[**So what are the Goals for this episode…**](#_bwt2kcnjxhdp) **2**

[**-----------**](#_rzy800y7x7h7) **6**

[**Parametric versus Non-parametric:  
Parametric and Nonparametric Machine Learning Algorithms**](#_yvej8a4m4has) **6**

[**---------------Branches of Machine Learning**](#_e3pmzyff3hwg) **10**

[Supervised Learning](#_aesga5v91c2m) 10

[---------------------](#_f5z64hg8mt4b) 16

[Unsupervised Learning](#_9q6x744o7wec) 16

[-------------------](#_uf1p07cafzr6) 20

[Semi-Supervised Learning:](#_av3g5hx5e6dh) 20

[-----------Reinforcement Learning](#_ep9q52rfsmo7) 21

Hello, my name is Ernesto Lee and this is my Machine Learning Course for Programmers and IT Professionals (Not Data Scientists).

You can also read about me on Ernesto.Net or LearningVoyage.com or just google me…

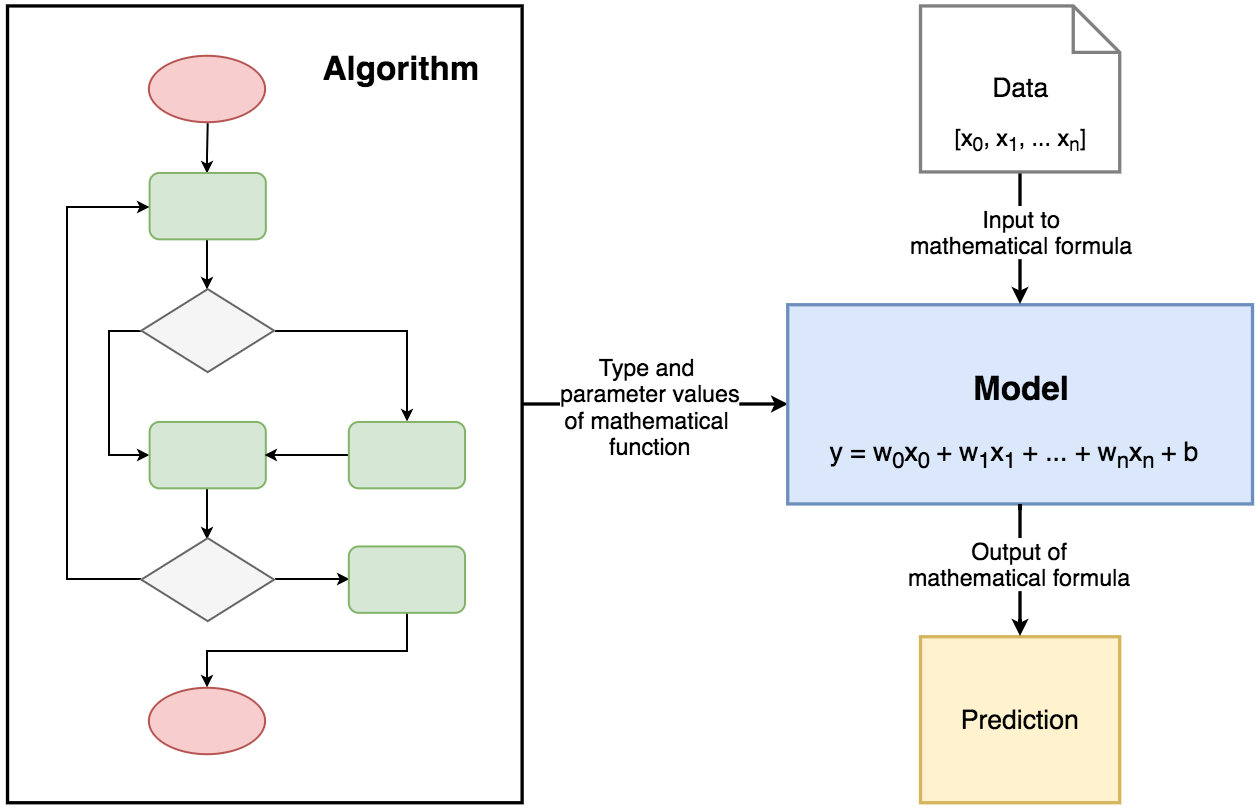
# **So what are the Goals for this episode…**

Let’s start with this… Getting started with Machine Learning is NOT easy. My grandmother - Thelma Elaine Eason - God rest her soul was one of the best teachers that I have ever known. She would always say that Learning only happens when you have a motivated instructor and a motivated student. For over 20 years I have been a high school teacher, college professor, and corporate trainer. Teaching and training is what I was born to do. **I am invested in your success.** With that, let’s start at the very beginning with Algorithms and Models.

