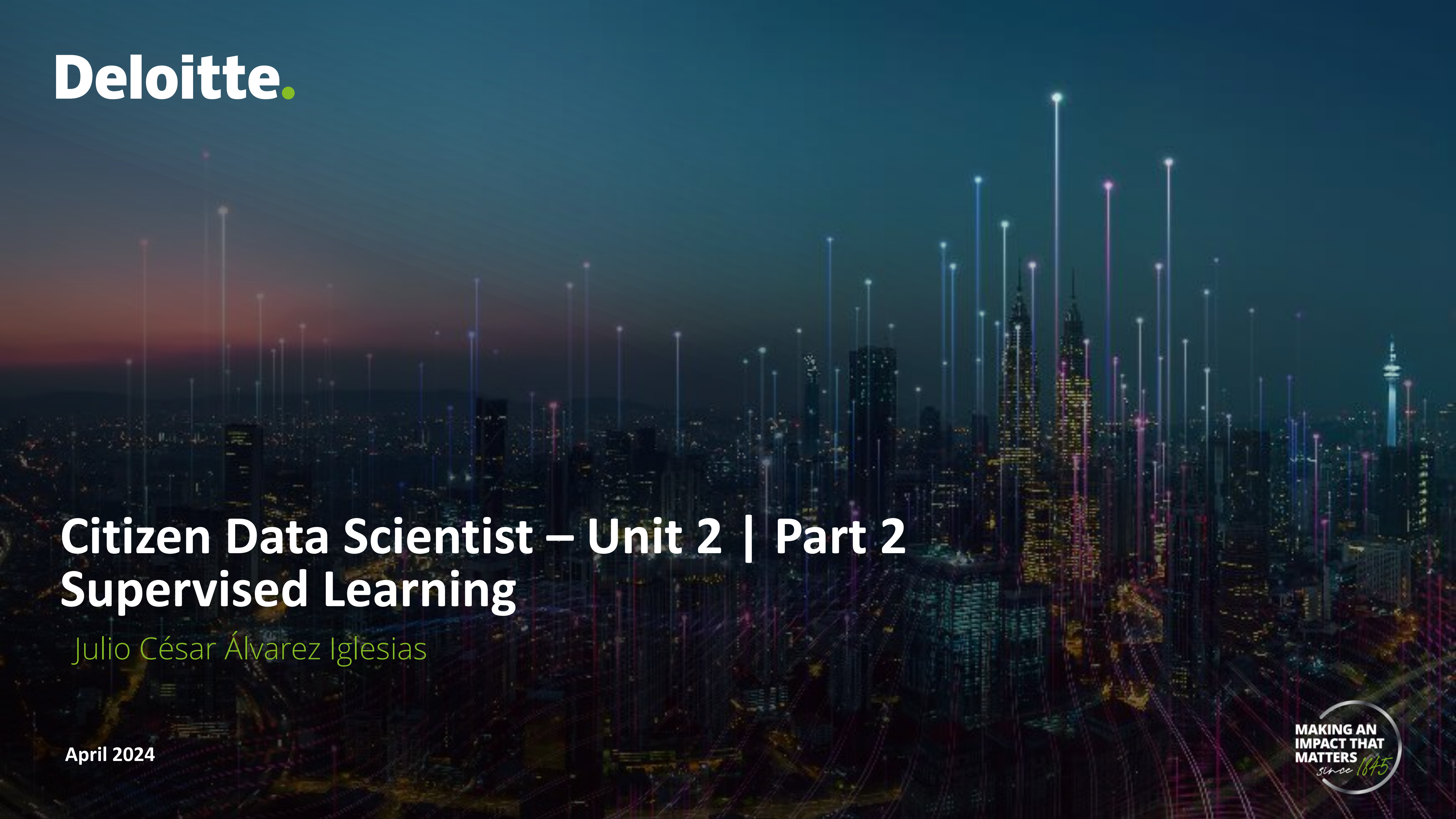


Citizen Data Scientist – Unit 2 | Part 2
Supervised Learning

Julio César Álvarez Iglesias

April 2024

MAKING AN IMPACT THAT MATTERS since 1845

What is Classification?

Classification is a type of supervised learning where the output (label, target or class) is categorical (not continuous). This is done by building a model based on one or more numerical and/or categorical variables (predictors, attributes or features).

So, in classification algorithm, a discrete output function(y) is mapped to input variable(X):

$$y = f(X),$$

where y = categorical output.

©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

What is Classification?

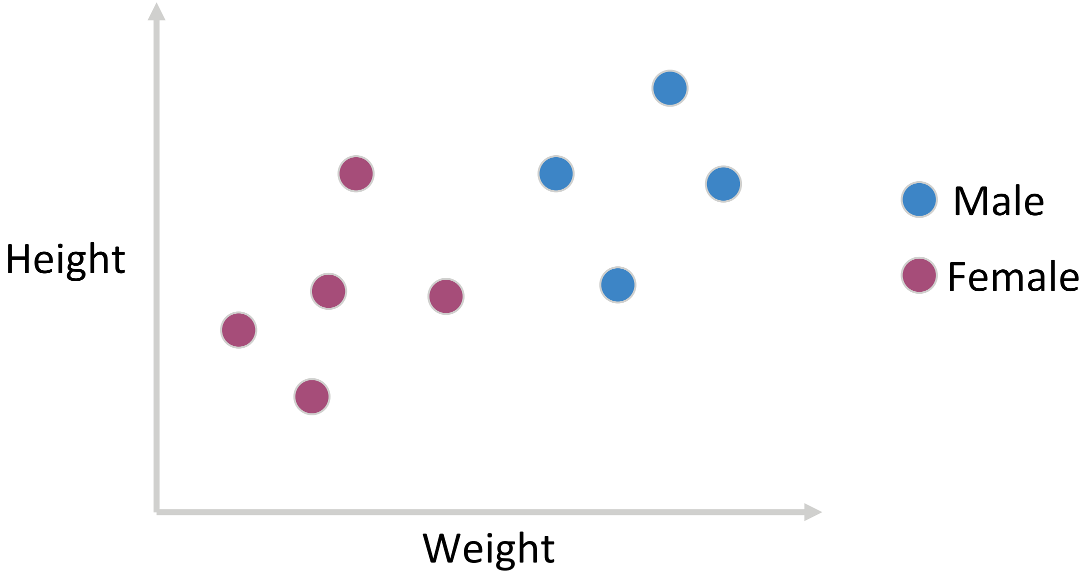
Classification is a type of supervised learning where the output (label, target or class) is categorical (not continuous). This is done by building a model based on one or more numerical and/or categorical variables (predictors, attributes or features).

So, in classification algorithm, a discrete output function(y) is mapped to input variable(X):

$$y = f(X),$$

where y = categorical output.

Classification algorithms can be better understood with a diagram. In this diagram, there are two classes: 'Male' and 'Female'. Each class has features (Height and Weight) that are distinct from the other yet exhibit similarities within the same class.



©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

What is Classification?

Classification is a type of supervised learning where the output (label, target or class) is categorical (not continuous). This is done by building a model based on one or more numerical and/or categorical variables (predictors, attributes or features).

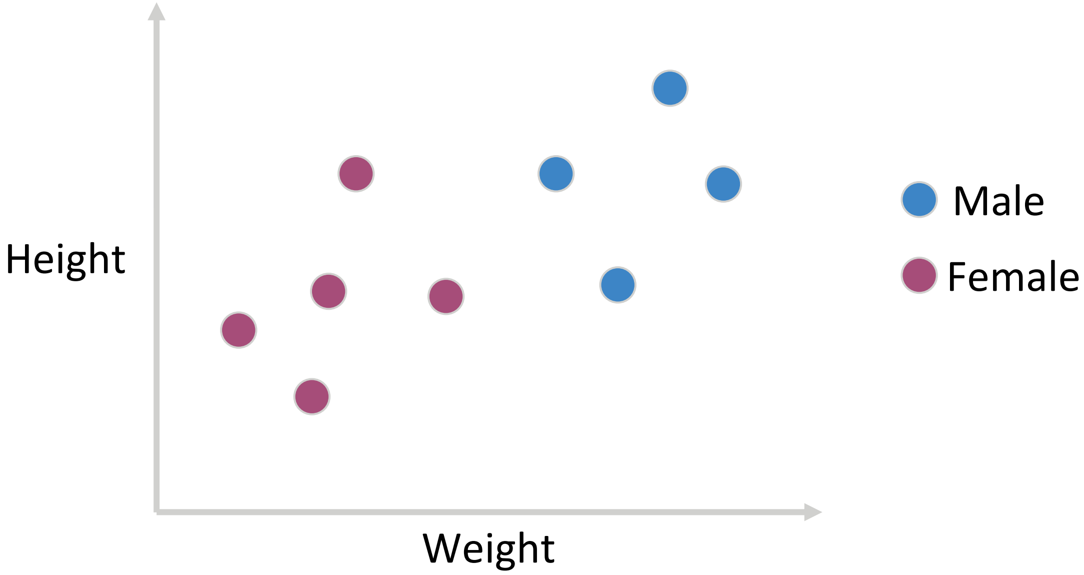
So, in classification algorithm, a discrete output function(y) is mapped to input variable(X):

$$y = f(X),$$

where y = categorical output.

Classification algorithms can be better understood with a diagram. In this diagram, there are two classes: 'Male' and 'Female'. Each class has features (Height and Weight) that are distinct from the other yet exhibit similarities within the same class.

How do we differentiate 'Male' and 'Female' based in this diagram?



©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

What is Classification?

Classification is a type of supervised learning where the output (label, target or class) is categorical (not continuous). This is done by building a model based on one or more numerical and/or categorical variables (predictors, attributes or features).

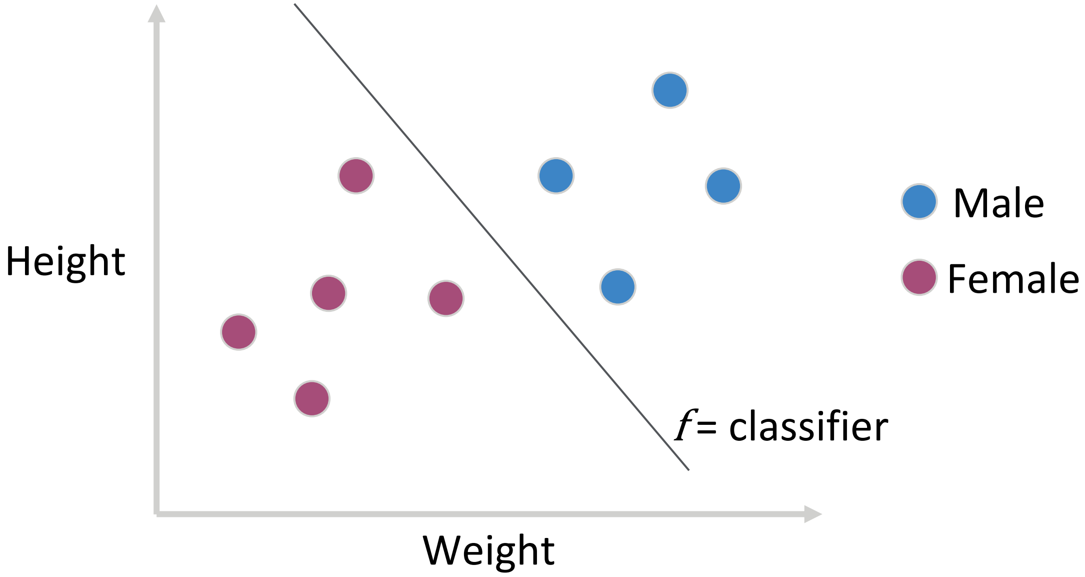
So, in classification algorithm, a discrete output function(y) is mapped to input variable(X):

$$y = f(X),$$

where y = categorical output.

Classification algorithms can be better understood with a diagram. In this diagram, there are two classes: 'Male' and 'Female'. Each class has features (Height and Weight) that are distinct from the other yet exhibit similarities within the same class.

How do we differentiate 'Male' and 'Female' based in this diagram?



©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

What is Classification?

Classification is a type of supervised learning where the output (label, target or class) is categorical (not continuous). This is done by building a model based on one or more numerical and/or categorical variables (predictors, attributes or features).

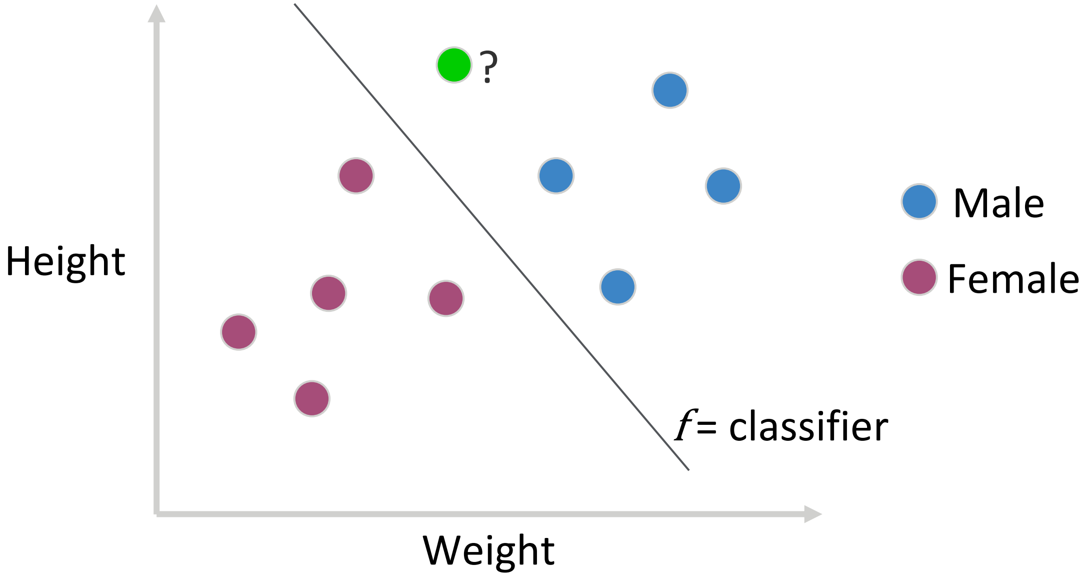
So, in classification algorithm, a discrete output function(y) is mapped to input variable(X):

$$y = f(X),$$

where y = categorical output.

Classification algorithms can be better understood with a diagram. In this diagram, there are two classes: 'Male' and 'Female'. Each class has features (Height and Weight) that are distinct from the other yet exhibit similarities within the same class.

What would happen if I wanted to classify a new person?



©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

What is Classification?

Classification is a type of supervised learning where the output (label, target or class) is categorical (not continuous). This is done by building a model based on one or more numerical and/or categorical variables (predictors, attributes or features).

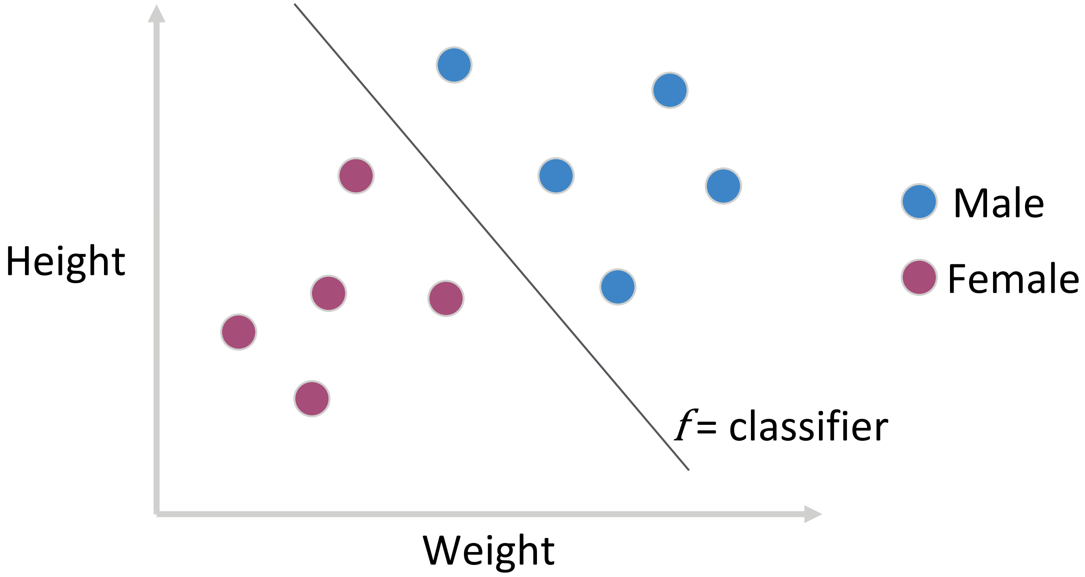
So, in classification algorithm, a discrete output function(y) is mapped to input variable(X):

$$y = f(X),$$

where y = categorical output.

Classification algorithms can be better understood with a diagram. In this diagram, there are two classes: 'Male' and 'Female'. Each class has features (Height and Weight) that are distinct from the other yet exhibit similarities within the same class.

What would happen if I wanted to classify a new person?

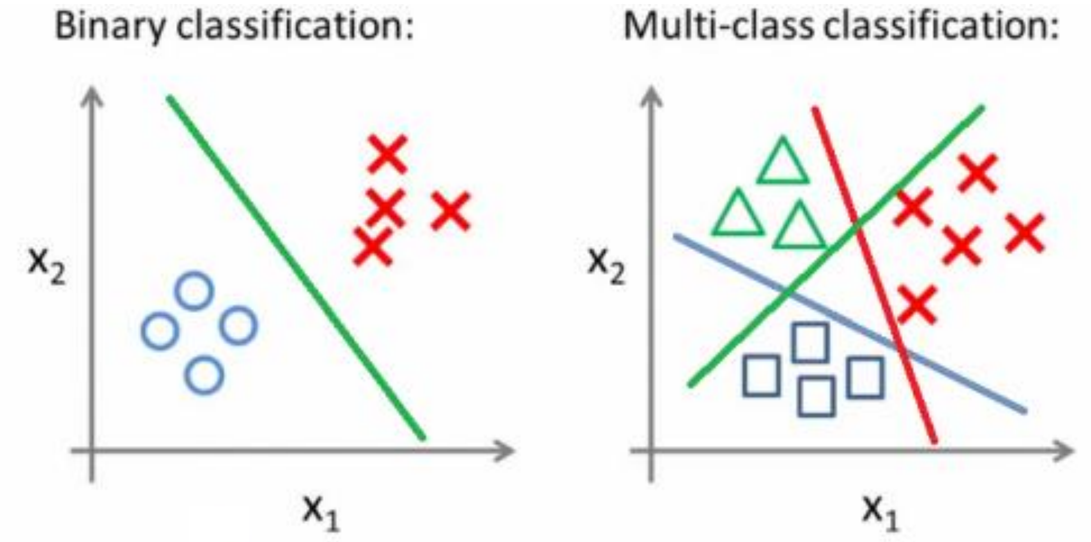


©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

Binary vs Multi-class Classification

There are several types of Classification Tasks in Machine Learning, but the two most important are:

1. Binary Classification: This involves distinguishing between exactly two classes. The output variable has two possible outcomes.
   - Example: 'YES' or 'NO', 'MALE' or 'FEMALE', 'SPAM' or 'NOT SPAM', 'CAT' or 'DOG', etc.
2. Multi-class Classification: This involves classifying instances into one of three or more classes. Each instance is assigned to one and only one class.
   - Example: Recognizing the type of fruit in an image (e.g., apples, oranges, bananas).



©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

Classification Algorithms

Given the extensive study of classification in statistics, numerous classification methods are available, tailored to specific datasets. Below are three principal machine learning algorithms commonly used in this domain:

1. Naive Bayes: A straightforward probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes determines whether a data point falls into a particular category. It can be used to classify phrases or words in text analysis as either falling within a predetermined classification or not.
2. K-Nearest Neighbors (K-NN): It calculates the likelihood that a data point will join the groups based on which group the data points closest to it are a part of. When using k-NN for classification, you determine how to classify the data according to its nearest neighbor.
3. Support Vector Machine (SVM): Support Vector Machine is a popular supervised machine learning technique for classification and regression problems. It goes beyond X/Y prediction by using algorithms to classify and train the data according to polarity.

©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

Understanding Learners in Classification Problems

In the realm of machine learning, especially in classification problems, learners are categorized into two types based on their approach to training and prediction:

1. Lazy Learners: Lazy learners, as the name suggests, delay the modeling of their data until a prediction is requested, meaning they store the training dataset and do not generate a model until new data (test data) is provided. This approach involves minimal training time but requires more time during the prediction phase, as the learner must process the entire dataset to make decisions. Classification is performed based on the proximity to the most relevant data stored during training.
    - Example: The K-NN algorithm is a classic example of a lazy learner. It classifies new cases based on a similarity measure (e.g., distance functions).
2. Eager Learners: In contrast, eager learners build a classification model as soon as the training data is available. These learners analyze and generalize the training data before receiving any data for prediction, thereby spending more time on training but allowing for quicker responses when making predictions. They are optimized to infer the classification as fast as possible.
    - Example: Naive Bayes and SVM.

Both types of learners offer advantages in specific scenarios. Lazy learners can be preferable when the dataset is constantly changing, as they avoid the need for retraining, and are often simpler and more flexible in adapting to new data. On the other hand, eager learners are beneficial when the cost of computing is less critical during the training phase and when fast response times during the prediction phase are crucial.

