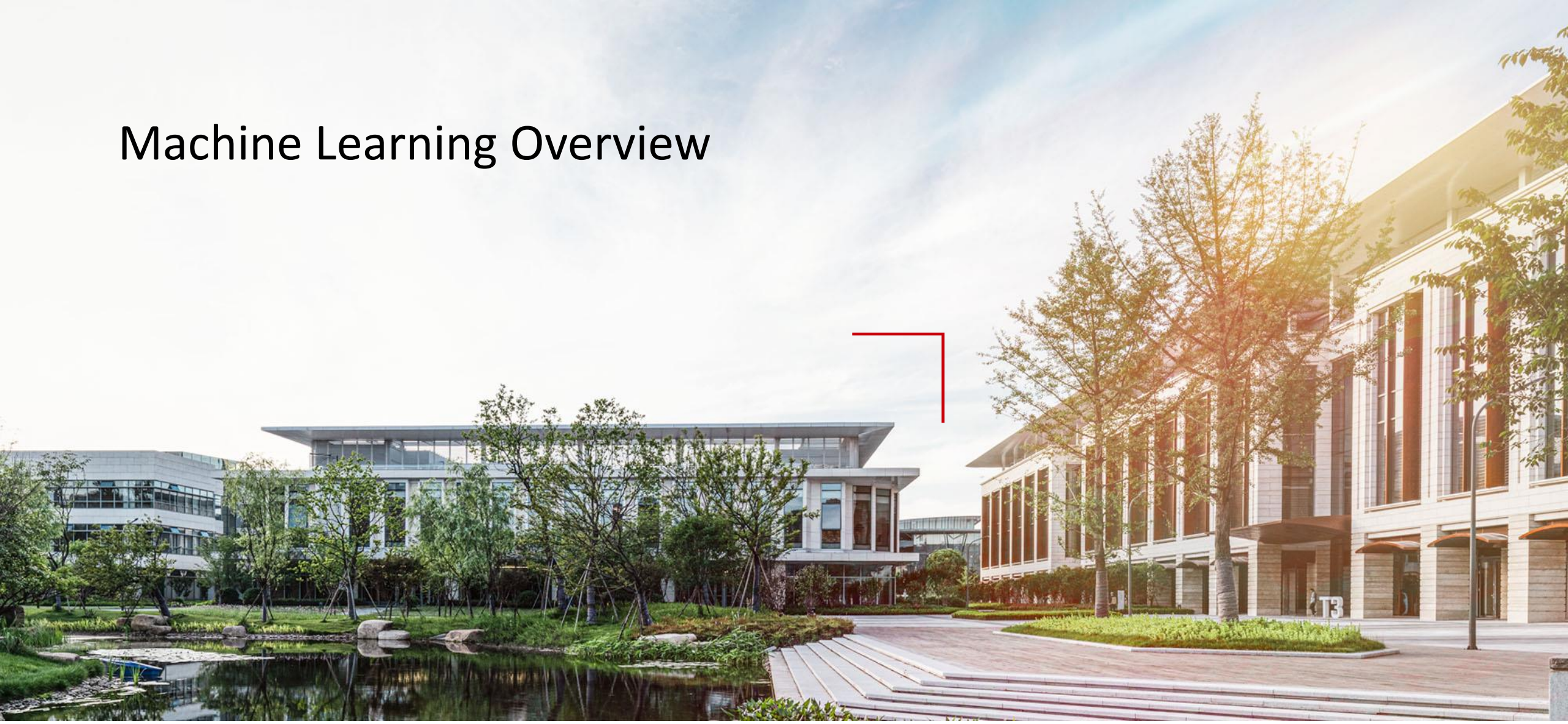
Machine Learning Overview



HUAWEI

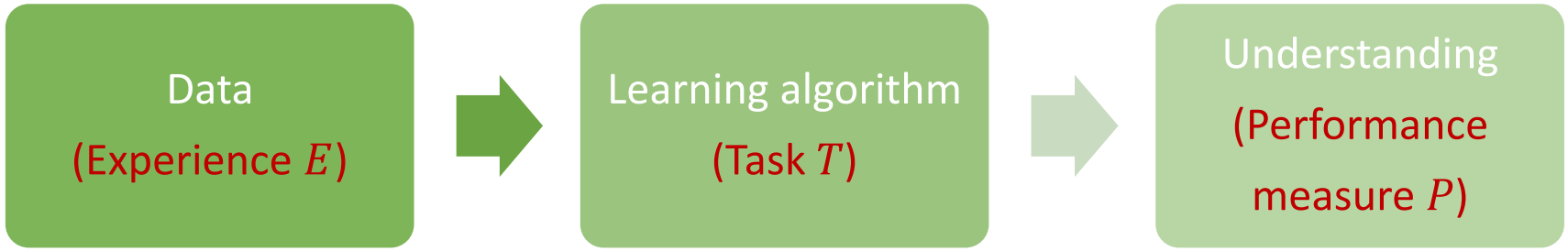
Objectives

- Upon completion of this course, you will understand:
  - Learning algorithm definitions and machine learning process
  - Related concepts such as hyperparameters, gradient descent, and cross-validation
  - Common machine learning algorithms

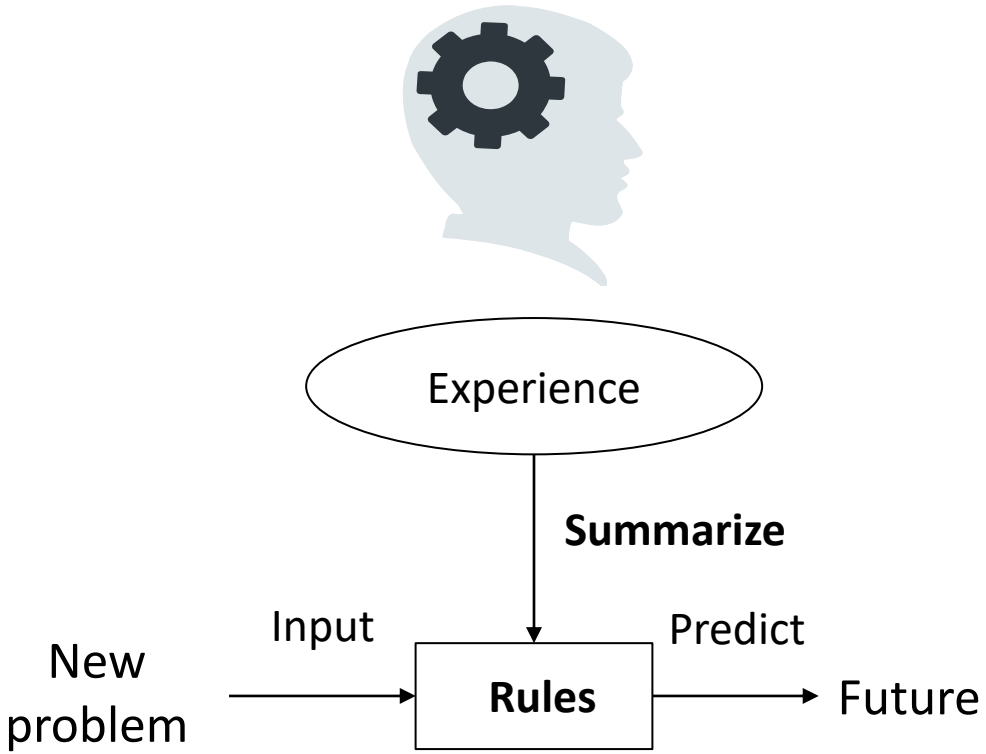
Contents

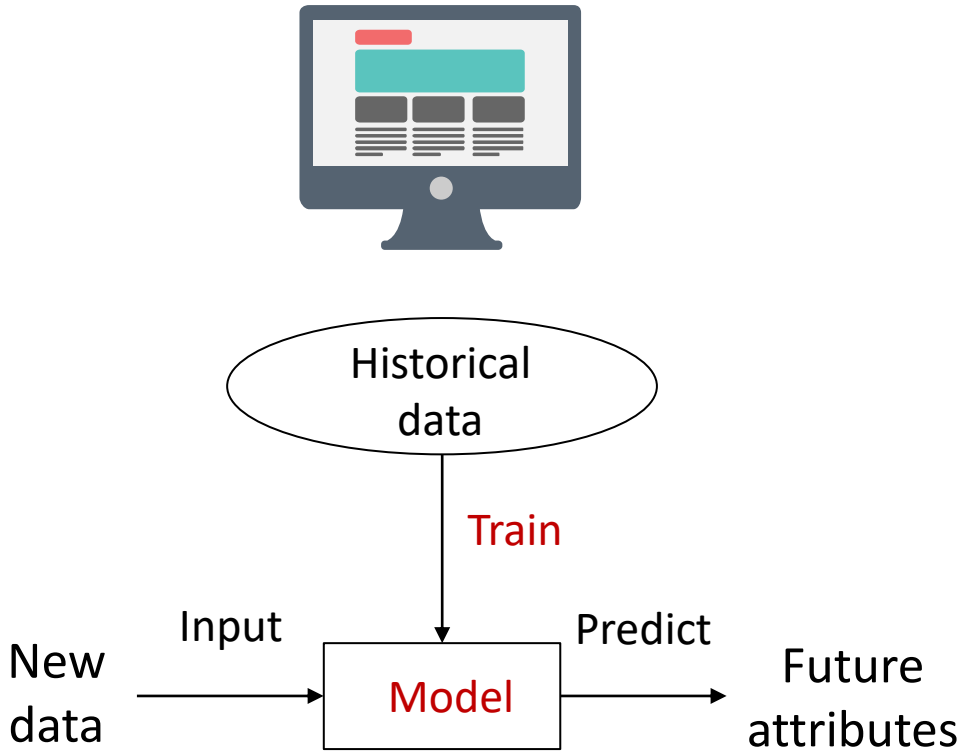
Machine Learning Algorithms (1)

- Machine learning is often combined with deep learning methods to study and observe AI algorithms. A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

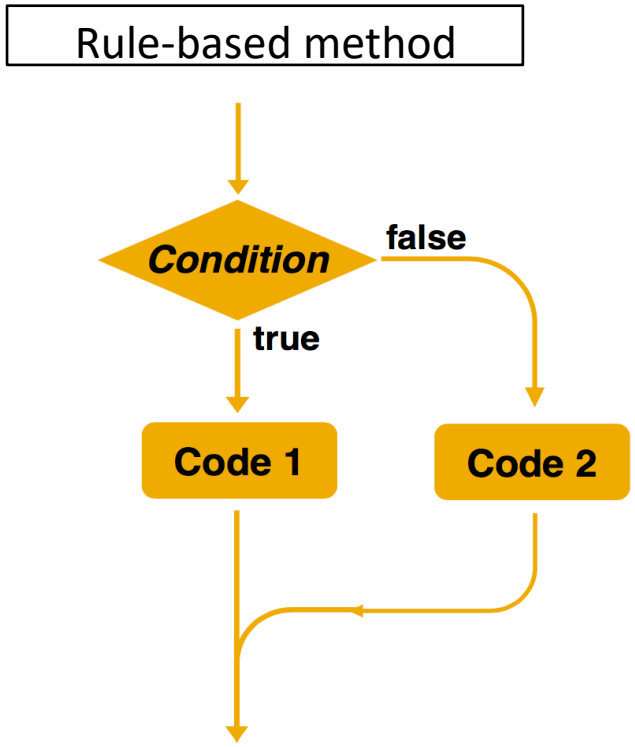


Machine Learning Algorithms (2)

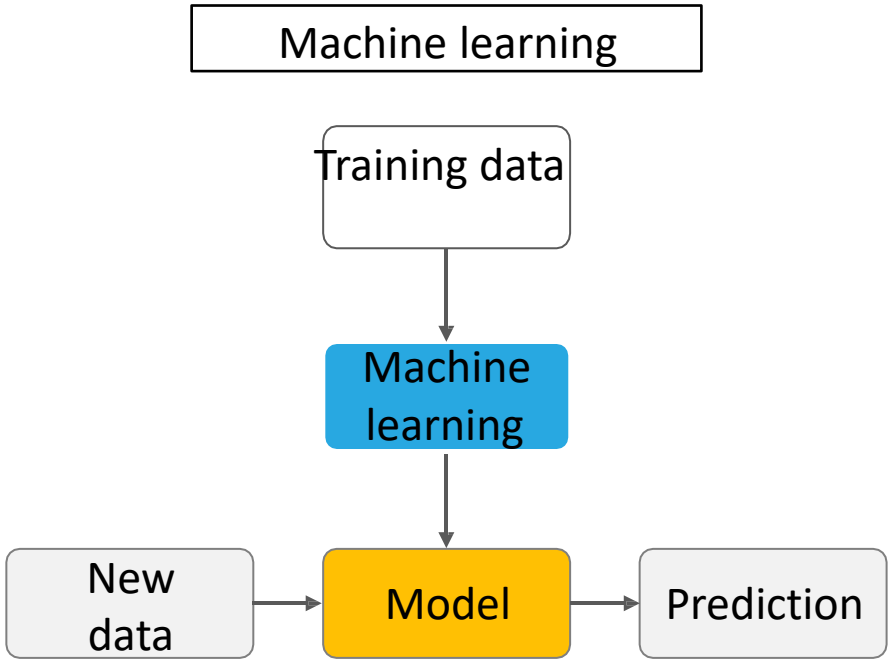




Differences Between Machine Learning Algorithms and Traditional Rule-based Methods



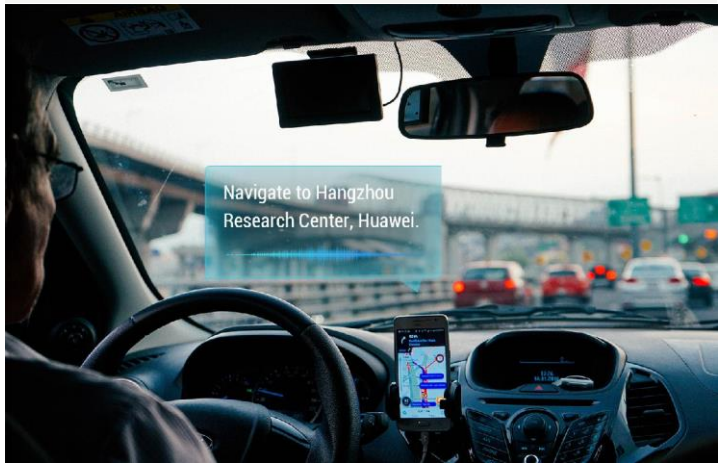
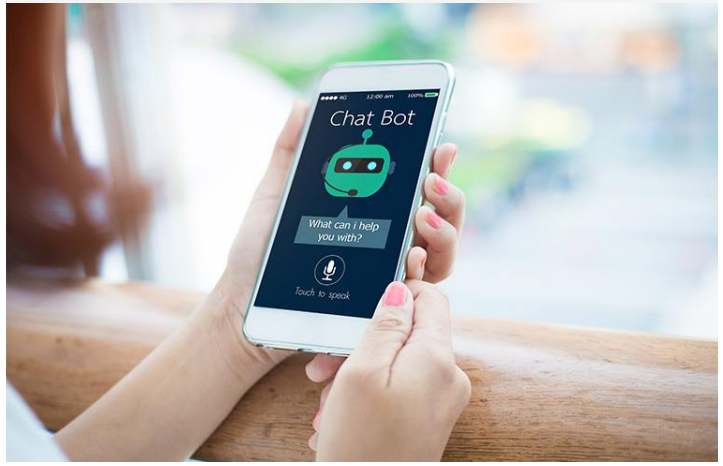
- Explicit programming is used to solve problems.
- Rules can be manually determined.



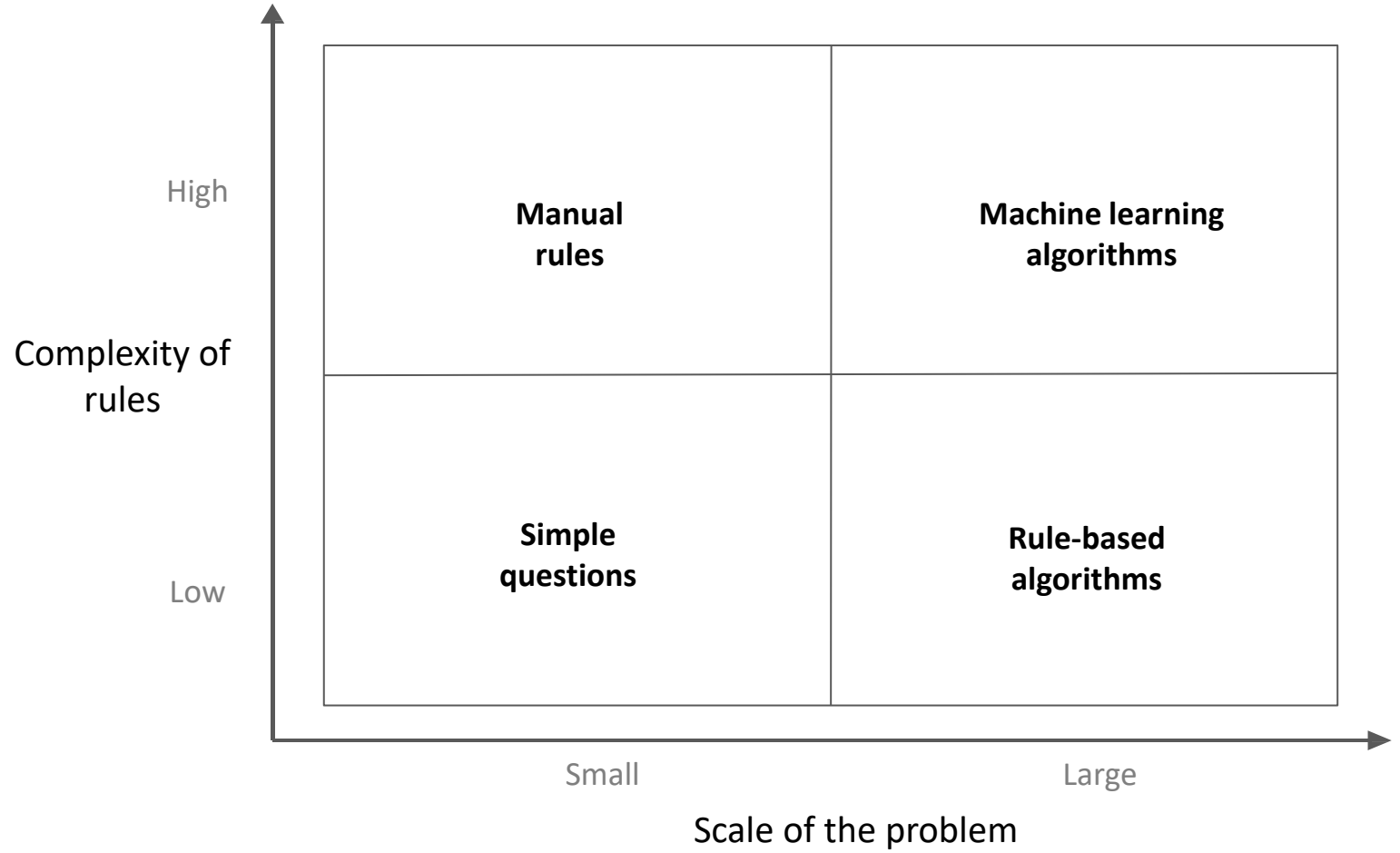
- Models are trained on samples.
- Decision-making rules are complex or difficult to describe.
- Machines automatically learn rules.

