An Introduction to the "How To" for AI and Machine Learning

Steven Kursh, Software Analysis Group, Cambridge, MA and Vanderbilt University
Arthur Schnure, Software Analysis Group, Cambridge, MA

ABSTRACT

This paper provides an overview of Artificial Intelligence (AI) and Machine Learning (ML) with a focus on discussing the processes used to develop and implement an AI application. We review the key components in building an AI application and then proceed to discuss the "how to." The review of key components covers six major components including data sources, source code, and algorithms used to train the application (*i.e.*, machine learning) and use of an AI application to fit specific business needs. The paper also provides background on biases with AI-related work as well as samples of different algorithms and models. Finally, we discuss next steps for the reader to consider if s/he wants to pursue AL and ML projects at her/his organization.

Keywords: Amazon Web Services, Artificial intelligence, Google Cloud, Machine Learning, Microsoft Azure, Natural Language Processing, Python.

INTRODUCTION

Just over a decade ago, Marc Andreessen, a successful venture capitalist who founded Netscape, the first company to develop and market a browser, wrote in *The Wall Street Journal* that "…software is eating the world." His words then were accurate, but today are even more relevant. Per an April, 2021 Report from Grand View Research annual revenues in the software industry now exceed $400 billion per year and are growing at a very fast rate. Nearly all of us are dependent on software in our lives, whether it's the applications on our mobile phones, the tools we use at our desks, and the services and products we depend on from local retailers, including restaurants, and merchants like Amazon. The software industry will continue to grow with software products becoming even more integrated into our daily lives.

An important and critical component behind the recent and future growth in software functionality is artificial intelligence and machine learning. Each of us already benefits from software with artificial intelligence and machine learning, and the best is yet to come. Consider, for example, how your mobile phone now automatically corrects your spelling and suggests words when you are texting. Or how Uber uses AI and machine learning to support its services that you may use. Another example of AI and machine learning in use today is Netflix's recommendations of movies that you would enjoy. Computer vision, language translators, and autonomous vehicles are also examples of AI and machine learning in use.

Many venture capital firms are making significant commitments to AI and machine learning. A recent report from Pitchbook, a well-respected research firm, notes that there were over 2,330 deals involving AI and machine learning, totaling about $57.5 billion, in the first half of 2021. These investments will yield even more robust AI and machine learning software with the attendant benefits (and risks) to us.

Unfortunately, AI and machine learning are often considered a mysterious and strange technology that only the most technically-advanced people can understand and use to grow corporate value. In fact, this isn't true. Yes, AI and machine learning are software and, yes, software in general is challenging for many people; a combination of art and science that initially appears to be far too difficult to master, albeit even for understanding how it can be used in our companies and organizations.

Our objective in this article is to provide an introduction to "the how" behind AI and machine learning. Your time spent with this article will give you a critical headstart and perspective for understanding AI and machine learning. Of course, there will be much more to learn, but as you will see, AI and machine learning are well worth your attention and time.

We'll begin with a review of this mysterious and strange technology with an overview diagram of AI and machine learning in operation. We'll then move to a discussion about the key components with an AI and machine learning application. We'll also discuss risks associated with AI and machine learning. The next section then describes how to create AI models and justification for investments for AI. Finally, we provide some background information on what to consider if you want to build software applications that use AI and machine learning.

AN OVERVIEW OF AI AND MACHINE LEARNING SOFTWARE

We'll begin our detailed discussion with some definitions.

