GRC Transactions, Vol. 46, 2022

What Did They Just Say?

Building a Rosetta Stone for Geoscience and Machine Learning

Stanley P. MORDENSKY[1], John J. LIPOR[2], Erick R. BURNS[1], Cary R. LINDSEY[1]

[1]U.S. Geological Survey, 2130 SW 5th Ave, Portland OR 97201, USA

[2]Portland State University, 19000 SW 4th Ave, Suite 160, Portland OR 97201, USA

Keywords

geoscience, machine learning, communication, cross-disciplinary

ABSTRACT

Modern advancements in science and engineering are built upon multidisciplinary projects that bring experts together from different fields. Within their respective disciplines, researchers rely on precise terminology for specific ideas, principles, methods, and theories. Hence, the potential for miscommunication is substantial, especially when common words have been adopted by one (or both) group(s) to represent very specific, precise, but, perhaps, different concepts. Under the best circumstances, misunderstanding key terms will lead toward a breakdown of efficiency. Under less optimal conditions, miscommunication will sow frustration, lead to errors, and inhibit scientific breakthroughs. Here, our research group of geoscientists and machine learning experts presents a process to help geoscientists understand the fundamentals of supervised learning by describing the general workflow (*i.e.,* a conceptual pipeline) for supervised learning that must be understood by all the parties involved in a geoscience-machine learning endeavor. Terms critical for machine learning are introduced, defined, and used within the context of an overly simplified mock hydrological study to illustrate their appropriate usage, and then used again in the context of a published geothermal-machine learning study. These key terms are divided into two groups, which are 1) essential to the field of machine learning but are predominantly absent in geoscience or 2) homonyms (*i.e.,* words with the same spelling or pronunciation but with different meanings) between the fields. Lastly, we discuss a few other important homonyms that were not introduced in the general workflow but arise regularly in machine learning applications.

1. Introduction – A Preamble About How to Read This Paper

This document is noticeably different in intent and structure compared to other manuscripts about geoscience or machine learning. Rather than serving as a report and discussion on the results of an experiment or as a literature review, this work introduces the key terms a geoscientist needs to collaborate with a machine learning expert. No prior machine learning experience is necessary to understand this document.

In Section 2, we present a general machine learning workflow of seven steps to provide a contextual framework for key terms. The description of each step (Sections 2.1-2.7) is divided into a short, simple, plain language overview of that step followed by three subsections that contain skill-building information. The first subsection (*i.e.,* 2.*x*.1 in which *x* is the step number and 1 is the first subsection) is a translation and expanded explanation of the ideas from the plain language overview. Critical terms for machine learning are introduced and defined. ***Terms that are predominantly absent in geoscience but crucial for machine learning are bolded and italicized*** and, until the reader is comfortable with this new jargon, the reader can refer to Table 1 for a precise definition. **Terms that are homonyms (*i.e.,* identical terms with different uses) between the two disciplines are bolded** and, for lookup purposes, definitions for these terms are provided in Table 2. In the next subsection (*i.e.,* 2.*x*.2), a simple hypothetical hydrological machine learning study is used to develop and define the core ideas, demonstrating how machine learning terms are used in practice. The final subsection of each step (*i.e.,* 2.*x*.3) then uses these newly introduced terms in the context of a recently published geothermal study where machine learning was used to predict geothermal resource favorability in the western United States (Mordensky et al., 2022). The final two sections of the manuscript conclude the discussion, with Section 3 listing a few additional terms that the reader may find helpful when working with machine learning experts, and Section 4 providing conclusions.

1.1 What Is Machine Learning?

Since the advent of electronic computers in the middle of the 20th century, the computational resources available to humans have increased by orders upon orders of magnitude. Classical computer software operates with exact sets of programmed rules to process data and return results. Rather than relying upon these rigid rules, researchers began asking if the computer could instead learn data-driven rules from general mathematical and statistical functions to process the input data into results without explicit direction (*i.e.,* without programmed rules) from the researchers, thereby finding patterns with little or no user bias. These newly learned data-driven rules could then be applied to other data. ***Machine learning*** is the field of study that explores the construction and study of the mathematical and statistical functions that **learn** (*i.e.,* create correlations and/or decision functions) without direct instruction.

Before proceeding to distinguish the language differences between geoscience and ***machine learning***, one must consider the distinct goals of each discipline. Geoscience is the practice of studying natural geological phenomena. A geoscientist seeks to understand a natural process, and, by doing so, a geoscientist creates a model (*i.e.,* a conceptual or mathematical representation of the process) to describe that process. In the eyes of a geoscientist, the mechanics of a model should, in some form, reflect the important physical processes that are observed. A ***machine learning*** expert, in comparison, specializes in developing a **model** to identify patterns in the data, typically for the purpose of making **predictions**. To a ***machine learning*** expert, the internal mechanics of the **model** need not reflect physical processes so long as the **model** performs well for its intended purpose (*e.g.,* identifying groups with differing behavior). That is, to the ***machine learning*** expert, the performance of the **model** may supersede its operational consideration for its relation to the natural world. To the ***machine learning*** expert, the mathematics of classification or discrimination is the science.

1.2 What Words Should We Use?

As in any new relationship, the potential for miscommunication is high in multi-disciplinary collaboration. It is particularly so when one clearly prefers to speak in terms of natural processes about the Earth and the other prefers to speak in mathematics and algorithms. This is not a disparaging comment on geoscientists or ***machine learning*** experts, but an acknowledgement that we geoscientists have learned to speak a common language to effectively discuss geoscience topics. Unfortunately, unlike foreign languages that have completely different words for almost everything, ***machine learning*** and geoscience experts use the same words to mean very different things, and sometimes, when words appear to have very similar meanings, their subtleties are important.

Geoscientists come from a wide range of disciplines and retain diction intrinsic to their research specialties (*e.g.,* geophysics versus geochemistry), but they have also learned to speak a common geoscience language to effectively discuss geoscience topics during seminars, cooperate in collaboration, and share common facilities. Deep comprehension for this shared technical vernacular among geoscience researchers diminishes the farther afield the researchers' professions are from one another (*e.g.,* civil engineering, though not strictly a geoscience discipline, is tied to geoscience). Consequently, the jargon used by ***machine learning*** specialists shares more similarities with that of a mathematician or statistician than that of a geoscientist. If a ***machine learning*** expert and a geoscientist seek to collaborate, how do they resolve their terminology?

To the extent possible, geoscientists should learn the ***machine learning*** dictionary, because effective communication enables the geoscience community to take advantage of a wide range of tools developed for a myriad of purposes (*e.g.,* predicting resource quality, locating resources). Doing so helps the geoscientists to formulate geoscience questions appropriate for mathematical constructs of ***machine learning*** and to supply the correct information when working with ***machine learning*** experts.

In any research environment, minimizing miscommunication within and beyond the research team increases the efficiency and quality of the work. Clear communication is central to avoiding misreporting and misinterpreting the data. Hereinafter, we seek to outline the general ***machine learning*** process (*i.e.,* a conceptual ***machine learning*** **pipeline**) and, in that process, clarify key terms. We emphasize that the nomenclature we present is non-exhaustive, but instead serves as a foundation upon which to build and grow one's parlance.

1.3 Background for the Published Geothermal-Machine Learning Study used (below) to Illustrate Terminology

In 2008, the U.S. Geological Survey released the most recent mid- to high-temperature geothermal resource assessment using data-driven analytical methods; however, some aspects of the analysis were dependent upon domain knowledge (*i.e.,* knowledge of the geothermal subdiscipline) and expert decisions (Williams and DeAngelo, 2008; Williams et al., 2008; Williams et al., 2009). In 2022, a collaborative effort between the U.S. Geological Survey and Portland State University used the same data to produce ***machine learning*** **models** to compare with the expert decision-dependent models from 2008 (Mordensky et al., 2022). This collaborative effort between the two institutions and the frequent need to explain terms led to the inspiration to develop the translational dictionary presented in this document. Below, we use Mordensky et al. (2022) to illustrate the proper use of key terms presented in the general ***machine learning*** workflow.

2. A Machine Learning Workflow for Supervised Learning

There are three primary domains of ***machine learning***: ***supervised learning***, ***unsupervised learning***, and ***reinforcement learning*** (Fig. 1). ***Supervised learning*** is a domain of ***machine learning*** that is used to make **predictions** of a desired condition (*e.g.,* the presence/absence of geothermal resources). ***Supervised learning*** seeks to find a mathematical or statistical relationship between measured or interpolated input data and known output data, so that new output values (*i.e.,* **predictions**) can be made where the output values are otherwise unknown. 'Supervision' is provided by measurements of the phenomenon/condition that will be predicted at other locations (*e.g.,* heat flow measurements in wells can be used to construct a continuous heat flow map). ***Supervised learning*** has two primary forms of **prediction** (*i.e.,* **regression** and **classification**). **Regression** is the **prediction** of a number corresponding to the magnitude of a property (*e.g.,* a measure of heat flow or probability). **Classification** is the **prediction** of one of a finite number of pre-defined, discrete **class** values (*e.g.,* rock type, mineral assemblage). ın other words, regression is used for continuous variables, and **classification** is used for categorical variables. In contrast to ***supervised learning*****,** ***unsupervised learning*** finds statistical relationships between data, identifying which samples are correlated and which samples belong to 'similar' groups. Finally, ***reinforcement learning*** is a domain of ***machine learning*** that sequentially makes decisions that are either rewarded for a correct decision or penalized for an incorrect decision, leading toward improved future decision making (*e.g.,* some strategies for play fairway analysis). Each of the primary forms of ***machine learning*** is comprised of a range of methods (Fig. 1).

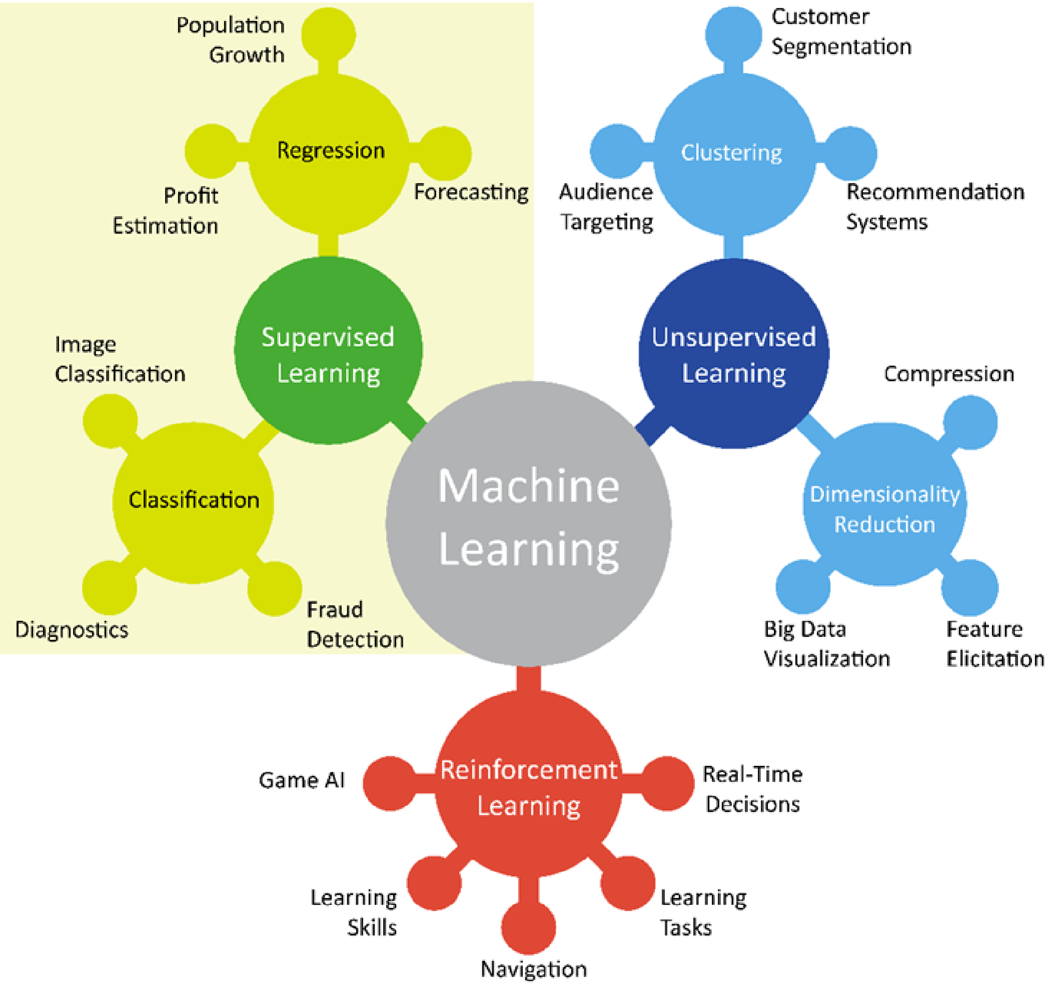


Figure 1. Machine learning family tree. Supervised learning, unsupervised learning, and reinforcement learning are commonly defined as the three primary domains of machine learning. Representative applications are given at end nodes. The yellow background highlights supervised learning, which is the focus of the workflow described in this document. Figure modified from the "Council of Europe" 2019).

Herein, we focus on a general workflow for ***supervised learning*** because the goal of making a **prediction** of interest based on available supporting geoscience data sets is a fundamental activity of geoscientists; however, we note that while the workflow we describe is for ***supervised learning***, the ***machine learning*** terms are widely applicable across other forms of ***machine learning***. We divide ***supervised learning*** into a workflow of seven distinct steps (the conceptual **pipeline**; Fig. 2):

1. Identify a Clear Question
2. Explore the Data
3. Engineer Features
4. Choose an Algorithm
5. Train-Test Split the Data
6. Optimize, Train, and Evaluate the Model
7. Make Predictions

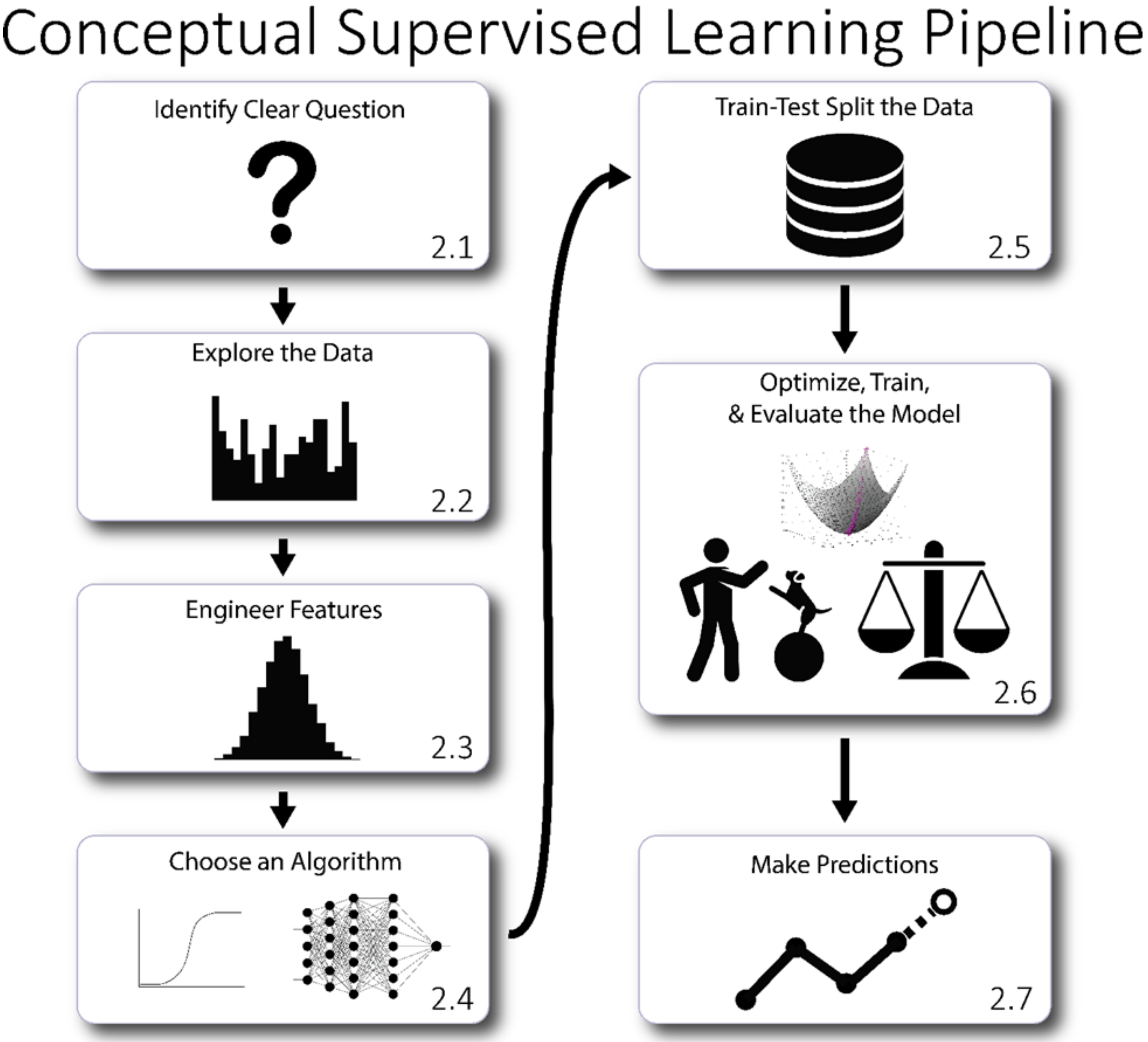


Figure 2. Conceptual representation of a supervised learning pipeline. Under ideal circumstances, the above image reflects the workflow one would follow to conduct supervised learning. Corresponding section numbers have been provided in the bottom-right of each step's pane.