When to Use Machine Learning (1)

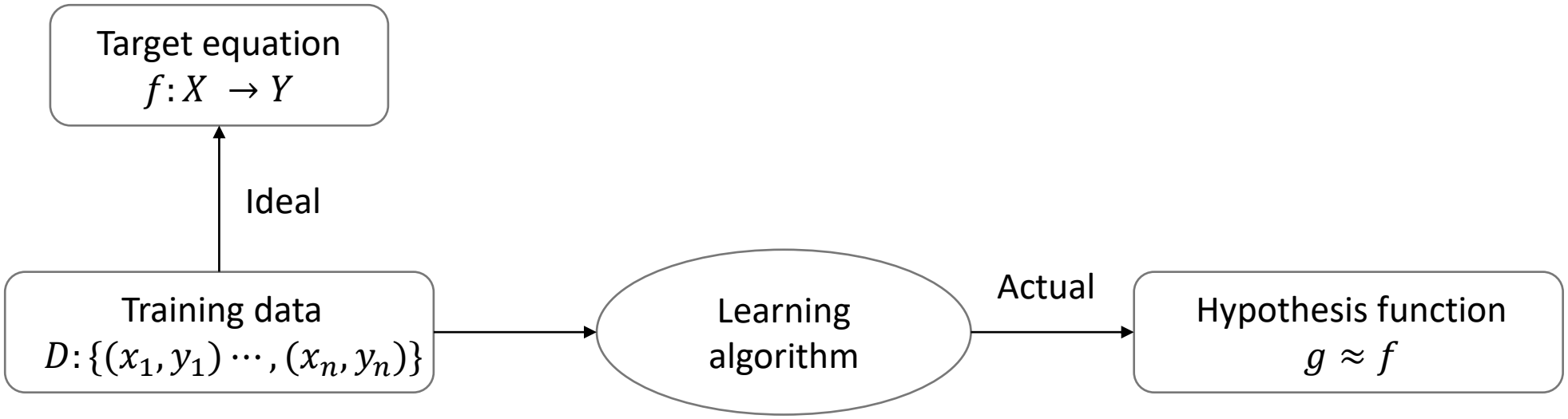
- Machine learning provides solutions to complex problems, or those involving a large amount of data whose distribution function cannot be determined.
- Consider the following scenarios:

Rules are complex or difficult to describe, for example, speech recognition.	Task rules change over time, for example, part-of-speech tagging, in which new words or word meanings can be generated at any time.	Data distribution changes over time and programs need to adapt to new data constantly, for example, sales trend forecast.
		

When to Use Machine Learning (2)



Rationale of Machine Learning Algorithms



- The objective function f is unknown, and the learning algorithm cannot obtain a perfect function f.
- Hypothesis function g approximates function f, but may be different from function f.

Main Problems Solved by Machine Learning

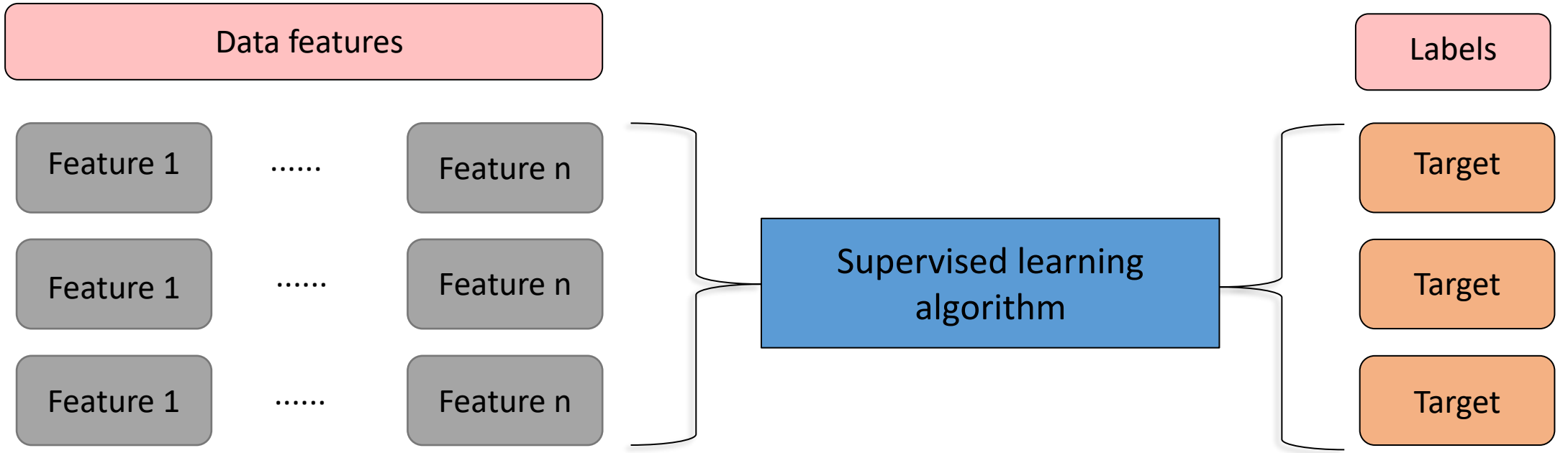
- Machine learning can solve many types of tasks. Three most common types are:
  - **Classification**: To specify a specific one of the *k* categories for the input, the learning algorithm usually outputs a function $f: R^n \rightarrow (1,2,\ldots,k)$. For example, image classification algorithms in computer vision solve classification tasks.
  - **Regression**: The program predicts the output for a given input. The learning algorithms usually output a function $f: R^n \rightarrow R$. Such tasks include predicting the claim amount of a policy holder to set an insurance premium or predicting the security price.
  - **Clustering**: Based on internal similarities, the program groups a large amount of unlabeled data into multiple classes. Same-class data is more similar than data across classes. Clustering tasks include search by image and user profiling.
- **Classification and regression are two major types of prediction tasks. The output of classification is discrete class values, and the output of regression is continuous values.**

Contents

Types of Machine Learning

- **Supervised learning**: The program takes a known set of samples and trains an optimal model to generate predictions. Then, the trained model maps all inputs to outputs and performs simple judgment on the outputs. In this way, unknown data is classified.
- **Unsupervised learning**: The program builds a model based on unlabeled input data. For example, a clustering model groups objects based on similarities. Unsupervised learning algorithms model the highly similar samples, calculate the similarity between new and existing samples, and classify new samples by similarity.
- **Semi-supervised learning**: The program trains a model through a combination of a small amount of labeled data and a large amount of unlabeled data.
- **Reinforcement learning**: The learning systems learn behavior from the environment to maximize the value of reward (reinforcement) signal function. Reinforcement learning differs from supervised learning of connectionism in that, instead of telling the system the correct action, the environment provides scalar reinforcement signals to evaluate its actions.
- Machine learning evolution is producing new machine learning types, for example, self-supervised learning, contrastive learning, generative learning.

Supervised Learning



Weather	Temperature	Wind Speed
Sunny	High	High
Rainy	Low	Medium
Sunny	Low	Low

Suitable for Exercise
Yes
No
Yes

Supervised Learning - Regression

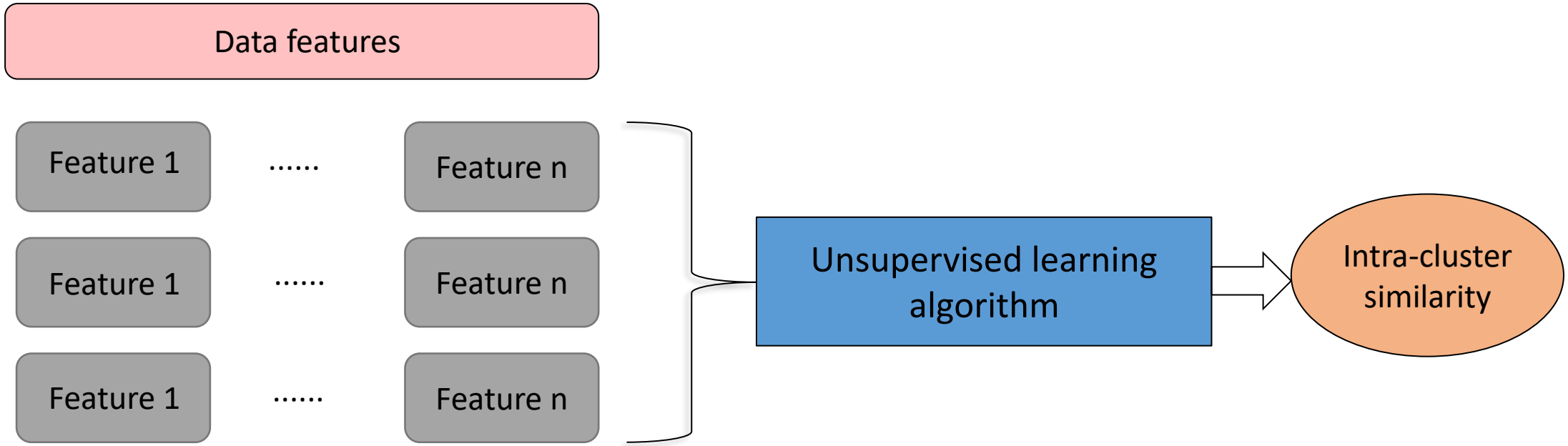
- **Regression** reflects the features of sample attributes in a dataset. A function is used to express the sample mapping relationship and further discover the dependency between attributes. Examples include:
  - How much money can I make from stocks next week?
  - What will the temperature be on Tuesday?

Monday	Tuesday
38°	?

Supervised Learning - Classification

- **Classification** uses a classification model to map samples in a dataset to a given category.
  - What category of garbage does the plastic bottle belong to?
  - Is the email a spam?

Unsupervised Learning



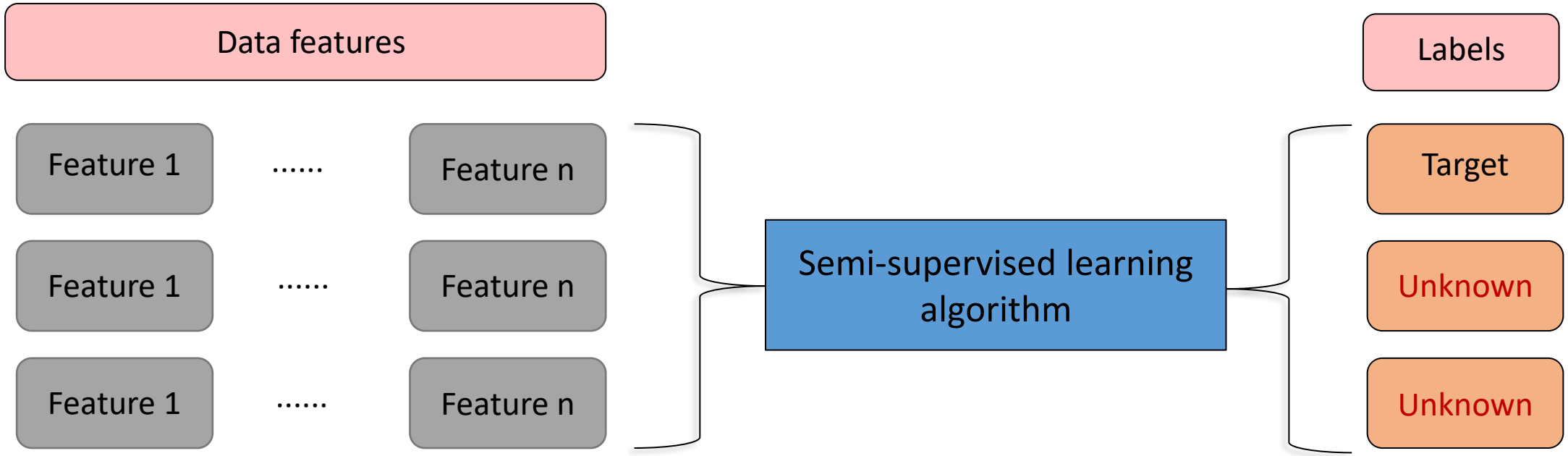
Monthly Sales Volume	Product	Sale Duration
1000-2000	Badminton racket	6:00-12:00
500-1000	Basketball	18:00-24:00
1000-2000	Game console	00:00-6:00

Category
Cluster 1
Cluster 2

Unsupervised Learning - Clustering

- **Clustering** uses a clustering model to classify samples in a dataset into several categories based on similarity.
  - Defining fish of the same species.
  - Recommending movies for users.

Semi-supervised Learning

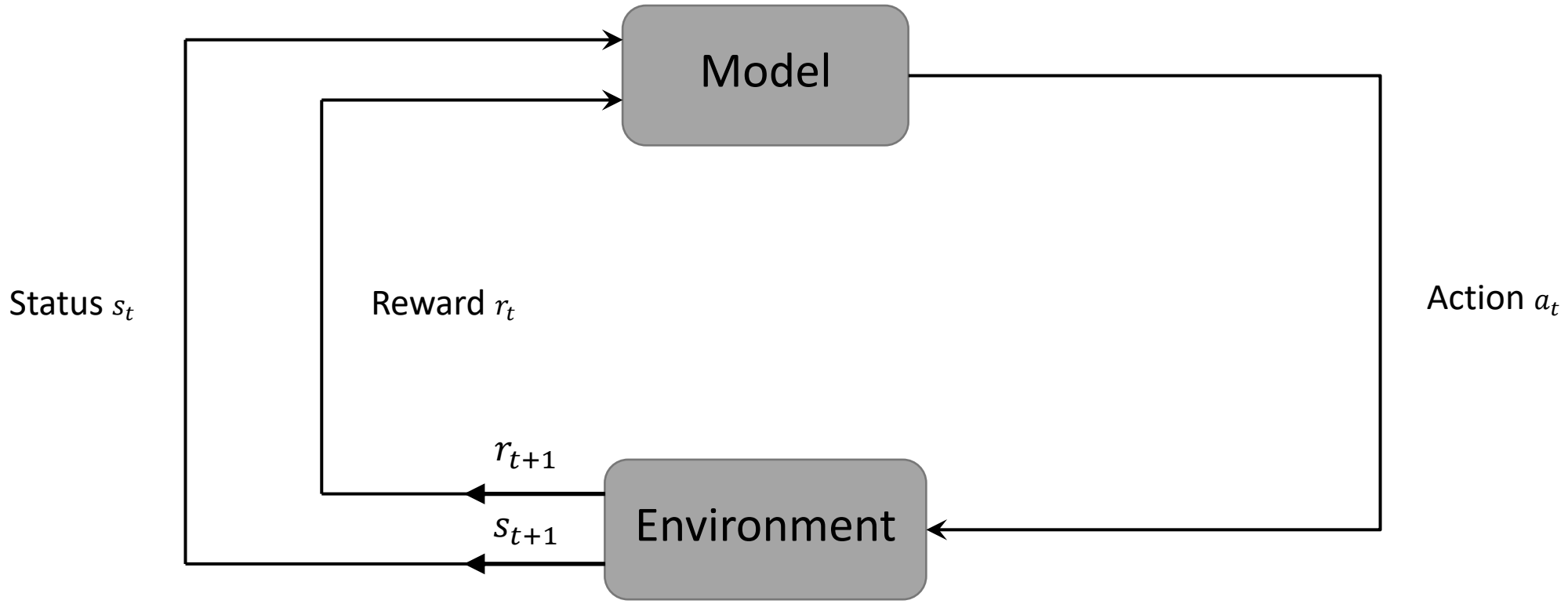


Weather	Temperature	Wind Speed
Sunny	High	High
Rainy	Low	Medium
Sunny	Low	Low

Suitable for Exercise
Yes
/
/

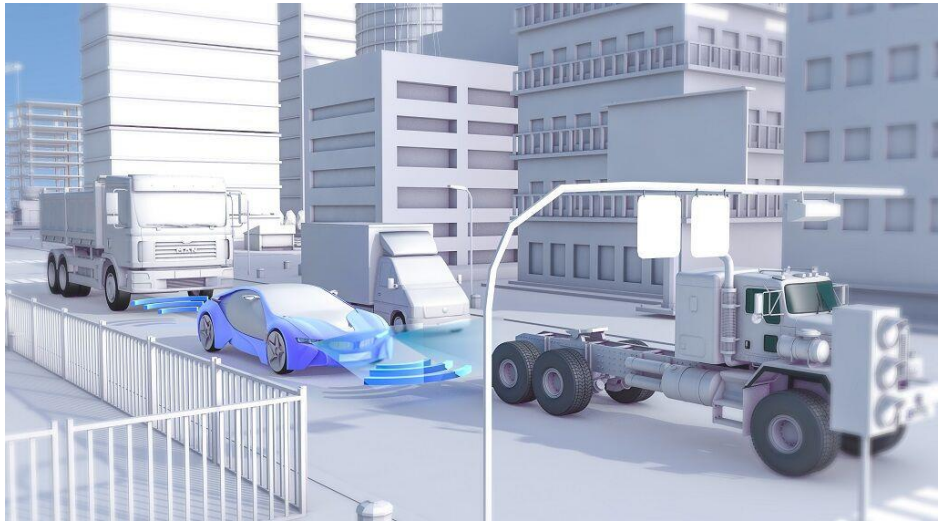
Reinforcement Learning

- A reinforcement learning model learns from the environment, takes actions, and adjusts the actions based on a system of rewards.



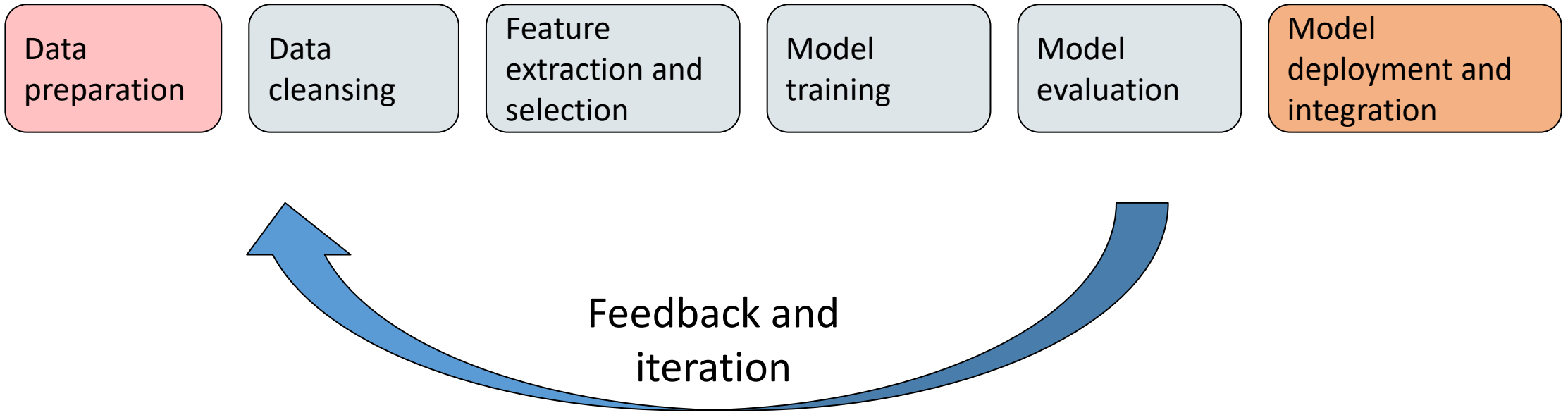
Reinforcement Learning - Best Action

- **Reinforcement learning** always tries to find the best action.
  - Autonomous vehicles: The traffic lights are flashing yellow. Should the vehicle brake or accelerate?
  - Robot vacuum: The battery level is 10%, and a small area is not cleaned. Should the robot continue cleaning or recharge?



Contents

Machine Learning Process



Machine Learning Basic Concept - Dataset

- **Dataset**: collection of data used in machine learning tasks, where each piece of data is called a sample. Items or attributes that reflect the presentation or nature of a sample in a particular aspect are called **features**.
  - **Training set**: dataset used in the training process, where each sample is called a training sample. **Learning (or training)** is the process of building a model from data.
  - **Test set**: dataset used in the testing process, where each sample is called a test sample. Testing refers to the process, during which the learned model is used for prediction.

