

Introduction to Supervised Learning



(<https://github.com/learn-co-curriculum/dsc-intro-to-supervised-learning-v2-1/issues/new/choose>)

Introduction

In this lesson, we'll examine what exactly the term "Supervised Learning" means, and where it fits in Data Science.

Objectives

- Describe the components of what makes something a supervised learning task

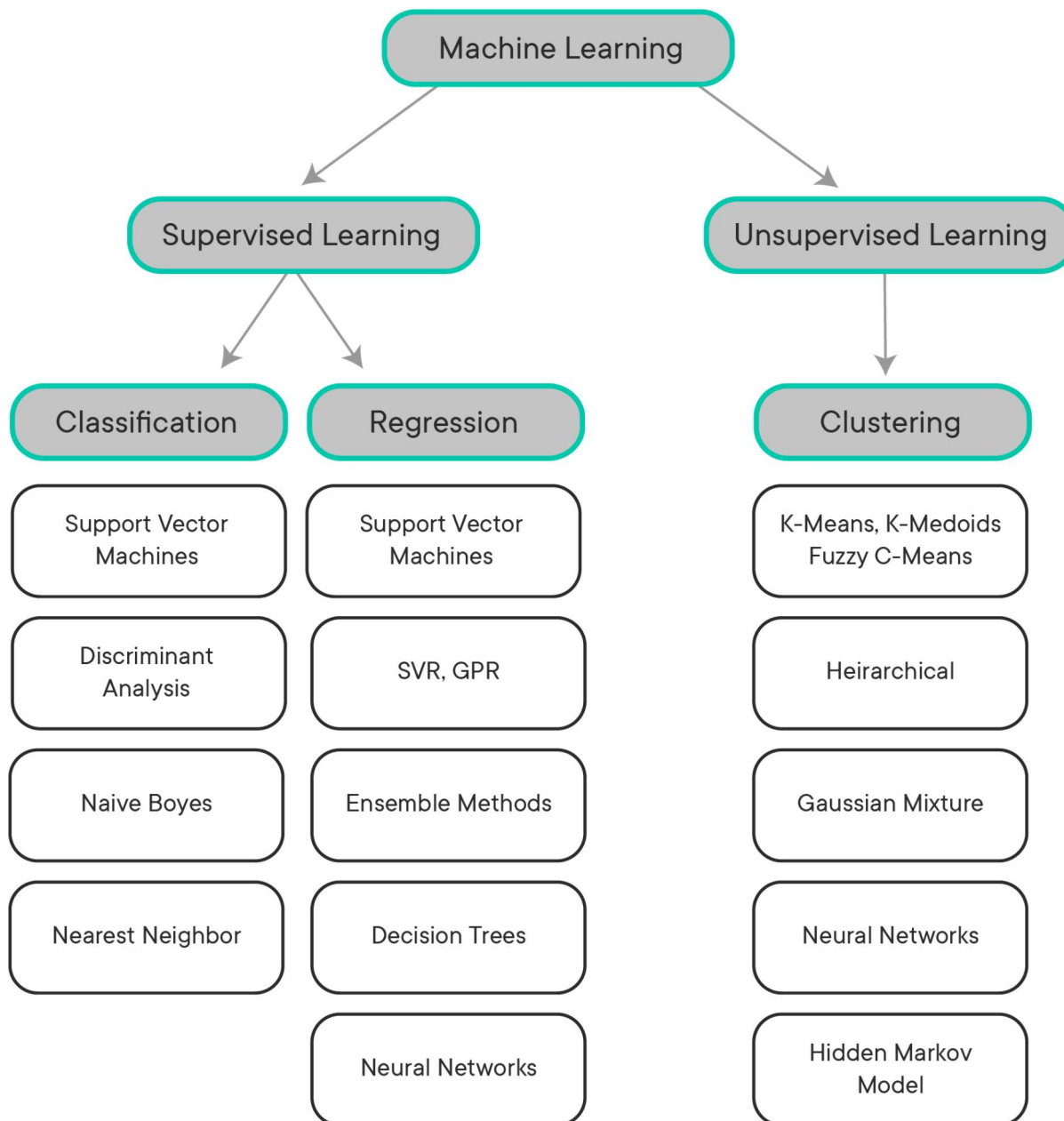
What is Supervised Learning?