**---------**

**Machine Learning model**

If we start at the ABCs of ML then that would lead us to begin with the concept of a “model”. The definition of a “model” which will appear quite often from now on. You will hear names like *Linear Regression*, *Logistic Regression*, *Decision Trees* etc. Those are all just the names of the algorithms that you will eventually learn but for now just know that they are just **theoretical concepts** that describe what to do in order to make a quality prediction. A model is slightly different from an algorithm. An algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are unambiguous specifications for performing calculation, data processing, automated reasoning, and other tasks. A Model is a **mathematical formula** which is the result of Machine Learning **algorithm implementation** (in the case of Data Science: a model is code). A model has measurable parameters that can be used for predictions. Models can be TRAINED by modifying their parameters (or weights or coefficients) in order to achieve better results. It is possible to say that models are representations of what a Machine Learning system has learned from the training data.



So let’s recap How machine learning algorithms work because this is a common principle that underlies all supervised machine learning algorithms… especially for predictive modeling. Really all that we are talking about is a mapping problem. This is what all supervised machine learning algorithms seek to solve.

Different machine learning algorithms represent different strategies for learning the mapping function.

Think of the Algorithm as a LEARNING FUNCTION. Machine learning algorithms are described as learning a target function (f) that best maps input variables (X) to an output variable (Y).

Y = f(X)

This is a general learning task where we would like to make predictions in the future (Y ) given new examples of input variables (X). We don’t know what the function (f) looks like or its form. If we did, we would use it directly and we would not need to learn it from data using machine learning algorithms. It is harder than you think. There is also error (e) that is independent of the input data (X).

Y = f(X) + e

This error might be caused by not having enough attributes to sufficiently describe the best mapping from X to Y . This error is called “the irreducible error” because no matter how good we get at estimating the target function (f), we can never reduce this error. This is to say, that the problem of learning a function from data is a difficult problem and this is the reason why the field of machine learning and machine learning algorithms exist.

The most common type of machine learning is to learn the mapping

Y = f(X)

So that we can make predictions of Y for new X. This is called predictive modeling or predictive analytics and our goal is to make the most accurate predictions possible.

Therefore, we are not really interested in the shape and form of the function (f) that we are learning, only that it makes accurate predictions. We could learn the mapping of Y = f(X) so that we could learn more about the relationship in the data and if we did this, this would be called statistical inference. If this were the goal, we would use simpler methods and value understanding the learned model and form of the function (f) above making accurate predictions.

When we learn a function (f) we are estimating its form from the data that we have available. And this estimate will have an error. It will not be a perfect estimate for the underlying hypothetical best mapping from Y given X. Much time in applied machine learning is spent attempting to improve the estimate of the underlying function and in turn improve the performance of the predictions made by the model.

Machine learning algorithms are techniques for estimating the target function (f) to predict the output variable (Y ) given input variables (X). Different representations make different assumptions about the form of the function being learned, such as whether it is linear or nonlinear.Different machine learning algorithms make different assumptions about the shape and structure of the function and how best to optimize a representation to approximate it. This is why it is so important to try a bunch of different algorithms on a machine learning problem, because we cannot know beforehand which approach will be best at estimating the structure of the underlying function we are trying to approximate.

So what did we learn? In this episode, you discovered the underlying principle that explains the objective of all machine learning algorithms for predictive modeling.

You learned that machine learning algorithms work to estimate the mapping function (f) of output variables (Y - the prediction) given input variables (X), or Y = f(X).

You also learned that different machine learning algorithms make different assumptions about the form of the underlying function.

You learned that when we don’t know much about the form of the target function we have to try a suite of different algorithms to see what works best.

You now know the fundamental principle that underlies all machine learning algorithms. In the next episode, you will discover the main classes and branches of machine learning algorithms: parametric and nonparametric algorithms.

# **-----------**

# **Parametric versus Non-parametric: Parametric and Nonparametric Machine Learning Algorithms**

What is a parametric machine learning algorithm and how is it different from a nonparametric machine learning algorithm? In this episode you will discover the difference between parametric

and nonparametric machine learning algorithms.

