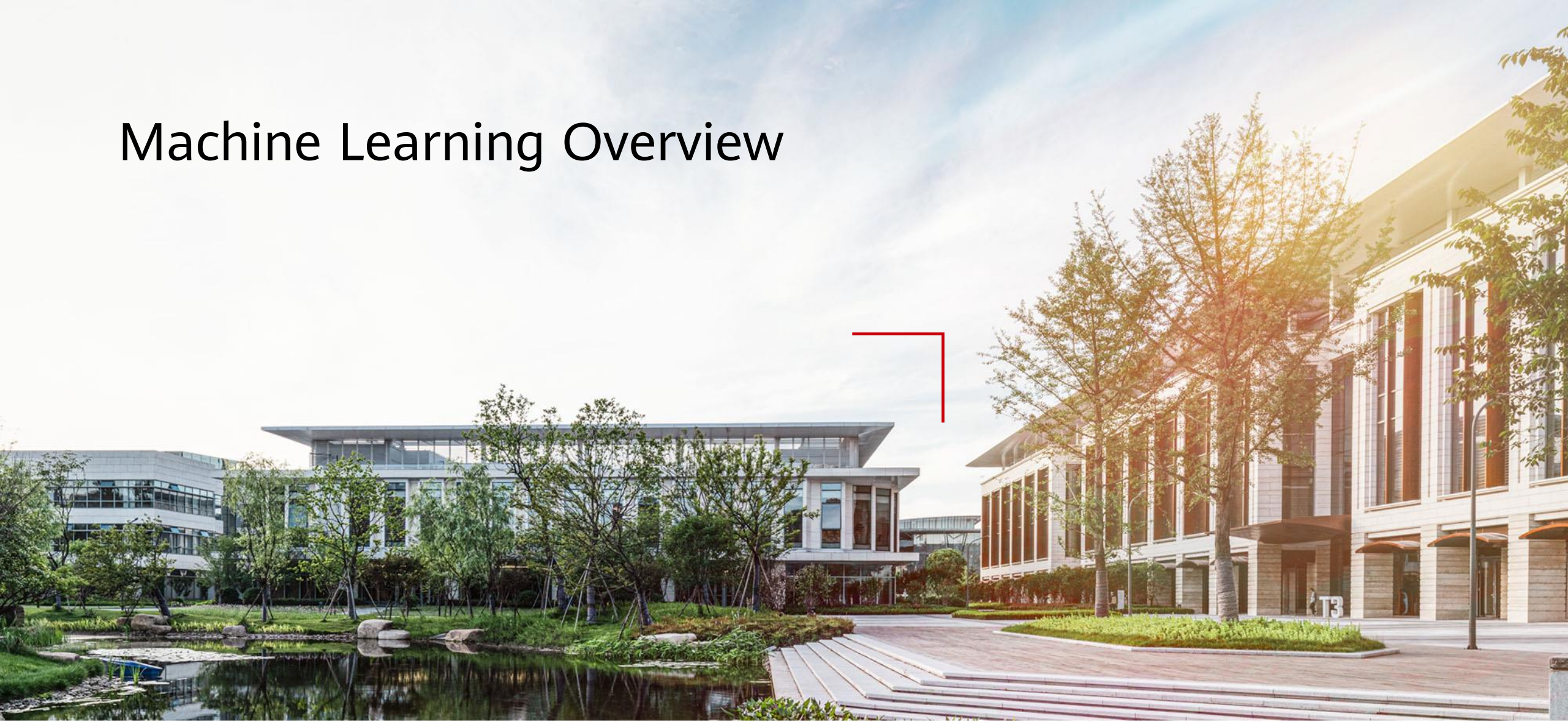
Machine Learning Overview



HUAWEI

Foreword

- Machine learning is a core research field of AI, and it is also a necessary knowledge for deep learning. Therefore, this chapter mainly introduces the main concepts of machine learning, the classification of machine learning, the overall process of machine learning, and the common algorithms of machine learning.

Objectives

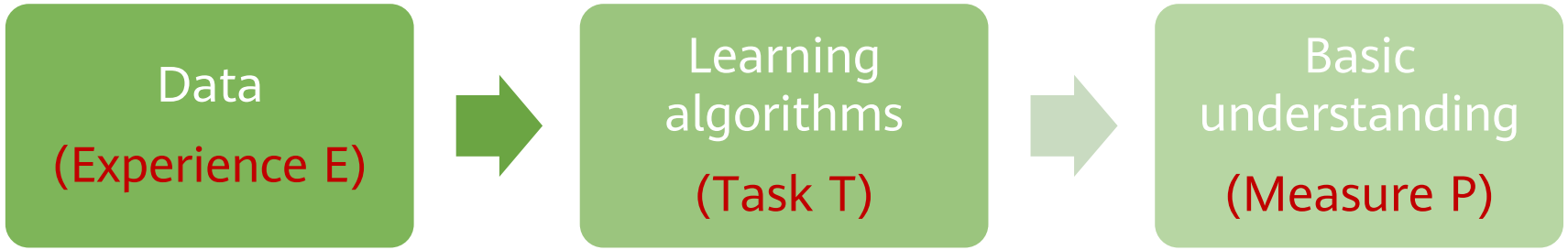
Upon completion of this course, you will be able to:

- Master the learning algorithm definition and machine learning process.
- Know common machine learning algorithms.
- Understand concepts such as hyperparameters, gradient descent, and cross validation.

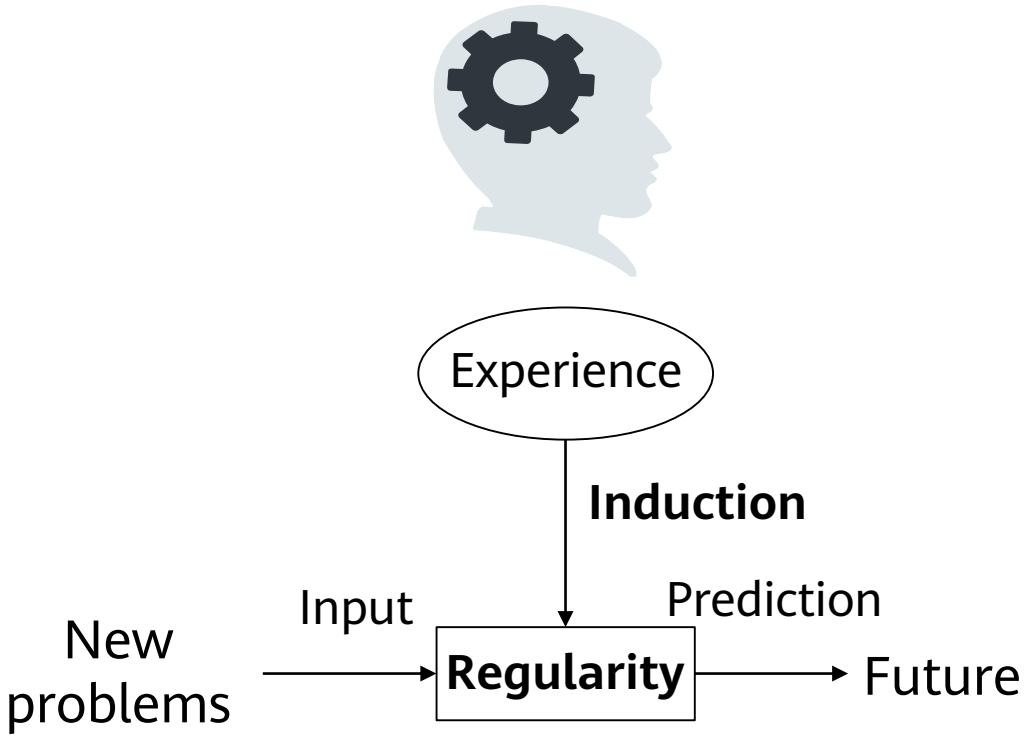
Contents

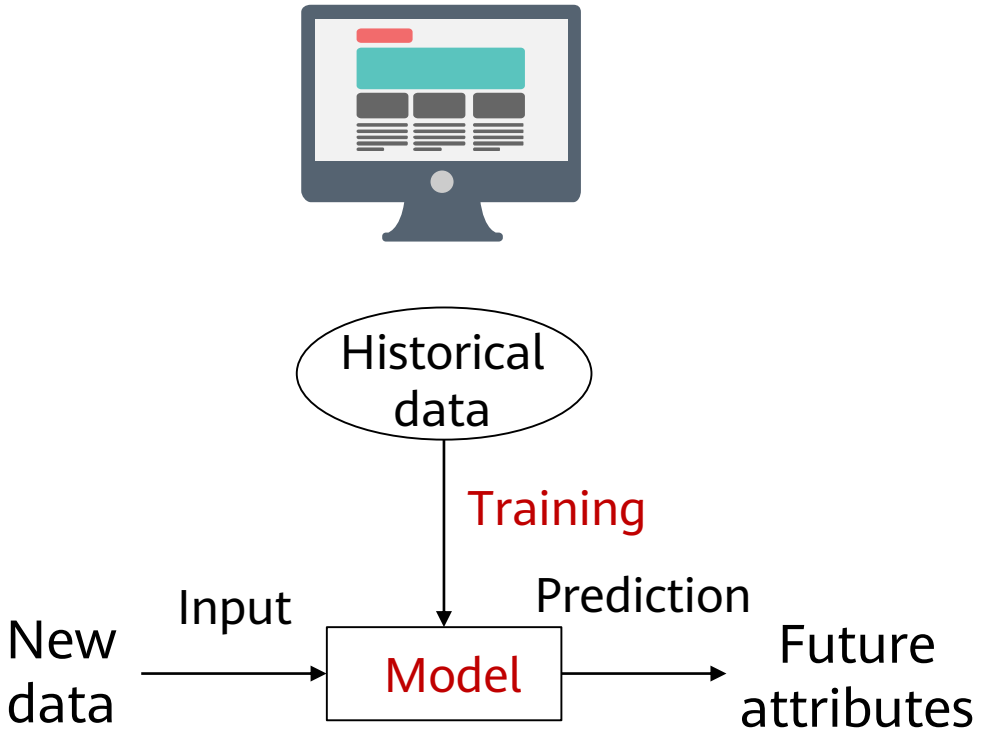
Machine Learning Algorithms (1)

- Machine learning (including deep learning) is a study of learning 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.

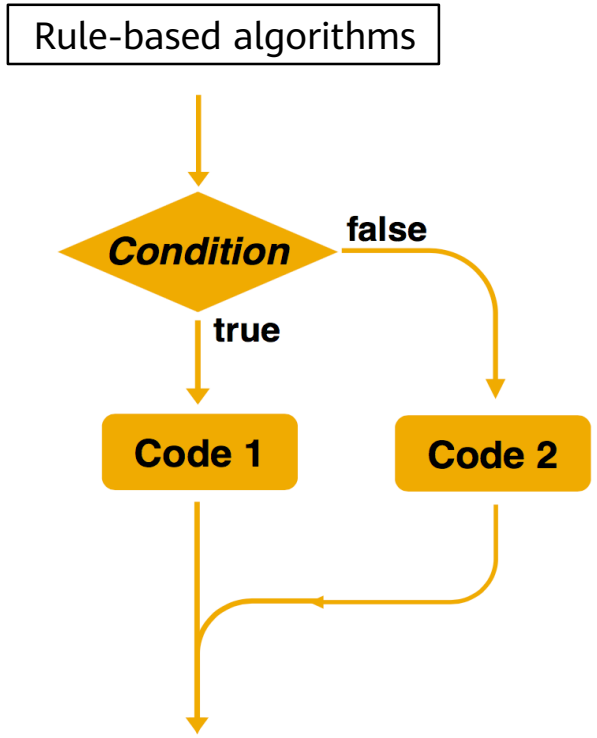


Machine Learning Algorithms (2)

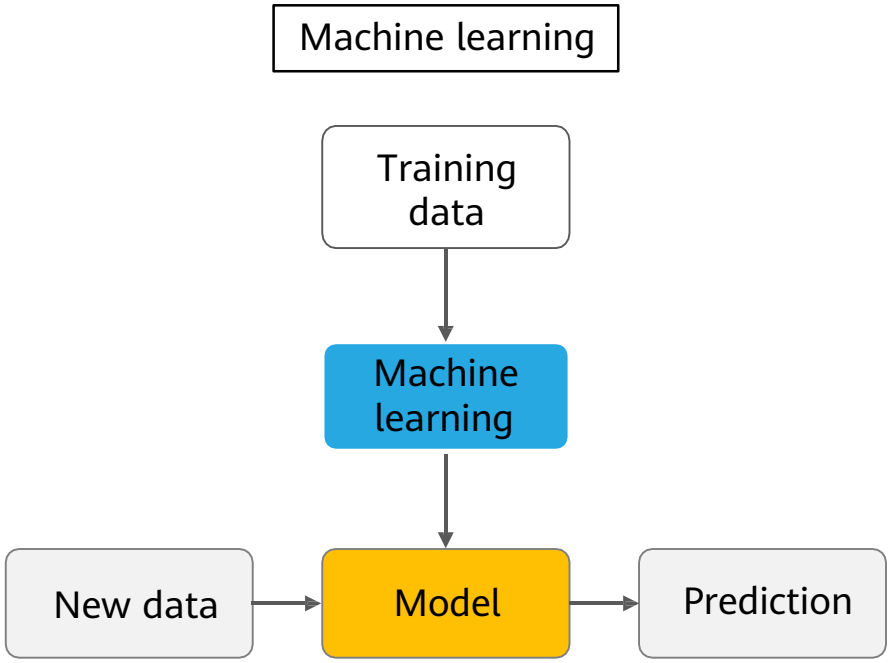




Differences Between Machine Learning Algorithms and Traditional Rule-Based Algorithms



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

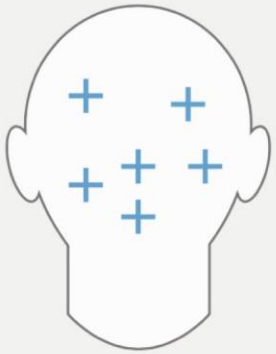


- Samples are used for training.
- The decision-making rules are complex or difficult to describe.
- Rules are automatically learned by machines.

Application Scenarios of Machine Learning (1)

- The solution to a problem is complex, or the problem may involve a large amount of data without a clear data distribution function.
- Machine learning can be used in the following scenarios:

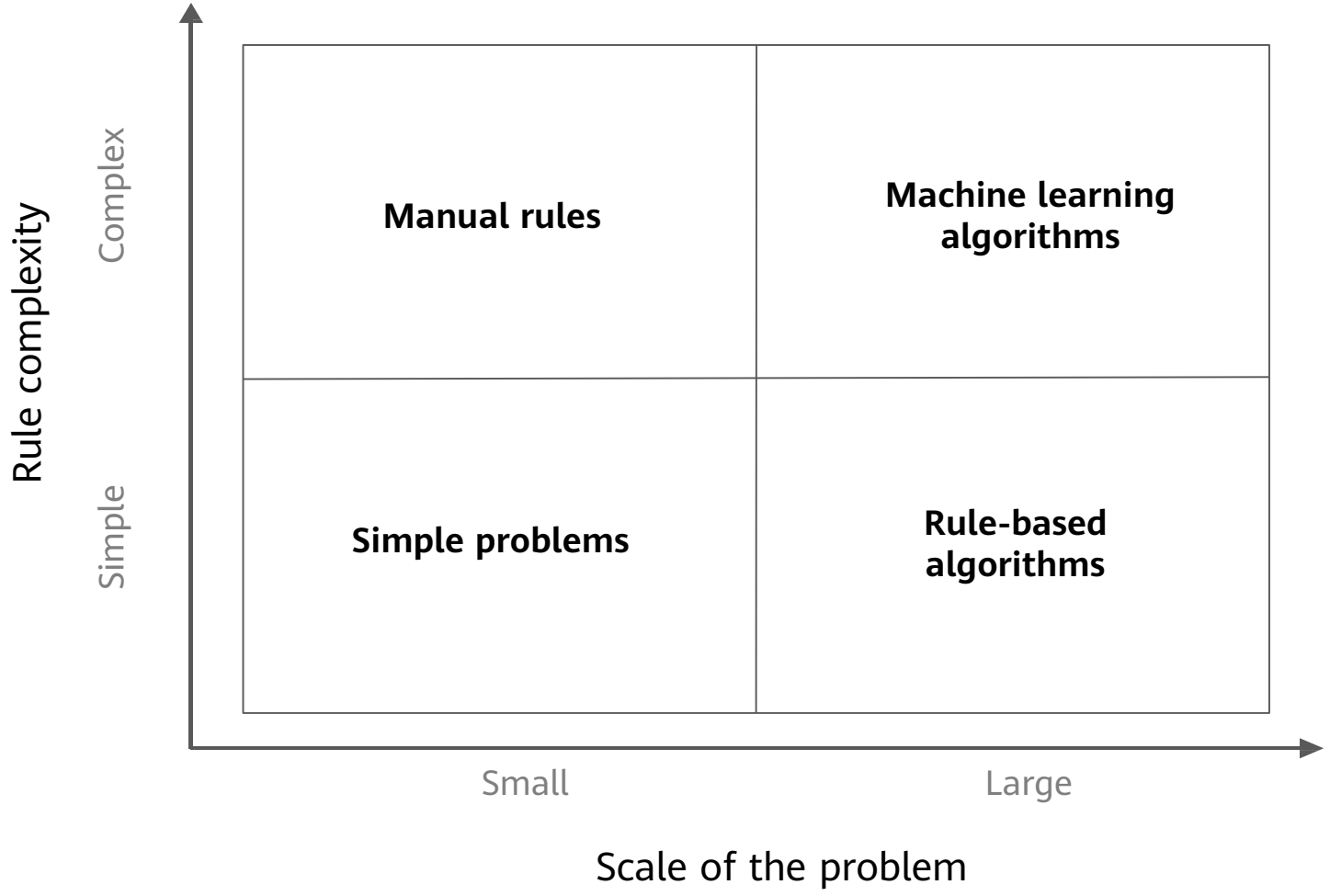
Rules are complex or cannot be described, such as facial recognition and voice recognition.



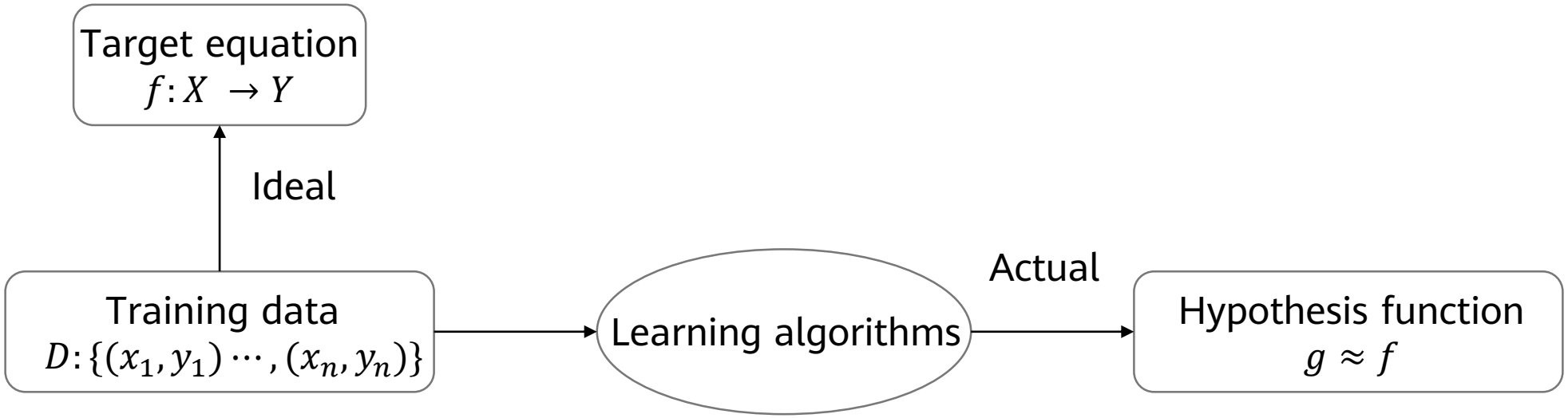
Task rules change over time. For example, in the part-of-speech tagging task, new words or meanings are generated at any time.

Data distribution changes over time, requiring constant readaptation of programs, such as predicting the trend of commodity sales.

Application Scenarios of Machine Learning (2)



Rational Understanding of Machine Learning Algorithms



- Target function *f* is unknown. Learning algorithms cannot obtain a perfect function *f*.
- Assume that hypothesis function *g* **approximates** function *f*, but may be different from function *f*.

Main Problems Solved by Machine Learning

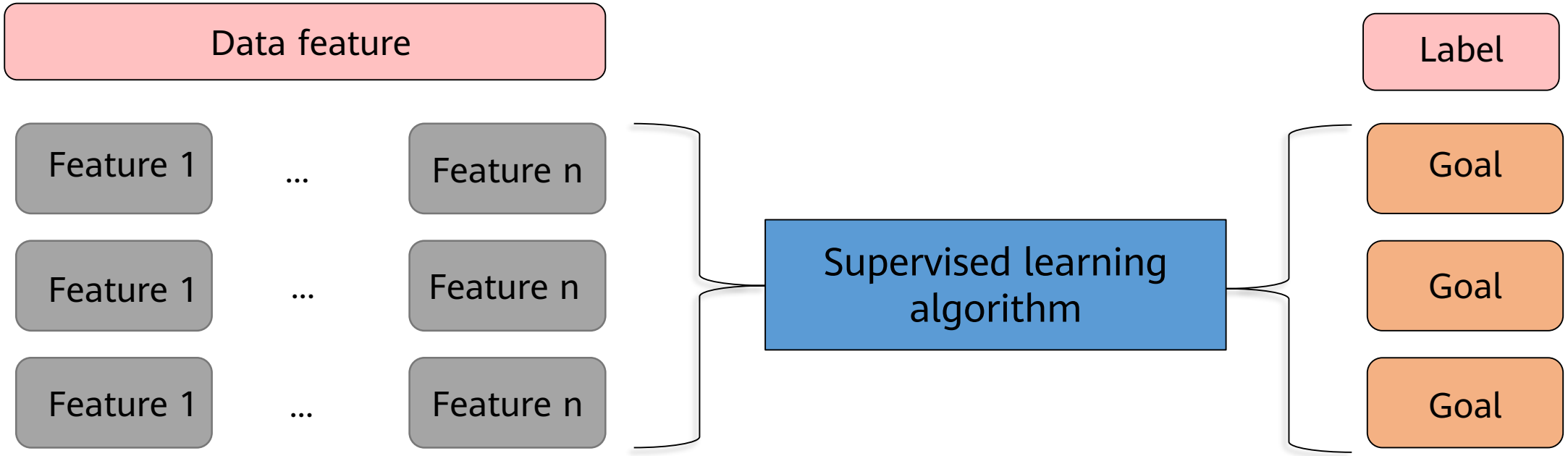
- Machine learning can deal with many types of tasks. The following describes the most typical and common types of tasks.
  - Classification: A computer program needs to specify which of the k categories some input belongs to. To accomplish this task, learning algorithms usually output a function $f: R^n \rightarrow (1,2,\ldots,k)$. For example, the image classification algorithm in computer vision is developed to handle classification tasks.
  - Regression: For this type of task, a computer program predicts the output for the given input. Learning algorithms typically output a function $f: R^n \rightarrow R$. An example of this task type is to predict the claim amount of an insured person (to set the insurance premium) or predict the security price.
  - Clustering: A large amount of data from an unlabeled dataset is divided into multiple categories according to internal similarity of the data. Data in the same category is more similar than that in different categories. This feature can be used in scenarios such as image retrieval and user profile management.
- Classification and regression are two main types of prediction, accounting from 80% to 90%. The output of classification is discrete category values, and the output of regression is continuous numbers.