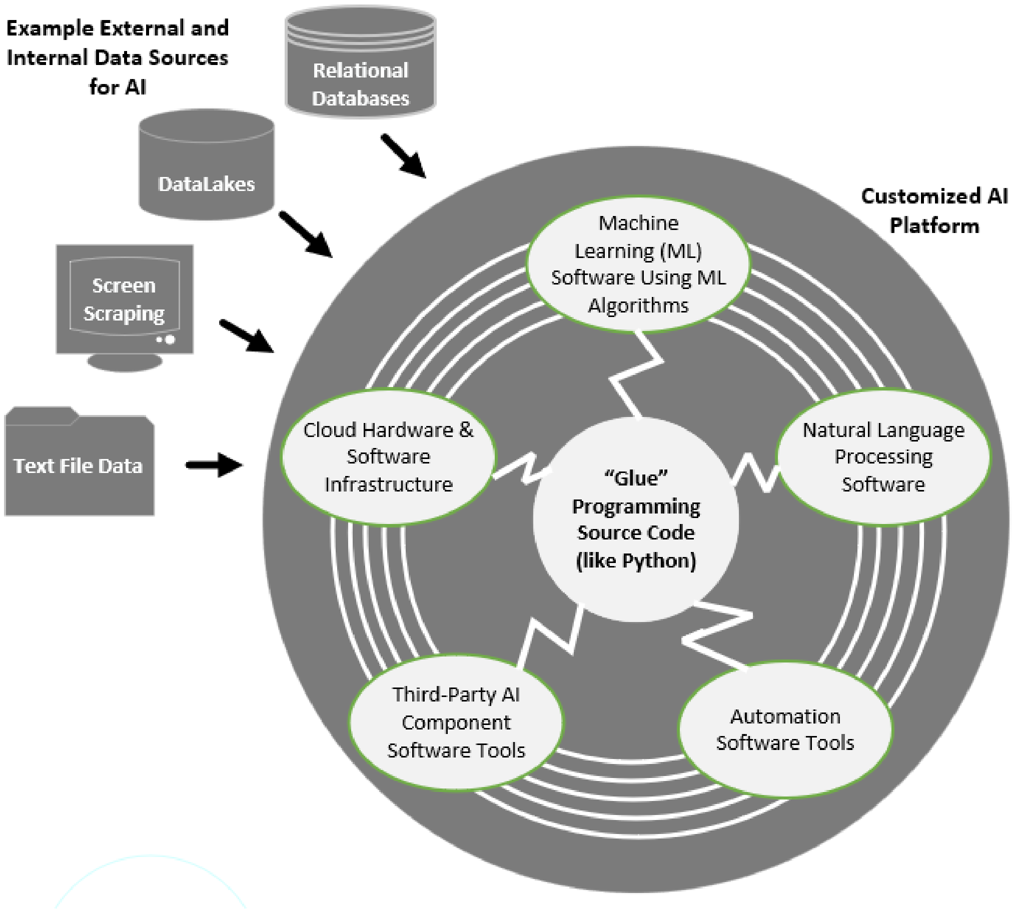
First, Artificial Intelligence. The term AI has been publicly known for decades and, accordingly, has many meanings in use. Reflecting the use and relevance of the term AI in business and science now, our definition is based on work by McKinsey published in 2018: AI is typically defined as the ability of a machine to perform cognitive functions associated with human minds, including perceiving, reasoning, learning, and problem solving.

Next, Machine Learning. Machine Learning is a tool within AI that involves the analysis of large data sets. The machine-learning algorithms detect patterns and make predictions as well as recommendations by processing the data. Additionally, the algorithms self adapt in response to new data and experiences.

Primary Components with AI and Machine Learning Software

There are six major software components with AI and machine learning. The diagram below, Exhibit One, provides a summary look at five key components used as well as referencing a sixth component, the AI programming source code, as shown in the center.

Exhibit 1. Data, Primary AI Components, and Source Code in a Customized AI Platform


Example External and Internal Data Sources for AI
Relational Databases
DataLakes
Screen Scraping
Text File Data
Customized AI Platform
Machine Learning (ML) Software Using ML Algorithms
Cloud Hardware & Software Infrastructure
"Glue" Programming Source Code (like Python)
Natural Language Processing Software
Third-Party AI Component Software Tools
Automation Software Tools

The four data sources at the upper left of this exhibit emphasize the preeminence of data for AI, since AI absolutely requires ongoing and quality input data. Data sources range from structured data (*e.g.*, text and numbers organized in rows and columns) and unstructured data. Unstructured AI data can be images, recordings, videos, graphics, emails, or web pages – showing that AI input data is not confined to text and numbers. Unstructured data can be stored in what is called a data lake, which is a data repository like the much-used software program Hadoop, part of the Apache Hadoop software library. (Hadoop is named after a child's stuffed elephant toy.)

The first (of the six components) is the central "glue" programming source code. Source code ties the components together, emphasizing the fact that AI is not out-of-the box software that we can license online and, thus, requires AI-experienced personnel and programming knowledge. Programmers often use the Python programming language (discussed below) for AI due to its relative simplicity, readability, widespread use, and flexibility. A key Python AI advantage stems from the multiple "add-on" Python libraries to handle AI and machine learning tasks that can be called from the lines of Python source code. Here we also have the algorithms used for machine learning.

There are many programming languages and even non-technical people may have been exposed and did programming work in the past, even if it was just macros in Microsoft Excel. Among the many programming languages available, Python has become the "defacto platform for new technologies," including AI and Machine Learning. *IEEE Spectrum*, a well-known and recognized publication from the IEEE (Institute for Electrical and Electronics Engineers) Computer Society, in an article published in August 2021 discussed how Python is the top-ranked programming language.

