DATA-MD

Data Augmented Technology Assisted Medical Decision Making (DATA-MD)

Lecture Notes and Study Guide

Module 1
Introduction to Artificial Intelligence (AI) and Machine Learning (ML) in Health Care

Learning Objectives:

After completing this module, you will be able to:

- Recognize the role that AI/ML will play in evidence-based, data-driven medical decision making.
- Explain the potential strengths and limitations of AI/ML in health care.
- Describe current applications of AI/ML in health care.
- Define key terms, concepts, and principles necessary for effective communication with other ML stakeholders in health care.

Lesson 1: Introduction

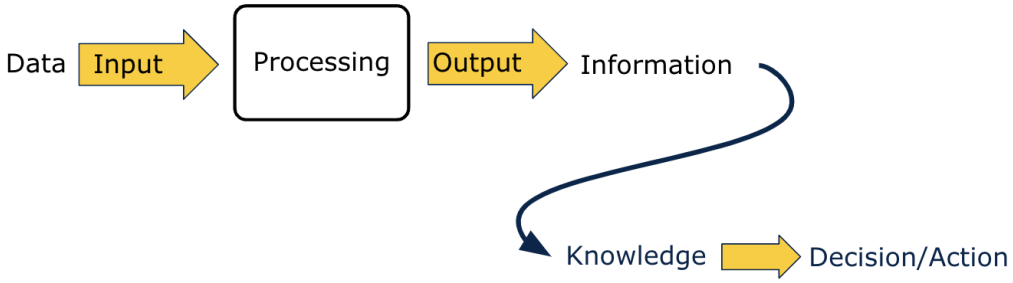
In this foundational module, we introduce data, artificial intelligence (AI), and machine learning (ML). Learning the vocabulary is an essential first step to beginning to appraise ML literature and apply ML outputs. In addition, it is important to learn the steps of ML model development. By the end of this module, you should be able to recognize the role that AI and ML play in evidence-based, data-driven medical decision making. Explain the potential strengths and limitations of AI and ML in healthcare, describe current applications of AI and ML in healthcare. And finally, define key terms and principles that are necessary for effective communication with other stakeholders.

Lesson 2: Big Data

Big Data in Health Care

Raw data refers to data that has not been processed or organized. Once data is processed, organized, and analyzed it becomes information (**Figure 1-1**). Information situates the data into context. Such contextualization clarifies how different data points are related to one another. Understanding this information results in knowledge, which may lead to decisions or actions based upon this knowledge.

Figure 1-1. Data, information, and knowledge



Every day, we produce large amounts of data when we use various web-based and mobile applications, search for entertainment on streaming services, listen to music, use wearables, shop online, engage in social media platforms and much more. These data can be processed and analyzed to generate information and knowledge. Beyond our day-to-day lives, there has also been a significant increase in the amount of health care data generated, mainly due to health care providers or health care systems use of electronic health records (EHR). Other sources of health care data include mobile health applications, smart devices, wearables, research, insurance registries, and government agencies.

This explosion of data or "data deluge" has led to "big data." Big data describes many types of data from many sources (variety) being generated rapidly (velocity) in large amounts (volume). Powerful computers with large storage capacity enable us to manage the massive and complex datasets often seen with big data.

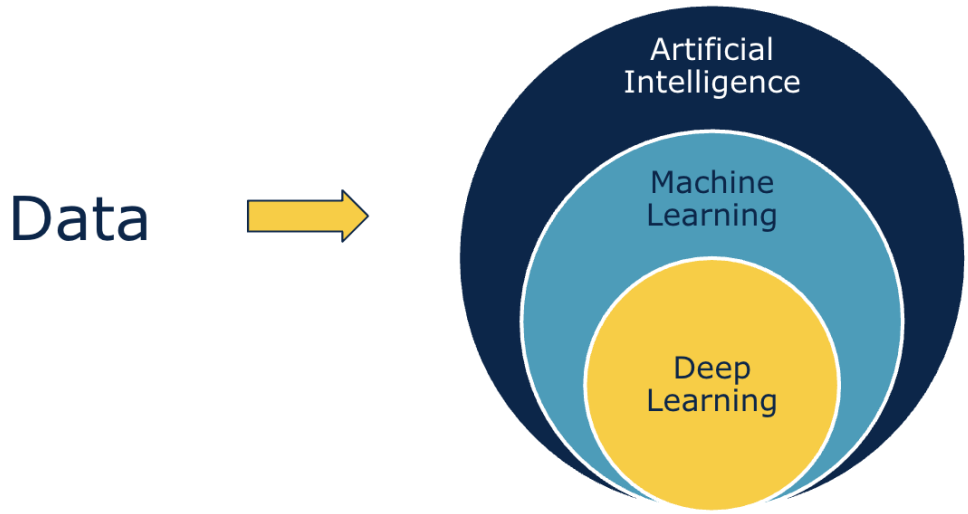
Big data has numerous applications in health care. As described in a New England Journal of Medicine Catalyst publication (NEJM Catalyst, 2018), big data may be used to identify causes of illness, prevent disease, make care more personalized, discover new treatments and medicines, identify medication errors, and flag possible adverse reactions. Big data can drive better patient care outcomes for long-term savings, identify disease trends based on demographics, geography, and socioeconomics, and much more.

Data and AI

Data alone are not very useful. Data must be analyzed and organized before it becomes useful information or knowledge. Artificial intelligence (AI) and machine learning (ML) may be helpful for organizing and analyzing data, especially large amounts of data.

AI is an umbrella term, and it involves the use of computers to perform tasks that typically require objective reasoning and understanding. AI has multiple domains as shown in **Figure 1-2**. In this course we focus on the ML domain.

Figure 1-2. Different domains of AI



ML has the potential to process large amounts of data and generate meaningful outputs or turn the data into knowledge. Before discussing ML in more detail, we need to understand data sources and structures.

Locating the Data

Data is the fuel that makes AI and ML go. Identifying potential sources of data is essential to developing health care ML models.

Data used to develop ML models originates from many places in the health care data ecosystem. The day-to-day health care that we provide may serve as a source of data. Examples include public health records,

immunization records, vital stats, birth and marriage certificates, and death certificates. Additionally, research generates data that may be used to train ML models. Environmental, geospatial, lifestyle, socioeconomic behavioral, and social data may also be used for model development.

We often refer to data as either qualitative or quantitative. Qualitative data is descriptive and is often in the form of words or language that can be observed but can't be computed. Quantitative data involves numbers that can be measured, calculated, and computed.

We may also describe data as structured or unstructured:

- **Structured data** is highly organized and often described as quantitative. This type of data includes records such as names, dates, addresses, phone numbers, bank account numbers, etc. Structured data is often stored in a *data warehouse*.
- **Unstructured data** is less organized compared to structured data and is often described as qualitative. This type of data may include video files, audio files, text, emails, and social media posts. Most data are unstructured. Unstructured data is often stored in *data lakes*.

Table 1-1 summarizes the characteristics of structured and unstructured data with some examples.

Table 1-1. Comparing structured and unstructured data

	Structured Data	**Unstructured Data**
Definition	Defined data that is defined, highly organized, and searchable.	Data that does not have a predefined model, identifiable structure, or organization.

Data Type	Quantitative	Qualitative
Storage	Data warehouse	Data lake
Example	Names, dates, addresses, phone numbers, bank account numbers, etc.	Video files, audio files, text, emails, social media posts, etc.

Datasets

Datasets are used to develop ML models. A dataset is a collection of data that is treated as a single unit by a computer. Because of the growth of big data, the datasets used to train ML models may be large and contain various types of data. In this course we refer to three types of datasets: *training sets*, *validation sets*, and *test* sets. We describe these data sets in greater detail in the upcoming lessons.

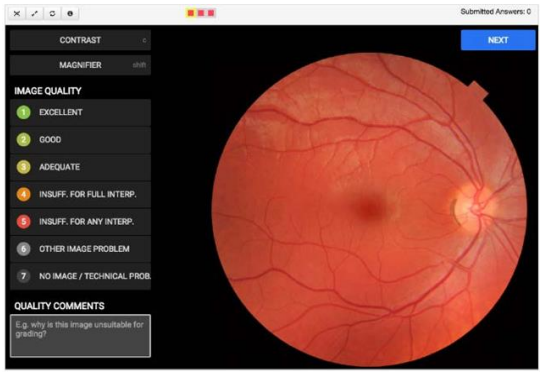
Lesson 2 Knowledge Check

1. Which of the following is NOT one of the 3-V's of big data?
    a. Volume
    b. Vast
    c. Variety
    d. Velocity

2. What type of data is likely to be stored in data lakes?
    a. Structured data
    b. Unstructured data
    c. Highly organized data

3. Refer to the Gulshan et al. (2016) Diabetic Retinopathy paper. Figure 1-3 is an example of the data used in this study. What type of data is this?
    a. Structured data
    b. Unstructured data

Figure 1-3. Data from Gulshan, et al. (2016)



Lesson 3: AI and ML in Health Care

A Brief History of AI and ML in Health Care

Over the years there has been a lot of hype around the use of AI and ML in health care settings. As you continue to learn about the potential uses of AI and ML in health care, remember to separate the potential promise of these technologies from the hype.