Contents

Machine Learning Classification

- **Supervised learning**: Obtain an optimal model with required performance through training and learning based on the samples of known categories. Then, use the model to map all inputs to outputs and check the output for the purpose of classifying unknown data.
- **Unsupervised learning**: For unlabeled samples, the learning algorithms directly model the input datasets. Clustering is a common form of unsupervised learning. We only need to put highly similar samples together, calculate the similarity between new samples and existing ones, and classify them by similarity.
- **Semi-supervised learning**: In one task, a machine learning model that automatically uses a large amount of unlabeled data to assist learning directly of a small amount of labeled data.
- **Reinforcement learning**: It is an area of machine learning concerned with how agents ought to take actions in an environment to maximize some notion of cumulative reward. The difference between reinforcement learning and supervised learning is the teacher signal. The reinforcement signal provided by the environment in reinforcement learning is used to evaluate the action (scalar signal) rather than telling the learning system how to perform correct actions.

Supervised Learning

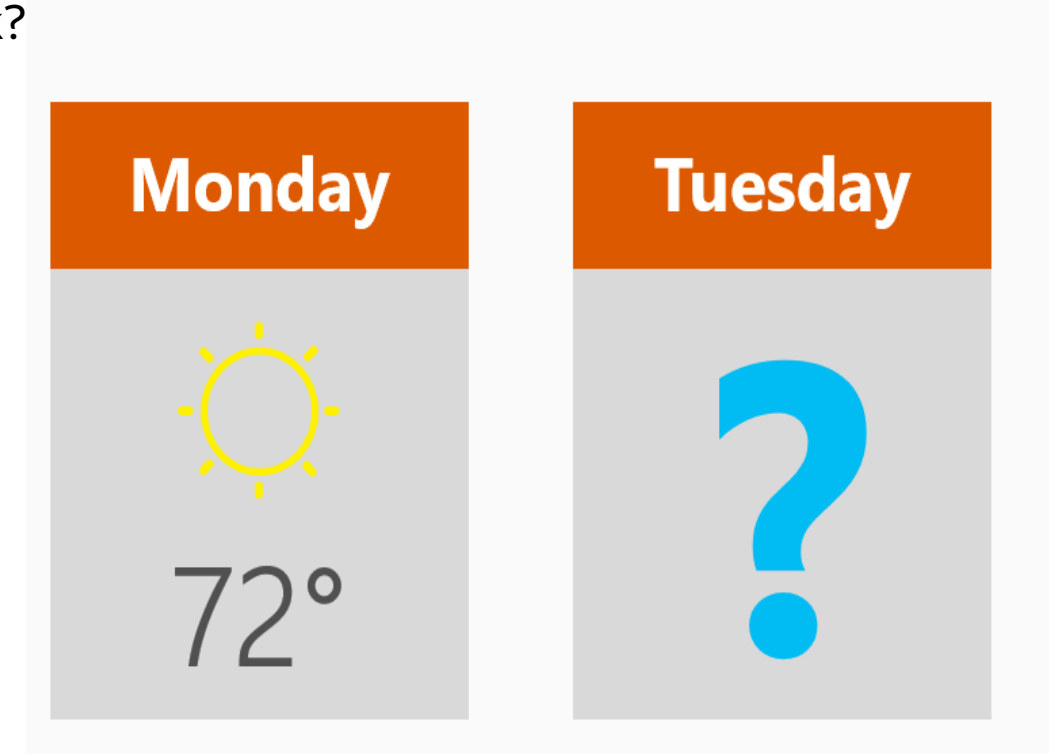


Weather	Temperature	Wind Speed
Sunny	Warm	Strong
Rainy	Cold	Fair
Sunny	Cold	Weak

Enjoy Sports
Yes
No
Yes

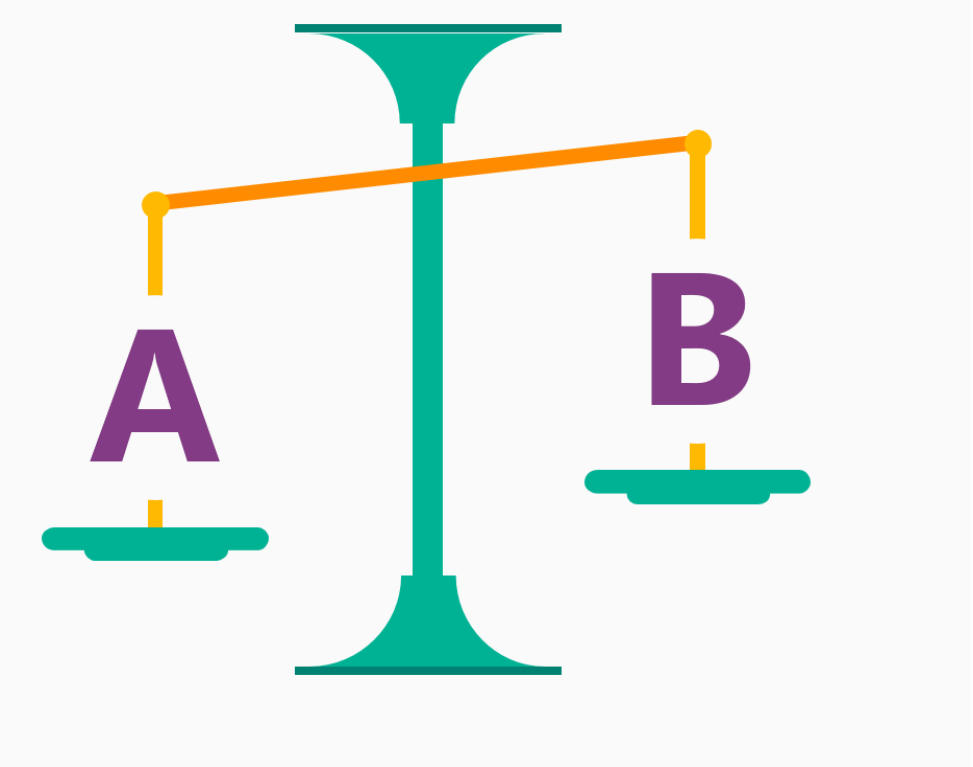
Supervised Learning - Regression Questions

- Regression: reflects the features of attribute values of samples in a sample dataset. The dependency between attribute values is discovered by expressing the relationship of sample mapping through functions.
  - How much will I benefit from the stock next week?
  - What's the temperature on Tuesday?

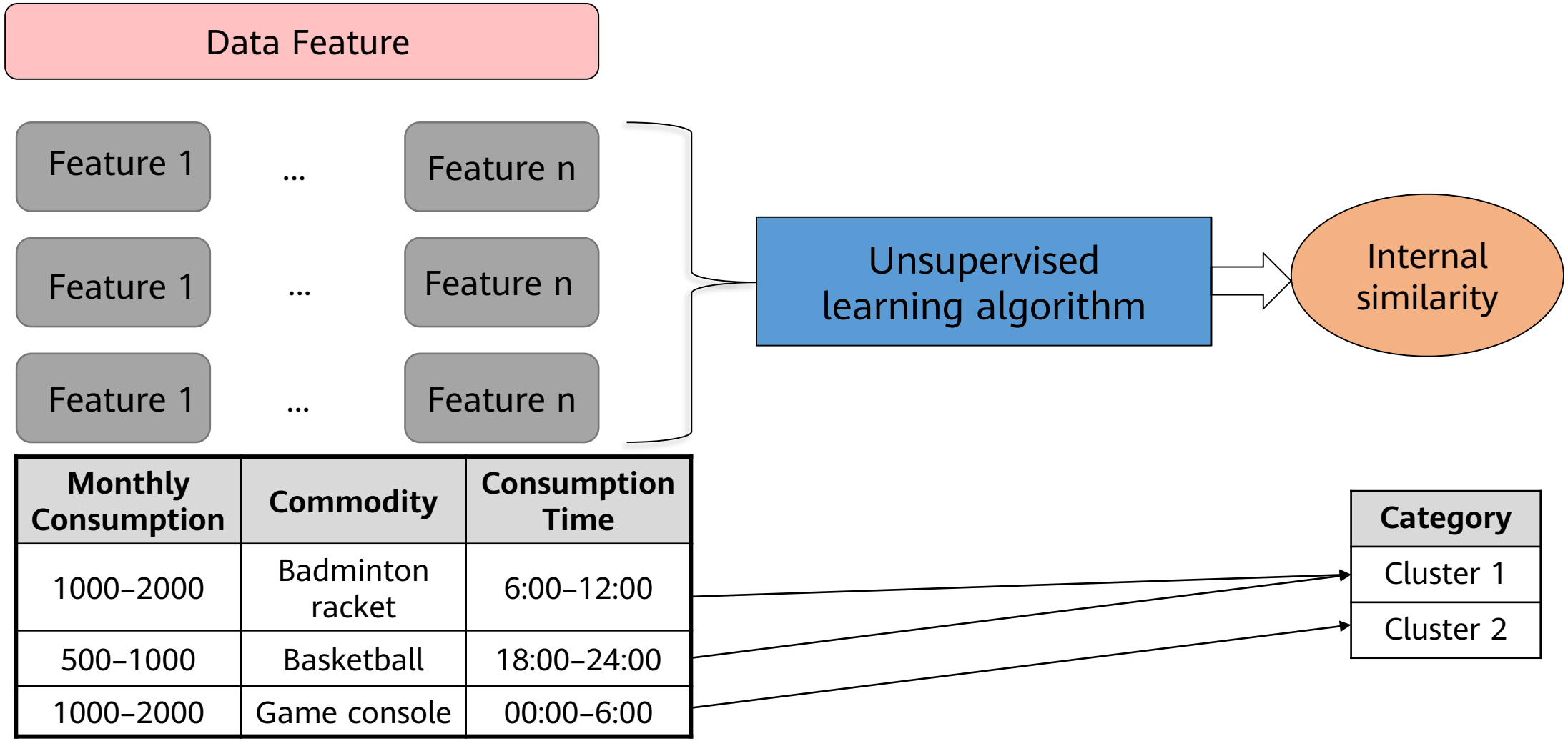


Supervised Learning - Classification Questions

- Classification: maps samples in a sample dataset to a specified category by using a classification model.
  - Will there be a traffic jam on XX road during the morning rush hour tomorrow?
  - Which method is more attractive to customers: 5 yuan voucher or 25% off?

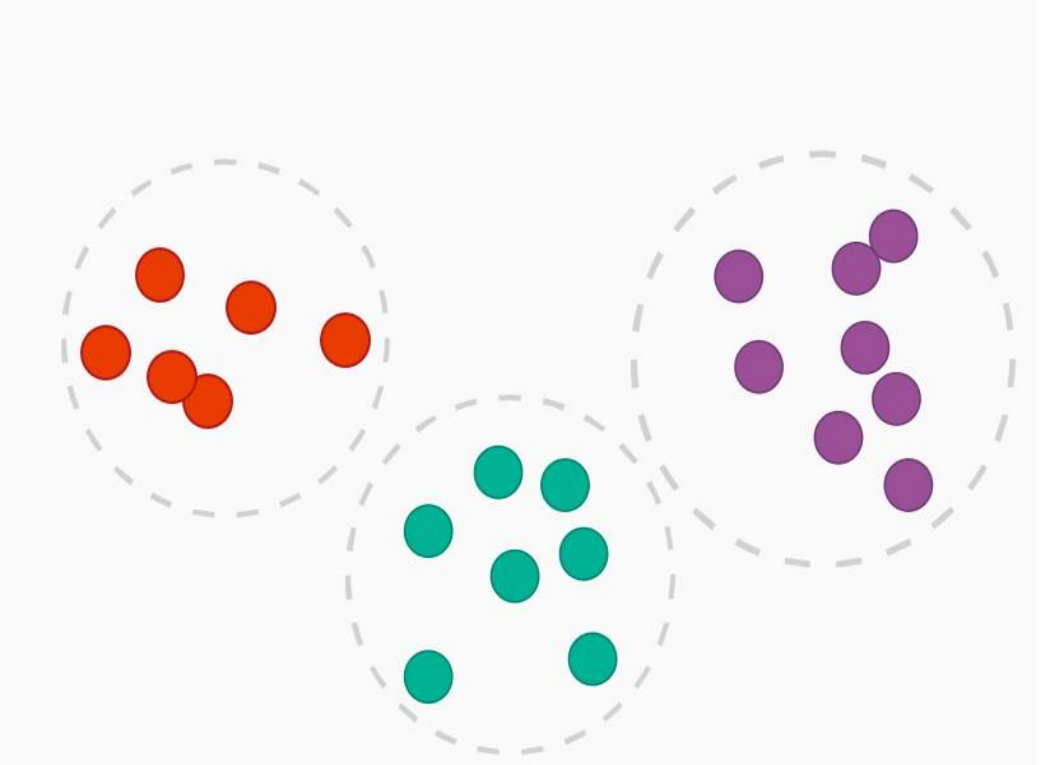


Unsupervised Learning

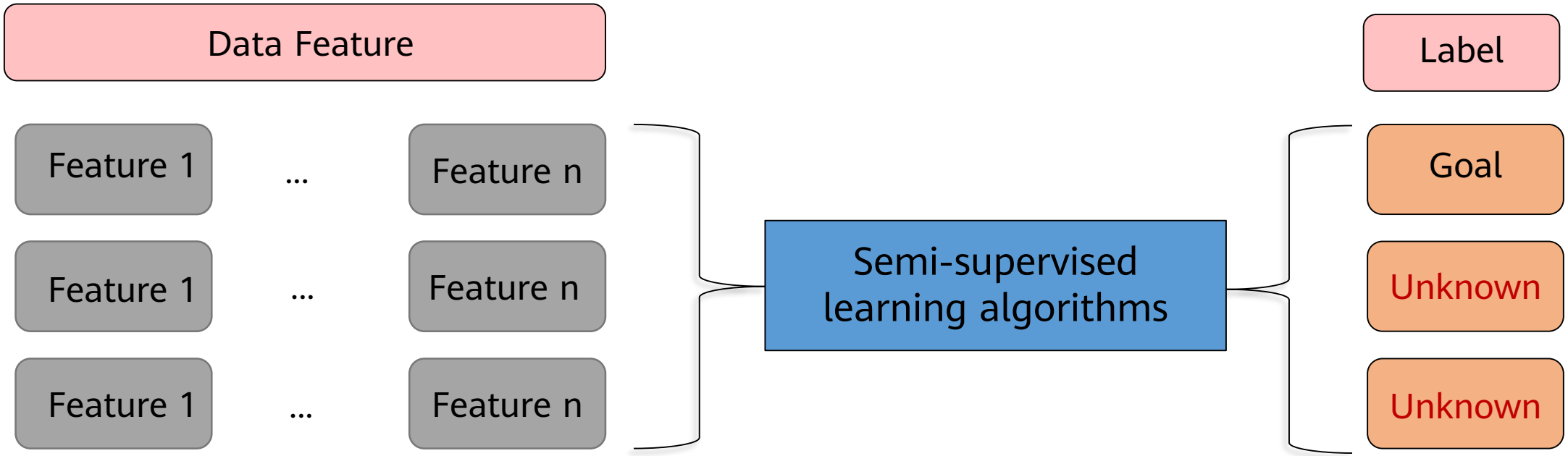


Unsupervised Learning - Clustering Questions

- Clustering: classifies samples in a sample dataset into several categories based on the clustering model. The similarity of samples belonging to the same category is high.
  - Which audiences like to watch movies of the same subject?
  - Which of these components are damaged in a similar way?



Semi-Supervised Learning

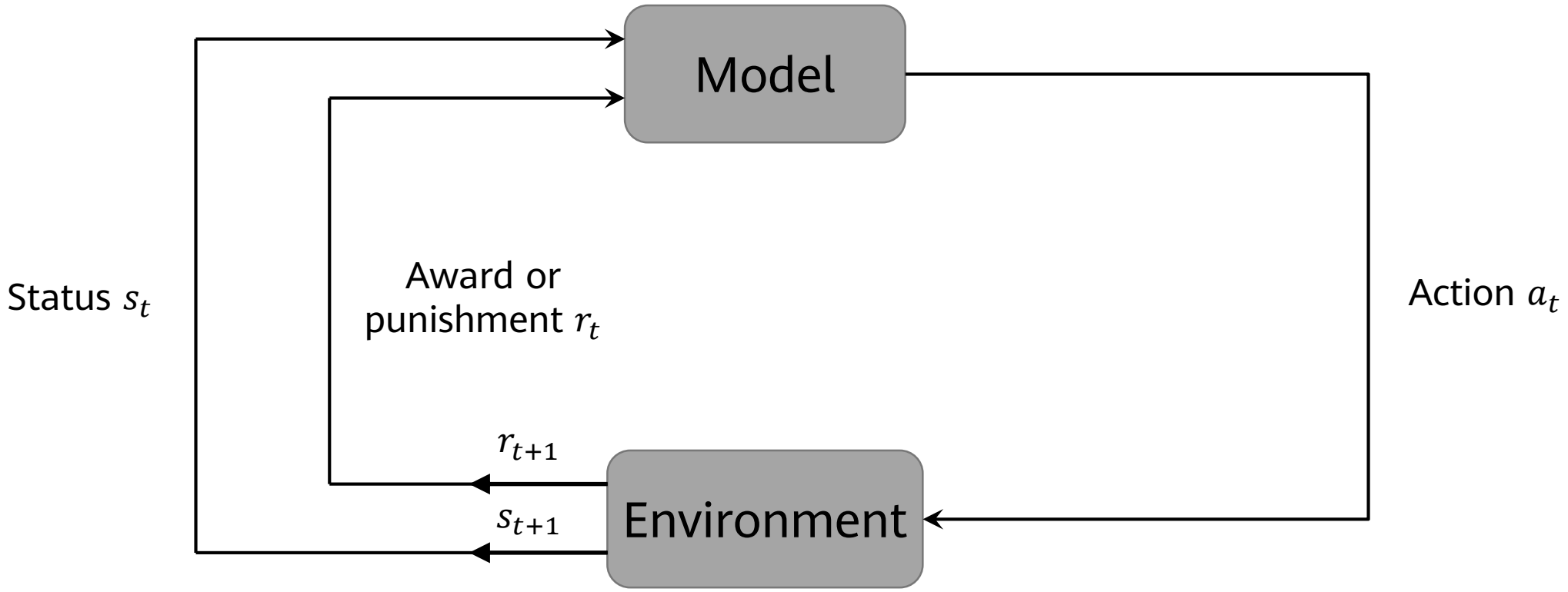


Weather	Temperature	Wind Speed
Sunny	Warm	Strong
Rainy	Cold	Fair
Sunny	Cold	Weak

Enjoy Sports
Yes
/
/

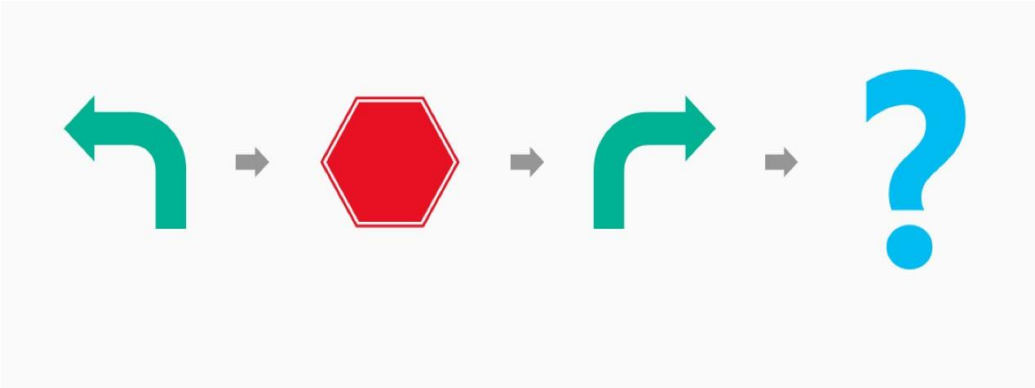
Reinforcement Learning

- The model perceives the environment, takes actions, and makes adjustments and choices based on the status and award or punishment.



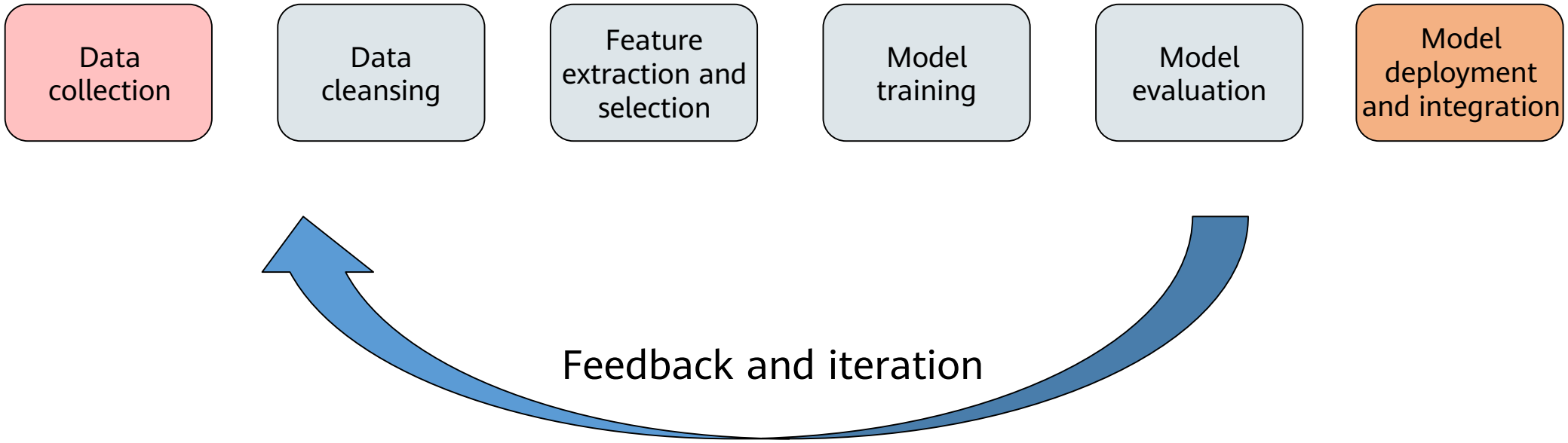
Reinforcement Learning - Best Behavior

- Reinforcement learning: always looks for best behaviors. Reinforcement learning is targeted at machines or robots.
  - Autopilot: Should it brake or accelerate when the yellow light starts to flash?
  - Cleaning robot: Should it keep working or go back for charging?



Contents

Machine Learning Process



Basic Machine Learning Concept — Dataset

- Dataset: a collection of data used in machine learning tasks. Each data record is called a sample. Events or attributes that reflect the performance or nature of a sample in a particular aspect are called features.
- Training set: a dataset used in the training process, where each sample is referred to as a training sample. The process of creating a model from data is called learning (training).
- Test set: Testing refers to the process of using the model obtained after learning for prediction. The dataset used is called a test set, and each sample is called a test sample.