2.1 Identify A Clear Question

Avoiding confusion and miscommunication begins at the pre-planning stages of a project. The bottom line with ***machine learning***, and even more so when working in collaboration with geoscientists, is that one must define what one would like to predict.

2.1.1 Identifying a Clear Question in Machine Learning Language

The goal of ***supervised learning*** is to create **predictions** of unknown conditions given measurements that are possibly correlated (in space, time, or type). When applying ***supervised learning*** tools to geoscience data, the following questions will be asked:

- What is being predicted?

- What are the input data?
- What will the **predictions** mean?

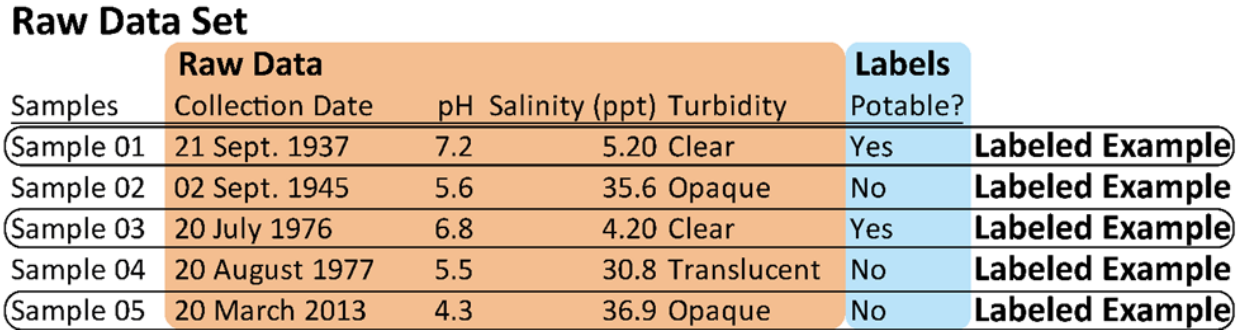
Perhaps the most confusing homonym for non-machine learning experts is the term **example**. An **example** is all of the information attached to a datapoint. For ***supervised learning***, this information consists of the input data that will be used to make **predictions** (*e.g.,* latitude, longitude, rock type, geological province), along with a measured value of the property that will be predicted (*e.g.,* heat flow). When data are being collected, each sample and its accompanying data are an **example**. ***Supervised learning*** **learns** from **examples**. More specifically, ***supervised learning*** **models learn** to predict by finding mathematical and statistical relationships between **examples'** independent variables (*i.e.,* the input data) and dependent variables (*i.e.,* **labels**). That is, ***supervised learning*** **models** use **labeled examples**, which are instances of a phenomenon replete with known values for the input data and **labels**, to find the mathematical and statistical relationships. **Labels** can be categorical or continuous. **Examples** without **labels** are called **unlabeled examples**. The **predictions** from ***supervised learning*** **models** are called ***response variables***.

Machine learning algorithms are the mathematical and statistical functions that can be **trained** into **models**, which associate the input data with the **labels** and, in doing so, **train** to become **models** and predict ***response variables*** from input data.

The data, as collected, are referred to as **raw data** and require processing before they are ready to **train** an ***algorithm*** to become a **model**. **Raw data** can be either categorical or continuous, and processing may depend on data type.

2.1.2 Identifying a Clear Question for the Hydrological Study

For our simple hydrological study, we ask 'Is the water potable?' (Fig. 3). Each sample in Figure 3 serves as a **labeled example**. The measured properties of each sample (*e.g.,* date, pH, salinity, and turbidity) are the **raw data**. The value for 'Potable?' (*i.e.,* the categorical variable Yes/No) is the **label**. An ***algorithm*** will **train** from **labeled examples** to become a model that **learns** how to predict ***response variables*** (*e.g.,* is the water potable?) for **unlabeled examples** (*e.g.,* new water samples for which we have **raw data**).



Raw Data Set

	Raw Data				**Labels**	
Samples	Collection Date	pH	Salinity (ppt)	Turbidity	Potable?	
Sample 01	21 Sept. 1937	7.2	5.20	Clear	Yes	**Labeled Example**
Sample 02	02 Sept. 1945	5.6	35.6	Opaque	No	**Labeled Example**
Sample 03	20 July 1976	6.8	4.20	Clear	Yes	**Labeled Example**
Sample 04	20 August 1977	5.5	30.8	Translucent	No	**Labeled Example**
Sample 05	20 March 2013	4.3	36.9	Opaque	No	**Labeled Example**

Figure 3. A visual depiction of the distinctions between 'labeled example', 'raw data', and 'labels' in a data set for a simple hydrological study. In this study, 'Potable?' indicates whether or not the water is safe to drink in a Yes/No format.

2.1.3 Identifying a Clear Question in the Geothermal Study

Mordensky et al. (2022) asked, how do the expert decision-dependent models from 2008 compare to purely data-driven ***machine learning*** **models** using the same **raw data**? In this geothermal-machine learning study, Mordensky et al. (2022) chose five sets of **raw data** (*i.e.,* heat flow, distance to quaternary faults, distance to magma bodies, maximum horizontal stress, and seismic-event density) for 700,000+ mostly **unlabeled examples** as cells in a 2-km-by-2-km grid across the western United States. Locations with confirmed geothermal systems were assigned positive **labels**, yielding 278 **labeled examples** for **training**. These **raw data** and **labels** were used with three ***machine learning algorithms*** to **train** ***machine learning*** **models** to predict ***response variables*** (*i.e.,* the probability of finding geothermally favorable conditions in a cell) for **unlabeled examples** for the western United States.

2.2 Explore Data

The collected data need to be inspected for their relationships and distributions. The correlation and statistics of the data are important considerations for selection of ***machine learning*** strategies and for transformation of the **raw data** into a format that will work with the selected ***algorithms***.

2.2.1 Exploring Data in Machine Learning Language

Exploratory data analysis (EDA) is the process by which one analyzes and investigates the primary characteristics of the **raw data** set (*e.g.,* statistics like mean and mode, data types, missing values). The key findings of the ***exploratory data analysis*** lend guidance toward how the **raw data** need to be processed and transformed so that they may be used by an ***algorithm*** to **train** to become a **model**.

It is common to use data visualization methods (*e.g.,* plots). Although the specifics steps for ***exploratory data analysis*** vary from institution to institution, or even individual to individual, the goal remains the same: to identify key statistics of the data (*e.g.,* Is there correlation between some data fields? Do some data express collinearity? Are some data values more common that others? Are the data normally distributed?). Data correlation is important, because input data that are highly correlated to each other have less unique information for **predictions**. Normally distributed data can lend confidence to **predictions** that are well represented by most of the data, but special care may be necessary when predicting extreme values.

2.2.2 Exploring Data in the Hydrological Example

Through examining the basic statistics of the hydrological study, we complete a simple ***exploratory data analysis***. For instance, 'Salinity' has greater minimum, maximum, mean, and standard deviation values than that of 'pH' in Figure 3. Similarly, the range of 'Salinity' is far greater than that of 'pH'. Also, 'Turbidity' and 'Potable?' are provided as categorical values. Date is provided in a Day-Month Name-Year format. Range of variables with different measurement units can be misleading (*e.g.,* perhaps a small difference in pH is more important than a large difference in salinity), so differences identified during ***exploratory data analysis*** can form the foundation of ***feature engineering*** (next step; Section 2.3).

2.2.3 Exploring Data in the Geothermal Study

In the 2008 geothermal resource assessment, ***exploratory data analysis*** found some **raw data** properties contained values spanning only a couple orders of magnitude (*e.g.,* heat flow [32 – 114 mW/m^2]). Other properties ranged six orders of magnitude (*i.e.,* distance to a quaternary fault [0 – 596,996 m] and distance to a magma body [0 – 362,580 m]). Similarly, depending upon the property in question, standard deviation ranged from single-digit values (*i.e.,* with seismic event density in *n*/km^2 and maximum horizontal stress in MPa) to greater than five-digit values (*i.e.,* distance to fault in m, distance to a magma body in m). Likewise, mean values also varied by several orders of magnitude between the properties.

2.3 Engineer Features

To those new to ***machine learning***, data manipulation may sound sinister in intent since all scientists are warned of a theme perhaps best articulated by Ronald C. Coase, "If you torture data long enough, it will confess [to anything]" (Good, 1972). Despite this important caution, collected data typically need preparation before the data are ready to **train *supervised learning* models**. ***Feature engineering*** can account for differences in measurement units (*e.g.,* distance in cm versus km) or for converting categorical variables into mathematical characterizations that may or may not have a natural rank order. For instance, if water content is an important input variable, then the categories saturated/damp/dry can be ranked from wetter to drier. But if mapped surficial geology is used as input, it may be more difficult to provide a meaningful rank (*e.g.,* Quaternary basalt/pre-Miocene basalt/sandstone/limestone).

2.3.1 Engineering Features in Machine Learning Language

Feature engineering is the process of using domain knowledge and an understanding of ***machine learning*** methods to select and transform the **raw data** into a format ideal for ***machine learning***. Once the data have been formatted, the **examples'** (*i.e.,* the samples') independent variables are called **features**. A ***feature vector*** is the numeric representation (*e.g.,* categorical variables are represented as numbers) of the combined **feature** values for an **example**. Writing each **example** as a ***feature vector*** prepares the data for numerical analyses. A **feature set** is the list of **feature** names (*e.g.,* column names in Fig. 4).

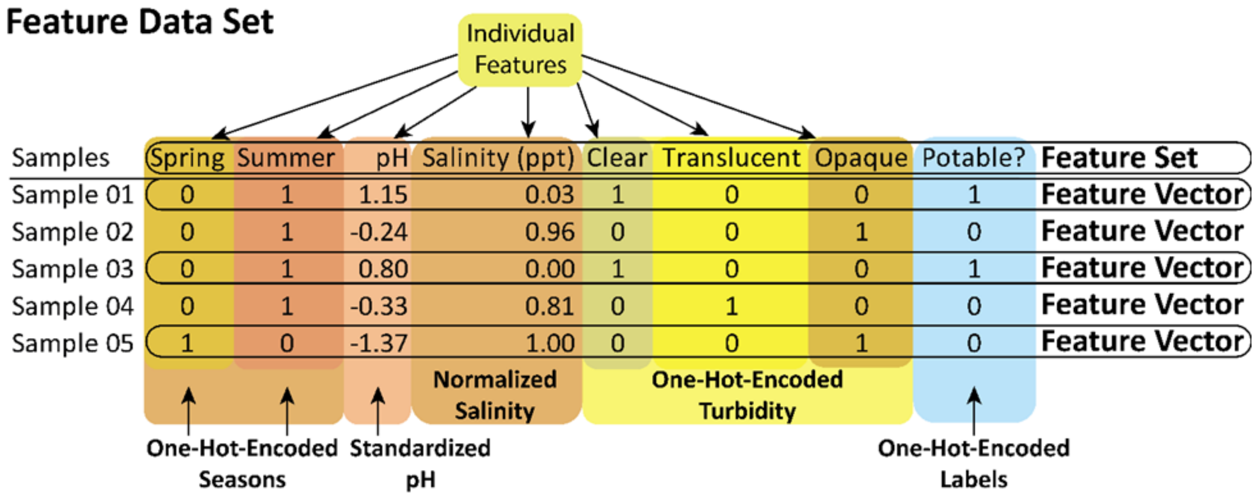
Continuous data are often either standardized or normalized so that the difference in scale between **features** does not impart unintended bias in the ***machine learning*** process. **Standardization** of continuous data is typically accomplished by subtracting the sample mean and dividing by the sample standard deviation, creating an engineered **feature** with a mean of 0 and a standard deviation of 1. ***Normalization*** is the process of scaling the allowable data range into a user-defined range defined by upper and lower limits (*e.g.,* 0 to 1). The choice of **standardization** or ***normalization*** is left to the ***machine learning*** expert based upon the qualities of the data, but the general goal is to make sure that the variability of each **feature** has a similar magnitude, so that machine learning ***algorithms*** can effectively examine correlations and contrasts between the **features**.

Categorical data commonly undergo a transformation known as ***one-hot-encoding***, which is the method of converting categorical data to a machine-readable format with no rank order. A categorical variable with *N* possible values is transformed into *N* **features** (*e.g.,* an **example** with three possible colors, red, blue, green, has N=3), with each **feature** having values of 0 or 1. In this case, the ‘red’ **feature** has a value of 1 only if the color is red, and 0 if it is any other color. The same process is used to create ‘blue’ and ‘green’ **features**.

Feature engineering also includes a wide range of transformations that optimize ***algorithm*** performance, and these transformations are often the result of the ingenuity of geoscientists or machine learning experts. Data may be transformed to isolate a signal representing a key process or characteristic, and strategies may be employed to isolate complex processes. For instance, given data sets A and B, where values in data set A are important only when the corresponding values of data set B are greater than a certain threshold, a new, third data set might be engineered to emphasize this behavior.

2.3.2 Engineering Features in the Hydrological Example

For the hydrological study, the ‘pH’ **feature** is standardized, and the ‘Salinity’ **feature** is normalized from 0 to 1 (Fig. 4). The three possible values of the single categorical variable ‘Turbidity’ are converted into three **features** using ***one-hot-encoding*** (*i.e.,* compare the values in Fig. 3 ‘Clear’, ‘Translucent’, and ‘Opaque’ to the **features** in Fig. 4). Similarly, the single column of sample dates (Fig. 3) are ***one-hot-encoded*** into multiple season **features** (Fig. 4). Finally, in addition to the input variables, the **labels** are also ***one-hot-encoded***, completing the transformation of the science question (Section 2.1.2) into a machine-readable format appropriate for ***machine learning***.



Samples	Spring	Summer	pH	Salinity (ppt)	Clear	Translucent	Opaque	Potable?	Feature Set
Sample 01	0	1	1.15	0.03	1	0	0	1	Feature Vector
Sample 02	0	1	-0.24	0.96	0	0	1	0	Feature Vector
Sample 03	0	1	0.80	0.00	1	0	0	1	Feature Vector
Sample 04	0	1	-0.33	0.81	0	1	0	0	Feature Vector
Sample 05	1	0	-1.37	1.00	0	0	1	0	Feature Vector

Figure 4. Engineered features using the raw data from Figure 3. pH has been standardized using a standard normal transformation. Salinity has been normalized. Season has been one-hot-encoded from the date. Turbidity has been one-hot-encoded. Labels have been converted from Yes and No to 1 and 0, respectively. Visual depiction of the distinctions between individual features, feature vectors, feature sets, examples, and labels are given.

2.3.3 Engineering Features in the Geothermal Study

Mordensky et al. (2022) standardized each of the five input’s **features**, so that every **feature** had a mean of 0 and a standard deviation of 1, effectively rendering the **features** unitless. The **labels** (*i.e.,* presence or absence of a geothermal system) were ***one-hot-encoded*** creating two **features**, a presence **feature** and an absence **feature**.

2.4 Choose an Algorithm

The ***machine learning*** expert chooses appropriate mathematical and statistical ***algorithms*** to **train** a **model** with consideration for the specific qualities of the phenomenon and the characteristics of the input data. Part of this process may entail testing several types of ***machine learning algorithms*** and strategies (*c.f.,* Mordensky et al., 2022). Use of multiple methods allows choosing ***algorithms*** that work best or allows an evaluation of confidence in **predictions** based on agreement (or disagreement) between methods (see Section 2.6 for evaluation methods).

2.4.1 Choosing an Algorithm in Machine Learning Language

There are many forms of ***machine learning algorithms*** with varying complexity ranging from ***shallow learning*** (*e.g.,* linear regression, logistic regression, decision trees, random forests, support vector machines) to ***deep learning*** in sequentially layered, neural networks (*e.g.,* multilayer perceptron neural networks, convolutional neural networks, recurrent neural networks). Every ***algorithm*** has its strengths and weaknesses that need to be considered in the context of the available data and the research question(s). For instance, tree-based classifiers provide **predictions** that are highly interpretable as behavior thresholds (*e.g.,* plant growth as a function of exceeding climate thresholds such as available sunlight or available water), so that the resulting decision process is easy for scientists to understand; whereas, interpreting the decision processes that led to a **prediction** from a neural network is not nearly as simple nor straightforward, especially when input variables are combined in many complicated mathematical operations.

Some ***algorithms*** (*e.g.,* support vector machines) directly predict a **classification** value, and some ***algorithms*** (*e.g.,* logistic regression) predict a probability of each possible **classification**. For instance, geoscience is replete with **classification labels** (*e.g.,* Yes/No and 1/0 for 'Potable?' in Figs. 3 and 4, respectively), and those classification values have clear meanings (*e.g.,* potability). In this sense, geoscientists are easily able to make use of classification values produced by **classification** ***machine learning*** methods. Other ***algorithms*** predict probability values for each **example** (*e.g.,* logistic regression). Then, given a chosen probability threshold (*e.g.,* a probability threshold = 0.5), a **classification label** is applied depending on whether the **example's** probability is greater than or less than the threshold probability. Understanding how probability values from ***machine learning algorithms*** are derived and knowing what factors lead to the development of those probability values is important for ensuring that the predicted probability values are properly interpreted, and their meaning understood.

Algorithms can also be described as ***linear*** or ***non-linear***. In ***linear algorithms***, the **feature** values of an **example** contribute proportionally toward a **prediction** value (*e.g.,* linear regression); whereas ***non-linear algorithms*** place particular emphasis on some **feature** values as being more important than other **feature** values in the decision-making process (*e.g.,* decision trees).

With the above considerations in mind, ***algorithm selection*** is a critically important decision in ***machine learning***. Specifically, ***algorithm selection*** is the method of selecting amongst the ***machine learning algorithms*** with consideration for their specific strengths, their specific weaknesses, the data available, and the requirements implicit to the research question (see generally Chapter 5 of Burkov [2019]). Understanding how the ***algorithm*** works helps with proper selection, and also helps prevent the misapplication of the method. Each method has implicit mathematical assumptions and restrictions that require verification and adherence.