AI: 1950s - 1970s

In the 1950s Alan Turing described a test that would eventually become known as the Turing test. This test was used to determine if a computer could simulate human intelligence. He set out to answer the question: can machines do what we can do? Also, in the 1950s John McCarthy coined the term AI and Arthur Samuel coined the term ML.

AI: 1970s-2000s

While there wasn't a lot of progress made between the 1970s and early 2000s, one significant development in the 1970s was the development of MYCIN. MYCIN used patient information that was input by physicians and about 600 rules to provide a list of potential pathogens and recommend antibiotics that were adjusted for the patient's weight.

During the 1970s the Stanford University Medical Experimental Artificial Intelligence and Medicine (SUMEX-AIM) system was also created. This time-shared system allowed networking amongst clinical and biomedical researchers from several institutions. These collaborations led to the first NIH-sponsored AI in Medicine Workshop in the 1970s.

In the 1980s a decision support system developed by the University of Massachusetts was released. This system generated a differential diagnosis based on symptoms input into the computer. Initially, the system contained 500 diseases and eventually it contained over 2,400 diseases.

AI Winters

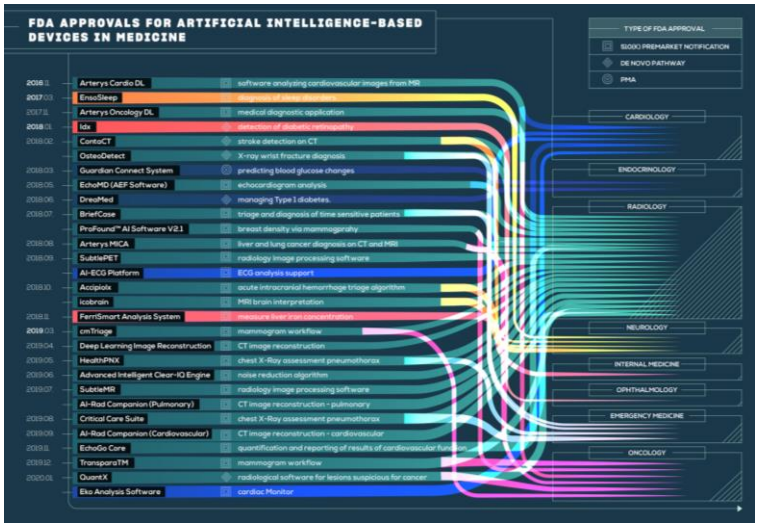
While there was some progress in the 1970s and 80s, there have been periods of decreased interest, funding, and hence, development of AI during this time period. These periods of decreased activity/interest are often referred to as AI Winters.

Digitization and Storage

Computers in the 20th century were large, expensive, and had smaller storage capacity. It wasn't until the 2000s that more powerful computers with increased storage capacity and faster speeds were readily available to many. This contributed to a proliferation of AI and ML models. We've also seen a proliferation of AI and ML models in health care. Much of this development could also be related to moving from paper-based systems to digital systems. For instance, health care records or data are now stored in digital form.

Figure 1-4 shows that the FDA has approved AI-based health care devices impacting all medical specialties to some extent.

Figure 1-4. FDA approved AI devices in medicine.



Currently, AI and ML are having a significant impact in diagnostics. These technologies may be used to augment diagnostic decision making and improve performance of the diagnostic process.

Diagnostic Applications of AI and ML in Health Care

AI algorithms may assist with detection of breast cancer on screening mammogram images (Shen et al., 2019) (see **Figure 1-5**). Each of the models tested in this study (Shen et al., 2019) demonstrated good performance as indicated by the area under the receiver operating characteristic curves in **Figure 1-6**.

Figure 1-5. Mammography images showing regions of interest and distinction between benign and malignant lesions (Shen et al., 2019).

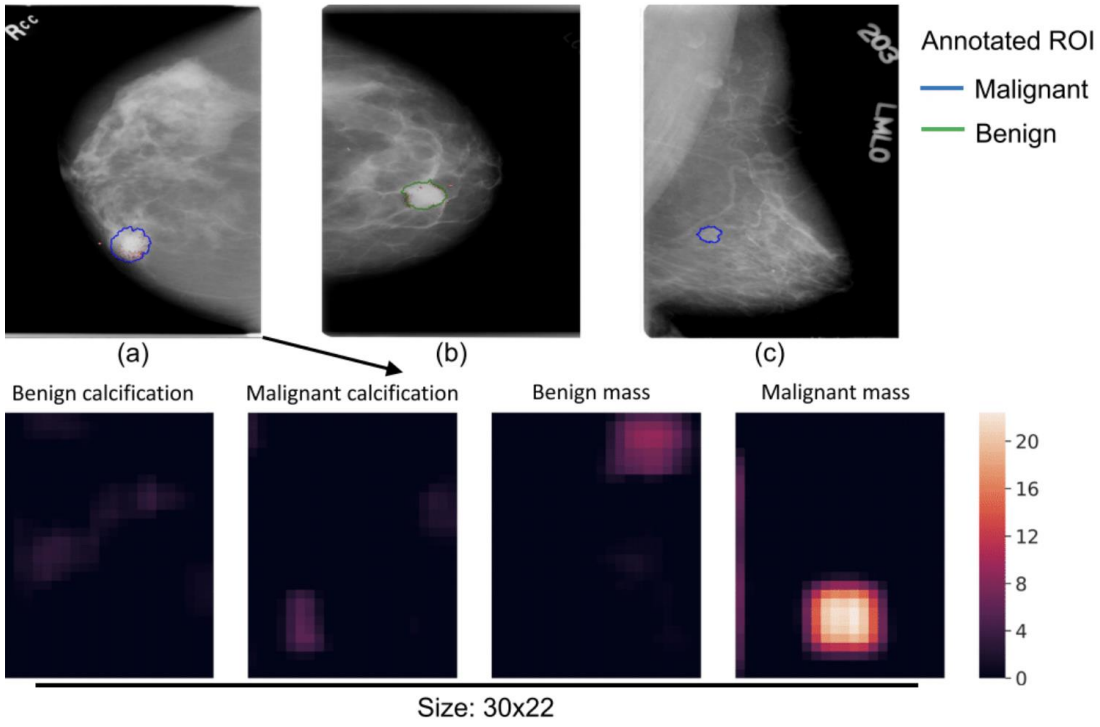
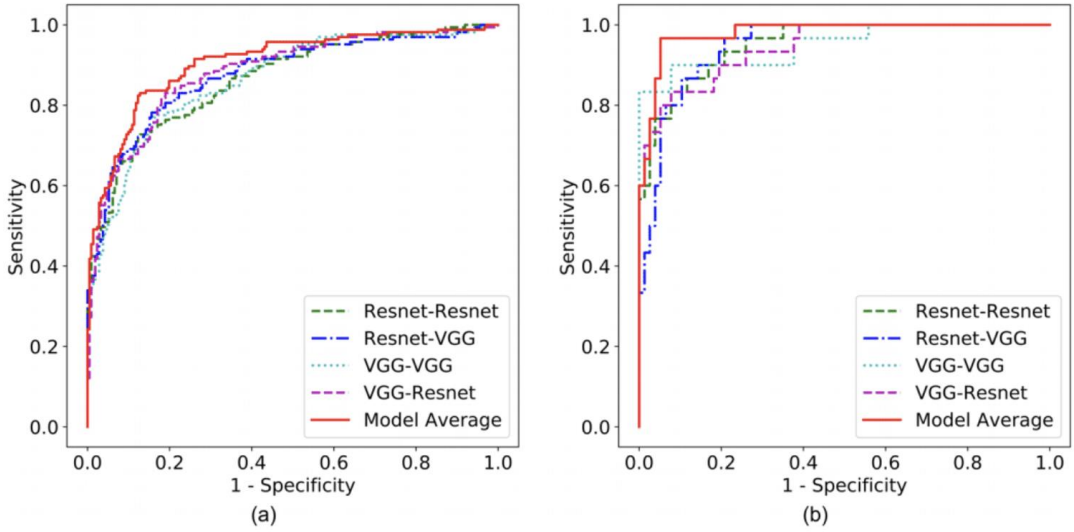


Figure 1-6. Receiver operating characteristic curves of four models developed to detect breast cancer on mammograms (Shen et al., 2019).



At this point, you may not be comfortable interpreting the results of receiver operating characteristic curves. This topic is covered in detail in Module 2: "Foundational Biostatistics and Epidemiology in AI/ML for Healthcare Professionals."

Another example of a diagnostic application of AI/ML are ML models that can autonomously diagnose diabetic retinopathy (Gargeya et al., 2017). Using fundoscopic images (**Figure 1-7**), this model demonstrated strong performance as noted by the area under the curve (**Figure 1-8**). It should also be noted that the IDX-DR model, a model designed to autonomously diagnose diabetic retinopathy, was the first ever FDA approved autonomous health care AI.