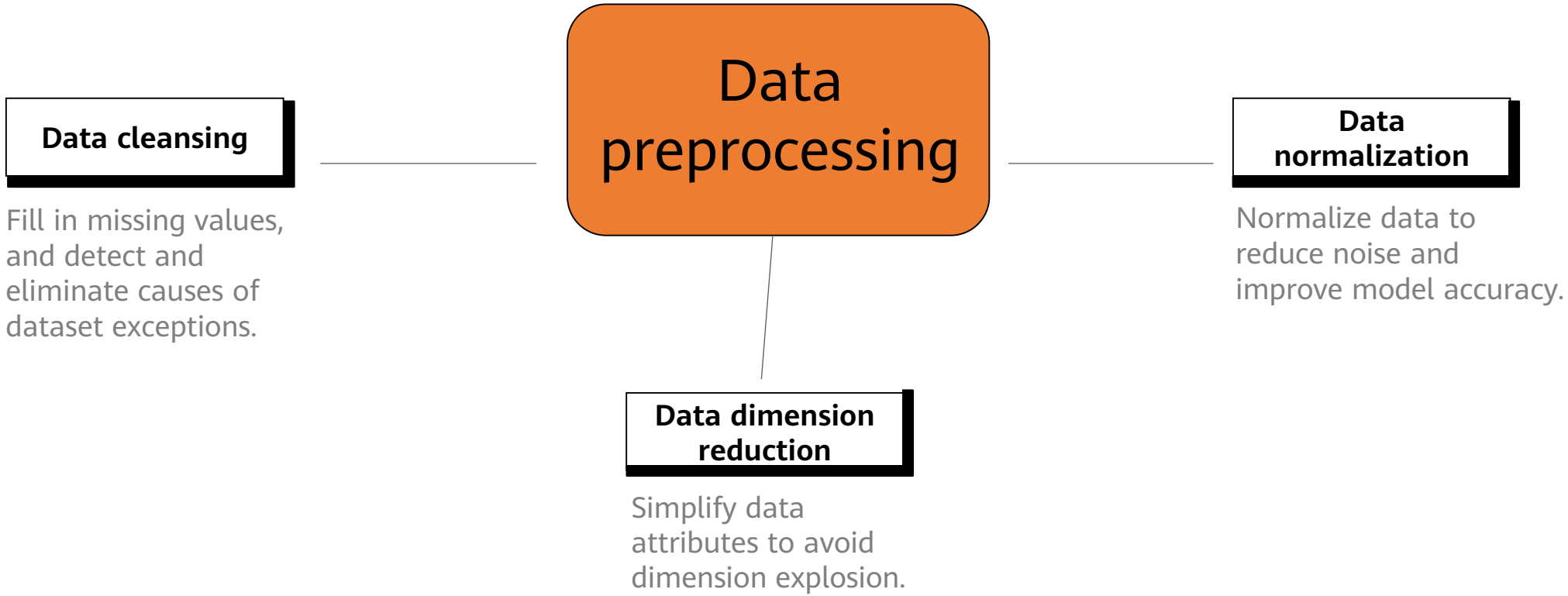
Checking Data Overview

- Typical dataset form

		Feature 1	Feature 2	Feature 3	Label
	No.	Area	School Districts	Direction	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

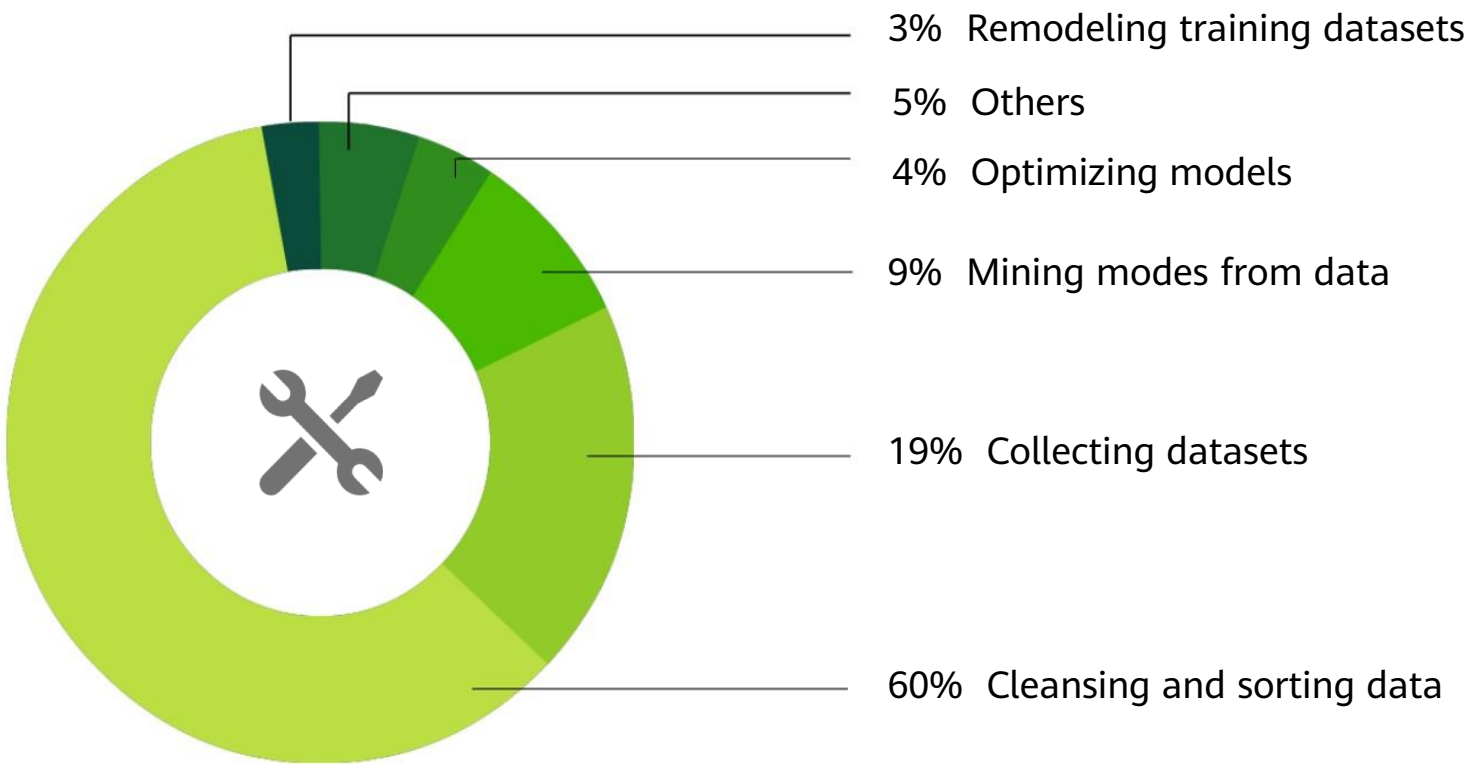
Importance of Data Processing

- Data is crucial to models. It is the ceiling of model capabilities. Without good data, there is no good model.



Workload of Data Cleansing

- Statistics on data scientists' work in 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, the collected data can be used by algorithms only after being preprocessed. The preprocessing operations include the following:
  - Data filtering
  - Processing of lost data
  - Processing of possible exceptions, errors, or abnormal values
  - Combination of data from multiple data sources
  - Data consolidation

Dirty Data (1)

- Generally, real data may have some quality problems.
  - Incompleteness: contains missing values or the data that lacks attributes
  - Noise: contains incorrect records or exceptions.
  - Inconsistency: contains inconsistent records.

Dirty Data (2)

#	Id	Name	Birthday	Gender	IsTeacher	#Students	Country	City
1	111	John	31/12/1990	M	0	0	Ireland	Dublin
2	222	Mery	15/10/1978	F	1	15	Iceland	
3	333	Alice	19/04/2000	F	0	0	Spain	Madrid
4	444	Mark	01/11/1997	M	0	0	France	Paris
5	555	Alex	15/03/2000	A	1	23	Germany	Berlin
6	555	Peter	1983-12-01	M	1	10	Italy	Rome
7	777	Calvin	05/05/1995	M	0	0	Italy	Italy
8	888	Roxane	03/08/1948	F	0	0	Portugal	Lisbon
9	999	Anne	05/09/1992	F	0	5	Switzerland	Geneva
10	101010	Paul	14/11/1992	M	1	26	Ytali	Rome

Missing value

Invalid value

Value that should be in another column

Misspelling

Invalid duplicate item

Incorrect format

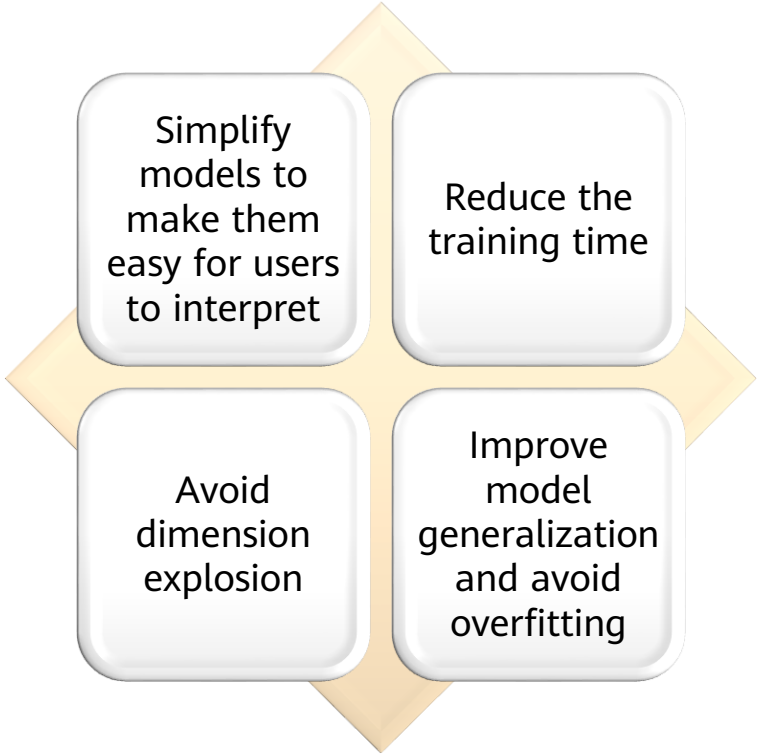
Attribute dependency

Data Conversion

- After being preprocessed, the data needs to be converted into a representation form suitable for the machine learning model. Common data conversion forms include the following:
  - With respect to classification, category data is encoded into a corresponding numerical representation.
  - Value data is converted to category data to reduce the value of variables (for age segmentation).
  - Other data
    - In the text, the word is converted into a word vector through word embedding (generally using the word2vec model, BERT model, etc).
    - Process image data (color space, grayscale, geometric change, Haar feature, and image enhancement)
  - Feature engineering
    - Normalize features to ensure the same value ranges for input variables of the same model.
    - Feature expansion: Combine or convert existing variables to generate new features, such as the average.

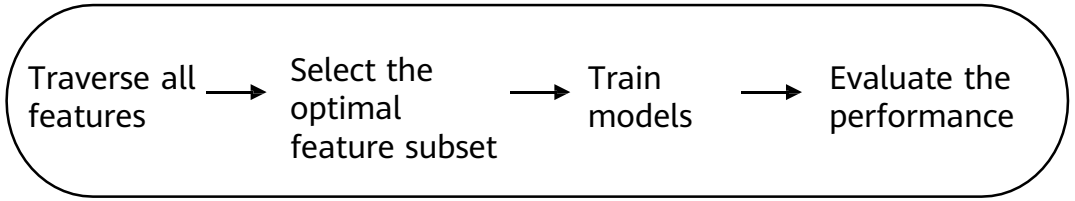
Necessity of Feature Selection

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



Feature Selection Methods - Filter

- Filter methods are independent of the model during feature selection.



Procedure of a filter method

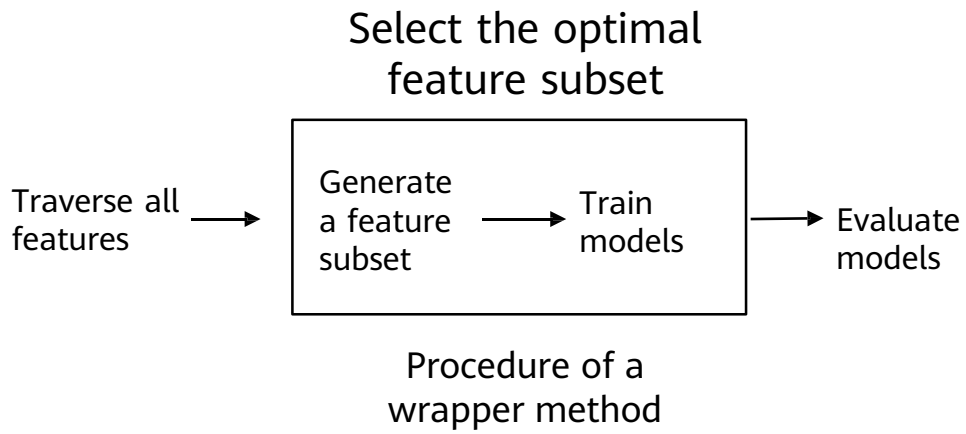
By evaluating the correlation between each feature and the target attribute, these methods use a statistical measure to assign a value to each feature. Features are then sorted by score, which is helpful for preserving or eliminating specific features.

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

Limitations
- The filter method tends to select redundant variables as the relationship between features is not considered.

Feature Selection Methods - Wrapper

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



Procedure of a wrapper method

Wrapper methods consider feature selection as a search issue for which different combinations are evaluated and compared. A predictive model is used to evaluate a combination of features and assign a score based on model accuracy.

Common methods

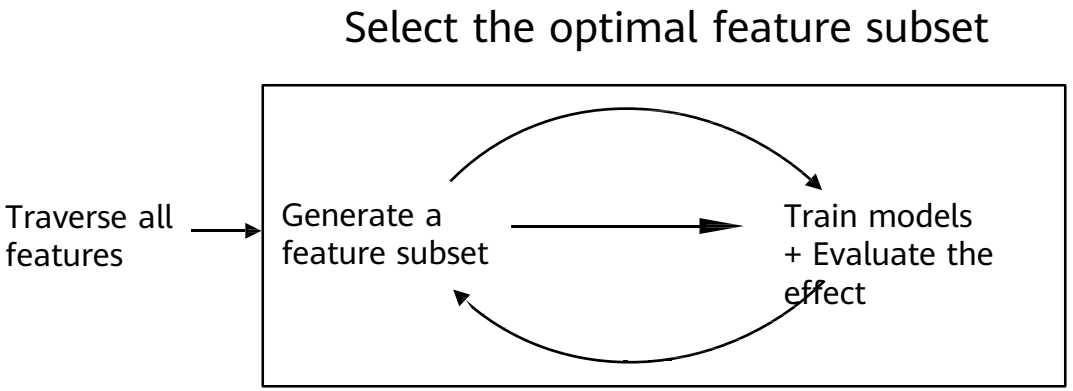
- Recursive feature elimination (RFE)

Limitations

- Wrapper methods train a new model for each subset, resulting in **a huge number of computations**.
- A feature set with the best performance is usually provided for a specific type of model.

Feature Selection Methods - Embedded

- Embedded methods consider feature selection as a part of model construction.



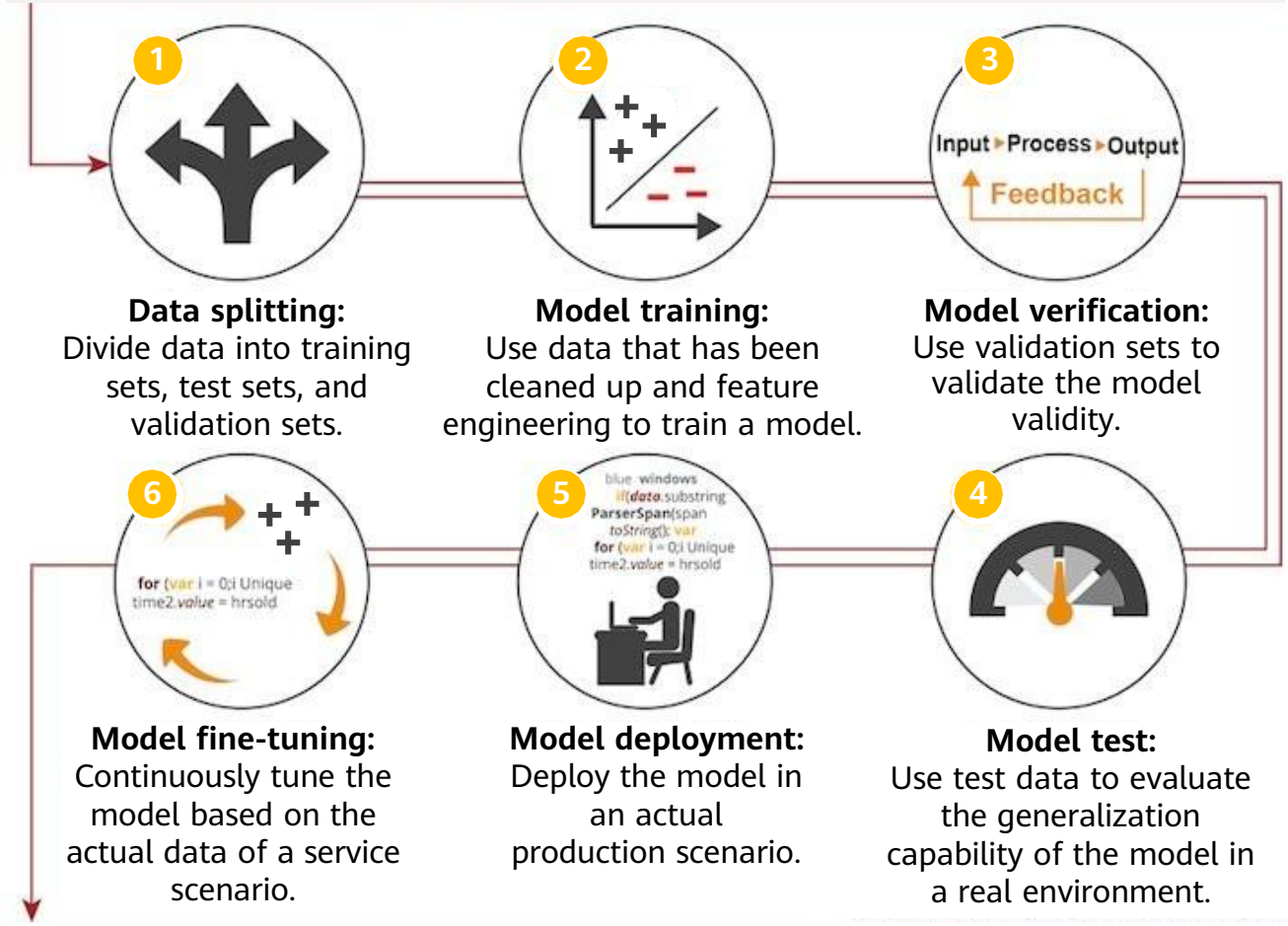
Procedure of an embedded method

The most common type of embedded feature selection method is the **regularization method**. Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm that bias the model toward lower complexity and reduce the number of features.

Common methods
- Lasso regression
- Ridge regression

Overall Procedure of Building a Model

Model Building Procedure



Data splitting:
Divide data into training sets, test sets, and validation sets.

Model training:
Use data that has been cleaned up and feature engineering to train a model.

Model verification:
Use validation sets to validate the model validity.

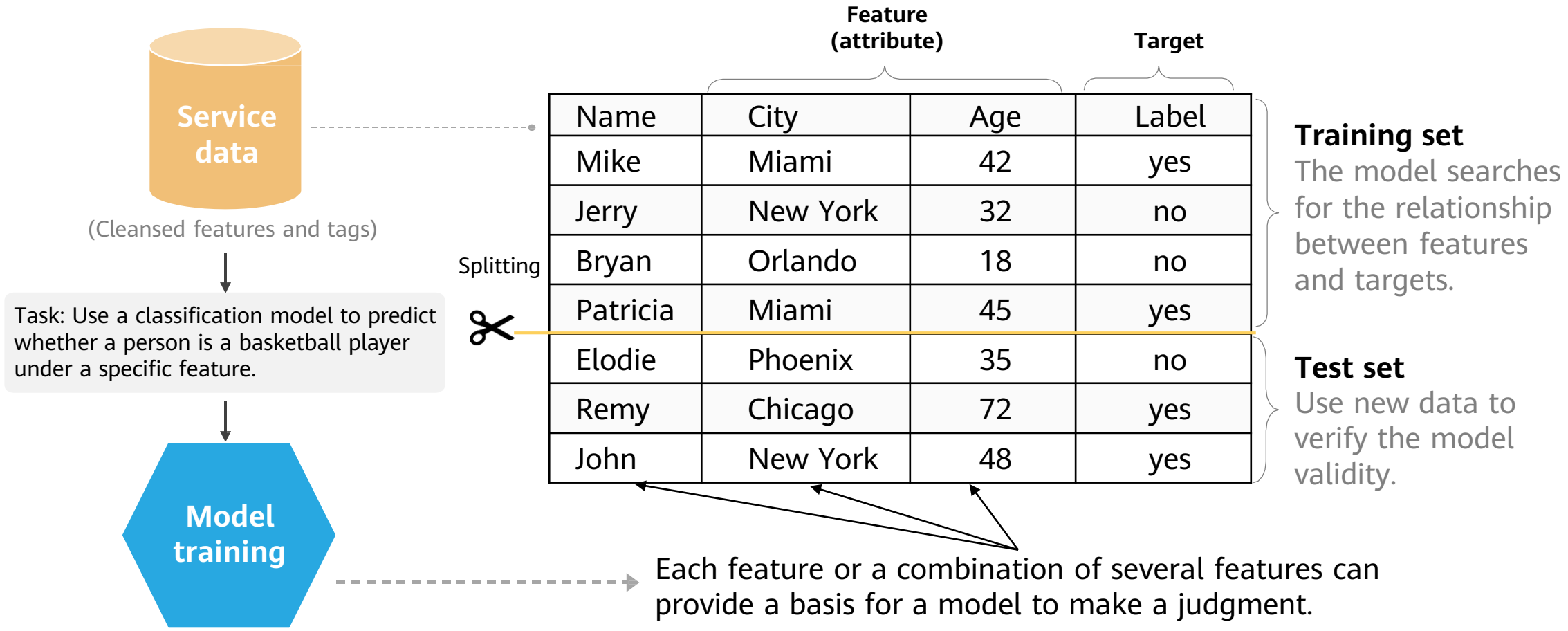
Model fine-tuning:
Continuously tune the model based on the actual data of a service scenario.

Model deployment:
Deploy the model in an actual production scenario.

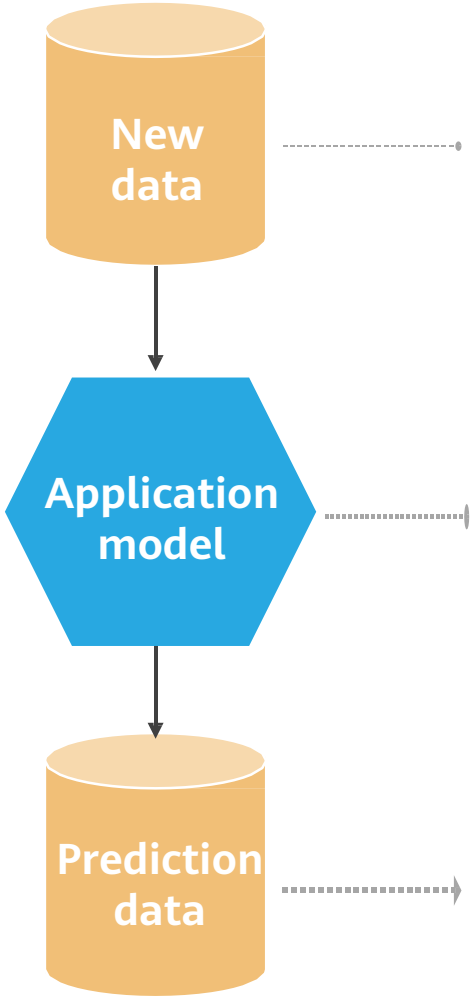
Model test:
Use test data to evaluate the generalization capability of the model in a real environment.

Examples of Supervised Learning - Learning Phase

- Use the classification model to predict whether a person is a basketball player.



Examples of Supervised Learning - 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, it is not known whether the people are basketball players.

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**

Possibility prediction
Apply the model to the new data to predict whether the customer will change the supplier.

What Is a Good Model?

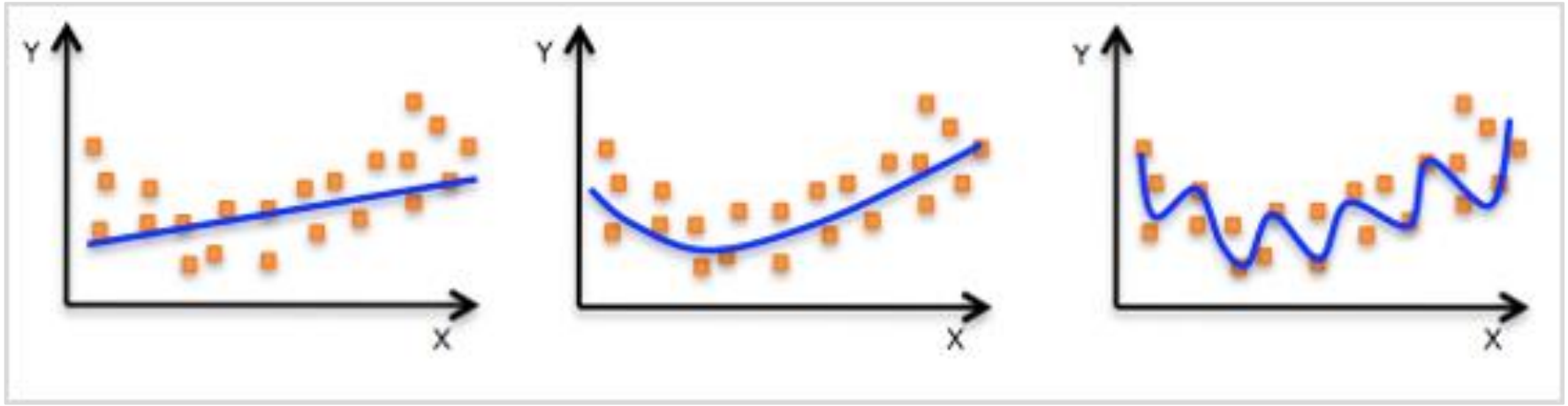
- **Generalization capability**
  Can it accurately predict the actual service data?
- **Interpretability**
  Is the prediction result easy to interpret?
- **Prediction speed**
  How long does it take to predict each piece of data?
- **Practicability**
  Is the prediction rate still acceptable when the service volume increases with a huge data volume?

Model Validity (1)

- Generalization capability: The goal of machine learning is that the model obtained after learning should perform well on new samples, not just on samples used for training. The capability of applying a model to new samples is called generalization or robustness.
- Error: difference between the sample result predicted by the model obtained after learning and the actual sample result.
  - Training error: error that you get when you run the model on the training data.
  - Generalization error: error that you get when you run the model on new samples. Obviously, we prefer a model with a smaller generalization error.
- Underfitting: occurs when the model or the algorithm does not fit the data well enough.
- Overfitting: occurs when the training error of the model obtained after learning is small but the generalization error is large (poor generalization capability).

Model Validity (2)

- Model capacity: model's capability of fitting functions, which is also called model complexity.
  - When the capacity suits the task complexity and the amount of training data provided, the algorithm effect is usually optimal.
  - Models with insufficient capacity cannot solve complex tasks and underfitting may occur.
  - A high-capacity model can solve complex tasks, but overfitting may occur if the capacity is higher than that required by a task.



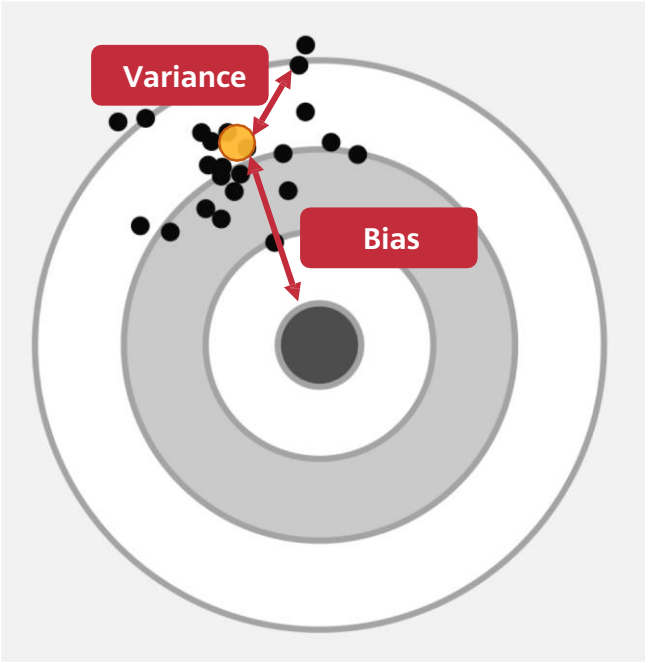
Underfitting
Not all features are learned.

Good fitting

Overfitting
Noises are learned.

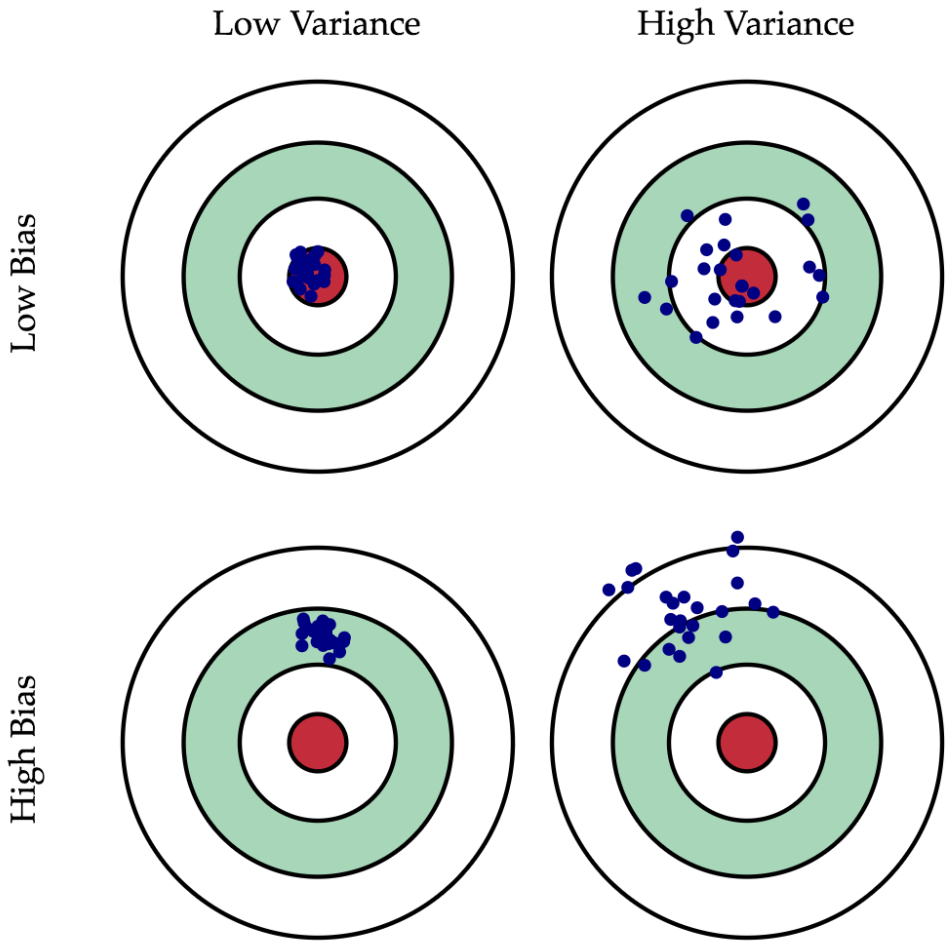
Overfitting Cause — Error

- Total error of final prediction = Bias2 + Variance + Irreducible error
- Generally, the prediction error can be divided into two types:
  - Error caused by "bias"
  - Error caused by "variance"
- Variance:
  - Offset of the prediction result from the average value
  - Error caused by the model's sensitivity to small fluctuations in the training set
- Bias:
  - Difference between the expected (or average) prediction value and the correct value we are trying to predict.



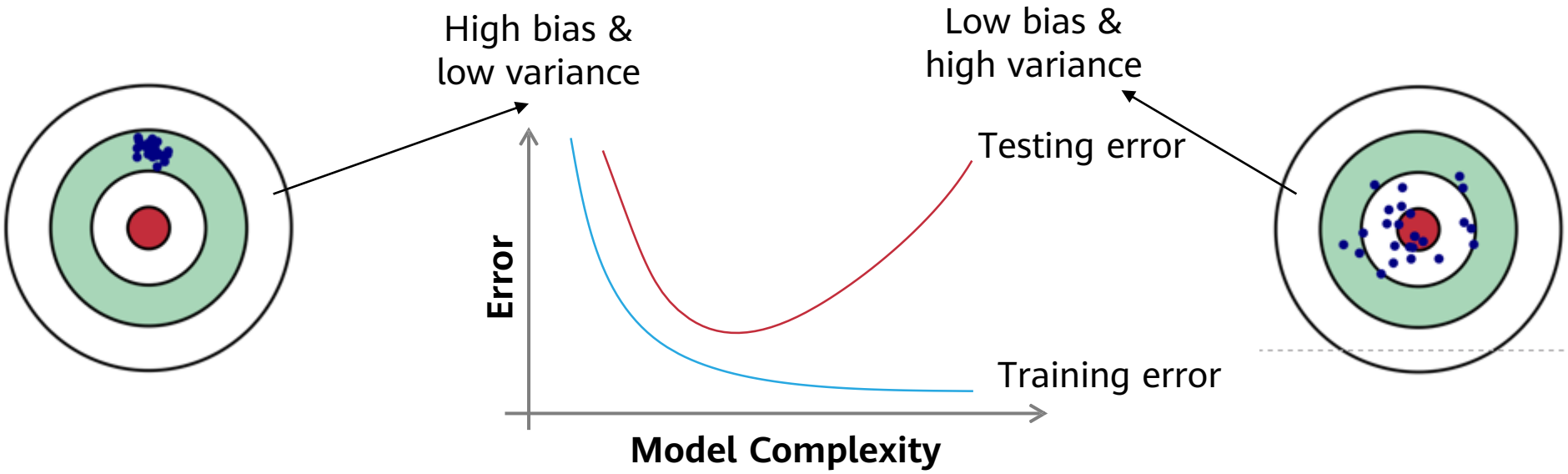
Variance and Bias

- Combinations of variance and bias are as follows:
  - Low bias & low variance –> Good model
  - Low bias & high variance
  - High bias & low variance
  - High bias & high variance –> Poor model
- Ideally, we want a model that can accurately capture the rules in the training data and summarize the invisible data (new data). However, it is usually impossible for the model to complete both tasks at the same time.



Model Complexity and Error

- As the model complexity increases, the training error decreases.
- As the model complexity increases, the test error decreases to a certain point and then increases in the reverse direction, forming a convex curve.



Machine Learning Performance Evaluation - Regression

- The closer the Mean Absolute Error (MAE) is to 0, the better the model can fit the training data.

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

- Mean Square Error (MSE)

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

- The value range of R^2 is (–∞, 1]. A larger value indicates that the model can better fit the training data. TSS indicates the difference between samples. RSS indicates the difference between the predicted value and sample value.

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

Machine Learning Performance Evaluation - Classification (1)

- Terms and definitions:
  - 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 by the classifier.
  - TN: true negative, indicating the number of negative cases that are correctly classified by the classifier.
  - FP: false positive, indicating the number of positive cases that are incorrectly classified by the classifier.
  - FN: false negative, indicating the number of negative cases that are incorrectly classified by the classifier.

Actual amount \ Estimated amount	yes	no	Total
yes	TP	FN	P
no	FP	TN	N
Total	P'	N'	$P+N$

Confusion matrix

- Confusion matrix: at least an $m \times m$ table. $CM_{i,j}$ of the first m rows and m columns indicates the number of cases that actually belong to class i but are classified into class j by the classifier.
  - Ideally, for a high accuracy classifier, most prediction values should be located in the diagonal from $CM_{1,1}$ to $CM_{m,m}$ of the table while values outside the diagonal are 0 or close to 0. That is, FP and FP are close to 0.

Machine Learning Performance Evaluation - Classification (2)

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

Example of Machine Learning Performance Evaluation

- We have trained a machine learning model to identify whether the object in an image is a cat. Now we use 200 pictures to verify the model performance. Among the 200 images, objects in 170 images are cats, while others are not. The identification result of the model is that objects in 160 images are cats, while others are not.

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

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

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

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

Contents

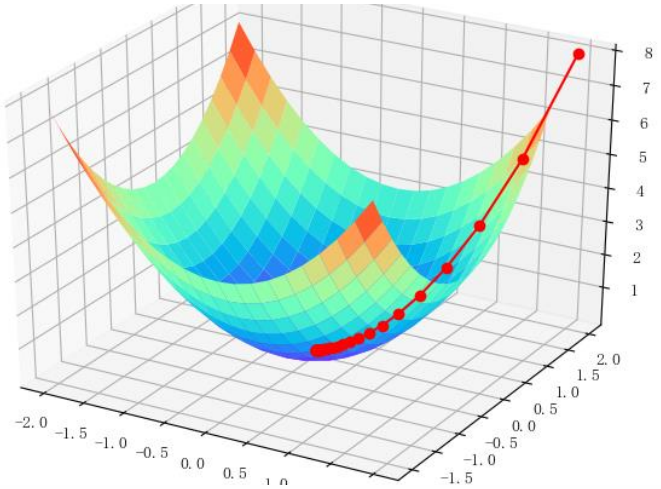
Machine Learning Training Method - Gradient Descent (1)

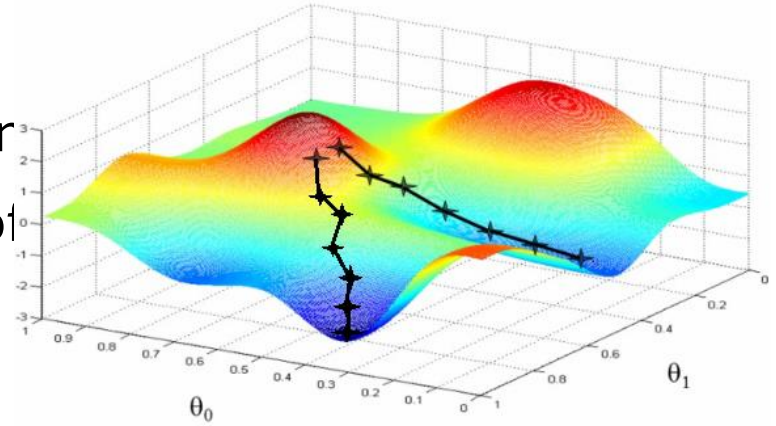
- The gradient descent method uses the negative gradient direction of the current position as the search direction, which is the steepest direction. The formula is as follows:

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

- In the formula, η indicates the learning rate and i indicates the data record number i. The weight parameter w indicates the change in each iteration.
- Convergence: The value of the objective functior changes very little, or the maximum number of iterations is reached.

Cost surface





HUAWEI

Machine Learning Training Method - Gradient Descent (2)

- Batch Gradient Descent (BGD) uses the samples (m in total) in all datasets to update the weight parameter based on the gradient value at the current point.

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

- Stochastic Gradient Descent (SGD) randomly selects a sample in a dataset to update the weight parameter based on the gradient value at the current point.

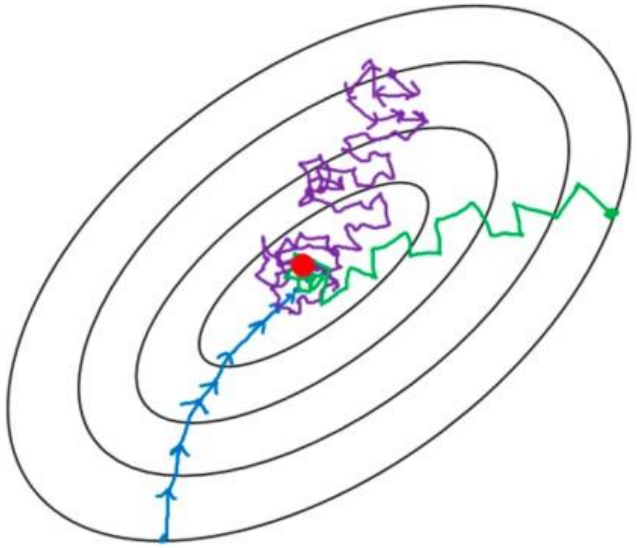
$$w_{k+1} = w_k - \eta \nabla f_{w_k}(x^i)$$

- Mini-Batch Gradient Descent (MBGD) combines the features of BGD and SGD and selects the gradients of n samples in a dataset 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 Method - Gradient Descent (3)

- Comparison of three gradient descent methods
  - In the SGD, samples selected for each training are stochastic. Such instability causes the loss function to be unstable or even causes reverse displacement when the loss function decreases to the lowest point.
  - BGD has the highest stability but consumes too many computing resources. MBGD is a method that balances SGD and BGD.



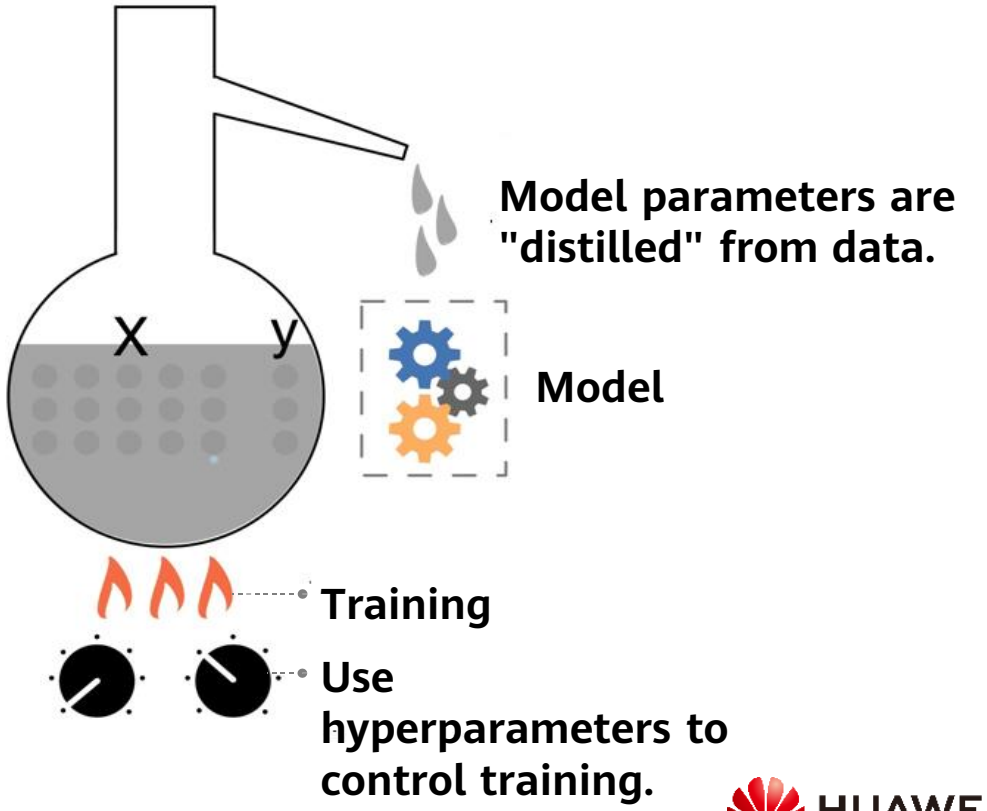
BGD
Uses **all** training samples for training each time.

SGD
Uses **one** training sample for training each time.

MBGD
Uses a certain number of training samples for training each time.

Parameters and Hyperparameters in Models

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



Hyperparameters of a Model

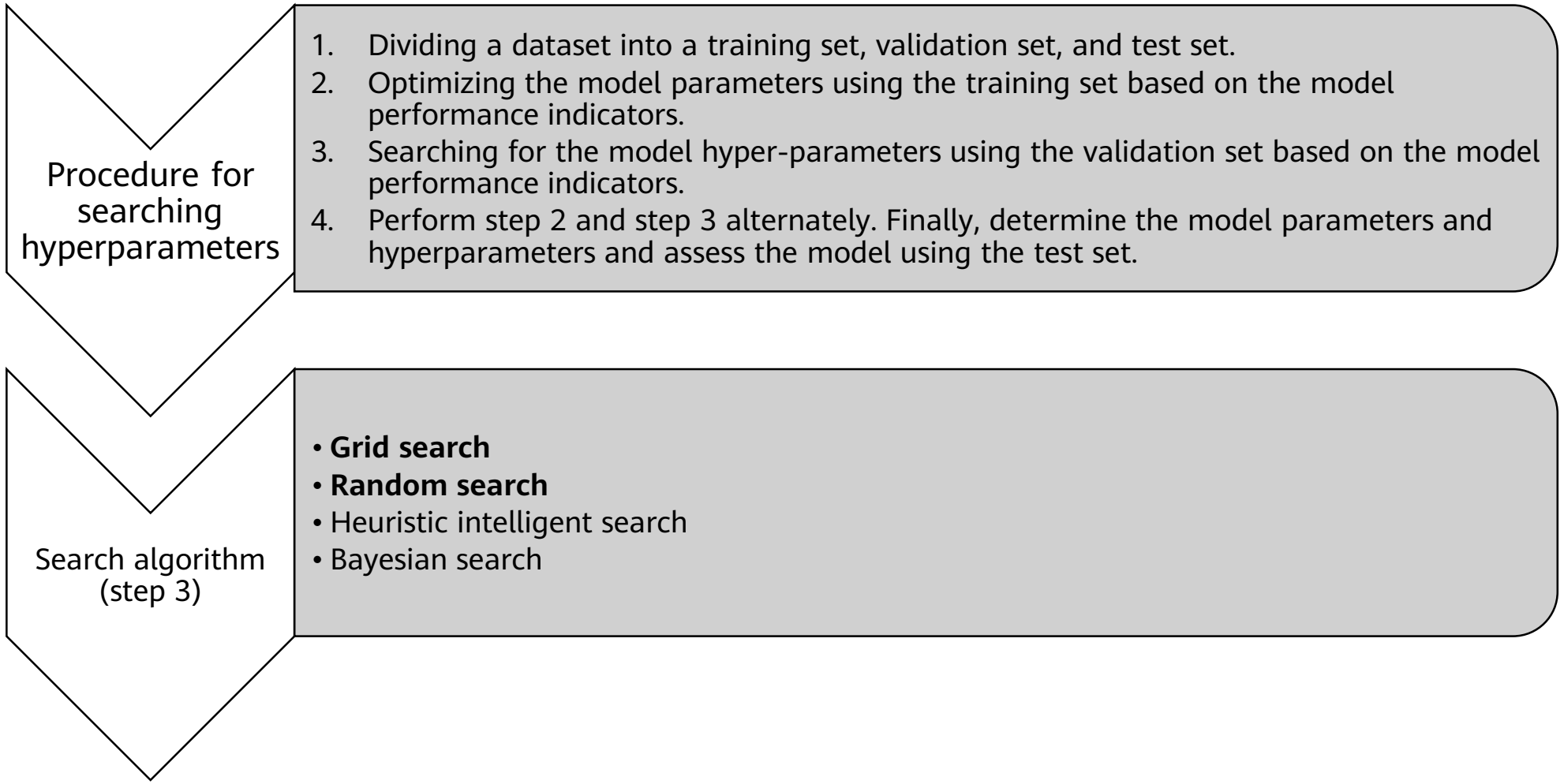
- Often used in model parameter estimation processes.
- Often specified by the practitioner.
- Can often be set using heuristics.
- Often tuned for a given predictive modeling problem.