The term **Supervised Learning** refers to a class of machine learning algorithms that can "learn" a task through **labeled training data**. We'll explore this definition more fully in a bit -- but first, it's worth taking some time to understand where supervised learning fits in the overall picture in regards to Data Science. By now, you've probably noticed that many of the things we've learned in Data Science and Computer Science are very hierarchical. This is especially true when it comes to AI and Machine Learning. Let's break down the hierarchy a bit, and see where **Supervised Learning** fits.

Artificial Intelligence

At the top of the hierarchy is **Artificial Intelligence**. AI is a catch-all term for various kinds of algorithms that can complete tasks that normally require human intelligence to complete. AI is made up of several subcategories, and is also a subcategory itself in the greater hierarchy of Computer Science. When data scientists talk about AI, we're almost focused on a single branch of AI, **Machine Learning**. Machine Learning is responsible for the boom in AI technologies and abilities in the last few decades, but it's worth noting that there are other areas of AI that do not fall under the umbrella of 'Machine Learning'. Other branches of AI include things like *Genetic Algorithms* for optimization, or rules-based AI for things like building a bot for players to play against in a video game. While these are still active areas of research, they have little to no application in Data Science, so they're beyond the scope of this lesson. In general, when you see the phrase 'Artificial Intelligence', it's generally safe to assume that the speaker is probably referring to the subfield of AI known as **Machine Learning** (which is also sometimes referred to by its older, more traditional name -- **Statistical Learning**).

The following graphic shows the breakdown of the 'Machine Learning' branch of AI:




Machine Learning

The field of *Machine Learning* can be further divided into two overall categories:

1. **Supervised Learning**
2. *Unsupervised Learning*

The main difference between these two areas of machine learning is the need for **labeled training data**. In **Supervised Learning**, any data used must have a **label**. These labels are the *ground truth*, which allows our supervised learning algorithms to 'check their work'. By comparing its predictions against the actual labels, our algorithm can learn to make less incorrect predictions and improve the overall performance of the task its learning to do. It helps to think of Supervised Learning as close to the type of learning we do as students in grade school. Imagine using practice exams to study for the SAT or ACT test. We can go through all the practice questions we want, but in order to learn from our

performance on those practice questions, we need to know what the correct answers are! Without them, we would have no way of knowing which questions we got right and which ones we got wrong, so we wouldn't be able to learn what changes we would need to make to improve our overall performance!

"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 ." -- [Tom Mitchell](http://www.cs.cmu.edu/%7Etom/)  (<http://www.cs.cmu.edu/%7Etom/>).


Let's pretend we've built and trained a model to detect if a picture contains a cat or not. Using the language from the definition above:

- **Task (T):** predict if a picture contains a cat or not
- **Performance Measure (P):** The objective function used to score the predictions made by our model for each image
- **Experience (E):** All of our labeled training data. The more training data we provide, the more 'experience' our model gets!

We'll spend some time learning about **Unsupervised Learning** in the next module, so don't worry about it for now!

Classification and Regression

The field of *Supervised Learning* can be further broken down into two categories -- **Classification** and **Regression**. At this point in your studies, you already have significant experience with regression -- specifically **Linear Regression**, probably the most foundational (and important) machine learning model. Recall that regression allows us to answer questions like "how much?" or "how many?". If our label is a real-valued number, then the supervised learning problem you're trying to solve is a *regression* problem.

The other main kind of supervised learning problem is **Classification**. Classification allows us to tell if something belongs to one class or the other. In the case of the [titanic](https://www.kaggle.com/c/titanic)  (<https://www.kaggle.com/c/titanic>) dataset, this may be something like survival. For example, given various characteristics of a passenger, predict whether they will survive or not. Questions that can be answered in a True/False format (in the titanic example, "Survived" or "Not survived") are a type of **Binary Classification**. To perform binary classification, you will be introduced to **Logistic Regression**. Don't let the name confuse you, although the name contains the word "regression," this important foundational technique is very important in understanding classification problems. There are several other classification techniques you will be learning in this module, but in order to gain a sound understanding of **Classification** tasks, this section will be focused exclusively on building and evaluating logistic regression models.

However, we are not limited to only two classes when working with classification algorithms -- we can have as many classes as we see fit. When a supervised learning problem has more than two classes, we refer to it as a **Multiclass Classification** problem.