Figure 1-7. Fundoscopic images of the retina (Gargeya et al., 2017)

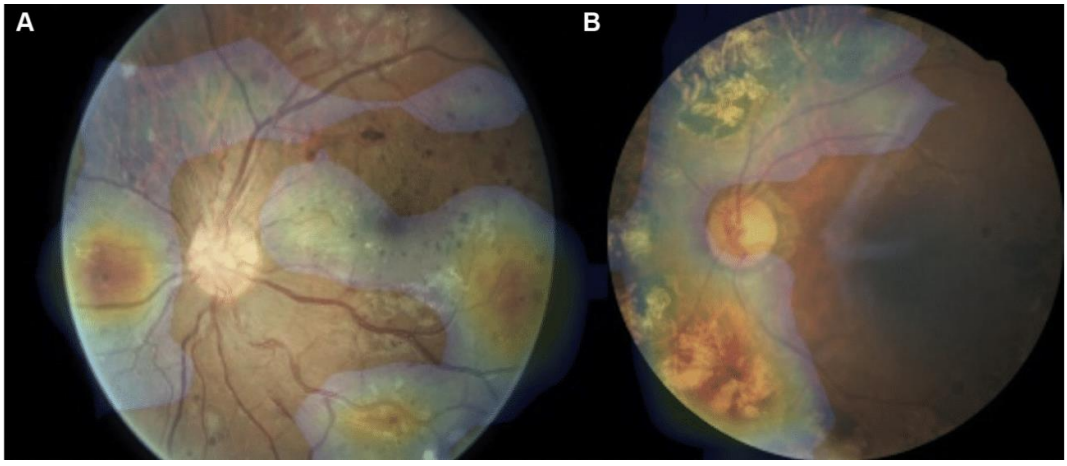
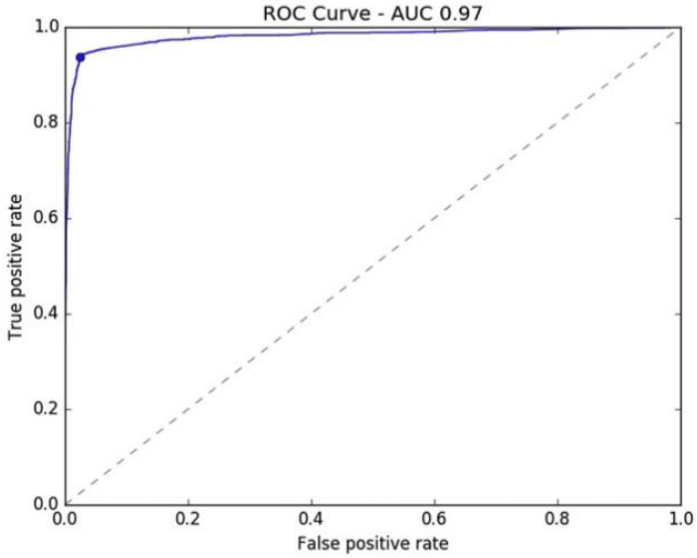
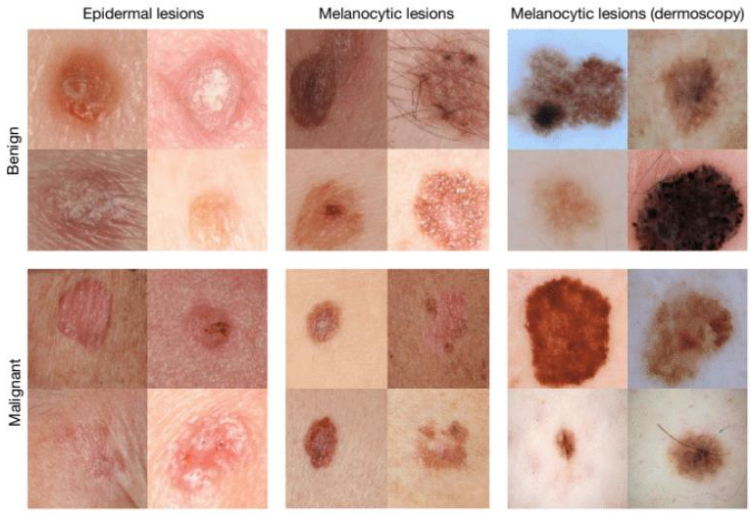


Figure 1-8. Receiver operating characteristic curve showing excellent performance by the IDX-DR model (Gargeya, et al., 2017).



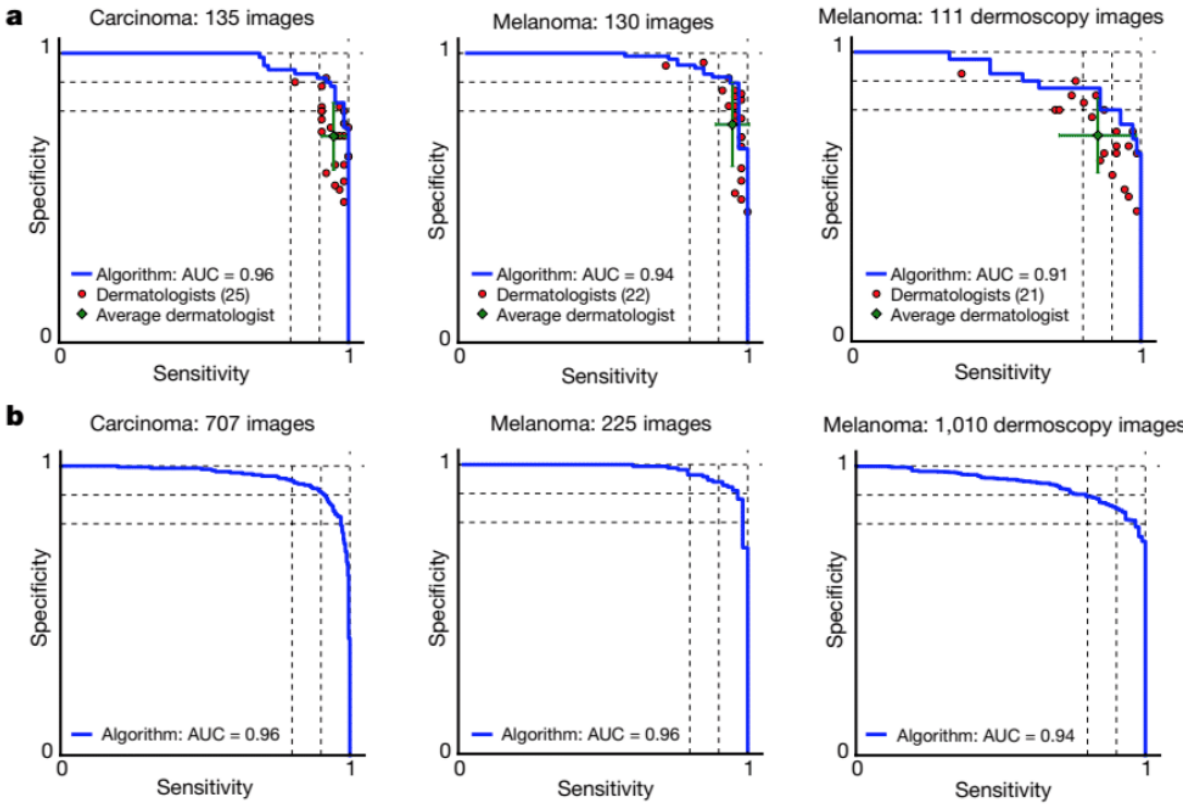
AI can also be used to assist with diagnosis of skin cancer on photographs. Esteva et al. (2017) showed that AI can achieve dermatologist level accuracy when diagnosing skin cancer on images (**Figure 1-9**).

Figure 1-9. Images of skin lesions (Esteva et al., 2017).



Notice the high area under the receiver operating characteristic curve for this algorithm (**Figure 1-10**).

Figure 1-10. Area under the receiver operating curve (Esteva et al., 2017)



Stakeholders

As we consider the impact of AI on health care and diagnostics, we must consider the specific groups and individuals who will be impacted by this technology. We must also consider how the implementation of AI into clinical practice and everyday life will impact the roles and responsibilities of various stakeholders. These stakeholder groups include but are not limited to health care leaders, developers, patients, researchers, insurers, policy makers, and clinicians.

- **Patients:** Patients are increasingly using AI-based technologies with

and without the knowledge of their health care providers. Patients will need to be educated on the role that AI plays in their health care. This education will be crucial to patients trusting that these technologies are both safe and effective.

- **Developers:** Those developing health care AI systems may not be clinicians or have clinical experience. They will need to know the right questions to ask as they develop AI algorithms that address important, clinically relevant issues. This will require partnering with clinicians and patients to understand their needs.
- **Researchers:** Those studying AI algorithms will also need to know the right clinical questions to ask as they evaluate the safety and efficacy of AI-based interventions.
- **Payers:** Regular use of AI-based technologies in health care is relatively nascent. Therefore, insurers will need to develop effective reimbursement mechanisms for the use of AI in clinical care.
- **Regulators:** Agencies like the United States Food and Drug Administration will play a key regulatory role in the development and implementation of AI-based technologies in health care. Policymakers will also consider issues related to data privacy, bias, fairness, economics, application of AI in clinical practice, and more.
- **Clinicians:** Ultimately frontline clinicians will be responsible for the use and application of outputs of AI that has been implemented into their clinical practice. Clinicians will also bear some responsibility in explaining to their patients the role that AI played in their care.
- **Health System Leaders:** Health system leaders will also be responsible for governance and regulation. When it comes to implementation of ML models in their health systems, they must ensure that AI is both safe and effective and that it continues to be

monitored after it's been implemented.