You will learn:

* That parametric machine learning algorithms simplify the mapping to a known functional form.
* That nonparametric algorithms can learn any mapping from inputs to outputs.
* That all algorithms can be organized into parametric or nonparametric groups.

Let’s get started with Parametric Machine Learning Algorithms.

Assumptions can greatly simplify the learning process, but it can also limit what can be learned.

Algorithms that simplify the function to a known form are called parametric machine learning algorithms.

A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model. No matter how much data you throw at a parametric model, it won’t change its mind about how man parameters it needs.

A parametric algorithm involve two steps:

1. Select a form for the function. In other words - this data looks linear so let's assume the form of the answer is a straight line.

2. Learn the coefficients for the function from the training data.

Remember that Universal Truth about all Parametric algos. You predict the form of the “solution” and then you look for the BEST coefficients.

An easy to understand functional form for the mapping function is a line, as is used in linear regression:

B0 + B1 × X1 + B2 × X2 = 0

Where B0, B1 and B2 are the coefficients of the line that control the intercept and slope, and X1 and X2 are two input variables. Assuming the functional form of a line greatly simplifies the learning process. Now, all we need to do is estimate the coefficients (B0, B1 and B2) of the line equation and we have a predictive model for the problem.

Often the assumed functional form is a linear combination of the input variables and as such parametric machine learning algorithms are often also called linear machine learning algorithms.

The problem is, the actual unknown underlying function may not be a linear function like a line.

It could be almost a line and require some minor transformation of the input data to work right.

Or it could be nothing like a line in which case the assumption is wrong and the approach will produce poor results.

Some more examples of parametric machine learning algorithms include:

* Logistic Regression
* Linear Regression obviously
* Linear Discriminant Analysis
* Perceptron

This is what I like about Parametric Machine Learning Algorithms:

They are Simpler: These methods are easier to understand and interpret results.

They have Faster Speed: Parametric models are very fast to learn from data.

They require Less Data: They do not require as much training data and can work well even if the fit to the data is not perfect.

These are some of the Limitations of Parametric Machine Learning Algorithms:

They are Constrained: By choosing a functional form these methods are highly constrained to the specified form.

They have Limited Complexity: The methods are more suited to simpler problems.

Often a Poor Fit: In practice the methods are unlikely to match the underlying mapping function.

Let’s move over to Nonparametric Machine Learning Algorithms..

These types of algorithms do not make strong assumptions about the form of the mapping function. By not making assumptions, they are free to learn any functional form from the training data.

Nonparametric methods are good when you have a lot of data and no prior knowledge, and when you don’t want to worry too much about choosing just the right features. In other words, you are given a bag of data and you are looking for patterns directly in the data. You don’t assume that it is of any specific form or shape.

Nonparametric methods seek to best fit the training data in constructing the mapping function, while maintaining some ability to generalize to unseen data. Because of that, they are able to fit a large number of functional forms. An easy to understand nonparametric model is the k-nearest neighbors algorithm that makes predictions based on the k most similar training patterns for a new data instance. The method does not assume anything about the form of the

mapping function other than patterns that are close are likely have a similar output variable.

Some more examples of popular nonparametric machine learning algorithms are:

* Decision Trees like CART and C4.5
* Naive Bayes
* Support Vector Machines
* Neural Networks

Benefits of Nonparametric Machine Learning Algorithms:

Flexibility: Capable of fitting a large number of functional forms.

Power: No assumptions (or weak assumptions) about the underlying function.

Performance: Can result in higher performance models for prediction.

These are some of the Limitations of Nonparametric Machine Learning Algorithms:

More data: Require a lot more training data to estimate the mapping function.

Slower: A lot slower to train as they often have far more parameters to train.

Overfitting: More of a risk to overfit the training data and it is harder to explain why specific predictions are made.

In this episode you have discovered

* the difference between parametric and nonparametric machine learning algorithms.
* You learned that parametric methods make large assumptions about the mapping of the input variables to the output variables and in turn are faster to train, require less data but may not be as powerful.
* You also learned that nonparametric methods make few or no assumptions about the target function and in turn require a lot more data, are slower to train and have a higher model complexity but can result in more powerful models.
* You now know the difference between parametric and nonparametric machine learning algorithms.

In the next episode you will discover another way to group machine learning algorithms by the way they learn: supervised and unsupervised learning.

