CHAPTER

1

What Is Machine Learning?

Let's start at the beginning, looking at what machine learning actually is, its history, and where it is used in industry. This chapter also describes some of the software used throughout the book so you can have everything installed and be ready to get working on the practical things.

History of Machine Learning

So, what is the definition of machine learning? Over the last six decades, several pioneers of the industry have worked to steer us in the right direction.

Alan Turing

In his 1950 paper, "Computing Machinery and Intelligence," Alan Turing asked, "Can machines think?" (See `www.csee.umbc.edu/courses/471/papers/turing.pdf` for the full paper.) The paper describes the "Imitation Game," which involves three participants—a human acting as a judge, another human, and a computer that is attempting to convince the judge that it is human. The judge would type into a terminal program to "talk" to the other two participants. Both the human and the computer would respond, and the judge would decide which response came from the computer. If the judge couldn't consistently tell the difference between the human and computer responses then the computer won the game.

The test continues today in the form of the Loebner Prize, an annual competition in artificial intelligence. The aim is simple enough: Convince the judges that they are chatting to a human instead of a computer chat bot program.

Arthur Samuel

In 1959, Arthur Samuel defined machine learning as, "[A] Field of study that gives computers the ability to learn without being explicitly programmed." Samuel is credited with creating one of the self-learning computer programs with his work at IBM. He focused on games as a way of getting the computer to learn things.

The game of choice for Samuel was checkers because it is a simple game but requires strategy from which the program could learn. With the use of alpha-beta evaluation pruning (eliminating nodes that do not need evaluating) and minimax (minimizing the loss for the worst case) strategies, the program would discount moves and thus improve costly memory performance of the program.

Samuel is widely known for his work in artificial intelligence, but he was also noted for being one of the first programmers to use hash tables, and he certainly made a big impact at IBM.

Tom M. Mitchell

Tom M. Mitchell is the Chair of Machine Learning at Carnegie Mellon University. As author of the book *Machine Learning* (McGraw-Hill, 1997), his definition of machine learning is often quoted:

> **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 the experience E.**

The important thing here is that you now have a set of objects to define machine learning:

- Task (T), either one or more
- Experience (E)
- Performance (P)

So, with a computer running a set of tasks, the experience should be leading to performance increases.

Summary Definition

Machine learning is a branch of artificial intelligence. Using computing, we design systems that can learn from data in a manner of being trained. The systems might learn and improve with experience, and with time, refine a model that can be used to predict outcomes of questions based on the previous learning.

Algorithm Types for Machine Learning

There are a number of different algorithms that you can employ in machine learning. The required output is what decides which to use. As you work through the chapters, you'll see the different algorithm types being put to work. Machine learning algorithms characteristically fall into one of two learning types: supervised or unsupervised learning.

Supervised Learning

Supervised learning refers to working with a set of labeled training data. For every example in the training data you have an input object and an output object. An example would be classifying Twitter data. (Twitter data is used a lot in the later chapters of the book.) Assume you have the following data from Twitter; these would be your input data objects:

```
Really loving the new St Vincent album!
#fashion I'm selling my Louboutins! Who's interested? #louboutins
I've got my Hadoop cluster working on a load of data. #data
```

In order for your supervised learning classifier to know the outcome result of each tweet, you have to manually enter the answers; for clarity, I've added the resulting output object at the start of each line.

```
music    Really loving the new St Vincent album!
clothing    #fashion I'm selling my Louboutins! Who's interested? #louboutins
bigdata    I've got my Hadoop cluster working on a load of data. #data
```

Obviously, for the classifier to make any sense of the data, when run properly, you have to work manually on a lot more input data. What you have, though, is a training set that can be used for later classification of data.

There are issues with supervised learning that must be taken into account. The *bias-variance dilemma* is one of them: how the machine learning model performs accurately using different training sets. High bias models contain restricted learning sets, whereas high variance models learn with complexity against noisy training data. There's a trade-off between the two models. The key is where to settle with the trade-off and when to apply which type of model.

Unsupervised Learning

On the opposite end of this spectrum is unsupervised learning, where you let the algorithm find a hidden pattern in a load of data. With unsupervised learning there is no right or wrong answer; it's just a case of running the machine learning algorithm and seeing what patterns and outcomes occur.

Unsupervised learning might be more a case of data mining than of actual learning. If you're looking at clustering data, then there's a good chance you're going to spend a lot of time with unsupervised learning in comparison to something like artificial neural networks, which are trained prior to being used.

The Human Touch

Outcomes will change, data will change, and requirements will change. Machine learning cannot be seen as a write-it-once solution to problems. Also, it requires human hands and intuition to write these algorithms. Remember that Arthur Samuel's checkers program basically improved on what the human had already taught it. The computer needed a human to get it started, and then it built on that basic knowledge. It's important that you remember that.

Throughout this book I talk about the importance of knowing what question you are trying to answer. The question is the cornerstone of any data project, and it starts with having open discussions and planning. (Read more about this in Chapter 2, "Planning for Machine Learning.")

It's only in rare circumstances that you can throw data at a machine learning routine and have it start to provide insight immediately.

Uses for Machine Learning

So, what can you do with machine learning? Quite a lot, really. This section breaks things down and describes how machine learning is being used at the moment.

Software

Machine learning is widely used in software to enable an improved experience with the user. With some packages, the software is learning about the user's behavior after its first use. After the software has been in use for a period of time it begins to predict what the user wants to do.

Spam Detection

For all the junk mail that gets caught, there's a good chance a Bayesian classification filter is doing the work to catch it. Since the early days of SpamAssassin to Google's work in Google Mail, there's been some form of learning to figure out whether a message is good or bad.

Spam detection is one of the classic uses of machine learning, and over time the algorithms have gotten better and better. Think about the e-mail program that you use. When it sees a message it thinks is junk, it asks you to confirm whether it is junk or isn't. If you decide that the message is spam, the system learns from that message and from the experience. Future messages will, hopefully, be treated correctly from then on.

Voice Recognition

Apple's Siri service that is on many iOS devices is another example of software machine learning. You ask Siri a question, and it works out what you want to do. The result might be sending a tweet or a text message, or it could be setting a calendar appointment. If Siri can't work out what you're asking of it, it performs a Google search on the phrase you said.

Siri is an impressive service that uses a device and cloud-based statistical model to analyze your phrase and the order of the words in it to come up with a resulting action for the device to perform.

Stock Trading

There are lots of platforms that aim to help users make better stock trades. These platforms have to do a large amount of analysis and computation to make recommendations. From a machine learning perspective, decisions are being made for you on whether to buy or sell a stock at the current price. It takes into account the historical opening and closing prices and the buy and sell volumes of that stock.

With four pieces of information (the low and high prices plus the daily opening and closing prices) a machine learning algorithm can learn trends for the stock. Apply this with all stocks in your portfolio, and you have a system to aid you in the decision whether to buy or sell.

Bitcoins are a good example of algorithmic trading at work; the virtual coins are bought and sold based on the price the market is willing to pay and the price at which existing coin owners are willing to sell.

The media is interested in the high-speed variety of algorithmic trading. The ability to perform many thousands of trades each second based on algorithmic prediction is a very compelling story. A huge amount of money is poured into these systems and how close they can get the machinery to the main stock trading exchanges. Milliseconds of network latency can cost the trading house millions in trades if they aren't placed in time.

About 70 percent of trades are performed by machine and not by humans on the trading floor. This is all very well when things are going fine, but when a problem occurs it can be minutes before the fault is noticed, by which time many trades have happened. The flash crash in May 2010, when the Dow Jones industrial average dove 600 points, is a good example of when this problem occurred.

Robotics

Using machine learning, robots can acquire skills or learn to adapt to the environment in which they are working. Robots can acquire skills such as object placement, grasping objects, and locomotion skills through either automated learning or learning via human intervention.

With the increasing amount of sensors within robotics, other algorithms could be employed outside of the robot for further analysis.

Medicine and Healthcare

The race is on for machine learning to be used in healthcare analytics. A number of startups are looking at the advantages of using machine learning with Big Data to provide healthcare professionals with better-informed data to enable them to make better decisions.

IBM's famed Watson supercomputer, once used to win the television quiz program *Jeopardy* against two human contestants, is being used to help doctors. Using Watson as a service on the cloud, doctors can access learning on millions of pages of medical research and hundreds of thousands of pieces of information on medical evidence.

With the number of consumers using smartphones and the related devices for collating a range of health information—such as weight, heart rate, pulse, pedometers, blood pressure, and even blood glucose levels—it's now possible to track and trace user health regularly and see patterns in dates and times. Machine learning systems can recommend healthier alternatives to the user via the device.

Although it's easy enough to analyze data, protecting the privacy of user health data is another story. Obviously, some users are more concerned about how their data is used, especially in the case of it being sold to third-party companies. The increased volume of analytics in healthcare and medicine is new, but the privacy debate will be the deciding factor about how the algorithms will ultimately be used.

Advertising

For as long as products have been manufactured and services have been offered, companies have been trying to influence people to buy their products. Since 1995, the Internet has given marketers the chance to advertise directly to our screens without needing television or large print campaigns. Remember the thought of cookies being on our computers with the potential to track us? The race to disable cookies from browsers and control who saw our habits was big news at the time.

Log file analysis is another tactic that advertisers use to see the things that interest us. They are able to cluster results and segment user groups according to who may be interested in specific types of products. Couple that with mobile location awareness and you have highly targeted advertisements sent directly to you.

There was a time when this type of advertising was considered a huge invasion of privacy, but we've gradually gotten use to the idea, and some people are even happy to "check in" at a location and announce their arrival. If you're thinking your friends are the only ones watching, think again. In fact, plenty of companies are learning from your activity. With some learning and analysis, advertisers can do a very good job of figuring out where you'll be on a given day and attempt to push offers your way.

Retail and E-Commerce

Machine learning is heavily used in retail, both in e-commerce and bricks-and-mortar retail. At a high level, the obvious use case is the loyalty card. Retailers that issue loyalty cards often struggle to make sense of the data that's coming back to them. Because I worked with one company that analyzes this data, I know the pain that supermarkets go through to get insight.

UK supermarket giant Tesco is the leader when it comes to customer loyalty programs. The Tesco Clubcard is used heavily by customers and gives Tesco a great view of customer purchasing decisions. Data is collected from the point of sale (POS) and fed back to a data warehouse. In the early days of the Clubcard, the data couldn't be mined fast enough; there was just too much. As processing methods improved over the years, Tesco and marketing company Dunn Humby have developed a good strategy for understanding customer behavior and shopping habits and encouraging customers to try products similar to their usual choices.

An American equivalent is Target, which runs a similar sort of program that tracks every customer engagement with the brand, including mailings, website visits, and even in-store visits. From the data warehouse, Target can fine-tune how to get the right communication method to the right customers in order for them to react to the brand. Target learned that not every customer wants an e-mail or an SMS message; some still prefer receiving mail via the postal service.

The uses for machine learning in retail are obvious: Mining baskets and segmenting users are key processes for communicating the right message to the customer. On the other hand, it can be too accurate and cause headaches. Target's "baby club" story, which was widely cited in the press as a huge privacy danger in Big Data, showed us that machine learning can easily determine that we're creatures of habit, and when those habits change they will get noticed.

TARGET'S PRIVACY ISSUE

Target's statistician, Andrew Pole, analyzed basket data to see whether he could determine when a customer was pregnant. A select number of products started to show up in the analysis, and Target developed a pregnancy prediction score. Coupons were sent to customers who were predicted to be pregnant according to the newly mined score. That was all very well until the father of a teenage girl contacted his local store to complain about the baby coupons that were being sent to his daughter. It turned out that Target predicted the girl's pregnancy before she had told her father that she was pregnant.

For all the positive uses of machine learning, there are some urban myths, too. For example, you might have heard the "beer and diapers" story associated with Walmart and other large retailers. The idea is that the sales of beer and diapers both increase on Fridays, suggesting that mothers were going out and dads would stock up on beer for themelves and diapers for the little ones they were looking after. It turned out to be a myth, but this still doesn't stop marketing companies from wheeling out the story (and believing it's true) to organizations who want to learn from their data.

Another myth is that the heavy metal band Iron Maiden would mine bittorrent data to figure out which countries were illegally downloading their songs and then fly to those locations to play concerts. That story got the marketers and media very excited about Big Data and machine learning, but sadly it's untrue. That's not to say that these things can't happen someday; they just haven't happened yet.

Gaming Analytics

We've already established that checkers is a good candidate for machine learning. Do you remember those old chess computer games with the real plastic pieces? The human player made a move and then the computer made a move. Well, that's a case of machine learning planning algorithms in action. Fast-forward a few decades (the chess computer still feels like yesterday to me) to today when the console market is pumping out analytics data every time you play your favorite game.

Microsoft has spent time studying the data from Halo 3 to see how players perform on certain levels and also to figure out when players are using cheats. Fixes have been created based on the analysis of data coming back from the consoles.

Microsoft also worked on Drivatar, which is incorporated into the driving game Forza Motorsport. When you first play the game, it knows nothing about your driving style. Over a period of practice laps the system learns your style, consistency, exit speeds on corners, and your positioning on the track. The sampling happens over three laps, which is enough time to see how your profile behaves. As time progresses the system continues to learn from your driving patterns. After you've let the game learn your driving style the game opens up new levels and lets you compete with other drivers and even your friends.

If you have children, you might have seen the likes of Nintendogs (or cats), a game in which a person is tasked with looking after an on-screen pet. (Think Tamagotchi, but on a larger scale.) Algorithms can work out when the pet needs to play, how to react to the owner, and how hungry the pet is.

It's still the early days of game companies putting machine learning into infrastructure to make the games better. With more and more games appearing on small devices, such as those with the iOS and Android platforms, the real learning is in how to make players come back and play more and more. Analysis can be performed about the "stickiness" of the game—do players return to play again or do they drop off over a period of time in favor of something else? Ultimately there's a trade-off between the level of machine learning and gaming performance, especially in smaller devices. Higher levels of machine learning require more memory within the device. Sometimes you have to factor in the limit of what you can learn from within the game.

The Internet of Things

Connected devices that can collate all manner of data are sprouting up all over the place. Device-to-device communication is hardly new, but it hadn't really hit the public minds until fairly recently. With the low cost of manufacture and distribution, now devices are being used in the home just as much as they are in industry.

Uses include home automation, shopping, and smart meters for measuring energy consumption. These things are in their infancy, and there's still a lot of concern on the security aspects of these devices. In the same way mobile device location is a concern, companies can pinpoint devices by their unique IDs and eventually associate them to a user.

On the plus side, the data is so rich that there's plenty of opportunity to put machine learning in the heart of the data and learn from the devices' output. This may be as simple as monitoring a house to sense ambient temperature—for example, is it too hot or too cold?

It's very early days for the Internet of things, but there's a lot of groundwork happening that is leading to some interesting outcomes. With the likes of Arduino and Raspberry Pi computers, it's relatively cheap to get started measuring the likes of motion, temperature, and sound and then extracting the data for analysis, either after it's been collated or in real time.

Languages for Machine Learning

This book uses the Java programming language for the working examples. The reasons are simple: It's a widely used language, and the libraries are well supported. Java isn't the only language to be used for machine learning—far from it. If you're working for an existing organization, you may be restricted to the languages used within it.

With most languages, there is a lot of crossover in functionality. With the languages that access the Java Virtual Machine (JVM) there's a good chance that you'll be accessing Java-based libraries. There's no such thing as one language being "better" than another. It's a case of picking the right tool for the job. The following sections describe some of the other languages that you can use for machine learning.

Python

The Python language has increased in usage, because it's easy to learn and easy to read. It also has some good machine learning libraries, such as scikit-learn, PyML, and pybrain. Jython was developed as a Python interpreter for the JVM, which may be worth investigating.

R

R is an open source statistical programming language. The syntax is not the easiest to learn, but I do encourage you to have a look at it. It also has a large number of machine learning packages and visualization tools. The RJava project allows Java programmers to access R functions from Java code. For a basic introduction to R, have a look at Chapter 12.

Matlab

The Matlab language is used widely within academia for technical computing and algorithm creation. Like R, it also has a facility for plotting visualizations and graphs.

Scala

A new breed of languages is emerging that takes advantage of Java's runtime environment, which potentially increases performance, based on the threading architecture of the platform. Scala (which is an acronym for *Sca*lable *La*nguage) is one of these, and it is being widely used by a number of startups.

There are machine learning libraries, such as ScalaNLP, but Scala can access Java jar files, and it can also implement the likes of Classifier4J and Mahout, which are covered in this book. It's also core to the Apache Spark project, which is covered in Chapter 11.

Clojure

Another JVM-based language, Clojure, is based on the Lisp programming language. It's designed for concurrency, which makes it a great candidate for machine learning applications on large sets of data.

Ruby

Many people know about the Ruby language by association with the Ruby On Rails web development framework, but it's also used as a standalone language. The best way to integrate machine learning frameworks is to look at JRuby, which is a JVM-based alternative that enables you to access the Java machine learning libraries.

Software Used in This Book

The hands-on elements in the book use a number of programs and packages to get the algorithms and machine learning working.

To keep things easy, I strongly advise that you create a directory on your system to install all these packages. I'm going to call mine `mlbook`:

```
$mkdir ~/mlbook
$cd ~/mlbook
```

Checking the Java Version

As the programs used in the book rely on Java, you need to quickly check the version of Java that you're using. The programs require Java 1.6 or later. To check your version, open a terminal window and run the following:

```
$ java -version
java version "1.7.0_40"
Java(TM) SE Runtime Environment (build 1.7.0_40-b43)
Java HotSpot(TM) 64-Bit Server VM (build 24.0-b56, mixed mode)
```

If you are running a version older than 1.6, then you need to upgrade your Java version. You can download the current version from `www.oracle.com/technetwork/java/javase/downloads/index.html`.

Weka Toolkit

Weka (Waikato Environment for Knowledge Acquisition) is a machine learning and data mining toolkit written in Java by the University of Waikato in New Zealand. It provides a suite of tools for learning and visualization via the supplied workbench program or the command line. Weka also enables you to retrieve data from existing data sources that have a JDBC driver. With Weka you can do the following:

- Preprocessing data
- Clustering
- Classification
- Regression
- Association rules

The Weka toolkit is widely used and now supports the Big Data aspects by interfacing with Hadoop for clustered data mining.

You can download Weka from the University of Waikato website at `www.cs.waikato.ac.nz/ml/weka/downloading.html`. There are versions of Weka available for Linux, Mac OSX, and Windows. To install Weka on Linux, you just need to unzip the supplied file to a directory. On Mac OSX and Windows, an installer program is supplied that will unzip all the required files for you.

Mahout

The Mahout machine learning libraries are an open source project that are part of the Apache project. The key feature of Mahout is its *scalability*; it works either on a single node or a cluster of machines. It has tight integration with the Hadoop Map/Reduce paradigm to enable large-scale processing.

Mahout supports a number of algorithms including

- Naive Bayes Classifier
- K Means Clustering
- Recommendation Engines
- Random Forest Decision Trees
- Logistic Regression Classifier

There's no workbench in Mahout like there is in the Weka toolkit, but the emphasis is on integrating machine learning library code within your projects. There are a wealth of examples and ready-to-run programs that can be used with your existing data.

You can download Mahout from `www.apache.org/dyn/closer.cgi/mahout/`. As Mahout is platform independent, there's one download that covers all the operating systems. To install the download, all you have to do is unzip Mahout into a directory and update your path to find the executable files.

SpringXD

Whereas Weka and Mahout concentrate on algorithms and producing the knowledge you need, you must also think about acquiring and processing data.

Spring XD is a "data ingestion engine" that reads in, processes, and stores raw data. It's highly customizable with the ability to create processing units. It also integrates with all the other tools mentioned in this chapter.

Spring XD is relatively new, but it's certainly useful. It not only relates to Internet-based data, it can also ingest network and system messages across a cluster of machines.

You can download the Spring XD distribution from `http://projects.spring.io/spring-xd/`. The link for the zip file is in the Quick Start section.

After the zip file has downloaded you need to unzip the distribution into a directory. For a detailed walkthrough of using Spring XD, read Chapter 9, "Machine Learning in Real Time with Spring XD."

Hadoop

Unless you've been living on some secluded island without power and an Internet connection, you will have heard about the savior of Big Data: Hadoop. Hadoop is very good for processing Big Data, but it's not a required tool. In this book, it comes into play in Chapter 10, "Machine Learning as a Batch Process."

Hadoop is a framework for processing data in parallel. It does this using the MapReduce pattern, where work is divided into blocks and is distributed across a cluster of machines. You can use Hadoop on a single machine with success; that's what this book covers.

There are two versions of Hadoop. This book uses version 1.2.1.

The Apache Foundation runs a series of mirror download servers and refers you to the ones relevant to your location. The main download page is at `www.apache.org/dyn/closer.cgi/hadoop/common/`.

After you have picked your mirror site, navigate your way to hadoop-1.2.1 releases and download `hadoop-1.2.1-bin.tar.gz`. Unzip and untar the distribution to a directory.

If you are running a Red Hat or Debian server, you can download the respective `.rpm` or `.deb` files and install them via the package installer for your operating system. If preferred, Debian and Ubuntu users can install Hadoop with the `apt-get` or `yum` command.

Using an IDE

Some discussions seem to spark furious debate in certain circles—for example, favorite actor/actress, best football team, and best integrated development environment (IDE).

I'm an Eclipse user. I'm also an IDEA user, and I have NetBeans as well. Basically, I use all three. There's no hard rule that IDE you should use, as they all do the same thing very well. The examples in this book use Eclipse (Juno release).

Data Repositories

One question that comes up again and again in my classes is "Where can I get data?" There are a few answers to this question, but the best answer depends on what you are trying to learn.

Data comes in all shapes and sizes, which is something discussed further in the next chapter. I strongly suggest that you take some time to hunt around the Internet for different data sets and look through them. You'll get a feel for how these things are put together. Sometimes you'll find comma separated variable (CSV) data, or you might find JSON or XML data.

Remember, some of the best learning comes from playing with the data. Having a question in mind that you are trying to answer with the data is a good start (and something you will see me refer to a number of times in this book), but learning comes from experimentation and improvement on results. So, I'm all for playing around with the data first and seeing what works. I hail from a very pragmatic background when it comes to development and learning. Although the majority of publications about machine learning have come from people with academic backgrounds—and I fully endorse and support them—we shouldn't discourage learning by doing.

The following sections describe some places where you can get plenty of data with which to play.

UC Irvine Machine Learning Repository

This machine learning repository consists of more than 270 data sets. Included in these sets are notes on the variable name, instances, and tasks the data would be associated with. You can find this repository at `http://archive.ics.uci.edu/ml/datasets`.

Infochimps

The data marketplace at Infochimps has been around for a few years. Although the company has expanded to cloud-based offerings, the data is still available to download at `www.infochimps.com/datasets`.

Kaggle

The competitions that Kaggle run have gained a lot of interest over the last couple of years. The 101 section on the site offers some data sets with which to experiment. You can find them at `www.kaggle.com/competitions`.

Summary

This chapter looked at what machine learning is, how it can be applied to different areas of business, and what tools you need to follow along with the remainder of the book.

Chapter 2 introduces you to planning for machine learning. It covers data science teams, cleaning, and different methods of processing data.