

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

Abstract

If we want nontechnical stakeholders to respond to artificial intelligence developments in an informed way, we must help them acquire a more-than-superficial understanding of artificial intelligence (AI) and machine learning (ML). Explanations involving formal mathematical notation will not reach most people who need to make informed decisions about AI. We believe it is possible to teach many AI and ML concepts without slipping into mathematical notation.

Introduction¹

Artificial intelligence (AI) and machine learning (ML) are transforming industries, societies, and economies, and the pace of change is accelerating. Businesspeople, lawyers, policymakers, and other stakeholders will increasingly face practical questions (e.g., “Should my firm adopt this AI-related product?”) as well as political, ethical, and legal questions (e.g., “What limits should we place on law enforcement’s use of facial recognition technology?”) related to AI.

If we want nontechnical stakeholders to respond to AI developments in an informed way, we must develop ways to help them acquire a reasonable understanding of what AI and ML are and how different techniques work. Many institutions and individuals have been developing teaching materials, but this remains an open problem.

General overviews are often insufficient for the purposes of sophisticated stakeholders such as judges and regulators. However, in-depth explanations risk presuming a shared basic literacy about AI and ML. And even introductory technical

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

explanations often fall into the trap of overestimating the learner's prior knowledge, particularly the learner's knowledge of math. That may be a mistake. Explanations involving mathematical notation will not reach many people who need to make informed decisions about AI.

Sophisticated but nontechnical stakeholders should be empowered to think about AI and ML at a more-than-superficial level even if they lack the mathematical and technical background of computer scientists. In some cases, curiosity will drive them to "dig deeper"; in other cases, a superficial understanding will be inadequate for the decision-making task they face. A judge faced with dispositive motions in a software patent case—or a judge considering whether to overturn a conviction on the grounds that the police used some algorithm or ML technique in their investigation—will have an ethical duty to try to understand the technology at issue in the case if doing so is necessary to achieve a just and legally correct outcome.

We believe it is possible to teach many AI and ML concepts without slipping into advanced mathematical notation. As Steven Skiena has written, "[t]he heart of any algorithm is an *idea*."² Mathematical notation is rarely the only way to communicate such an idea, and it is possible to explain many ML concepts by analogies and examples while avoiding the terminology of calculus and linear algebra.

We have therefore developed a prototype website to communicate fundamental AI and ML concepts to an educated but nontechnical audience. This website can be found at <https://www.aiwithoutmath.com>.³ The site's name is a bit of a misnomer, as some machine learning concepts (such as "gradient descent") are inherently mathematical, and sometimes simple math examples make a concept easier to understand. Nevertheless, we intend to make the site's explanations as accessible as possible to people lacking mathematics backgrounds beyond high school algebra.

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

We aim to promote accessibility by providing simple but rigorous explanations of AI and ML concepts. These explanations go beyond high-level overviews and glossaries and attempt to teach the intuition behind various algorithms. To that end, they avoid formal mathematical notation and reduce the time commitment and cognitive effort required to learn the material.

We hope to create articles and learning materials that are as clear, concise, and self-contained as possible. We want to enable busy professionals to get the understanding they need without wading through details that may be relevant only to computer scientists or machine learning developers.

Further, instead of providing a linear walkthrough characteristic of a textbook or college course, we want to create modular, self-contained explanations of each concept that teach (or at least link to) the necessary background concepts. This will encourage non-linear navigation of the material that is driven by the reader's needs and curiosity. Many professionals lack the time or resources to complete an entire course on machine learning, but need a level of understanding that goes beyond that provided by high-level overviews or glossary entries.

Examples: Four AI/ML concepts explained without advanced math

In the following subsections, we illustrate our idea by explaining four AI concepts without relying on advanced math. Note that we also define all technical vocabulary as it is introduced. We try to give the reader the intuition behind each idea while assuming as little as possible about the reader's prior knowledge.

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

A. Rational agents

In AI, an *agent* is something that acts (such as a software “bot” or a robot). A *rational agent* is an agent that tries to achieve the best outcome. Agents are programmed to view some outcomes as better than others. The measuring stick by which the agent determines the “best outcome” is called the agent’s *objective function*.