2.4.2 Choosing an Algorithm for the Hydrological Example

While no analyses are performed in this manuscript for either described studies, we recommend simple linear **models** first, and moving towards more complex **models** incrementally. The sequential addition of complexity allows the science team to better understand correlations and possibly even causation. Simple **models** also tend to work better with fewer data (*e.g.,* Fig. 4) and to be very fast, allowing rapid data exploration of data characteristics and **predictions**. Adding **model** complexity incrementally (*e.g.,* moving from a linear to a non-linear **model**) allows the research team to understand basic characteristics of the data (*e.g.,* if **model** performance improves with a non-linear **model**, the process may intrinsically be non-linear). For the hydrological study, logistic regression (Fig. 5) would be a good choice of a robust linear **model** for initial data analyses. If results are sufficient for study purposes, then the study is complete. If desired, a next step might be to employ a non-linear method such as XGBoost (Chen and Guestrin, 2016) to see if performance improves. This process of selecting more complicated (but generally harder to interpret) methods can continue until a **model** is deemed sufficient for a defined purpose (*e.g.,* predicting potable water with sufficiently high accuracy), or until it becomes apparent that more complex **models** do not provide improved **predictions**.

2.4.3 Choosing an Algorithm in the Geothermal Study

The three ***algorithms*** chosen during ***algorithm selection*** in Mordensky et al. (2022) are forms of ***shallow learning***. These three ***algorithms*** (*i.e.,* logistic regression, support vector machines, and XGBoost) were selected primarily for two reasons: 1) to compare the ***machine learning*** approaches to the strategies of the 2008 expert decision-dependent assessment; and 2) to compare the performance of ***algorithms*** that function differently from one another (*e.g.,* linear versus non-linear). Mordensky et al. (2022) used this diverse suite of ***shallow learning algorithms*** to establish a foundational understanding of how ***machine learning algorithms*** treat geothermal data.

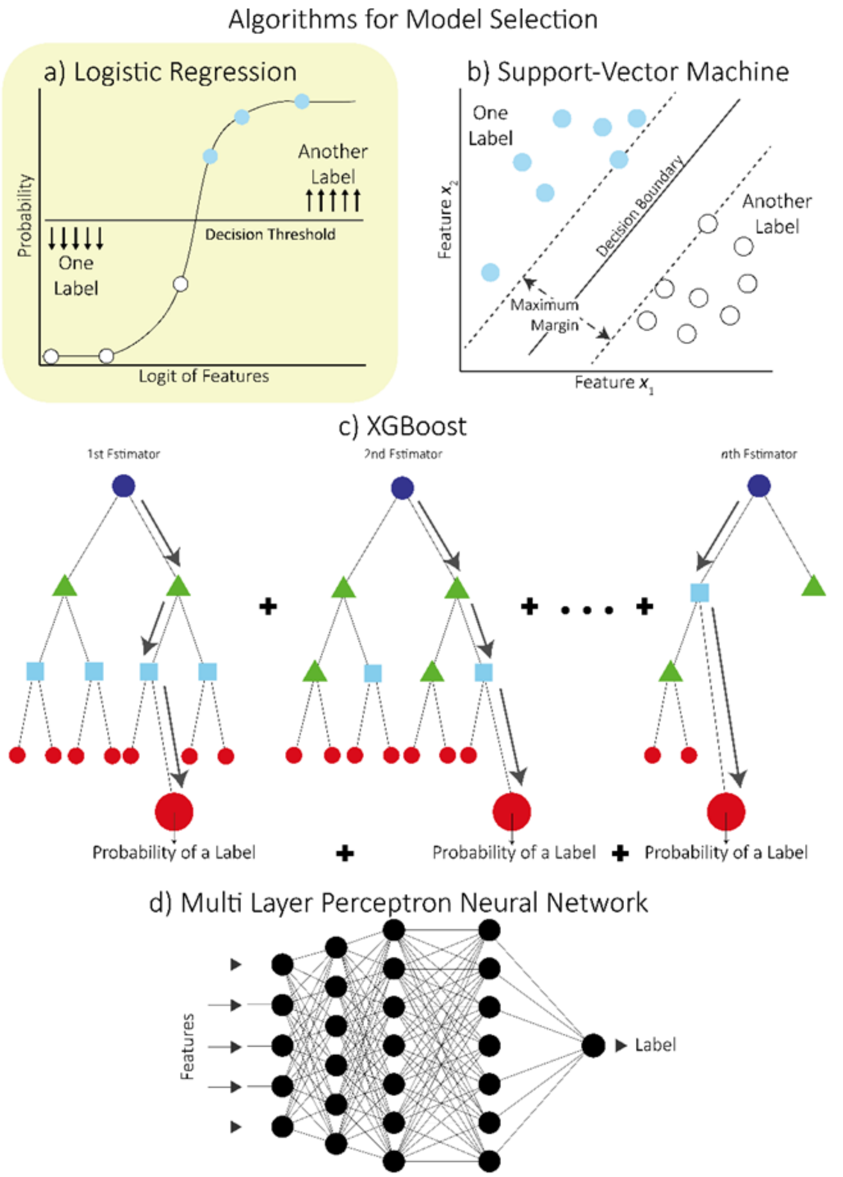


Figure 5. Illustrated conceptual frameworks for a non-exhaustive assortment of common machine learning algorithms (*i.e.,* [a] logistic regression, [b] a support vector machine, [c] XGBoost, and [d] a multilayer perceptron neural network). See Berkson (1944) for more information on logistic regression. See Cortes and Vapnik (1995) for more information on support vector machines. See Chen and Guestrin (2016) for more information on XGBoost. See Chollet (2021) for more information on neural networks. The conceptual model figures for logistic regression, support vector machines, and XGBoost are modified from Mordensky et al. (2022).

2.5 Train-Test Split the Data

A large minority of the data (*e.g.,* 5 – 20 %; see generally Chapter 5 of Burkov [2019]) needs to be set aside and not used to **train** the **model**. The remaining 80 – 95 % of the data are used to **train** the **model**. The data that were set aside will be used to test the **model** once it is **trained**. Testing ensures that the newly **trained model** predicts well from new data not seen during **training**.

2.5.1 Train-Test Splitting Data in Machine Learning Language

Supervised learning **models** require **examples** from which to **train** and other **examples** to test how well the **model** has **learned**. The training **examples** provide the 'supervision' in ***supervised learning***. However, the **models** produced from this training need to be evaluated to determine the quality of their performance. Put simply, the ***machine learning*** expert needs to gauge if the **model** performs well enough to be used to predict ***response variables*** for **unlabeled examples**. This evaluation of the **model** requires its own data separate from the data used for training. Hence, the initially complete **feature** data set needs to be split. A ***train-test split*** refers to randomly dividing the **feature** data set into ***training data*** and **testing data** (Fig. 6). The ***training data*** are used to **train** the **model**. The **testing data** are used to evaluate the performance of that **model**.

2.5.2 Train-Test Splitting Data in the Hydrological Example

At this step, the low number of samples given as the hydrological data proves to be a detriment for its use to **train** a ***machine learning*** **model** because there are too few **examples**. Although we are able to conduct a 4:1 split of the **feature** data, only one of the five **labeled examples** would then be used as the **testing data**. While there is no exact number of **examples** needed to conduct supervised ***machine learning***, both the **testing data** and ***training data*** should contain many **examples**. Additional consideration needs to be given to the complexity of the phenomenon being modeled and the complexity of the selected ***algorithm*** (*e.g.,* ***linear*** versus ***non-linear***). As the complexity of the phenomenon being modeled and/or the complexity of the selected ***algorithm*** increase(s), ***machine learning*** requires more **examples**. Unfortunately, a consistently reoccurring challenge faced by geoscientists is having too few **examples**; however, there are strategies to address this issue (*e.g.,* creating synthetic **examples** with the same statistical properties of the collected **examples** to supplement the collected **examples**).

2.5.3 Train-Test Splitting Data in the Geothermal Study

In Mordensky et al. (2022), the **feature** data were subject to a 4:1 ***train-test split***, which corresponds to 80% of the data becoming ***training data*** and 20% of the data becoming **testing data**.

2.6 Optimize, Train, and Evaluate the Model

All ***machine learning algorithms*** have parameters that are fit with the data, but many ***algorithms*** also have parameters that can only be manually tuned to improve **model** performance. By changing the values of these manually adjusted parameters, the performance of the resultant **models** also changes. After the best values for these parameters are identified, a final **model** can be produced and evaluated.

2.6.1 Optimizing, Training, and Evaluating the Model in Machine Learning Language

Every **model** is the product of a combination of an ***algorithm*** (*e.g.,* Fig. 5) and a **feature** data set. Many of the ***algorithm's*** parameters within the **model** are adjusted as the **model learns** from the data. However, many ***algorithms*** also have parameters that do not **learn** from the data and must be manually tuned to improve **model** performance. These parameters, called ***hyperparameters,*** are (usually, but not always) numerical values (*e.g.,* class weight, inverse regularization strength; see generally Pedregosa et al., 2011 for details) that must be set by the researcher before **training** the

model. ***Hyperparameter optimization*** (also termed **optimization** and ***hyperparameter*** tuning) refers to the exploration and selection of ***hyperparameter*** values that produce the best performing **model**. After the optimal values for the ***hyperparameters*** are identified, a final **model** is **trained** and evaluated.

