< 机器学习 | 目录 | Scikit-Learn简介 >

Open in Colab

What Is Machine Learning?

什么是机器学习?

> Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

在我们开始学习机器学习方法的细节之前，让我们先来了解机器学习是什么以及不是什么。机器学习经常被归为人工智能的一个子领域，但作者发现这种分类方式常常一开始就导致了误解。对机器学习的研究肯定是在这个环境中发展出来的，但是机器学习方法在数据科学应用中，它更适合被看成是*数据的构造模型*。

> Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.

机器学习基本上就是关于构建数学模型来帮助我们理解数据。当我们为这些模型提供了*可调整的参数*时，“学习”能让我们从观察到的数据中调整这些参数。也就是说，这个过程可以被认为我们从数据中“学习”。一旦这些模型已经适应（拟合）了观察到的数据之后，它们就可以用来预测和理解新的数据。作者把这个问题的哲学思考留给读者，基于模型的“学习”确实与人脑展示的“学习”类似。

> Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

理解机器学习中的各种概念是有效使用这些工具的基础，因此我们首先介绍机器学习的分类以及方法的类型。

Categories of Machine Learning

机器学习分类

> At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

在最基础的层次上，机器学习可以被分为两大类：有监督学习和无监督学习。

> *Supervised learning* involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

*有监督学习*指的是在除了数据本身外，我们还拥有对数据进行的标记，有监督学习就是要建立两者之间的联系模型，然后这个模型就可以应用在新的数据上进行标记。它可以进一步分为*分类*和*回归*任务：在分类中，标记的是离散的分组，而在回归中，标记的是连续的量。我们在后续章节中会看到这两种有监督学习的例子。

> *Unsupervised learning* involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

*无监督学习*是从没有标记的数据中建立模型，它常被描述为“让数据集自己说话”。这样的模型包括*聚类*和*降维*。聚类算法能识别数据中的分组，而降维算法寻找数据更简洁的表达形式。我们在后续章节中会看到这两种无监督学习的例子。

> In addition, there are so-called *semi-supervised learning* methods, which falls somewhere between supervised learning and unsupervised learning. Semi-supervised learning methods are often useful when only incomplete labels are available.

除此之外，还有一种被成为*半监督学习*的方法，介于有监督学习和无监督学习之间。半监督学习方法经常应用在不完整的数据标记的场合中。

Qualitative Examples of Machine Learning Applications

机器学习应用的定性例子

> To make these ideas more concrete, let's take a look at a few very simple examples of a machine learning task. These examples are meant to give an intuitive, non-quantitative overview of the types of machine learning tasks we will be looking at in this chapter. In later sections, we will go into more depth regarding the particular models and how they are used. For a preview of these more technical aspects, you can find the Python source that generates the following figures in the Appendix: Figure Code.

要更具体的说明这些内容，我们来看一些非常简单的机器学习任务例子。这些例子为了给读者提供一个直观的，非定量的机器学习任务的概要介绍。在后续章节中，我们会深入介绍每一个模型以及它们是如何使用的。产生下面的图像的代码可以在附录：产生图像的代码中找到。

Classification: Predicting discrete labels

分类：预测离散的标签

> We will first take a look at a simple *classification* task, in which you are given a set of labeled points and want to use these to classify some unlabeled points.
>
> Imagine that we have the data shown in this figure:

我们首先看一个简单的*分类*任务，你有一组标记过的点，然后你使用这些数据来标记新的未标记过的数据点。我们有下图展示的数据：

附录中生成图像的代码

> Here we have two-dimensional data: that is, we have two *features* for each point, represented by the *(x,y)* positions of the points on the plane. In addition, we have one of two *class labels* for each point, here represented by the colors of the points. From these features and labels, we would like to create a model that will let us decide whether a new point should be labeled "blue" or "red."