©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

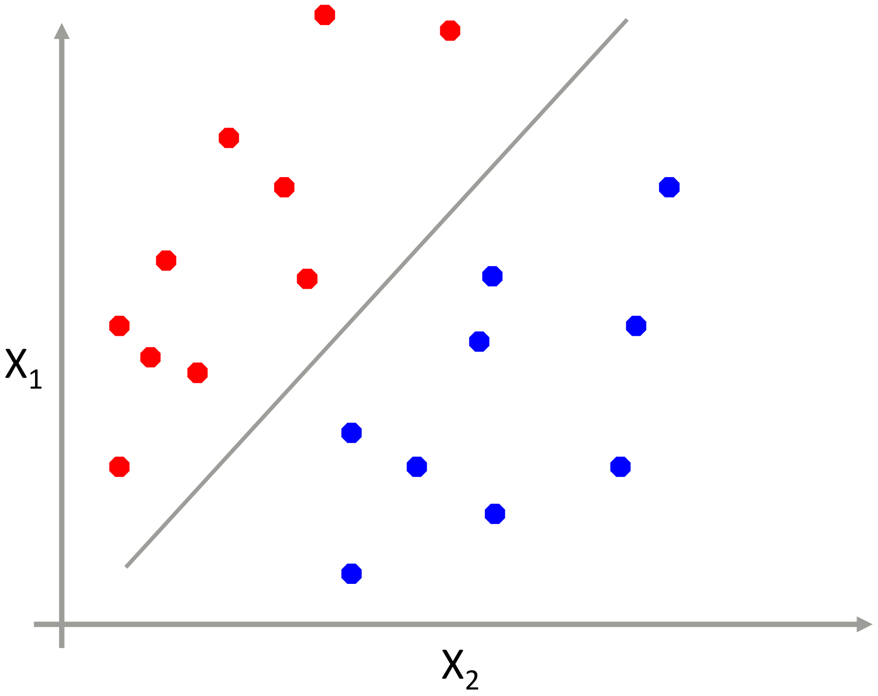
Types of ML Classification Algorithms

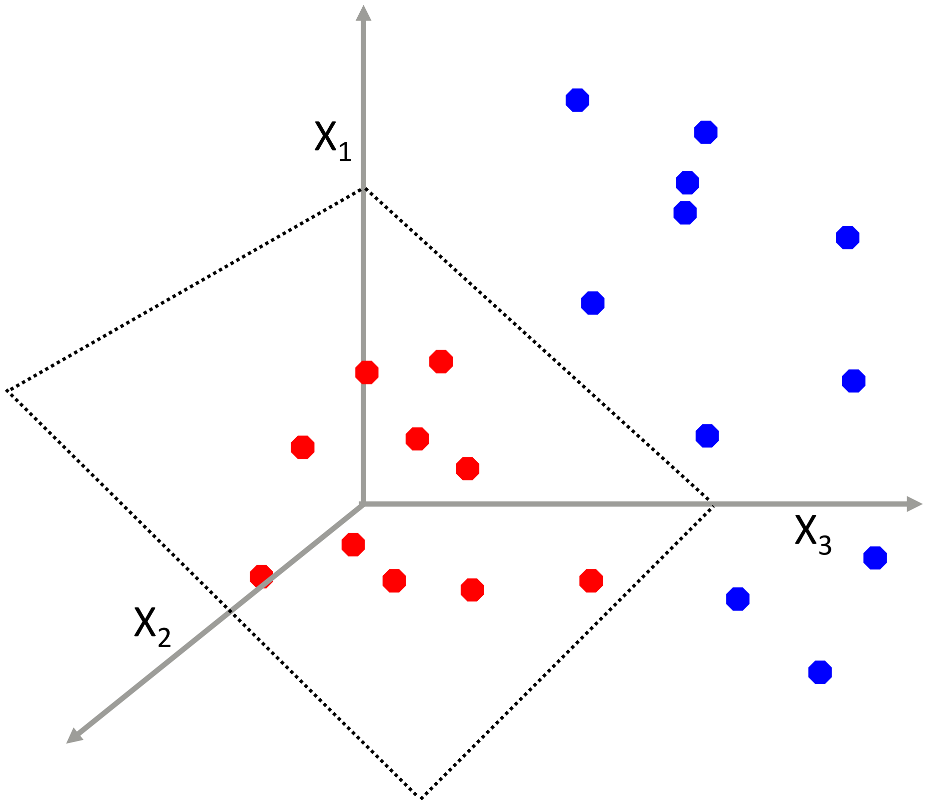
Classification algorithms can be mainly divided into two categories based on the nature of their decision boundaries:

1. Linear Models:
    - SVM: SVMs are typically used for linear classification, though they can also handle non-linear classification using kernel tricks..
2. Non-linear Models:
    - K-NN: This algorithm classifies data points based on the majority label among the nearest neighbors, which inherently makes it capable of handling complex, non-linear decision boundaries.
    - Naïve Bayes: Although often associated with linear boundaries due to its assumption of feature independence, Naïve Bayes can exhibit non-linear behavior depending on the distribution of the data and the specific variant of the algorithm used (e.g., Gaussian Naïve Bayes can model non-linear boundaries in certain distributions).