Each of these individuals or groups will be impacted by health care AI. Therefore, we all have a responsibility to ensure that we are informed stakeholders who can play the role that we've been called to play.

In the DATA-MD course, we aim to prepare you to become a vocal stakeholder. We encourage you to take a few moments to think about how AI may be used in your clinical practice and the role that you will play in its successful implementation and use.

Lesson 3 Knowledge Check

1. Describe at least two factors that have led to proliferation of artificial intelligence in health care.

2. What role do you see yourself playing as artificial intelligence becomes more common in health care?

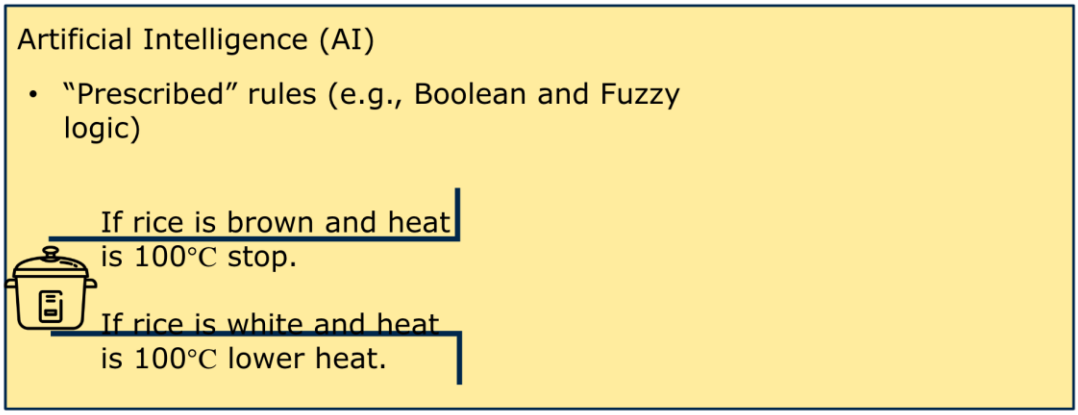
Lesson 4: Methodologies

What is ML?

The unifying goal of ML as a discipline is to use data to discover patterns or rules that generalize. The typical pipeline of ML is that we observe some data, learn a pattern, and then use this pattern as a rule that we can apply to unseen data. You might have heard that ML is related to AI. ML is a subfield of AI.

AI is the collection of algorithms. An algorithm is a set of steps or procedures that a computer takes. These algorithms are designed to mimic human reasoning in some form. Some of the examples of AI algorithms include Boolean or fuzzy logic, which is something that you'll find incorporated in kitchen appliances such as rice cookers. This kind of AI looks like a set of rules that are encoded in the system for example in a rice cooker (**Figure 1-11**).

Figure 1-11. An example of prescribed rules.



The AI algorithm in the rice cooker is a set of rules illustrated in **Figure 1-11**:

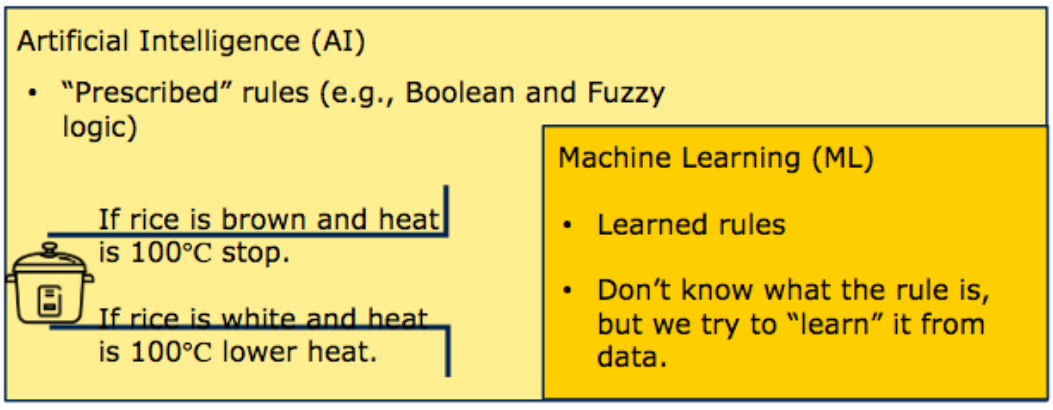
- If you're cooking brown rice and a given amount of water remains in the cooker, or if the heat is 100^0C, then you might want to stop cooking.
- If you're cooking white rice and the heat is 100^0C, then lower the heat and continue cooking.

In this example, a human being, based on experimentation or maybe by virtue of being a good cook, decided that these are the right rules to cook perfect rice.

How is ML Different?

What makes ML different from other types of software or programming paradigms is that the designer of the ML algorithm does not explicitly define a rule or a mapping from the inputs to the outputs. Rather, the designer gives the computer explicit instructions on how to search for that good mapping and the characteristics of what makes a good or accurate mapping. It is then the computer's role to find this good mapping (**Figure 1-12**).

Figure 1-12. ML as a subfield of AL.



ML Methodologies

We discuss three main types of ML methodologies and review examples of how they have been used in clinical or health care applications. Each methodology has a different goal and requires different types of data:

- **Supervised Learning**: there is a specific outcome or disease that we want to predict.
- **Unsupervised Learning**: there isn't a specific outcome that we want to predict but we want to discover new patterns.
- **Reinforcement Learning (RL)**: the goal is to learn the optimal sequence of actions to optimize long-term outcomes.

Supervised Learning

Supervised learning is the most common type of ML used in health care. In supervised learning we have a specific outcome or disease that we want to predict. We refer to that specific outcome as our *label*.

To develop supervised learning models, we need our data to come in pairs. These pairs may include the patient characteristics, and the label that we want to predict. In this situation the model may be trying to predict an outcome that will occur in the future.

For example, we may use supervised learning to create models that predict the likelihood that a patient will develop type II diabetes in the future, or we can use supervised learning to detect an existing health condition such as detecting the presence of pneumonia on chest x-rays.

The Setup Phase

Data Component

First, we start with a dataset. This dataset will have features that we will use to create our predictions. Features may include electrocardiogram (ECG) signals, or the data could include patient characteristics such as weight, height, body mass index, age, prior medications, medical history, or genomic data. These kinds of data are usually stored in a table similar to **Table 3-1**. Each row stores information about a single patient or a single patient encounter

Table 3-1. An example of a feature table.

Patient ID	Feature 1	Feature 2	….	Feature 100

Features can be images, e.g. medical images (**Figure 1-13**), or in textual form, e.g. a discharge note (**Figure 1-14**).

Figure 1-13. A chest x-ray, stored in pixel form, is used as an input to the model.

121	115	3	251	132
23.5	101.2	234	48.1	145
42	10	47.5	47.9	57.1
250	198	12	78	90

Figure 1-14. A discharge note used as an input to the model.

30-year-old female arrived at the emergency room with concerns about right lower abdominal pain that increases with walking

We usually use the variable x to denote these inputs regardless of what type they are.

Label Component

Labels represent the outcome that we're trying to predict. We usually use y to denote the outcome that we wish to predict. The type of label used depends upon the purpose of the model:

- **Binary classification model**: These labels could encode the presence or absence of a disease.
- **Multi-class classification model**: The labels could encode multiple diseases (e.g. pulmonary embolism, heart failure, pneumonia).
- **Regression model**: Used when predicting a continuous outcome, such as insulin levels or blood pressure.

The Learning (Training) Phase

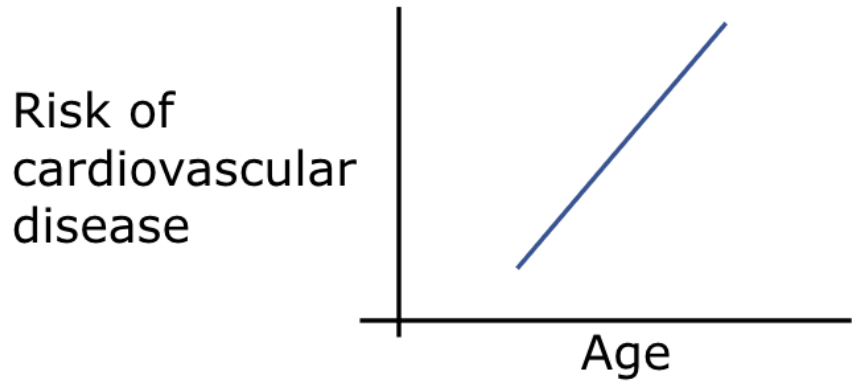
There are three main steps in the learning (training) phase.

Step 1: Define a model class.