这里我们有二维的数据：即这里面的每个点我们都有两个*特征*，使用平面中的*(x,y)*位置表示。除此之外，我们对每个点都有一个标记，标记一共有两种，上图中使用了颜色进行区分。使用这些特征和标记，我们可以建立一个模型，然后我们就可以对一个新的数据点进行标记，判断它属于“蓝色”还是“红色”。

> There are a number of possible models for such a classification task, but here we will use an extremely simple one. We will make the assumption that the two groups can be separated by drawing a straight line through the plane between them, such that points on each side of the line fall in the same group. Here the *model* is a quantitative version of the statement "a straight line separates the classes", while the *model parameters* are the particular numbers describing the location and orientation of that line for our data. The optimal values for these model parameters are learned from the data (this is the "learning" in machine learning), which is often called *training the model*.

对于这个分类任务来说可以有很多可能的模型，但是我们会使用一个特别简单的模型。我们假设这两组数据点可以使用一条平面上的直线进行区分，直线两边分别属于两个不同的组。这里的*模型*是“一条分类直线”说法的定量版本，而*模型中的参数*就是用来描述直线位置和方向的特殊数字。优化后的模型参数值是从数据中学习得到的，这个学习过程我们通常成为*训练模型*。

> The following figure shows a visual representation of what the trained model looks like for this data:

下面展示了一个训练好的模型的可视化图像：

附录中生成图像的代码

> Now that this model has been trained, it can be generalized to new, unlabeled data. In other words, we can take a new set of data, draw this model line through it, and assign labels to the new points based on this model. This stage is usually called *prediction*. See the following figure:

当模型训练好之后，它就能泛化到新的未标记的数据上。换一种说法是，我们可以取一组新的数据，将模型的直线画上去穿过它们，然后给新的数据点定义标签。这个阶段通常被称为*预测*。参见下面的图：

附录中生成图像的代码

> This is the basic idea of a classification task in machine learning, where "classification" indicates that the data has discrete class labels. At first glance this may look fairly trivial: it would be relatively easy to simply look at this data and draw such a discriminatory line to accomplish this classification. A benefit of the machine learning approach, however, is that it can generalize to much larger datasets in many more dimensions.

上面就是机器学习中分类任务的基本概念，这里的*分类*表明数据具有离散的类别标签。第一眼看上去这个任务显得很琐碎：观察数据并画出这样一条分类的直线显得相对来说很容易。但是机器学习方法的优势在于，它可以泛化到非常大的数据集上，以及更多的维度上。

> For example, this is similar to the task of automated spam detection for email; in this case, we might use the following features and labels:
>
> - *feature 1*, *feature 2*, etc. → normalized counts of important words or phrases ("Viagra", "Nigerian prince", etc.)
> - *label* → "spam" or "not spam"

例如，类似自动垃圾电子邮件识别，在这种情况下，我们可能会用到下面的特征和标签：

- *特征1*、*特征2*等 → 正则化后的重要单词或短语的计数（“伟哥”，“尼日利亚王子”等）
- *标签* → “垃圾邮件”或“非垃圾邮件”

> For the training set, these labels might be determined by individual inspection of a small representative sample of emails; for the remaining emails, the label would be determined using the model. For a suitably trained classification algorithm with enough well-constructed features (typically thousands or millions of words or phrases), this type of approach can be very effective. We will see an example of such text-based classification in In Depth: Naive Bayes Classification.

对于这个训练集来说，这些标签可以通过检查一部分电子邮件的典型样本来获得，对于剩余的电子邮件，标签可以使用模型得到。对于一个良好训练的分类算法而言，它包括足够多的特征（上千或上百万的单词或短语），这样的方法会非常有效。我们会在深入：朴素贝叶斯分类一节中看到一个文本分类的例子。

> Some important classification algorithms that we will discuss in more detail are Gaussian naive Bayes (see In Depth: Naive Bayes Classification), support vector machines (see In-Depth: Support Vector Machines), and random forest classification (see In-Depth: Decision Trees and Random Forests).