As noted above one of the reasons why Python is so popular is the depth and breadth of Python tools; examples of well-known and readily available, Python-friendly machine learning (ML) software tools are H_2O AutoML, TensorFlow, and Scikit-learn. Another Python library is NumPy, providing multiple ways to manipulate data. Pandas is yet another tool good for reshaping matrix data from sources like Excel and plotting output graphs. More information on these software tools is readily available via a Google search.

Second, and going clockwise from the exhibit's top, the machine learning (ML) software uses algorithms and statistics to learn and to adapt from its exposure to input data over time. Simply put, ML software gradually learns from multiple passes through the ever-growing input data, with an AI professional or the ML software automatically tweaking an AI model's parameter values with the data passes.

Unlike a computer program written by a human, machine learning creates its program (called a model) based on the desired result guiding the ML software to learn from the input data. The constant passes through input data are a critical aspect of how AI and machine learning software "learn and adjust" with more information. Large data sets are critical, as we discuss below in more detail.

A critical part of the ML software is the algorithms used for learning from the analysis of the data. We'll discuss in more detail below, but a quick example can be helpful. Those of us who have taken statistics or research methods are likely familiar with regression models. Essentially, regression enables us to model relationships between independent variables, say one's blood pressure, with dependent variables, in this example, weight, BMI, age, and type of diet. While not showing causation, we know that the dependent variables can help to explain variance in weights among people. Linear regression is a simple and popular algorithm for machine learning. So, if you are familiar with linear regression, you are already on the way of learning how to do AI and machine learning.

Third, working closely with machine learning, natural language processing (NLP) software is another essential since it enables computers to understand written and spoken words. NLP software is so critical that earlier in 2021 Microsoft acquired Nuance Communications, a company known for conversational AI, particularly in medical applications. Microsoft spent $16 billion and justified the purchase publicly by stating that the acquisition will enable it to expand its addressable market in healthcare alone by $250 billion. NLP is used in various ways with AI and Machine Learning projects. Bain & Company noted in a Bain Insights article published in March, 2021 that it used NLP and machine learning in a survey study for a global retailer to yield richer and more robust findings, particularly in regards to the analysis of responses to open-ended questions.

Fourth, due to the massive amount of data required and the steps needed to create an ML model, automation software is essential to manage many mundane AI tasks, since you want your skilled AI personnel focused on high-level work. Automated tasks include searching/collecting data from near and far, performing data cleansing tasks, and starting/monitoring machine learning tasks.

Fifth, the proliferation of third-party AI component software tools offers a wide choice of AI customization functionalities to suit your AI business needs. An example is PyTorch, a third-party add-on component for the Python programming language, which is "an open-source machine learning library framework that accelerates the path from research prototyping to production deployment." (https://pytourch.org.) The software toolset market also has components that work with other programming languages, but Python is highlighted due to its current dominance in AI. It's common to have scores of such open-source products used to create and maintain a customized AI platform, ranging from programming language extensions like PyTorch to non-AI tools like security scanners.

Sixth, AI platforms are usually located in the cloud, due to AI's need for massive data storage and processing power for developing and hosting production AI platforms. The primary cloud vendors including Google Cloud, Microsoft Azure, AWS, and others have created specialized AI tools to develop and host AI platforms. These toolsets will continue to grow in options, speed, and functionality.

AI'S DATA SOURCES AND ETL (EXTRACT, TRANSFORM, LOAD)

We noted above the importance of having data, lots and lots of data, to create and use an AI application. Much like we learned in our lives from experiences, AI and machine learning need "experiences" to analyze, hence, the tremendous demand for data. Consider, for example, an application like Netflix uses for recommending movies. Similarly, one company now offers driverless lawn mowers for golf courses with AI functionality that gets more efficient and better based on its experiences mowing.

Data: Critical to Your AI Application

An initial creation of an AI platform may be limited to an initial set of data sources, with more data sources planned in later releases. The depth and breadth of available data is expanding rapidly worldwide, so responsible organizations should do ongoing data searches to discover new valuable data "grist" to add to their AI platforms.

If the AI input data is incomplete, unrepresentative of the real world, or just wrong – the AI output information will also not represent the real world. Like humans, ML software learns through experience through data, but the software can make sense of massive input data sets that humans can't comprehend. Sometimes what the ML software "learns" is incorrect. Nevertheless, when the software learns correctly it is astonishingly powerful and useful. Consider AI's use in medicine. A more prosaic use is software offered by companies that through the use of AI and machine learning enable recognition of a vehicle often without the complete image of the license plate, a task that humans can't do.