©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

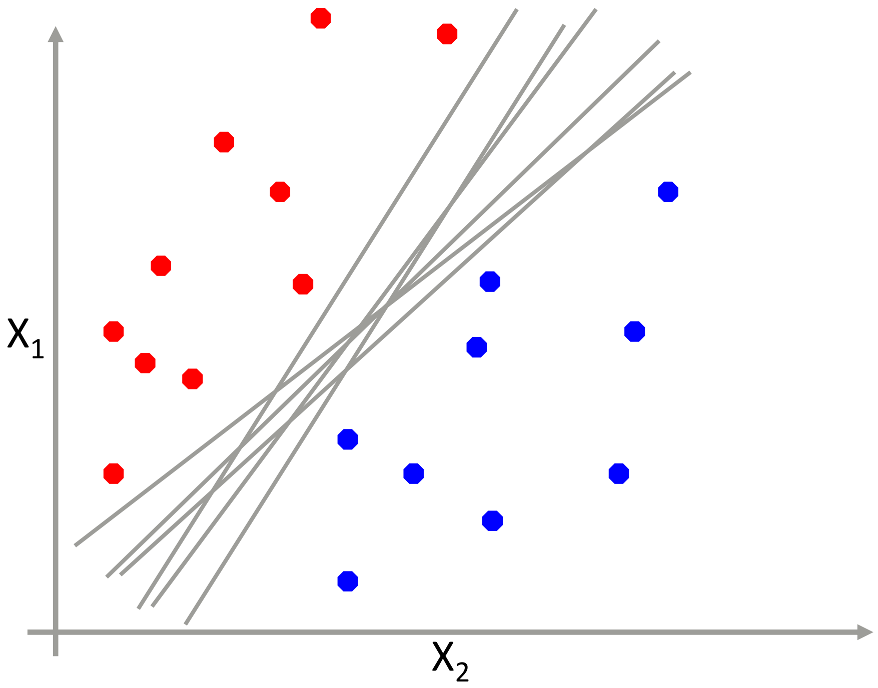
Linear Decision Boundary





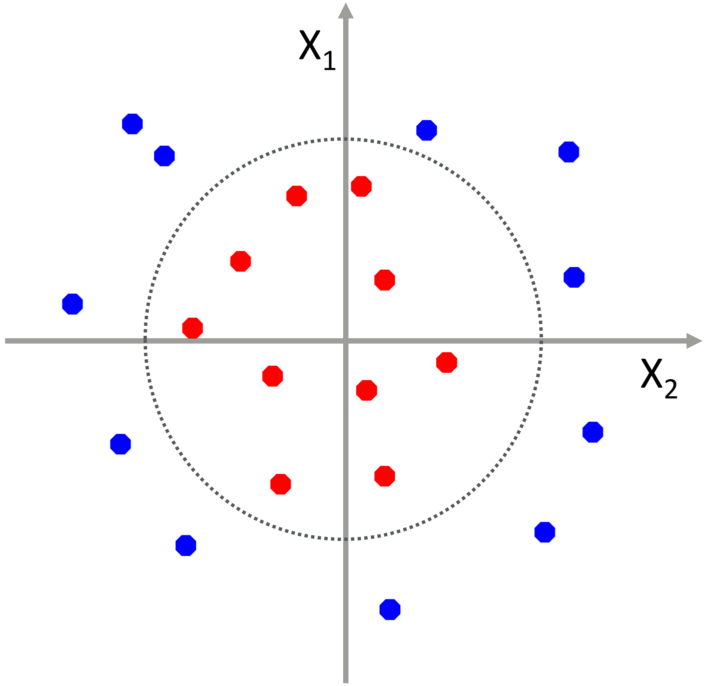
©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