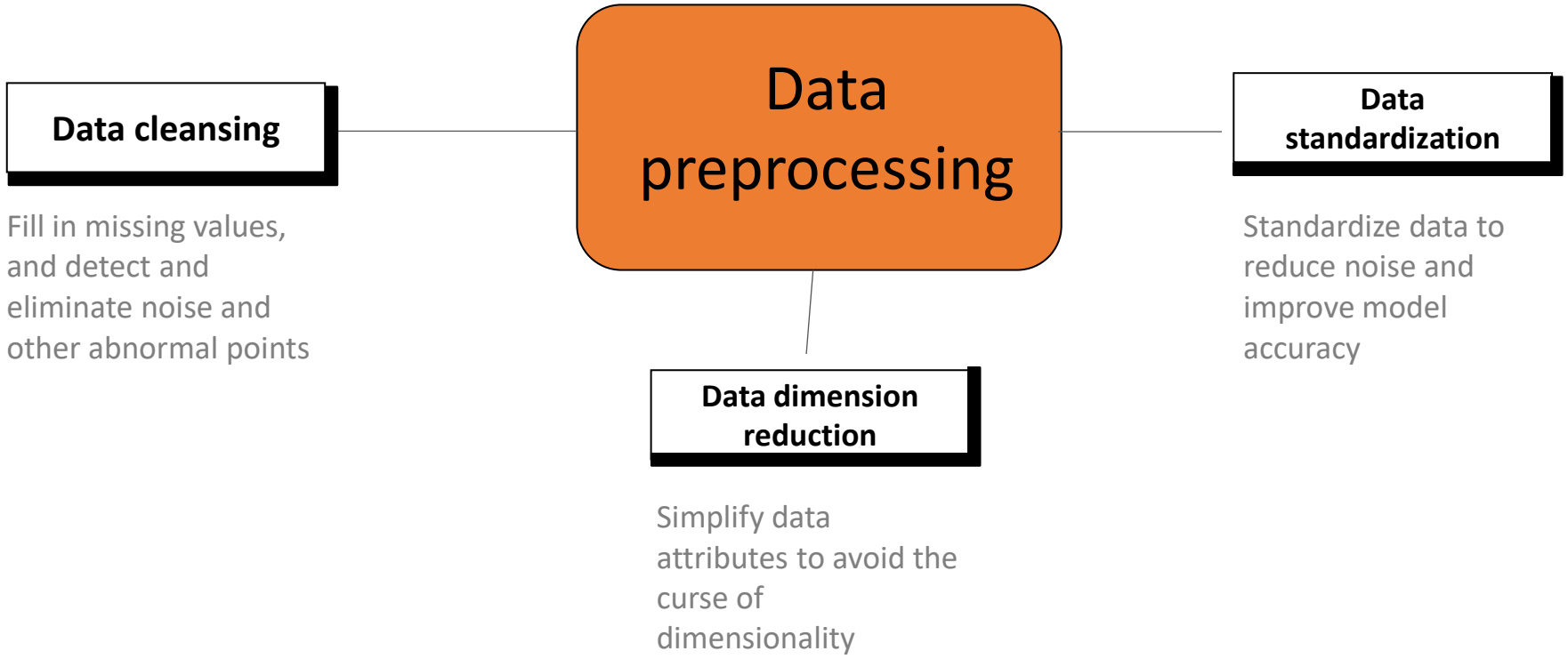
Data Overview

- Typical dataset composition

		Feature 1	Feature 2	Feature 3	Label
	No.	Area	Location	Orientation	House Price
Training set	1	100	8	South	1000
	2	120	9	Southwest	1300
	3	60	6	North	700
	4	80	9	Southeast	1100
Test set	5	95	3	South	850

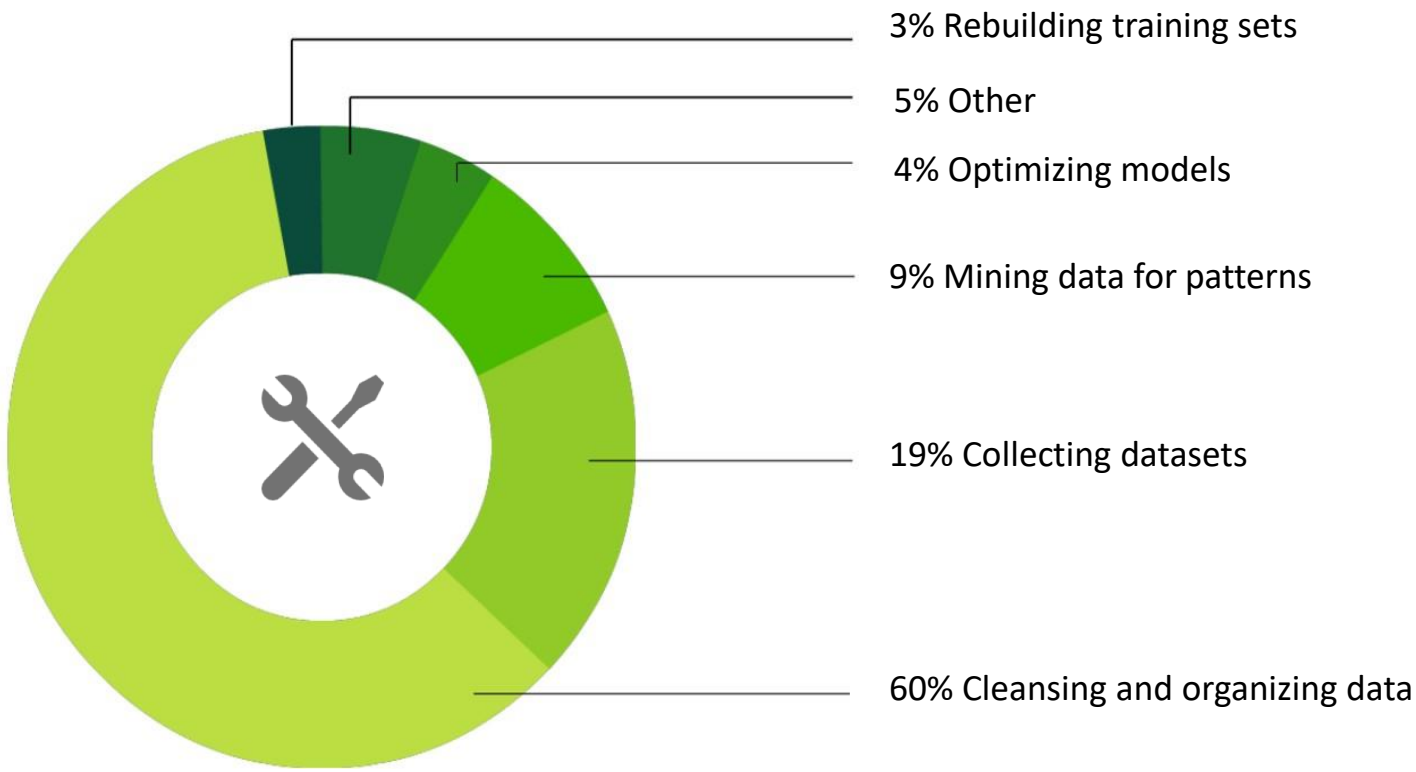
Importance of Data Processing

- Data is crucial to models and determines the scope of model capabilities. All good models require good data.



Data Cleansing Workloads

- What data scientists spend time doing for machine learning:



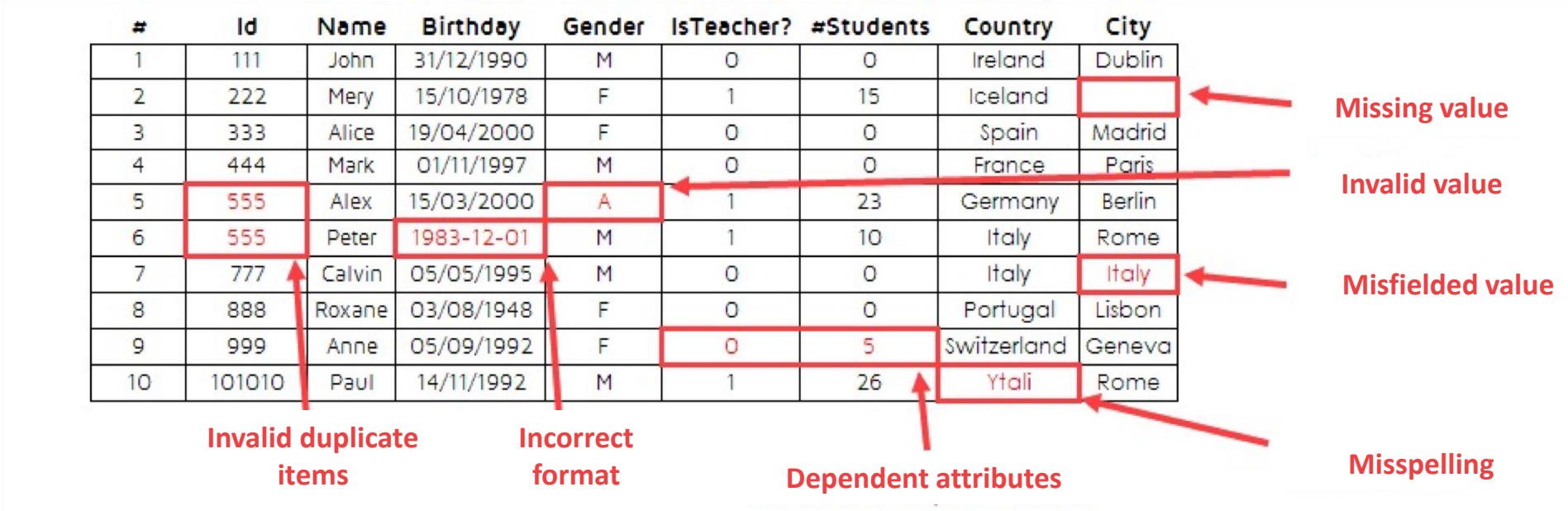
CrowdFlower Data Science Report 2016

Data Cleansing

- Most machine learning models process features, which are usually numeric representations of input variables that can be used in the model.
- In most cases, only preprocessed data can be used by algorithms. Data preprocessing involves the following operations:
  - Data filtering
  - Data loss handling
  - Handling of possible error or abnormal values
  - Merging of data from multiple sources
  - Data consolidation

Dirty Data

- Raw data usually contains data quality problems:
  - Incompleteness: Incomplete data or lack of relevant attributes or values.
  - Noise: Data contains incorrect records or abnormal points.
  - Inconsistency: Data contains conflicting records.

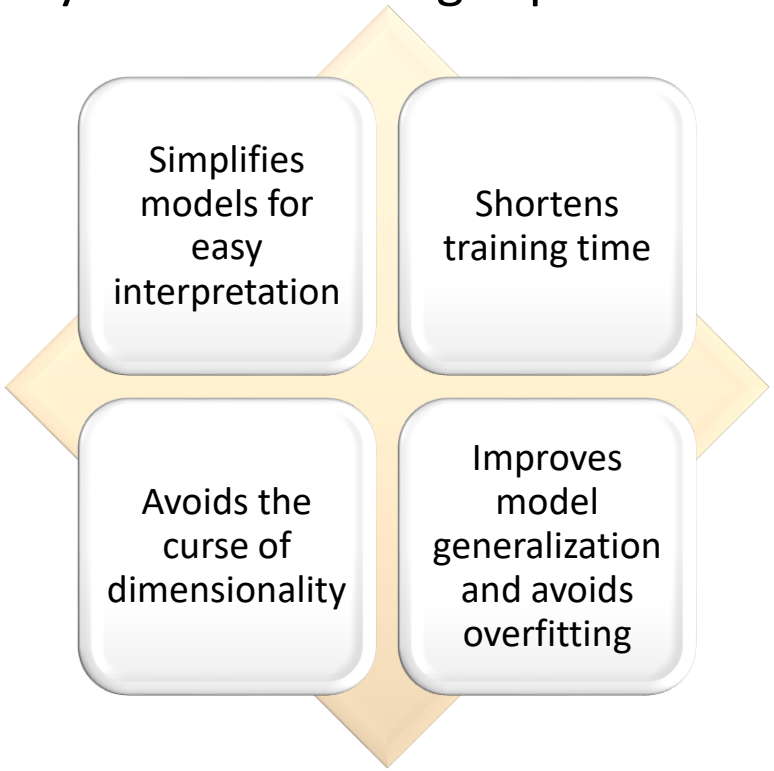


Data Conversion

- Preprocessed data needs to be converted into **a representation suitable for machine learning models**. The following are typically used to convert data:
  - Encoding categorical data into numerals for classification
  - Converting numeric data into categorical data to reduce the values of variables (for example, segmenting age data)
  - Other data:
    - Embedding words into text to convert them into word vectors (Typically, models such as word2vec and BERT are used.)
    - Image data processing, such as color space conversion, grayscale image conversion, geometric conversion, Haar-like features, and image enhancement
  - Feature engineering:
    - Normalizing and standardizing features to ensure that different input variables of a model fall into the same value range
    - Feature augmentation: combining or converting the existing variables to generate new features, such as averages.

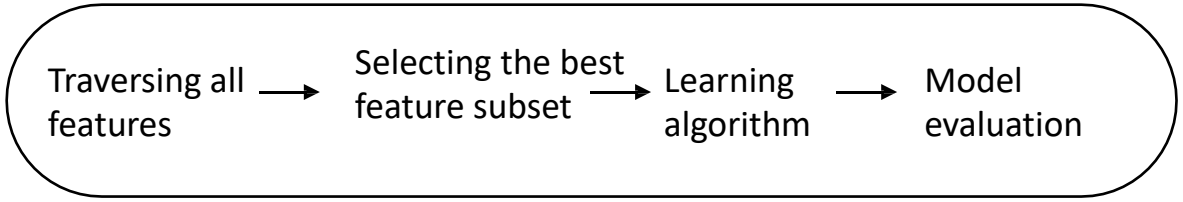
Necessity of Feature Selection

- Generally, a dataset has many features, some of which may be **unnecessary or irrelevant to the values to be predicted**.
- Feature selection is necessary in the following aspects:



Feature Selection Methods - Filter

- Filter methods are independent of models during feature selection.



Filter method process

By evaluating the correlation between each feature and target attribute, a filter method scores each feature using a statistics measurement and then sorts the features by score. This can preserve or eliminate specific features.

Common methods:

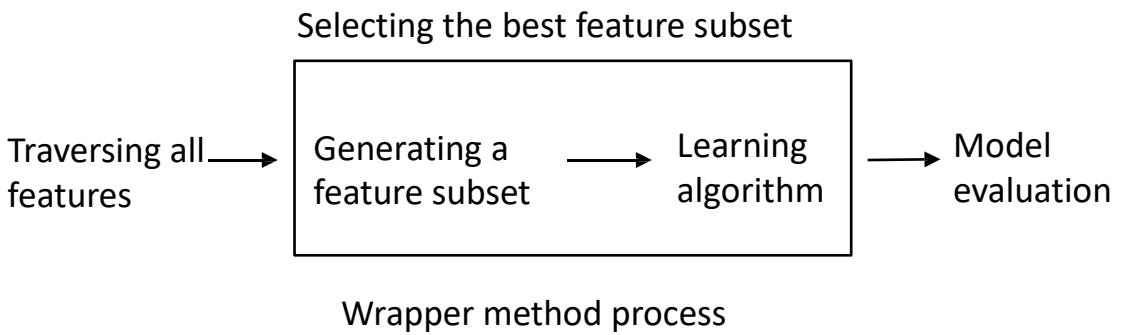
- Pearson correlation coefficient
- Chi-square coefficient
- Mutual information

Limitations of filter methods:

- Filter methods tend to select redundant variables because they do not consider the relationships between features.

Feature Selection Methods - Wrapper

- Wrapper methods use a prediction model to score a feature subset.



Wrapper method process

Wrapper methods treat feature selection as a search issue and evaluate and compare different combinations. Wrapper methods use a predictive model to evaluate the different feature combinations, and score the feature subsets by model accuracy.

Common method:

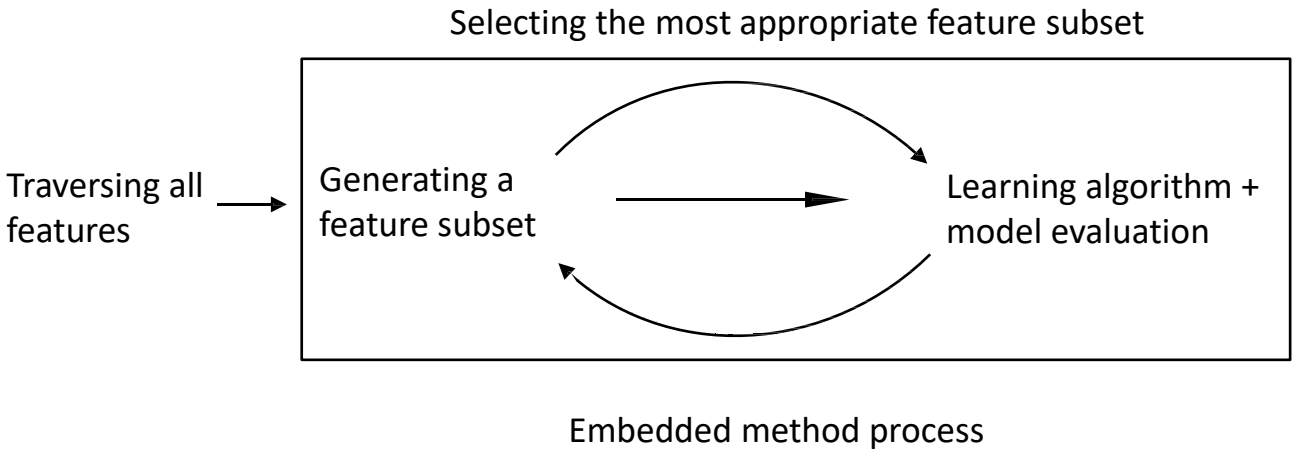
- Recursive feature elimination

Limitations of wrapper methods:

- Wrapper methods train a new model for each feature subset, which can be **computationally intensive**.
- Wrapper methods usually provide high-performance feature sets for **a specific type of model**.

Feature Selection Methods - Embedded

- Embedded methods treat feature selection as a part of the modeling process.



Embedded method process

Regularization is the most common type of embedded methods.

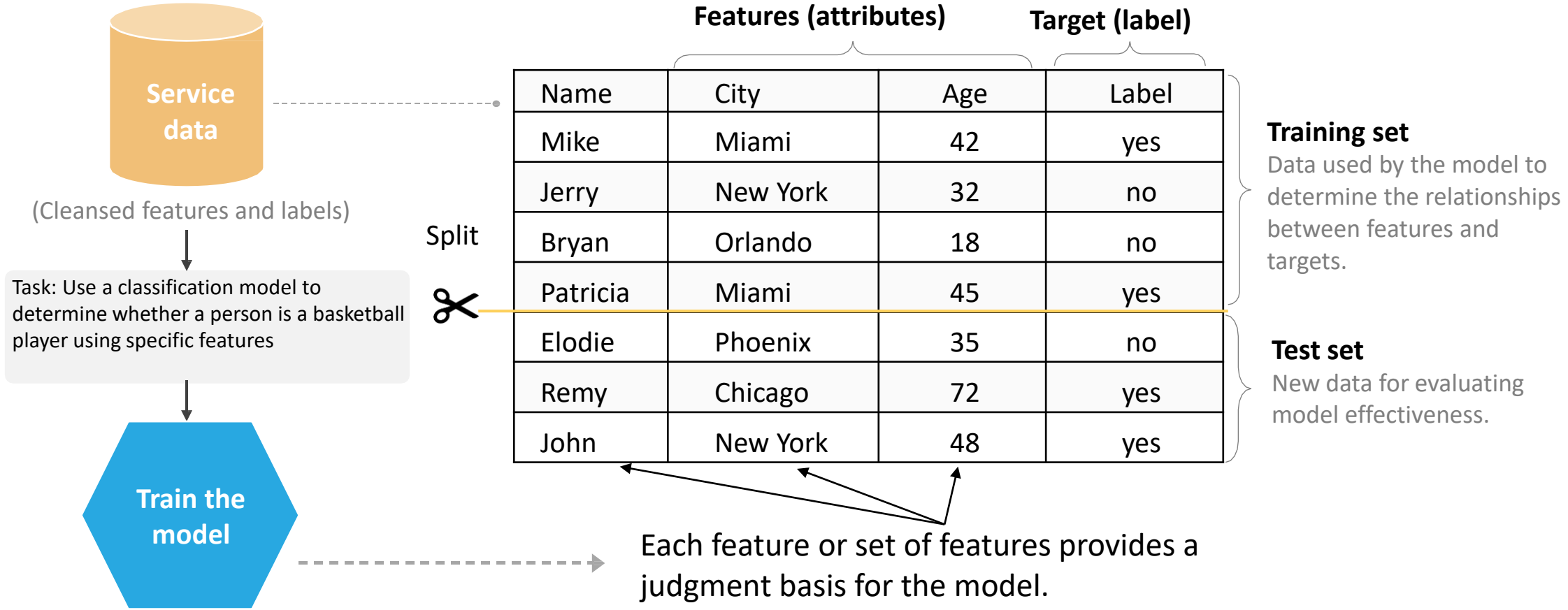
Regularization methods, also called penalization methods, introduce additional constraints into the optimization of a predictive algorithm to bias the model toward lower complexity and reduce the number of features.