我们后续会讨论到的一些重要的分类算法包括高斯朴素贝叶斯（参见深入：朴素贝叶斯分类），支持向量机（参见深入：支持向量机）和随机森林分类（参见深入：决策树和随机森林）。

Regression: Predicting continuous labels

回归：预测连续标签

> In contrast with the discrete labels of a classification algorithm, we will next look at a simple *regression* task in which the labels are continuous quantities.

对比离散标签分类算法，我们下面来看一个简单的*回归*任务，它的标签是一个连续的数量。

> Consider the data shown in the following figure, which consists of a set of points each with a continuous label:

考虑如下图展示的数据，包含着一组的数据点每一个都有一个连续的标签：

附录中生成图像的代码

> As with the classification example, we have two-dimensional data: that is, there are two features describing each data point. The color of each point represents the continuous label for that point.

就像分类例子中那样，我们有着二维的数据：即每个数据点都有两个特征。每个点的颜色代表这这个点的连续标签。

> There are a number of possible regression models we might use for this type of data, but here we will use a simple linear regression to predict the points. This simple linear regression model assumes that if we treat the label as a third spatial dimension, we can fit a plane to the data. This is a higher-level generalization of the well-known problem of fitting a line to data with two coordinates.

对于这个数据集来说，可以有很多种可能的回归模型，但是这里我们会使用一种简单的线性回归来预测数据点。这个简单的线性回归模型假设我们将数据标签作为第三个空间维度，我们可以在上面使用一个平面来拟合数据。这是在两个坐标中使用一根直线来拟合数据的泛化版本。

> We can visualize this setup as shown in the following figure:

可以使用下图可视化这个设置：

附录中生成图像的代码

> Notice that the *feature 1-feature 2* plane here is the same as in the two-dimensional plot from before; in this case, however, we have represented the labels by both color and three-dimensional axis position. From this view, it seems reasonable that fitting a plane through this three-dimensional data would allow us to predict the expected label for any set of input parameters. Returning to the two-dimensional projection, when we fit such a plane we get the result shown in the following figure:

注意上图中的*特征1 - 特征2*平面与前面二维图中数据点是一致的；我们使用了颜色以及三维坐标表示数据点的标签。从上图中我们可以看到，通过这个平面可以让我们对任意输入的数据点参数进行标签的预测。返回到二维投射，当我们拟合了这个平面我们会得到下图的结果：

附录中生成图像的代码

> This plane of fit gives us what we need to predict labels for new points. Visually, we find the results shown in the following figure:

拟合得到的平面能为我们提供预测新数据点标签的能力。下面的图像展示了预测的结果：

附录中生成图像的代码

> As with the classification example, this may seem rather trivial in a low number of dimensions. But the power of these methods is that they can be straightforwardly applied and evaluated in the case of data with many, many features.

同样的，这个方法在维度较少时显得很普通。但是当数据的特征很多时，这个方法的威力就显现出来了。

> For example, this is similar to the task of computing the distance to galaxies observed through a telescope—in this case, we might use the following features and labels:
>
> - *feature 1*, *feature 2*, etc. → brightness of each galaxy at one of several wave lengths or colors
> - *label* → distance or redshift of the galaxy

例如，类似通过望远镜计算星系之间距离任务时，我们会使用下面的特征和标签：

- *特征1*、*特征2*等 → 每个星系在不同波长或颜色范围上的亮度值
- *标签* → 星系的距离或红移

> The distances for a small number of these galaxies might be determined through an independent set of (typically more expensive) observations. Distances to remaining galaxies could then be estimated using a suitable regression model, without the need to employ the more expensive observation across the entire set. In astronomy circles, this is known as the "photometric redshift" problem.

少量的星系距离可以通过独立的观测方式（通常更加昂贵）来获得。剩余的星系距离可以使用合适的回归模型进行估算，避免了在所有星系上使用昂贵观测方法的需要。在天文学领域，这被称为*光度红移*问题。

> Some important regression algorithms that we will discuss are linear regression (see In Depth: Linear Regression), support vector machines (see In-Depth: Support Vector Machines), and random forest regression (see In-Depth: Decision Trees and Random Forests).