Assume that we want to predict the likelihood or risk of cardiovascular disease as a function of the patient's age.

We might believe that the relationship between cardiovascular disease and age is linear. This means that as the age increases the risk of cardiovascular disease increases by a linear factor. In that case we might wish to consider the class of linear models. **Figure 1-15** illustrates one possible model within this class. But there are many different linear models that can encode the relationship between age and risk of cardiovascular disease. All the possible linear models that can encode the relationship between age and risk of cardiovascular disease are called the "*class of linear models*."

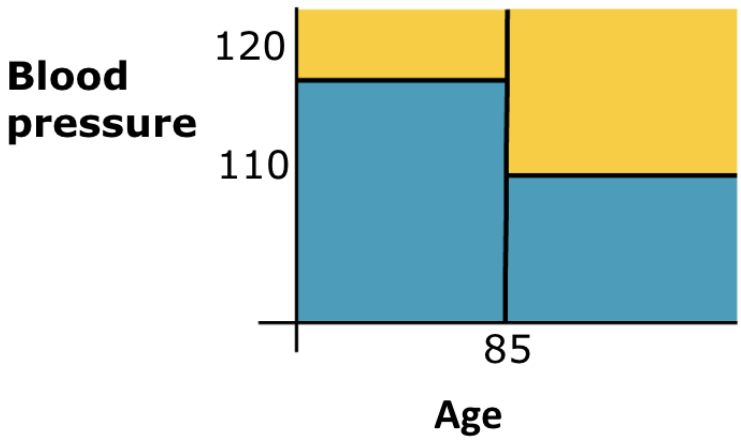
Figure 1-15. An example of a linear model



Alternatively, we could have a prior understanding that to predict whether a patient will have a cardiovascular disease, we need to consider not just age, but also blood pressure. And we may know, for example, that the relationship between the two variables and cardiovascular disease is non-linear. In this case, we might choose a different model class, the "*class of decision trees*."

Figure 1-16 shows one possible decision tree. This decision tree indicates that if the patient's age is above 85 and their blood pressure is higher than 110, then they may be at risk for cardiovascular disease. But if the age is below 85 and the BP is greater than 120, they may be at risk for cardiovascular disease. There are many other possible decision trees that set different thresholds for age and blood pressure and how they relate to cardiovascular disease. All the possible decision trees make up what we call the "*class of decision trees*."

Figure 1-16. An example of a decision tree

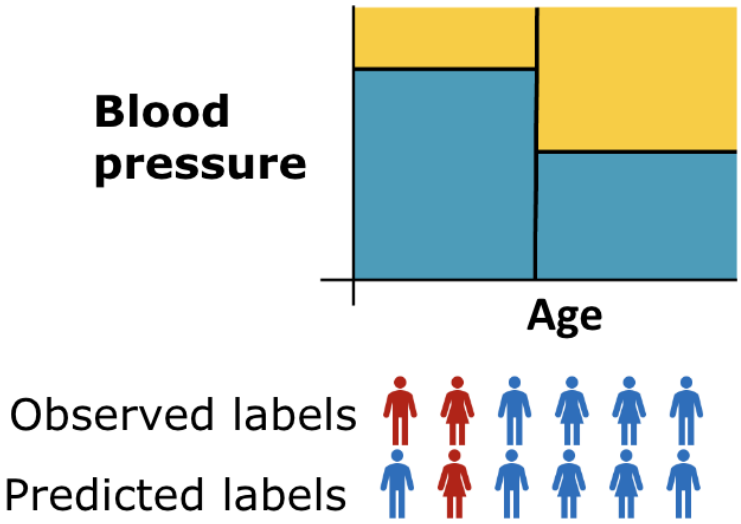


There are many other model classes. There are also challenges related to choosing a model class.

Step 2: Define Loss/Evaluate Model

Let's assume that we chose the class of decision trees as our preferred model class (**Figure 1-17**). Our dataset, including observed labels, has 6 patients. In this example the red patients are patients who have cardiovascular disease, and the blue patients do not have cardiovascular disease. Predicted labels show the predictions that this model makes. As shown **in Figure 1-17**, the model has mislabeled the first patient in its prediction.

Figure 1-17. Decision tree model



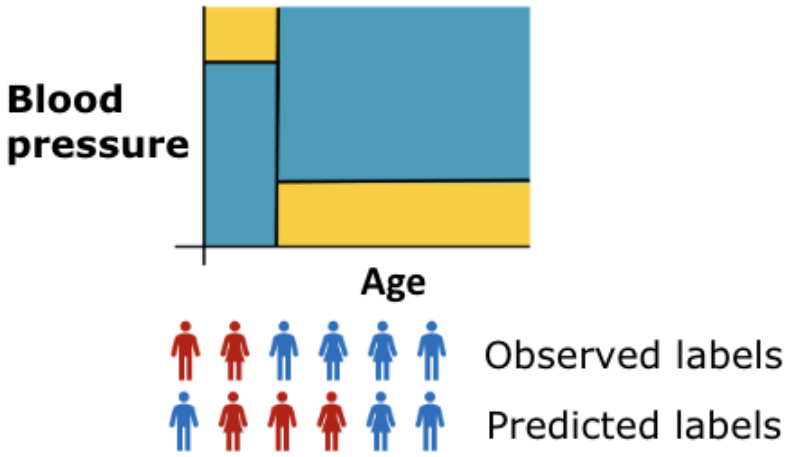
A reasonable evaluation criterion here is the "*misclassification rate*", which is the number of mistakes the model makes divided by the total dataset size, which in this scenario is the total number of patients. Remember that different settings or situations might require different evaluation criteria. In binary classification, we can compare the observed labels to the predicted labels to see if they match. This is referred to as the misclassification rate. There are many other evaluation criteria that generally measure the extent to which the predicted labels match the observed labels.

$$misclassification\ rate\ = \frac{number\ of\ incorrect\ predictions}{total\ number\ of\ datapoints}$$

For the example in **Figure 1-17** the misclassification rate is ⅙.

Now consider a second model shown in **Figure 1-18**. It uses the same dataset and as the predicted labels show, this model mislabels more patients and has a higher error or misclassification rate of ½.

Figure 1-18. Another example of a decision tree model



We want to choose the model with the lower error rate because we expect it to give better predictions for new patients.

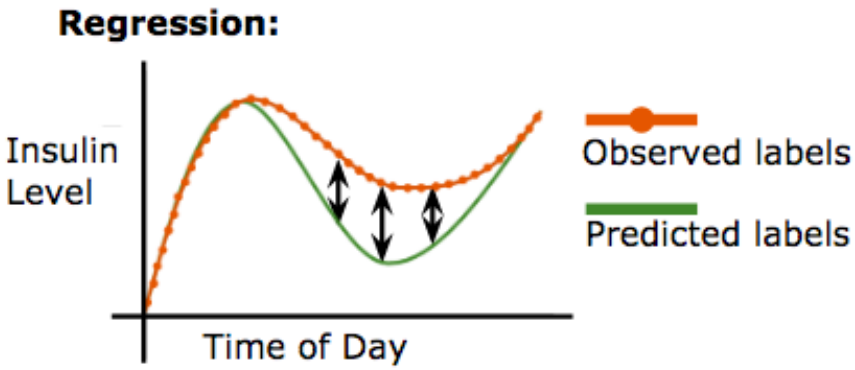
Thus far, we defined our model class, decision trees, which contains all the possible partitions based on age and blood pressure, e.g. models shown in **Figure 1-17** and **Figure 1-18**. Each model leads to different predictions about which patients are going to have cardiovascular disease. We used misclassification rate as our evaluation criteria.

An inefficient strategy would be to try to enumerate all the possible models in a model class and estimate their misclassification rate. This strategy is inefficient for most model classes because there might be an

infinite number of models in a model class. Consequently, we need a learning algorithm that efficiently selects the best model for our intended purpose.

Note: A different evaluation criteria is used for regression. Consider a different prediction problem where we're trying to predict the insulin level of a patient using the time of day. On the plot shown in **Figure 1-19**, the predicted labels are green, and the true observed labels are orange.

Figure 1-19. A regression model.

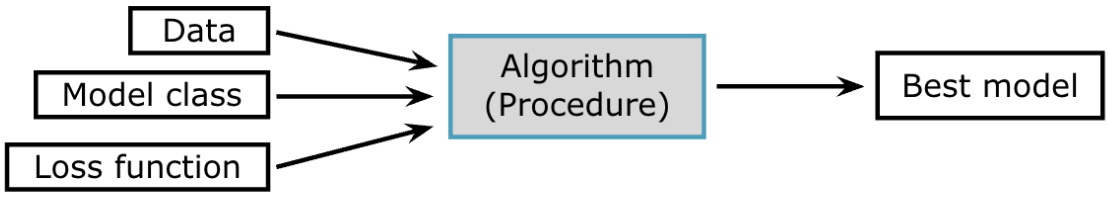


In this example, we could look at *mean squared error*, the squared distance between the predicted and the true labels, and average over all these squared distances in our dataset. Or we could compute the absolute squared error, measure absolute distances, and calculate their means. The Huber loss is another type of evaluation criteria that combines both the mean and absolute errors.