Common method:

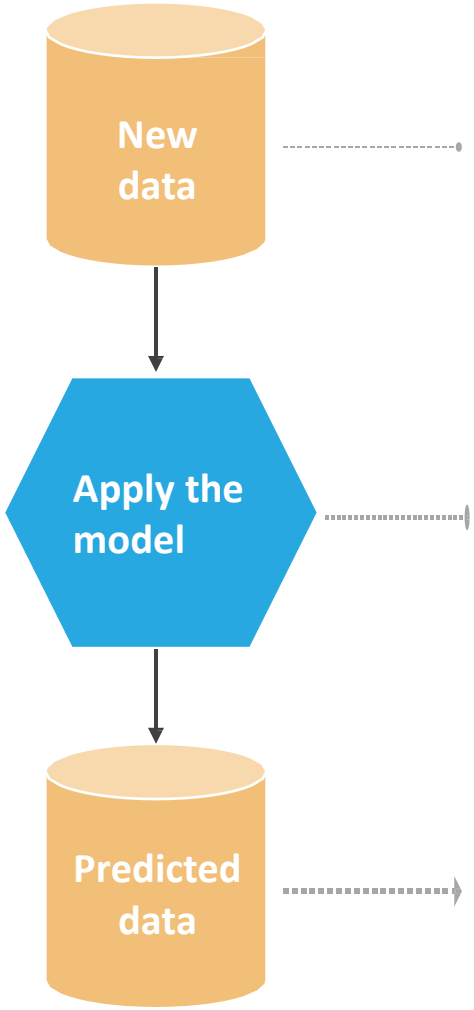
- LASSO regression

Supervised Learning Example - Learning Phase

- Use a classification model to determine whether a person is a basketball player based on specific features.



Supervised Learning Example - Prediction Phase



Name	City	Age	Label
Marine	Miami	45	?
Julien	Miami	52	?
Fred	Orlando	20	?
Michelle	Boston	34	?
Nicolas	Phoenix	90	?

Unknown data

Recent data cannot determine whether they are a basketball player.

IF city = Miami → Probability = +0.7
IF city= Orlando → Probability = +0.2
IF age > 42 → Probability = +0.05*age + 0.06
IF age <= 42 → Probability = +0.01*age + 0.02

Name	City	Age	Prediction
Marine	Miami	45	**0.3**
Julien	Miami	52	**0.9**
Fred	Orlando	20	**0.6**
Michelle	Boston	34	**0.5**
Nicolas	Phoenix	90	**0.4**

Predicted possibility

Use the model against new data to predict the possibility of them being basketball players.

What Is a Good Model?

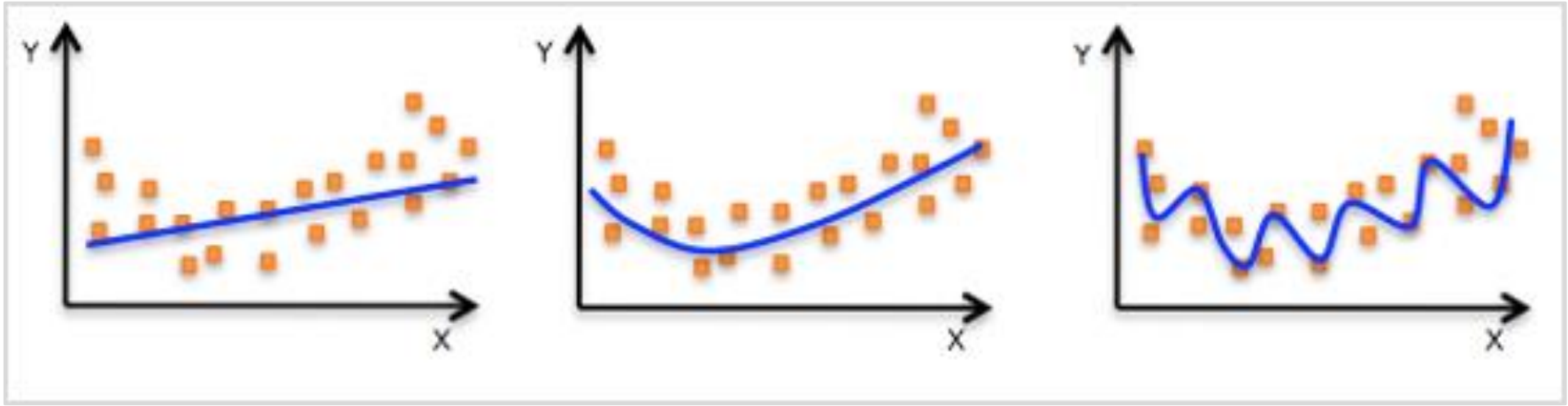
- **Generalization**
  The accuracy of predictions based on actual data
- **Explainability**
  Predicted results are easy to explain
- **Prediction speed**
  The time needed to make a prediction

Model Effectiveness (1)

- **Generalization capability**: Machine learning aims to ensure models perform well on new samples, not just those used for training. Generalization capability, also called **robustness**, is the extent to which a learned model can be applied to new samples.
- **Error** is the difference between the prediction of a learned model on a sample and the actual result of the sample.
  - Training error is the error of the model on the training set.
  - Generalization error is the error of the model on new samples. Obviously, we prefer a model with a smaller generalization error.
- **Underfitting**: The training error is large.
- **Overfitting**: The training error of a trained model is small while the generalization error is large.

Model Effectiveness (2)

- **Model capacity**, also known as model complexity, is the capability of the model to fit various functions.
  - With sufficient capacity to handle task complexity and training data volumes, the algorithm results are optimal.
  - Models with an insufficient capacity cannot handle complex tasks because **underfitting** may occur.
  - Models with a large capacity can handle complex tasks, but **overfitting** may occur when the capacity is greater than the amount required by a task.



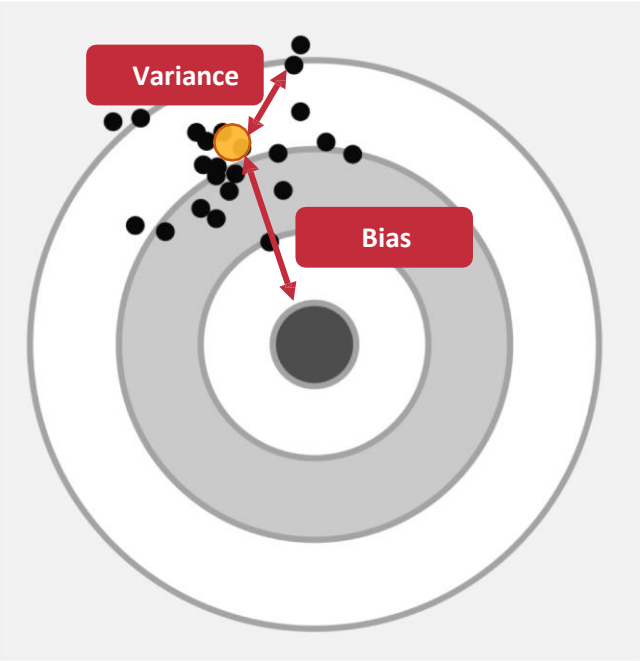
Underfitting:
features not learned

Good fitting

Overfitting:
noises learned

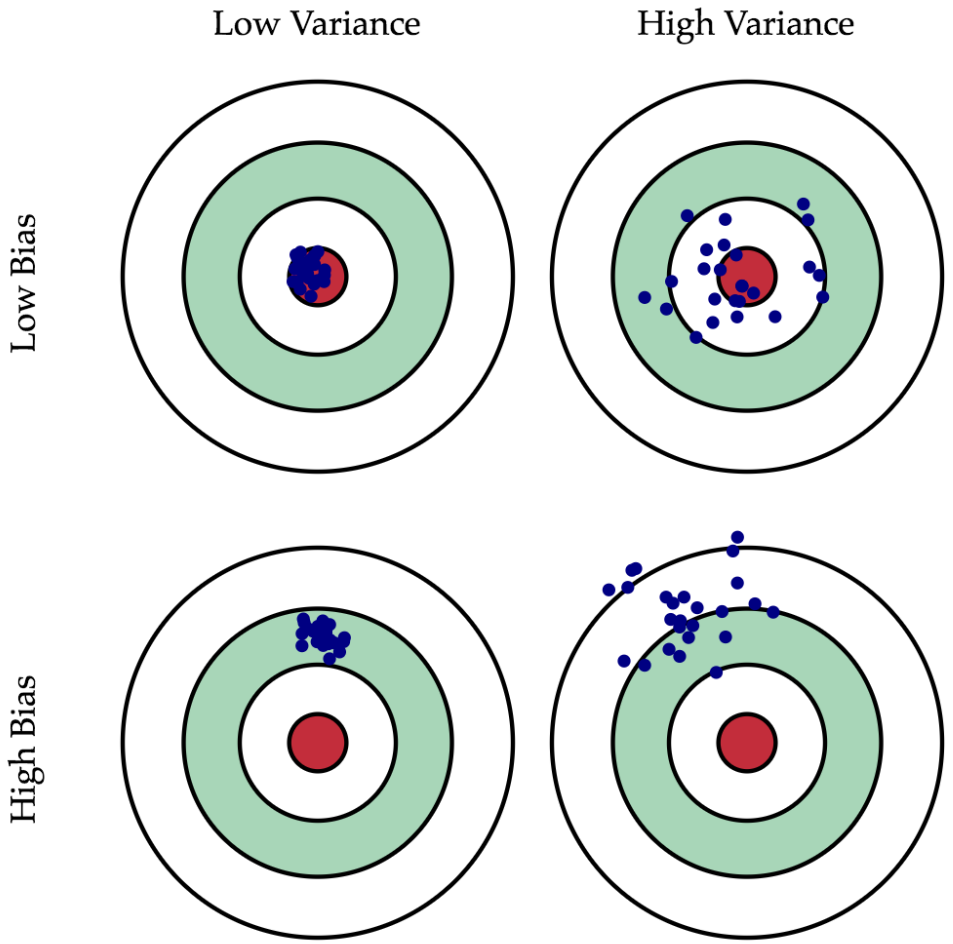
Cause of Overfitting - Errors

- Prediction error = **Bias**^{2} + **Variance** + Ineliminable error
- In general, the two main factors of prediction error are variance an bias.
- Variance:
  - How much a prediction result deviates from the mean
  - Variance is caused by the sensitivity of the model to small fluctuations a training set.
- Bias:
  - Difference between the average of the predicted values and the actual values.



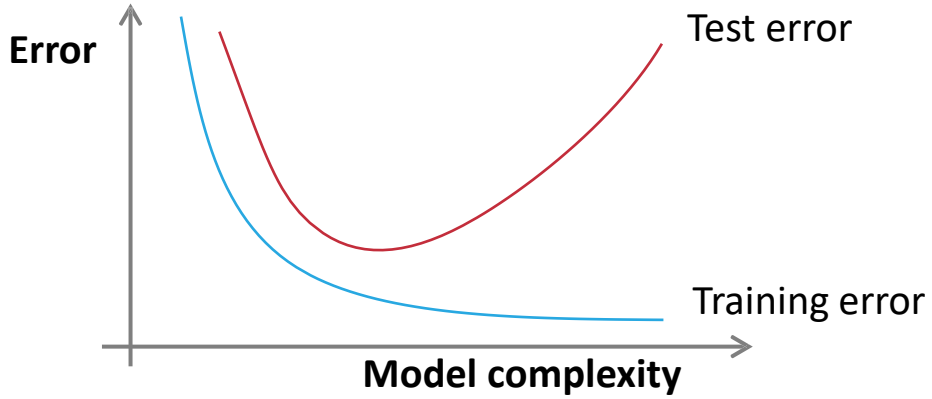
Variance and Bias

- Different combinations of variance and bias are as follows:
  - Low bias & low variance → good model
  - Low bias & high variance → inadequate model
  - High bias & low variance → inadequate model
  - High bias & high variance → bad model
- An ideal model can accurately capture the rules in the training data and be generalized to invisible (new) data. However, it is impossible for a model to complete both tasks at the same time.



Complexity and Errors of Models

- The more complex a model is, the smaller its training error is.
- As the model complexity increases, the test error decreases before increasing again, forming a convex curve.



Performance Evaluation of Machine Learning - Regression

- Mean absolute error (MAE). An MAE value closer to 0 indicates the model fits the training data better.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$$

- Mean squared error (MSE).

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

- The value range of R^2 is [0,1]. A larger value indicates that the model fits the training data better. TSS indicates the difference between samples, and RSS indicates the difference between the predicted values and sample values.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y}_i)^2}$$

Performance Evaluation of Machine Learning - Classification (1)

- Terms:
  - P: positive, indicating the number of real positive cases in the data.
  - N: negative, indicating the number of real negative cases in the data.
  - TP : true positive, indicating the number of positive cases that are correctly classified.
  - TN: true negative, indicating the number of negative cases that are correctly classified.
  - FP: false positive, indicating the number of positive cases that are incorrectly classified.
  - FN: false negative, indicating the number of negative cases that are incorrectly classified.

Actual \ Predicted	Yes	No	Total
Yes	TP	FN	P
No	FP	TN	N
Total	P'	N'	$P+N$

Confusion matrix

- The confusion matrix is an $m \times m$ table at minimum. The entry $CM_{i,j}$ in the first m rows and m columns indicates the number of cases that belong to class i but are labeled as j.
  - For classifiers with high accuracy, most of the cases should be represented by entries on the diagonal of the confusion matrix from $CM_{1,1}$ to $CM_{m,m}$,while other entries are 0 or close to 0. That is, FP and FN are close to 0.

Performance Evaluation of Machine Learning - Classification (2)

Measurement	Formula
Accuracy, recognition rate	$\frac{TP + TN}{P + N}$
Error rate, misclassification rate	$\frac{FP + FN}{P + N}$
True positive rate, sensitivity, recall	$\frac{TP}{P}$
True negative rate, specificity	$\frac{TN}{N}$
Precision	$\frac{TP}{TP + FP}$
F_1 value, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β value, where β is a non-negative real number	$\frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

Performance Evaluation of Machine Learning - Example

- In this example, an ML model was trained to identify an image of a cat. To evaluate the model's performance, 200 images were used, of which 170 of them were cats.
- The model reported that 160 images were cats.

Precision: $P = \frac{TP}{TP+FP} = \frac{140}{140+20} = 87.5\%$

Recall: $R = \frac{TP}{P} = \frac{140}{170} = 82.4\%$

Accuracy: $ACC = \frac{TP+TN}{P+N} = \frac{140+10}{170+30} = 75\%$

Predicted / Actual	yes	no	Total
yes	140	30	170
no	20	10	30
Total	160	40	200

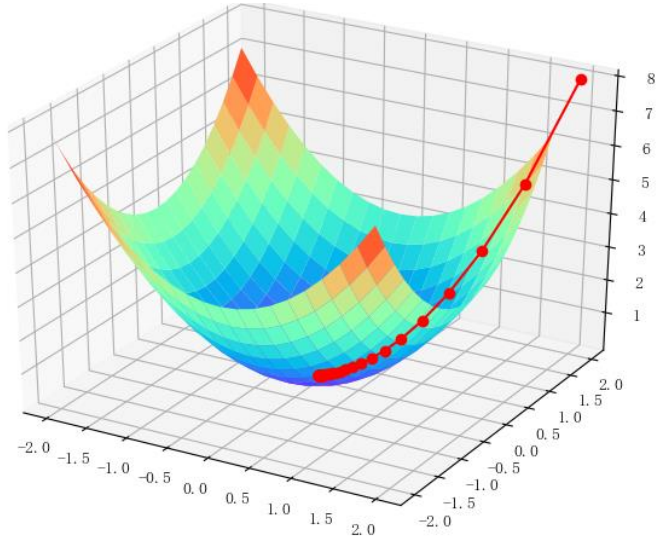
Contents

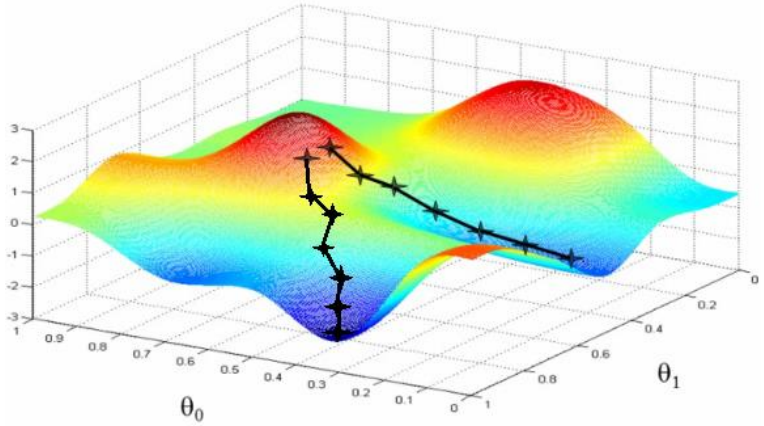
Machine Learning Training Methods - Gradient Descent (1)

- This method uses the negative gradient direction of the current position as the search direction, which is the fastest descent direction of the current position. The formula is as follows:

$$w_{k+1} = w_k - \eta \nabla f_{w_k}(x^i)$$

- η is the learning rate. i indicates the i-th data record. $\eta \nabla f_{w_k}(x^i)$ indicates the change of weight parameter w in each iteration.
- Convergence means that the value of the objective function changes very little or reaches the maximum number of iterations.





Machine Learning Training Methods - Gradient Descent (2)

- Batch gradient descent (BGD) uses the sum of gradients of all m samples of the dataset at the current point to update the weight parameter.

$$w_{k+1} = w_k - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla f_{w_k}(x^i)$$

- Stochastic gradient descent (SGD) randomly uses the gradient of a random sample of the dataset at the current point to update the weight parameter in the current gradient.

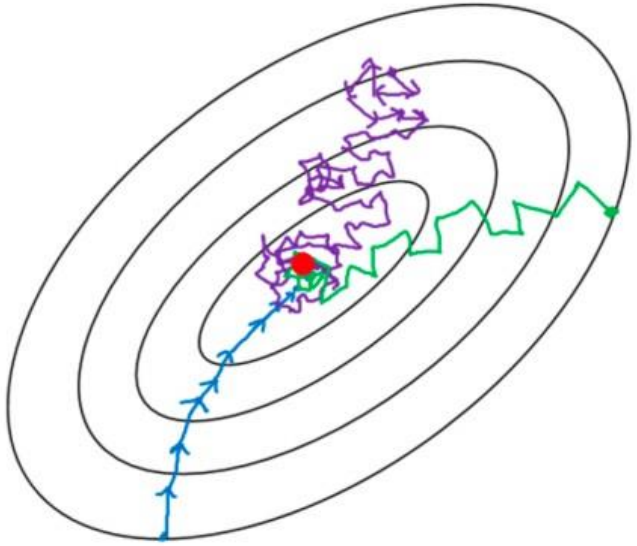
$$w_{k+1} = w_k - \eta \nabla f_{w_k}(x^i)$$

- Mini-batch gradient descent (MBGD) combines the features of BGD and SGD, and chooses the gradients of n samples in a dataset each time to update the weight parameter.

$$w_{k+1} = w_k - \eta \frac{1}{n} \sum_{i=\mathrm{t}}^{t+n-1} \nabla f_{w_k}(x^i)$$

Machine Learning Training Methods - Gradient Descent (3)

- Comparison of gradient descent methods
  - SGD randomly chooses samples for each training pass, causing instability. As a result, the loss function fluctuates or even produces reverse displacement during the process of dropping to the minimum.
  - BGD is the most stable, but it consumes too many compute resources. MBGD is a balance between BGD and SGD.



