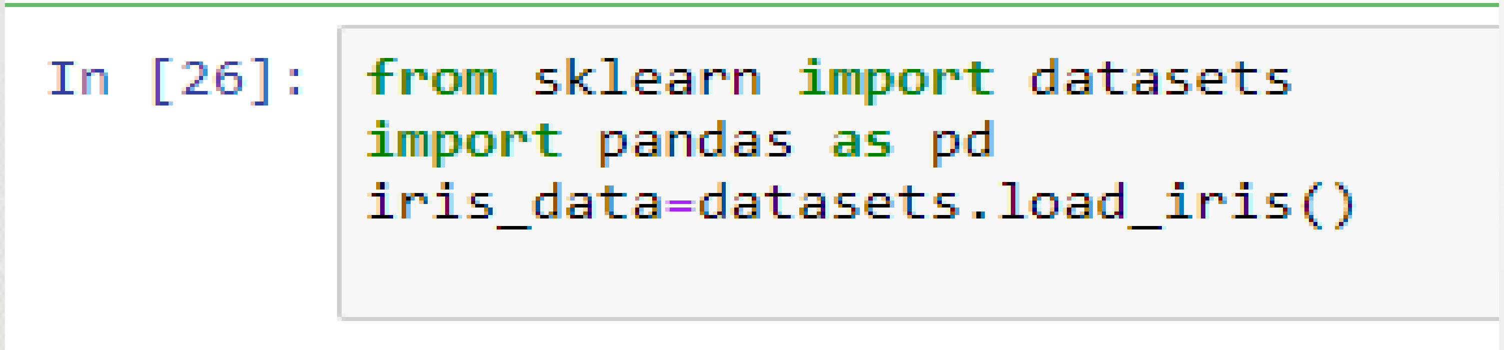
# ScikitLearn

| Pre-Processing | |
|---|---|
| **Function** | **Description** |
| sklearn.preprocessing.StandardScaler | Standardize features by removing the mean and scaling to unit variance |
| sklearn.preprocessing.Imputer | Imputation transformer for completing missing values |
| sklearn.preprocessing.LabelBinarizer | Binarize labels in a one-vs-all fashion |
| sklearn.preprocessing.OneHotEncoder | Encode categorical integer features using a one-hot a.k.a one-of-K scheme |
| sklearn.preprocessing.PolynomialFeatures | Generate polynomial and interaction features |

# ScikitLearn Libraries

Loading  Datasets:

scikit-learn comes with a few standard datasets, for instance the **iris and digits datasets** for classification and the **Boston house prices** dataset for regression.



```
In [26]:  from sklearn import datasets
          import pandas as pd
          iris_data=datasets.load_iris()
```

# ScikitLearn Libraries

Available Datasets:

scikit-learn comes with a few small standard datasets that do not require to download any file from some external website.

**load_boston([return_X_y])** Load and return the boston house-prices dataset (regression)
**load_iris([return_X_y])** Load and return the iris dataset (classification).
**load_diabetes([return_X_y])** Load and return the diabetes dataset (regression).
**load_digits([n_class, return_X_y])** Load and return the digits dataset (classification).
**load_linnerud([return_X_y])** Load and return the linnerud dataset (multivariate regression)
**load_wine([return_X_y])** Load and return the wine dataset (classification).
**load_breast_cancer([return_X_y])** Load and return the breast cancer wisconsin dataset (classification).

# ScikitLearn Libraries

| Regression | |
|---|---|
| **Function** | **Description** |
| sklearn.tree.DecisionTreeRegressor | A decision tree regressor |
| sklearn.svm.SVR | Epsilon-Support Vector Regression |
| sklearn.linear_model.LinearRegression | Ordinary least squares Linear Regression |
| sklearn.linear_model.Lasso | Linear Model trained with L1 prior as regularized (a.k.a the lasso) |
| sklearn.linear_model.SGDRegressor | Linear model fitted by minimizing a regularized empirical loss with SGD |
| sklearn.linear_model.ElasticNet | Linear regression with combined L1 and L2 priors as regularizor |
| sklearn.ensemble.RandomForestRegressor | A random forest regressor |
| sklearn.ensemble.GradientBoostingRegressor | Gradient Boosting for regression |
| sklearn.neural_network.MLPRegressor | Multi-layer Perceptron regressor |