# 

# **---------------Branches of Machine Learning**

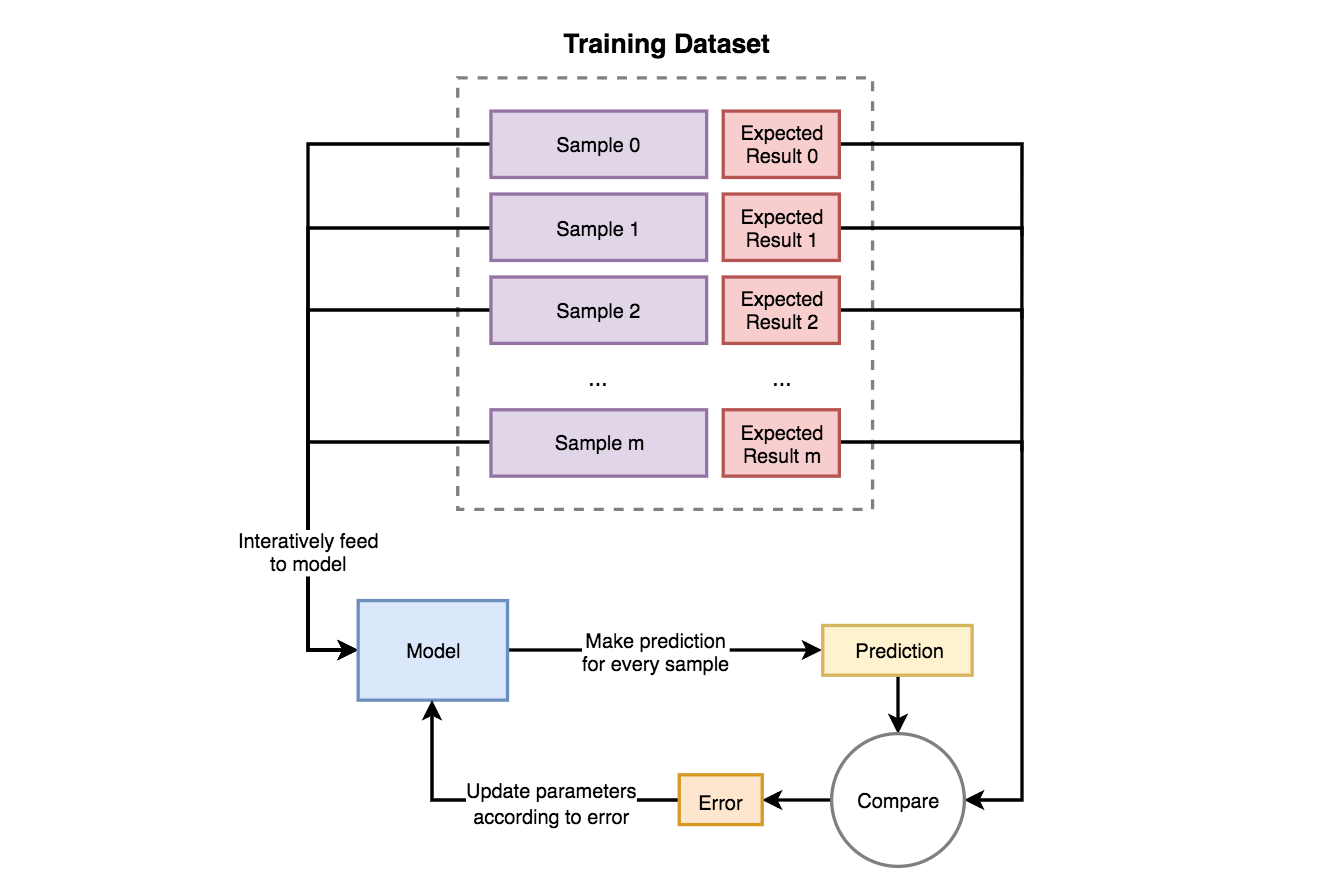
There are three most regularly listed categories of Machine Learning:

* Supervised Learning
* Unsupervised Learning
* Reinforcement Learning

## **Supervised Learning**

There is a group of algorithms that **require** **datasets which consists of example input-output pairs and this group of Algorithms are all considered Supervised Learning Algorithms**. Each pair consists of **data samples that are** used to make predictions and the expected outcome is called a **label**. Think about the word - “supervised”. The word “supervised” comes from the fact that labels need to be assigned to the data by a **human supervisor**.