Once AI data has been acquired via various means, it must be cleaned and enhanced to be made usable for AI using ETL (Extract, Transform, and Load) software. Unstructured raw data is often stored in a data lake repository, where it can (if needed) later undergo the ETL process to make it usable for AI. In fact, even structured data can require the ETL treatment, with the resulting cleansed data stored in a structured database.

Getting (Extracting) and storing (Loading) are easier tasks compared to "Transforming," since the transformation process can require substantial logic to make the needed data changes. ETL software conceptually might seem easy, but data is frequently "dirty" or poorly formed. For example, incoming city names may be misspelled, correctable by ETL software through zip code lookups or through referencing information for common misspellings of cities. (Think, for example, of "De Moines" rather than the correct spelling of "Des Moines," the capital city of the State of Iowa.)

Consequently, much software has ETL built in, such as the Excel data import feature, but sophisticated AI data cleansing and enhancing requires a high-powered ETL tool. Two examples of non-trivial transformations are: a) transforming a time to a 24-hour format to remove the AM and PM time formats, and b) cleaning data to remove typos and bad zip codes.

ETL software has been around for many years and ETL is now available via the cloud. Examples of cloud ETL tools are AWS Data Pipeline, Azure Data Factory, Google Cloud Data Fusion, and Informatica Intelligent Data Management Cloud. The AWS ETL tool is a web service API (application programming interface that enables two applications to share information, for example, the weather app on your phone) that gets transformed "…according to a predefined chain of data dependencies, operations, and a given schedule." Similarly, the Azure, Google, and Informatica cloud-based services provide ETL tools to move, transform, and store data.

The Risks of Bias

Accurate data is critical for AI, so it's important to realize a number of biases could well exist in data sources, consisting of four primary bias areas. We'll list multiple detailed biases within each area to emphasize the myriad ways data can be biased: The first bias area is Data Creation Biases which includes Sampling Bias, Measurement Bias, Label Bias, and Negative Set Bias. A common Sampling Bias is over-representation of one type of AI learning data to the detriment of another data type, such as face recognition AI data concentrating on light-skinned faces with darker-skinned faces under-represented, causing poorer AI recognition of darker-skinned faces.

Second, a Problem Formulation Bias is caused by a Framing Effect Bias, which is the original AI business intent unduly framed to favor a certain outcome. For instance, AI predictions of credit-worthiness might be framed to maximize a company's profit margin or to maximum loan repayments, which benefits a company and not the applicant.

Third, Data Analysis Biases includes Sample Selection Bias, Confounding Bias, and Design-Related Bias. An example of a Design-Related Bias comes from AI algorithms relying on randomness to accurately distribute results. However, insufficient computing power or the algorithm itself skews data selections towards the beginning or the end of lists, making results non-randomly distributed which causes inaccurate results.

Fourth, Validation and Testing Biases can consist of Sample Treatment Bias, Human Evaluation Bias, and Test Dataset Bias. A Sample Treatment Bias can be caused in AI input data preparation testing when certain people speaking a different language don't see ads shown to everyone else during the testing, skewing the results since certain testers saw a subset of the data.

AI Biases can throw off machine learning results, so forewarned is forearmed to minimize the errors possibly inherent in the data and the processes used by AI to learn. As such, data biases are important to ensure an organization's data is accurate at the beginning. Ask questions about data biases at your company when the AI effort is getting underway.

A CUSTOMIZED AI PLATFORM – COLLABORATION AMONGST THE COMPONENTS

Machine Learning Software

An AI platform is built from many components, but the most elemental component of all is the machine learning software, which learns from input data using automated statistical methods. In effect, the computer gains experience over time so it can make decisions on its own, make predictions, and improve outcomes. As an example, often when you use Amazon, the company's software looks at recent purchases and recommends products that you may be interested in purchasing.

Going in, it's clear to understand the degree of flexibility needed with machine learning software, since some ML software might lack the flexibility to address your AI needs. Machine learning software (services) from the Big Three cloud vendors are as follows: Google has Cloud AutoML and BigQueryML; Azure has Azure ML; and AWS has SageMaker. However, there are multiple third-party ML software open-source vendors such as H_2O AI and Scikit-Learn. The Scikit-Learn ML software offers "Simple and efficient tools for predictive [ML] data analysis." Overseen by data scientists, these ML software tools ingest large amounts of AI input data to gradually learn over time (like humans,) leading to more accurate predictive results.