Step 3: Define a Learning Algorithm

An algorithm is a procedure consisting of a set of steps. It takes an input and generates an output. In our specific case, a learning algorithm takes as an input a dataset, the model class, and the evaluation criteria to select the best model in the model class (see **Figure 1-20**). "Best" is defined with respect to the evaluation criteria that we chose.

Figure 1-20. Input and output of a learning algorithm.



Note that an algorithm component usually involves an optimization step. This optimization step helps us find the model that minimizes or maximizes our evaluation criteria allowing us to select the best model in the model class without evaluating every model in the model class.

Unsupervised Learning

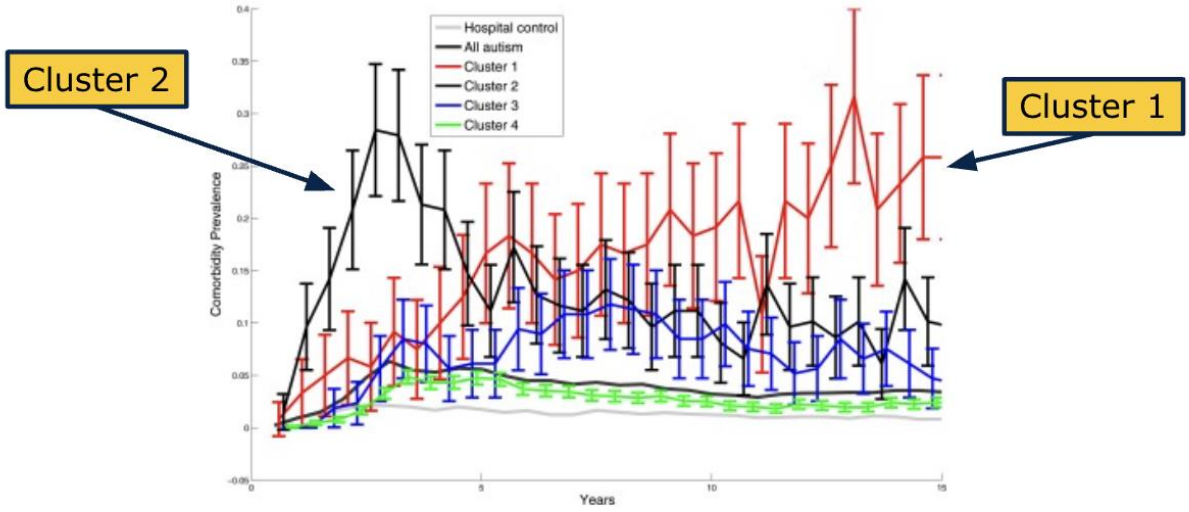
In unsupervised learning we don't have a specific outcome that we want to predict. Instead, we want to discover new patterns. For example, we

may want to divide patients into groups or clusters that have similar health care utilization patterns.

Let's discuss an example. Researchers from Harvard developed an approach that takes as input data from EHRs of patients on the autism spectrum (Doshi-Velez et al., 2014) to discover different subtypes of individuals on the autism spectrum (**Figure 1-21**).

In this example, there were no labels, and the goal was to find distinct trajectories in clinical manifestations beyond the neurobehavioral criteria in the Diagnostic and Statistical Manual (DSM). Doshi-Velez et al. (2014) examined comorbidities and utilization patterns of patients with autism spectrum disorders between the ages of 0 to 15 and they found 4 different trajectories. This plot in **Figure 1-21** highlights temporal patterns of developmental delays that varied between the subgroups. Note that cluster 2, characterized by multisystem disorders, had a spike in diagnoses for developmental delays at age 2.5. Specific developmental delays for individuals in cluster 1 rose steadily through age 15.

Figure 1-21. Trajectories of patients with autism spectrum disorder: Temporal patterns of comorbidities in subgroups of patients with autism spectrum disorder (Doshi-Velez et al., 2014).



Similar to supervised learning, the input features can be EHR, imaging data, or text such as discharge notes. However, we do not have a specific label that we want to predict. Instead, the goal is typically to discover ways in which patients cluster, or to reduce the dimension of a large feature set to something that is more manageable.

These models are developed using these the same three steps described for supervised learning:

1. Define a model class.
2. Define evaluation criteria.
3. Define an algorithm that takes in the model class, evaluation criteria and the data to provide the best model.

The difference here is that the types of suitable model classes that we can consider are different. Instead of considering decision trees, we can choose the class of all possible "clustering" that give three clusters, and there are many other possible choices of classes.

Remember that all the evaluation criteria used in the supervised learning setting relied on having an observed label. None of those evaluation criteria are used for unsupervised learning.

Reinforcement Learning (RL)

With RL the goal is to find optimal sequences of actions or decisions that optimize long term outcomes that we're interested in. For example, we might want to learn the optimal intervention strategy to treat patients with sepsis.

There are two different types of RL:

- Online
- Offline

Online RL

Online RL requires active experimentation or exploration by a RL agent, such as a robotic arm. This agent interacts with its environment to collect data that is then used to learn which is the optimal sequence of decisions.

One example of online RL is Alpha Zero, which is an algorithm created by researchers at DeepMind. This algorithm learned how to play the game of Go by playing games against itself. It takes a sequence of

moves, and observes the outcome, that is, which side won. By doing this several times, it can learn which sequence of moves leads to winning.

As you can imagine, applying this kind of learning in a setting where we are evaluating sequences of interventions on a patient is both costly and unethical. Therefore, online RL is infrequently used in a health care setting because of the potential for patient harm.

Offline RL

Offline RL which is more appropriate for health care settings. In offline RL, instead of having an agent collect data by interacting with its environment, the agent learns by observing decisions made by humans, and the corresponding long-term outcomes after these decisions.

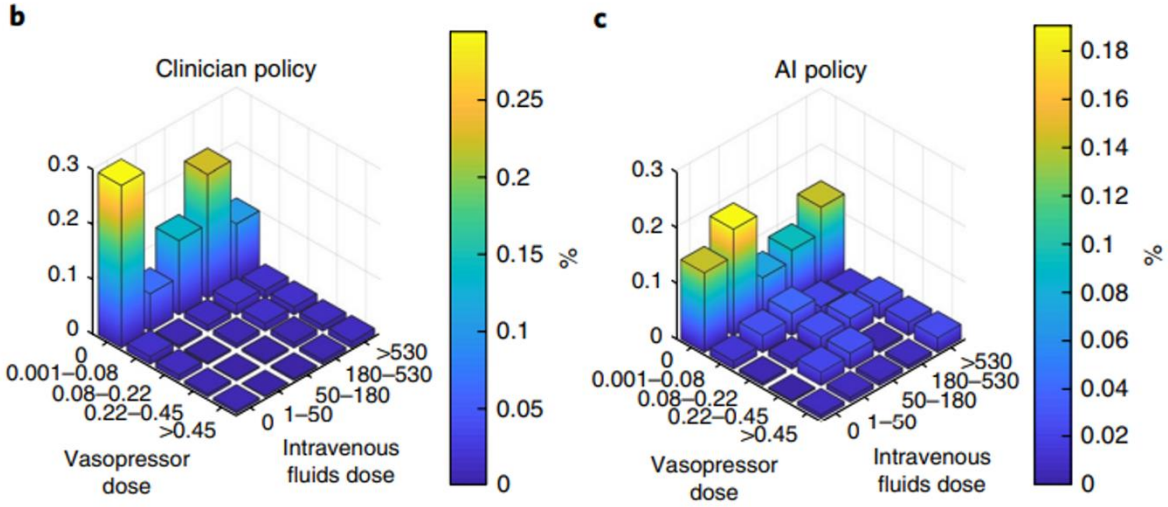
To conduct offline RL, we need three pieces of information or data:

- *State data*: The data that describes the patient's state at each point in time of the observation.
- *Action data*: The data that describes what was done to the patient. For example, on day one the patient was given IV fluids and vasopressors. On day two we reduced the amount of vasopressors given, and so
- *Reward*: This data tells us how good the intervention strategy has been. Did it end with the patient expiring? Or did it end with the patient getting discharged home after a short stay? What was the final outcome?