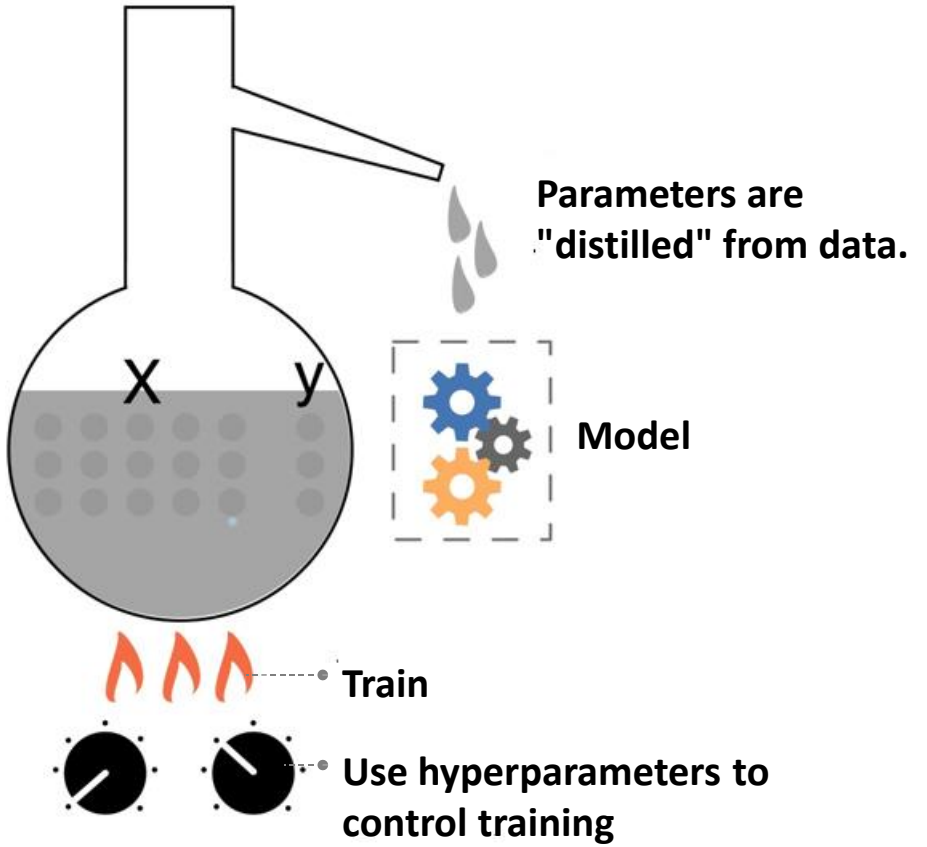
BGD
Use **all** training samples for each training pass.

SGD
One training sample is used for each training pass.

MBGD
A certain number of training samples are used for each training pass.

Parameters and Hyperparameters

- A model contains not only parameters but also hyperparameters. Hyperparameters enable the model to learn the optimal configurations of the parameters.
  - Parameters are automatically learned by models.
  - Hyperparameters are manually set.



Hyperparameters

- Commonly used for model parameter estimation.
- Specified by the user.
- Set heuristically.
- Often tuned for a given predictive modeling problem.

Hyperparameters are configurations outside the model.

- λ of Lasso/Ridge regression
- Learning rate, number of iterations, batch size, activation function, and number of neurons of a neural network to be trained
- C and σ of support vector machines (SVMs)
- k in the k-nearest neighbors (k-NN) algorithm
- Number of trees in a random forest

Common hyperparameters

Hyperparameter Search Process and Methods

Hyperparameter search general process

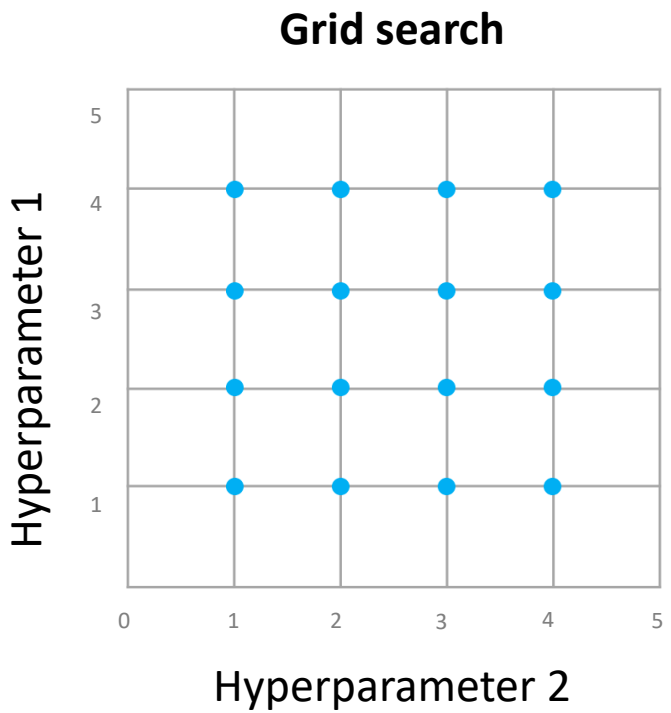
1. Divide a dataset into a training set, validation set, and test set.
2. Optimize the model parameters using the training set based on the model performance metrics.
3. Search for model hyperparameters using the validation set based on model performance metrics.
4. Perform step 2 and step 3 alternately until model parameters and hyperparameters are determined, and assess the model using the test set.

Search algorithms (step 3)

- **Grid search**
- **Random search**
- Heuristic intelligent search
- Bayesian search

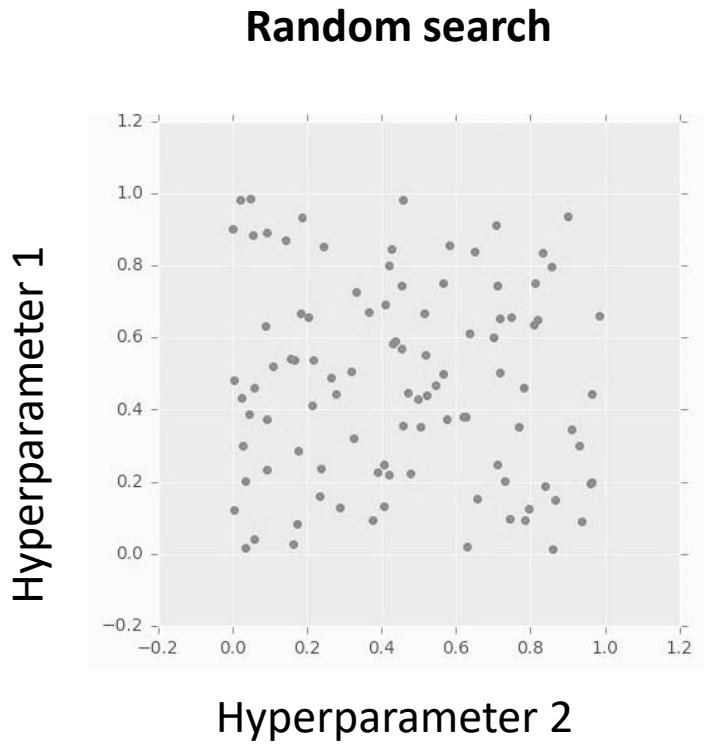
Hyperparameter Tuning Methods - Grid Search

- Grid search performs an **exhaustive search** of all possible hyperparameter combinations to form a hyperparameter value grid.
- In practice, the hyperparameter ranges and steps are manually specified.
- Grid search is expensive and time-consuming.
  - This method works well when there are relatively few hyperparameters. Therefore, it is feasible for general machine learning algorithms, but not for neural networks (see the deep learning course).



Hyperparameter Tuning Methods - Random Search

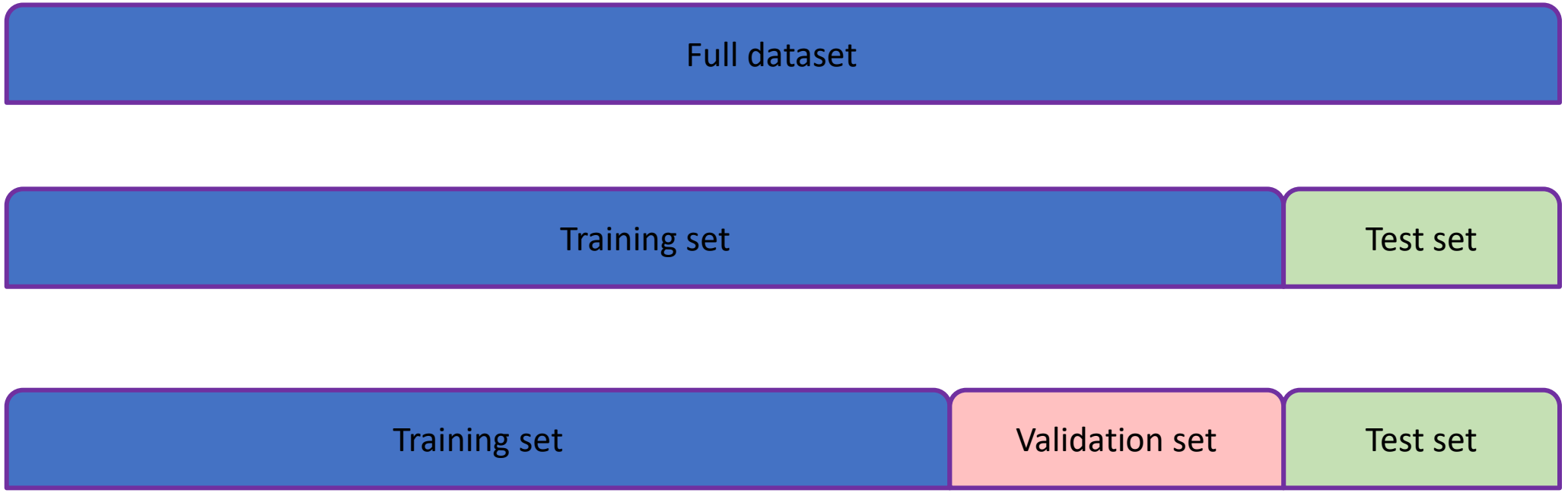
- If the hyperparameter search space is large, **random search** is more appropriate than grid search.
- In a random search, each setting item is sampled from possible parameter values to find the most appropriate parameter subset.
- Note:
  - In a random search, a search is first performed within a broad range, and then the range is narrowed based on the location of the best result.
  - Some hyperparameters are more important than others and affect random search preferences.



Cross-Validation (1)

- **Cross-validation** is a statistical analysis method used to check the performance of classifiers. It splits the original data into the training set and validation set. The former is used to train a classifier, whereas the latter is used to evaluate the classifier by testing the trained model.
- ***k*-fold cross-validation (*k*-fold CV):**
  - Divides the original data into k (usually equal-sized) subsets.
  - Each unique group is treated as a validation set, and the remaining $k-1$ groups are treated as the training set. In this way, k models are obtained.
  - The average classification accuracy score of the k models on the validation set is used as the performance metric for *k*-fold CV classifiers.

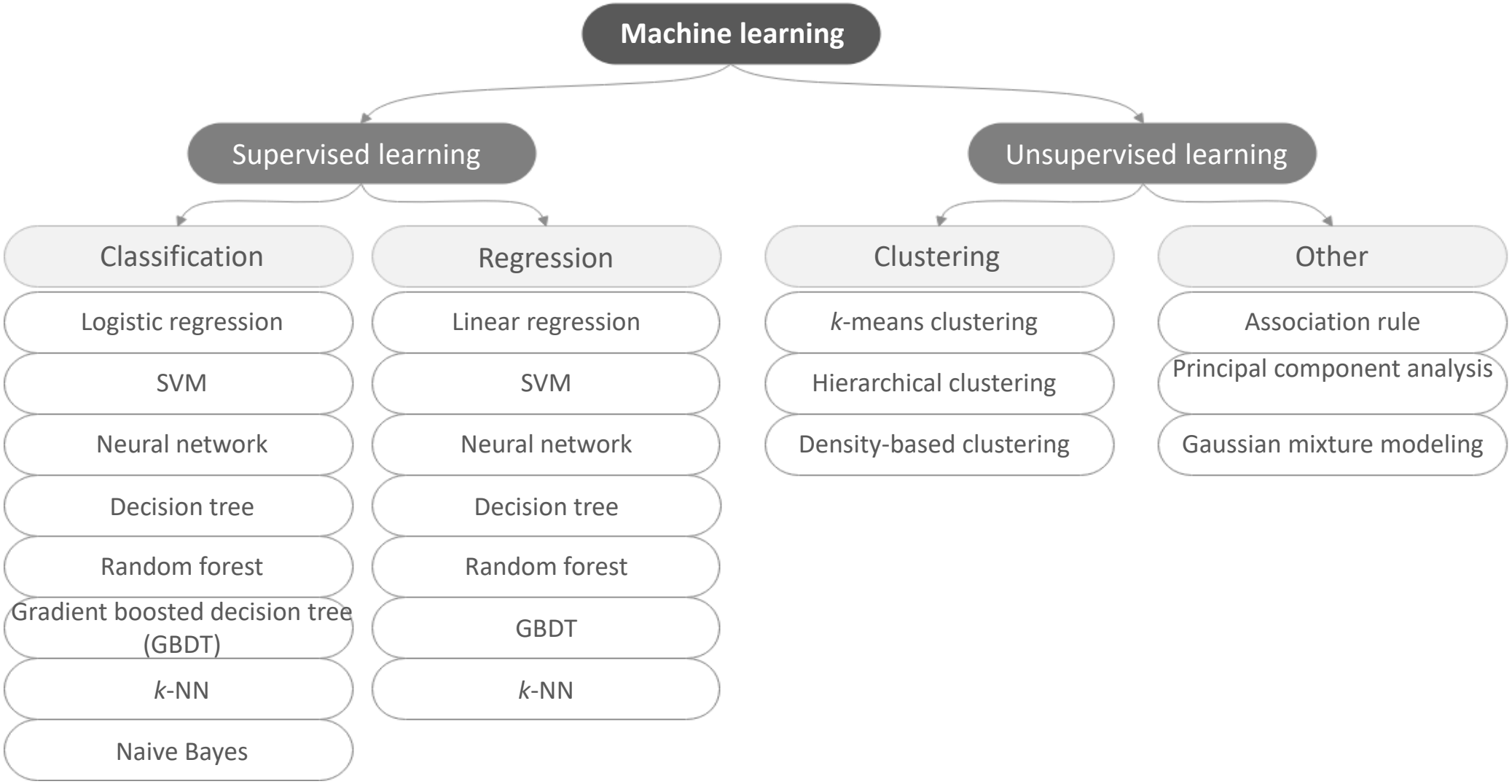
Cross-Validation (2)



- Note: *k* in *k*-fold CV is a hyperparameter.

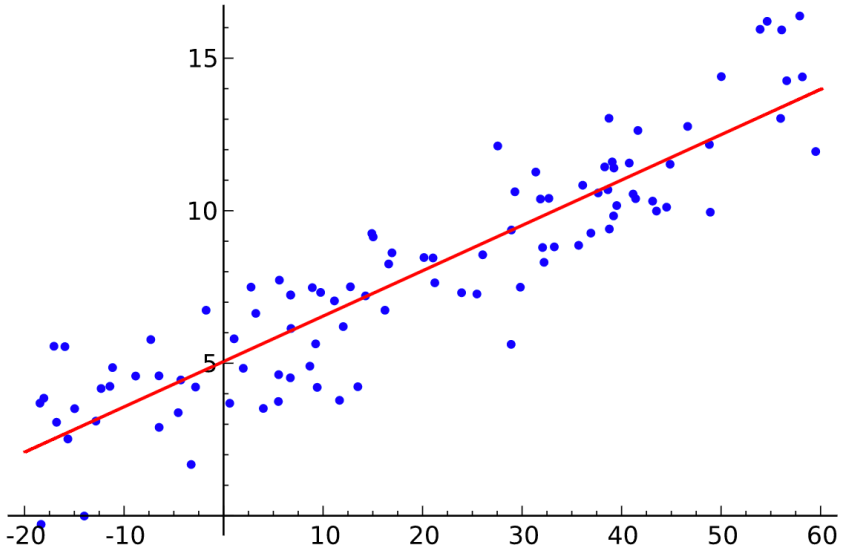
Contents

Machine Learning Algorithm Overview

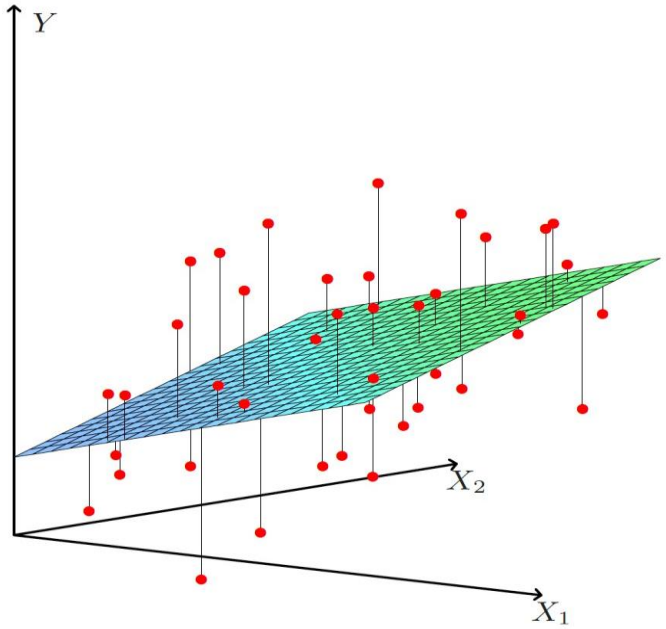


Linear Regression (1)

- Linear regression uses the regression analysis of mathematical statistics to determine the quantitative relationship between two or more variables.
- Linear regression is a type of supervised learning.



Simple linear regression



Multiple linear regression

Linear Regression (2)

- The model function of linear regression is as follows, where w is the weight parameter, b is the bias, and x represents the sample:

$$h_w(x) = w^T x + b$$

- The relationship between the value predicted by the model and the actual value is as follows, where y indicates the actual value, and ε indicates the error:

$$y = w^T x + b + \varepsilon$$

- The error ε is affected by many independent factors. Linear regression assumes that the error ε follows normal distribution. The loss function of linear regression can be obtained using the normal distribution function and maximum likelihood estimation (MLE):

$$J(w) = \frac{1}{2m}\sum\left(h_w(x) - y\right)^2$$

- We want the predicted value approaches the actual value as far as possible, that is, to minimize the loss value. We can use a gradient descent algorithm to calculate the weight parameter w when the loss function reaches the minimum, thereby complete model building.

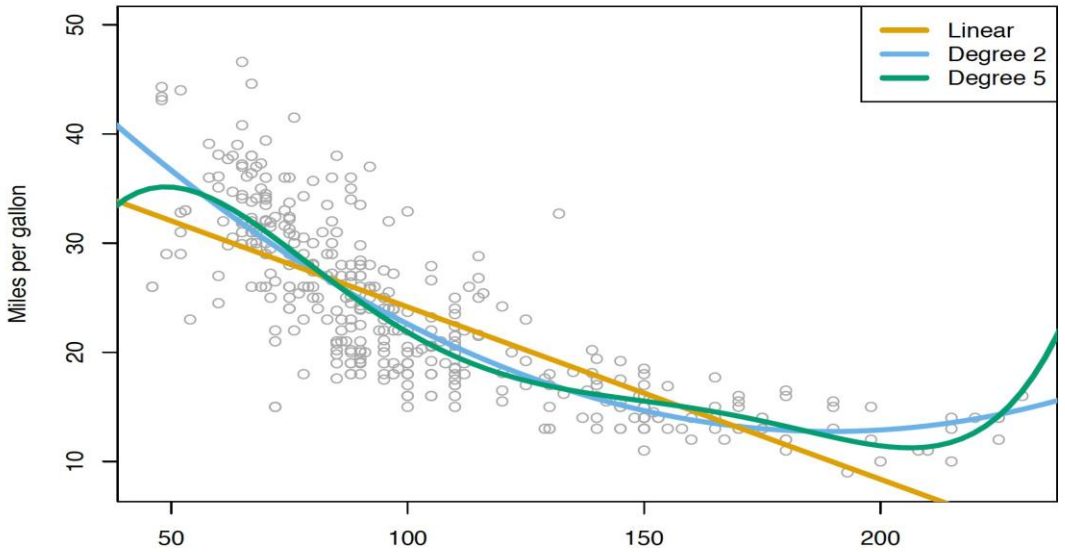
Linear Regression Extension - Polynomial Regression

- Polynomial regression is an extension of linear regression. Because the complexity of a dataset exceeds the possibility of fitting performed using a straight line (obvious underfitting occurs if the original linear regression model is used), polynomial regression is used.

$$h_w(x) = w_1 x + w_2 x^2 + \cdots + w_n x^n + b$$

Here, n-th power indicates the degree of the polynomial.