Good software design applies to all kinds of software, but the following characteristics are particularly notable for AI software. The user interface should be easy and intuitive to create/manage the data inputs required for AI jobs and for viewing/analyzing AI outputs. As with all software, documentation should amply describe the sophisticated algorithms, required ML inputs, and available ML outputs.

Given the massive data required and extreme processing time, ML software should be fast and, thus, may require specialized software to scale with speed. Cloud vendors offer readily-available abilities to scale out or scale up. Scaling up focuses on the server by making it more efficient as well as adding more robust hardware capacity. For instance, an application-specific integrated circuit (ASIC) name "Tensor Processing Unit" (TPU) has been developed by Google to work with its neural network ML TensorFlow software, resulting in less energy use and speed increases. Better-known GPU (Graphical Processing Unit) ASICs can also improve a computer's energy use and speed for AI.

Thus, scaling up increases the computational work one server can do. Scaling out, also called scaling horizontally, distributes the computational workload among multiple ML servers via load balancing. Load balancing software acts like a traffic cop by monitoring incoming requests and distributing the work to multiple servers so no one server is overloaded.

ML software should be readily customized or configured to suit an organization's business needs. Of course, 100 percent customization through the ML software isn't possible for AI platforms, since programming languages are often needed to tie together AI platform components and to address AI data science needs. For instance, the Python programming language can customize an AI platform by calling third-party Python libraries to access sophisticated ML algorithms, extracting data, loading data, and providing sophisticated statistical graphs.

ML software should handle the needs of all levels of data science personnel, complete with specialized security roles to expose "need to know" information to the appropriate user roles that align with the various personnel skill sets in an organization. Microsoft's Azure Machine Learning has three default roles: Reader (allows read-only rights); Contributor (allows create/read/update/delete rights for many functionalities); and Owner (allows create/read/update/delete rights for all functionalities.) Azure also allows customized user roles to be created, improving an organization's security by ensuring sensitive data is only seen by personnel who need to know the information.

ML Model and Data Governance

ML model governance is important to keep machine language outcomes accurate and well maintained. Top organization executives should delegate ML governance so a mix of organization AI stakeholders can manage the overall functioning of ML. Regular meetings allow oversight of new model development, monitoring the AI inputs and outputs to ensure optimal training (learning) done by an ML model.

Overall, governance lets an organization align their business goals with the business outputs produced by AI. In our experience, some organizations combine AI data governance and ML model governance into one unit, since both data and models are the main drivers of AI effectiveness. Governance provides a process to analyze continuously input data against set standards, compare AI model outputs against set standards, and to apply governance contingency plans for use when standards are not met. Through governance, an organization's efficiency and bottom-line are benefited when ML prediction outputs are originally developed accurately and kept accurate thenceforth.

Natural Language Processing

As an adjunct to ML, Natural Language Processing (NLP) is a less invisible AI component, but it's critical. NLP goes hand-in-hand with machine learning, since a computer can't learn from written or spoken text unless NLP understands the content and context, such as performing co-reference resolution, which is a synonym-like ability to identify various words referring to one object. NLP can also translate text from one language to another, increasing the amount of data available to AI, using a tool like Google Translate. Or, NLP lets Google find references similar to your requests when doing a search. Many of us are familiar (and often, likely frustrated) by voice recognition software in use at many companies. Some of this software uses NLP.

Perhaps most importantly for AI, NLP can do "deep learning," which is a subset of machine learning. Deep learning uses neural networks, which is software mimicking the human brain, to understand language spoken or in text that can reveal important data nuggets amidst massive data sets. Many NLP products are on the market, such as Google Translate, Amazon Comprehend IBM Watson, OpenNLP, and MonkeyLearn.

Some NLP products are best suited for large organizations and other products, like InBenta that specializes in a customer experience data niche. Keep NLP's functionality in mind when thinking about an ML product to understand the extent of NLP functionality beforehand.

The Importance of Automation for AI

AI platforms require massive inputs of data and are invariably composed of many discrete processes before useful AI results emerge. Due to the magnitude of the work and the steps needed, automation is absolutely critical to make AI accurate and cost-effective by reducing labor hours and reducing the time required spent by skilled personnel to do menial tasks.

One automation tool example is Azure Automation to automate cloud-based tasks, provided you have an Azure cloud account, though the tool can access other cloud vendors and on-premises data centers. Another automation tool is Run:AI, featuring intelligence for neural network deep learning that requires massive computing power to manage GPUs efficiently to increase speed and save energy.

Third Party AI Component Software Tools

A robust third-party market exists to help users construct an efficient AI platform. The tools range from major components like ML software to discrete software services such as tools to detect "data drift," which detects changes in AI input data over time (that can negatively impact AI results. For example, demand for some consumer and enterprise products changes over time due to various factors like the weather and fashion trends; accordingly, the AI software needs to be updated with fresh and more current information.