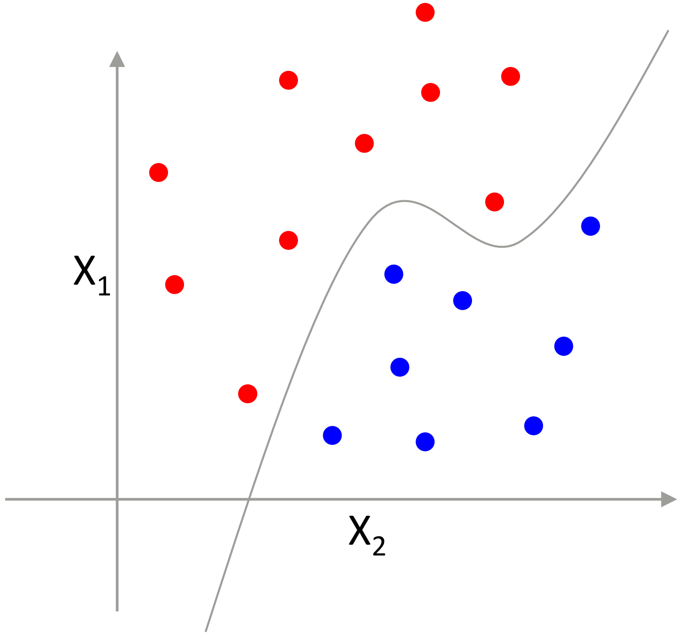
Linear Decision Boundary



©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

Nonlinear Decision Boundary





©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

Support Vector Machine

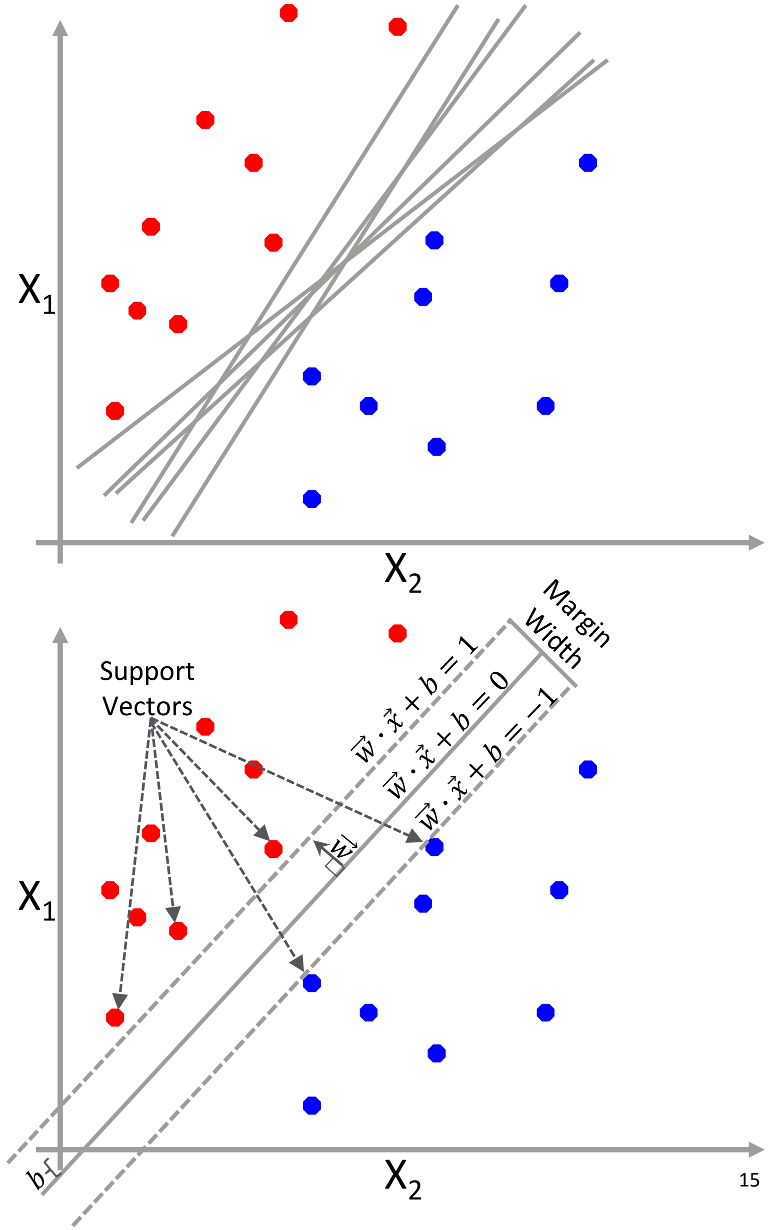
The decision boundary in SVM is defined by the equation of a hyperplane:

$$\vec{w} \cdot \vec{x} + b = 0$$

Where:

- $\vec{w}$ is the weight vector perpendicular to the hyperplane.
- $\vec{x}$ are the input data points.
- b is the bias term, offsetting the hyperplane from the origin.

- How to find the best decision boundary?
  - All lines in the figure correctly separate the 2 classes.
  - The line that is farthest from all training examples will have better generalization capabilities.
- The SVM algorithm solves an optimization problem:
  - First, identify a decision boundary that correctly classifies the examples.
  - Next, increase the geometric margin between the boundary and all examples.
- The data points that define the maximum margin width are called support vectors.
- Find $\vec{w}$ and b by solving:
  - Minimize $||\vec{w}||^2$to maximize the margin.
  - Subject to the constraint that all data points are correctly classified: $y_i(\vec{w} \cdot \vec{x} + b) \geq 1$ for each i, where $y_i$are the labels of the training examples, taking values -1 or +1 depending on the class.



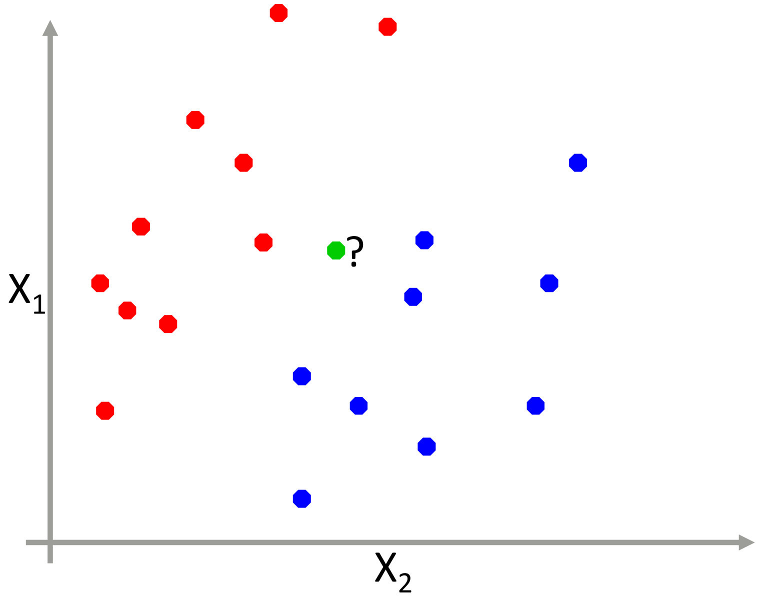
X1
X2
Support Vectors
Margin Width
$\vec{w} \cdot \vec{x} + b = 1$
$\vec{w} \cdot \vec{x} + b = 0$
$\vec{w} \cdot \vec{x} + b = -1$
X1
X2
b