The idea that an agent tries to achieve the best outcome is often stated in technical vocabulary. For example, some might say that the agent tries to *maximize utility*, *maximize expected utility*, or *maximize its objective function*.

B. Naive Bayes classifier

A Naive Bayes classifier uses probability to classify (categorize) an object.

What is a classifier?

A classifier is a program that categorizes items as one type of thing or another. For example, a picture might be classified as a “cat picture” or a “dog picture.” If you feed (input) an image into a classifier designed to distinguish cat pictures from dog pictures, it would output the *label* “cat” or “dog.”

Spam filter example

Email systems include “spam filters” that automatically determine whether each incoming email is likely to be spam. Their input is an email and their output is the probability that the email is spam.

Each time we estimate the probability that a particular email is spam, we should take into consideration the overall probability that an email is spam. Suppose that 45% of emails are spam (this is called a *base rate*). If we know that an email arrived but we

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

do not know anything else about it, we should conclude that it is probably not spam, because most emails (55%) are not spam.

However, each incoming email may have certain features that make it more likely to be spam. For example, every email has a certain number of exclamation points; “number of exclamation points” is a feature. We could say that an email containing more than two exclamation points is 80% likely to be spam. We can keep identifying more features and comparing them across the emails we see.

After considering many emails which are already labeled as “spam” or “not spam,” the algorithm will know that some features have certain values for spam emails and other values for non-spam emails. This knowledge is called a “model”; generating that knowledge is called “training the model.” Once we have a trained model, we can compare the feature values of any new email to the values in the model to estimate whether the new email is spam or not.

What makes Naive Bayes different from other classification methods?

A major advantage of the Naive Bayes method is that Naive Bayes models are relatively simple and can be trained quickly with a small set of data.

One potential disadvantage is that, in the Naive Bayes algorithm, each feature is considered independently of the others. This *simplifying assumption* is why the algorithm is called “naive.” In reality, many variables “travel together” and are not independent. For example, if the weather is rainy, it is probably also cloudy. But in a Naive Bayes analysis, “raininess” and “cloudiness” might be treated as two separate and independent features. For many tasks, however, treating these features as independent does not affect the outcome.

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

C. Linear regression

What is it?

Regression is a way of predicting an output value based on one or more input variables. For example, a regression model might attempt to predict house prices (an output value) based on input variables such as number of rooms, school district, and proximity to the ocean.

Input variables are also called *explanatory variables* because they attempt to “explain” the reasons for the output value. In the housing prices example, the input variables “explain” why each house is sold at a certain price.

What are some business use cases for regression?

- Given past prices and economic indicators, should we expect copper prices to rise or fall?
- How might sales change if, instead of investing \$100k in TV advertising, \$50K is invested in TV advertising, and \$50K is invested in social media advertising?
- If we hire additional doctors, how much can we decrease patient wait time?
- What are the top five factors that can cause a customer to default on their loan payment?

When should you use linear regression as opposed to another machine learning algorithm?

Linear regression should be used when (1) your target output value is a continuous numerical value, and (2) you expect a *linear* relationship between the input variables and the output value. A linear relationship means that if the input variable (e.g., number of rooms) goes up, the output value (e.g., housing price) should go up as

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

well.

Note that one strength of machine learning and neural networks is that they can learn complex, *nonlinear* relationships between input data and the target variable. However, more complex algorithms are not necessarily better for a particular task. It is important to choose algorithms whose complexity is similar to the complexity of the underlying data relationship. If there is reason to suspect that there is *not* a linear relationship between input data and a target variable, however, a linear regression model will be too simple, and a neural network or alternative algorithm (such as *polynomial* regression) will be a better fit.

How does linear regression work?

A linear regression model works by fitting a line (when there is one input variable) or a plane (when there are two or more input variables) to historical data. “Fitting” means that the algorithm finds the line or plane which best explains the output value, given the historical data.

The image below shows a linear regression model with a line fit between one input variable (x) and one output value. The x -axis shows the value of a single input variable (such as a neighborhood’s safety rating), while the dots show the historical data—the value of different output value and input variable pairs (such as house price and neighborhood safety rating). The linear regression model determines the fit between these variables by finding a line which minimizes the distance between the line and the data points. The values of y on the line give the prediction of the output value. For example, a neighborhood safety rating of 20 would predict a house price of 8.