Polynomial regression is a type of linear regression. Although its features are non-linear, the relationship between its weight parameters w is still linear.



Comparison between linear and polynomial regression

Preventing Overfitting of Linear Regression

- Regularization terms help reduce overfitting. The w value cannot be too large or too small in the sample space. You can add a square sum loss to the target function:

$$J(w) = \frac{1}{2m}\sum(h_w(x) - y)^2 + \lambda\|w\|_2^2$$

- Regularization term: This regularization term is called L2-norm. Linear regression that uses this loss function is called **Ridge regression**.

$$J(w) = \frac{1}{2m}\sum(h_w(x) - y)^2 + \lambda\|w\|_1$$

- Linear regression with an absolute loss is called **Lasso regression**.

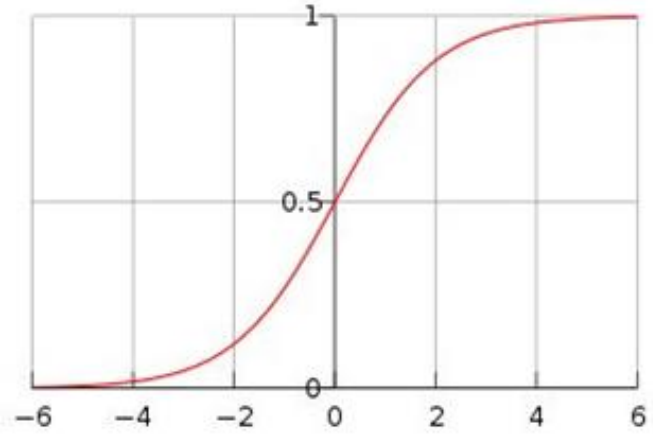
Logistic Regression (1)

- The logistic regression model is a classification model used to resolve classification problems. The model is defined as follows:

$$P(Y=0|x)=\frac{e^{-(wx+b)}}{1+e^{-(wx+b)}}$$

$$P(Y=1|x)=\frac{1}{1+e^{-(wx+b)}}$$

w represents the weight, b represents the bias, and $wx+b$ represents a linear function with respect to x. Compare the preceding two probability values. x belongs to the type with a larger probability value.



Logistic Regression (2)

- Logistic regression and linear regression are both linear models in broad sense. The former introduces a non-linear factor (sigmoid function) on the basis of the latter and sets a threshold. Therefore, logistic regression applies to binary classification.
- According to the model function of logistic regression, the loss function of logistic regression can be calculated through maximum likelihood estimation as follows:

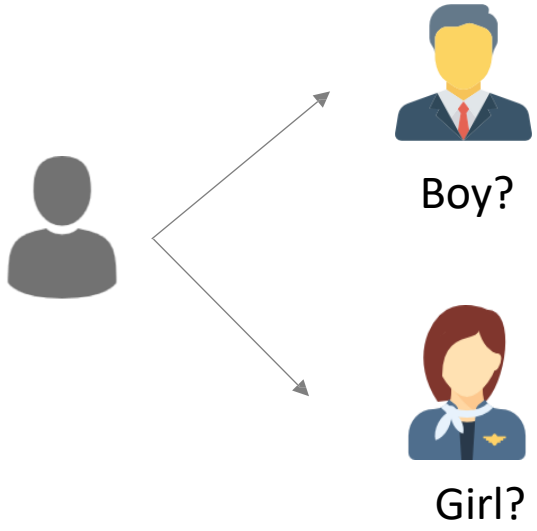
$$J(w) = -\frac{1}{m}\sum\left(y \ln h_w(x) + (1-y)\ln(1-h_w(x))\right)$$

- In the formula, w indicates the weight parameter, m indicates the number of samples, x indicates the sample, and y indicates the actual value. You can also obtain the values of all the weight parameters w by using a gradient descent algorithm.

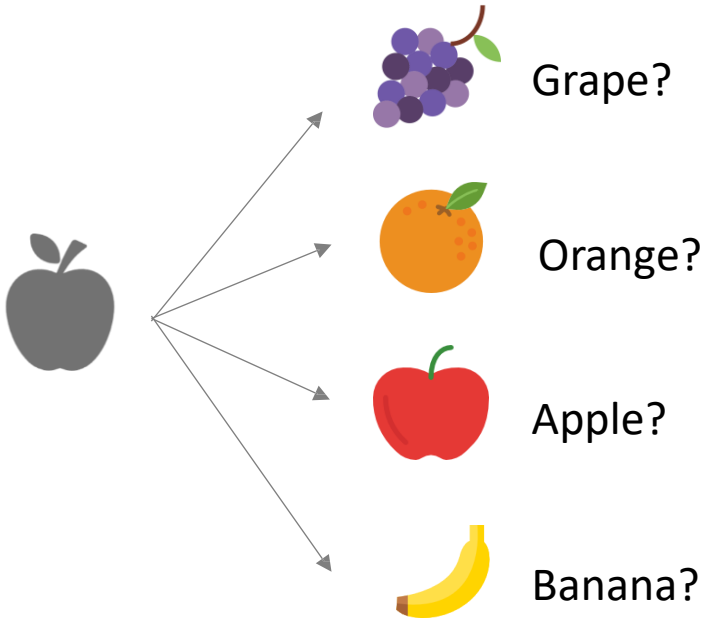
Logistic Regression Extension - Softmax (1)

- Logistic regression mainly applies to binary classification. For multi-class classification, the softmax function is typically used.

Binary classification problem



Multi-class classification problem



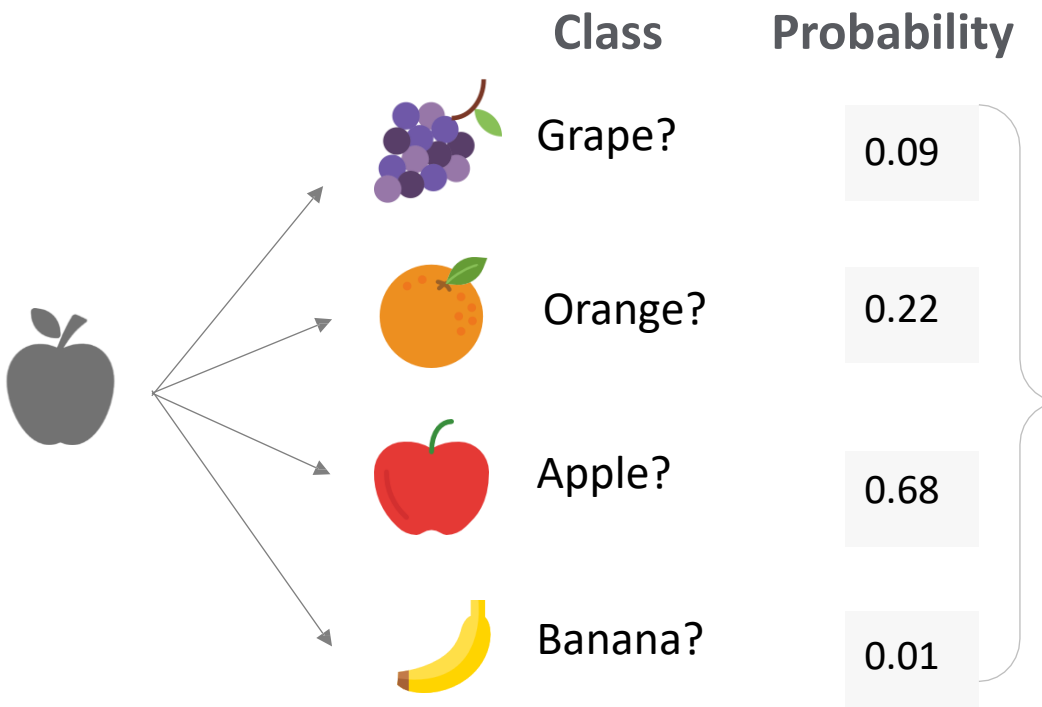
Logistic Regression Extension - Softmax (2)

- Softmax regression is a generalization of logistic regression and applies to *k*-class classification.
- The softmax function compresses (maps) a *k*-dimensional vector of arbitrary real values to another *k*-dimensional vector of real values, where each vector element is in (0, 1).
- The Softmax regression probability function is:

$$p(y=k \mid x;w)=\frac{e^{w_k^T x}}{\sum_{l=1}^{K} e^{w_l^T x}}, k=1,2\cdots,K$$

Logistic Regression Extension - Softmax (3)

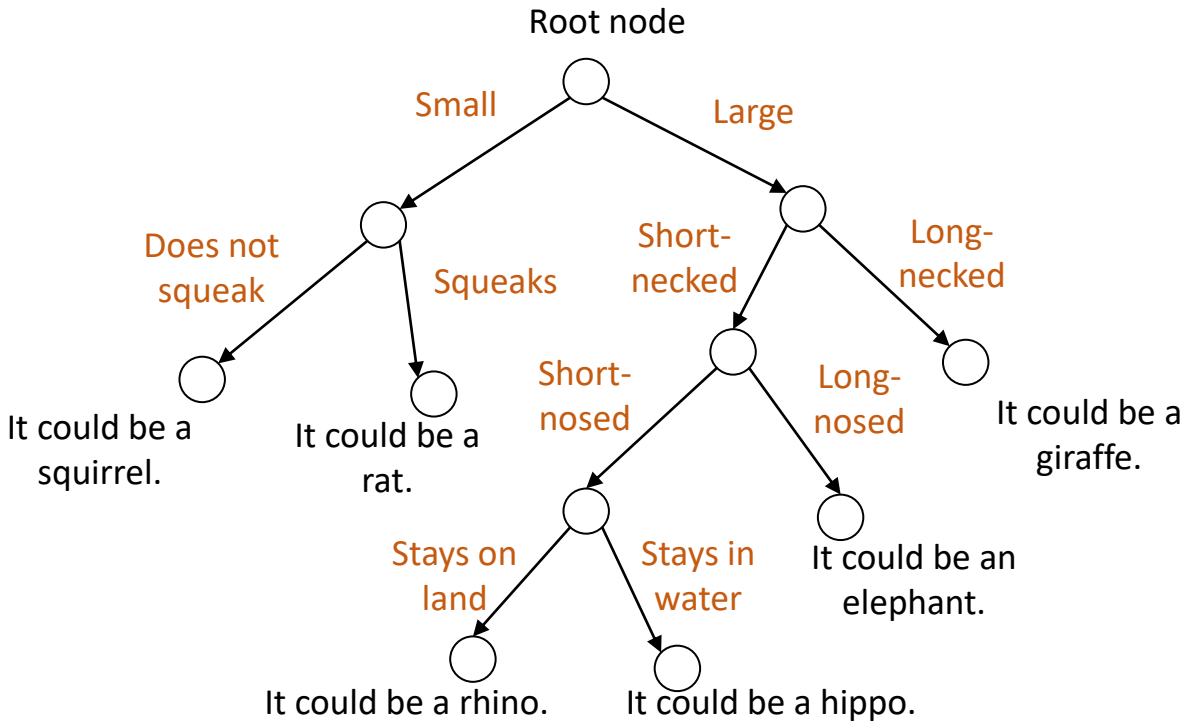
- Softmax assigns a probability value to each class in a multi-class classification problem. The sum of all the probabilities is 1.
  - Softmax may present the generated classes as follows:



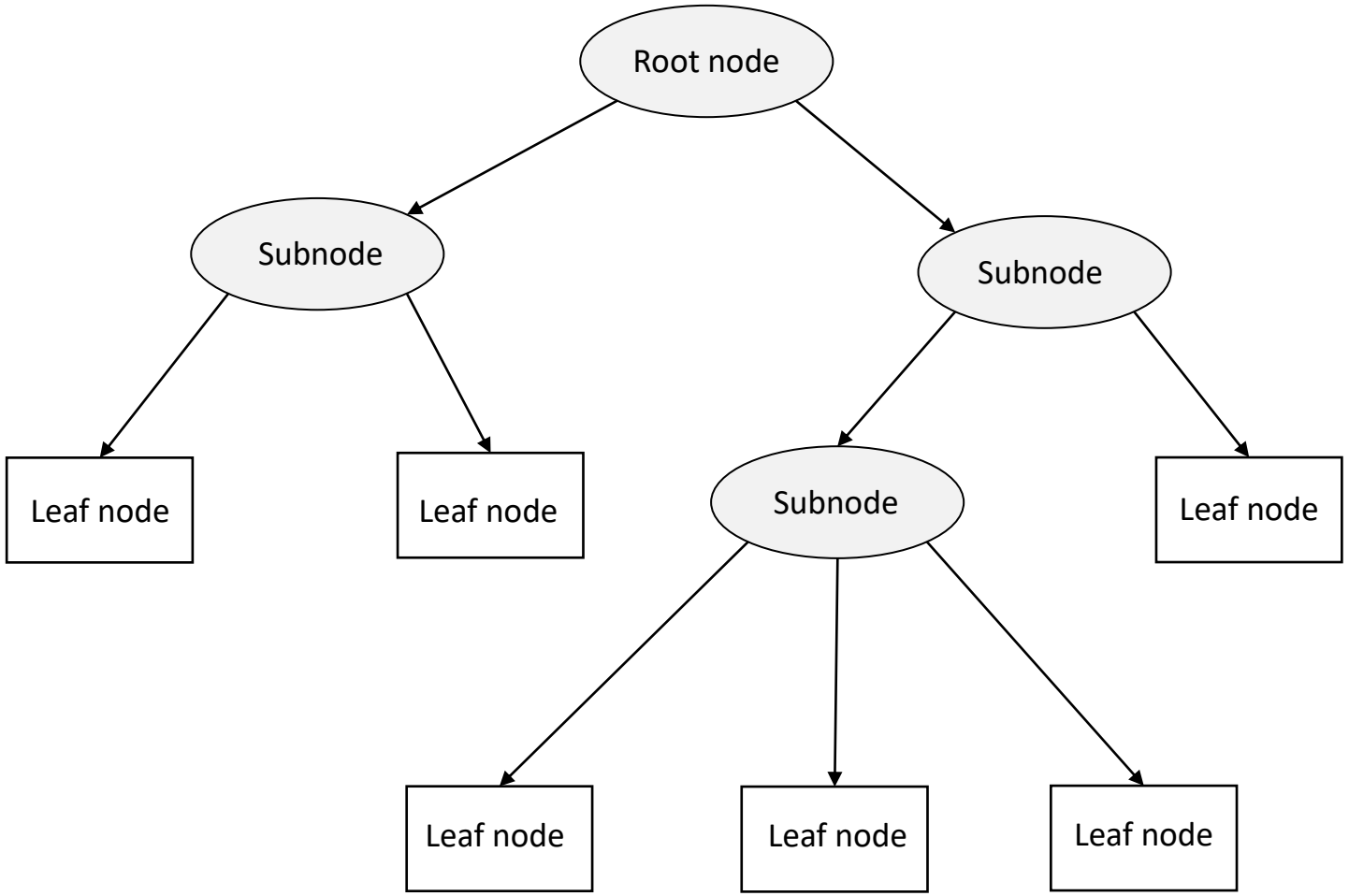
- **Sum of all probabilities:**
  - 0.09 + 0.22 + 0.68 + 0.01 =1
- The most possible object in the image: Apple

Decision Tree

- Each non-leaf node of the decision tree denotes a test on an attribute; each branch represents the output of a test; and each leaf (or terminal) node holds a class label. The algorithm starts at the root node (topmost node in the tree), tests the selected attributes on the intermediate (internal) nodes, and generates branches according to the output of the tests. Then, it saves the class labels on the leaf nodes as the decision results.



Structure of a Decision Tree



Key to Decision Tree Construction

- A decision tree requires feature attributes and an appropriate tree structure. The key step of constructing a decision tree is to divide data of all feature attributes, compare the result sets in terms of purity, and select the attribute with the highest purity as the data point for dataset division.
- Purity is measured mainly through the information entropy and GINI coefficient. The formula is as follows:

$$H(X)= -\sum_{k=1}^{K} p_k \log_2(p_k) \qquad Gini = 1 - \sum_{k=1}^{K} p_k^2$$

$$min_{j,s}[min_{c1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + min_{c2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

- p_k indicates the probability that a sample belongs to category k (in a total of *K* categories). A larger purity difference between the sample before and after division indicates a better decision tree.
- Common decision tree algorithms include ID3, C4.5, and CART.

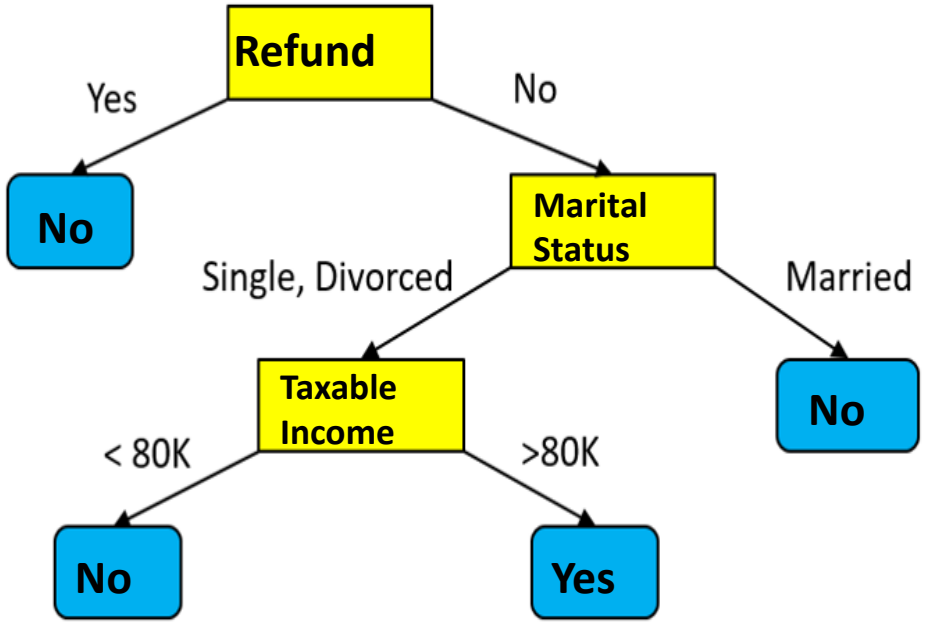
Decision Tree Construction Process

- **Feature selection**: Select one of the features of the training data as the split standard of the current node. (Different standards distinguish different decision tree algorithms.)
- **Decision tree generation**: Generate subnodes from top down based on the selected feature and stop until the dataset can no longer be split.
- **Pruning**: The decision tree may easily become overfitting unless necessary pruning (including pre-pruning and post-pruning) is performed to reduce the tree size and optimize its node structure.

Decision Tree Example

- The following figure shows a decision tree for a classification problem. The classification result is affected by three attributes: refund, marital status, and taxable income.