Hyperparameter optimization is completed through a technique known as ***validation***. ***Validation*** begins by splitting the ***training data*** yet another time so that a sub-group (or several sub-groups) of ***validation data*** are set aside from the rest of the ***training data*** (Fig. 6). A **model** is **trained** with the remaining ***training data*** (Fig. 6). The ***validation data*** are then used by the **model** to find the ***validation error*** (also called ***validation loss***). ***Loss*** is a measure of error, as defined by a ***loss function***, between an **example's label** and its associated ***response variable*** from the **model**. A ***cost function*** provides the average value of ***loss*** over all the examined **examples**. Common ***loss functions*** are mean squared error and logistic ***loss***. ***Regularizers*** are mathematical terms that can be added to the ***loss function*** to achieve some desired behavior (*e.g.,* avoid violating physical principles, honor physical processes, limit the number of non-unique solutions) but are not required. ***Validation*** explores how different ***hyperparameter*** values influence the ***validation error***. The ***hyperparameter*** values that contribute to the lowest ***validation error*** (*i.e.,* ***average loss***) are said to be optimal. If the ***average loss*** is not minimized during ***validation***, the researcher adjusts the ***hyperparameter*** values being explored and ***hyperparameter optimization*** is started again. This process is repeated until ***average loss*** is minimized with the ***validation data*** and the optimal ***hyperparameter*** values are identified.

Multiple sets of ***training data*** and ***validation data*** allow for a means to assess a **model's variance** (*i.e.,* changes in the **model's** ***response variables***) when different data subsets are used to **train** the **model**.

Some ***algorithms*** have only a couple ***hyperparameters*** that have a major impact on performance. Other ***algorithms*** have several ***hyperparameters***, and the proper choice of these ***hyperparameters*** is essential for strong **model** performance. ***Hyperparameter*** exploration can be a computationally intensive task. Depending on the time available, the number of ***hyperparameters*** involved with an ***algorithm*** may be a consideration in ***algorithm selection***.

Once the ***average loss*** is minimized on the ***validation data***, the ***hyperparameter optimization*** is complete. A final **model** is then **trained** using all ***training data*** and the optimal ***hyperparameter*** values (Fig. 6). The **model** is then ready to predict using the **testing data**.

The ***response variables*** predicted from the **testing data** by the **model** are compared with the researcher-assigned **labels** to assess the **testing error** (*i.e.,* a measure of error between the **labels** and ***response variables*** of the **testing data**). Similarly, when the **trained model** is used to predict ***response variables*** for the ***training data***, one can examine the ***training error*** **(***i.e.,* a measure of error between the **labels** and ***response variables*** of the ***training data***). If the ***training error*** is low and **testing error** high, the **model** may be ***overfit*** to the ***training data***. ***Overfitting*** occurs when a **model** perfectly (or nearly perfectly) fits the ***training data*** but does not predict well from new data (*e.g.,* test data or new samples). Reciprocally, ***underfitting*** occurs when a **model** is some combination of not complex enough or has not **trained** from enough **examples** to predict the ***training data*** or **testing data** well, leading to high ***training error*** and high **testing error**.

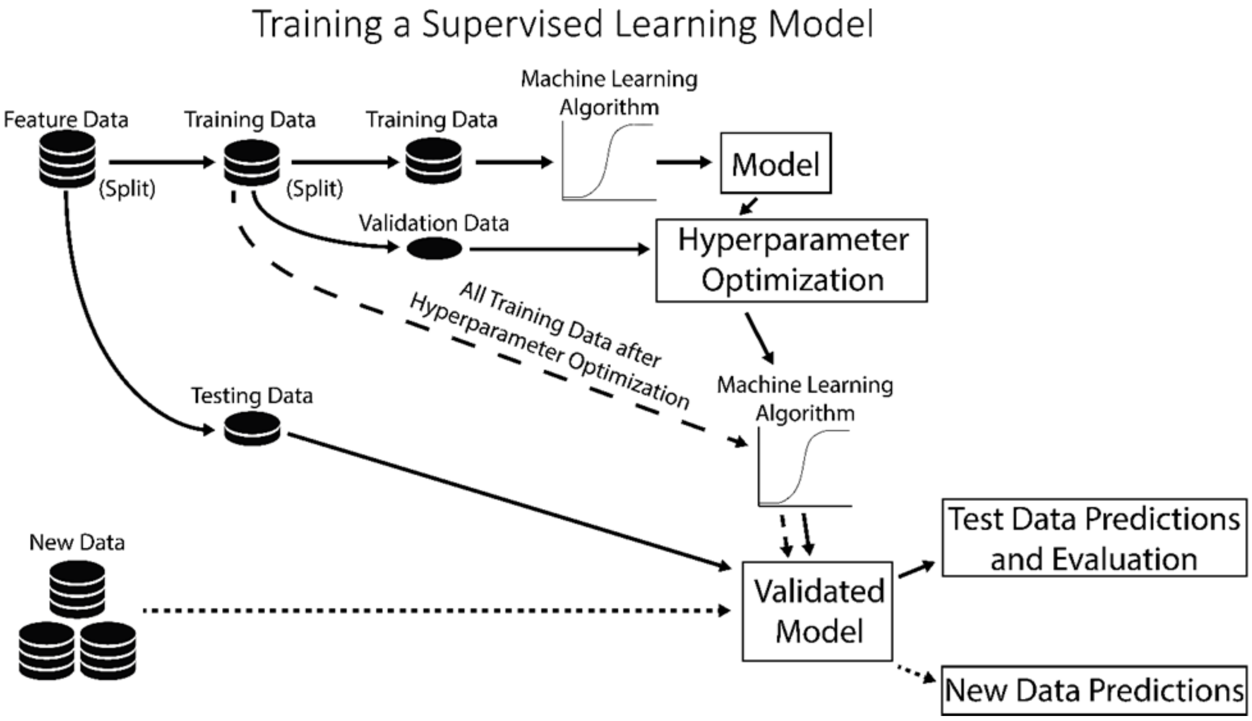

Training a Supervised Learning Model
Feature Data
(Split)
Training Data
(Split)
Training Data
Machine Learning Algorithm
Model
Validation Data
Hyperparameter Optimization
All Training Data after Hyperparameter Optimization
Machine Learning Algorithm
Testing Data
New Data
Validated Model
Test Data Predictions and Evaluation
New Data Predictions

Figure 6. Workflow for optimizing, training, evaluating, and predicting from a supervised learning model. Testing data are used to infer how the validated model will perform when new data are processed.

2.6.2 Optimizing, Training, and Evaluating a Model from the Hydrological Study

Presuming a researcher had only four **examples** in their ***training data***, like with the simple hydrological **feature** data, one **example** would need to be used for ***validation***, leaving only three **examples** for **training**. Under these circumstances, it is highly likely that the resulting **model** would be ***underfit***. Many more **examples** would be needed during **training** to produce a well-performing **model**.

2.6.3 Optimizing, Training, and Evaluating the Models from the Geothermal Study

During ***validation***, Mordensky et al. (2022) used the default ***loss functions*** in Python's Scikit-Learn and XGBoost modules for each of the selected ***algorithms*** to gauge the performance of the thousands of **models** resulting from the thousands of unique combinations of ***hyperparameter*** values (Pedregosa et al., 2011; Chen and Guestrin, 2016). The ***validation data*** were used to calculate ***loss*** values through the ***loss functions***. The ***hyperparameter*** values that contributed to the **model** that produced the lowest ***average loss*** from the ***validation data*** were then used with all the ***training data*** to produce a final **model** (*i.e.,* the proposed best **model** for **prediction**). The final **model** was then evaluated using the **testing data** to verify that the **model** performs equally well for new data.

2.7 Make Predictions

Once the **model** is **trained** and performs well during evaluation, the **model** can be used to make **predictions** using new data.

2.7.1 Making Predictions in Machine Learning Language

Following **training**, ***validation***, and evaluation, the ***supervised learning*** **model** is ready to predict for new, **unlabeled examples**. When discussing ***machine learning*** **models**, the term generalize is sometimes used in lieu of predict and **generalization** in lieu of **prediction**. When generalizing with the **model**, the **testing error** provides a metric of reliability regarding the new **generalizations** of the **model** from **unlabeled examples**. However, should that **model** be used with new, **unlabeled examples** that have values well beyond the ranges of the **feature** values in the ***training data*** (*i.e.,* new data are very different from previous data), the confidence in that **testing error** lessens.

2.7.2 Making Predictions with the Hydrological Example

With the hydrological data, the limited number of the **examples** used in **training** would likely mean that the **model** would not be properly **trained** to generalize from the full range of values that would be found with newly collected data; hence, the **model** would be ***underfit***. This can be seen by the fact that for the sample data set, every time the water is 'clear', it is potable, and any lack of clarity indicates that water is not potable. This means that a very accurate **model** for the hydrological study data would use only the turbidity to predict potability, but seawater is very clear in some areas; yet seawater is not potable. In this case, more **labeled examples** would be needed for **training** to produce an accurate **model** for a wide range of natural waters.

