

انواع الگوریتم‌های یادگیری ماشین

انواع الگوریتم‌های یادگیری ماشین

- Supervised Learning (یادگیری با نظارت)
- Unsupervised Learning (یادگیری بدون ناظر)
- Reinforcement Learning (یادگیری تقویتی)

**SUPERVISED
LEARNING**



**UNSUPERVISED
LEARNING**

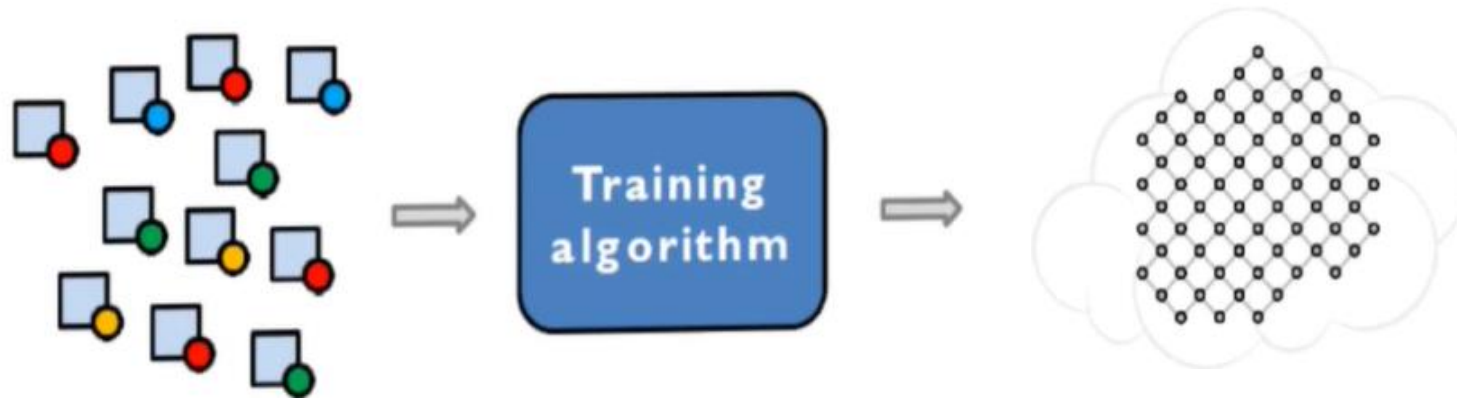


**REINFORCEMENT
LEARNING**



یادگیری با نظارت (Supervised Learning)

فاز آموزش



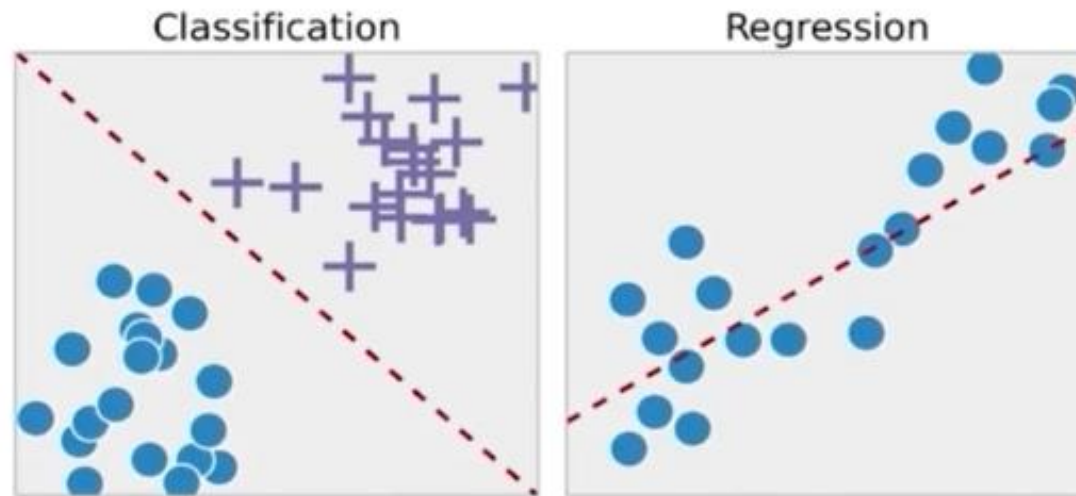
متغیرهای اندازه گیری شده به همراه برچسب

آموزش پارامتر/ساختار

یادگیری با نظارت (Supervised Learning)

یادگیری نظارت شده (Supervised Learning) از داده‌های برچسب‌دار استفاده می‌کند تا با توجه به ویژگی‌ها، یک برچسب را پیش‌بینی کند.

اگر برچسب پیوسته باشد، مسئله از نوع رگرسیون (Regression) است، و اگر برچسب از نوع دسته‌ای باشد، مسئله طبقه‌بندی (Classification) نامیده می‌شود.



یادگیری با نظارت (Supervised Learning)

فاز تست

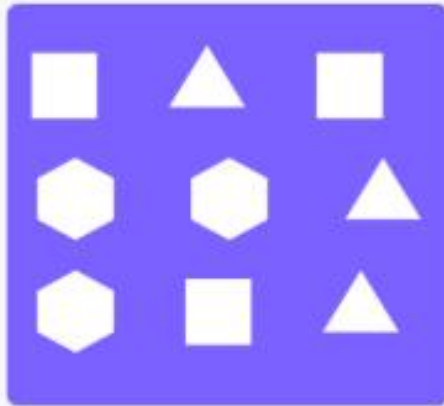
مدل آموزش دیده



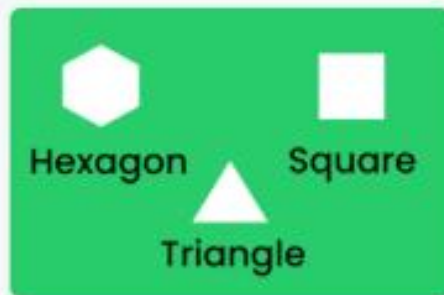
برچسب متغیر

متغیرهای اندازه گیری شده بدون برچسب

Labeled Data



Lables



Model Training



Prediction



Test Data



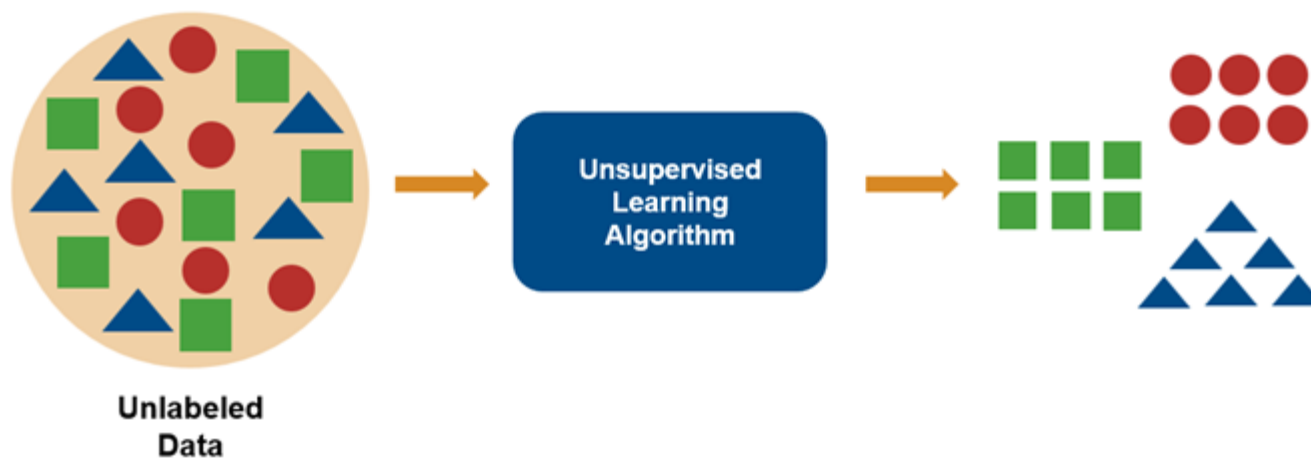
Square

Triangle

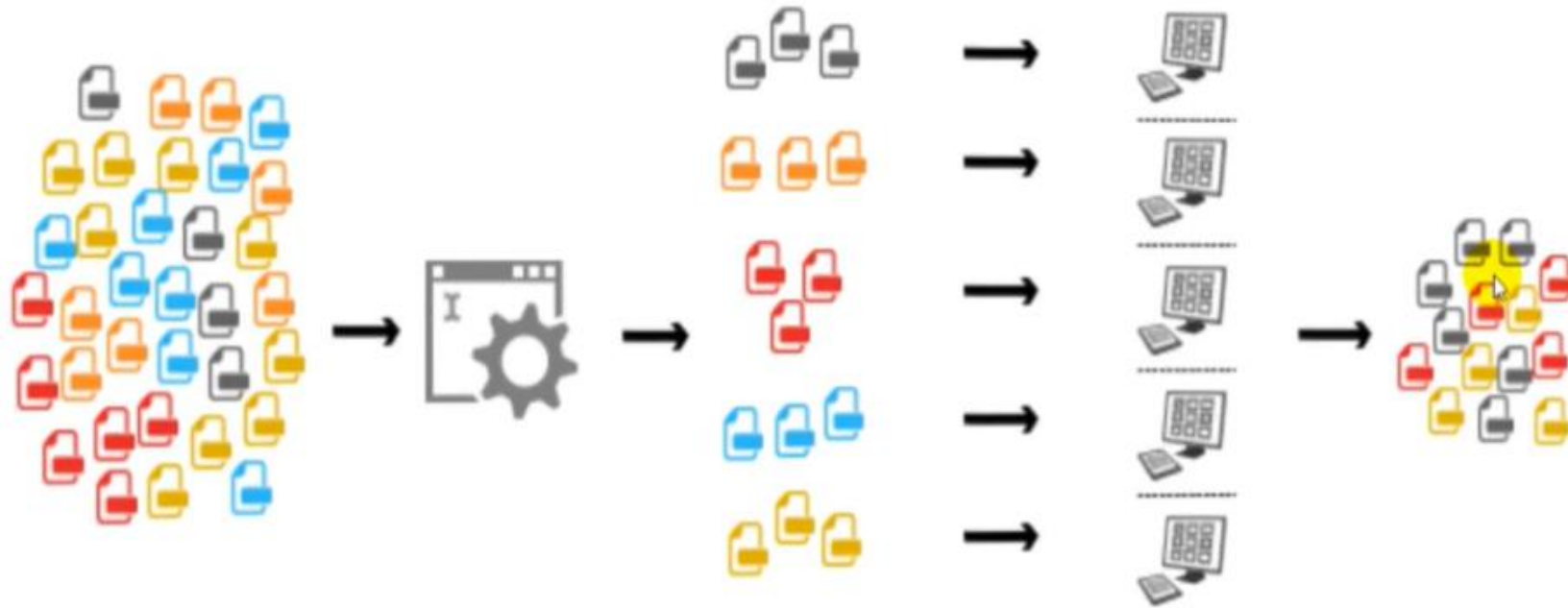


یادگیری بدون ناظر (Unsupervised Learning)

ویژگی‌ها: قد و وزن نژادهای مختلف سگ‌ها.
برچسب: در یادگیری بدون نظارت، برچسب وجود ندارد.
وظیفه: داده‌ها باید در گروه‌های مشابه (خوشه‌ها) قرار گیرند. سپس این وظیفه دانشمند داده است که خوشه‌ها را تفسیر کند.



یادگیری بدون ناظر (Unsupervised Learning)

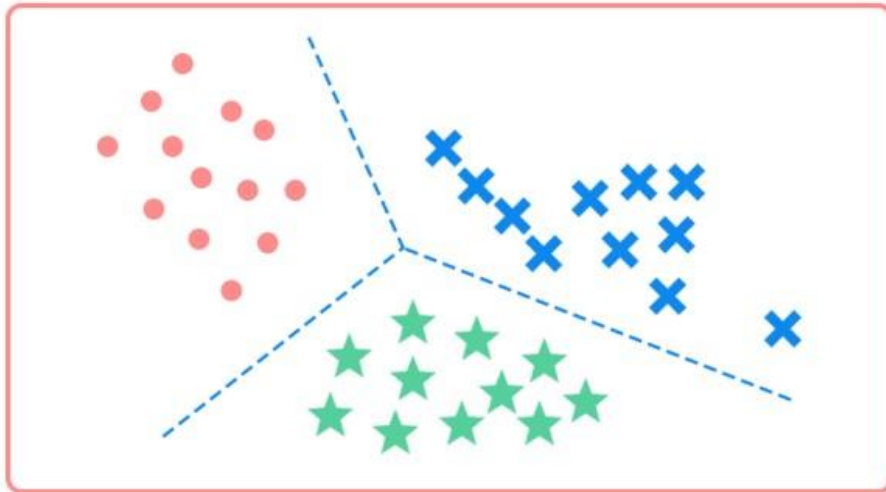


Supervised Learning vs Unsupervised Learning



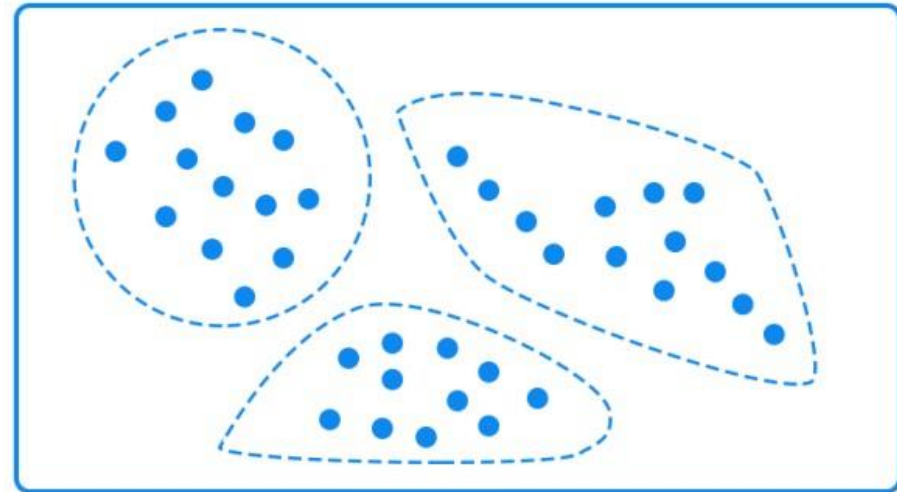
Supervised vs. Unsupervised Learning

Classification



Supervised learning

Clustering

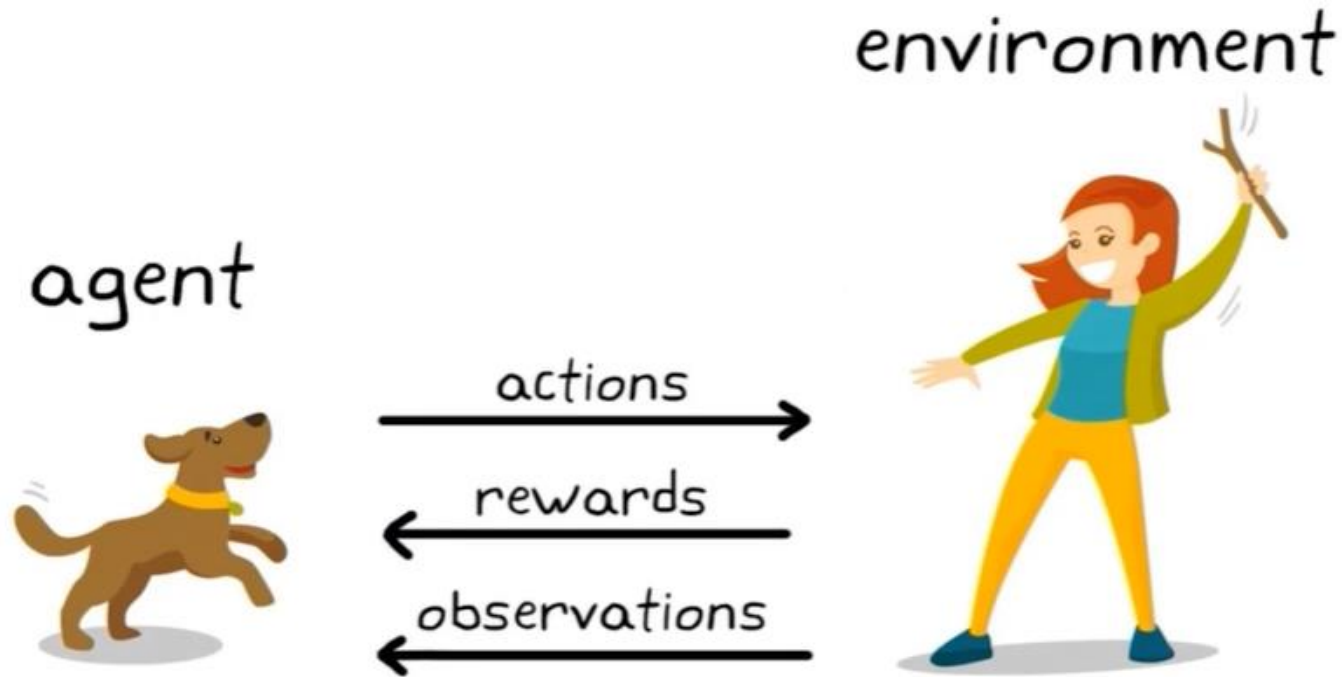


Unsupervised learning

Supervised Learning vs Unsupervised Learning

Supervised Learning	Unsupervised Learning
Input data is labelled.	Input data is unlabelled.
Uses a training data set.	Uses the input data set.
Used for prediction.	Used for analysis.
Enables classification and regression.	Enables Classification, Density Estimation, & Dimension Reduction

یادگیری تقویتی (Reinforcement Learning)