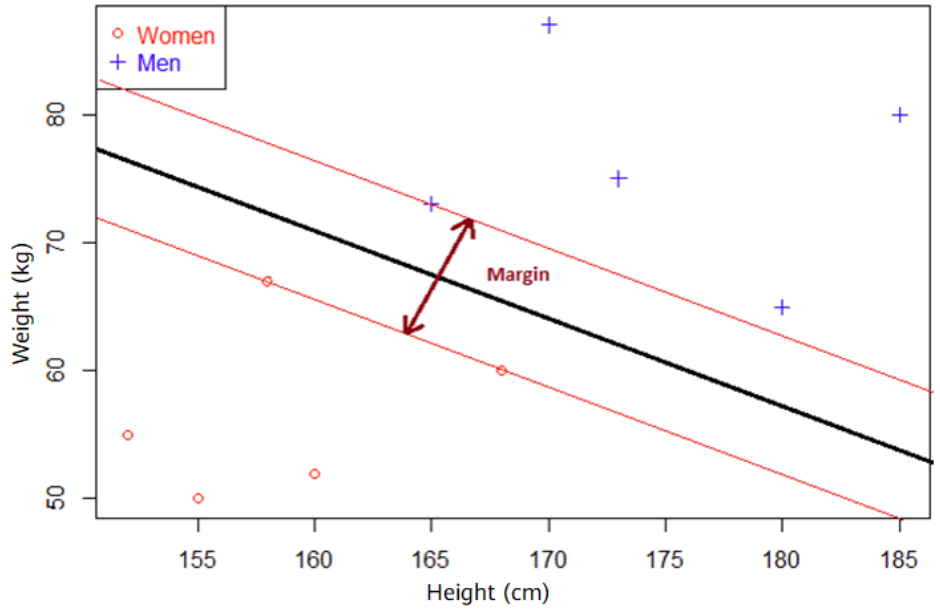
TID	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

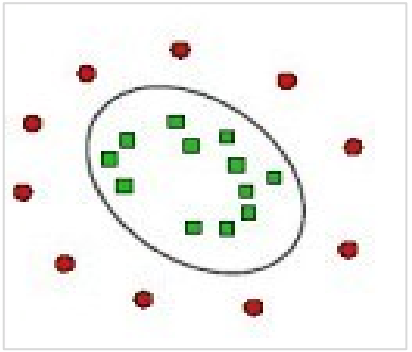


HUAWEI

Support Vector Machine

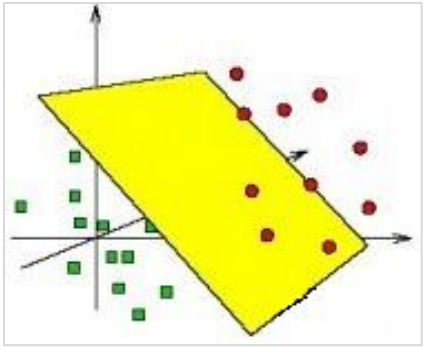
- Support vector machines (SVMs) are binary classification models. Their basic model is the linear classifier that maximizes the width of the gap between the two categories in the feature space. SVMs also have a kernel trick, which makes it a non-linear classifier. The learning algorithm of SVMs is the optimal algorithm for convex quadratic programming.





Difficult to split in a low-dimensional space.

Mapping

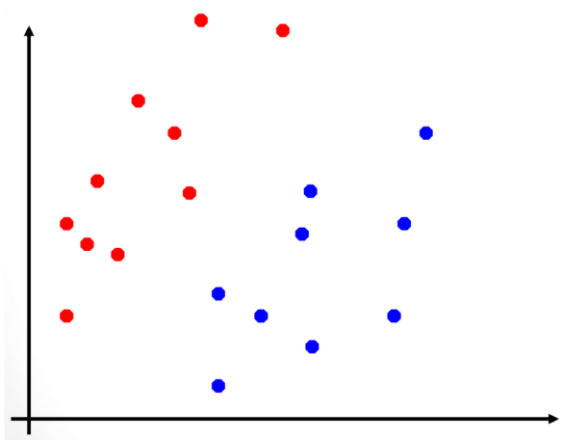


Easy to split in a high-dimensional space.

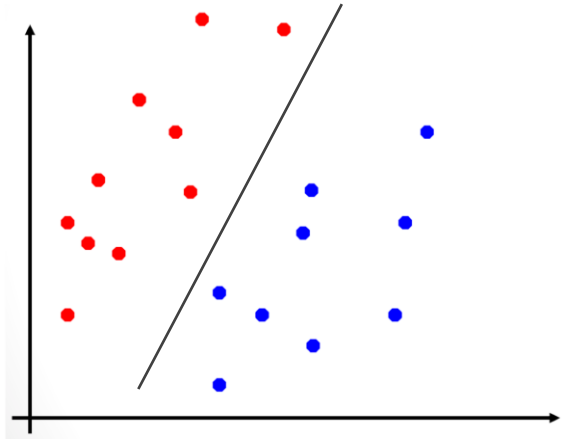
HUAWEI

Linear SVM (1)

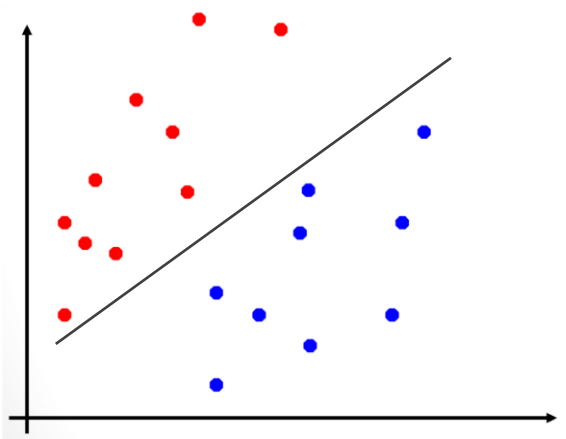
- How can we divide the red and blue data points with just one line?



Two-dimensional data set with two sample categories



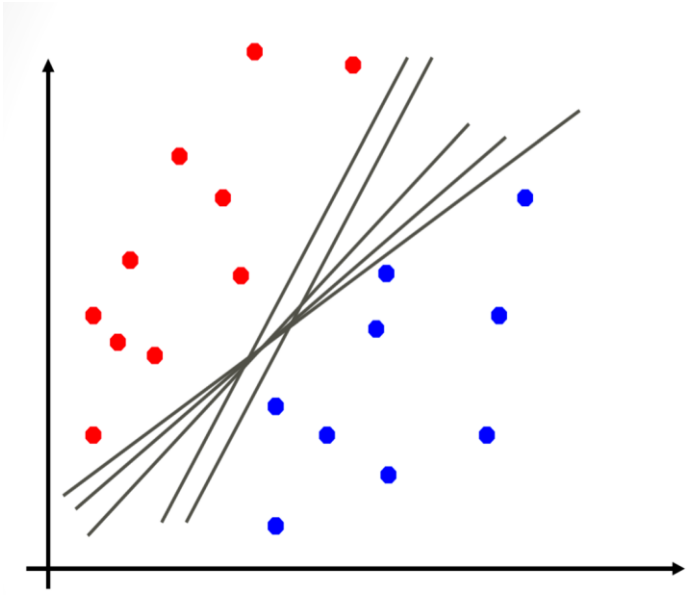
or

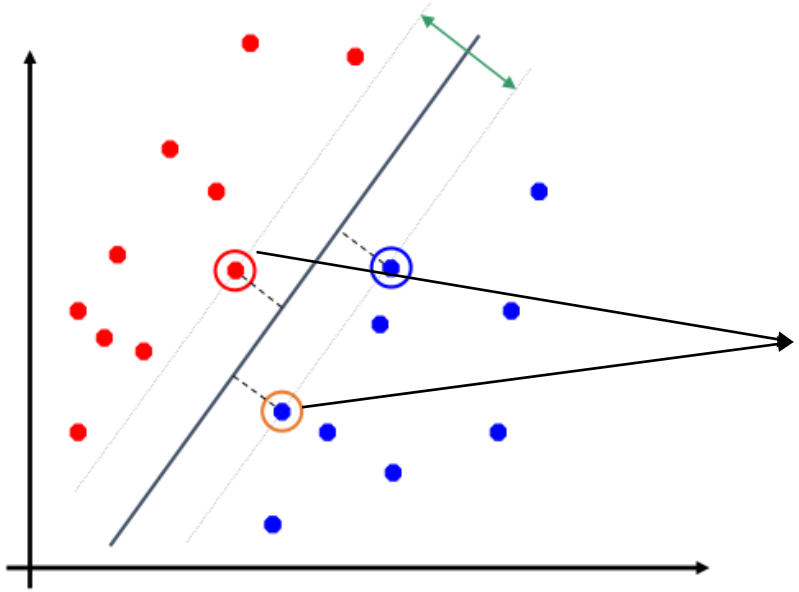


Both the division methods on the left and right can divide data. But which is correct?

Linear SVM (2)

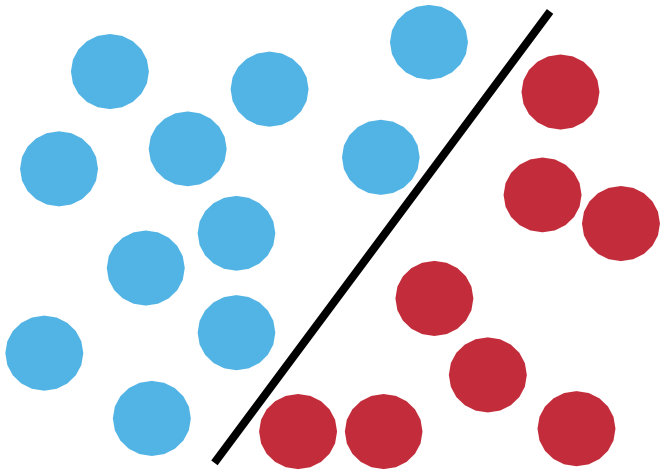
- We can use different straight lines to divide data into different categories. SVMs find a straight line and keep the most nearby points as **far** from the line as possible. This gives the model a strong generalization capability. These most nearby points are called **support vectors**.
- In the two-dimensional space, a straight line is used for division; in the high-dimensional space, a **hyperplane** is used for division.



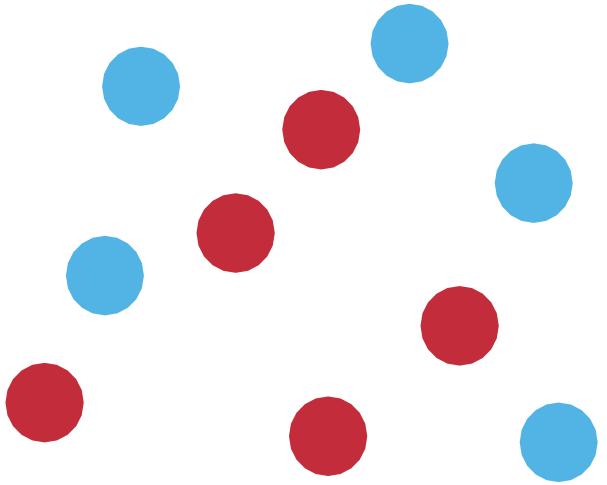


Non-linear SVM (1)

- How can we divide a linear inseparable data set?



Linear SVM works well on a linear separable data set.

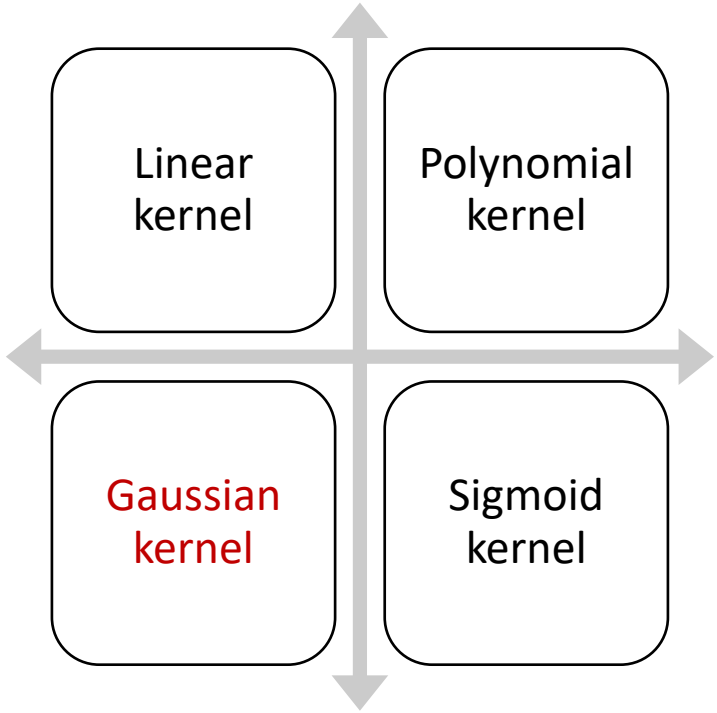


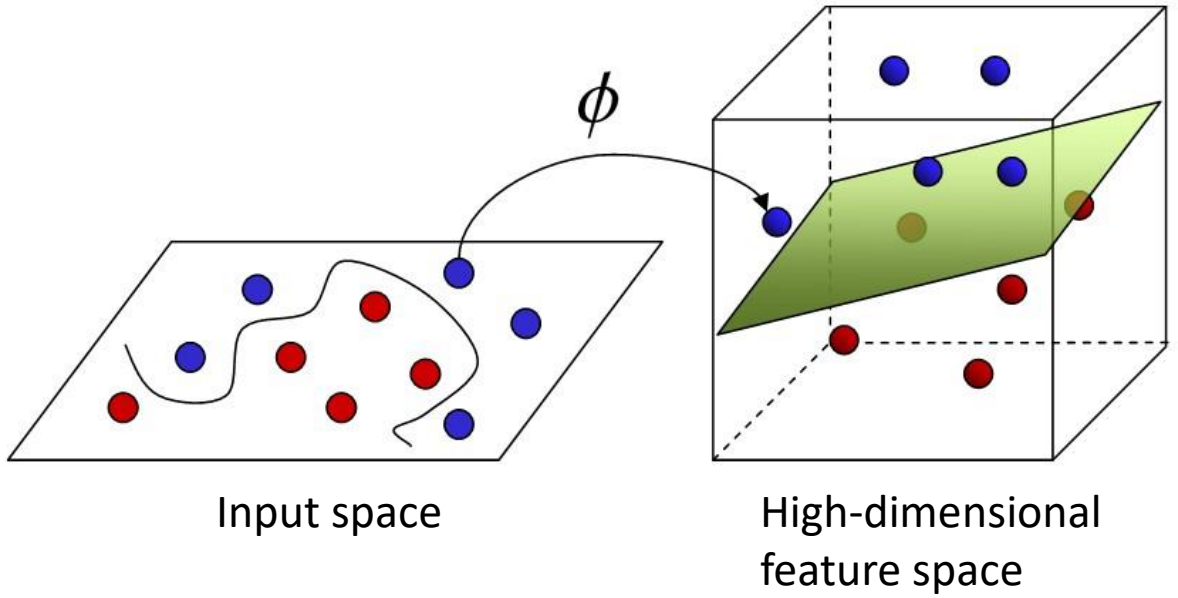
A non-linear data set cannot be divided using a straight line.

Non-linear SVM (2)

- Kernel functions can be used to create non-linear SVMs.
- Kernel functions allow algorithms to fit a maximum-margin hyperplane in a transformed high-dimensional feature space.

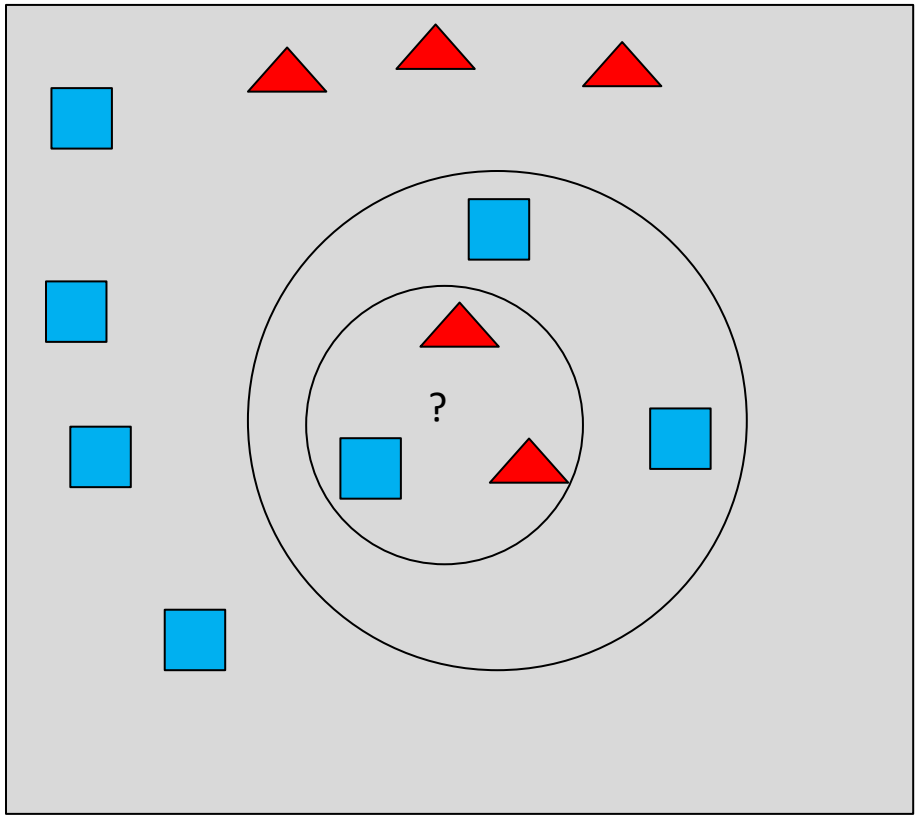
Common kernel functions





k-Nearest Neighbors Algorithm (1)

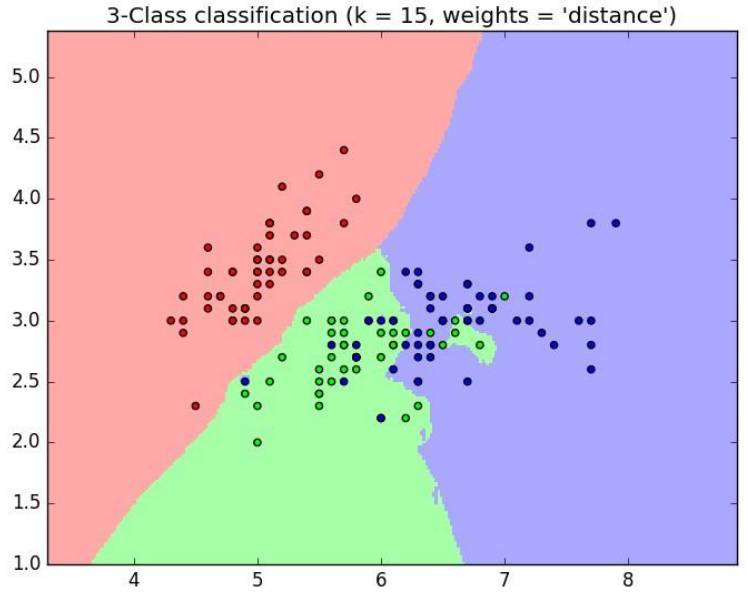
- The *k*-nearest neighbor (*k*-NN) classification algorithm is a theoretically mature method and one of the simplest machine learning algorithms. The idea of *k*-NN classification is that, if most of *k* closest samples (nearest neighbors) of a sample in the feature space belong to a category, the sample also belongs to this category.



The category of point **?** varies according to how many neighbor nodes are chosen.