2.7.3 Making Predictions with the Geothermal Study

Many ***machine learning*** studies only use one ***machine learning algorithm*** to **train** and evaluate a **model**. Mordensky et al. (2022) used three ***algorithms*** and, in doing so, were able to compare the different **models** from the different ***algorithms*** trained using the same data. Each **model** was used to predict geothermal favorability across the western United States (see favorability maps in Mordensky et al. [2022]). The favorability maps that were produced by the different ***algorithms*** generally agreed in terms of areas of high geothermal favorability but expressed greater variability in **predictions** between **models** in areas of low favorability. If the goal of study is to find high-favorability areas, the differences between the **models** may not be critical, and the general agreement between the **models** adds confidence in the **predictions**. The **models** trained from ***non-linear algorithms*** (*i.e.,* support vector machines and XGBoost) predicted greater geospatial granularity than that of the ***linear algorithm*** (*i.e.,* logistic regression).

3. Additional Homonyms

While many of the key terms needed to discuss ***machine learning*** are novel to a geoscientist (Table 1), many other key terms do not appear as new but, instead, are homonyms (Table 2). Here, we present prominent homonyms shared by geoscience and ***machine learning*** that were not discussed in the conceptual **pipeline** of Section 2.

3.1 Survey

To a geoscientist, 'a survey' refers to the systematic investigation of the geology beneath a specific area. The word can also be used as a verb for the conduction of that systematic investigation. A geoscientist may also refer to a state's geological survey or the U.S. Geological Survey as 'the Survey'. To a ***machine learning*** expert, a **survey** is akin to what a geoscientist knows as a literature review paper. That is, a **survey** paper in ***machine learning*** focuses on summarizing the findings from several works with a unifying theme to synthesize additional understanding of a topic.

3.2 Risk

Geological studies involving risk define risk as a function of the likelihood of occurrence for a given hazard (*e.g.,* a landslide, a volcanic eruption, an earthquake) and the damage that hazard would produce. In ***machine learning***, **risk** refers to a measure of the possibility that a ***machine learning*** process will produce a **model** that makes less reliable **predictions** with new data. In ***machine learning***, **risk** can be measured by comparing ***loss***.

3.3 Epoch

In geoscience, an epoch refers to a (reasonably short) length of time on the scale of (only) tens of millions of years. In geologic time, epochs are used to subdivide the next longer segment of time (*i.e.,* periods). For instance, the 'Jurassic' in 'Late Jurassic' would be a period and the 'late' in 'Late Jurassic' would define the epoch. In ***machine learning***, some ***algorithms*** pass through the data several times during training (*e.g.,* multilayer perceptron neural networks). Each of these passes of the entire training data is referred to as an **epoch**. Hundreds or thousands of **epochs** may be used to fully train a **model**.

3.4 Entropy

In science, the common definition of entropy refers to the measure of molecular disorder addressed in thermodynamics; although, the term is also used more generally as a qualitative reference to disorder. While not starkly different, the nuanced derivation of ***machine learning*'s entropy** is a mathematical measure of randomness, but that measure is not necessarily of a subatomic nature. **Examples' features** inherently contain some degree of uncertainty with their values. The less constrained those **feature** values are, the more **entropy** (*i.e.,* the greater uncertainty) they hold.

4. Conclusion

In this study, a general workflow (*i.e.,* a conceptual pipeline) for supervised learning is presented. In that process, key machine learning terms that may be new to geoscientists are defined and their context explained. Homonyms between the disciplines are identified and defined. Lastly, common homonyms not presented in the general workflow are presented and briefly discussed. We emphasize that the terms provided here are non-exhaustive. The intent of this work is to produce an initial resource to which geoscientists and machine learning experts may refer when working together.

Acknowledgements

This work was supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE), Geothermal Technologies Office (GTO) under Contract No. DEAC02-05CH11231 with Lawrence Berkeley National Laboratory, Conformed Federal Order No. 7520443 between Lawrence Berkeley National Laboratory and the U.S. Geological Survey (Award Number DE-EE0008105), and Standard Research Subcontract No. 7572843 between

Lawrence Berkeley National Laboratory and Portland State University. Support for Cary Lindsey was provided by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Geothermal Technologies Office, under Award Number DE-EE0008762. Additional support for John Lipor was provided by the National Science Foundation awards NSF CRII CIF-1850404 and NSF CAREER CIF-2046175. Support for Erick Burns was provided by the U.S. Geological Survey Energy Resources Program. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

RECOMMENDED MACHINE LEARNING TEXTS

Burkov, A. (2019). The Hundred-Page Machine Learning Book: Andriy Burkov.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). The Elements of Statistical Learning (2nd ed.). New York, New York: Springer.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. Cambridge, Massachusetts: The MIT Press.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge Massachusetts: Cambridge University Press.

REFERENCES

Berkson, J. (1944). Application of the Logistic Function to Bio-Assay. *Journal of the American Statistical Association, 39*, 357-365.

Burkov, A. (2019). *The Hundred-Page Machine Learning Book*: Andriy Burkov.

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System.* Paper presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA.

Chollet, F. (2021). Chapter 1. What is deep learning? In *Deep Learning with Python* (2nd ed.): Simon and Schuster.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*, 273-297. doi:doi.org/10.1007/BF00994018

Council of Europe. (2019). Retrieved from www.coe.int

Good, I. J. (1972). Statistics and Today's Problems. *The American Statistician, 26*(3), 11-19. doi:10.2307/2682859

Mordensky, S. P., Lipor, J. J., DeAngelo, J., Burns, E. R., & Lindsey, C. R. (2022). *Predicting Geothermal Favorability in the Western United States by Using Machine Learning: Addressing Challenges and Developing Solutions*. Paper presented at the 47th Stanford Geothermal Workshop, Stanford, California (Virtual).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . others. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*, 2825-2830.

Williams, C. F., & DeAngelo, J. (2008). Mapping Geothermal Potential in the Western United States. *GRC Transactions, 32*, 181-188.

Williams, C. F., Reed, M. J., DeAngelo, J., & Galanis, S. P. (2009). Quantifying the undiscovered geothermal resources of the United States. *Transactions, 33*, 882-889.

Williams, C. F., Reed, M. J., Mariner, R. H., DeAngelo, J., & Galanis, S. P. (2008). Assessment of Moderate-and High-Temperature Geothermal Resources of the United States. *U.S. Geological Survey Fact Sheet 2008-3082*, 1-4.

Table 1. Novel Key Machine Learning Terms

Term	Definition & Implications	What a Geoscientist Should Think
algorithm	The mathematical or statistical function(s) that map(s) inputs to outputs. Also known as hypothesis, concept, predictor, or prediction rule.	The underlying idea or structure technique used to build a model. An untrained model. Examples include linear regression, decision trees, and neural networks.
algorithm selection	A method of selecting among algorithms or algorithm parameters to avoid overfitting.	The process to consider which algorithm and its parameters are most appropriate when considering the qualities of a dataset.
average loss	Loss averaged by all the examined examples. Given by the cost function.	Average of the loss function for examined samples.
cost function	Provides average of loss across all the examined examples (see Table 2 for 'example').	Gives the average of the error (*i.e.,* loss) function. The function that is minimized during optimization. Minimizing average loss (*i.e.,* error) generally improves the performance of a machine learning model. There may be multiple terms encapsulated in the cost function to achieve multiple goals (*e.g.,* minimize error, find a solution that honors physical constraints, find a solution that minimizes uncertainty). Also called the objective or objective function, although a cost function is but one type of objective function.
deep learning	A form of machine learning with algorithms based on numerous sequential layers in a neural network.	Highly flexible machine learning algorithms that perform well at the cost of interpretability (*i.e.,* a black-box problem). Deep learning can result in models with low training error but are prone to overfitting.
exploratory data analysis	The process analyzing and investigating the primary characteristics of the data set. Often referred to as EDA.	Examining the distribution, types, and relationships of data.
feature engineering	The process of using domain knowledge (*i.e.,* knowledge of the discipline) to select and transform the most relevant variables from raw data so that machine learning may better use the data.	Processing data into a format usable by machine learning algorithms.