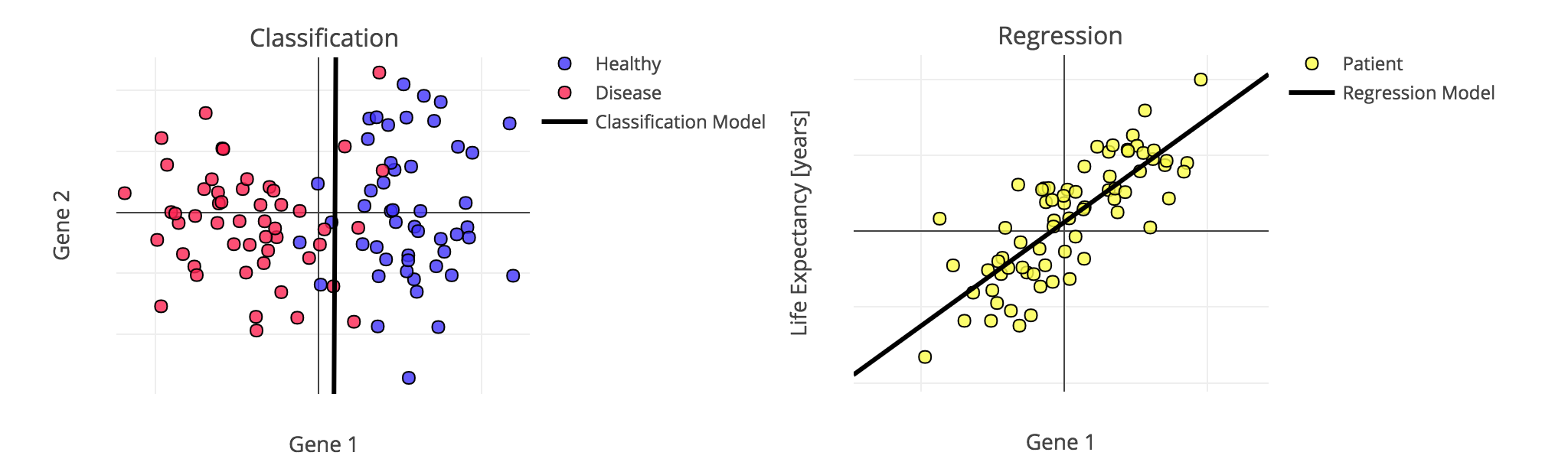
In the training process, samples are being iteratively fed to the model. For every sample, the model uses the current state of parameters and returns a prediction. The prediction is compared to the label (or the known “correct” answer), and the difference is called an error. **The error is a feedback for the model of what went wrong and how to update itself in order to decrease the error in future predictions.** This means that model will change the values of its parameters according to the algorithm based on which it was created.



Here is the important bit… Supervised Learning models are **trying to find parameter values that will allow them to perform well based on historical data**. Then they are **used for making predictions on new and unknown data**, that was not a part of training dataset.

There are two main problems that can be solved with Supervised Learning:

* **Classification** — process of *assigning category to input* data sample. Example usages: predicting whether a person is ill or not, detecting fraudulent transactions, face classifier.
* **Regression** - process of *predicting a continuous, numerical value* for input data sample. Example usages: assessing the house price, forecasting grocery store food demand, temperature forecasting.



So let me recap Supervised Learning:  
The majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (X) and an output variable (Y ) and you use an algorithm to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (Y ) for that data. It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We already know the correct answers, so we let the algorithm iteratively makes predictions on the training data and it is corrected by the teacher. The learning stops when the algorithm achieves an acceptable level of performance. Supervised learning problems can be further grouped into regression and classification problems.

Classification: A classification problem is when the output variable is a category, such as red or blue or disease versus no disease.

Regression: A regression problem is when the output variable is a real value, such as dollars or weight.

Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively.

Some popular examples of supervised machine learning algorithms are:

Linear regression for regression problems.

Random forest for classification and regression problems.

