CHAPTER 4

Machine Learning—Basics

Taking our AI discussion further from Chapter 3, we will discuss here further a special type of artificial intelligence called machine learning.

■ **Note** The primary audience for this book is business users and higher management (both business corporations and IT service providers). Still, a basic knowledge of artificial intelligence and machine learning concepts will help provide better understanding for cognitive computing concepts in later chapters.

Machine Learning

Machine learning is a special type of AI. This is the ability of machines to learn and work on new problem dimensions without being explicitly programmed to do so. The programs are changed as they are exposed to new data and scenarios.

Machine learning plays a very important role in the so-called industry revolution 4.0.

This revolution is centered around IoT (Internet of Things) and cognitive computing. Machine learning has an important role to play here, as this becomes an essential subcomponent in this ecosystem.

Google and lot of other companies are now even able to provide machine learning as a cloud service. This is in line with preparations to move to the industry 4.0 level: see the various industry generations in Figure 4-1.

R. Kumar, *Machine Learning and Cognition in Enterprises*,
https://doi.org/10.1007/978-1-4842-3069-5_4

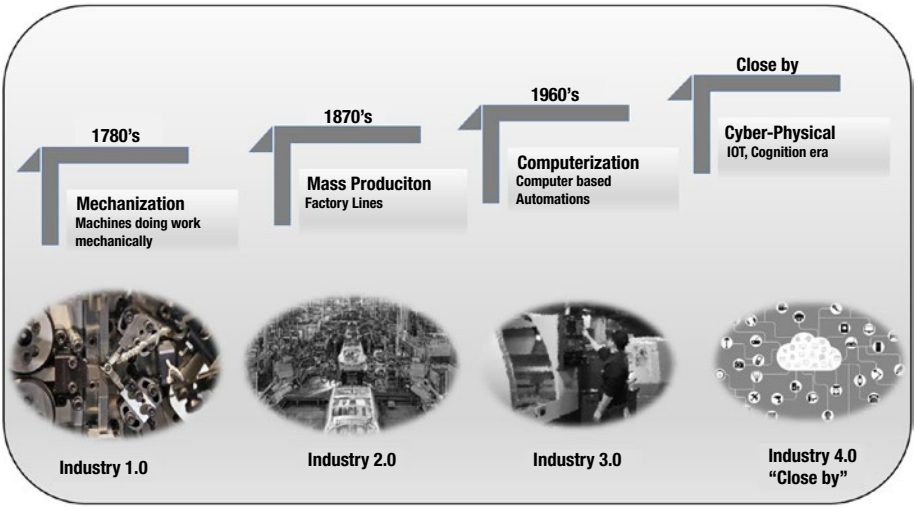

Figure 4-1. *Industry generations*

Machine learning has some similarity to data mining or pattern analysis. These activities help to search through data points to generate outcomes as outputs. So, the primary action for these functionalities is to analyze and understand data and then to use it to comprehend outcomes or outputs for users. Similarly, machine learning also keeps on understanding the data it comes in touch with while processing, but the result of such findings is not an outcome for the user but a continuous process of adjusting and updating the artificial intelligence program.

So, to compare patterns of basic activities between machine learning and data mining, in principle they both sift through mounds of data to get some coordinates fixed at desired outcome points. But the purpose of data mining is to display outwardly the essence of such mining activity with "fixed" outcome purposes and to help users see the patterns found as graphs or tables, while that of machine learning is to internally churn and derive multiple meanings out of it and internally consume it to evolve its algorithm to help optimize or upgrade different output somewhere else.

Therefore, data points consumed by data mining are the ones outputted as result, while in machine learning, the data points consumed might not be same data which is required to be output directly.

For example, rain and soil moisture data from a particular geographic region might be data mined to come up with some trend analysis using data mining, and so the same data is consumed and output, just in a different format as a result. In machine learning, this data pattern and forecasting might be used for running machines based on the weather to run irrigation systems in the most optimized way. So, this data of rain and soil moisture is still being consumed but to optimize the capabilities of a machine running irrigation in fields and influencing a different set of data to get optimized, say, ROI on running such a machine in the most optimized way.

The root of machine learning is in subjects like pattern analysis and various computational learning theories of artificial intelligence.

Tom Mitchell defined it as follows: "A machine learns with respect to a particular task T, performance metric P, and type of experience E, if the system reliably improves its performance P at task T, following experience E" (July 2006 CMU-ML-06-108) [http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf].

Machine learning is a field of computer science which takes care of how computers could improvise their programs based on tasks and outcomes and in a way program themselves toward a better output.

Machine learning–based artificial intelligence was a late entrant onto the market, and almost no commercial applications were on the market until the late 1980s.

Some prominent applications available using machine learning now are speech recognition, face recognition, robotics applications, robot chat bots, self-driving cars, and so on.

The machine learning training of software candidates is an area where programming in advance could be difficult or even impossible. You cannot program a machine to learn all expressions of all possible types of faces or patterns of voice. Even if that were possible, it would be cumbersome and time-consuming, and it would come preloaded with a huge volume of program logic data, maybe in terabytes. A better way is to pretrain it to train itself based on certain vectors and then leave it naïve as a baby and allow it to learn for the user(s) using it. This is mostly how these systems work. For example:

- IBM Watson for medical applications can identify details from CT scan results for a patient and provide suggestions using comparisons with all such patients with similar CT scan results and other parameters. This data could be fed in as an example of multiple CT scan reports, and various label markers could be added, but it is definitely impossible to create such data for Watson via programming alone.
- A facial recognition or fingerprint recognition system could be trained to allow only authorized images through the system based on such recognitions.

So, there is a category of data which could be sent only via data labeling and feeding into the system and not by programming, mostly in sensory perception situations like those involving image, sound, or other sensor data. Data labeling means identifying individual data with various vector points to mark it as something unique and discrete and then recording it as learning for the system.

The special characteristics of an individual's voice (in the case of a speech-recognizing system) are extracted from training with the user. These are then repeatedly retrained with the same user, and all information is stored as possible clusters with a designated centroid vector position. This is the training data for the speaker. The extraction happens using digital signal processing (DSP)–related tools. All such user training data are then identified in the system with a unique user ID. The system may accept or reject a sample for testing based on certain thresholds like loudness, clarity, background sounds, and distortions. Finally, the system with all training data is ready with a reference model or template for that user identified via a user ID (Figure 4-2). The more it trains, the more precision it gains. That is the reason Google Assistant on your phone or Microsoft Word requests you to train the system with your voice at least two to three times. So, when the user is being matched on voice invocation, it is nothing but pattern or feature matching happening at the back end with the individual user's template as recorded in the system and identified by a unique user ID.

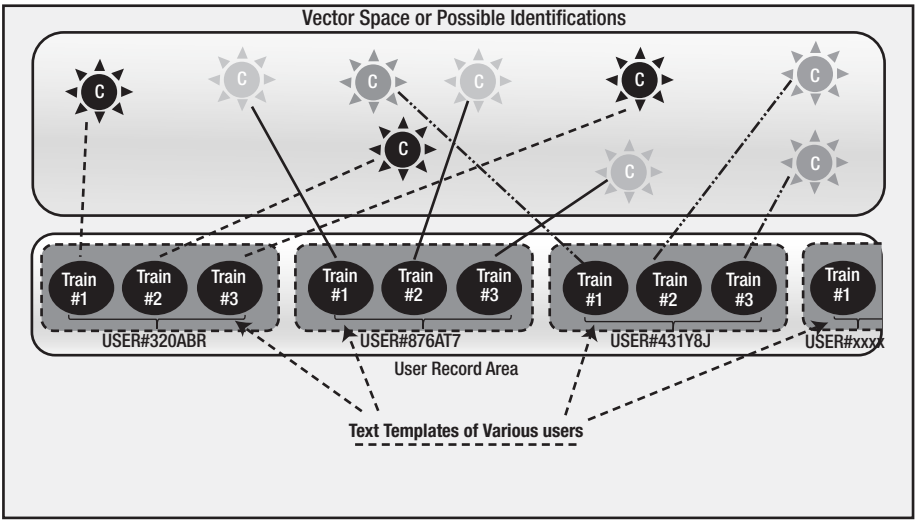

Figure 4-2. *Voice training models*

But there is a limitation: the learning is done according to the provided to that system. It cannot pick up new subjects to create a program related to a new subject area by itself. For example, a person proficient in singing might observe some street dance form over time and learn that as well. It could be willful learning, or he might observe it more passively and still learn it to an extent.

But a machine learning machine expert in cancer cannot just pick up music knowledge and become an expert into it; likewise, a voice recognition system learns voice wavefronts and not image pixels for facial recognition. Since this limitation remains, infinitely learning machines have not yet been created. What we have now is curriculum-driven machines which learn within their subject areas.

Conversely, self-initiated and –supervised learning would involve something like, say, a robot with the job of bringing tea and coffee to those working at a Recording studios also picking up some music lessons from people around by noticing and hearing them sing.

All these concepts are common in human and machine learning concepts. In fact, the latter tries to pick up everything from the former.

Some research says the human brain has one single algorithm to learn; if brain nerve connections from the eyes and ears are cut and swapped with each other at their primary connecting locations in the brain, the swapped locations learn the new perceptions. So, the hearing part of the brain when connected with the optical nerve starts to process the visuals and the vision-related brain now when getting connected with auditory nerves starts to process sounds. This is the basis for neural networks, which are very much a part of machine learning.

Machine Learning Tasks

Machine learning is a type of artificial intelligence. So, the types of learning tasks are as applicable for artificial intelligence and as mentioned in detail in Chapter 3 (supervised, unsupervised, and reinforcement learning)

Based on output, certain additional tasks could be listed. They are as follows:

Classification

This is an operation to predict into classes based on observations made. This is achieved using supervised learning only. So, the classification categories are already in place and well defined. The data so classified could be of any type, be it in normal text format or even images. See the previous example in this chapter on voice template creation for users and the image is Figure 4-2.

The training of the model is done to further classify any data into correct categories. The context of every word is calculated as a probability by the classifying program, and with this probabilistic weightage the whole connected content is moved under or labeled under one such predefined category of classification. The most common applications in this category are spam-detecting applications. Additionally, these applications could also be further trained by the user to identify more types of spam. The user manually designates some mails as spam, and the system further trains itself to use this criterion as well. Google search and Google news also fall in this category.

Clustering

Clustering is an unsupervised task. The inputs are divided into groups, but no predefined classification exists. These groups are called clusters. The clustering could take any of the multiple top-down or bottom-up approaches that are available. The inputs are marked with some values. The clusters are then defined based on all such objects present near or closest to this mean (K-means clustering). Every cluster defined could be related further into a tree, or else all objects may first be considered as a big blob slowly divided into clusters and related with the other clusters as a tree (hierarchical clustering) (Figure 4-3). Online e-commerce shopping applications like Amazon work on clustering principles.

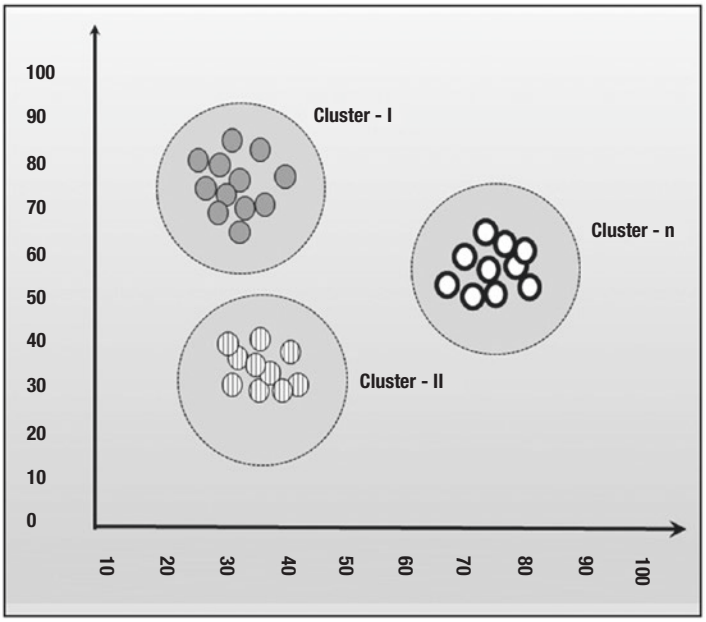

Figure 4-3. *Clustering example*

The online stores have millions of products in connected inventories for sale on their web sites. There could be two major issues here.

- The products for sale on these web sites could be so numerous at a given point of time that quickly grouping them into correct product groups might become difficult.
- If a user is interested in a specific subtype of product, the method by which the group is determined should be flexible or intelligent enough to help that customer make the purchase.

This is where clustering helps. While uploading details by the seller, there are certain questions asked to have a quick tagging. For example, a supplier might upload a shirt which has details like men's apparel; formal; sizes M, L, and XL; material cotton; and colors red, white, purple, and yellow. Based on these properties (tags) and clustering algorithms, the system automatically registers it under the men's apparel category.

Now if a customer is interested in buying blue formal shirts, the system should be again able to do a clustering of products based on the specific search terms, earlier buying patterns, or clone customer properties (any other customer with a similar pattern, e.g., geographic location, search term language, age, etc.).

There may not be a "blue, men's apparel" category as such. But based on the tagging and smart recommendation and prediction clustering algorithms at the back end, the e-commerce site starts to show relevant products on the screen, and this should increase user satisfaction and sales.

Regression

This is a supervised task. The relationships between mean values of multiple outputs are compared, and corresponding values for other connected outputs are predicted linearly. Training data is used to predict (regression) using statistical analysis. A famous example is predicting house prices from related training data. It does not group objects as clusters around mean values as seen in clustering or classify them under different labels as seen in classification; instead, it predicts values for individual candidates (Figure 4-4).

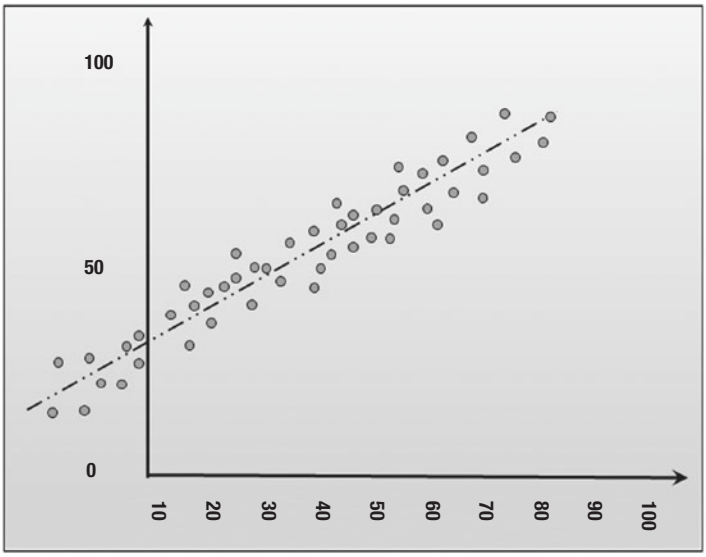

Figure 4-4. *Regression example*

Connected Key Concepts

Machine learning has many key concepts associated with it. These could be considered as approaches to machine learning and are applied to achieve machine learning solutions. Some of them are described in the following sections.

Deep Learning

This is also known as deep machine learning (DML). Artificial neural networks (discussed in Chapter 3) with multiple hidden layers, when studied together with algorithms, are called deep learning or DL.

A quick differentiator between machine learning and deep learning could be an example of image recognition. It could be both, depending upon what is required. Imagine that there is a factory-based robot whose purpose is to learn and stop defective parts from going to the packaging unit. It has some sort of image variations and original expected images fed into its system. This robot could use a full mug shot of every article

finished and running on the factory line and quickly do a comparison to see whether it's defective or not. These machines could do this with extreme speed and accuracy. This is classic machine learning.

Now let us look at another example: in a crowd, moving trucks, buses, cars, cycles, cats, dogs, and people are all there together. The requirement is to only find people's faces and compare them with the mug shots in a database of criminals to see if someone is fleeing the law and then to arrest that person immediately. This has other challenges apart from those the previous example had.

In this case, there is no specific image or images that could be captured with calibrated focus adjustments and arrangements; it could be a lot of things captured in the surveillance cameras apart from the face. It could be a face from any angle; it could be a man with a big cat in his arms or with a cart full of melons. Before comparing the face and the mug shot, the machine at the back end should be able to differentiate from continuous exposures, the collection of faces, and then process the comparison exercise. This is where machine learning fails. It needs data input from a wider range and processing of a much higher order. This is where deep learning comes into play.

DL is more in-depth learning, as the data learning is transferred through a multilayer artificial neuron–based neural network, and thus deeper meaning, with many times more variables and components, can be derived from the data. Thus, it provides deeper understanding and in turn learning. The percolation through this multilayer complex neural network can turn shallow or not-so-deep input into deep and rich learning.

This could be used for either supervised or unsupervised learning, but the best usage is for unsupervised learning, as this deep neural network–based percolation provides richer inferences and thus richer learning for machines. Therefore, the usability factor fits more with unsupervised learning. Also, major problem areas always lie waiting in the unsupervised area.

In simple languages, multiple dimensions of possible details on any subject are fed into the system, and then the system can get the needed perception even if only a partial dimension of information is available. This is the technique behind face recognition systems used around the world (one recently in the news is that being used in the Gold Coast by Australian federal police in preparation for the Gold Coast Commonwealth Games 2017). An array of CCTV cameras will be using such algorithms to capture facial images of people in the crowd; in turn, their facial features will be mapped, measured, and compared with terrorist image databases.

So, the deep learning machines are a step ahead of machine learning machines. The machine learning applications start to fail when the dimensionality of the information becomes too high. In other words, the more complex the problem, the more it becomes dimensional. Deep learning, which uses neural networks to help it break the problem's dimensions into various abstraction layers and then to solve them one by one, combining the result at the highest abstraction layer, is used for all of these applications.

Facial recognition from partial face images and being able to detect a known terrorist in a crowd is a problem too complex and multidimensional to be solved by machine learning. It has so many dimensions: the edges to understand that this part captured in the image is a nose, partial lips, and some part of eye or some other combination of facial features captured. There are dimensions of skin color, skin texture, eye color, proportion of the eye or nose to face, and so on, along with everything related to movement of muscle stretching it while laughing or even full-face details of people to be compared with. The neural networks help here.

The neural networks follow the Markovian hidden model concept and have an input interface, an output interface, and in between multiple hidden layers. Every layer processes a different dimension; then, all of them are combined using mathematical models to come up with an output result at the highest possible abstraction level, and the face is recognized.

See Figure 4-5 to see the same example there.

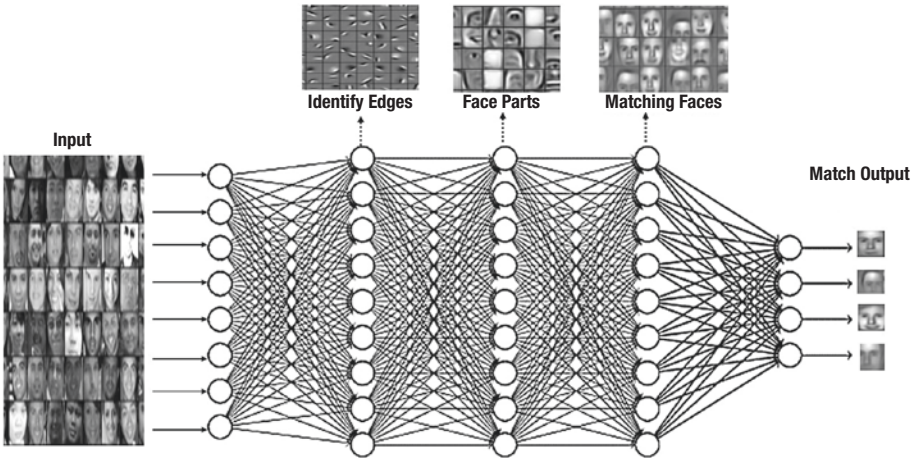

Figure 4-5. *Example of deep learning*

The figure shows a neural network with multiple hidden layers in between. The input and output layers are at both ends. The feed of images from cameras is entering from input, and then every layer of the neural network is processing the various properties and goes to higher abstraction layers with increasing closeness in terms of output or the reach toward the resolution. The first layer looks at the various edges and contours of images and then tries to resolve them as understanding of face parts in the next layer, which is a higher layer of abstraction. The next layer then matches these identified attributes with the database images and could differentiate faces from background in the original images and compare them easily with the database images, identifying, say, four people from the database as matches in the crowd.

This is how deep learning works, as shown in one very famous and simple example.

The deep network combines the various simple networks doing multiple simple tasks together with increasing levels of complexity of result, and then formulates the result. This could be very thick in terms of the number of layers in real-life applications, considering the dimensions of information that need to be processed.

Genetic Algorithms

These are algorithms that mimic the biological genetics theory of natural selection. Where biological theories talk about various crossovers and mutations of genes for a sustainable species, these algorithms use heuristic concepts to find better solutions. The genetic algorithms have genetic factors for calculations and success parameters to judge the success of each possible variation hypothesized and simulated. Machine learning algorithms use these to optimize their efficiency, and genetic algorithms are refined using machine learning.

Decision Tree and Association Rule

Machine learning uses data mining like computation to get the most structured learning to improve its program. So, decision trees are exactly like data mining decision trees. These trees are helpful in predicting solutions better and structuring data for learning even in unsupervised learning.

By comparison, the association rule helps machine learning machines to create combined learning from deducing associations in data and further bringing combined learning out of the scenarios to add more complex and effective changes in programs.

Bayesian Network

There are two major problem focus areas in engineering, and managing both together is a challenge always: complexity and probability or uncertainty. This could be modeled using graphical modeling and random variables in Bayesian or belief networks. This graphical model is a combination of smaller graphical components. Each smaller component is a representation of a smaller subset of the whole problem. Graphical models have always been considered the most intuitive way of representation by humans. This makes it a more interesting subject for artificial intelligence and machine learning areas. Bayesian networks could be simple or even dynamic.

Speech Recognition

These applications are unique in a way; they're shipped and then trained afterwards by users. Speech recognition isn't something which can be preprogrammed to recognize all speech patterns; Instead, it must be trained for the use of specific users. Even aside from regional accent differences, all people produce slightly different sounds when pronouncing the same word. With more than 6,900 living languages across the world and even with thousands of dialects in just one country or region, it is impossible to have the software to be trained in advance in all these. It is again the physical configuration of the person that influences how he or she speaks that language in that dialect, and this of course makes it much more complicated.

Even this would have been impossible to do: just consider the amount of training data needed to put into this. For instance, imagine a smartphone coming with 50GB or even 1TB of data preloaded just to have a training content for such a voice recognition system.

All this combined makes the speech recognition task a perfect candidate for machine learning, where a system based on preset training and rules is shipped and then does its training with the specific user to interpret his voice.

Biosurveillance

Many countries globally now use machine learning for biosurveillance of hospitalization data to understand admission patterns for many infectious diseases. On the basis of already understood patterns, the system detects any anomalies. This helps in the setup of biosurveillance to help stop the spread of such diseases.

Machine Learning vs. Statistics

The similarities and differences between machine learning and statistics have long been a topic for debate. Some people refer to them collectively as the statistics of computer science, while others prefer to consider them as two distinct areas based on origin: statistics coming from mathematics and machine learning from computer science (artificial intelligence).

Both machine learning and statistics work on data to derive patterns and churn out some meaning.

One major difference is that while statistics looks for data with a predefined pinpoint to discern any number of tendencies, machine learning must jump into a complete new data scenario and deduce meaning out of it. But if we consider that machine learning is an application for one just subject specifically, then this is equivalent to predefining data pinpoints as seen in statistics.

To explain this: say the machine is primarily designed to respond to problem A. Over time, the machine develops some new learnings about the environment and starts to develop meaning out of it. More specifically, let's say that the machine was required to bring a glass of water or cake or an apple from the kitchen to some user's desk in a specific room, and the user says "bring water [or cake, or an apple, etc.] to my desk." Now, the user has moved to other side of the desk and calls by saying "bring water to the other side of the desk." This will require adjustments in the path. Subsequently, when the user calls for cake or an apple by using the phrase "other side of the table," the machine will know that it must go to the other side and thus will calculate its path of travel to take it there. The machine has learned. But consider the scenario in which a natural and normal statement is made. This statement could be made in any number of tones, styles, and other variations by the same user or a different one, and yet all would mean the same thing: bring water to my table. This works at the back end exactly the same as statistical analysis, and the mean value of weightage of statements for the same scenario is being used. The learning curve brings in nonstatistical behavior by shifting the central mean to another locus, where again its multiple variants with same meaning at the other side of table would be there. So, for the same situation and to understand that routine input, it works in a statistical pattern, while a new one is to be generated for new learning and it must deviate from that basic behavior to be able to generate new training data which could be stored further.

There is also interrelation between these two. Statistics has given means to machine learning for it to write algorithms, while statistics has also gained from machine learning usage.

One very nice representation of the relationship between statistics and machine learning can be seen in a publication from SAS (http://blogs.sas.com/content/subconsciousmusings/2014/08/22/looking-backwards-looking-forwards-sas-data-mining-and-machine-learning/).

Statistics is only concerned with data points or results coming out of the data set under consideration, while machine learning has a purpose to generate further meaning out of it and use it to upgrade its program and in turn the complete system.

So, it is very difficult to come up with a conclusion no matter whether both are the same, one is a subset of the other, or they are complimentary to each other.

In my opinion, statistics, a well-established subject for hundreds of years, is more into pure mathematical modeling or deduction of data for various sets of data. It is a tool to play with data. It is said to be a branch of mathematics dedicated to collection, analysis, interpretation, presentation, and organization of data.

On the other hand, machine learning a newer field of computer science, or rather a sub-branch of artificial intelligence, and uses most of the techniques of statistics to solve its purpose of inferring meaning out of data given to it through various means. So, the end stage for a machine learning application is to get data organized through statistics and then also to make some conclusions in the form of learning. This is then used to create changes in programs or to create new ones based on rules to do so.

However, statistical methods could also do lot of analysis using various machine learning applications to generate relations or test such relationships.

In other words, they are complimentary to each other. But machine learning is based on (some part of) statistics, but the reverse cannot be true.

Also, machine learning is not only statistics. It is lot of other subjects also combined to come up with such a solution.

Business Use Case Example

Netflix is one good example of this. The Netflix Prize is one of the famous competitions in this area; its purpose is to understand how user data could be utilized to train the system in predicting user choice in advance and propose the most probable options to them.

Even part from the publicity from the competition itself, it was a very smart business idea to have a competition or challenge brought onto the market using sample data. By paying $1 million to the winning team, they had paid only to for the winning idea and saved themselves many hours by having multiple teams research all the options and select the most relevant and suitable. So, it became a very low-cost affair relative to going only to a selected team in house for the same purpose. Also, the restrictions of knowledge reach-out were overcome due to this. Anyone who is eligible and could think of something could compete, removing all limitations to work only with people selected via recruitment limitations.

However, there was one problem: there was a lawsuit filed against them for disclosing private/confidential customer data to third parties without any prior consent.

But to return to our business use case topic, let's look at how Netflix runs on an extremely advanced recommendation system. The bases for this system are as follows:

- People are unique and every single person using the same subscription in a household is unique.
- People respond positively by liking a solution or service when it provides them with favorable solutions.

Netflix wanted this kind of machine learning solution for its viewers. Starting off as a store-based DVD movie supplier, they had moved to being a completely web-based business.

The biggest shift in this from a client interaction perspective was the missing store-level personal touch. Even though the library was so immense by now, a store-based service might still have advantages in helping individual customers select the right stuff, since it was still possible to interact with or even know a customer personally to recommend something relevant for them.

But the online store environment at times works as a black box from this perspective. You come to know who liked what if the customer provides feedback and again if the customer hits the right content by searching and browsing through the sea of content and multiple genres.

This was the problem solved by Netflix: to deliver content even better than they might have done in store with the human touch.

The system was designed from the get-go so that it would recommend the right stuff to people based on their prior interactions and on rule-based recommendation algorithms.

So, a smart system would need to make recommendations based on the actual person from the household who is using that subscription; that is why Netflix recommends adding individual subusers from the household to the subscription account.

Not only the videos but even the trailers for movies are handled that way. For instance, one person might be interested in watching war movies, and another in the same family would rather watch comedy more; in such cases, trailer versions shown to each might differ by emphasizing the more interesting factor for him or her.

So, this takes care of everything from TV serials to movies to trailers that are to be recommended and presented to individual users as per their choice.

This is all within the scope of machine learning. The system algorithm was first designed in the Netflix Prize. At one time, the algorithm was designed for a smaller set of data; about 100 million user ratings. It was then recreated to accommodate billions of records. Once the system was ready with this algorithm and trained with this data, a baseline intelligent system was ready. This was based on understanding these customers and getting inferences from this learning to make recommendations for other customers, even new ones.

Over time, this system learned from both perspectives. One perspective is of the overall preferences of the various customers, and this is to help make recommendations even for new customers when it has not started training itself about their preferences. The other perspective is more of a continual and complete learning process to identify the preferences of existing customers to be able to recommend relevant content to them.

This is how machine learning works: getting trained via data to be able to create outputs more and more preferable to that event and in this case to individual customers or members using this subscription at home.

■ **Note** As we saw in Chapter 1, Netflix went a step ahead and started producing movies based on such choice patterns. Based on customer base matrix and their choices, Netflix could understand in advance or predict what kind of story and which artist combination might give a superhit program cocktail. The result was a show like *House of Cards*.

The genres for content were also divided into subgenres. This was required to be able to make recommendations with the benefit of additional parameters. For example, a single genre like comedy was divided into subgenres like

- Dark comedies (e.g., *Look Who's Back*)
- Mockumentaries (e.g., *People Just Do Nothing*)
- Political comedies (e.g., *War Machine*)
- Sports comedies (e.g., *Chalk It Up*)
- Satires (e.g., *Tropic Thunder*)

There are about 12 such categories in comedy itself. So, Netflix broke every genre into subgenres to fit every subtype of viewer identified and recommend relevant stuff to him or her.

The recommendation system is such that it takes care of recommendations based on preference and popularity. Popularity, though, is in some senses just the opposite of preference, since it refers to something being watched because one likes it only, rather than because others are watching. But this important factor too is counted while training the system for that individual from a group or category, as ascertained by machine learning algorithms.

It also takes care of users' social data to see what is popular among the users' own networks.

So, Netflix is a well-running and successful example, and a model like theirs is my personal choice, since its lets organizations not only to successfully implement it and reap its benefits but also to bring in a new era in the entertainment industry: a shift from the physical CD/DVD renting business to online content streaming.