我们还会介绍其他一些重要的回归算法，包括线性回归（参见深入：线性回归），支持向量机（参见深入：支持向量机）和随机森林回归（参见深入：决策树和随机森林）。

Clustering: Inferring labels on unlabeled data

聚类：在未标记的数据上推断标签

> The classification and regression illustrations we just looked at are examples of supervised learning algorithms, in which we are trying to build a model that will predict labels for new data. Unsupervised learning involves models that describe data without reference to any known labels.

上面介绍的分类和回归为我们展示了使用有监督学习算法的例子，我们会从数据中学习得到一个模型然后使用它预测新数据的标签。无监督学习用来描述数据的模型是从没有任何已知标签的数据中获得的。

> One common case of unsupervised learning is "clustering," in which data is automatically assigned to some number of discrete groups. For example, we might have some two-dimensional data like that shown in the following figure:

最常见的无监督学习场景是“聚类”，其中的数据自动组合成一些离散的分组。例如下图中展示的二维数据：

附录中生成图像的代码

> By eye, it is clear that each of these points is part of a distinct group. Given this input, a clustering model will use the intrinsic structure of the data to determine which points are related. Using the very fast and intuitive *k*-means algorithm (see In Depth: K-Means Clustering), we find the clusters shown in the following figure:

肉眼观察可以知道很显然这些数据点是不同分组的组成部分。对于这个输入来说，一个聚类模型会使用输入数据的内在结构来找到哪些点是关联的。使用下面快速直观的*k均值*算法（参见深入：k均值聚类），我们会发现如下如的聚类：

附录中生成图像的代码

> *k*-means fits a model consisting of *k* cluster centers; the optimal centers are assumed to be those that minimize the distance of each point from its assigned center. Again, this might seem like a trivial exercise in two dimensions, but as our data becomes larger and more complex, such clustering algorithms can be employed to extract useful information from the dataset.

*k均值*会适应训练出一个包括*k*个聚类中心点的模型；优化后的中心点应该是属于这个聚类群的所有点距离之和最小的点。还是需要说明的是在二维的情况下，这看起来有点平淡无奇，但是当我们数据变得更大更复杂时，这种聚类算法可以用来从数据集中提取出有用的信息。

> We will discuss the *k*-means algorithm in more depth in In Depth: K-Means Clustering. Other important clustering algorithms include Gaussian mixture models (See In Depth: Gaussian Mixture Models) and spectral clustering (See Scikit-Learn's clustering documentation).

我们会在深入：k均值聚类一节中深入讨论k均值算法。其他重要的聚类算法包括高斯混合模型（参见深入：高斯混合模型）和谱聚类（参见Scikit-Learn聚类在线文档）。

Dimensionality reduction: Inferring structure of unlabeled data

降维：推断无标记数据的结构

> Dimensionality reduction is another example of an unsupervised algorithm, in which labels or other information are inferred from the structure of the dataset itself. Dimensionality reduction is a bit more abstract than the examples we looked at before, but generally it seeks to pull out some low-dimensional representation of data that in some way preserves relevant qualities of the full dataset. Different dimensionality reduction routines measure these relevant qualities in different ways, as we will see in In-Depth: Manifold Learning.

降维是另一个无监督算法的例子，它能从数据集本身的结构推断标签或其他的信息。降维的例子比起前面那些算法的例子稍微复杂一些，总的来说，降维通过用更少维度的数据表达但是却保留了完整数据集的相关关键信息。不同的降维算法从不同方面衡量这些相关信息，就像我们会在深入：流形学习中看到的那样。

> As an example of this, consider the data shown in the following figure:

使用下图展示的数据作为例子：

附录中产生图像的代码

> Visually, it is clear that there is some structure in this data: it is drawn from a one-dimensional line that is arranged in a spiral within this two-dimensional space. In a sense, you could say that this data is "intrinsically" only one dimensional, though this one-dimensional data is embedded in higher-dimensional space. A suitable dimensionality reduction model in this case would be sensitive to this nonlinear embedded structure, and be able to pull out this lower-dimensionality representation.

