**Difference Between Algorithm and Model in Machine Learning**

Machine learning involves the use of machine learning **algorithms** and **models**.

For beginners, this is very confusing as often “*machine learning algorithm*” is used interchangeably with “*machine learning model*.” Are they the same thing or something different?

As a developer, your intuition with “*algorithms*” like sort algorithms and search algorithms will help to clear up this confusion.

In this post, you will discover the difference between machine learning “*algorithms*” and “*models*.”

After reading this post, you will know:

* Machine learning algorithms are procedures that are implemented in code and are run on data.
* Machine learning models are output by algorithms and are comprised of model data and a prediction algorithm.
* Machine learning algorithms provide a type of automatic programming where machine learning models represent the program.

**Overview**

This tutorial is divided into four parts; they are:

1. What Is an Algorithm in Machine Learning
2. What Is a Model in Machine Learning
3. Algorithm vs. Model Framework
4. Machine Learning Is Automatic Programming

**What Is an “*Algorithm*” in Machine Learning**

An “*algorithm*” in machine learning is a procedure that is run on data to create a machine learning “*model*.”

Machine learning algorithms perform “*pattern recognition*.” Algorithms “*learn*” from data, or are “*fit*” on a dataset.

There are many machine learning algorithms.

For example, we have algorithms for classification, such as k-nearest neighbors. We have algorithms for regression, such as linear regression, and we have algorithms for clustering, such as k-means.

Examples of machine learning algorithms:

* Linear Regression
* Logistic Regression
* Decision Tree
* Artificial Neural Network
* k-Nearest Neighbors
* k-Means

You can think of a machine learning algorithm like any other algorithm in computer science.

For example, some other types of algorithms you might be familiar with include bubble sort for sorting data and best-first for searching.

As such, machine learning algorithms have a number of properties:

* Machine learning algorithms can be described using math and pseudocode.
* The efficiency of machine learning algorithms can be analyzed and described.
* Machine learning algorithms can be implemented with any one of a range of modern programming languages.

For example, you may see machine learning algorithms described with pseudocode or [linear algebra](https://machinelearningmastery.com/linear-algebra-machine-learning-7-day-mini-course/) in research papers and textbooks. You may see the computational efficiency of a specific machine learning algorithm compared to another specific algorithm.

Academics can devise entirely new machine learning algorithms and machine learning practitioners can use standard machine learning algorithms on their projects. This is just like other areas of computer science where academics can devise entirely new sorting algorithms, and programmers can use the standard sorting algorithms in their applications.

You are also likely to see multiple machine learning algorithms implemented together and provided in a library with a standard application programming interface (API). A popular example is the scikit-learn library that provides implementations of many classification, regression, and clustering machine learning algorithms in Python.

**What Is a “*Model*” in Machine Learning**

A “*model*” in machine learning is the output of a machine learning algorithm run on data.

A model represents what was learned by a machine learning algorithm.

The model is the “*thing*” that is saved after running a machine learning algorithm on training data and represents the rules, numbers, and any other algorithm-specific data structures required to make predictions.

Some examples might make this clearer:

* The linear regression algorithm results in a model comprised of a vector of coefficients with specific values.
* The decision tree algorithm results in a model comprised of a tree of if-then statements with specific values.
* The neural network / backpropagation / gradient descent algorithms together result in a model comprised of a graph structure with vectors or matrices of weights with specific values.

A machine learning model is more challenging for a beginner because there is not a clear analogy with other algorithms in computer science.

For example, the sorted list output of a sorting algorithm is not really a model.

**The best analogy is to think of the machine learning model as a “*program*.”**

The machine learning model “*program*” is comprised of both data and a procedure for using the data to make a prediction.

For example, consider the linear regression algorithm and resulting model. The model is comprised of a vector of coefficients (data) that are multiplied and summed with a row of new data taken as input in order to make a prediction (prediction procedure).

We save the data for the machine learning model for later use.

We often use the prediction procedure for the machine learning model provided by a machine learning library. Sometimes we may implement the prediction procedure ourselves as part of our application. This is often straightforward to do given that most prediction procedures are quite simple.

**Algorithm vs. Model Framework**

So now we are familiar with a machine learning “*algorithm*” vs. a machine learning “*model*.”

Specifically, an algorithm is run on data to create a model.

* Machine Learning => Machine Learning Model

We also understand that a model is comprised of both data and a procedure for how to use the data to make a prediction on new data. You can think of the procedure as a prediction algorithm if you like.

* Machine Learning Model == Model Data + Prediction Algorithm

This division is very helpful in understanding a wide range of algorithms.

For example, most algorithms have all of their work in the “*algorithm*” and the “*prediction algorithm*” does very little.

Typically, the algorithm is some sort of optimization procedure that minimizes error of the model (data + prediction algorithm) on the training dataset. The linear regression algorithm is a good example. It performs an optimization process (or is solved analytically using linear algebra) to find a set of weights that minimize the sum squared error on the training dataset.

**Linear Regression:**

* **Algorithm**: Find set of coefficients that minimize error on training dataset
* **Model**:
  + **Model Data**: Vector of coefficients
  + **Prediction Algorithm**: Multiple and sum coefficients with input row

Some algorithms are trivial or even do nothing, and all of the work is in the model or prediction algorithm.

The k-nearest neighbor algorithm has no “*algorithm*” other than saving the entire training dataset. The model data, therefore, is the entire training dataset and all of the work is in the prediction algorithm, i.e. how a new row of data interacts with the saved training dataset to make a prediction.

**k-Nearest Neighbors**

* **Algorithm**: Save training data.
* **Model**:
  + **Model Data**: Entire training dataset.
  + **Prediction Algorithm**: Find k most similar rows and average their target variable.

You can use this breakdown as a framework to understand any machine learning algorithm.

**What is your favorite algorithm?**  
Can you describe it using this framework in the comments below?

Do you know an algorithm that does not fit neatly into this breakdown?

**Machine Learning Is Automatic Programming**

We really just want a machine learning “*model*” and the “*algorithm*” is just the path we follow to get the model.

Machine learning techniques are used for problems that cannot be solved efficiently or effectively in other ways.

For example, if we need to classify emails as spam or not spam, we need a software program to do this.

We could sit down, manually review a ton of email, and write if-statements to perform this task. People have tried. It turns out that this approach is slow, fragile, and not very effective.

Instead, we can use machine learning techniques to solve this problem. Specifically, an algorithm like [Naive Bayes](https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/) can learn how to classify email messages as spam and not spam from a large dataset of historical examples of email.

We don’t want “*Naive Bayes*.” We want the model that Naive Bayes gives is that we can use to classify email (the vectors of probabilities and prediction algorithm for using them). We want the model, not the algorithm used to create the model.

In this sense, the machine learning model is a program automatically written or created or learned by the machine learning algorithm to solve our problem.

As developers, we are less interested in the “*learning*” performed by machine learning algorithms in the artificial intelligence sense. We don’t care about simulating learning processes. Some people may be, and it is interesting, but this is not why we are using machine learning algorithms.

Instead, we are more interested in the automatic programming capability offered by machine learning algorithms. We want an effective model created efficiently that we can incorporate into our software project.

Machine learning algorithms perform automatic programming and machine learning models are the programs created for us.

**Summary**

In this post, you discovered the difference between machine learning “*algorithms*” and “*models*.”

Specifically, you learned:

* Machine learning algorithms are procedures that are implemented in code and are run on data.
* Machine learning models are output by algorithms and are comprised of model data and a prediction algorithm.
* Machine learning algorithms provide a type of automatic programming where machine learning models represent the program.