طبقه بند نزدیکترین همسایه

یادگیری نظارت شده – طبقه بندی

طبقه بندی یا classification

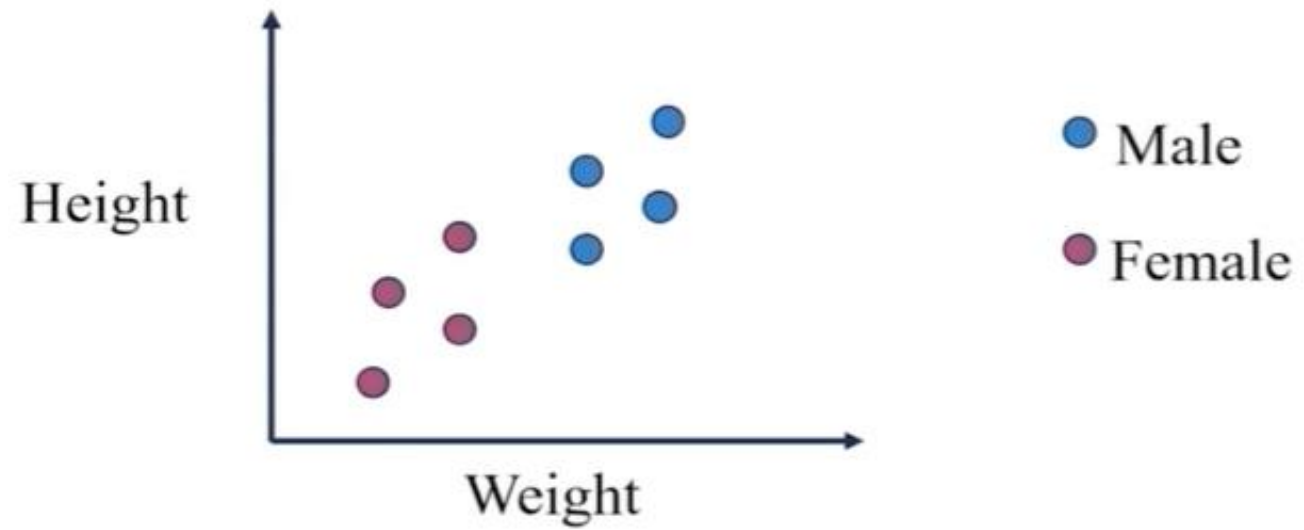
- **Feature:** قد و وزن

- **Label:** جنسیت

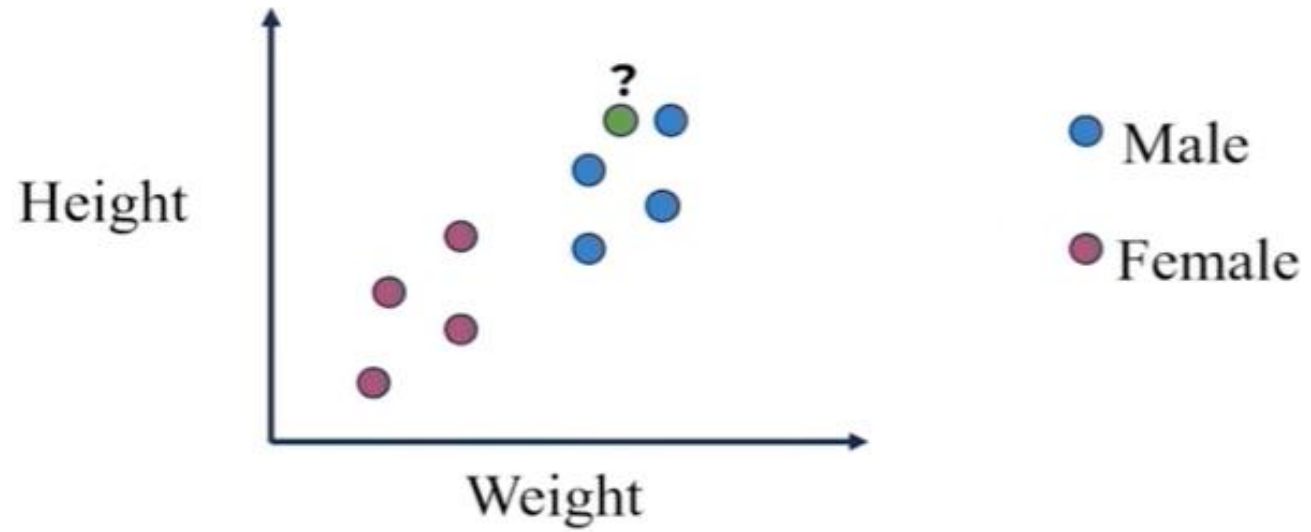
هدف:

- تشخیص جنسیت از روی قد و وزن

یادگیری نظارت شده - طبقه بندی

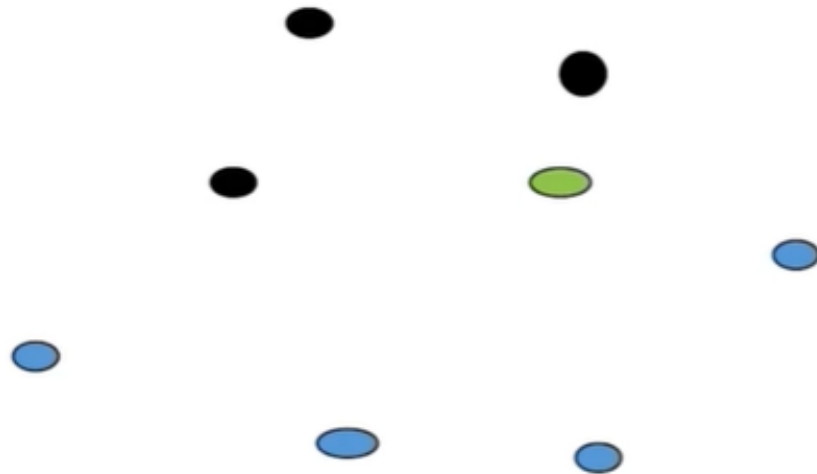


یادگیری نظارت شده - طبقه بندی



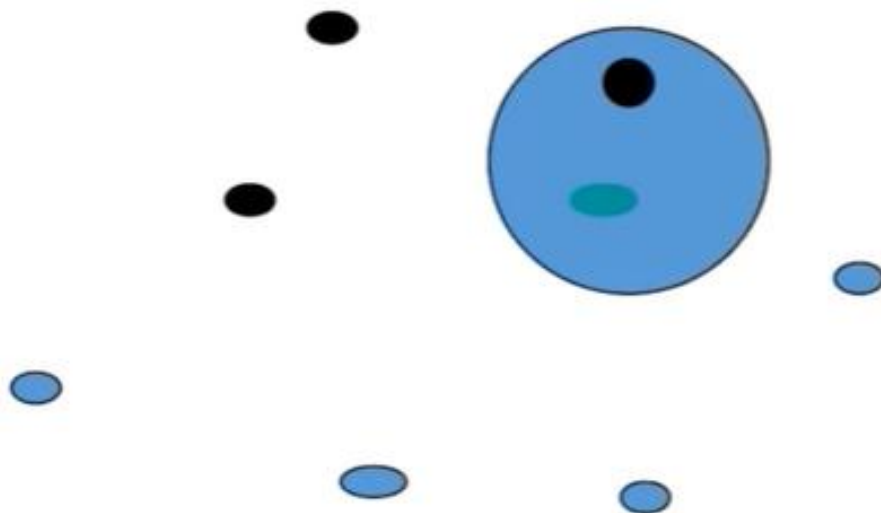
طبقه بندی: الگوریتم نزدیکترین همسایه

- 1-Nearest Neighbor



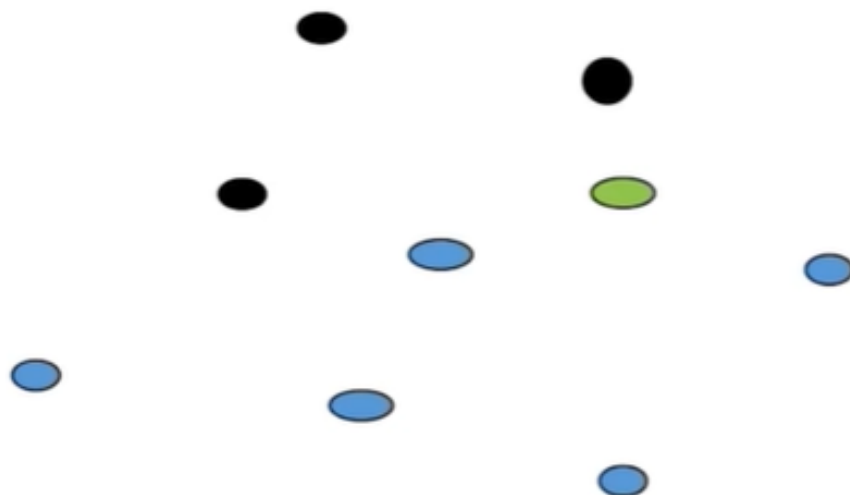
طبقه بندی: الگوریتم نزدیکترین همسایه

- 1-Nearest Neighbor



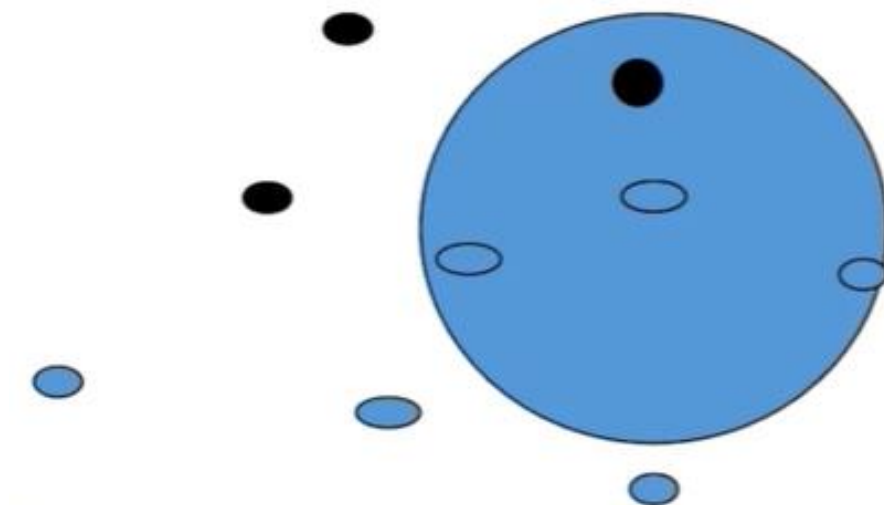
طبقه بندی: الگوریتم نزدیکترین همسایه

- 3-Nearest Neighbor



طبقه بندی: الگوریتم نزدیکترین همسایه

- 3-Nearest Neighbor



scikit-learn

Machine Learning in Python

Getting Started

Release Highlights for 1.7

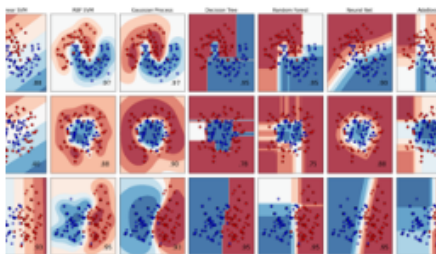
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [logistic regression](#), and [more...](#)



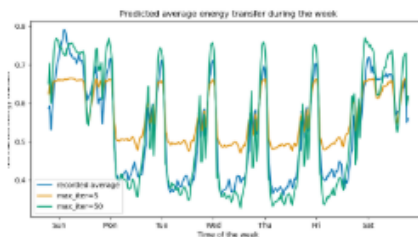
Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [ridge](#), and [more...](#)



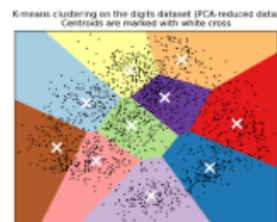
Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.

Algorithms: [k-Means](#), [HDBSCAN](#), [hierarchical clustering](#), and [more...](#)



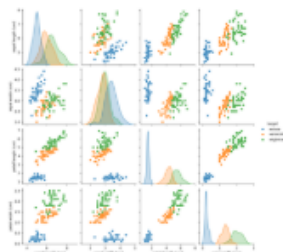
Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, increased efficiency.

Algorithms: [PCA](#), [feature selection](#), [non-negative matrix factorization](#), and [more...](#)



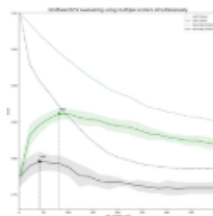
Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning.

