Scikit-Learn : ( دارای الگوریتم‌های طبقه‌بندی، رگرسیون و خوشه‌بندی مختلف از جمله ماشین‌های بردار پشتیبانی، جنگل‌های تصادفی، تقویت گرادیان ، k-means و DBSCAN است و برای تعامل با پایتون طراحی شده است. )

{

<https://scikit-learn.org/>

***Features of Scikit-Learn***

***Are Scikit-Learn and Sklearn same***

***Classification***

***Loading Dataset***

***Breaking data into Training and Test Set***

1. ***Test Set***
2. ***Train Set***
3. ***Validation Set***
4. ***from sklearn.model\_selection import train\_test\_split ()***

***Learning and Predicting***

1. ***classifier.fit(labels,targets)***
2. ***classifier.predict()***

***Performance Analysis***

1. ***classifier.confusion\_matrix()***
2. ***classifier.accuracy\_score()***

***Scikit-Learn Functionalities***

|  |  |  |
| --- | --- | --- |
| ***Sno.*** | ***Function Name*** | ***Description*** |
| ***1*** | ***sklearn.base*** | ***Base Class for all the Estimators*** |
| ***2*** | ***sklear.calibration*** | ***Calibration of Predicted Probabilities*** |
| ***3*** | ***sklear.cluster*** | ***It provides various unsupervised learning algorithms*** |
| ***4*** | ***sklearn.cluster.k\_means*** | ***Provides all the functionalities of the K-Means clustering algorithm*** |
| ***5*** | ***sklear.cluster.bicluster*** | ***It provides the spectral biclustering algorithms*** |
| ***6*** | ***sklearn.compose*** | ***Meta-estimators for building composite models with transformers*** |
| ***7*** | ***sklearn.covariance*** | ***This module includes methods and algorithms to robustify estimate the covariance of features given a set of points. The precision matrix defined as the inverse of the covariance is also estimated*** |
| ***8*** | ***sklearn.cros\_decomposition*** | ***It provides methods and algorithms to support cross decomposition*** |
| ***9*** | ***sklearn.datasets*** | ***This module includes utilities to load datasets, including methods to load and fetch popular reference datasets. It also provides artificial data generators*** |
| ***10*** | ***sklearn.decomposition*** | ***This module includes matrix decomposition algorithms, including among others PCA, NMF, or ICA.*** |
| ***11*** | ***sklearn.discriminant\_analysis*** | ***It provides Linear Discriminant Analysis and Quadratic Discriminant Analysis*** |
| ***12*** | ***sklearn.dummy*** | ***It provides Dummy Estimatators which are helpful to get a baseline value of those metrics for random predictions*** |
| ***13*** | ***sklearn.ensemble*** | ***This module includes ensemble-based methods for classification, regression and anomaly detection*** |
| ***14*** | ***sklearn.exceptions*** | ***This module contains all custom warnings and error classes used across scikit-learn*** |
| ***15*** | ***sklearn.experimental*** | ***This module provides importable modules that enable the use of experimental features or estimators*** |
| ***16*** | ***sklearn.feature\_extraction*** | ***This module deals with features extraction from raw data. It can currently extract features from text and images*** |
| ***17*** | ***sklearn.feature\_selection*** | ***This module implements feature selection algorithms. It currently provides univariate filter selection methods and the recursive feature elimination algorithm*** |
| ***18*** | ***sklearn.gaussian\_process*** | ***This module implements Gaussian Process-based regression and classification*** |
| ***19*** | ***sklearn.isotonic*** | ***This module provides us with capabilities to implement isotonic regression*** |
| ***20*** | ***sklearn.impute*** | ***It provides transformers for missing value imputation*** |
| ***21*** | ***sklearn.kernel\_approximation*** | ***This module implements several approximate kernel feature maps based on Fourier Transforms*** |
| ***22*** | ***sklearn.kernel\_ridge*** | ***It provides capabilities to help us implement kernel ridge regression*** |
| ***23*** | ***sklearn.linear\_model*** | ***It module implements generalized linear models. It includes Ridge regression, Bayesian Regression, Lasso and Elastic Net estimators computed with Least Angle Regression and coordinate descent. It also implements Stochastic Gradient Descent related algorithms.*** |