A well-studied application of RL in clinical care is management of sepsis (Komorowski et al., 2018). Here, the task is to find the best

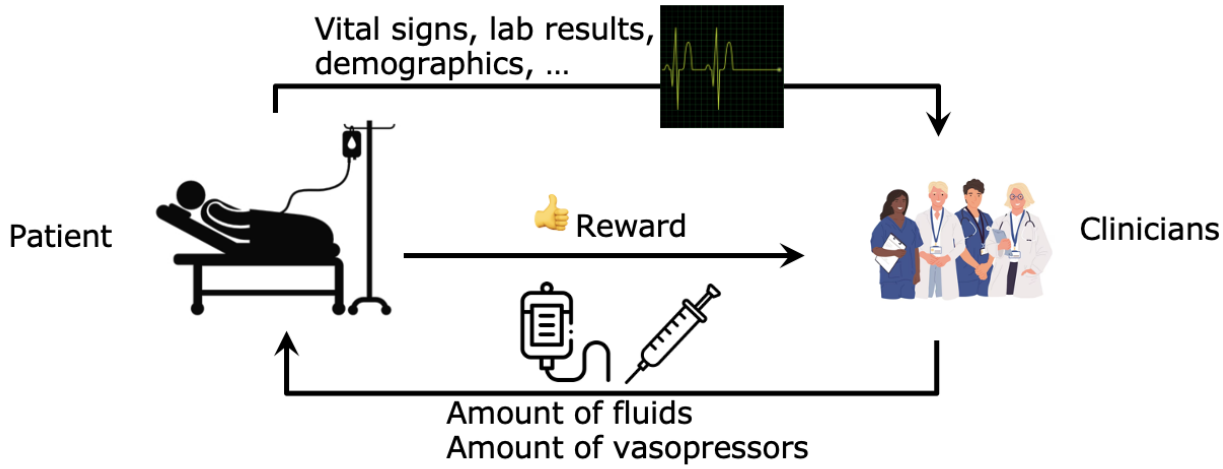
combinations of vasopressors and IV fluid doses over time that lead to recovery from sepsis. **Figure 1-22** shows the difference between the clinician's policy (actions), the graph on the left, vs. the AI policy, the graph on the right. The RL agent tended to give more IV fluids than the clinicians.

Figure 1-22. Differences between the clinician's policy vs the AI policy.



Let's look deeper into the sepsis example to understand what the data looks like. We can think of sepsis treatment as a decision-making loop (**Figure 1-23**), where at any particular time point, the clinicians can observe the patient's health status, including vital signs, lab results, demographics, and decide how much fluid and how much vasopressor to give to the patient. After the treatments they will see if the patient fully recovers and survives sepsis. If not, they can proceed to select the next set of treatments.

Figure 1-23. Treatment decision making loop.



Let's consider what the data looks like in the offline RL setting. Here we have not only features that describe the patient state but we also have the actions or interventions that were taken to treat the patient. This is not a single bulk of data but sequences over time. We have:

- Patient's state for day 1, e.g. vitals and lab results, collected on day 1 of the inpatient visit.
    - Amounts of vasopressor and IV fluids administered on day 1.
- Patient's state for day 2
    - Amounts of vasopressor and IV fluids administered on day 2.
- …
- Patient's state for day n
    - Amounts of vasopressor and IV fluids that were administered on day n.

Instead of a label, as in supervised learning, we have a reward. This reward can be coded as 1 if the patient was discharged home or 0 if the patient died.

Using this data we can train a model. The model takes in the patient's states, and outputs the best sequence of actions that we should follow depending on the patient state. While the model learning process is overall similar to the supervised and unsupervised learning, it includes more caveats.

Semi-Supervised Learning

Semi-supervised learning uses a mixture of labeled and unlabeled data. The goal is usually similar to supervised learning where we want to generate predictions for our new patients. However, in some cases of semi-supervised learning, we may have a dataset with many missing labels. Instead of discarding these unlabeled examples, semi-supervised learning makes use of the unlabeled data in clever ways to provide even better models.

Deep Learning

Deep learning is a subfield of ML that is inspired by the organization of the human brain. As we learn, neurons in our brains are activated and connections between neurons are made. These connections are called "neural networks."

In deep learning a computer learns by seeing many examples and extracting features from these examples to classify things or make predictions. For example, a computer may be assigned the task of identifying birds in photos. This requires seeing many photos of birds.

The first layer of neurons in the artificial neural network is the *input layer*. Images of birds are fed to the input layer. At this layer, the network may extract features from pictures such as the pixels in the image. After processing, the input layer neurons transfer information to the hidden layers of the artificial neural network via channels.

Artificial neural networks can have one layer. But for it to be a deep neural network it should have *at least two layers*. Information continues to be transferred to other hidden layers in the network. Eventually, the information reaches the output layer, and the image is classified based on the learning that took place in the neural network.

Deep learning is present in many applications that we use daily. For example, Amazon uses deep learning techniques to make recommendations for purchases based on our prior purchases and viewing. Voice assistants such as Siri and Alexa also employ deep learning

techniques. Deep learning is also used in identifying the location of pictures.

Some limitations of deep learning include the need for very large datasets to train the model. In addition, powerful graphical processing units to process the data are needed, and deep learning models can take hours to months to train.

Lesson 4 Knowledge Check

1. Which machine learning methodology involves training a model using labeled data.
    a. Supervised learning
    b. Unsupervised learning
    c. Reinforcement learning

2. Match the machine learning method with the appropriate description.

Description	Method
Requires labeled data to generate a prediction for new examples.	Reinforcement Learning
Uses unlabeled data to discover new patterns (clusters)	Supervised Learning
Model trained using a mixture of labeled and unlabeled data to generate predictions for new examples	Unsupervised Learning
Goal is to identify a sequence of actions to optimize long term rewards.	Semi-Supervised Learning

Lesson 5: Model Development

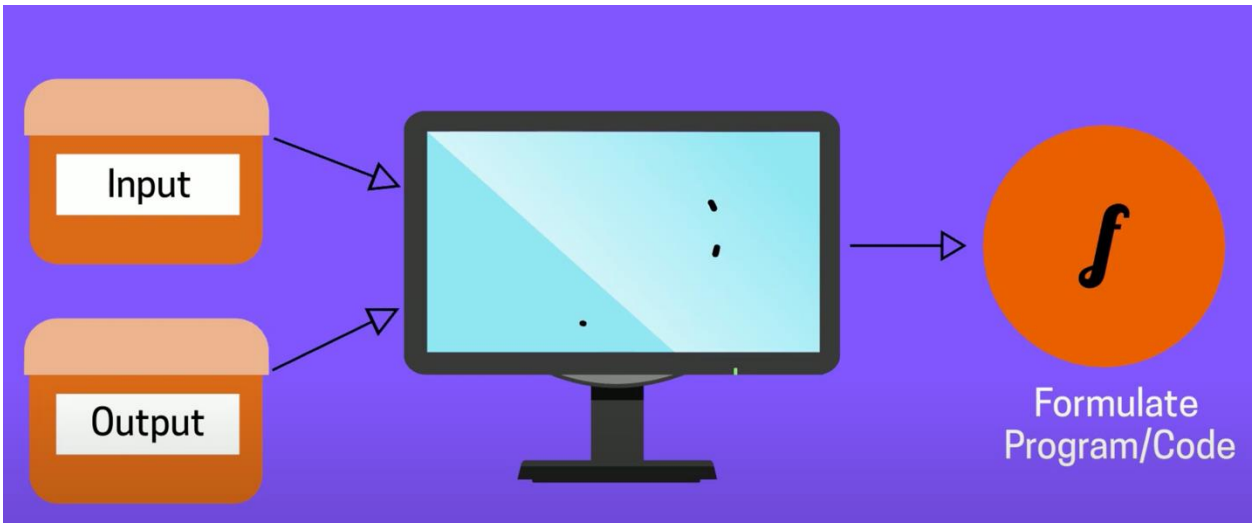
Model Development

How are ML models developed? We need to first understand traditional computer programming:

1. A person writes the rules for the computer to follow.
2. The computer process processes data based on those rules and provides an output.

In ML, explicit programming by a human is not necessary. Instead, both the input and output (e.g., supervised learning) are fed into the computer and the computer learns the association between the two to formulate the rules, program, or model (**Figure 1-24**).

Figure 1-24. Programming in ML



Steps for developing a health care ML model include:

1. Define a problem that needs to be solved and can be solved
2. Ensure that an effective intervention or solution exists to address the problem.
3. Determine if the necessary data is available to develop the algorithm.
4. Splitting data into three datasets: an 80/20 split of often described. 80% of the data is used for training and validation, and 20% is used for testing the model.
    a. *Training*: 80 % of the data is used for training. The training set is the data set that the computer learns from. For example it may learn the associations between an input and an output (e.g., supervised learning) or it may learn similarities between inputs and place different types of input into clusters or categories (e.g., unsupervised learning).
    b. *Validation*: The validation set sometimes referred to as the tuning or holdout set is used to ensure that the model is not overfitting or underfitting to the data used to train it.
    c. *Test*: The test set is used to evaluate the model's performance on new data that it has never seen.

By splitting the data set into these three parts we can ensure that the model is accurate and reliable.

Training Set

In a typical supervised learning model, during the training phase the model uses the training set to learn the *parameters* that map the input features (independent variables) to the labels or output features (dependent variables). This is referred to as "fitting the data to the algorithm."

The parameters are values that are learned by the model and make up the model determining its output and performance:

- During training, the values for the parameters may be updated iteratively as the model learns.
- The final value of the parameters will depend on tuning of the *hyperparameters* to optimize the performance of the model.

The training set is critical for developing an accurate model. Choosing the right training set and ensuring it is properly curated is one of the most important steps in developing a good ML model.

Validation Set

Once the model has learned the parameters, it should be evaluated on the validation set. The model's performance on the validation set helps developers understand how the model might generalize to unseen data, and whether it has learned the maps or associations from the inputs to outputs at an acceptable level. This involves using the validation set, an unseen, held-out portion of the training set to evaluate the performance of potential models.