# ScikitLearn Libraries

**classification**

| Function | Description |
|---|---|
| sklearn.neural_network.MLPClassifier | Multi-layer Perceptron classifier |
| sklearn.tree.DecisionTreeClassifier | A decision tree classifier |
| sklearn.svm.SVC | C-Support Vector Classification |
| sklearn.linear_model.LogisticRegression | Logistic Regression (a.k.a logit, Max Ent) classifier |
| sklearn.linear_model.SGDClassifier | Linear classifiers (SVM, logistic regression, a.o.) with SGD training |
| sklearn.naive_bayes.GaussianNB | Gaussain Naïve Bayes |
| sklearn.neighbors.KNeighborsClassifier | Classifier implementing the k-nearest neighbors vote |
| sklearn.ensemble.RandomForestClassifier | A random forest classifier |
| sklearn.ensemble.GradientBoostingClassifier | Gradient Boosting for classification |

# ScikitLearn Libraries

| Clustering | |
|---|---|
| **Function** | **Description** |
| sklearn.cluster.Kmeans | K-Means clustering |
| sklearn.cluster.DBSCAN | perform DBSCAN clustering from vector array or distance matrix |
| sklearn.cluster.AgglomerativeClustering | Agglomerative clustering |
| sklearn.cluster.SpectralBiclustering | Spectral bi-clustering |

# ScikitLearn Libraries

| Dimensionality Reduction | |
|---|---|
| **Function** | **Description** |
| sklearn.decomposition.PCA | Principal component analysis (PCA) |
| sklearn.decomposition.LatentDirichletAllocation | Latent Dirichlet Allocation with online variational Bayes algorithm |
| sklearn.decomposition.SparseCoder | Sparse coding |
| sklearn.decomposition.DictionaryLearning | Dictionary learning |

# ScikitLearn Libraries

| Dimensionality Reduction | |
|---|---|
| **Function** | **Description** |
| sklearn.decomposition.PCA | Principal component analysis (PCA) |
| sklearn.decomposition.LatentDirichletAllocation | Latent Dirichlet Allocation with online variational Bayes algorithm |
| sklearn.decomposition.SparseCoder | Sparse coding |
| sklearn.decomposition.DictionaryLearning | Dictionary learning |

# ScikitLearn Libraries

**Model Selection**

| Function | Description |
|---|---|
| sklearn.model_selection.Kfold | K-Folds cross-validator |
| sklearn.model_selection.StratifiedKFold | Stratified K-Flods cross-validator |
| sklearn.model_selection.TimeSeriesSplit | Time Series cross-validator |
| sklearn.model_selection.train_test_split | Split arrays or matrices into random train and test subsets |
| sklearn.model_selection.GridSearchCV | Exhaustive search over specified parameter value for an estimator |
| sklearn.model_selection.cross_val_score | Evaluate a score by cross-validation |

