

Encapsulation

Encapsulation

In object-oriented computer programming languages, the notion of encapsulation refers to the bundling of data, along with the methods that operate on that data, into a single unit. Many programming languages use encapsulation frequently in the form of classes. A class is a program-code-template that allows developers to create an object that has both variables (data) and behaviours (functions or methods). A class is an example of encapsulation in computer science in that it consists of data and methods...

Encapsulation may also refer to a mechanism of restricting the direct access to some components of an object so users cannot access state values for all of the variables of a particular object. Encapsulation can be used to hide both data members and data functions or methods associated with an instantiated class or object.

Encapsulation in programming has a few key benefits. It hides data: users will have no idea how classes are being implemented or stored; all that users will know is that values are being passed and initialised. It gives more flexibility: enables you to set variables as read or write-only. It's easy to reuse: it's easy to change and adapt to new requirements.

Инкапсуляция

В объектно-ориентированных языках программирования понятие инкапсуляции означает объединение данных вместе с методами, которые работают с этими данными, в одну единицу. Многие языки программирования часто используют инкапсуляцию в виде классов. Класс — это шаблон программного кода, который позволяет разработчикам создавать объект, содержащий как переменные (данные), так и поведения (функции или методы). Класс является примером инкапсуляции в информатике, поскольку он объединяет данные и методы в одну единицу.

Инкапсуляция также может означать механизм ограничения прямого доступа к некоторым компонентам объекта, чтобы пользователи не могли получить доступ к значениям состояния всех переменных определённого объекта. Инкапсуляция может использоваться для сокрытия как членов данных, так и функций или методов, связанных с созданным классом или объектом.

Инкапсуляция в программировании имеет несколько ключевых преимуществ. Она скрывает данные: пользователи не знают, как реализованы или хранятся классы; всё, что они знают — это то, что значения передаются и инициализируются. Она даёт большую гибкость: позволяет задавать переменные как только для чтения или только для записи. Это облегчает повторное использование: проще вносить изменения и

адаптировать код под новые требования.

Machine Learning Methods

Machine Learning Methods

Supervised learning algorithms are trained using labelled examples, such as an input where the desired output is known. For example, a piece of equipment could have data points labelled either “F” (failed) or “R” (runs). The learning algorithm receives a set of inputs along with the corresponding correct outputs, and the algorithm learns by comparing its actual output with correct outputs to find errors. It then modifies the model accordingly. Through methods like classification, regression, prediction and gradient boosting, supervised learning uses patterns to predict the values of the label on additional unlabelled data. Supervised learning is commonly used in applications where historical data predicts likely future events.

Unsupervised learning is used against data that has no historical labels. The system is not told the “right answer”. The algorithm must figure out what is being shown. The goal is to explore the data and find some structure within. Unsupervised learning works well on transactional data. For example, it can identify segments of customers with similar attributes who can then be treated similarly in marketing campaigns. Or it can find the main attributes that separate customer segments from each other. Popular techniques include self-organising maps, nearest-neighbour mapping, k-means clustering and singular value decomposition.

Методы машинного обучения

Алгоритмы обучения с учителем (supervised learning) обучаются с использованием размеченных примеров, таких как входные данные, для которых известен желаемый результат. Например, оборудование может иметь метки данных: “F” (отказ) или “R” (работает). Алгоритм получает набор входных данных и соответствующих выходов, сравнивает свои результаты с правильными и корректирует модель, чтобы уменьшить ошибки. Через такие методы, как классификация, регрессия, предсказание и градиентный спуск, обучение с учителем использует шаблоны для прогнозирования значений на новых неразмеченных данных. Такой подход широко используется там, где исторические данные позволяют предсказать будущие события.

Обучение без учителя (unsupervised learning) используется для данных без исторических меток. Алгоритму не сообщают «правильный ответ». Его цель — изучить данные и найти в них структуру. Такой подход хорош для транзакционных данных. Например, можно определить сегменты клиентов с похожими характеристиками для маркетинга или найти ключевые признаки, отличающие одну группу от другой. Популярные методы: самоорганизующиеся карты, кластеризация k-средних и сингулярное разложение.