Many of the available third-party components are open-source software constructed to be included within an AI platform. Multiple external open–source AI products need to collaborate well. Responsible software developers, including those creating AI functionalities, must ensure external open-source functionality comes with source code and meets certain standard criteria such as trustworthiness, good maintenance practices, legally-sound, popular, and likely to be supported over time. Be sure to ask questions in regards to open-source functionality if and when you get involved with an AI project at your organization.

Once external dependencies are placed into an organization's source code, they're often difficult to remove. To guard against inclusion of open-source vulnerabilities in an AI platform, third-party dependency scanning security tools can be used to detect and remedy vulnerable open – source components. An example is Dependabot, a bot to scan the GitHub source code repository widely used to store an organization's source code. Dependabot "creates pull requests to keep your dependencies secure and up-to-date." GitHub's free repository hosts source code, provides version control and access control. GitHub also provides collaboration tools to track bugs / enhancement requests and to do task management.

Another third-party product is NumPy, short for Numerical Python, which offers mathematical functionality to deal with sound waves and images. The NumPy library is open-source software written in the C programming language and compiled to increase AI computational speed when accessed.

Cloud Hardware and Software Infrastructure

The growth in cloud services has been nothing short of astounding. Notably, AWS's debut in 2006 featured just eight services, whereas Amazon had 262 in 2019. Among other new cloud services, AI has been added by cloud vendors to meet burgeoning demands. The following list of features emphasizes the broad range of AI functionalities available in the cloud: machine learning, natural language processing to understand written and spoken text, business intelligence to make sense of AI data, purchase predictions via AI-powered forecasts, incident automation to sense problems and automatically handle them, and multi-lingual support for language translation.

Naturally, more traditional ancillary services are important for AI, such as unstructured data lake repositories to store raw data, structured relational databases to store data in an AI-usable organized fashion, scaling up or out to increase speed and handle loads, disaster recovery to ensure AI system restoration in case of floods or hurricanes, backup & archiving to prevent data loss in case of hardware or network failures, and heightened security to prevent hacks.

HOW MACHINE LEARNING AI MODELS WORK

AI Model Creation

To get insight into the skill sets required to create models, it's useful to get a sense of the model creation process, the gradual learning done by ML software, and the challenges faced to produce a model that meets predefined success criteria.

ML software uses data to train a model, with the model constituting an AI "product" that can be reused over time via regular AI input data updates. ML software has four basic learning types: Supervised, Unsupervised, Semi-Supervised, and Reinforced. These four basic learning types are defined as:

- Supervised… involves making the algorithm learn the data while providing the correct answers using labels placed on the data. This essentially means that the classes or the values to be predicted are known and well defined for the algorithm from the very beginning.

- Unsupervised… Unlike supervised methods the algorithm doesn't have correct answers or any answers at all, it is up to the algorithms discretion to bring together similar data and understand it.
- Semi-Supervised… A hybrid with a mix of Supervised and Unsupervised learning.
- Reinforcement… in Reinforcement Learning, there are rewards given to the algorithm upon every correct prediction, thus driving the accuracy higher up.

Data science expertise is needed to determine the best statistical algorithms for use in the ML software to fit your particular data set. For instance, a long and accurate data history would suggest use of a certain statistical algorithm. Volatile data with a short history could require another algorithm. Product data with a high seasonality, such as big sales before Christmas, could require yet another algorithm.

The following algorithms are often used in AI: Naive Bayes used for sentiment analysis, spam detection, and recommendations; Decision Tree used for outcome predictions; Random Forest merges multiple decision trees to improve predictions; Logistic Regression used for binary classifications (A or B); Linear Regression used for categorizing a large data set; AdaBoost, Gaussian mixture, Recommender, and K-Means Clustering to organize data into groups like market segmentation or finding crime-prone areas.

Many of us may be familiar with the general concepts in these algorithms based on prior work in statistics, finance, research methods, epidemiology, physics, econometrics, and other fields. Consider Bayesian statistics, which you may recall from an earlier statistics course. Data set features are rarely independent of one another, but the Bayes algorithm "naively" assumes that they are independent. The naïve assumption combined with the Bayes statistical probability algorithm produces a good way to classify data, such as determining if an email is spam or not.