Algorithms: [Grid search](#), [cross validation](#), [metrics](#), and [more...](#)



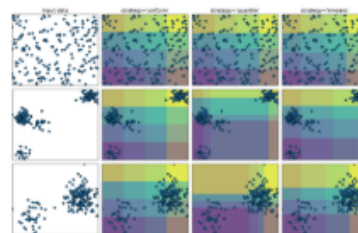
Examples

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

Algorithms: [Preprocessing](#), [feature extraction](#), and [more...](#)



Examples

1. Supervised learning

1.1. Linear Models

- [1.1.1. Ordinary Least Squares](#)
- [1.1.2. Ridge regression and classification](#)
- [1.1.3. Lasso](#)
- [1.1.4. Multi-task Lasso](#)
- [1.1.5. Elastic-Net](#)
- [1.1.6. Multi-task Elastic-Net](#)
- [1.1.7. Least Angle Regression](#)
- [1.1.8. LARS Lasso](#)
- [1.1.9. Orthogonal Matching Pursuit \(OMP\)](#)
- [1.1.10. Bayesian Regression](#)
- [1.1.11. Logistic regression](#)
- [1.1.12. Generalized Linear Models](#)
- [1.1.13. Stochastic Gradient Descent - SGD](#)
- [1.1.14. Perceptron](#)
- [1.1.15. Passive Aggressive Algorithms](#)
- [1.1.16. Robustness regression: outliers and modeling errors](#)
- [1.1.17. Quantile Regression](#)
- [1.1.18. Polynomial regression: extending linear models with basis functions](#)

1.2. Linear and Quadratic Discriminant Analysis

- [1.2.1. Dimensionality reduction using Linear Discriminant Analysis](#)
- [1.2.2. Mathematical formulation of the LDA and QDA classifiers](#)
- [1.2.3. Mathematical formulation of LDA dimensionality reduction](#)
- [1.2.4. Shrinkage and Covariance Estimator](#)
- [1.2.5. Estimation algorithms](#)

1.3. Kernel ridge regression

1.4. Support Vector Machines

- [1.4.1. Classification](#)
- [1.4.2. Regression](#)
- [1.4.3. Density estimation, novelty detection](#)
- [1.4.4. Complexity](#)
- [1.4.5. Tips on Practical Use](#)
- [1.4.6. Kernel functions](#)
- [1.4.7. Mathematical formulation](#)
- [1.4.8. Implementation details](#)

1.5. Stochastic Gradient Descent

[1.5.1. Classification](#)

[1.5.2. Regression](#)

[1.5.3. Online One-Class SVM](#)

[1.5.4. Stochastic Gradient Descent for sparse data](#)

[1.5.5. Complexity](#)

[1.5.6. Stopping criterion](#)

[1.5.7. Tips on Practical Use](#)

[1.5.8. Mathematical formulation](#)

[1.5.9. Implementation details](#)

1.6. Nearest Neighbors

[1.6.1. Unsupervised Nearest Neighbors](#)

[1.6.2. Nearest Neighbors Classification](#)

[1.6.3. Nearest Neighbors Regression](#)

[1.6.4. Nearest Neighbor Algorithms](#)

[1.6.5. Nearest Centroid Classifier](#)

[1.6.6. Nearest Neighbors Transformer](#)

[1.6.7. Neighborhood Components Analysis](#)

1.7. Gaussian Processes

[1.7.1. Gaussian Process Regression \(GPR\)](#)

[1.7.2. Gaussian Process Classification \(GPC\)](#)

[1.7.3. GPC examples](#)

[1.7.4. Kernels for Gaussian Processes](#)

1.8. Cross decomposition

[1.8.1. PLSCanonical](#)

[1.8.2. PLSSVD](#)

[1.8.3. PLSRegression](#)

[1.8.4. Canonical Correlation Analysis](#)

1.9. Naive Bayes

[1.9.1. Gaussian Naive Bayes](#)

[1.9.2. Multinomial Naive Bayes](#)

[1.9.3. Complement Naive Bayes](#)

[1.9.4. Bernoulli Naive Bayes](#)

[1.9.5. Categorical Naive Bayes](#)

[1.9.6. Out-of-core naive Bayes model fitting](#)

1.10. Decision Trees

[1.10.1. Classification](#)

[1.10.2. Regression](#)

[1.10.3. Multi-output problems](#)

[1.10.4. Complexity](#)

[1.10.5. Tips on practical use](#)

[1.10.6. Tree algorithms: ID3, C4.5, C5.0 and CART](#)

[1.10.7. Mathematical formulation](#)

[1.10.8. Missing Values Support](#)

[1.10.9. Minimal Cost-Complexity Pruning](#)

1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking

[1.11.1. Gradient-boosted trees](#)

[1.11.2. Random forests and other randomized tree ensembles](#)

[1.11.3. Bagging meta-estimator](#)

[1.11.4. Voting Classifier](#)

[1.11.5. Voting Regressor](#)

[1.11.6. Stacked generalization](#)

[1.11.7. AdaBoost](#)

1.12. Multiclass and multioutput algorithms

[1.12.1. Multiclass classification](#)

[1.12.2. Multilabel classification](#)

[1.12.3. Multiclass-multioutput classification](#)

[1.12.4. Multioutput regression](#)

1.13. Feature selection

[1.13.1. Removing features with low variance](#)

[1.13.2. Univariate feature selection](#)

[1.13.3. Recursive feature elimination](#)

[1.13.4. Feature selection using SelectFromModel](#)

[1.13.5. Sequential Feature Selection](#)

[1.13.6. Feature selection as part of a pipeline](#)

1.14. Semi-supervised learning

[1.14.1. Self Training](#)

[1.14.2. Label Propagation](#)

1.15. Isotonic regression

1.16. Probability calibration

[1.16.1. Calibration curves](#)

[1.16.2. Calibrating a classifier](#)

[1.16.3. Usage](#)

1.17. Neural network models (supervised)

[1.17.1. Multi-layer Perceptron](#)

[1.17.2. Classification](#)

[1.17.3. Regression](#)

[1.17.4. Regularization](#)

[1.17.5. Algorithms](#)

[1.17.6. Complexity](#)

[1.17.7. Tips on Practical Use](#)

[1.17.8. More control with warm_start](#)

Iris dataset



Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)

A diagram of an Iris flower is shown to the right of the table. Yellow arrows indicate measurements: one arrow points to a petal and is labeled 'Petal', and another arrow points to a sepal and is labeled 'Sepal'. The label 'Class labels (targets)' is positioned below the table.