Objective Functions

Whenever we're dealing with supervised learning, we have an **Objective Function** (also commonly called a **Loss Function**) that we're trying to optimize against. Regardless of the supervised learning model we're working with, we can be sure that we have some sort of function under the hood that we're using to grade the predictions made by our model against the actual ground-truth labels for each prediction. In the quote from Tom Mitchell listed above, objective functions are P . While classification and regression models use different kinds of objective functions to evaluate their performance, the concept is the same -- these functions allow the model to evaluate exactly how right or wrong a prediction is, which the algorithm can then "learn" from. These objective functions serve an important purpose, because they act as the ground-truth for determining if our model is getting better or not.

The Limitations of Labeled Data

Because supervised learning requires **Labels** for any data used, this severely limits the amount of available data we have for use with supervised learning algorithms. Of all the data in the world, only a very, very small percentage is labeled. Why? Because labeling data is a purposeful activity that can only be done by humans, and is therefore time-consuming and expensive. In supervised learning, labels are not universal -- they are unique to the problem we're trying to solve. If we're trying to train a model to predict if someone survived the titanic disaster, we need to know the survival results of every passenger in our dataset -- there's no way around it. However, if we're trying to predict how much a person paid for a ticket on the titanic, survival data now no longer works as a label -- instead, we need to know how much each passenger paid for a ticket. In a more generalized sense, this means that for whatever problem we're trying to train a supervised learning model to solve, we need to have a large enough dataset containing examples where humans have already done the things we're trying to get our model to learn how to do.

Although labeled data is still expensive and time-consuming to get, the internet has made the overall process of getting labeled data a bit easier than it used to be. Nowadays, when companies need to construct a dataset of labeled training data to solve a problem, they typically make use of services like Amazon's **AWS Mechanical Turk** [↗\(https://docs.aws.amazon.com/mturk/index.html\)](https://docs.aws.amazon.com/mturk/index.html), or 'MTurk' for short. Services like this obtain labels by paying people for each label they generate. In this way, a company can crowdsource the work to label the training data needed. The company simply uploads unlabeled training data like an image, and a "turker" will then provide a label for that image according to the instructions from the company. Depending on the problem the company is trying to solve, the

label for the image might be something as simple as the word "cat", or as complex as as boxes drawn around all the cats in the image.

Negative Examples

When creating a labeled dataset for a classification problem, it is worth noting that negative examples are just as important to be included in the dataset as positive examples. If our training data in the titanic dataset only contained data on passengers that all survived, no supervised learning algorithm would be able to learn how to predict if a passenger survived or died with any sort of accuracy. **Positive Examples** are data points that belong to the class we're training our model to recognize. For instance, let's pretend we're building a model to tell if a picture is of a cat or not. All the pictures of cats in our dataset would be positive examples. However, in order to build a good cat classifier, our dataset would also need to contain many different kinds of pictures that don't include cats. Intuitively, this makes sense -- if every picture that our model ever saw had a cat in it, then the only thing that model will learn is that everything is a cat. To truly learn what we need it to learn, this model will also need to learn what a cat *isn't*, by looking at pictures that don't include cats -- our **Negative Examples**. In this way, with a complex enough model and enough labeled training data, our classifier will eventually learn that the differentiating factor between images with positive labels and images with negative labels are the shapes and patterns common to cats, but not dogs (or other animals). In this way, supervised learning can be a bit tricky. For instance, if all of the negative examples in our cat classifier dataset are of cars and houses, then the model will almost certainly get a picture of a dog incorrect by predicting that the picture is of a cat. Why does this happen? Because the model hasn't seen a dog before, and therefore has no idea whether this fits. In this particular example, we can guess that any picture of a dog will look more like a cat than it would a house or car, which from the model's perspective means that this is probably a picture of a cat.

In summary, this part of supervised learning can often be more art than science -- when creating a dataset, make sure that your dataset contains enough negative examples, and that you are very thoughtful about what those negative examples actually contain!

Summary

In this lesson, we learned about *Supervised Learning*, and where it fits in relation to Machine Learning and Artificial Intelligence.

How do you feel about this lesson?



Have specific feedback?

[Tell us here! \(https://github.com/learn-co-curriculum/dsc-intro-to-supervised-learning-v2-1/issues/new/choose\)](https://github.com/learn-co-curriculum/dsc-intro-to-supervised-learning-v2-1/issues/new/choose)