Plunging into model creation, there are three distinct learning stages for machine learning: Training Stage A; Validation Stage B; and Testing Stage C. Before starting the entire process, it's necessary to ensure the data is as well-organized and immaculate as feasible. Though the concept is simple, getting data wrangled into orderliness is a time-consuming and detail-oriented process to make the data free from duplicates and disconnected data. After cleansing, the data is divided up randomly into three sets to be used for each of the three training stages. The random data division is meant to discourage selection data biases.

Three relevant definitions in model creation are:

- Parameter. Model parameters are values learned automatically by the ML software from the AI input data as training progresses, although a user can manually change a parameter value during the training process. Examples are the maximum number of passes to be made during a session and the training data maximum model size in bytes.
- Hyperparameter. Values external to ML that are input beforehand by a data scientist user, so hyperparameter values are not derived from AI data and can be changed during the training process. Examples of hyperparameters are the number of clusters to be returned when using a clustering algorithm and the number of layers in a neural network.
- Variable. The particular AI data input fields chosen for consideration by the ML software, which can be modified as training progresses. Variables can be age, height, and weight.

Before starting Stage A (Training), it's important have labels added to the data to the ML software that provides clues to help it learn (unsupervised learning does not need labels). For example, the below exhibit for supervised learning shows two rows of data containing product sales data in text, potentially including a product brand name label and a product type label attached to each row of product data, per the below exhibit.

Exhibit 2. Example AI Input Data Labels for Supervised ML Training

Blog text to be analyzed by ML	Data "Helper" Labels
I chose the Craftsman 18" chain saw due to its ability to cut medium-diameter trees	Craftsman, chain saw
While I was at the store, I bought both a chain saw for $32.50 and loppers	chain saw, lopper

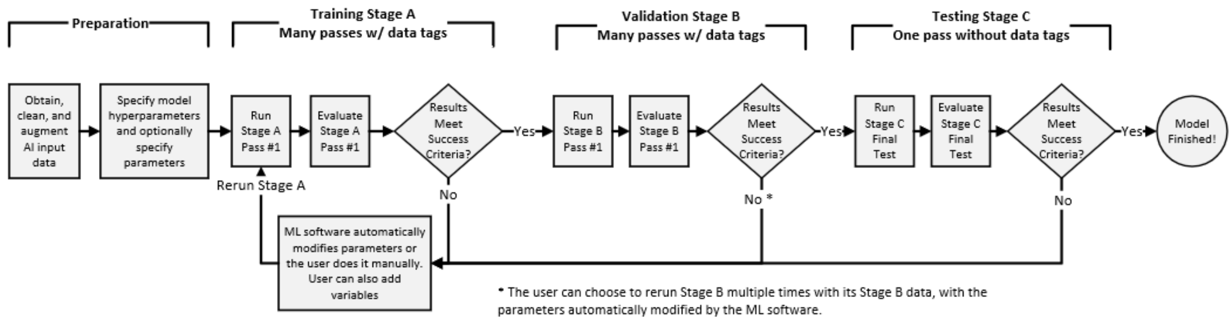
For Stage A, you can use the ML software default parameter values or you can change the parameters yourself. After Stage A Pass #1 finishes, parameters are automatically changed by the ML software through its learning – or a user can modify them before running Stage A Pass #2. You can proceed to Stage B once the Stage A results meet the success criteria. But most likely, it won't be successful after Pass #1, so Pass #2 and additional passes will likely be needed to proceed until the ML software reaches its preset maximum pass count or no new patterns are found by the ML software. It's very possible Stage A will show the input data needing more cleansing and augmentation.

Validation Stage B Pass #1 uses a new set of data. If the Stage B Pass #1 result exceeds the success criteria, you may proceed directly to Testing Stage C. Stage B passes can continue until the ML software shows no new patterns or it reaches the maximum number of passes. Negative results require a return to Training Stage A, where additional input data variables may need to be added. The parameters are automatically modified by the ML software or by you as training progresses. Stages A and B can alternatively be run in tandem, with the results of Stage A compared to the results of Stage B until agreement is achieved.

Testing Stage C is the "final exam" against a new set of data - but this time lacking the "helper" data labels (for supervised learning only). If it passes the test, you now have a working model. If not, it's back to Stage A for the user to potentially add new variables suggested by training done thus far. As before, you can manually modify parameters or let the ML software automatically modify parameters as training progresses.

The following exhibit lays out the AI training process flow.