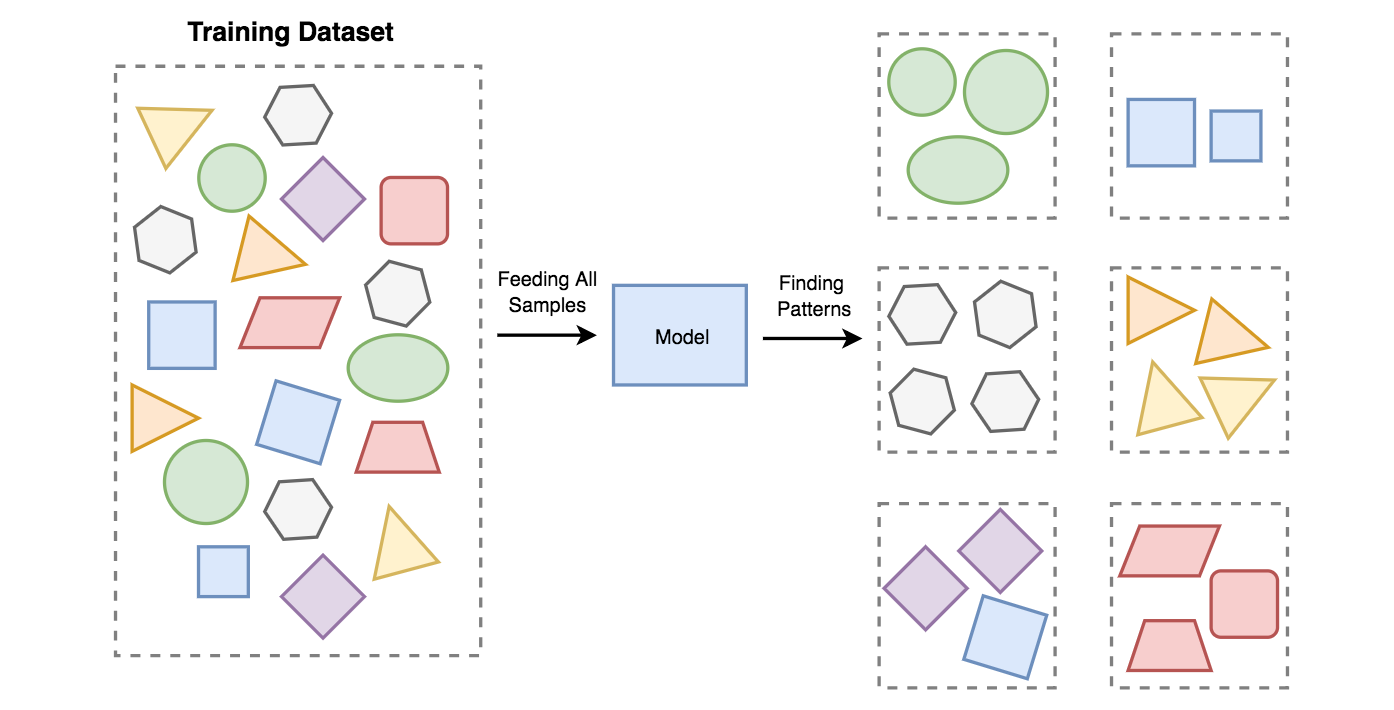
Support vector machines for classification problems.

This is just the overview. We will go into detail later in the course. Thank you.