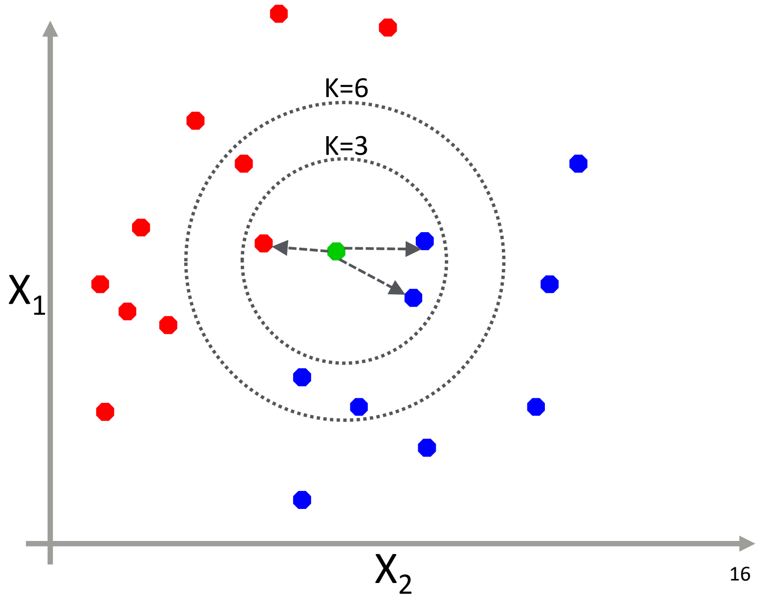
©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

K-Nearest Neighbors

The K-Nearest Neighbors (K-NN) algorithm is a simple and widely-used method for classification and regression tasks in machine learning. Here's a brief explanation of how it works for classification:

- Basic Principle: K-NN classifies a new data point based on the majority vote of its 'k' nearest neighbors in the feature space. The 'k' is a user-defined constant, and the nearest neighbors are determined by a distance metric, typically Euclidean distance.
- Distance Calculation: When a new data point needs to be classified, K-NN calculates the distance from this point to all other points in the training dataset.
- Neighbor Selection: After computing these distances, the algorithm identifies the 'k' closest training examples, which are termed the nearest neighbors.
- Majority Voting: The classification decision is made based on the most common class among these k neighbors. The new data point is assigned to the class that appears most frequently within its closest k neighbors.





©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

Naive Bayes

The Naive Bayes classifier is a probabilistic machine learning model used for classification tasks, which applies Bayes' Theorem with the assumption of independence between the features. Here’s a brief explanation:

- Bayes' Theorem: Naive Bayes classifiers use Bayes' Theorem to predict the probability of a class given a set of features. The theorem is expressed as:

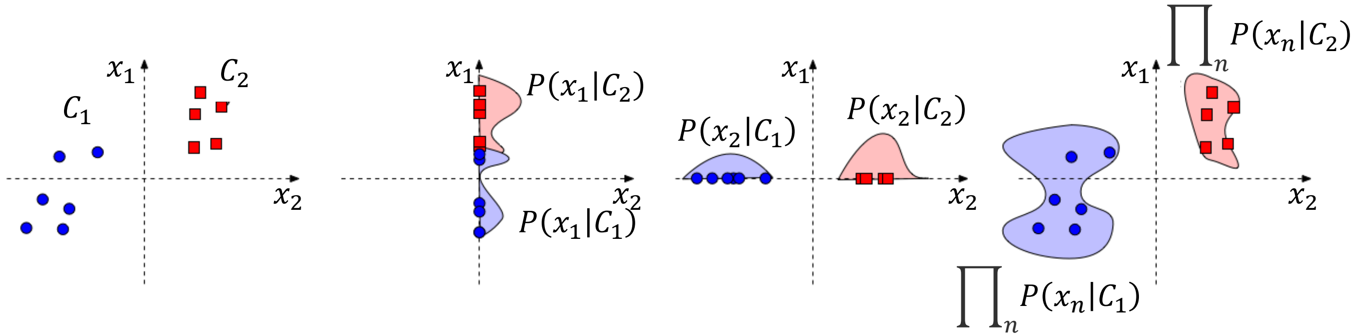
$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)}$$

where $P(C_k|x)$ is the posterior probability of class C_k given feature(s) x, $P(x|C_k)$ is the likelihood which is the probability of feature(s) x given class C_k, $P(C_k)$ is the prior probability of class C_k, and $P(x)$ is the prior probability of feature(s).

- Independence Assumption: It simplifies the calculation of $P(x|C_k)$ by assuming that each feature x_i is independent of the others within each class, which allows the product of the individual probabilities to be taken as the overall likelihood:

$$P(x|C_k)=P(x_1|C_k) \times P(x_2|C_k) \times \cdots \times P(x_n|C_k) = \prod_n P(x_n|C_k)$$

- Model Training: During training, the model calculates the probabilities $P(x_i|C_k)$ from the training data for each feature x_i and each class C_k, as well as the prior probabilities $P(C_k)$.
- Prediction: For a given new instance, the product of the likelihoods of all features and the class prior is calculated for each class, and the class with the highest posterior probability is predicted as the outcome.



©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.

SVM vs K-NN vs Naive Bayes

Here's a concise summary of the pros and cons of the three machine learning algorithms:

- SVM
  - Pros:
    - Effective in high-dimensional spaces.
    - Versatile with various kernel functions.
    - Robust against overfitting.
  - Cons:
    - Inefficient with large datasets.
    - Does not provide probability estimates.
    - Sensitive to noisy data.
- KNN
  - Pros:
    - Simple to implement and understand.
    - Effective for multi-class problems.
    - Flexible to classification, regression, and search.
  - Cons:
    - High computational cost.
    - Sensitive to irrelevant features and scale.
    - Affected by the curse of dimensionality.
- Naive Bayes
  - Pros:
    - Fast and efficient with large datasets.
    - Easy to implement.
    - Good with categorical data.
  - Cons:
    - Assumes feature independence.
    - Can perform poorly with numerical features.
    - Probability outputs are often biased.

©2023 Deloitte Touche Tohmatsu. Todos os direitos reservados.