Exhibit 3. Process Flow of ML Model Training Stages A, B, and C


Preparation
Training Stage A
Many passes w/ data tags
Validation Stage B
Many passes w/ data tags
Testing Stage C
One pass without data tags
Obtain, clean, and augment AI input data
Specify model hyperparameters and optionally specify parameters
Run Stage A Pass #1
Evaluate Stage A Pass #1
Results Meet Success Criteria?
Yes
Run Stage B Pass #1
Evaluate Stage B Pass #1
Results Meet Success Criteria?
Yes
Run Stage C Final Test
Evaluate Stage C Final Test
Results Meet Success Criteria?
Yes
Model Finished!
Rerun Stage A
No
No *
No
ML software automatically modifies parameters or the user does it manually. User can also add variables

* The user can choose to rerun Stage B multiple times with its Stage B data, with the parameters automatically modified by the ML software.

In short, machine learning is a repetitious replay of the ML software's exposure to data, with parameters automatically changed iteratively by the ML software (and/or by humans) to make the model smarter after each pass of the data. ML software does multiple passes of the data until it realizes no new patterns are being detected, causing it to stop.

AI Model Ongoing Maintenance

Constant vigilance (monitoring) is the price of AI freedom. To determine how well an AI model is doing, an obvious tack is to monitor how closely the actual performance matches the AI prediction. If the AI predictions worsen, it's time to reenter the model training process to correct the model using up-to-date data.

As mentioned earlier, input data can easily change over time - called data drift in the trade. Data drift can cause the AI model's accuracy to deteriorate, so early data drift warnings are important to stay ahead of problems. AI tools are available to track data drift and find outlier data, such as Fiddler, Neptune, and Azure ML, which can supply early warnings to so data problems can be addressed by ML updates sooner rather than later.

Justifying and Explaining AI

It's one thing to have AI provide accurate predictions, but how does one assign numbers to an AI model to determine ROI? All organizations will want to know how much business value has been created over time by AI models. Many AI vendors have in-house tools to figure if AI models have provided returns that exceed the time and expense spent on AI development and implementation. These vendors have use cases to monitor the business value derived from AI models.

We suggest that investments in AI at this time should not be analyzed as compared to investments in technologies and other assets that have historical data on financial returns. Instead, AL investments should be considered in the context of the necessity to gain experience with a technology that will have a major role in the operations of many companies.

Try It!

To get started, there are many sources of free or payment-based AI training via on-line videos ranging from minutes to 12 hours using a web search of "artificial intelligence courses." Online articles, magazines and books also offer various levels of instruction from beginners to experts. You can start your AI quest by registering for accounts with cloud vendors AWS, Azure, Google Cloud, and many others, with some vendors even offering a free AI try before you buy. Providers of courses include Udacity, edX, MonkeyLearn, OpenAI, and many universities.

REFERENCES

Andreessen, Marc. "Why Software is Eating the World." *The Wall Street Journal*, August 20, 2011.

Cass, Stephen. "Top Programming Languages 2021." *IEEE Spectrum*, August 24, 2021.

Derman, Emanuel. *Models.Behaving.Badly.: Why Confusing Illusion with Reality Can Lead to Disaster, on Wall Street and in Life*, (New York: Free Press), 2012.

Grand View Research. "Business Software and Services Market Report, 2021-2028," April 2021.

Hellemons, Ruud and Rober Zhu, "How Machine Language Learning and Natural Language Processing Produce Deeper Survey Insights." *Bain Insights*, March 23, 2021.

Killalea, Tom."A Second Conversation with Werner Vogels." *Communications of the ACM*, March 2021, p. 50.

McKinsey, "An Executives Guide to AI," McKinsey Analytics, 2018, p. 1. Available at McKinsey.com, last accessed on August 31, 2021.

O'Neil, Kathy. *Mass Destruction: How Big Data Increases Inequality and Threatens Democracy.* (New York: Crown), 2016.

Pitchbook, "Emerging Technologies: Artificial Intelligence & Machine Learning," Q2, 2021, August, 2021.

Srinivasan, Ramya and Ajay Chander, "Biases in AI Systems," *Communications of the ACM*, August, 2021, pp. 46 – 48.

Websites

https://dependabot.com/.
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-assign-roles.
https://pytorch.org/.
www.openalpr.com.
https://pytorch.org/.
https://scikit-learn.org/stable/index.html.
https://www.upgrad.com/blog/what-is-aws-data-pipeline-and-components/.

Steven R. Kursh, Ph.D., CSDP, CLP is the founder of the Software Analysis Group and an Adjunct Professor in the School of Engineering at Vanderbilt University. His research interests focus on software intellectual property, customs and practices in the software industry, and the intersections between technology and business.

F. Arthur Schnure, PE, PMP, MSDP is a Senior Consultant at the Software Analysis Group. His research interests include software design, development and implementations. He has worked with large mission-critical software deployments.