# ScikitLearn Libraries

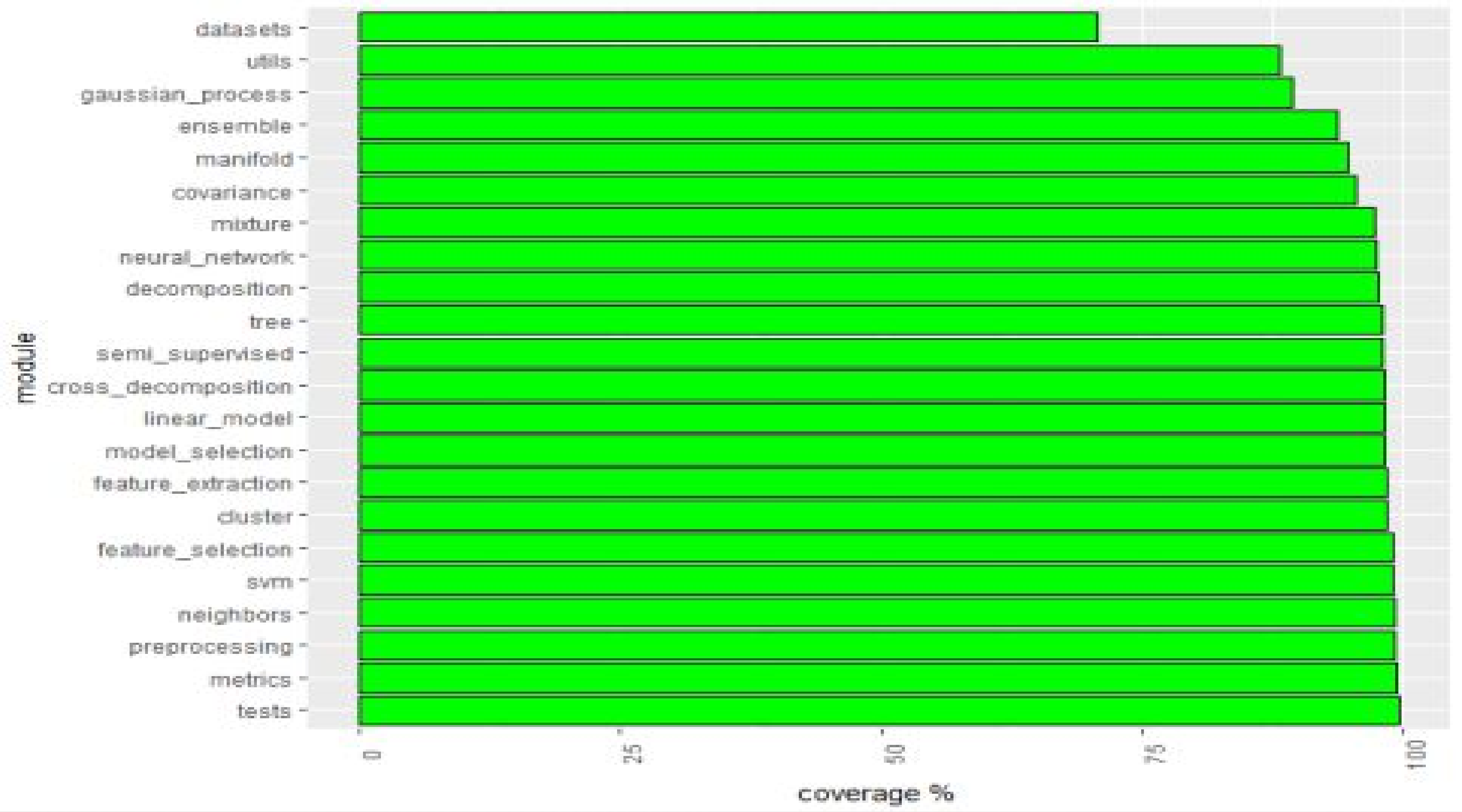
| Metric | |
|---|---|
| **Function** | **Description** |
| sklearn.metrics.accuracy_score | Classification Metric: Accuracy classification score |
| sklearn.metrics.log_loss | Classification Metric: Log loss, a.k.a logistic loss or cross-entropy loss |
| sklearn.metrics.roc_auc_score | Classification Metric: Compute Receiver operating characteristics ROC |
| sklearn.metrics.mean_absolute_error | Regression Metric: Mean absolute error regression loss |
| sklearn.metrics.r2_score | Regression Metric: R^2 (coefficient of determination) regression score |
| sklearn.metrics.label_ranking_loss | Ranking Metric: Compute Ranking loss measure |
| sklearn.metrics.mutual_info_score | Clustering Metric: Mutual Information between two clustering. |

# ScikitLearn Libraries

**Miscellaneous**

| Function | Description |
|---|---|
| sklearn.datasets.load_boston | Load and return the Boston house prices data set (regression) |
| sklearn.datasets.make_classification | Generate a random n-class classification problem |
| sklearn.feature_extraction.FeatureHasher | Implements feature hashing, a.k.a the hashing trick |
| sklearn.feature_selection.SelectKBest | Select features according to the k highest scores |
| sklearn.pipeline.Pipeline | Pipeline of transforms with a final estimator |
| sklearn.semi_supervised.LabelPropagation | Label Propagation classifier for semi-supervised learning |

# ScikitLearn Libraries



# THANK YOU