Statistical Machine Learning Approaches to Liver Disease Prediction

TEAM ID: PNT2022TMID42133

Prior Knowledge

Supervised and unsupervised learning Machine learning:

Within artificial intelligence (AI) and machine learning, there are two basic approaches: supervised learning and unsupervised learning. The main difference is one uses labeled data to help predict outcomes, while the other does not. However, there are some nuances between the two approaches, and key areas in which one outperforms the other. This post will clarify the differences so you can choose the best approach for your situation.

supervised learning:

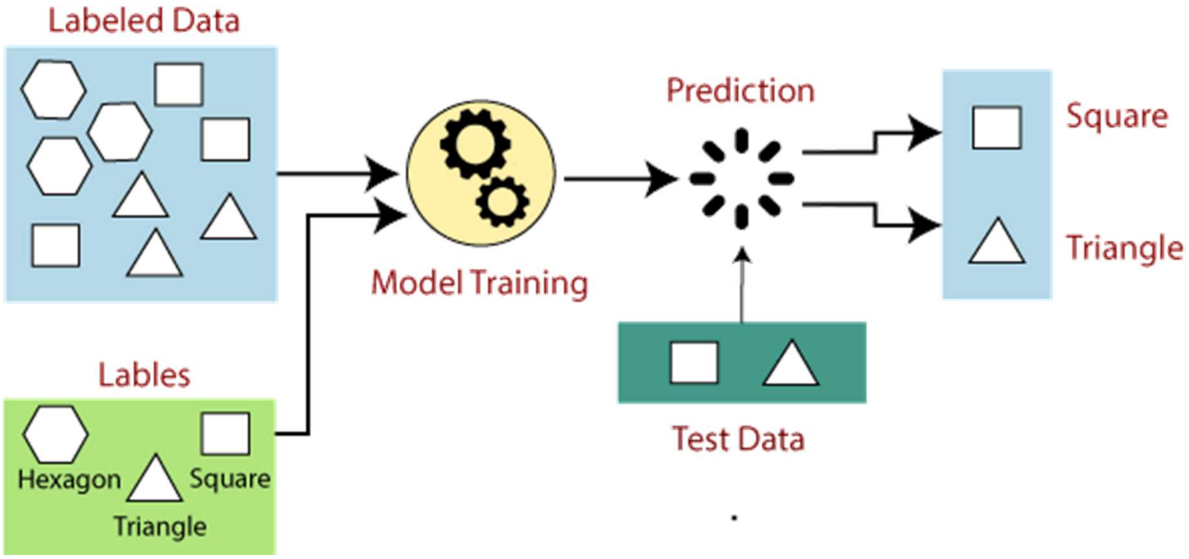
Supervised learning is a machine learning approach that's defined by its use of labeled datasets. These datasets are designed to train or "supervise" algorithms into classifying data or predicting outcomes accurately. Using labeled inputs and outputs, the model can measure its accuracy and learn over time.

Supervised learning can be separated into two types of problems when data mining: classification and regression:

- Classification problems use an algorithm to accurately assign test data into specific categories, such as separating apples from oranges. Or, in the real world, supervised learning algorithms can be used to classify spam in a separate folder from your inbox. Linear classifiers, support vector machines, decision trees and random forest are all common types of classification algorithms.
- Regression is another type of supervised learning method that uses an algorithm to understand the relationship between dependent and independent variables. Regression models are helpful for predicting numerical values based on different data points, such as sales

revenue projections for a given business. Some popular regression algorithms are linear regression, logistic regression and polynomial regression.