Model hyperparameters are external configurations of models.

- λ during Lasso/Ridge regression
- Learning rate for training a neural network, number of iterations, batch size, activation function, and number of neurons
- C and σ in support vector machines (SVM)
- K in k-nearest neighbor (KNN)
- Number of trees in a random forest

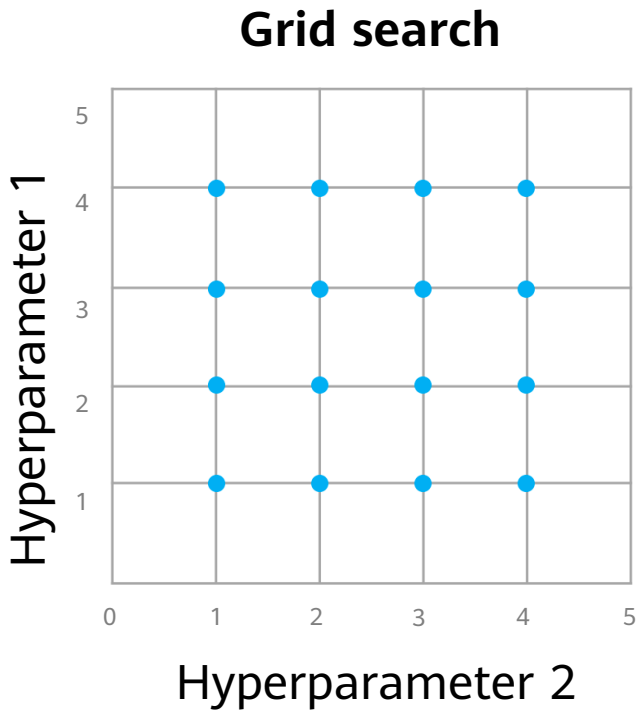
Common model hyperparameters

Hyperparameter Search Procedure and Method



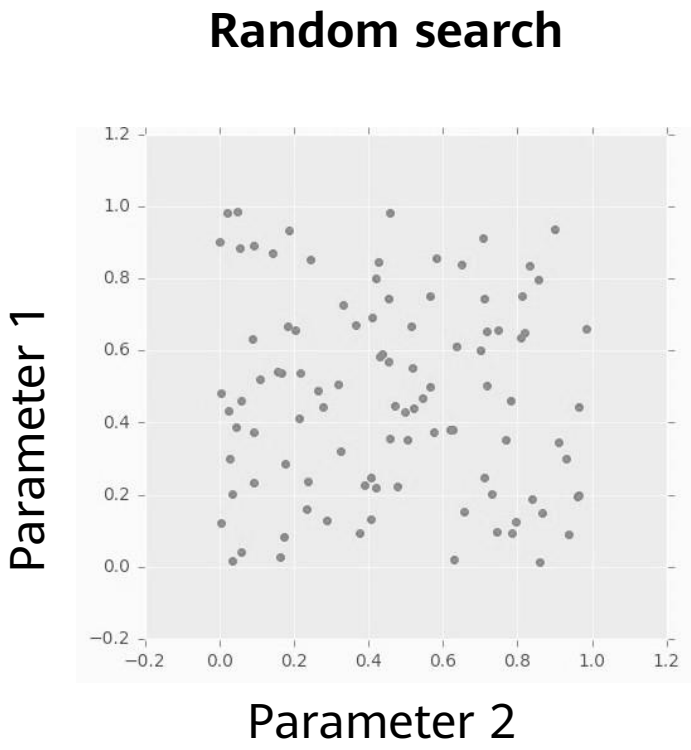
Hyperparameter Searching Method - Grid Search

- Grid search attempts to **exhaustively search** all possible hyperparameter combinations to form a hyperparameter value grid.
- In practice, the range of hyperparameter values to search is specified manually.
- Grid search is an expensive and time-consuming method.
  - This method works well when the number of hyperparameters is relatively small. Therefore, it is applicable to generally machine learning algorithms but inapplicable to neural networks
    (see the deep learning part).



Hyperparameter Searching Method - Random Search

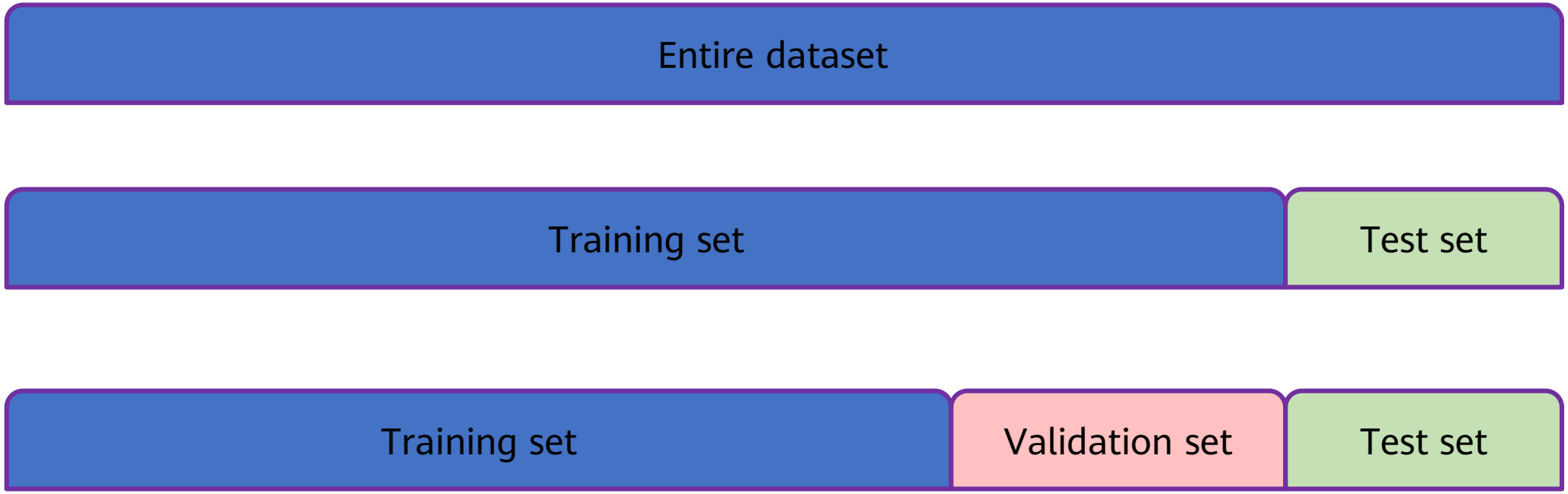
- When the hyperparameter search space is large, **random search** is better than grid search.
- In random search, each setting is sampled from the distribution of possible parameter values, in an attempt to find the best subset of hyperparameters.
- Note:
  - Search is performed within a coarse range, which then will be narrowed based on where the best result appears.
  - Some hyperparameters are more important than others, and the search deviation will be affected during random search.



Cross Validation (1)

- **Cross validation**: It is a statistical analysis method used to validate the performance of a classifier. The basic idea is to divide the original dataset into two parts: training set and validation set. Train the classifier using the training set and test the model using the validation set to check the classifier performance.
- **k-fold cross validation ($K-CV$):**
  - Divide the raw data into k groups (generally, evenly divided).
  - Use each subset as a validation set, and use the other $k-1$ subsets as the training set. A total of k models can be obtained.
  - Use the mean classification accuracy of the final validation sets of k models as the performance indicator of the $K-CV$ classifier.

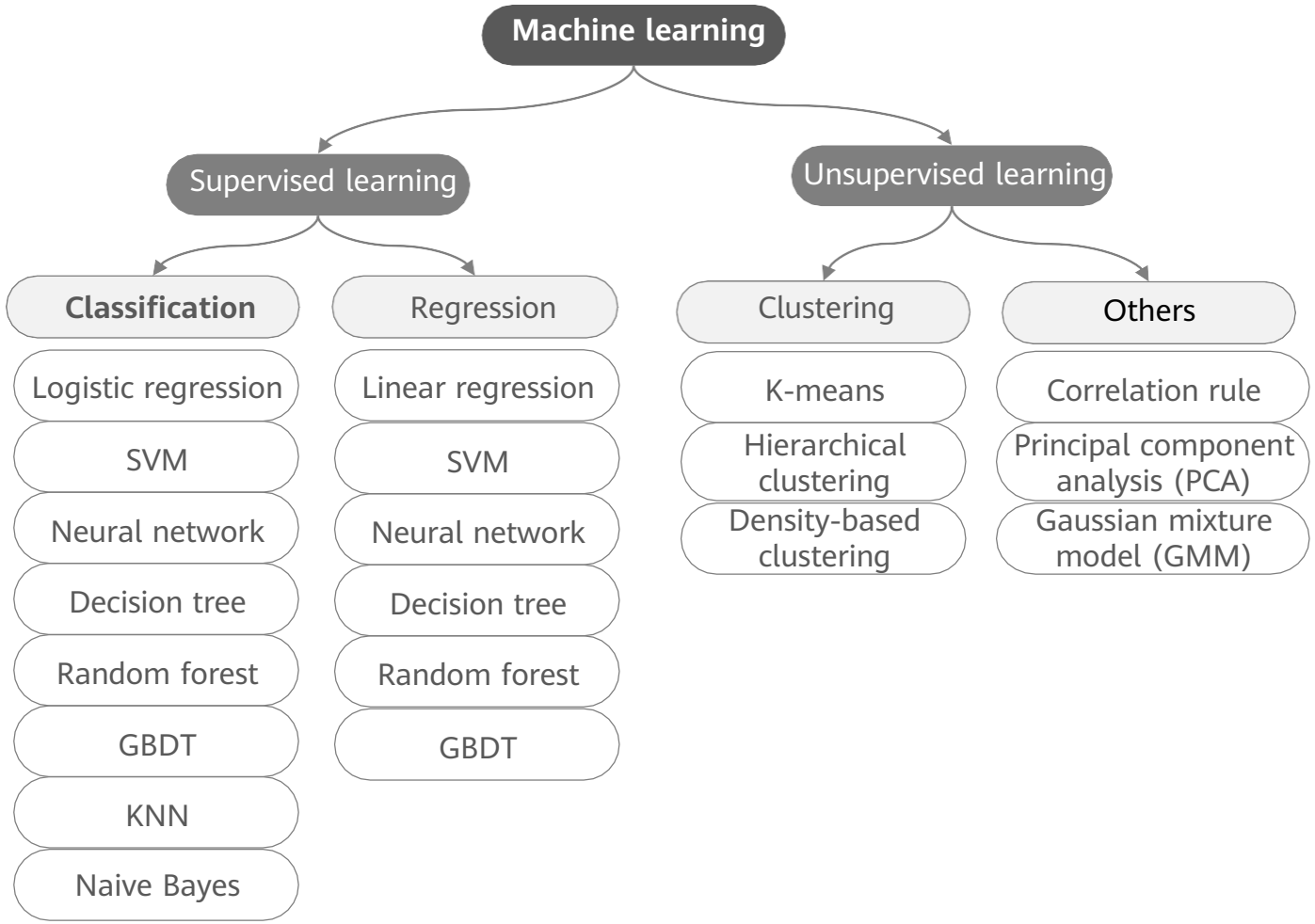
Cross Validation (2)



- Note: The K value in K-fold cross validation is also a hyperparameter.

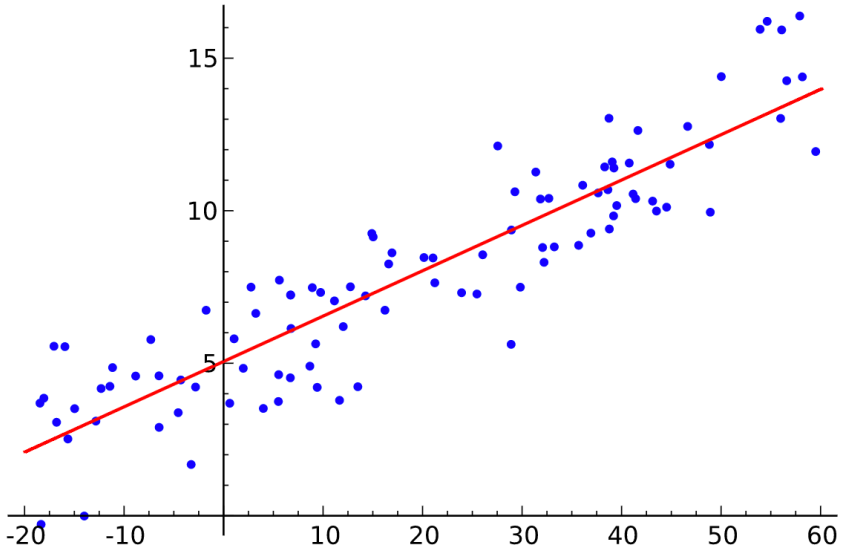
Contents

Machine Learning Algorithm Overview

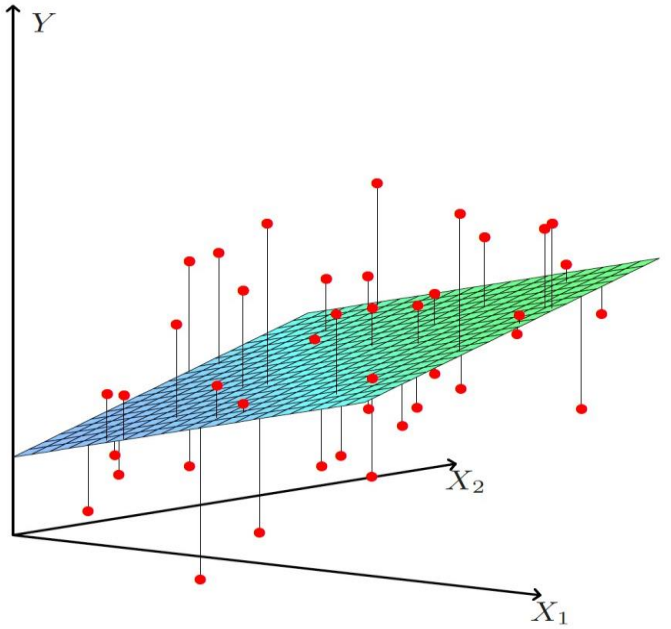


Linear Regression (1)

- Linear regression: a statistical analysis method to determine the quantitative relationships between two or more variables through regression analysis in mathematical statistics.
- Linear regression is a type of supervised learning.



Unary linear regression



Multi-dimensional linear regression

Linear Regression (2)

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

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

- The relationship between the value predicted by the model and 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 influenced by many factors independently. According to the central limit theorem, the error ε follows normal distribution. According to the normal distribution function and maximum likelihood estimation, the loss function of linear regression is as follows:

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

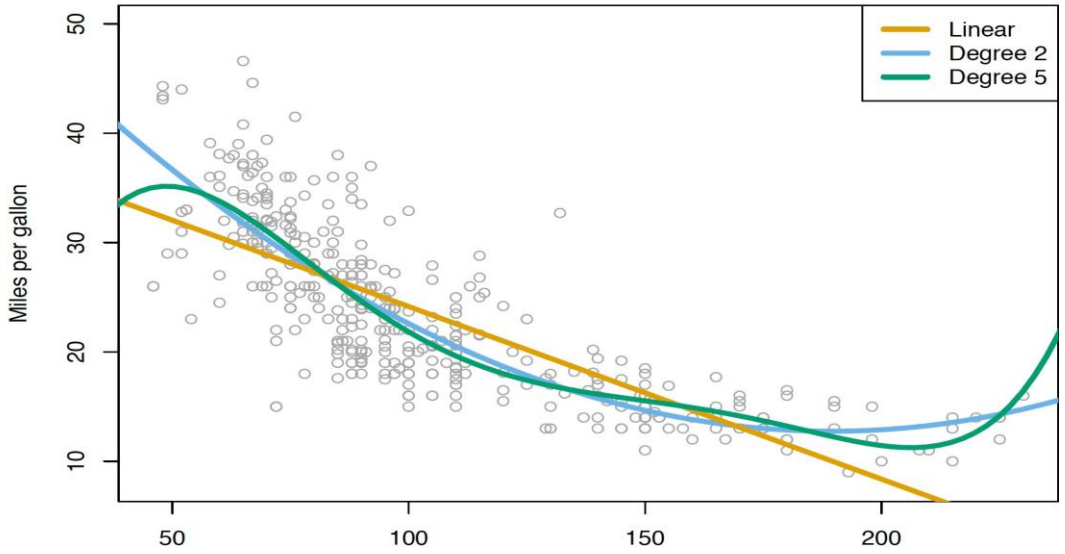
- To make the predicted value close to the actual value, we need to minimize the loss value. We can use the gradient descent method to calculate the weight parameter w when the loss function reaches the minimum, and then complete model building.

Linear Regression Extension - Polynomial Regression

- Polynomial regression is an extension of linear regression. Generally, the complexity of a dataset exceeds the possibility of fitting by a straight line. That is, obvious underfitting occurs if the original linear regression model is used. The solution is to use polynomial regression.

$$h_w(x) = w_1 x + w_2 x^2 + \mathrm{L} + w_n x^n + b$$

- where, the nth power is a polynomial regression dimension (degree).
- Polynomial regression belongs to linear regression as the relationship between its weight parameters w is still linear while its nonlinearity is reflected in the feature dimension.



Comparison between linear regression and polynomial regression

HUAWEI

Linear Regression and Overfitting Prevention

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

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

- Regularization terms (norm): The regularization term here is called L2-norm. Linear regression that uses this loss function is also called Ridge regression.

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

- Linear regression with absolute loss is called Lasso regression.

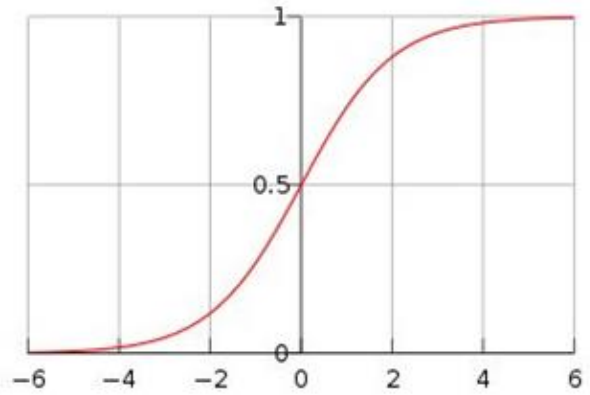
Logistic Regression (1)

- Logistic regression: The logistic regression model is used to solve classification problems. The model is defined as follows:

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

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

where w indicates the weight, b indicates the bias, and $wx + b$ is regarded as the linear function of x. Compare the preceding two probability values. The class with a higher probability value is the class of x.



Logistic Regression (2)

- Both the logistic regression model and linear regression model are generalized linear models. Logistic regression introduces nonlinear factors (the sigmoid function) based on linear regression and sets thresholds, so it can deal with binary classification problems.
- According to the model function of logistic regression, the loss function of logistic regression can be estimated as follows by using the maximum likelihood estimation:

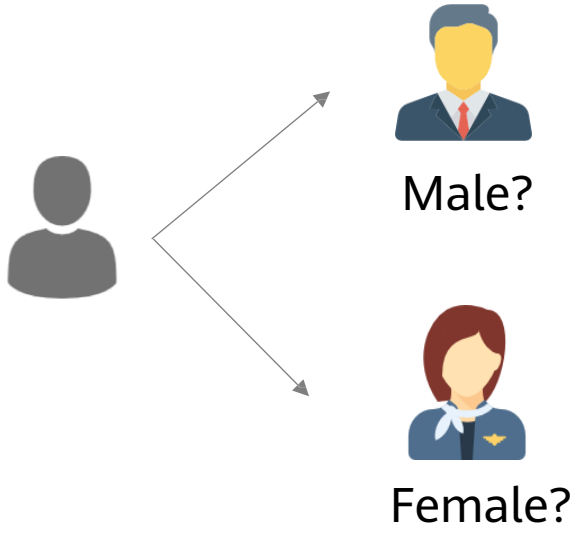
$$J(w) = -\frac{1}{m}\sum\left(y \ln h_w(x) + (1-y)\ln(1-h_w(x))\right)$$

- where w indicates the weight parameter, m indicates the number of samples, x indicates the sample, and y indicates the real value. The values of all the weight parameters w can also be obtained through the gradient descent algorithm.

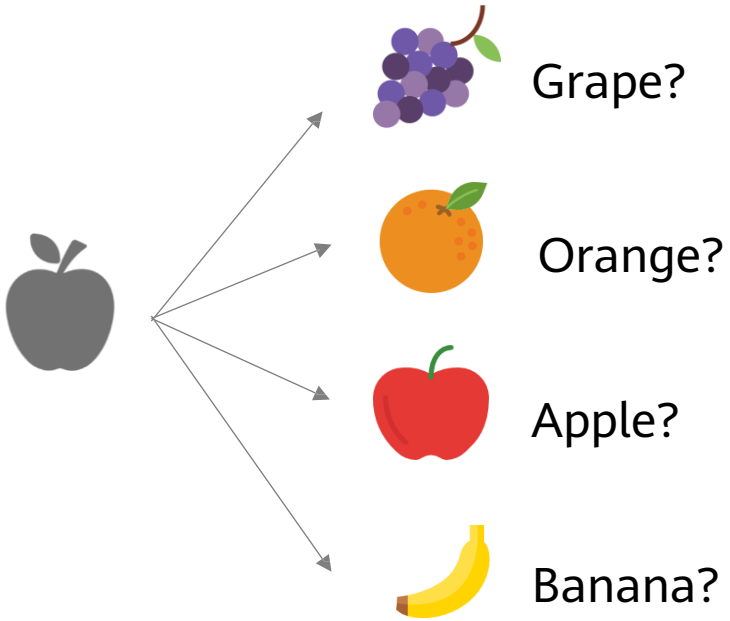
Logistic Regression Extension - Softmax Function (1)

- Logistic regression applies only to binary classification problems. For multi-class classification problems, use the Softmax function.