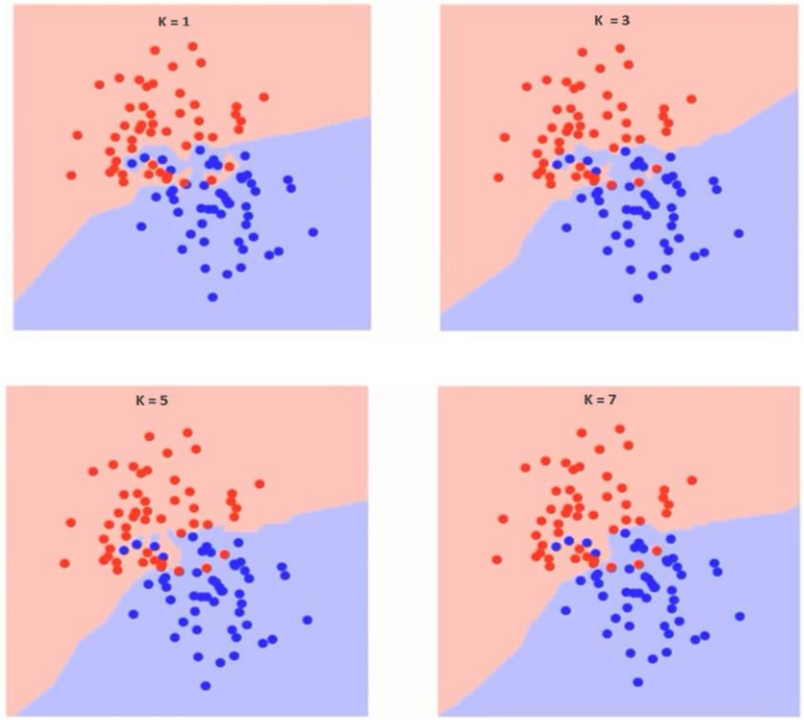
k-Nearest Neighbors Algorithm (2)

- The logic of *k*-NN is simple: If an object's *k* nearest neighbors belong to a class, so does the object.
- *k*-NN is a non-parametric method and is often used for datasets with irregular decision boundaries.
  - *k*-NN typically uses the **majority voting** method to predict classification, and uses the **mean value** method to predict regression.
- *k*-NN requires a very large amount of computing.



k-Nearest Neighbors Algorithm (3)

- Typically, a larger *k* value reduces the impact of noise on classification, but makes the boundary between classes less obvious.
  - A large *k* value indicates a higher probability of underfitting because the division is too rough; while a small *k* value indicates a higher probability of overfitting because the division is too refined.



- As seen from the figure, the boundary becomes smoother as the *k* value increases.
- As the *k* value increases, the points will eventually become all blue or all red.

Naive Bayes (1)

- **Naive Bayes** classifiers are a family of simple "probabilistic classifiers" based on **Bayes' theorem** with **strong independence assumptions between the features**. For a given sample feature X, the probability that the sample belongs to category H is:

$$P(C_k \mid X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n \mid C_k)P(C_k)}{P(X_1,\ldots,X_n)}$$

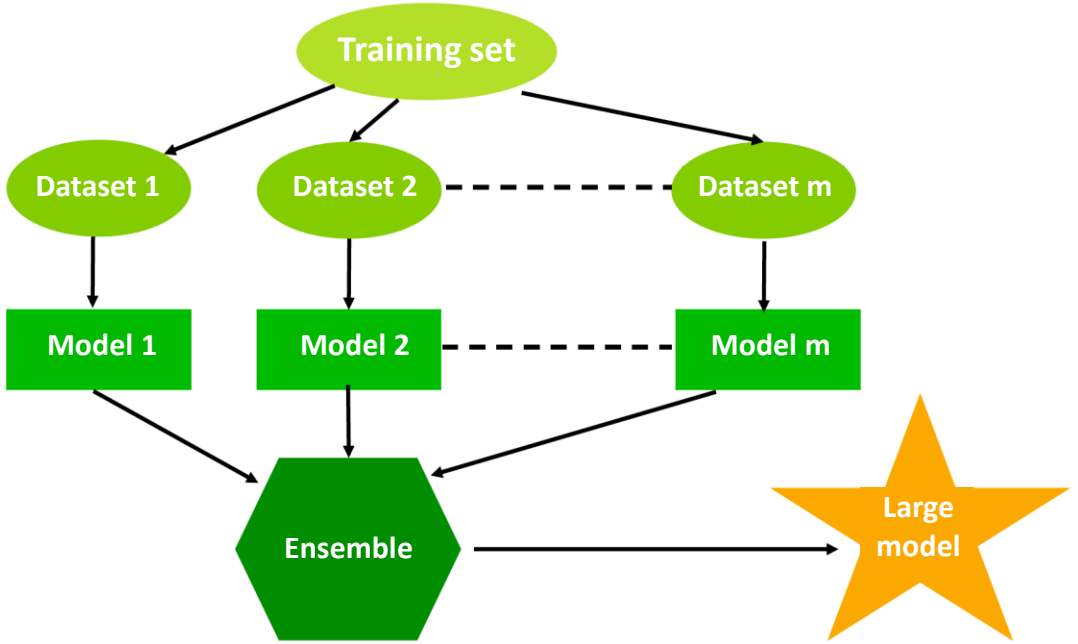
  - $X_1, X_2, \ldots, X_n$ are data features, which are usually described by *m* measurement values of the attribute set.
    - For example, the attribute of the color feature may be red, yellow, and blue.
  - C_k indicates that the data belongs to a specific class C.
  - $P(C_k|X_1, X_2, \ldots, X_n)$ is the posterior probability, or the posterior probability of H under condition C_k.
  - $\mathrm{P}(C_k)$ is the prior probability independent of $X_1, X_2, \ldots, X_n$.
  - $P(X_1, X_2, \ldots, X_n)$ is the prior probability of X.

Naive Bayes (2)

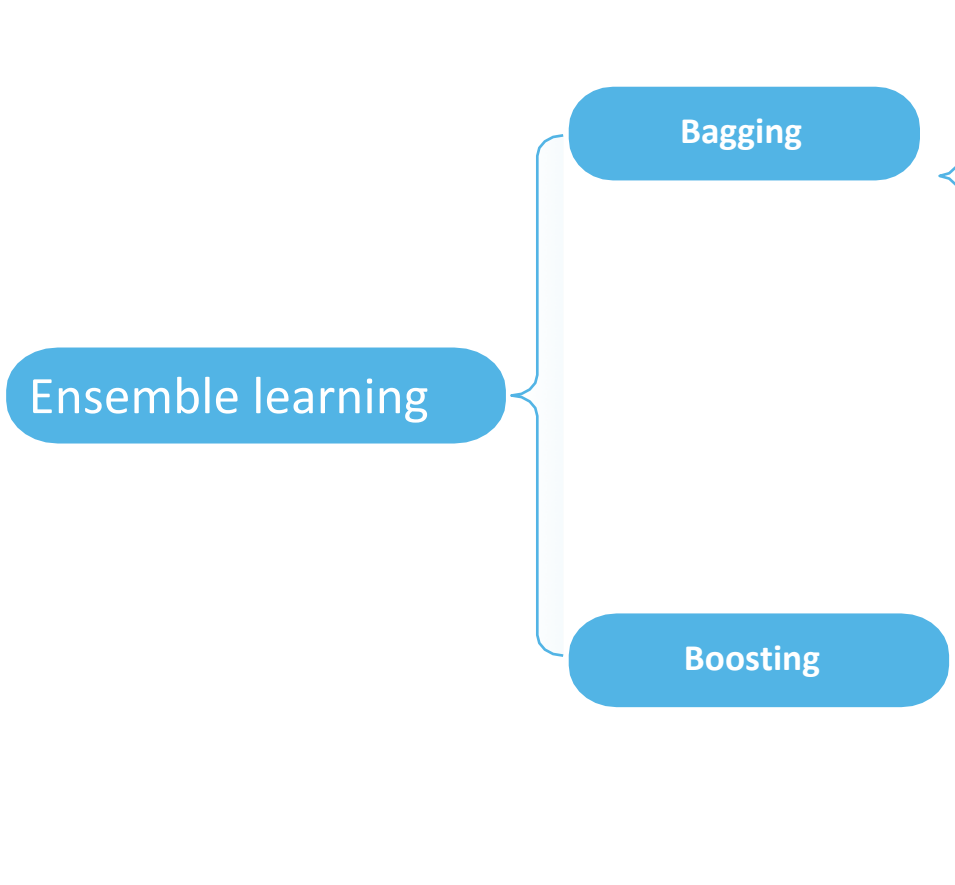
- Feature independent hypothesis example:
  - If a fruit is red, round, and about 10 cm in diameter, it can be considered an apple.
  - A Naive Bayes classifier believes that each of these features independently contributes to the probability of the fruit being an apple, regardless of any possible correlation between color, roundness, and diameter features.

Ensemble Learning

- Ensemble learning is a machine learning paradigm in which multiple learners are trained and combined to resolve an issue. When multiple learners are used, the generalization capability of the ensemble can be much stronger than that of a single learner.
- For example, If you ask thousands of people at random a complex question and then summarize their answers, the summarized answer is more accurate than an expert's answer in most cases. This is the **wisdom of the crowd**.



Types of Ensemble Learning



Example: random forest

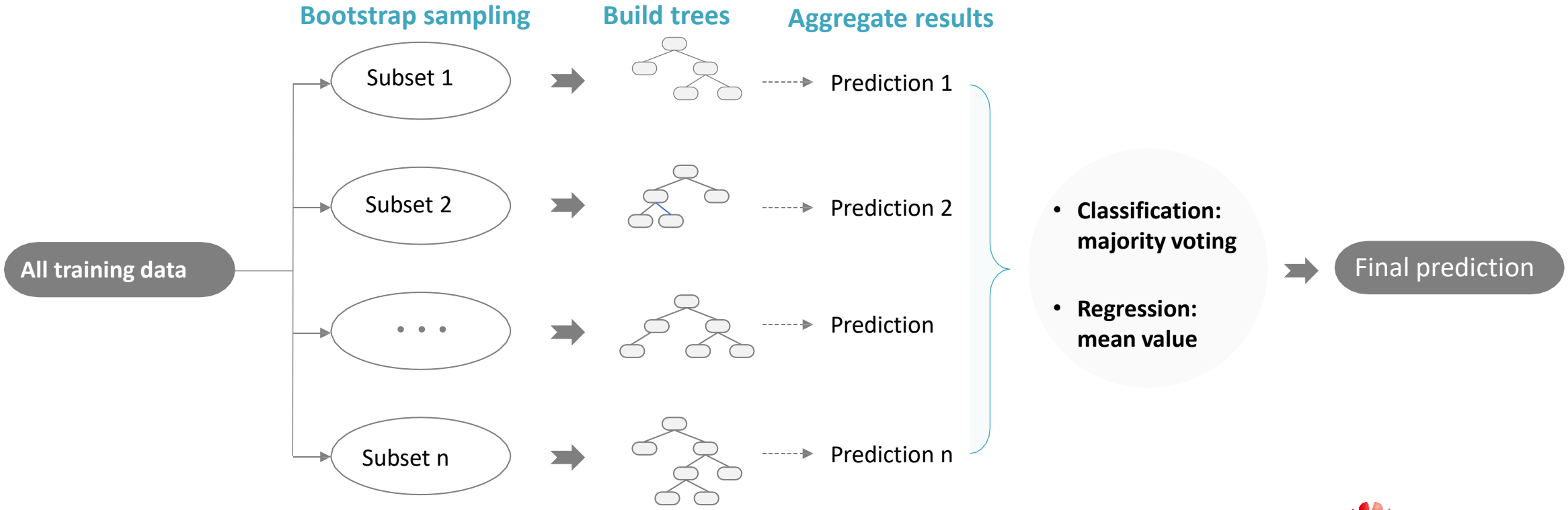
- Bagging independently builds multiple basic learners and then averages their predictions.
- On average, an ensemble learner is usually better than a single-base learner because of a smaller variance.

Example: AdaBoost, GBDT, XGBoost

Bagging builds basic learners in sequence and gradually reduces the biases of the ensemble learner. An ensemble learner has a strong fitting capability but may overfit.

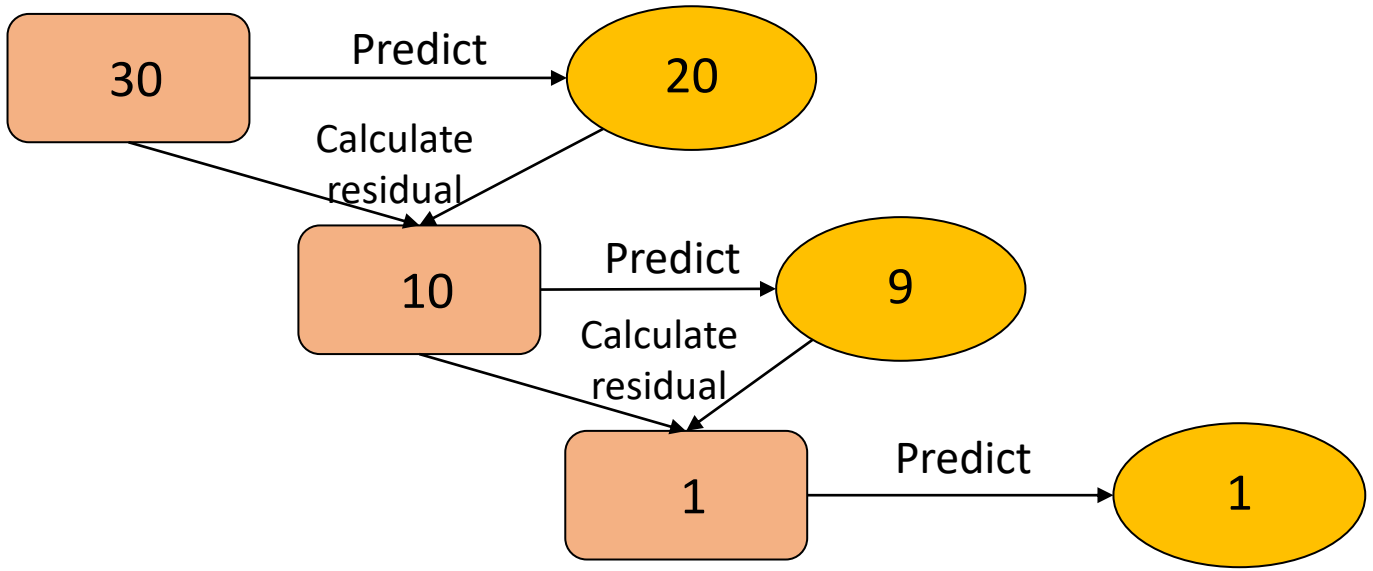
Ensemble Learning - Random Forest

- Random forest = Bagging + Classification and regression tree (CART)
- Random forest builds multiple decision trees and aggregates their results to make prediction more accurate and stable.
  - The random forest algorithm can be used for classification and regression problems.



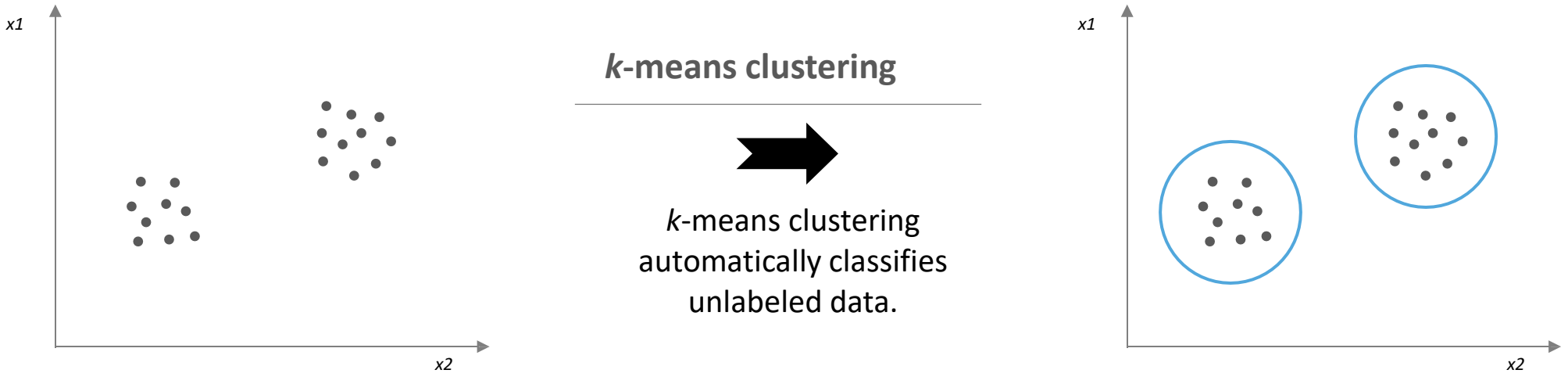
Ensemble learning - Gradient Boosted Decision Tree

- Gradient boosted decision tree (GBDT) is a type of boosting algorithm.
- The prediction result of the ensemble model is the sum of results of all base learners. The essence of GBDT is that the next base learner tries to fit the residual of the error function to the prediction value, that is, the residual is the error between the prediction value and the actual value.
- During GBDT model training, the loss function value of the sample predicted by the model must be as small as possible.



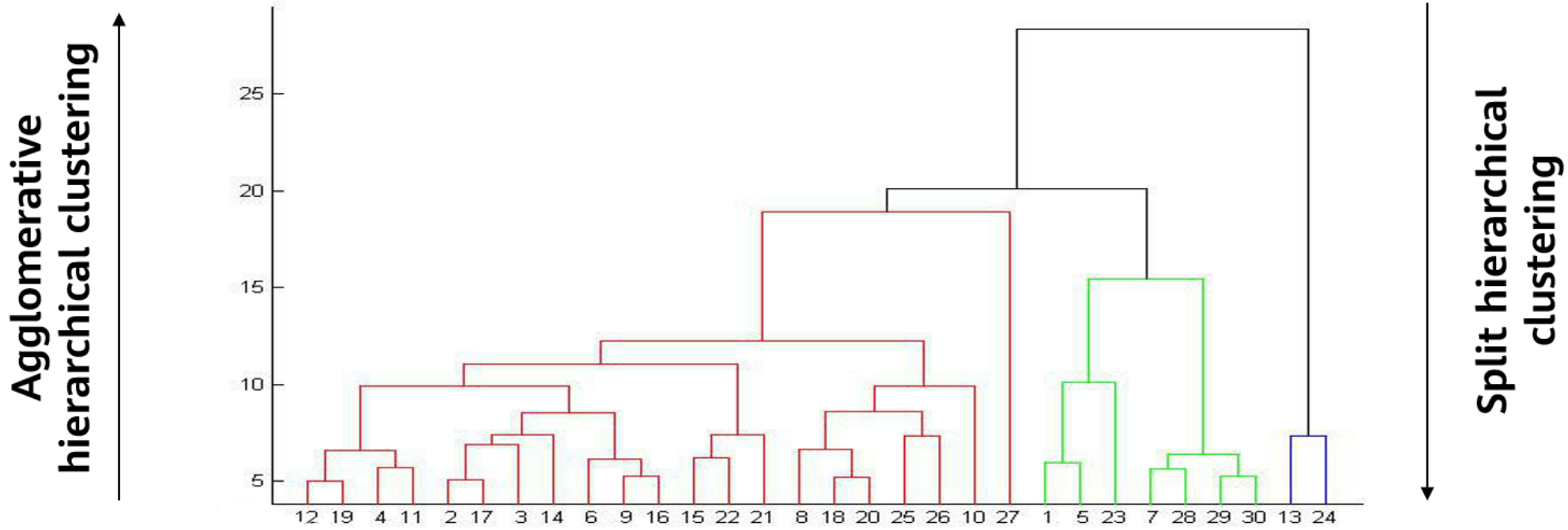
Unsupervised Learning - *k*-Means Clustering

- *k*-means clustering takes the number of clusters *k* and a dataset of *n* objects as inputs, and outputs *k* clusters with minimized within-cluster variances.
- In the *k*-means algorithm, the number of clusters is *k*, and *n* data objects are split into *k* clusters. The obtained clusters meet the following requirements: high similarity between objects in the same cluster, and low similarity between objects in different clusters.



Unsupervised Learning - Hierarchical Clustering

- Hierarchical clustering divides a dataset at different layers and forms a tree-like clustering structure. The dataset division may use a "bottom-up" aggregation policy, or a "top-down" splitting policy. The hierarchy of clustering is represented in a tree diagram. The root is the only cluster of all samples, and the leaves are clusters of single samples.



Summary

- This course first describes the definition and types of machine learning, as well as problems machine learning solves. Then, it introduces key knowledge points of machine learning, including the overall procedure (data preparation, data cleansing, feature selection, model evaluation, and model deployment), common algorithms (including linear regression, logistic regression, decision tree, SVM, Naive Bayes, *k*-NN, ensemble learning, and k-means clustering), and hyperparameters.

Quiz

1. (Single-answer) Which of the following is not a supervised learning algorithm? ( )

   A. Linear regression

   B. Decision tree

   C. *k*-NN

   D. *k*-means clustering

2. (True or false) Gradient descent is the only method of machine learning. ( )

Recommendations

- Huawei Talent
  - https://e.huawei.com/en/talent/portal/#/
- Huawei knowledge base
  - https://support.huawei.com/enterprise/en/knowledge?lang=en

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.