| ***24*** | ***sklearn.linear\_model.LinearRegression*** | ***It provides functionalities to implement Linear Regression*** |
| ***25*** | ***sklearn.linear\_model.LogisticRegression*** | ***It provides functionalities to implement Logistic Regression*** |
| ***26*** | ***sklearn.mainifold*** | ***This module implements data embedding techniques*** |
| ***27*** | ***sklearn.metrics*** | ***It includes score functions, performance metrics and pairwise metrics and distance computations*** |
| ***28*** | ***sklearn.metrics.accuracy\_score*** | ***It gives the accuracy classification score*** |
| ***29*** | ***sklearn.metrics.confusion\_matrix*** | ***It gives the confusion matrix*** |
| ***30*** | ***sklearn.metrics.f1\_Score*** | ***It gives the F1 score or balanced F-score or F-measure*** |
| ***31*** | ***sklearn.metrics.classification\_report*** | ***It builds a text report showing the main classification metrics*** |
| ***32*** | ***sklearn.metrics.precision\_score*** | ***It gives the precision of the classification*** |
| ***33*** | ***sklearn.metrics.mean\_absolute\_error*** | ***It gives the mean absolute error regression loss*** |
| ***34*** | ***sklearn.metrics.mean\_squared\_error*** | ***It gives the mean squared error regression loss*** |
| ***35*** | ***sklearn.mixture*** | ***This module implements mixture modelling algorithms*** |
| ***36*** | ***sklearn.model\_selection*** | ***This module contains model selections functions*** |
| ***37*** | ***sklearn.multiclass*** | ***This module provides functionalities for implementation of multiclass and multilabel classification*** |
| ***38*** | ***sklearn.multioutput*** | ***This module implements multioutput regression and classification.The estimators provided in this module are meta-estimators: they require a base estimator to be provided in their constructor. The meta-estimator extends single output estimators to multi-output estimators.*** |
| ***39*** | ***sklearn.naive\_bayes*** | ***This module implements Naive Bayes algorithms. These are supervised learning methods based on applying Bayes’ theorem with strong (naive) feature independence assumptions.*** |
| ***40*** | ***sklearn.neighbours*** | ***This module implements the k-nearest neighbors' algorithm.*** |
| ***41*** | ***sklearn.neaural\_network*** | ***This module includes models based on neural networks*** |
| ***42*** | ***sklearn.pipeline*** | ***This module implements utilities to build a composite estimator, as a chain of transforms and estimators*** |
| ***43*** | ***sklearn.inspection*** | ***This module includes tools for model inspection*** |
| ***44*** | ***sklearn.preprocessing*** | ***This module includes scaling, centering, normalization, binarization and imputation methods*** |
| ***45*** | ***sklearn.random\_projection*** | ***It provides random\_projection. Random Projections are a simple and computationally efficient way to reduce the dimensionality of the data by trading a controlled amount of accuracy (as additional variance) for faster processing times and smaller model sizes.*** |
| ***46*** | ***sklearn.semi\_supervised*** | ***This module implements semi-supervised learning algorithms. These algorithms utilized small amounts of labeled data and large amounts of unlabeled data for classification tasks. This module includes Label Propagation.*** |
| ***47*** | ***sklearn.svm*** | ***This module includes Support Vector Machine algorithms*** |
| ***48*** | ***sklearn.tree*** | ***This module includes decision tree-based models for classification and regression.*** |
| ***49*** | ***sklearn.utils*** | ***It includes various utilities*** |

***Transformer :***

***transformer.fit(X) # Train the transformer***

***transformer.transform(X) # Transform the data using an already trained transformer***

***transformer.fit\_transform(X) # Train the transformer and transform the data***

***Predictor :***

***predictor.fit(x, y) # Training the predictor***

***predictor.predict(X\_new) # Predicting the target for new instances once the predictor is trained.***

مرحله پیش پردازش دیتا شامل کار با ترانسفورماتورها () است و ما در مرحله مدل سازی با پیش بینی کننده ها (به طور خاص با مدل ها) [] کار می کنیم.

{