Binary classification problem



Multi-class classification problem



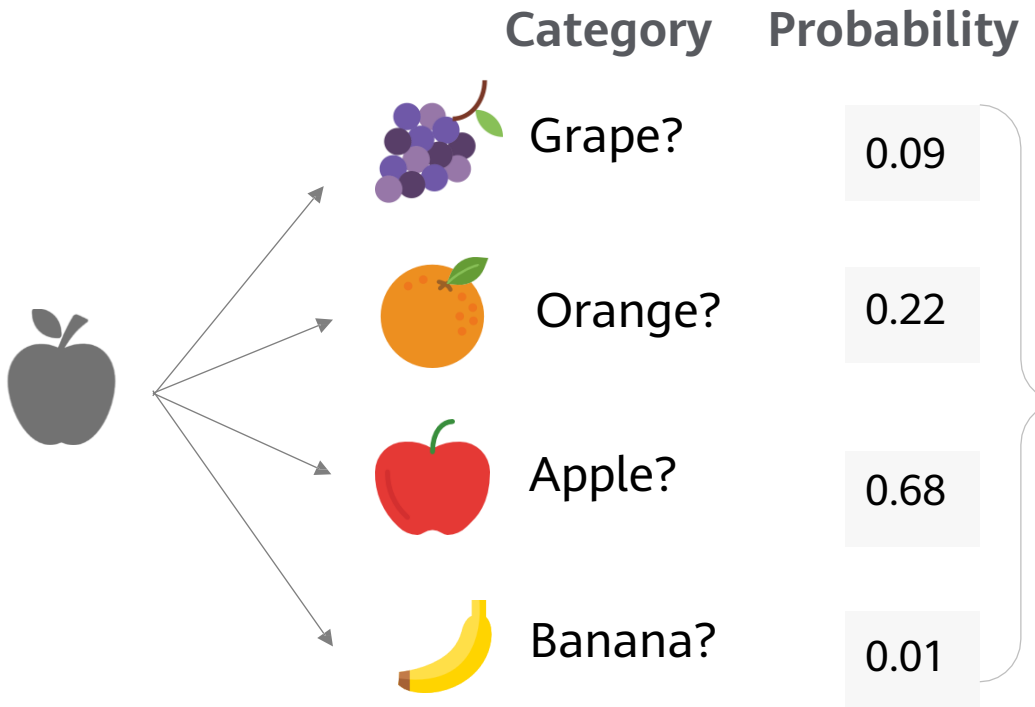
Logistic Regression Extension - Softmax Function (2)

- Softmax regression is a generalization of logistic regression that we can use for K-class classification.
- The Softmax function is used to map a K-dimensional vector of arbitrary real values to another K-dimensional vector of real values, where each vector element is in the interval (0, 1).
- The regression probability function of Softmax is as follows:

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

Logistic Regression Extension - Softmax Function (3)

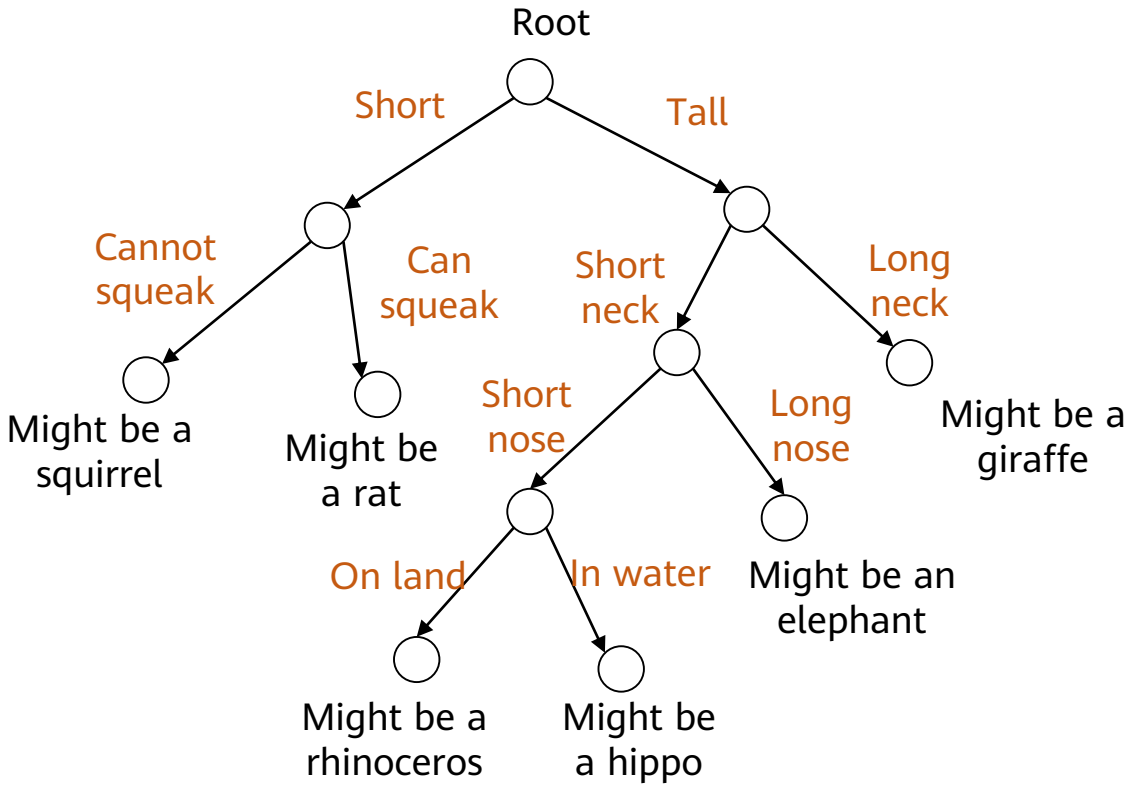
- Softmax assigns a probability to each class in a multi-class problem. These probabilities must add up to 1.
  - Softmax may produce a form belonging to a particular class. Example:



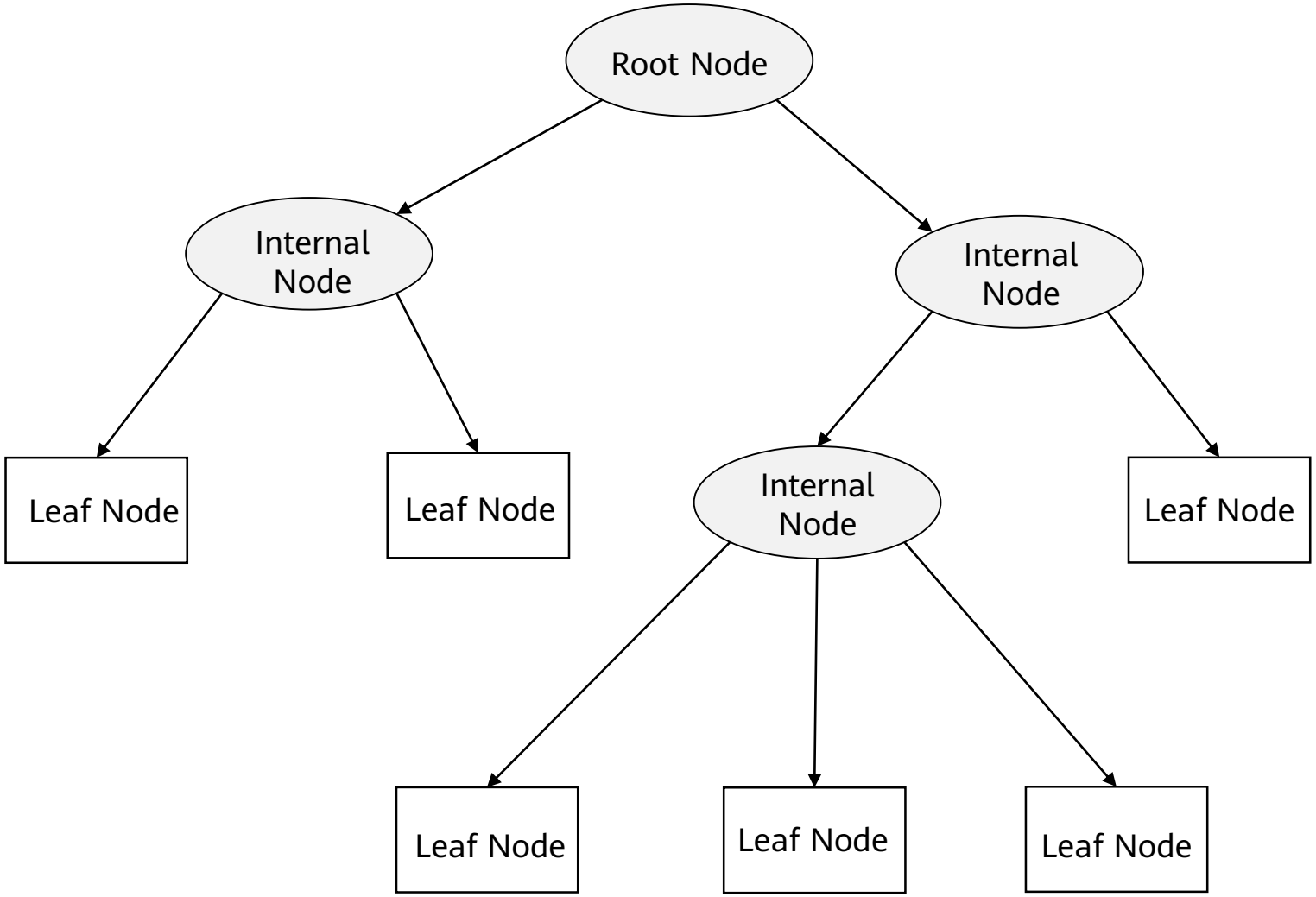
- **Sum of all probabilities:**
  - 0.09 + 0.22 + 0.68 + 0.01 =1
- Most probably, this picture is an apple.

Decision Tree

- A decision tree is a tree structure (a binary tree or a non-binary tree). Each non-leaf node represents a test on a feature attribute. Each branch represents the output of a feature attribute in a certain value range, and each leaf node stores a category. To use the decision tree, start from the root node, test the feature attributes of the items to be classified, select the output branches, and use the category stored on the leaf node as the final result.



Decision Tree Structure



Key Points of Decision Tree Construction

- To create a decision tree, we need to select attributes and determine the tree structure between feature attributes. 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.
- The metrics to quantify the 'purity' include the information entropy and GINI Index. The formula is as follows:

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

- where p_k indicates the probability that the sample belongs to class k (there are K classes in total). A greater difference between purity before segmentation and that after segmentation indicates a better decision tree.
- Common decision tree algorithms include ID3, C4.5, and CART.

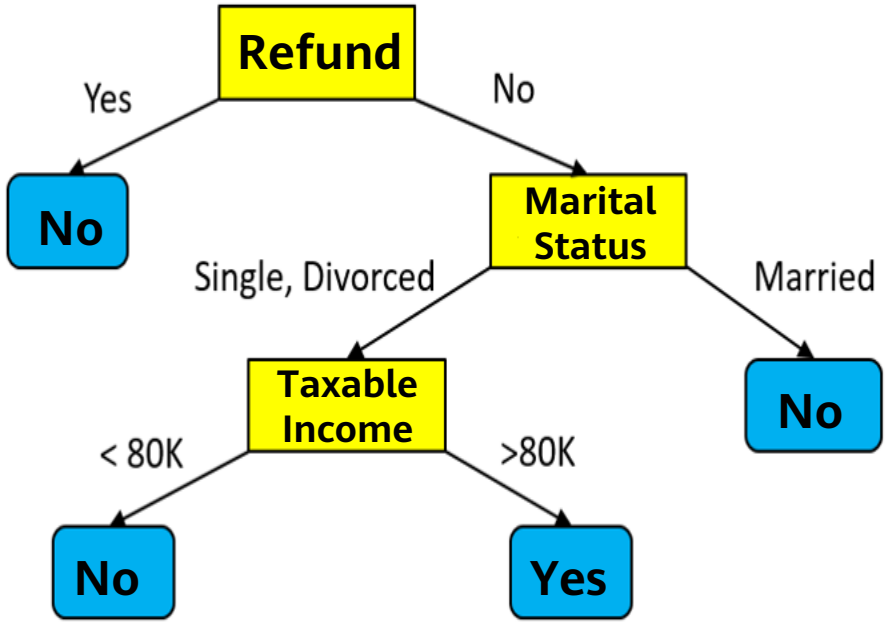
Decision Tree Construction Process

- **Feature selection:** Select a feature from the features of the training data as the split standard of the current node. (Different standards generate different decision tree algorithms.)
- **Decision tree generation:** Generate internal node upside down based on the selected features 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

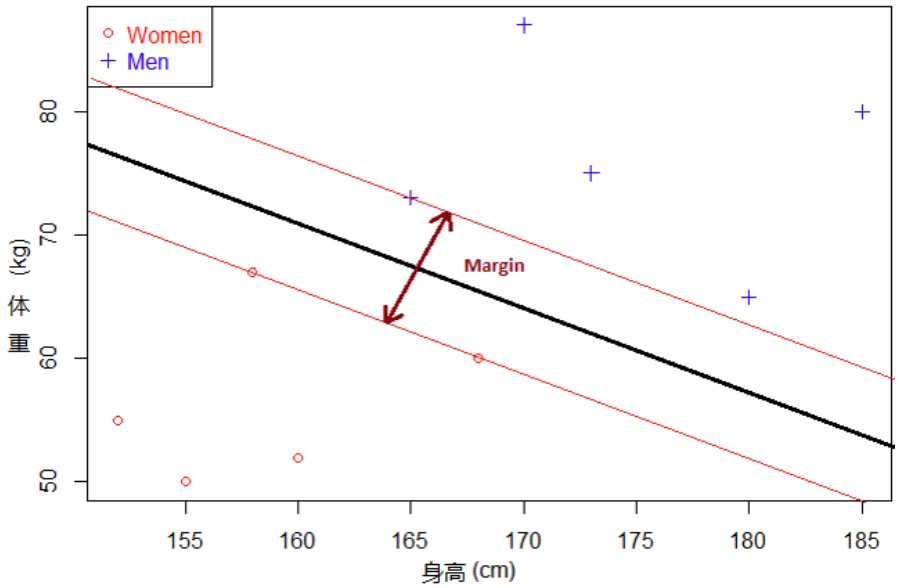
- The following figure shows a classification when a decision tree is used. The classification result is impacted by three attributes: Refund, Marital Status, and Taxable Income.

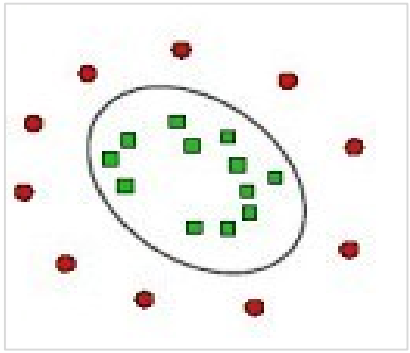
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125,000	No
2	No	Married	100,000	No
3	No	Single	70,000	No
4	Yes	Married	120,000	No
5	No	Divorced	95,000	Yes
6	No	Married	60,000	No
7	Yes	Divorced	220,000	No
8	No	Single	85,000	Yes
9	No	Married	75,000	No
10	No	Single	90,000	Yes



SVM

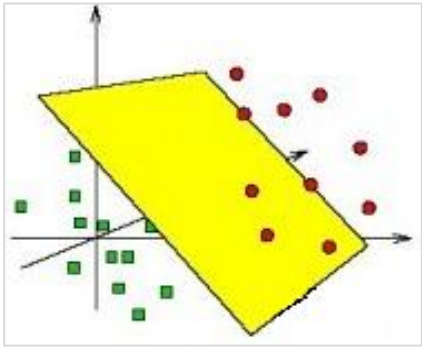
- SVM is a binary classification model whose basic model is a linear classifier defined in the eigenspace with the largest interval. SVMs also include kernel tricks that make them nonlinear classifiers. The SVM learning algorithm is the optimal solution to convex quadratic programming.





Complex segmentation in low-dimensional space

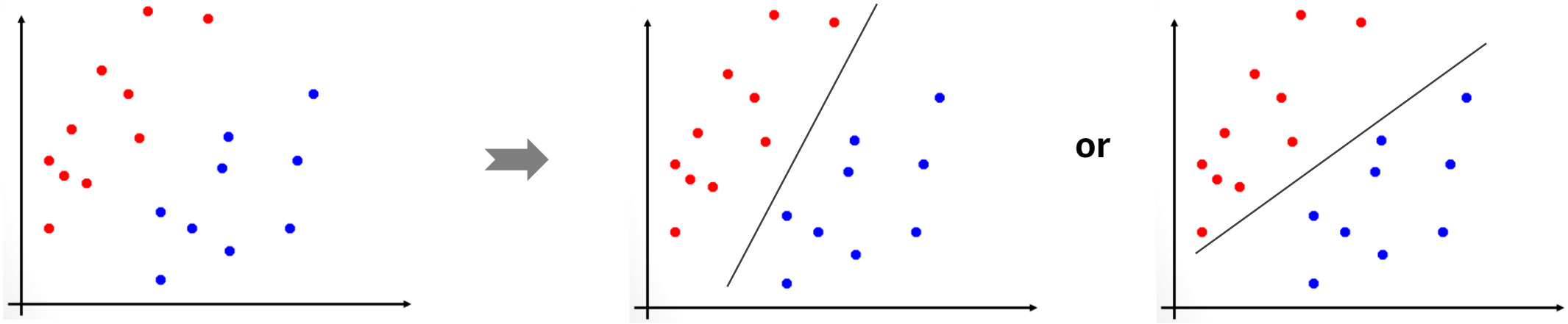
Projection



Easy segmentation in high-dimensional space

Linear SVM (1)

- How do we split the red and blue datasets by a straight line?

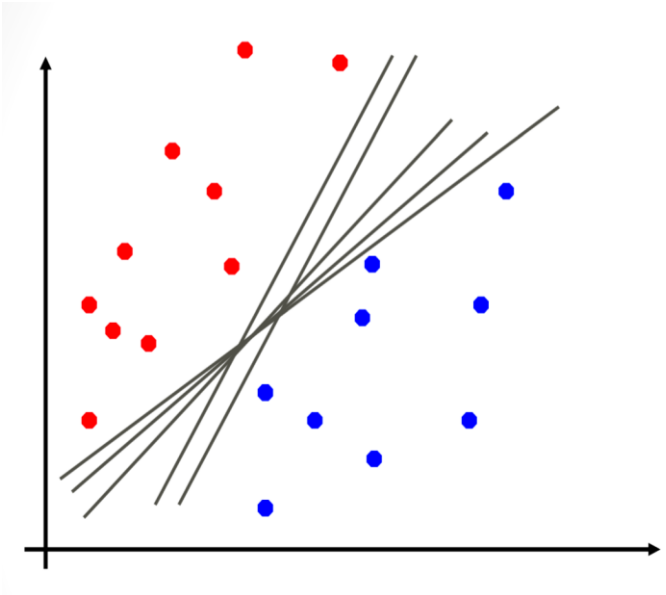


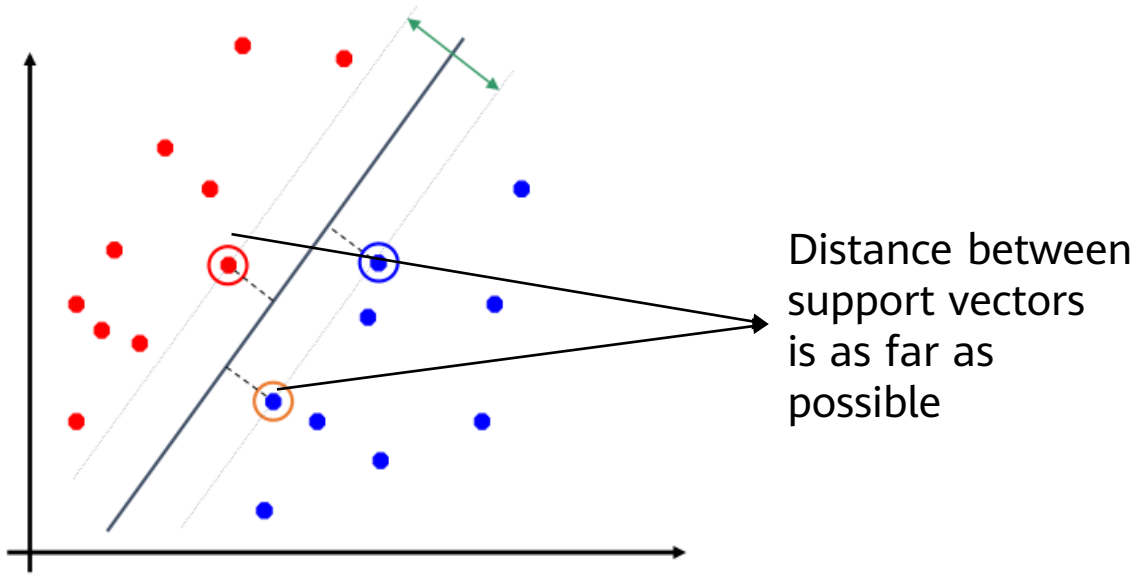
With binary classification
Two-dimensional dataset

Both the left and right methods can be used to divide datasets. Which of them is correct?

Linear SVM (2)

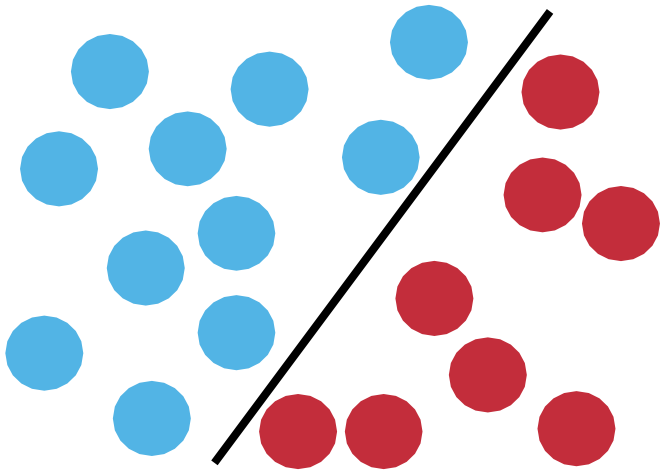
- Straight lines are used to divide data into different classes. Actually, we can use multiple straight lines to divide data. The core idea of the SVM is to find a straight line and keep the point close to the straight line as **far** as possible from the straight line. This can enable strong generalization capability of the model. These points are called **support vectors**.
- In two-dimensional space, we use straight lines for segmentation. In high-dimensional space, we use **hyperplanes** for segmentation.



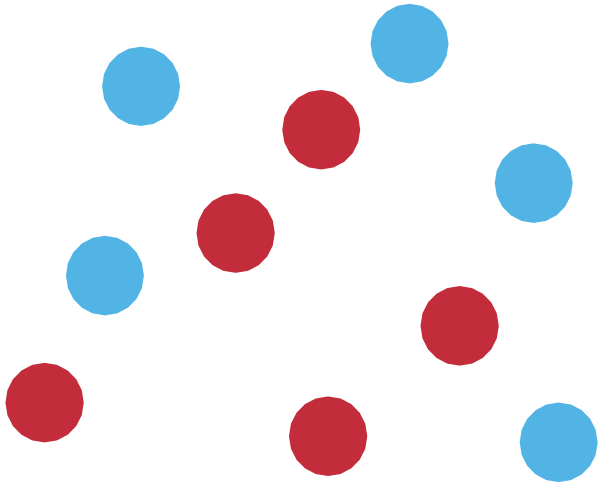


Nonlinear SVM (1)

- How do we classify a nonlinear separable dataset?



Linear SVM can function well for linear separable datasets.