## **---------------------**

## **Unsupervised Learning**

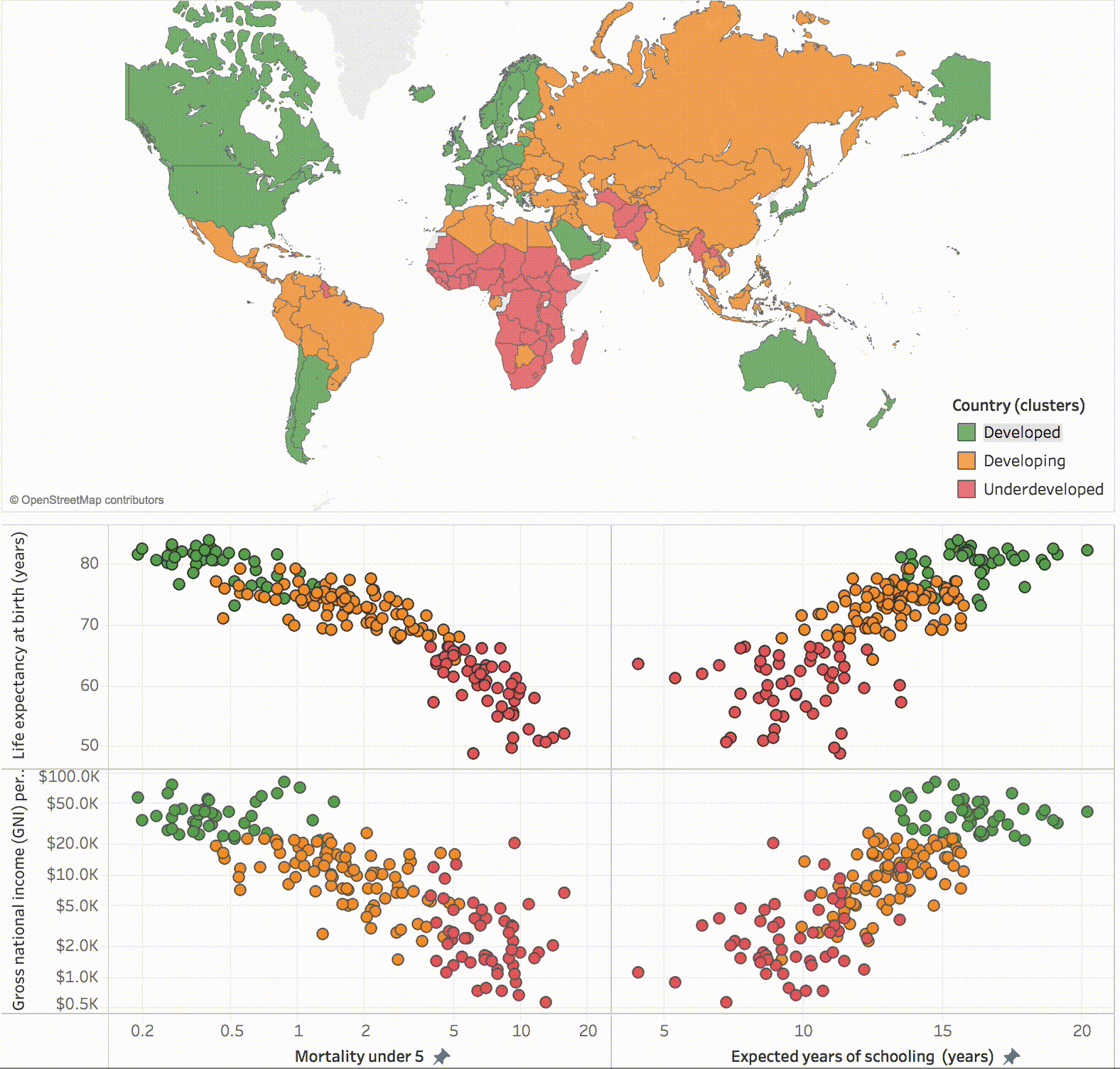
Is the Group of algorithms that try to **draw inferences from non-labeled data** (without reference to known or labeled outcomes). In Unsupervised Learning, **there are no correct answers**. Models based on this type of algorithms can be used for discovering unknown data patterns and data structure itself.



On your screen, you can see the Unsupervised Learning concept. All data is fed to the model and it produces an output on it’s own based on similarity between samples and algorithm used to create the model.

The most common applications of Unsupervised Learning are:

* **Pattern recognition and data clustering** - The Process of *dividing and grouping similar data* samples together. Groups are usually called clusters. Example usages: segmentation of supermarkets, user base segmentation, signal denoising.
* **Reducing data dimensionality** - Data dimension is the number of features needed to describe data sample. Dimensionality reduction is a *process of compressing features into so-called principal values* which conveys similar information concisely. By choosing only a few components, the amount of features is reduced and a small part of the data is lost in the process. Example usages: speeding up other Machine Learning algorithms by reducing numbers of calculations, finding a group of the most reliable features in data.



Dividing data from various countries around the world into three clusters representing Developed, Developing and Underdeveloped nations (source: [Tableau blog](https://www.tableau.com/about/blog/2016/7/uncover-patterns-your-data-tableau-10s-clustering-feature-56373)).

So let me recap Unsupervised Learning:  
Unsupervised learning is where you only have input data (X) and no corresponding output variables. The goal of unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

This is called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devices to discover and

present the interesting structure in the data. Unsupervised learning problems can be further grouped into clustering and association problems.

Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, things like people that buy A also tend to buy B.

Some popular examples of unsupervised learning algorithms are:

k-means for clustering problems.

Apriori algorithm for association rule learning problems.