从图上很容易看出数据有一些内在的结构：数据是由一维的线卷曲成螺旋状的二维形状。或者直觉上你可以认为数据本质上是一维的，不过是嵌入在一个更高维度的空间中。一个合适的降维模型可以在这个情况下感知这种非线性的内嵌结构，并且能够将其低维度的数据表现方式提取出来。

> The following figure shows a visualization of the results of the Isomap algorithm, a manifold learning algorithm that does exactly this:

下面展示了使用Isomap算法的可视化结果，这是一种适合该应用场景的流形学习算法：

附录中生成图像的代码

> Notice that the colors (which represent the extracted one-dimensional latent variable) change uniformly along the spiral, which indicates that the algorithm did in fact detect the structure we saw by eye. As with the previous examples, the power of dimensionality reduction algorithms becomes clearer in higher-dimensional cases. For example, we might wish to visualize important relationships within a dataset that has 100 or 1,000 features. Visualizing 1,000-dimensional data is a challenge, and one way we can make this more manageable is to use a dimensionality reduction technique to reduce the data to two or three dimensions.

注意到上图中的颜色（代表着提取出来的一维隐变量）是沿着螺旋线均匀变化的，这表明算法确实能够检测到我们肉眼观察到的结构。降维算法的威力同样可以在更高维度的数据中更好的展现出来。例如，我们希望将具有100或1000个特征的数据集的重要关联关系在图中可视化出来，可视化1000维度的数据是非常具有挑战性的，我们可以通过降维技术将数据维度减少到二维或三维，这就很容易实现可视化了。

> Some important dimensionality reduction algorithms that we will discuss are principal component analysis (see In Depth: Principal Component Analysis) and various manifold learning algorithms, including Isomap and locally linear embedding (See In-Depth: Manifold Learning).

我们在本章中会介绍一些重要的降维算法，包括主成分分析（参见深入：主成分分析）和不同的流形学习算法，如Isomap和局部线性嵌入（参见深入：流形学习）。

Summary

总结

> Here we have seen a few simple examples of some of the basic types of machine learning approaches. Needless to say, there are a number of important practical details that we have glossed over, but I hope this section was enough to give you a basic idea of what types of problems machine learning approaches can solve.

本节中我们看到了一些基本机器学习方法的简单例子。无需说明也看得出来，我们只是一笔带过的进行了相关介绍，但通过本节的内容希望能为读者提供了关于机器学习方法能够解决的问题类型的基本概念。

> In short, we saw the following:
>
> - *Supervised learning*: Models that can predict labels based on labeled training data
> - *Classification*: Models that predict labels as two or more discrete categories
> - *Regression*: Models that predict continuous labels
> - *Unsupervised learning*: Models that identify structure in unlabeled data
> - *Clustering*: Models that detect and identify distinct groups in the data
> - *Dimensionality reduction*: Models that detect and identify lower-dimensional structure in higher-dimensional data

简单来说，有如下的主要几个方面：

- *有监督学习*：建立一个能够根据带标记的训练数据对数据进行标签预测的模型
  - *分类*：建立一个能够预测两个或多个离散分组标签的模型
  - *回归*：建立一个能够预测连续标签的模型
- *无监督学习*：建立一个能够识别未标记数据内在结构的模型
  - *聚类*：建立一个检查和识别数据不同分组的模型
  - *降维*：建立一个能发现高维度数据在低维度情况下结构的模型

> In the following sections we will go into much greater depth within these categories, and see some more interesting examples of where these concepts can be useful.

在后续章节中，我们会深入到上述的这些机器学习方法类型中，还有看到更多这些方法能发挥作用的有趣的例子。

> All of the figures in the preceding discussion are generated based on actual machine learning computations; the code behind them can be found in Appendix: Figure Code.

本节中所有的图像都是使用真实的机器学习计算生成的；产生图像的代码可以在附录：生成图像的代码中找到。

< 机器学习 | 目录 | Scikit-Learn简介 >

Open in Colab