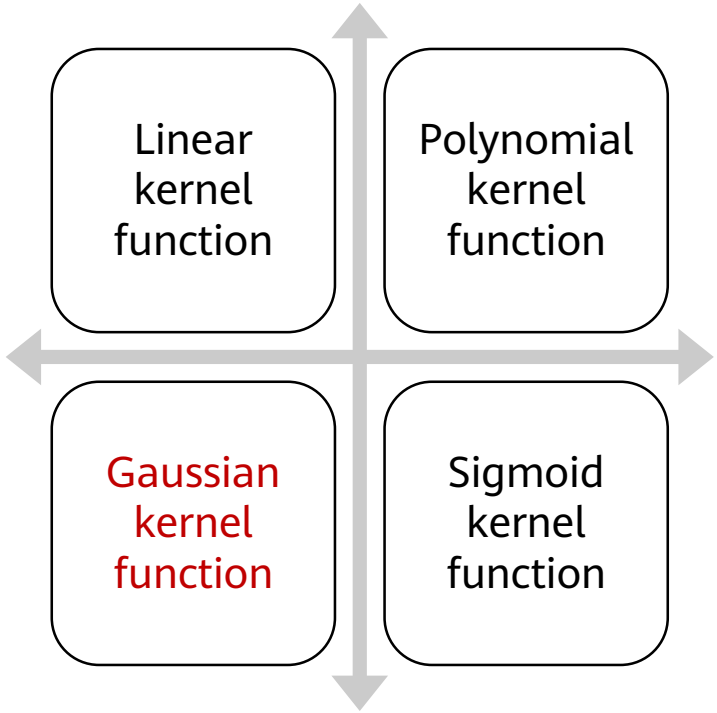


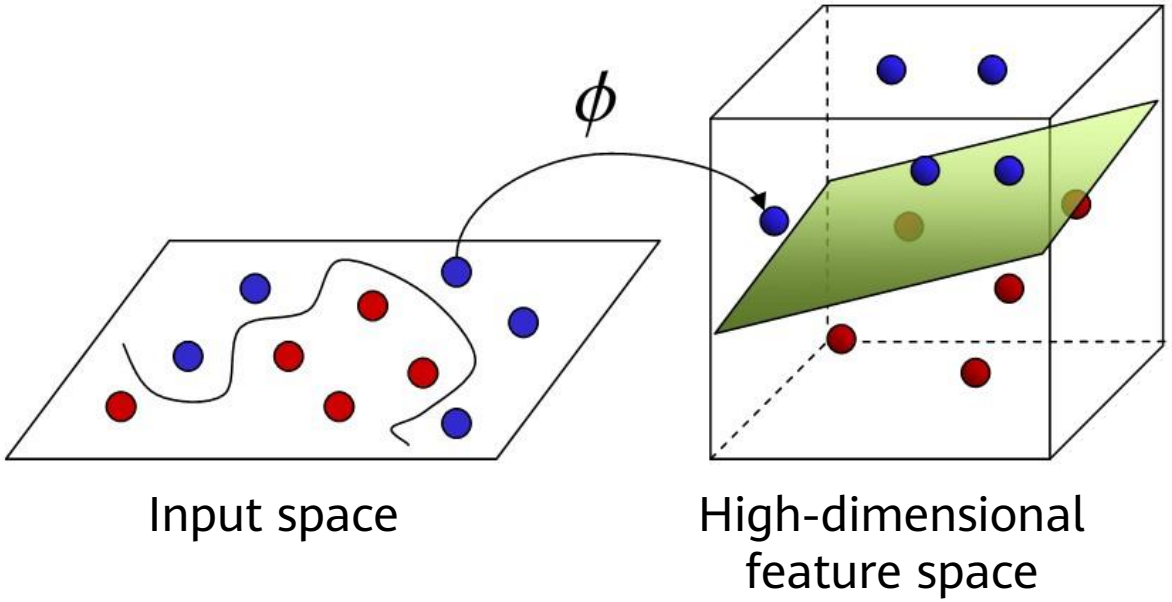
Nonlinear datasets cannot be split with straight lines.

Nonlinear SVM (2)

- Kernel functions are used to construct nonlinear SVMs.
- Kernel functions allow algorithms to fit the largest hyperplane in a transformed high-dimensional feature space.

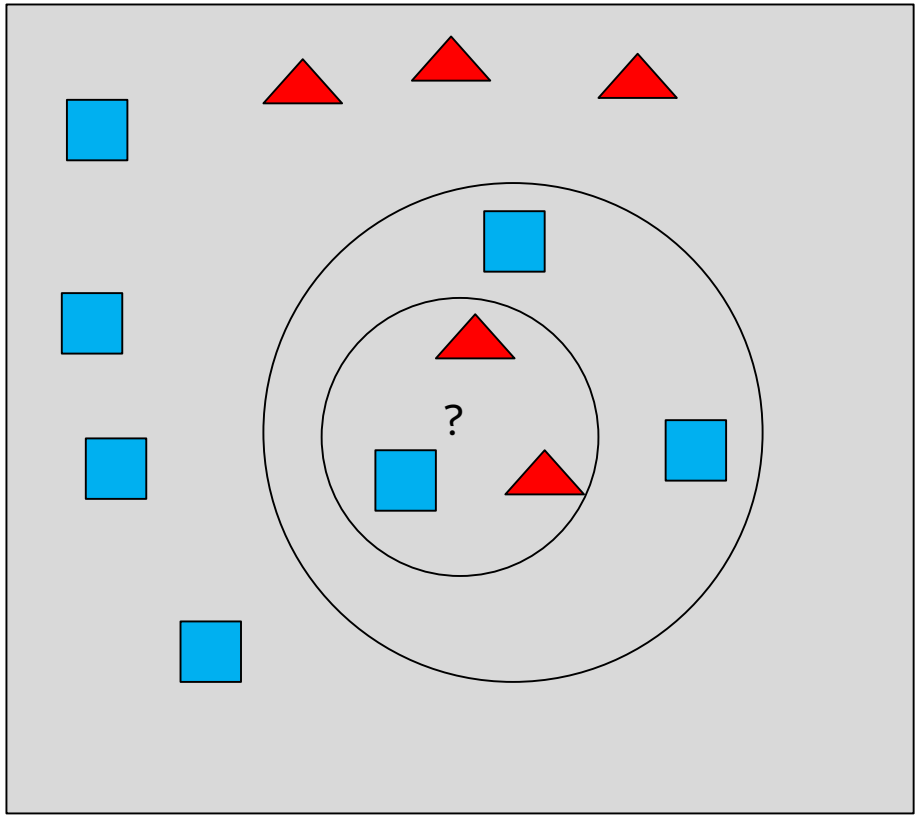
Common kernel functions





KNN Algorithm (1)

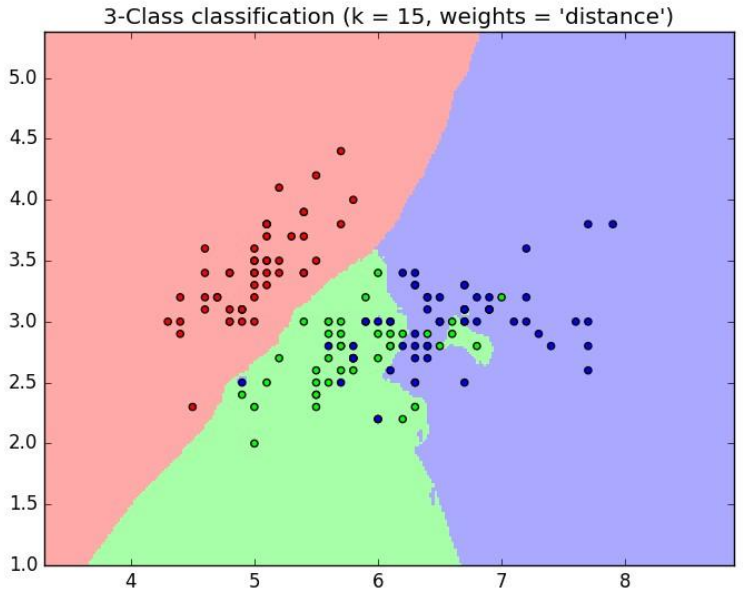
- The KNN classification algorithm is a theoretically mature method and one of the simplest machine learning algorithms. According to this method, if the majority of k samples most similar to one sample (nearest neighbors in the eigenspace) belong to a specific category, this sample also belongs to this category.



The target category of point ? varies with the number of the most adjacent nodes.

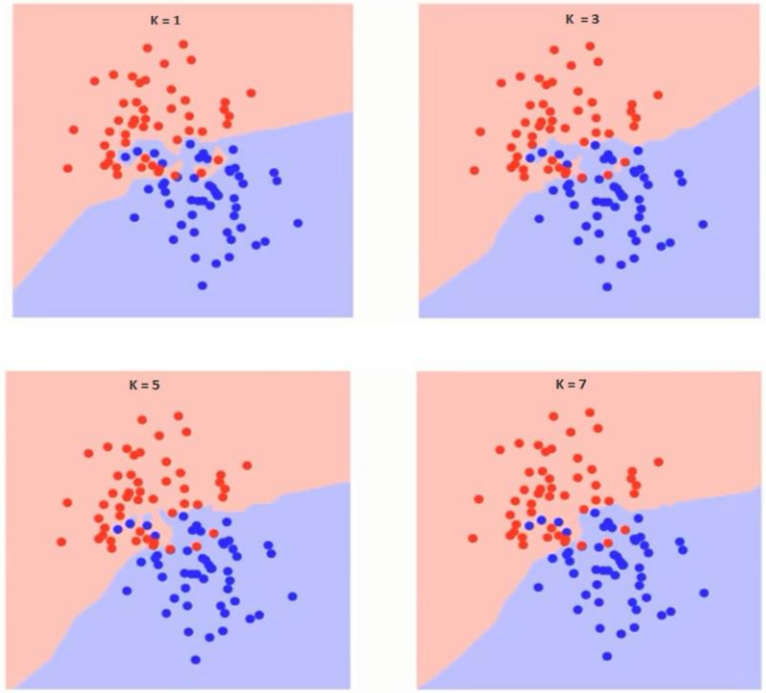
KNN Algorithm (2)

- As the prediction result is determined based on the number and weights of neighbors in the training set, the KNN algorithm has a simple logic.
- KNN is a non-parametric method which is usually used in datasets with irregular decision boundaries.
  - The KNN algorithm generally adopts the majority voting method for classification prediction and the average value method for regression prediction.
- KNN requires a huge number of computations.



KNN Algorithm (3)

- Generally, a larger k value reduces the impact of noise on classification, but obfuscates the boundary between classes.
  - A larger k value means a higher probability of underfitting because the segmentation is too rough. A smaller k value means a higher probability of overfitting because the segmentation is too refined.



- The boundary becomes smoother as the value of k increases.
- As the k value increases to infinity, all data points will eventually become all blue or all red.

Naive Bayes (1)

- Naive Bayes algorithm: a simple multi-class classification algorithm based on the Bayes theorem. It assumes that features are independent of each other. For a given sample feature X, the probability that a sample belongs to a category H is:

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

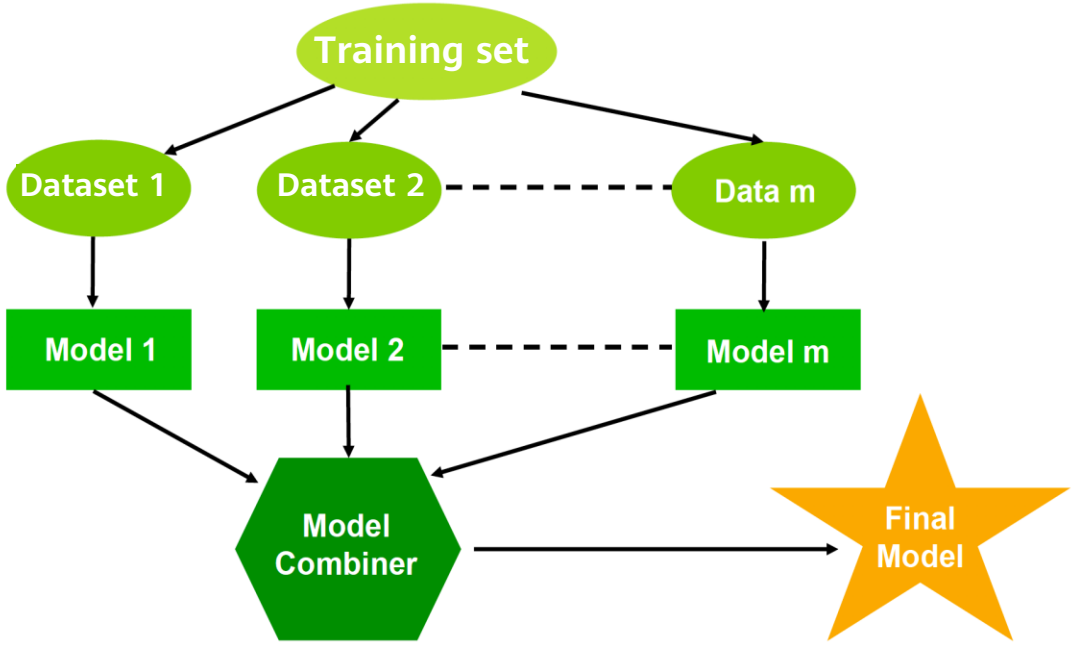
  - $X_1, \ldots, X_n$ are data features, which are usually described by measurement values of m attribute sets.
    - For example, the color feature may have three attributes: red, yellow, and blue.
  - C_k indicates that the data belongs to a specific category C
  - $P(C_k|X_1, \ldots, X_n)$ is a posterior probability, or a posterior probability of under condition C_k.
  - $P(C_k)$ is a prior probability that is independent of $X_1, \ldots, X_n$
  - $P(X_1, \ldots, X_n)$ is the priori probability of X.

Naive Bayes (2)

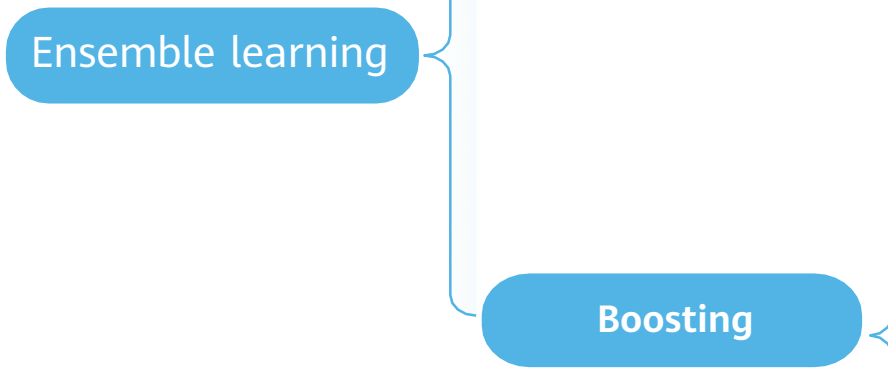
- Independent assumption of features.
  - For example, if a fruit is red, round, and about 10 cm (3.94 in.) in diameter, it can be considered an apple.
  - A Naive Bayes classifier considers that each feature independently contributes to the probability that the fruit is an apple, regardless of any possible correlation between the color, roundness, and diameter.

Ensemble Learning

- Ensemble learning is a machine learning paradigm in which multiple learners are trained and combined to solve the same problem. When multiple learners are used, the integrated generalization capability can be much stronger than that of a single learner.
- If you ask a complex question to thousands of people at random and then summarize their answers, the summarized answer is better than an expert's answer in most cases. This is the wisdom of the masses.



Classification of Ensemble Learning



Bagging (Random Forest)

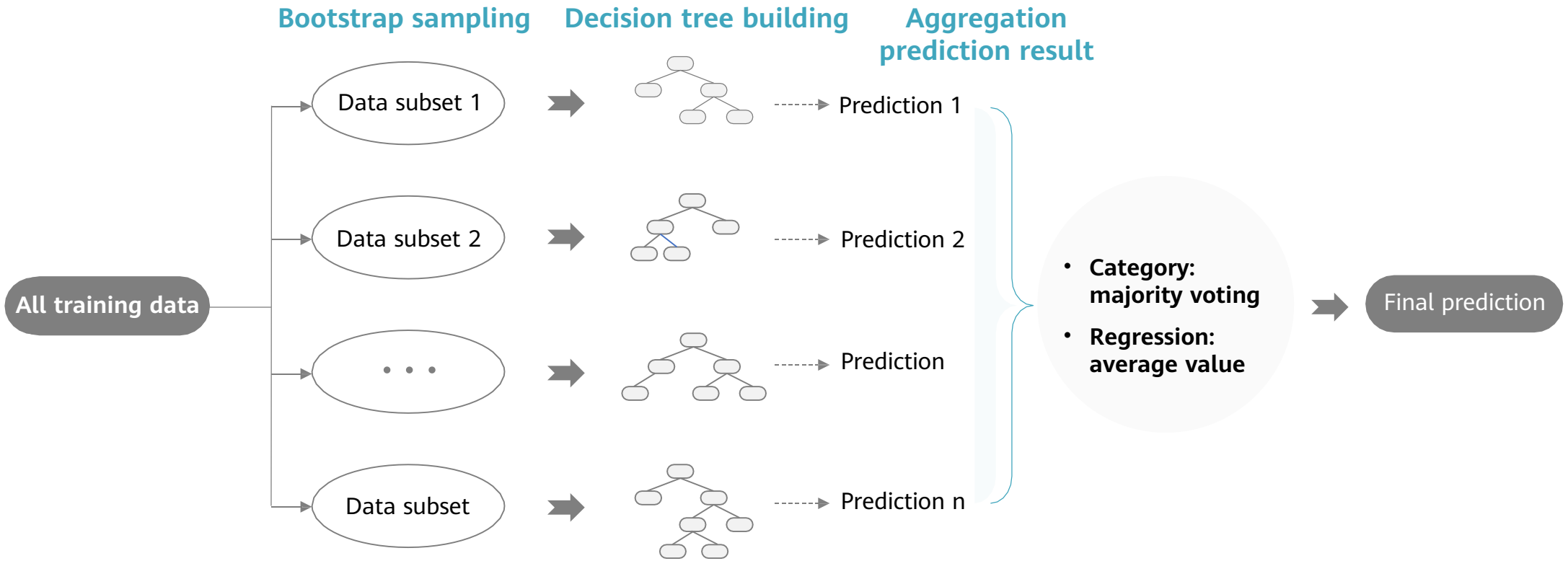
- Independently builds several basic learners and then averages their predictions.
- On average, a composite learner is usually better than a single-base learner because of a smaller variance.

Boosting (Adaboost, GBDT, and XGboost)

Constructs basic learners in sequence to gradually reduce the bias of a composite learner. The composite learner can fit data well, which may also cause overfitting.

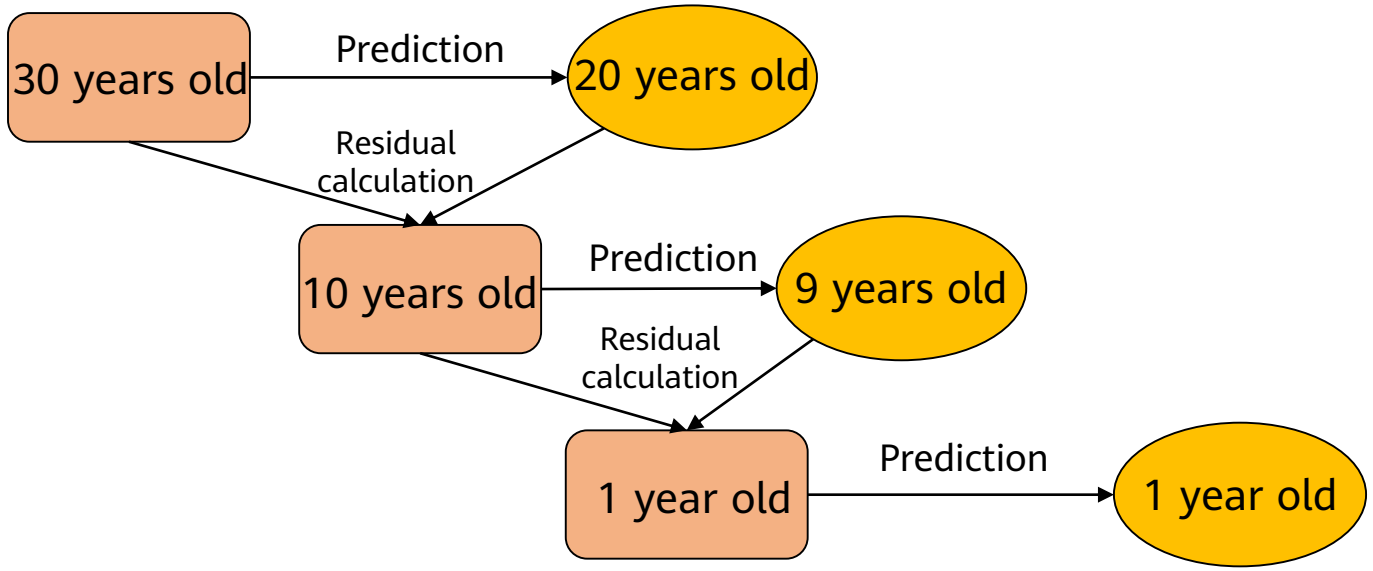
Ensemble Methods in Machine Learning (1)

- Random forest = Bagging + CART decision tree
- Random forests build multiple decision trees and merge them together to make predictions more accurate and stable.
  - Random forests can be used for classification and regression problems.



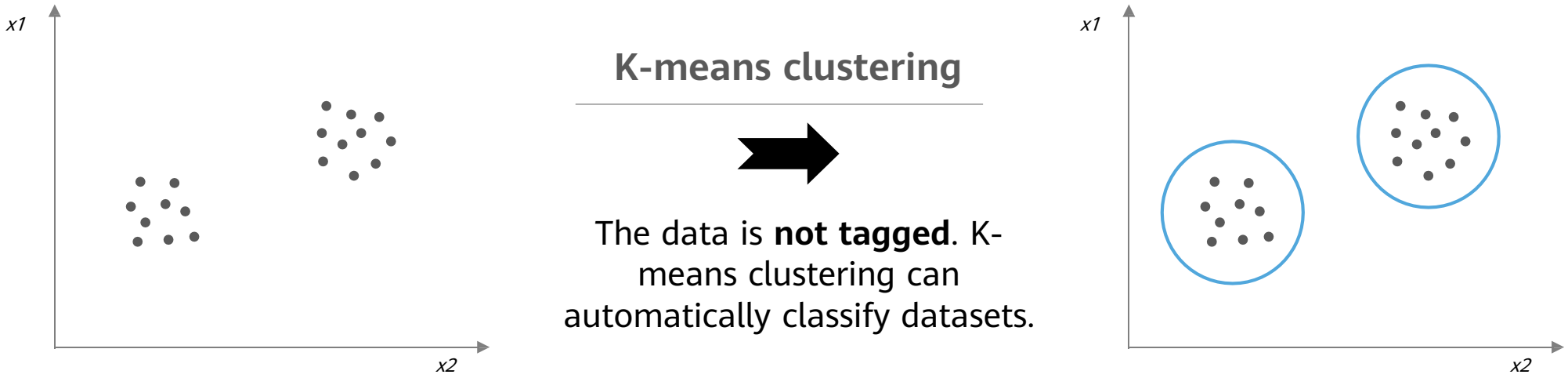
Ensemble Methods in Machine Learning (2)

- GBDT is a type of boosting algorithm.
- For an aggregative mode, the sum of the results of all the basic learners equals the predicted value. In essence, the residual of the error function to the predicted value is fit by the next basic learner. (The residual is the error between the predicted value and the actual value.)
- During model training, GBDT requires that the sample loss for model prediction be as small as possible.