## **-------------------**

## **Semi-Supervised Learning:**

Problems where you have a large amount of input data (X) and only some of the data is labeled (Y ) are called semi-supervised learning problems. These problems sit in between both supervised and unsupervised learning. A good example is a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled. Many real world machine learning problems fall into this area. This is because it can be expensive or time consuming to label data as it may require access to domain experts. Whereas unlabeled data is cheap and easy to collect and store.

You can use unsupervised learning techniques to discover and learn the structure in the input variables. You can also use unsupervised learning techniques to make best guess predictions for the unlabeled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data.

In this episode you learned the difference between supervised, unsupervised and semi-supervised learning. You now know that:

Supervised: All data is labeled and the algorithms learn to predict the output from the input data.

Unsupervised: All data is unlabeled and the algorithms learn to inherent structure from the input data.

Semi-supervised: Some data is labeled but most of it is unlabeled and a mixture of supervised and unsupervised techniques can be used.

You now know that you can group machine learning algorithms as supervised, unsupervised and semi-supervised learning. In the future episodes you will discover the two biggest sources of error when learning from data, namely bias and variance and the tension between these two concerns.

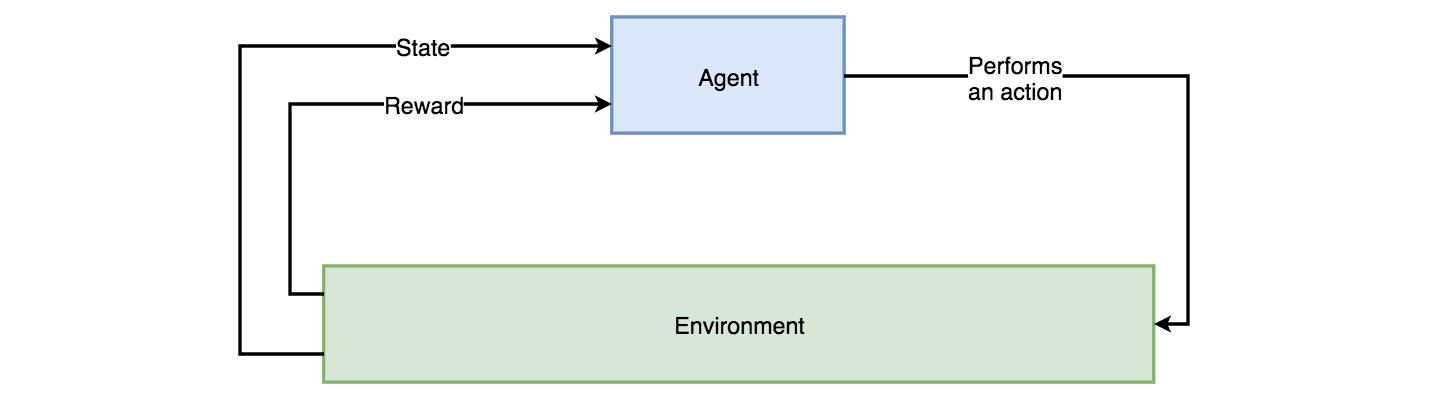
## 

## 

## **-----------Reinforcement Learning**

Is the Branch of Machine Learning algorithms which produces so-called **agents**. The agent role is slightly different than classic model. It’s to **receive information from the environment and react to it** by performing an **action**. The information is fed to an agent in the form of numerical data, called a **state,** which isstored and then used for choosing the right action. As a result, an agent receives a **reward** that can be either positive or negative. The reward is feedback that can be used by an agent to update its parameters.

The Training of an agent is a process of **trial and error**. It needs to find itself in various situations and get punished every time it takes the wrong action in order to learn. The goal of optimization can be set in many ways depending on the Reinforcement Learning approach e.g. based on *Value Function*, *Gradient Policy* or *Environment Model*.



In all cases, RL is always an Interaction between Agent and Environment.

There is a broad group of Reinforcement Learning applications. The Majority of them are the inventions that are regularly mentioned as most innovative accomplishments of AI.



Examples of solutions where Reinforcement Learning is used range from self-driving cars through various games such as Go, Chess, Poker or computer ones — Dota or Starcraft, to manufacturing.

Simulating the movement of 3D models is a complicated task. These models need to interact with different models in a given environment. Reinforcement Learning is becoming more actively used as a tool for solving this type of problem, as the results it produces seem very trustworthy for the human eye and RL algorithms are capable of automatically adjusting to rules describing the environment.