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

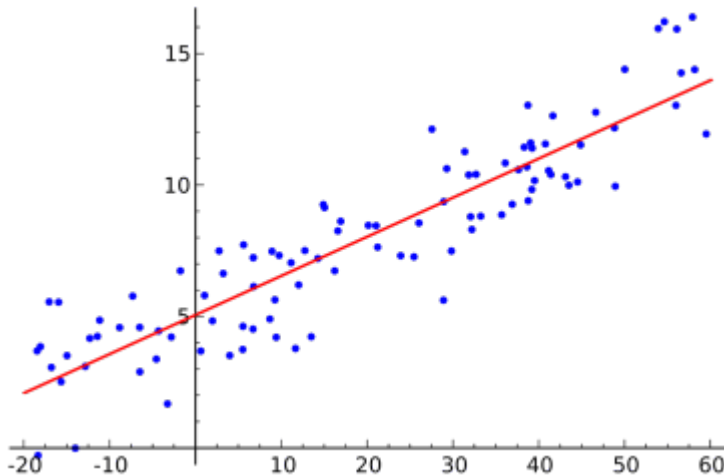


Image source: Wikipedia.org

D. Convolutional neural networks

Researchers designed *convolutional neural networks* (CNNs) because they needed better tools to process images. Most systems that “see” the world—self-driving cars, medical diagnostics, etc.—use a convolutional neural network to do so.

Neural networks are a class of algorithms that are used in many machine learning problems. Their basic building block is a neuron, which performs a simple operation on its input. These neurons are arranged into layers, which are connected in a network that can perform complex tasks. Engineers have lots of flexibility when structuring layers and connections, making these algorithms suitable for many problems.

Convolution is a special type of operation that answers the question, “How much of B is in A?” where A is often an image, and B is often a pattern. For instance, if A is an image of a house, and B is a horizontal edge, the convolution might return an abstract image of the house showing only its horizontal lines.

By connecting many such operations into a neural network, CNNs are able to detect

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

increasingly complex features. For example, in a CNN for face detection, early layers look for edges, intermediate layers look for facial components, and later layers look for full faces.

Future work

We hope to expand the AI Without Math website to include many more technical topics as well as nontechnical concepts (such as “explainability”) that form the shared vocabulary of AI researchers. Other ideas for the website include offering alternative explanations for complicated topics (perhaps with some voting mechanism, similar to that used on StackOverflow and Quora, in which readers can upvote and downvote explanations); linking to off-site explanations; and illustrating concepts with multimedia resources such as videos, games, and demonstrations.

Conclusion

Most resources on AI and ML are either too general or too technical. There are many high-level overviews that can give stakeholders a sense of what these concepts refer to, but many stakeholders will need more than just a broad overview or glossary. Some will want (or need) to peek “under the hood” of AI and ML technologies to get a basic understanding of how they work and why one might use one technique as opposed to another. There is an urgent need for educational resources that will help more people participate in decisions requiring a more-than-superficial understanding of AI and ML. We hope that the AI Without Math website can become a hub for those resources.

1. The ideas in this paper, and the website <https://www.aiwithoutmath.com>, were

by: Ryan McCarl, Jodie Lobana, Heather von Stackelberg, Kristen Schell, Brandon Leshchinskiy, Dania Humaidan and Gursimran Singh

imagined and prototyped at the 2019 Summer Institute on AI and Society hosted by the Alberta Machine Intelligence Institute and sponsored by CIFAR and the PULSE Project at the UCLA School of Law. The following people participated in the workgroup and developed the initial website and its content: Ryan McCarl, Dirk Hovy, Heather von Stackelberg, Jodie Lobana, Dania Humaidan, Gursimran Singh, Kristen Schell, Martha White, and Brandon Leshchinskiy.

2. Steven Skiena, *The Algorithm Design Manual* 12 (Springer 2d ed. 2009).
3. This website is a work in progress; in its current version, it is meant to be only a prototype rather than a finished product. However, we are confident that it fills an urgent need and ought to be developed as soon as possible. If you can contribute in any way (content, design, funding, publicity), please contact Ryan McCarl at mccarl@law.ucla.edu.