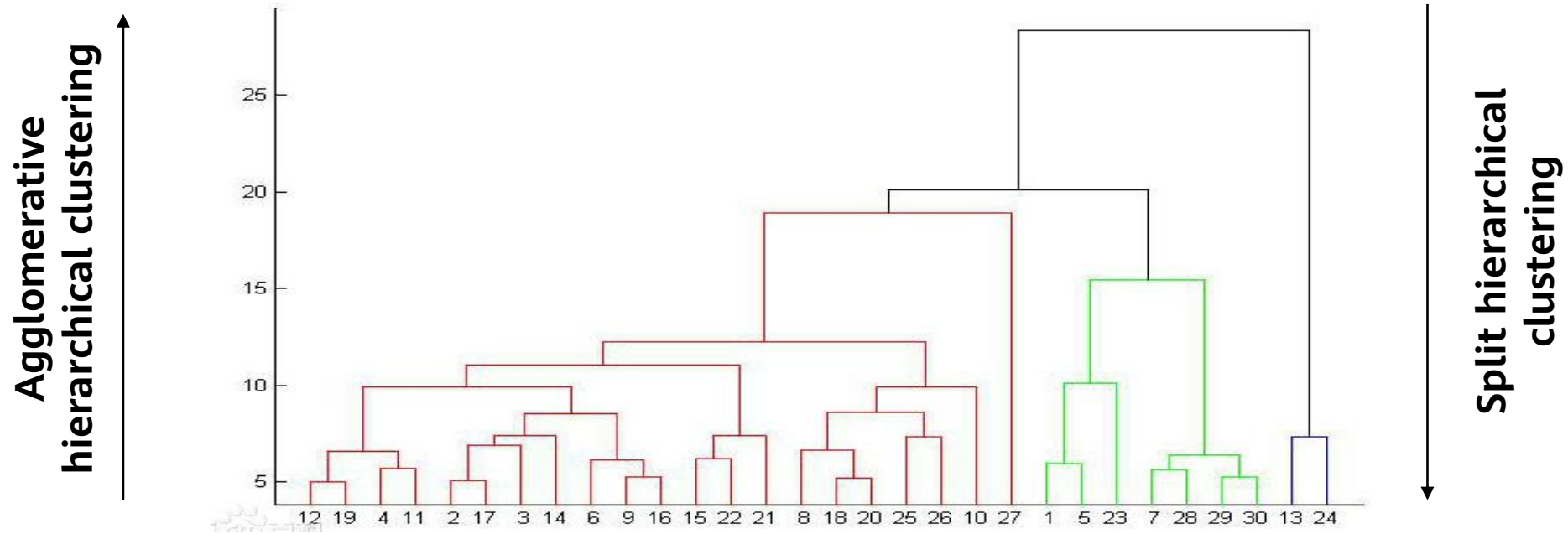
Unsupervised Learning - K-means

- K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.
- For the k-means algorithm, specify the final number of clusters (k). Then, divide n data objects into k clusters. The clusters obtained meet the following conditions: (1) Objects in the same cluster are highly similar. (2) The similarity of objects in different clusters is small.



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 graph. The root is the unique cluster of all samples, and the leaves are the cluster of only a sample.



Contents

Comprehensive Case

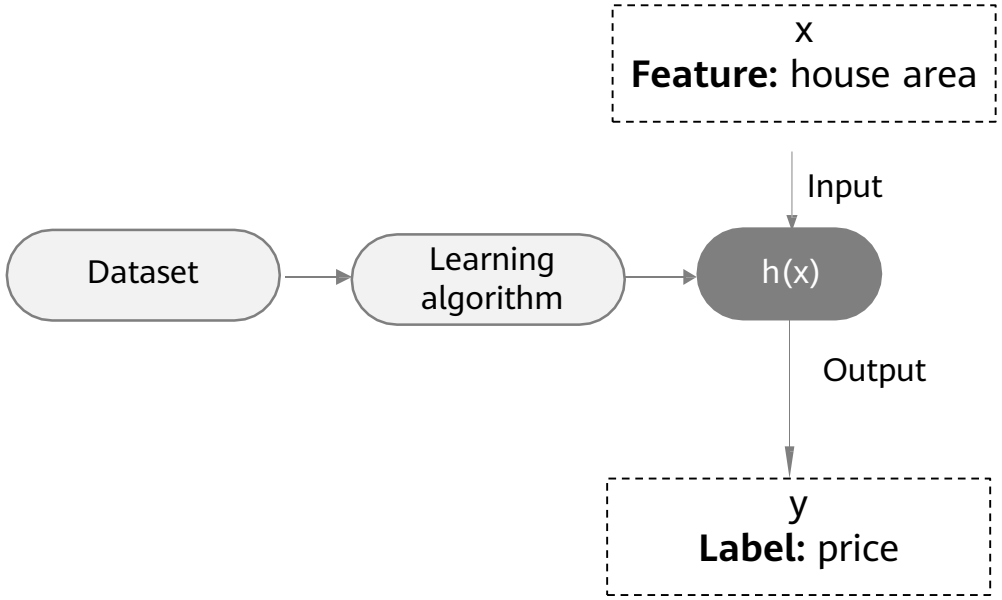
- Assume that there is a dataset containing the house areas and prices of 21,613 housing units sold in a city. Based on this data, we can predict the prices of other houses in the city.

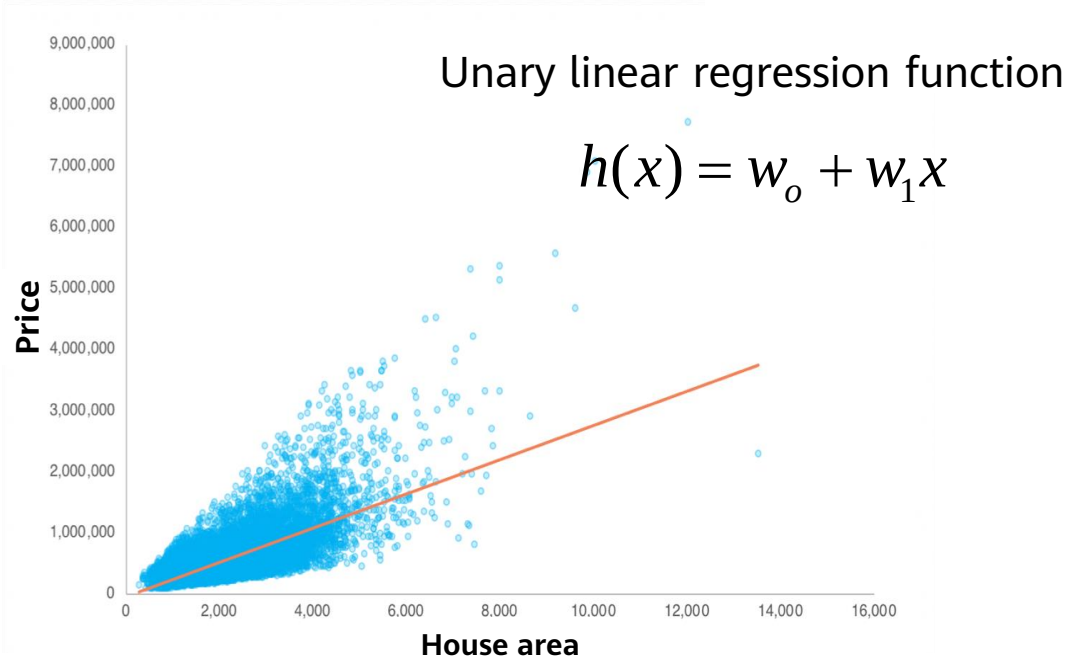
House Area	Price
1,180	221,900
2,570	538,000
770	180,000
1,960	604,000
1,680	510,000
5,420	1,225,000
1,715	257,500
1,060	291,850
1,160	468,000
1,430	310,000
1,370	400,000
1,810	530,000
...	...

Dataset

Problem Analysis

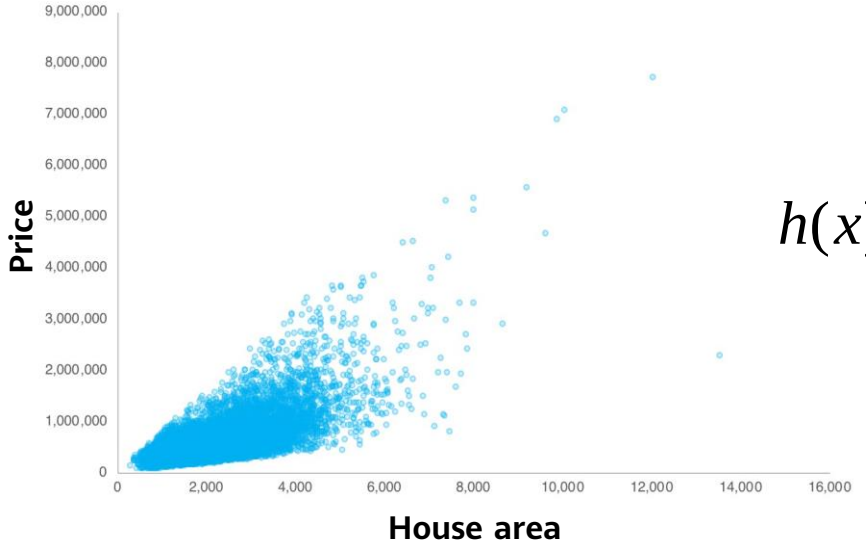
- This case contains a large amount of data, including input x (house area), and output y (price), which is a continuous value. We can use **regression** of **supervised learning**. Draw a scatter chart based on the data and use **linear regression**.
- Our goal is to build a model function h(x) that infinitely approximates the function that expresses true distribution of the dataset.
- Then, use the model to predict unknown price data.





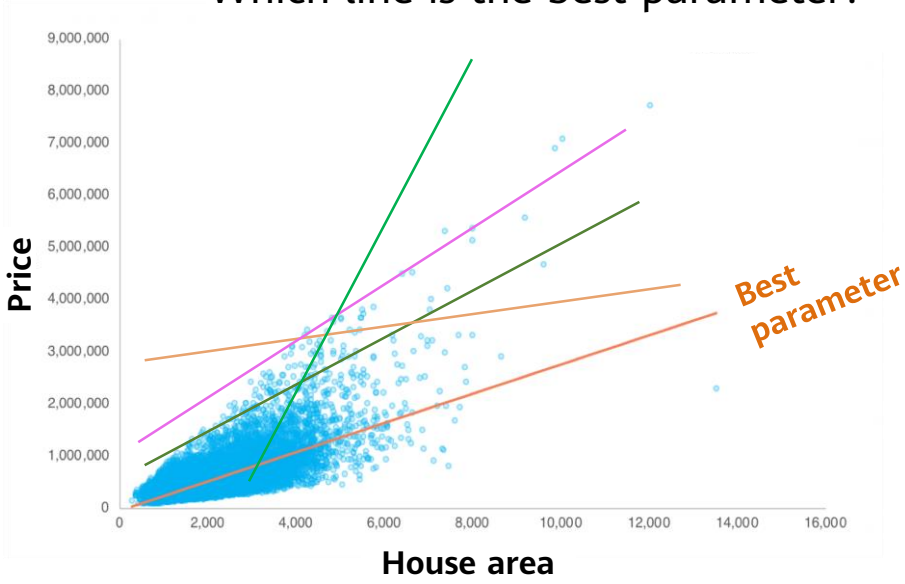
Goal of Linear Regression

- Linear regression aims to find a straight line that best fits the dataset.
- Linear regression is a parameter-based model. Here, we need learning parameters w_0 and w_1. When these two parameters are found, the best model appears.



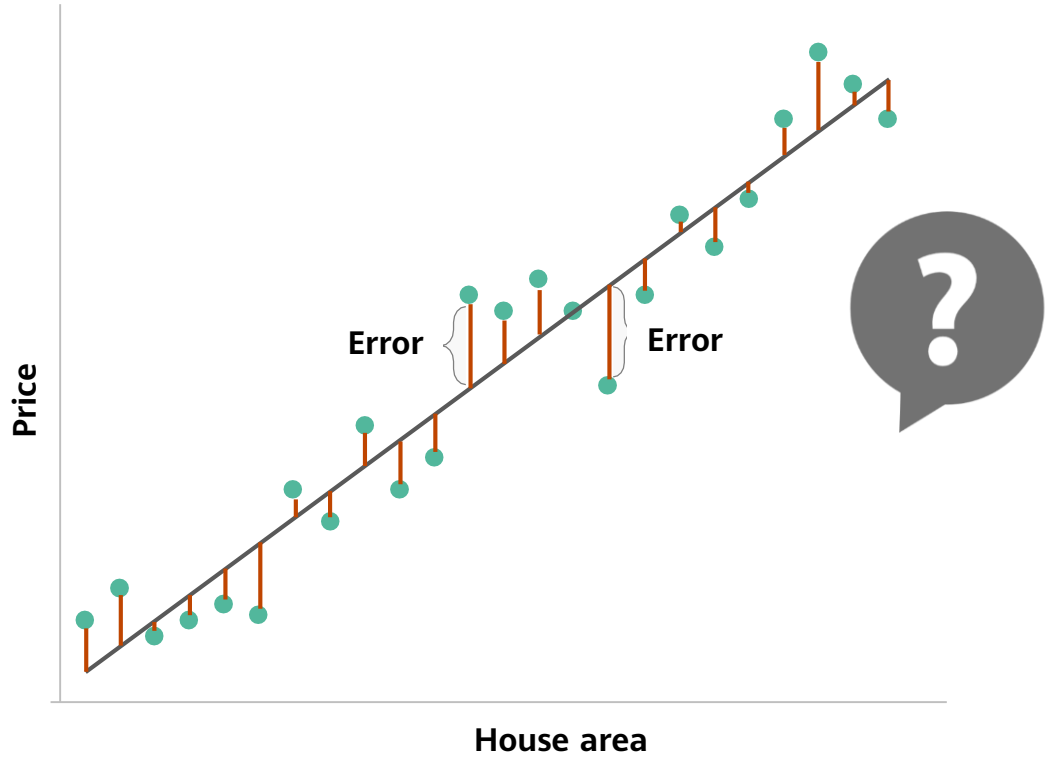
$$h(x) = w_o + w_1 x$$

Which line is the best parameter?



Loss Function of Linear Regression

- To find the optimal parameter, construct a loss function and find the parameter values when the loss function becomes the minimum.



Loss function of linear regression:

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

Goal:

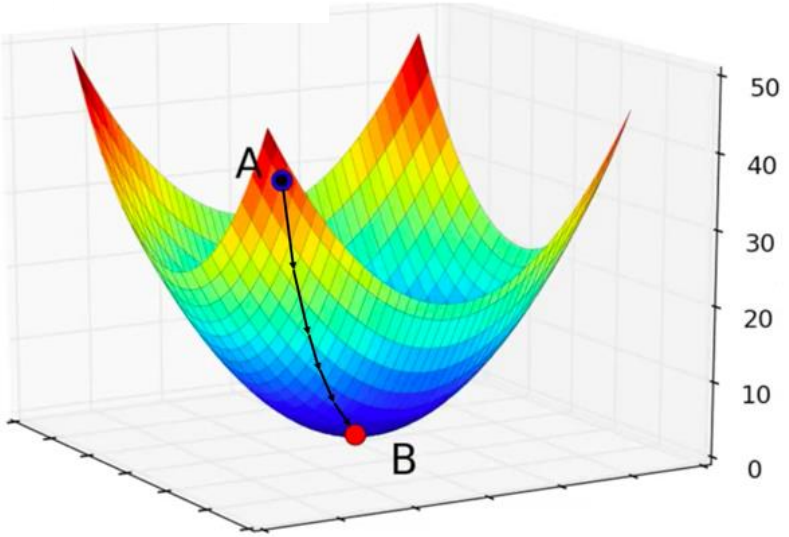
$$\arg\min_{w} J(w)=\frac{1}{2m}\sum\left(h(x)-y\right)^2$$

- where, *m* indicates the number of samples,
- *h(x)* indicates the predicted value, and *y* indicates the actual value.

Gradient Descent Method

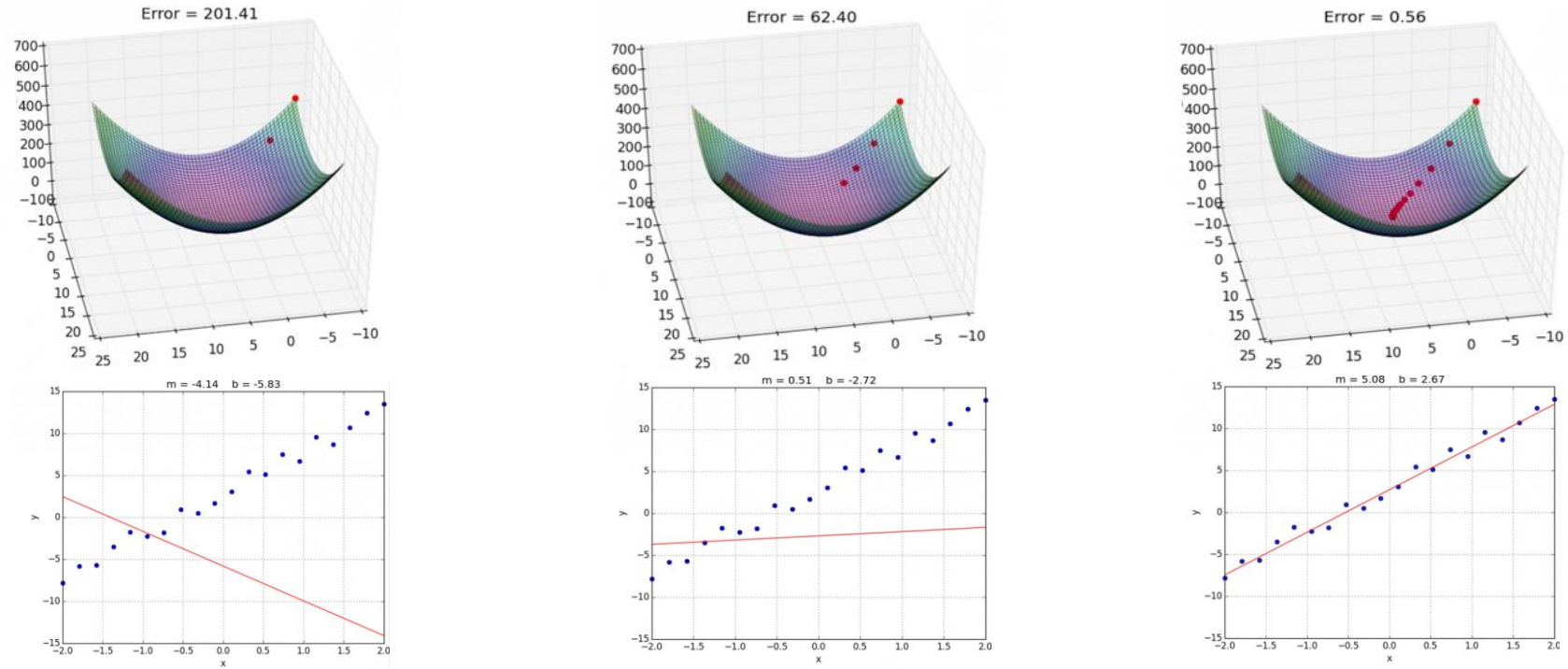
- The gradient descent algorithm finds the minimum value of a function through iteration.
- It aims to randomize an initial point on the loss function, and then find the global minimum value of the loss function based on the negative gradient direction. Such parameter value is the optimal parameter value.
  - **Point A:** the position of w_0 and w_1 after random initialization. w_0 and w_1 are the required **parameters.**
  - **A-B connection line:** a track formed based on descents in a negative gradient direction. Upon each descent, values w_0 and w_1 **change**, and the regression line also changes.
  - **Point B:** global minimum value of the loss function. Final values of w_0 and w_1 are also found.

Cost surface



Iteration Example

- The following is an example of a gradient descent iteration. We can see that as red points on the loss function surface gradually approach a lowest point, fitting of the linear regression red line with data becomes better and better. At this time, we can get the best parameters.

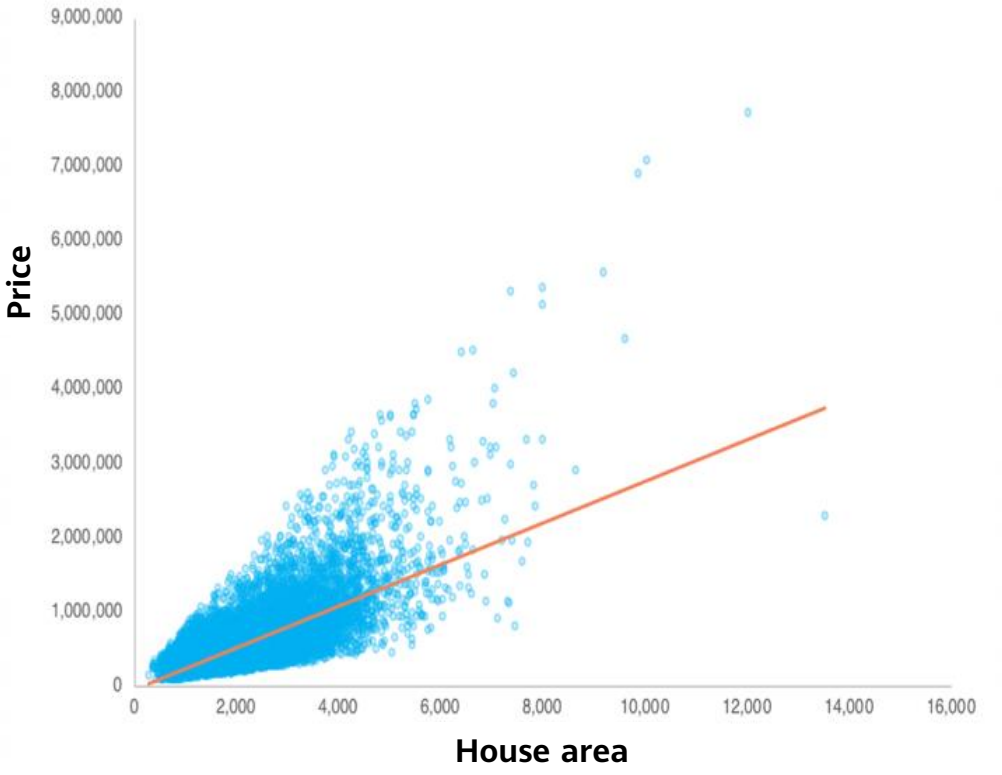


Model Debugging and Application

- After the model is trained, test it with the test set to ensure the generalization capability.
- If overfitting occurs, use Lasso regression or Ridge regression with regularization terms and tune the hyperparameters.
- If underfitting occurs, use a more complex regression model, such as GBDT.
- Note:
  - For real data, pay attention to the functions of data cleansing and feature engineering.

The final model result is as follows:

$$h(x) = 280.62x - 43581$$



Summary

- First, this course describes the definition and classification of machine learning, as well as problems machine learning solves. Then, it introduces key knowledge points of machine learning, including the overall procedure (data collection, data cleansing, feature extraction, model training, model training and evaluation, and model deployment), common algorithms (linear regression, logistic regression, decision tree, SVM, naive Bayes, KNN, ensemble learning, K-means, etc.), gradient descent algorithm, parameters and hyper-parameters.
- Finally, a complete machine learning process is presented by a case of using linear regression to predict house prices.