-



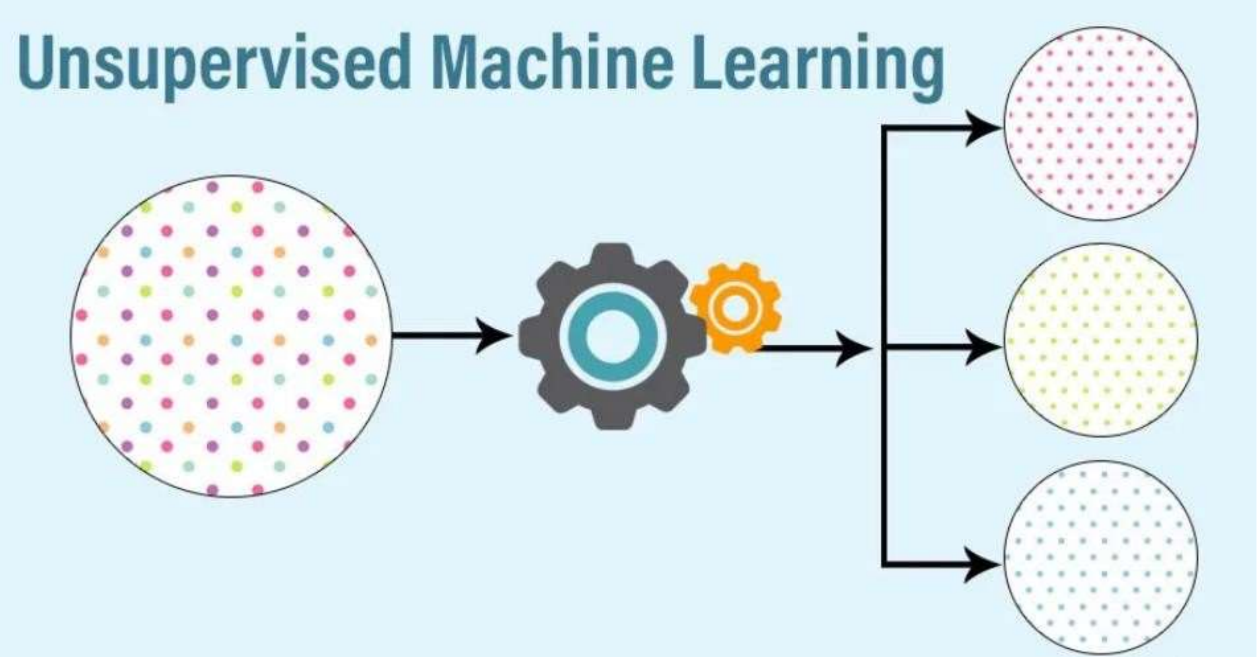
unsupervised learning

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns in data without the need for human intervention (hence, they are “unsupervised”).

Unsupervised learning models are used for three main tasks: clustering, association and dimensionality reduction:

- Clustering is a data mining technique for grouping unlabeled data based on their similarities or differences. For example, K-means clustering algorithms assign similar data points into groups, where the K value represents the size of the grouping and granularity. This technique is helpful for market segmentation, image compression, etc.

- Association is another type of unsupervised learning method that uses different rules to find relationships between variables in a given dataset. These methods are frequently used for market basket analysis and recommendation engines, along the lines of "Customers Who Bought This Item Also Bought" recommendations.
- Dimensionality reduction is a learning technique used when the number of features (or dimensions) in a given dataset is too high. It reduces the number of data inputs to a manageable size while also preserving the data integrity. Often, this technique is used in the preprocessing data stage



Machine Learning in R - Classification, Regression and Clustering Problems:

First up is Classification. A **classification problem** involves predicting whether a given observation belongs to one of two or more categories. The simplest case of classification is called binary classification. It has to decide between two categories, or classes. Remember how I compared machine learning to the estimation of a function? Well, based on earlier observations of how the input maps to the output, classification tries to estimate a classifier that can generate an output for an arbitrary input, the observations. We say that the classifier labels an unseen example with a class. The possible applications of classification are very broad. For example, after a set of clinical examinations that relate vital signals to a disease, you could predict whether a new patient with an unseen set of vital signals suffers that disease and needs further treatment. Another totally different example is classifying a set of animal images into cats, dogs and horses, given that you have trained your model on a bunch of images for which you know what animal they depict. Can you think of a possible classification problem yourself? What's important here is that first off, the output is qualitative, and second, that the classes to which new observations can belong, are known beforehand. In the first example I mentioned, the classes are "sick" and "not sick". In the second examples, the classes are "cat", "dog" and "horse". In chapter 3 we will do a deeper analysis of classification and you'll get to work with some fancy classifiers!

Moving on ... A ***Regression problem*** is a kind of Machine Learning problem that tries to predict a continuous or quantitative value for an input, based on previous information. The input variables, are called the predictors and the output the response. In some sense, regression is pretty similar to classification. You're also trying to estimate a function that maps input to output based on earlier observations, but this time you're trying to estimate an actual value, not just the class of an observation. Do you remember the example from last video, there we had a dataset on a group of people's height and weight. A valid question could be: is there a linear relationship between these two? That is, will a change in height correlate linearly with a change in weight, if so can you describe it and if we know the weight, can you predict the height of a new person given their weight ? These questions can be answered with linear regression! Together, \beta_0 and \beta_1 are known as the model coefficients or parameters. As soon as you know the coefficients beta 0 and beta 1 the function is able to convert any new input to output. This means that solving your machine learning problem is actually finding good values for beta 0 and beta 1. These are estimated based on previous input to output observations. I will not go into details on how to compute these coefficients, the function `lm()` does this for you in R. Now, I hear you asking: what can regression be useful for apart from some silly weight and height problems? Well, there are many different applications of regression, going from modeling credit scores based on past payements, finding the trend in your youtube subscriptions over time, or even estimating your chances of landing a job at your favorite company based on your college grades. All these problems have two things in common. First off, the response, or the thing you're trying to predict, is always quantitative. Second, you will always need input knowledge of previous input-output observations, in order to build your model. The fourth chapter of this course will be devoted to a more comprehensive overview of regression. Soooo.. Classification: check. Regression: check. Last but not least, there is clustering. In clustering, you're trying to group objects that are similar, while making sure the clusters themselves are dissimilar. You can think of it as classification, but without saying to which classes the observations have to belong or how many classes there are.

Take the animal photo's for example. In the case of classification, you had information about the actual animals that were depicted. In the case of clustering, you don't know what animals are depicted, you would simply get a set of pictures. The clustering algorithm then simply groups similar photos in clusters. You could say that **clustering** is different in the sense that you don't need any knowledge about the labels. Moreover, there is no right or wrong in clustering. Different clusterings can reveal different and useful information about your objects. This makes it quite different from both classification and regression, where there always is a notion of prior expectation or knowledge of the result.

How to Build a Predictive Model in Python?

One of the great perks of Python is that you can build solutions for real-life problems. This applies in almost every industry. From building models to predict diseases to building web apps that can forecast the future sales of your online store, knowing how to code enables you to think outside of the box and broadens your professional horizons as a data scientist.

Whether you've just learned the Python basics or already have significant knowledge of the programming language, knowing your way around predictive programming and learning how to build a model is essential for machine learning. In this practical tutorial, we'll learn together how to build a binary logistic regression in 5 quick steps.

Table of Contents: