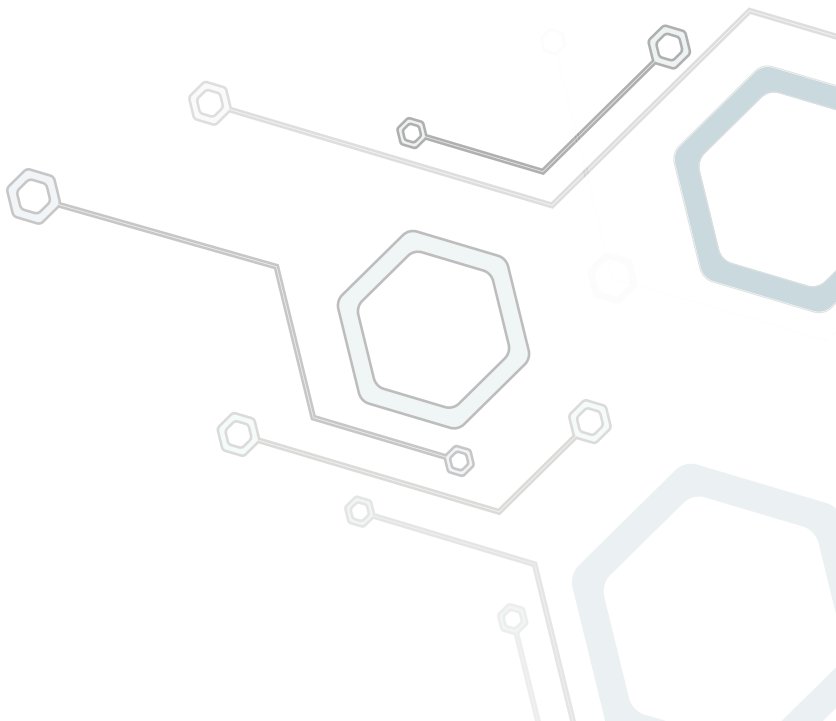
Demystifying Machine Learning

Enter

ORACLE

Table of Contents



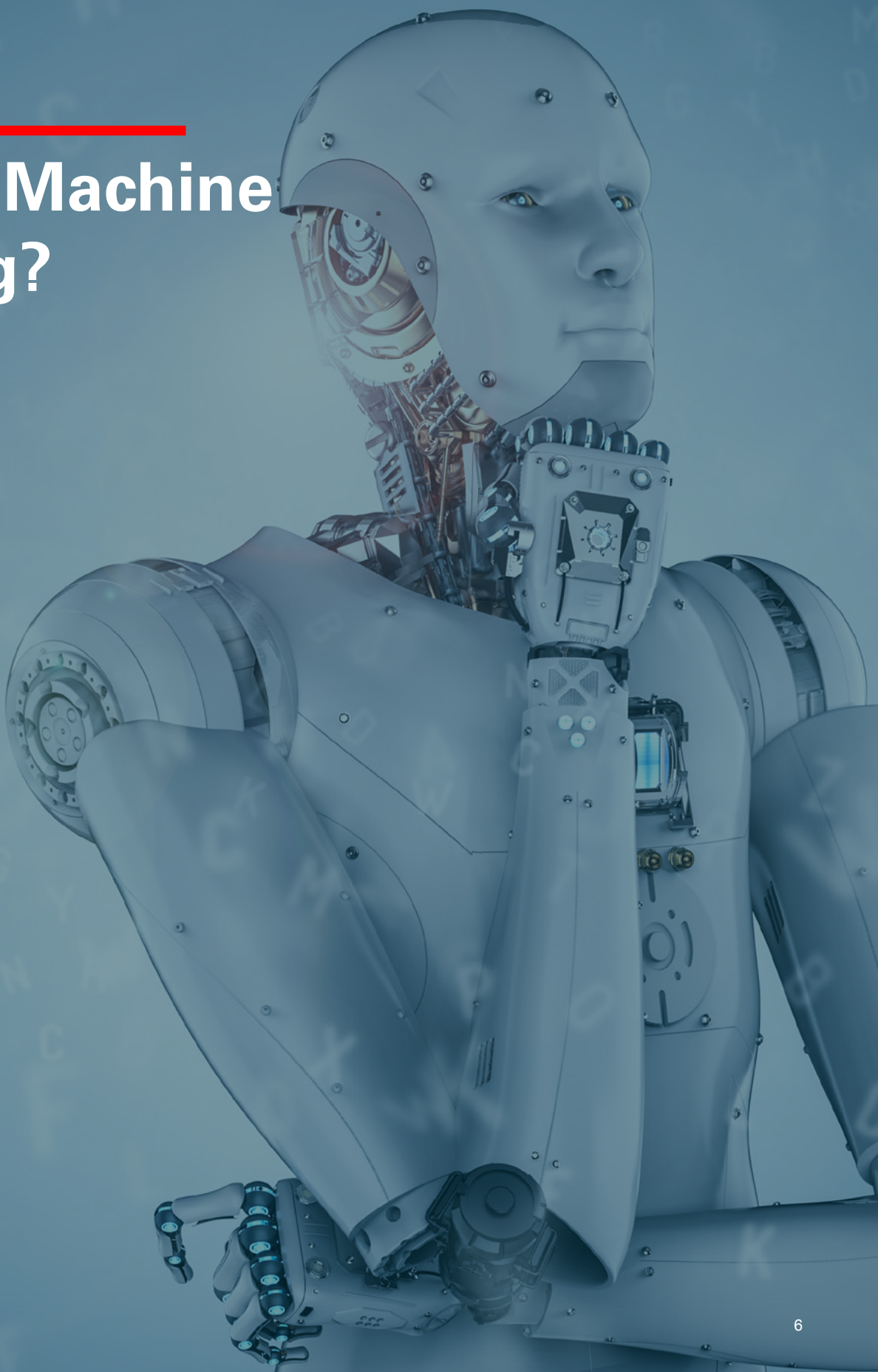
Introduction

Have you ever had a credit card transaction declined when it shouldn't have? Or been on the receiving end of a personalized email or web ad? Have you ever noticed that when you're shopping online, the site often gives you recommendations for things you might be interested in? And my last example, have you ever had an offer from a company designed to stop you from leaving them as a customer?

If any of these things have happened to you, then you've probably been on the receiving end of a machine learning algorithm, employed by a company or organization you do business with (or in some cases, have merely considered doing business with).

We're going to take you behind the scenes and give you a layman's view of machine learning so you can see what kind of problems they can solve. If you're a data scientist, then you might be more interested in this "big data journey" about accelerating data science, which is more detailed. But this series is designed for less technical people who hear the buzzword, who know that it's something important, but don't really know what it is or what it can do. You'll get just enough information to make you dangerous.

What Is Machine Learning?



A McKinsey article describes machine learning as "[...]based on algorithms that can learn from data without relying on rules-based programming." Put another way, with big data you have got a lot of data. Determining what to do to with it and figuring out what it's telling you isn't easy. So you can understand the appeal of machine learning, which basically allows you to find processing power and the right algorithm and tell them to figure things out for you.

The analogy is to how we learn as human beings, experiencing the world around us and working things out for ourselves. When I taught my kids how to ride a bike, I didn't give them "The Rules of Bike Riding." I put them on a bike, held onto it/them, and let them work it out. They took data inputs from their eyes, their ears and, on one occasion a large bush, and started to discover what would keep the bike upright.

So it is with machine learning. Take the data, work with it, and see what comes out.

A Simple Example

Supposed you've been tasked with finding out more about your customer base. In that snapshot below you can see they're a diverse and pretty happy group. But what else do you know?

Well, a simple query in your database might reveal things like age, gender, or how they like to be contacted (mail, email, phone, text).

You could run a query with some analytics to calculate, say, RFM, a measure of customer value based on how much customers spend and how often they spend. You can see who is more valuable to you, but you wouldn't really know what to do with that data.

Machine learning algorithms could do much more. For example, you could group your customers into segments that show similar behavior, or you could also figure out how likely they are to purchase a given new product of yours. In that picture below we have customers in five different behavioral segments: "retired cosmopolitan", "affluent executive", "new home mom", "young, successful startup" and "executive product collector". And we know that their likelihood of purchasing that product we're looking at promoting ranges from 21 percent to 72 percent.

Now you have something potentially more powerful. Armed with this information, you can:

- Tailor your marketing campaigns
- Use different language for those different groups
- Prioritize campaigns
- Market that product only to the subset of customers likely to buy it

In this situation machine learning gives you much more insight into your customers and, perhaps most importantly, it can predict what they might do or respond to. These days, we're starting to take deep learning to another level and use it to solve real business problems—which is very new and exciting.

But how does machine learning actually do that? Let's walk through some common techniques so you have a good foundation for understanding what's going on in that much-hyped machine learning world.

If you are a data scientist, remember that this series is for the non-expert.

But first, let's talk about terminology. I'll use three different terms which I've seen used interchangeably (and sometimes not accurately): techniques, algorithms and models. Let me explain each one.

A **technique** is a way of solving a problem. For example, classification (which we'll see later on) is a technique for grouping things that are similar. To actually do classification on some data, a data scientist would have to employ a specific **algorithm** like Decision Trees (though there are many other classification algorithms to choose from). Finally, having applied an algorithm to some data, the end result would be a trained **model** which you can use on new data or situations with some expectation of accuracy. It should all be clearer after these examples, so read on.

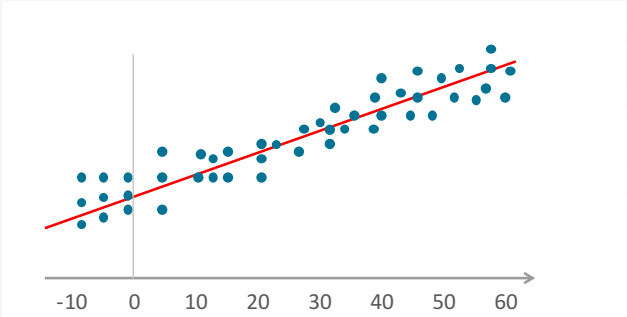
Machine Learning Technique #1: Regression

If you're looking for a great conversation starter at the next party you go to, you could always start with "you know, machine learning is not so new; why, the concept of regression was first described by Francis Galton, Charles Darwin's half cousin, all the way back in 1875." Of course, it will probably be the last party you get an invite to for a while.

But the concept is simple enough. Galton was looking at the sizes of sweet peas over many generations. We know that if you selectively breed peas for size, you can get larger ones. But if you let nature take its course, you see a variety of sizes. Eventually, even bigger peas will produce smaller offspring and "regress to the mean." Basically, there's a typical size for a pea and although things vary, they don't "stay varied" (as long as you don't selectively breed).

The same principle applies to monkeys picking stocks. On more than one occasion there have been stock-picking competitions (WSJ has done them, for example) where a monkey will beat the pros. Great headline. But what happens next year or the year after that? Chances are that monkey, which is just an entertaining way of illustrating "random," will not do so well. Put another way, its performance will eventually regress to the mean.

What this means is that in this simple situation, you can predict what the next result will be (with some kind of error). The next generation of pea will be the average size, with some variability or uncertainty (accommodating smaller and larger peas). Of course, in the real world things are a little more complicated than that.



In the image above, we don't have a single mean value like pea size. We have a straight line with a slope and two values to work with, not just one. Instead of variability around a single value, here we have variability in a two-dimensional plane based on an underlying line.

You can see all the various data points in blue, and that red line is the line that best fits all that data. And based on that red line, you could make a prediction about what would happen if, say, the next data point was a 70 on the X axis. (That prediction would not be a single definitive value, but rather a projected value with some degree of uncertainty, just like for the pea sizes we looked at earlier).

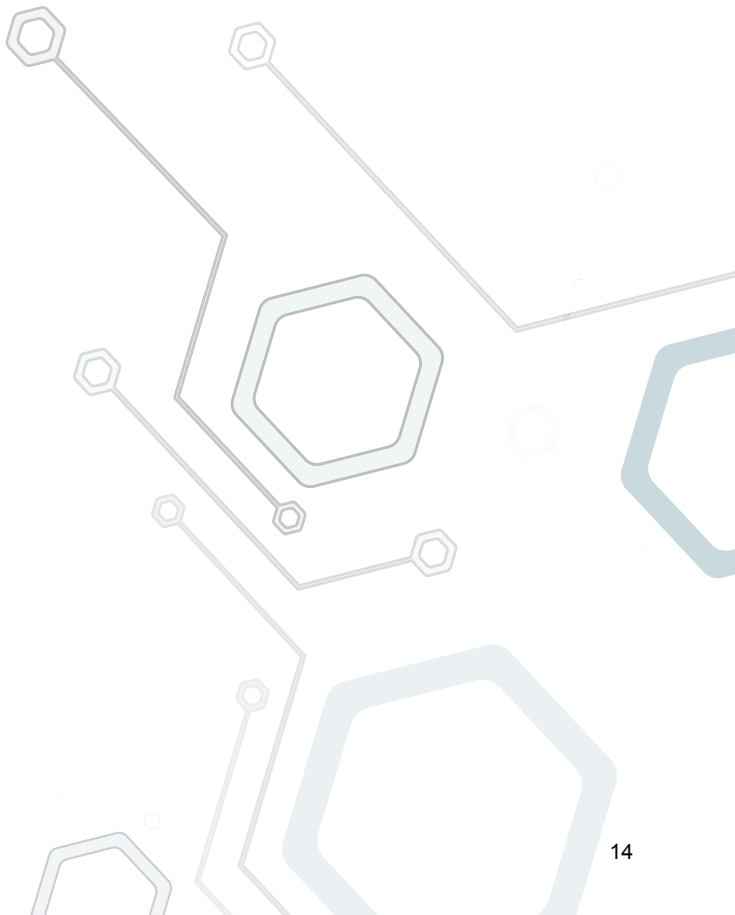
Regression algorithms are used to make predictions about numbers. For example, with more data, we can:

1. Make predictions about customer lifetime value, perhaps spotting potentially valuable customers before they have declared themselves by the volume of their purchases
2. Predict the optimal pricing for a product to maximize revenue or profit
3. Predict house prices, for companies that want to send out those property newsletters

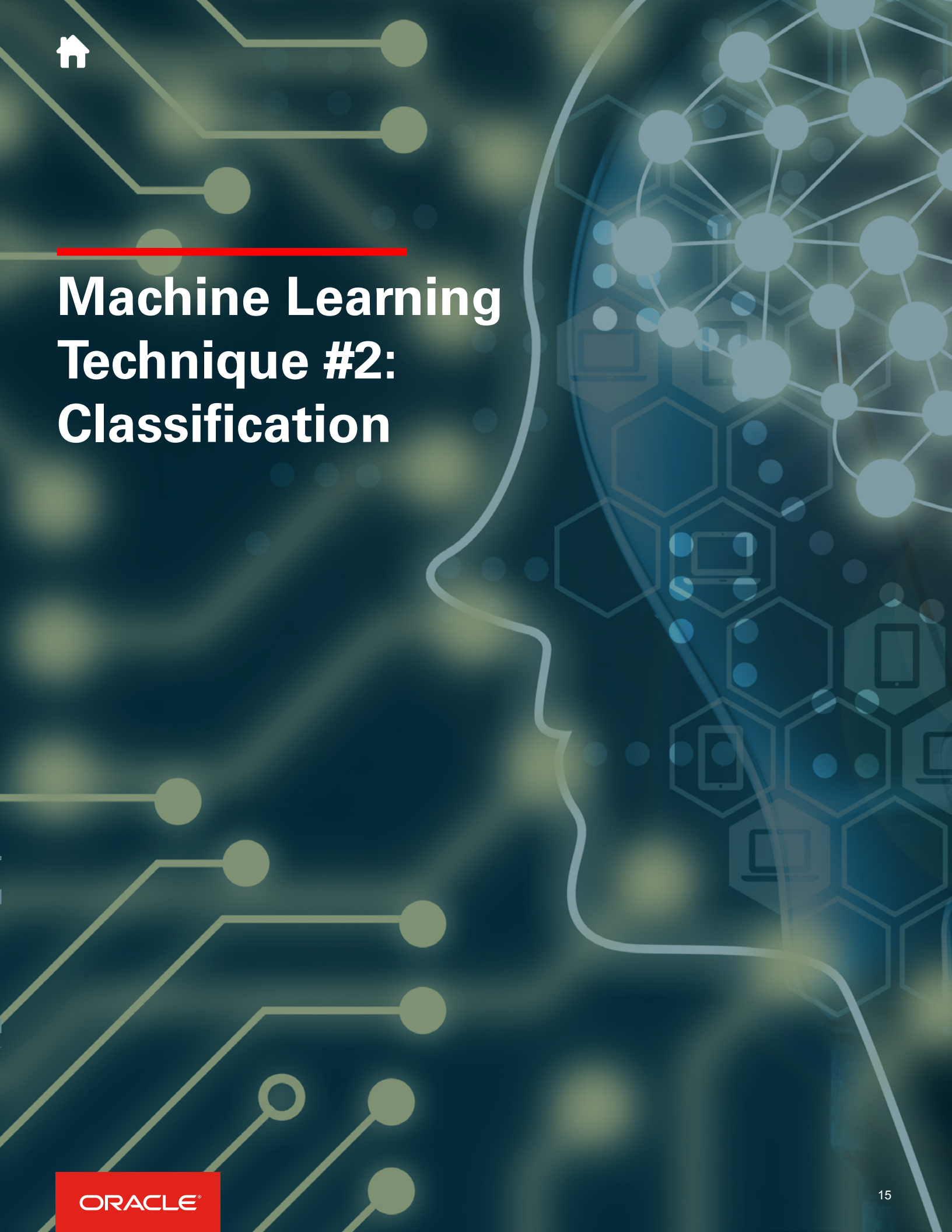
The straight line in the graph is an example of linear regression, but looking at those three examples above, I'd be surprised if any of them fit well to a straight line. And in fact, the underlying line behind your data doesn't have to be straight. It could be an exponential, a sine wave or some arbitrary curve. And there are algorithms and techniques to find the best fit to the underlying data no matter what shape the underlying line is.

Furthermore, I've given you a two-dimensional diagram there. If you were trying to predict house prices, for example, you'd include many more factors than just two: size, number of rooms, school scores, recent sales, size of garden, age of house and more.

Finally, perhaps my favorite example of regression is this approach to measuring the quality of Bordeaux wine.



Machine Learning Technique #2: Classification



Let's move on to classification. And now I want you to pretend you're back in preschool and I'll play the role of teacher trying hard to teach a room of children about fruit (presumably fruit-hating children if they have to this age without knowing what a banana is).

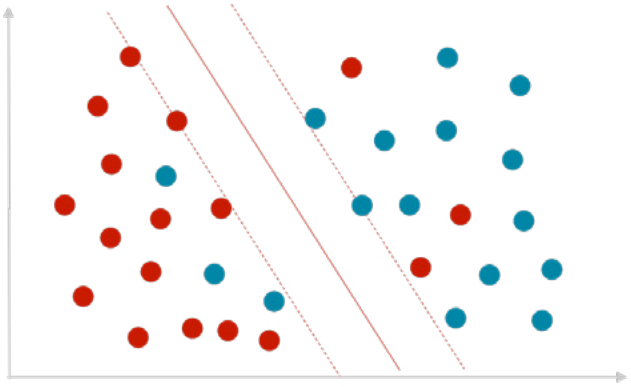
While you kids don't know about fruit, the good news for you is that I do. You don't have to guess (at least initially). I'm going to show you many pieces of fruit and tell you what each one is. And so, like children in a preschool, you will learn how to classify fruit. You'll look at things like size, color, taste, firmness, smell, shape and whatever else strikes your fancy as you attempt to figure out what it is that makes an apple, an apple, as opposed to a banana.

Once I've gone through 70 percent to 80 percent of the basket, we can move onto the next stage. I'll show you a fruit (that I have already identified) and ask you, "What is it?" Based on the learning you've done, you should be able to classify that new fruit correctly.

In fact, by testing you on fruit that I've already classified correctly, I can see how well you've learned. If you do a good job, then you're ready for some real work which in a non-kindergarten situation, would mean deploying that trained model into production. If of course the results of the test weren't good enough that would mean the model wasn't ready. Perhaps we need to start again with more data, better data, or a different algorithm.

We call this approach "supervised learning" because we're testing your ability to get the right answers, and we have a lot of correct examples to work with since we have a whole basket that has been correctly classified. That idea of using part of the basket for training and the rest for testing is also important. That's how techniques like this make sure that the training worked or, alternatively, that the training didn't work and a new approach is needed.

Note that the basket of fruit we worked with had only four kinds of fruit: apples, bananas, strawberries (you can't see them in the picture, but I assure you they are there) and oranges. So, what happens if you were presented with a cherry to classify is somewhat unpredictable. It would depend what the algorithm found to be important in differentiating the others. The point here of course is that if you want to recognize cherries then the model should be trained on them.



Here's an example of a chart showing a data set that has been grouped into two different classes. We have a few outliers in this diagram, a few colored points that are on the wrong side of the line. I show this to emphasize the point that these algorithms aren't magic and may not get everything right. It could also be the case that with different approaches or algorithms, we could do a better job classifying these data points and identifying them correctly.

Summarizing the previous entry, classification enables you to find membership in a known class. Examples of known classes? Let's go back to customer segmentation. I know who my high-value customers are today. What did they look like some time ago? By using them as a training class, I could train a model to spot a valuable customer earlier.

Another example is customer churn. We know who's left us. Let's train a model on that class and then see if we do a better job of spotting other churners before they churn. This kind of approach is what triggers those unexpected offers from companies who think you are about to leave them.

Insurance companies pay out on claims and they have a historical set of claims that they have already classified into "good claims" and ones that need "further investigation." Train a classification algorithm on all those old claims, and perhaps you can do a better job of spotting dubious claims when they come in.

One additional point.

In all these cases, it's important to have lots of data available to train on. The more data you have, the better the training (more accurate, wider range of situations etc.). One of the reasons (of course there are others) for building a data lake is to have easy access to more data for machine learning algorithms.

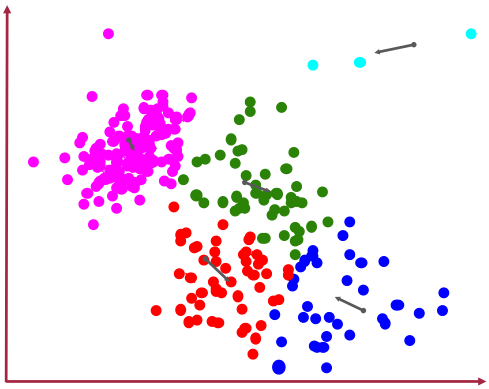
Machine Learning Technique #3: Clustering

Alert readers should have noticed that this is the same basket of fruit used in the classification example. Yes, this was done on purpose. Same fruit, but a different approach.

This time we're going to do clustering, which is an example of unsupervised learning. You're back in preschool and the same teacher is standing in front of you with the same basket of fruit. But this time, as I hand the stuff out, I'm not going to tell you "This is a banana." Instead I'm effectively going to say "do these things have any kind of natural grouping?." (Which is a complex concept for a preschooler, but work with me for a moment).

You'll look at them and their various characteristics, and you might end up with several piles of fruit that look like "squidgy red things, ""curved yellow things", "small green things," and "larger red or green things."

To clarify, what you did (in your role as preschoolers/machine learning algorithm) is to group the fruits in that way. What the teacher (or the human supervising the machine learning process) did was to come up with meaningful names for those different piles. This is likely the same process used to do the customer segmentation mentioned earlier with a shorthand way to name or describe each grouping.



This approach is very common for customer segmentation. You could evaluate credit risk, or even things like the similarity between written documents. Basically, if you look at a mass of data and don't know how to logically group it, then clustering is a good place to start.

Here's a real-world cluster diagram. With these data points you can see five separate clusters. Those little arrows represent part of the process of calculating the clusters and their boundaries: basically pick arbitrary centers, calculate which points belong in which cluster, and then move your arbitrary point to the actual center of the cluster and repeat until you have close enough (movements of the centers are sufficiently small).

Machine Learning Technique #4: Anomaly Detection

Sometimes you're not trying to group like things together. Maybe you don't much care about all the things that blend in with the flock. What you're looking for is something unusual, something different, something that stands out in some way. This approach is called anomaly detection. You can use this to find things that are different, even if you can't say upfront how they are different. It's fairly easy to spot the outliers here, but in the real world, those outliers might be harder to find.

One health provider used anomaly detection to look at claims for medical services and found a dentist billing at the extraordinarily high rate of 85 fillings per hour. That's 42 seconds per patient to get the numbing shot, drill the bad stuff out and put the filling in. Clearly that's suspicious and needs further investigation. Just by looking at masses of data (and there were millions of records) it would not have been obvious that you were looking for something like that.

Of course, it might also throw up that fact that one doctor only ever billed on Thursdays. Anomalous, yes. Relevant, probably not. Anomaly detection can throw up the outliers for you to evaluate to see if they need further investigation.

Finding a dentist billing for too much work is a relatively simple anomaly. If you knew to look at billing rates (which will not always be the case), you could find this kind of issue using other techniques. But anomaly detection could also apply to more complex scenarios. Perhaps you are responsible for some mechanical equipment where things like pressure, flow rate and temperature are normally in sync with each other: one goes up, they all go up; one goes down, they all go down. Anomaly detection could identify the situation where two of those variables go up and the other one goes down. That would be really hard to spot with any other technique.

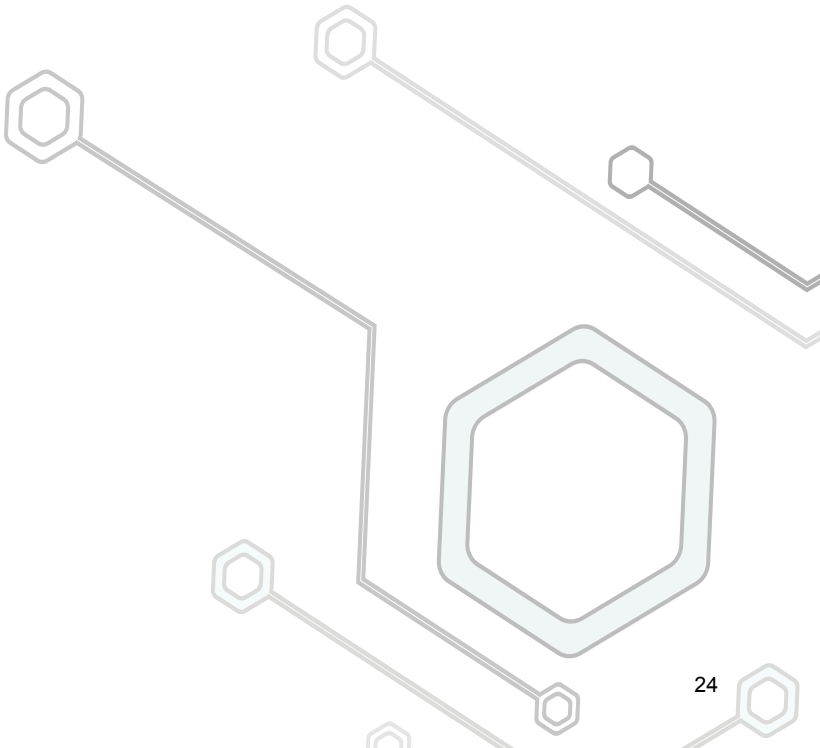
Machine Learning Technique #5: Market Basket Analysis

Association rules are typically used for "market basket analysis." Basically, you can look at things people buy and use it to predict other things that they might buy. How? Because by examining enough of these things, we can determine rules about which purchases are associated with other ones. For example, if somebody buys peanut butter then there is an X percent chance of buying jelly. A more complex rule (a rule of length four meaning four items are involved) would be "if somebody buys spaghetti, tomato sauce and ground beef, then there is an X percent chance of buying parmesan cheese." These rules drive things like "you might be interested in..." when you are shopping on a website.

It turns out that you can use these rules in other ways, though. Perhaps you are responsible for operating some mechanical equipment.

You could use association rules to track and uncover the relationships between, say, high temperature, pressure and vibration with a particular failure mode. So these kind of rules can help with root cause analysis.

Did you ever enjoy Gerber Singles? Or Colgate's packaged meals? What about Pepsi AM? All of those products and many others failed. But it turns out that some people liked them. And further research from MIT shows that some people consistently like products that turn out to fail. They called them "harbingers of failure." If you have this kind of business, perhaps you'd like to find this kind of consumer. Great way to get new insight about the likely success or failure of a new product you're launching.



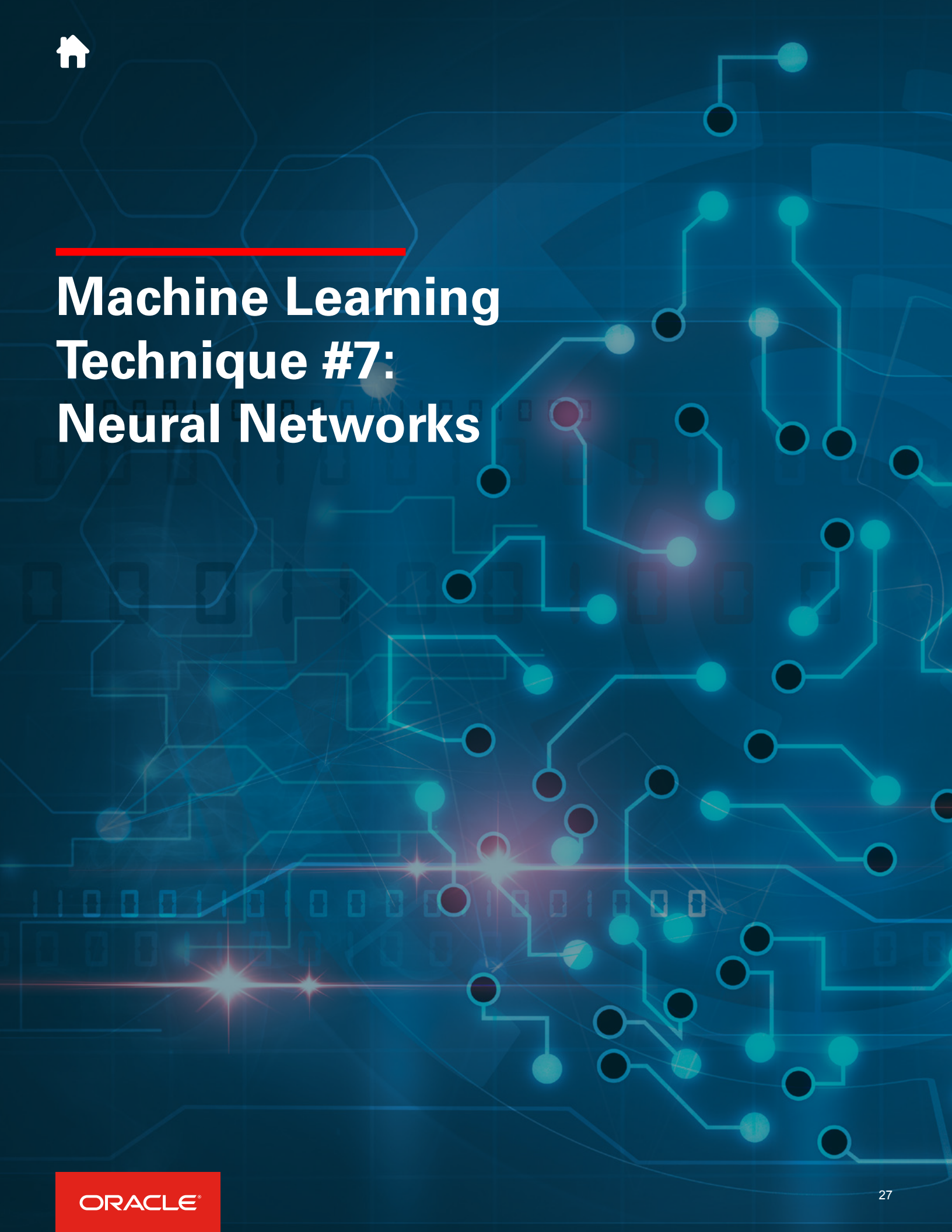
Machine Learning Technique #6: Time Series Data

You all experience time series data because your heartbeat, as captured in this EKG, is an example of time series data. Every 50-150 or more times a minute, your heart beats. Multiple sensors on your body might be used to capture all the data about your heartbeat and muscle contractions many times a second and in aggregate that will give you a time series data set. And there are many other examples of time series data that you might experience in daily life:

- Stock market opening and closing prices occur every 24 hours (holidays and weekends excluded).
- Vibration, noise, pressure or similar readings from mechanical devices might be measured every hour, minute, second or less (depending on needs).
- You could log energy consumption for houses, factories, subdivisions or entire cities.
- Even economic data like unemployment figures can be treated as time series data.

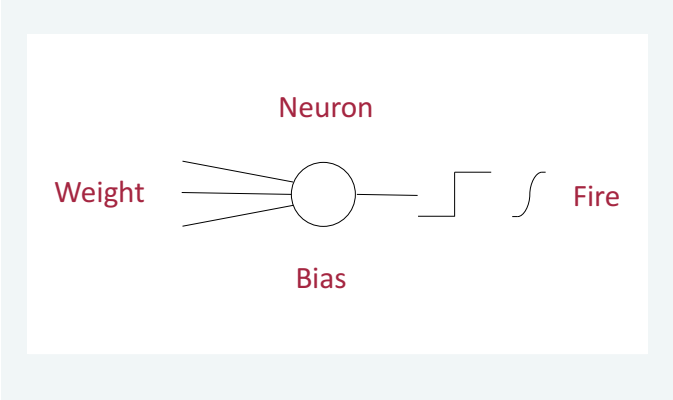
With this kind of data you can make predictions about when some reading might hit a threshold, what energy use might be at some given point in the future, or remove systematic time-based biases from data as when unemployment figures are adjusted to reflect seasonal ups and downs based on things like temporary holiday workers, people to bring in the harvest etc. Unemployment among ski instructors, for example, is higher in the summer than winter.

Machine Learning Technique #7: Neural Networks



Strictly speaking, neural networks are a type of algorithm. I cover them here as a technique because there's so much interest in understanding what they are and how they work.

The idea behind neural networks is to simulate the way in which human brains work using artificial neurons. Let's start by taking a brief look at what they are.

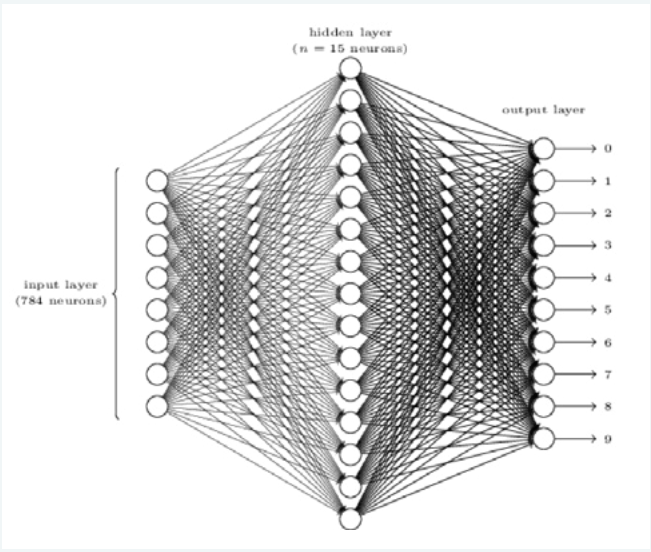


A neuron is conceptually fairly simple. It has one or more inputs and an output. Depending on the value of those inputs, the neuron may "fire." Simplistically, "firing" means that the output goes from off to on (think of it like a binary switch going from 0 to 1). In practice, neural networks based on neurons that flip like a binary switch can be unstable, so they generally have a pattern more like that s-shaped curve: 0 for a bit, then a fast (but not instantaneous) transition to a 1.

There are two other concepts that you should know about, called "weight" and "bias". Not all inputs to a neuron are equal; some are more important than others and are given more weight. For example, if you are going to a fancy dinner with friends, when picking the restaurant you might want to give more weight to the opinion of the foodie than to the fast food aficionado.

Finally, for this section at least, there is the concept of bias. What that really means is how likely is the neuron to fire in the first place. For example, when I say "let's go for a walk" our labrador puppy is at the front door before I've found the leash while the greyhound takes some more persuading. In this respect, some neurons are labrador puppies while others are retired racing greyhounds.

So now let's look at a real problem that can be tackled by neural networks: recognizing handwriting, specifically the numbers 0-9. You can get a set of sample scanned digits from here if you want to try this yourself and you can find an excellent and in-depth explanation on this problem here. There are some detailed mathematics if you want that level of detail, but you can read it and skip the math if you prefer. In this entry, I'll just summarize the story to give you the general idea of how it all works.



Those handwritten numbers were all scanned into 28 by 28 pixel images or 784 pixels in total. We'd start by having 784 input neurons on the left-hand side, each of which will fire based on the content of a single pixel in the scanned image. That hidden layer does some additional processing. And on the right-hand side we'd need a total of 10 output neurons representing the possible answers, and we'd expect one and only one of them to fire with each image.

So how does this network generate accurate answers? The details are complex (again, here's the article with more details), but basically we train it with many known images and look at the answers generated. Using some complex math it's possible to tune all the weights and biases in the network so we end up with a highly accurate tool for identifying handwritten numbers.

In fact, because they are more accurate than other algorithms on problems like this neural networks are used to process hand-written checks, so you've almost certainly encountered them.

If that description of training the network on a set of known answers sounds familiar, then you were paying attention when you read the earlier part of the article. That's actually the process for classification and what I described here is using an algorithm (a neural network) to do classification. Neural networks can also be used for regression problems.

Finally, since we're covering all the buzzwords in this article, let's touch on deep learning. It's broader than just neural networks, but one approach to deep learning could involve them. And as the name implies, rather than having a shallow network (here we have an input layer, and output layer, and just one hidden layer for a total of 3 layers), you could have a deep network with many more layers. Why do this? Because it enables looking at more data, and in a more sophisticated or nuanced way. This makes machine learning even more like a human.

So there you have it: different techniques for machine learning, and how they're used to solve real business problems. Perhaps you'd like to use them in your organization.

Machine Learning From Oracle

When it comes to incorporating new capabilities powered by machine learning into applications and tools, Oracle is on the cutting edge. The Oracle Autonomous Database is enabled by machine learning, while Oracle Management Cloud uses machine learning to automate the process of intrusion detection (amongst other things). And every day, Oracle is working to release new capabilities powered by machine learning.

But what if you want to deliver new capabilities powered by machine learning in your own applications and tools? To do that you need powerful tools to build, train, and test models against a wide range of data.

The Oracle Big Data Platform provides just that, and is built around three core principles:

OPEN

Open source is a key component of the Oracle big data strategy, and it remains the case here with support for technologies like:

- R language - Enhanced for greater scalability, performance, and integration
- Apache Spark - As a framework to run distributed applications
- Apache Zeppelin - Notebook to simplify the process of interactive data analytics

FAST

Open source technologies provide a great foundation, but there's always room for improvement. Oracle has optimized many source algorithms, making them work well in-memory and in parallel. So they are going to run fast. For example, one Oracle-improved algorithm runs 32 times faster on Spark than the standard implementation available with Spark MLlib.

INTEGRATED

Oracle machine learning capabilities are integrated with data management: keeping the analytics with the data helps to get results quicker since data movement is minimized. So Oracle Database Cloud and Oracle Big Data Cloud both include machine learning analytics that work directly on the data stored in those two services.

GETTING STARTED

If you would like to find out more about machine learning technologies there is in-depth information available about R technologies from Oracle as well as Oracle Advanced Analytics.

If you or somebody in your organization wants to test out machine learning on the Oracle Big Data Platform, then take a look at Oracle's Big Data Journey, a free trial.

Available journeys include "Building a Data Lake," "Self-Service Analytics," and "Data Science Acceleration" and they all include self-paced, guided trials that will enable you to get productive with big data and machine learning in the cloud.

THE LEADING CHOICE FOR MACHINE LEARNING

OPEN

Based on open source

FAST

Optimized in-memory algorithms

Up to 32X faster

INTEGRATED

Bringing analytics to the data

Oracle Corporation

WORLDWIDE HEADQUARTERS

500 Oracle Parkway
Redwood Shores
CA 94065
USA

WORLDWIDE INQUIRIES

Phone: +1.650.506.7000
+1.800.ORACLE1

Fax: +1.650.506.7200

oracle.com

Integrated Cloud Applications & Platform Services

Oracle is committed to developing practices and products that help protect the environment

CONNECT WITH US

ORACLE