feature vector	The combined values for an example’s features (see Table 2 for 'example').	The combined values for the different data fields tied to an observation. Example: the feature vector for a rock sample 01 with a feature set of (age, density, and rock type) is (6.7 Ma, 3.4 g/cm^3, igneous).
hyperparameter	A property of an algorithm, usually (but not always) having a numerical value. This value influences the way the model works and is not learned from the data. Instead, it is set by the data analyst before training the algorithm.	Hyperparameters are variables that control how a model learns but cannot be learned from the data. One must explore and select hyperparameters when training new models.
hyperparameter optimization	The process of finding a set of algorithm parameters that results in the best performance according to the chosen metric. That is, the process of iteratively training a model that results in the best performance through the adjustment hyperparameters. Also termed hyperparameter tuning (see 'optimization' in Table 2).	The process of identifying and selecting hyperparameter values that produce the best performing model with a given algorithm.
linear algorithm	An algorithm in which the feature values of an example are linearly combined to produce a label value (see 'example' and 'label' in Table 2).	An algorithm in which features' values contribute proportionally toward a prediction.
loss	A measure of prediction performance on a single example. A measure of error at a single training site (commonly, some measure of the difference between the prediction and the training data at that location). A common example would be the squared error.	The error between a prediction and its associated label.
loss function	Computes the error between the response variable and the expected label (*i.e.,* an error function).	How the error between a prediction and a label (see Table 2 for 'label') is calculated. An example of loss would be the squared error (*i.e.,* quadratic loss) between a prediction and a known value.
machine learning	The field of study that explores the construction and study of mathematical and statistical models that learn without direct instruction.	An entire field of study with many sub-disciplines but the unifying component of these disciplines is that models are learning from data.

non-linear algorithm	An algorithm that produces a label value by combining features using a mathematically non-linear function (*e.g.,* a decision tree).	Algorithms that can form more complicated prediction functions to distinguish between examples.
normalization	The process of scaling data into a pre-selected range (*e.g.,* commonly 0 to 1) or simply transforming data onto the unit sphere.	A type of transformation in feature engineering that allows the different datasets to share the same scale.
one-hot-encoding	A means to quantify categorical data.	Every categorical value is converted to a feature. These new features have a value of 0 except where the categorical value for that example and the feature match; this feature's value is set to 1. For example, if we were using flower color as a category with [red, green, yellow, blue] as potential colors and the flower in question was green, the corresponding feature values would be [0,1,0,0].
overfitting	Training a predictor that achieves low training error but has high variance.	When an algorithm creates a model that matches the training dataset very well, too well (*i.e.,* low training error), but does not predict well from data not used during training.
regularizer	A term added to the objective function to achieve some desired behavior.	Regularizers are additional terms in the objective function to improve the machine learning algorithm. Some regularizers ensure rapid convergence to an answer *(i.e.,* mathematical techniques to speed up optimization). Some prevent overfitting. Some ensure convergence to answers that honor likely physical conditions (*e.g.,* temperature varies smoothly in space, heat flow is likely similar to the regional average) and avoid violating physical principles.
reinforcement learning	Decision making over time with consequences dependent on external, possibly delayed feedback.	A domain of machine learning that considers stimuli (*e.g.,* previous decisions, conditions, events) to make decisions that either lead toward a reward (for a correct decision) or a punishment (for an incorrect decision).

response variable	The variable that corresponds to a prediction that is made by a machine learning-derived model. It is a subtle distinction, but the machine learning scientist frequently uses the term "prediction" when discussing the value of the response variable for one set of input variables, and "response variable" when discussing all possible predictions that are made from all possible input data combinations.	What a model predicts.
shallow learning	Learning that does not involve multiple layers of a neural network.	Algorithms apart from neural networks, including linear regression, logistic regression, support vector machines, and decision trees. Neural networks classify as shallow learning if they have only one hidden layer.
supervised learning	A domain of machine learning in which the algorithm is given input-output pairs to learn from, so that predictions can be made with new data.	A machine learning approach that uses labeled data sets to predict values for unlabeled datasets (see Table 2 for 'label').
training data	Set of input-output data used for supervised learning.	The examples used by a model to learn the relationship between inputs and outputs. Bad training data will result in a bad model.
training error	A measure of error between the training data and the prediction made using the model with the training data. Empirical estimate of risk over the training dataset.	A measure of a model's performance with the training data.
train-test split	Partitioning training data from testing data for supervised machine learning.	The feature data are divided so that a large percentage of the data are used for training and a smaller percentage are used to evaluate (*i.e.,* to test) the trained model.
underfitting	Training a predictor that fails to predict well with the training data and the testing data. Producing an undertrained model.	Training a model that needs more complexity (*e.g.,* more structure or more training examples) to produce reliable predictions.
unsupervised learning	Machine learning when prediction labels are not provided. Finding relationships between input data and then grouping based upon these relationships.	A machine learning approach that groups data in different ways or simplifies data by finding internal relationships.

validation	A technique to estimate the model's ability to predict on unseen data (*i.e.,* data outside the training data set).	An evaluation of the model while tuning hyperparameters.
validation data	Set of input-output data used to evaluate a model for validation (*i.e.,* for hyperparameter optimization).	Validation data are used to measure the error of a model to tune hyperparameters during validation.
validation error	The measure of error between a model's predictions for the validation data and those data labels.	Optimal hyperparameters are chosen when hyperparameter values minimize the validation error.

Table 2. Geoscience-Machine Learning Homonyms

Term	What a Machine Learning Scientist Thinks	What a Geoscientist May Think
class	One of a set of finite target values for a label.	A category or subdivision with very specific definitions in some fields such as mineralogy and petrology.
classification	A supervised learning approach to predict and apply class labels to examples.	Generally, the grouping of similar objects within a system.
entropy	Mathematical construct of disorder; entropy of a random variable is the average level of “information”, “surprise”, or “uncertainty” inherent in the variable’s possible outcomes. That is, the more certain or the more deterministic an event is, the less information it will contain. In a nutshell, the information is an increase in uncertainty or entropy.	Most commonly refers to the thermodynamics definition of a measure of unavailable energy which is a function of molecular disorder, also used as a general measure of disorder or uncertainty.
epoch	A full pass of all the training data during an optimization procedure.	The geological time period when a rock was deposited. Geological eras are composed of geological periods.
example	From a dataset, a single datum replete with values for the features of the feature set. That is, one row of a dataset, containing one or more features and possibly a label.	An instance generally representative of a larger population.
feature	An independent attribute or variable for an example used to make predictions.	Any aspect of a system (*e.g.,* distance from a fault), characteristic, or structure of a rock.
feature set	The features that lead toward a prediction, specifically the field titles for the attribute data (*e.g.,* age, weight, height, and blood pressure might be considered a feature set for predicting an individual's health).	A collection of geological observations, sometimes qualitative (*e.g.,* structures).
generalization	The ability of a model to predict on unseen (*i.e.,* general) data. Similarly, generalize is sometimes used in place of predict.	Often the lay definition of broad statement is meant, but generalization is also used in cartography and geographic information systems to refer to methods used to limit the symbology needed or shown on map products.
label	The true condition (*i.e.,* dependent variable) of what is trying to be predicted by supervised machine learning.	A means of specimen identification or classification term applied to samples or data.
labeled example	Examples with corresponding labels (see 'examples' and 'labels' for additional information).	Representative specimens with identification indices or tags.
(to) learn	To create and tune decision functions using mathematical and statistical rules.	To gain or acquire knowledge by study, experience, or being taught.

model	What is produced by training an algorithm and can then be used to make predictions. A combination of decision functions composed of procedures dependent upon the algorithm chosen during model selection and specific values learned during training.	A spatial, conceptual, or mathematical representation of a phenomenon.
optimization	The process of finding a set of inputs to an objective function (*e.g.,* a loss function) that results in a maximum or minimum function evaluation.	Improving efficiency (*e.g.,* streamlining).
pipeline	A workflow with discrete steps for a complete machine learning task.	A pipe for conveying fluid or gas.
predictions	The output of a machine learning model.	An interpretation, forecast, or prognosis dependent on previous data but not necessarily dependent on a mathematical or statistical model.
raw data	Untouched data before engineering.	Data directly from measurement with no modifications.
regression	A numerical value estimate produced from a trained model.	Marine regression is a geological process when areas of submerged seafloor become exposed during changes in sea level.
risk	A measure of the possibility that the machine learning process will produce a model that makes less reliable predictions with new data. Risk is estimated by validation.	A combination of hazard, value, and vulnerability.
standardization	Data are transformed to a mean of 0 and a standard deviation of 1.	Adhering to specific methods and units of measurement to follow industry standards.
survey	A literature review of machine learning studies.	A systematic investigation of the geology beneath a specific area or a government institution.
testing data	Same as training data but are not part of the data used to train the algorithm. Used to assess performance of the model with data yet unseen by the model.	Experimental results.
testing error	A measure of the error between the testing data and the predictions made using the model with the testing data. That is, the error when a trained model is used to predict results on data from which the model has not been trained. The less the testing error, the better. If testing error is similar to validation error, this is evidence that the algorithm produced a reliable predictor.	Uncertainty or variability introduced into data during lab experiments, analyses, or other "tests".

train	Adjust (*i.e.,* change) the weights (*i.e.,* parameters) of an algorithm using label examples to reduce loss and risk.	Some form of personnel education.
unlabeled examples	Examples without labels (see 'examples' and 'labels' for additional information).	Representative specimens without a proper identifier.
variance	In addition to the statistical definition (see What a Geoscientist May Think), changes in the model when using different portions of the training data set; simply, variance is the variability in the model prediction. The degree of overfitting or underfitting.	A statistical definition providing a measure of how far data are distributed about the mean.