During validation, a process called "*hyperparameter tuning*" occurs. Models with various hyperparameter values are evaluated to find the best

hyperparameter settings that produce the best performance for our specific task of interest.

Adjusting the hyperparameters has an impact on the parameters learned by the model. The model with the best performance is identified and is then trained on the full training set, including the validation set.

Hyperparameters are parameters whose values are set prior to the commencement of the training process. In contrast to parameters, hyperparameters are directly controlled by developers. They determine the values of the parameters that the model learns. Hyperparameters are optimized during training in a process called *tuning*.

Validation involves evaluating the model's performance on the validation data set and allows us to estimate how well the model is fit to the training data set. If the model's performance is acceptable at this step, we can stop training.

Overfitting and Underfitting

We want to ensure that the model is not *overfitted* or *underfitted* to the training data and that it can be generalized to data that it has never seen before.

Overfitting means the model predicts the data that it was trained on too well. Underfitting means that the model does not predict it well enough. Both of these issues will lead to a model that does not provide accurate predictions when it sees new data.

We can evaluate the performance of the model on the *validation set* to see if it is overfitting or underfitting. If model performance on the

validation data set is similar to the performance on the training data set, we can be confident overfitting or underfitting isn't present.

Test Set

Once we have trained and validated the model, we can use our test set which the model has not seen. During testing, the goal is to determine how accurately a model performs a given task, for example identifying lung cancer on a chest x-ray.

If the performance of the model on the test set is similar to its performance on the validation set, we can be confident that overfitting or underfitting has not occurred.

Challenges with Model Development

There are some of the common challenges that we might encounter when developing ML models. We will focus on challenges in the context of supervised learning, where the goal is to learn a model that predicts a specific outcome given input such as patient characteristics. But these challenges also appear in unsupervised learning and RL. These can be challenges with the data or with the modeling. We will look at the concept of overfitting and challenges with the loss function.

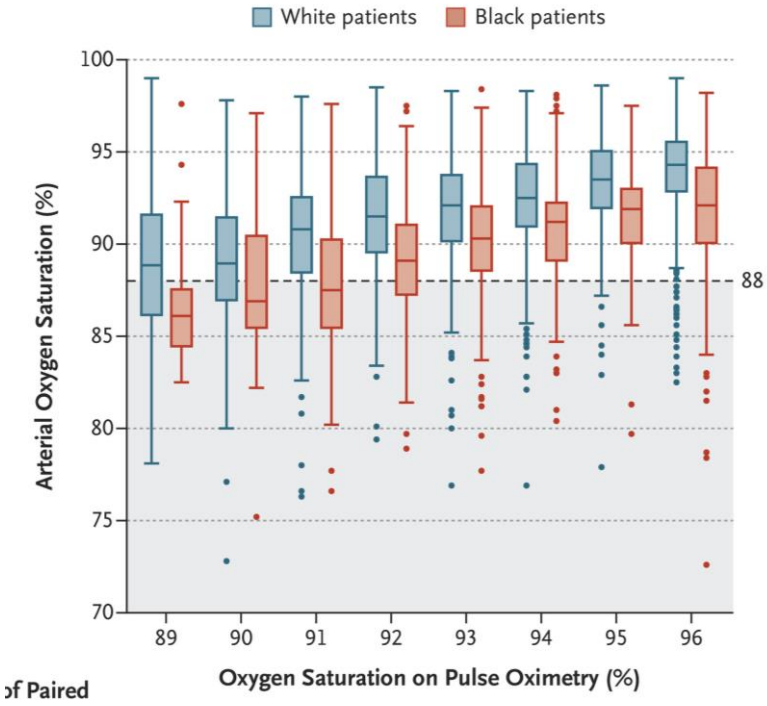
Data Challenges

One of the challenges that we might face is that the training data are not representative of the population in which we want to deploy the model, i.e. the testing data. For example, the training data could be predominantly older men, while the testing data is predominantly younger women.

In some cases, we might be able to address this challenge without collecting additional data, but that's not always possible. Appropriate solutions will vary, but one solution may involve re-weighting the data to create a pseudo population that matches our target population. This solution is only appropriate if we know what our target population is, and if our training data has some representation of the target population.

There are instances when the data measurement process isn't perfect. In many cases this imperfection is not random. For example, we may want to predict some outcome using patient vital signs such as pulse oximetry. As the plot in **Figure 1-25** shows, pulse oximetry measurements have some inaccuracy when compared to the gold standard lab measurement of arterial oxygen saturation (Sjoding et al., 2020). Furthermore, pulse oximetry measurements have been shown to be less accurate for Black patients compared to White patients.

Figure 1-25. Plot comparing the accuracy pulse oximetry readings in Black patients to pulse oximetry readings in White patients.



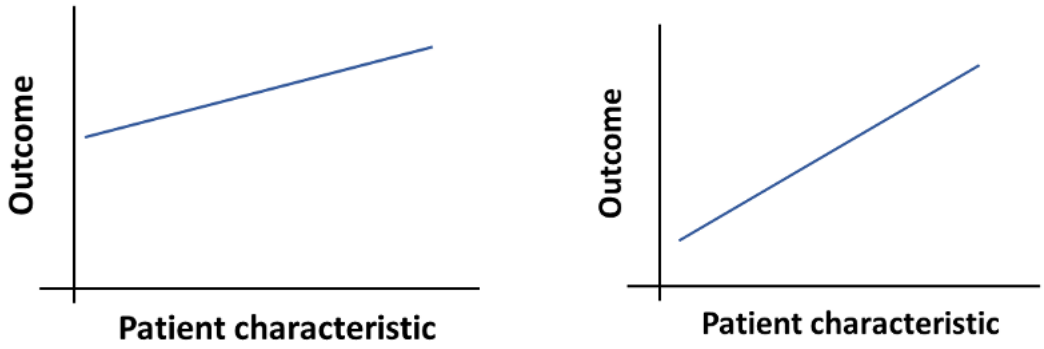
A model trained to predict lung conditions using vital signs including pulse oximetry might be inaccurate, especially when making predictions for Black patients. In some cases, we may be able to design algorithms to address these data limitations or maybe collect better data. But in general, if you are developing the model a good strategy is to transparently describe the data used for model development and communicate its limitations. If you're reading a paper about ML or implementing an existing model, it is always good to exercise caution in interpreting the results of an existing model.

Choosing a Model Class

In addition to data challenges, we might also have challenges with the modeling process. In the Methodology Lesson of this module, we discussed choosing a model class as an important step in model development. A model class is the set of functions or mappings that we will search within to find the best model that fits our data.

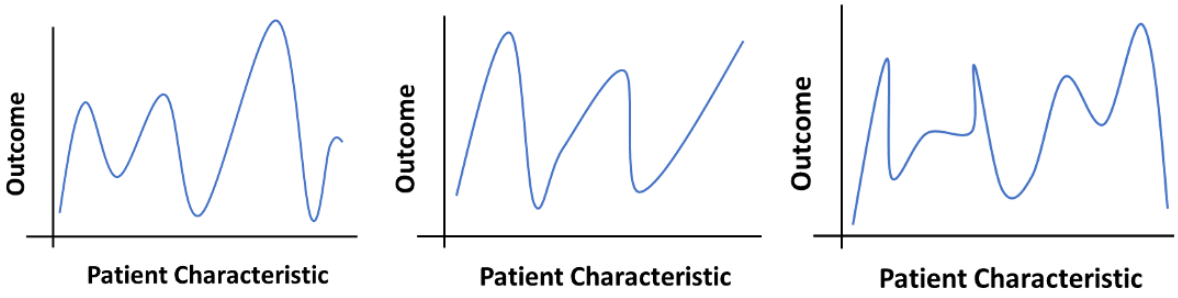
We can choose the set of all linear functions as a model class. Figure **1-26** shows different functions in the class of linear functions that model the relationship between patient characteristics and an outcome of interest.

Figure 1-26. Functions in the class of linear functions



Or, we can choose a different class, such as deep neural networks, that allows us to model more complex relationships between the patient characteristics and the outcome (**Figure 1-27**).

Figure 1-27. Class of deep neural networks



One challenge that we may encounter when developing a model is choosing an appropriate model class if we have knowledge about the relationship between the inputs and the labels. We want to pick a model or a model class that can encode this relationship, but in many cases we don't have such knowledge. Another strategy is to choose model classes that are very *expressive*, or capable of performing many complex computations.

Some model classes are expressive to the point that they can be universal approximators, meaning they can approximate almost any function. This is true of specific types of deep neural networks. If we're using these model classes we don't usually need to know anything about the relationship between the inputs and the label. A downside of using these very flexible models is that they typically require more data.

If we have access to a small dataset, we may be unable to develop reliable models if we choose a very flexible model class. In fact, if our model class is very expressive overfitting may occur.

Overfitting and Regularization

Overfitting is one of the most important and most commonly encountered problems when developing ML models. The term overfitting refers to models being so expressive that they can memorize the data instead of learning a meaningful pattern. Another way to think about it is that the model is so expressive that it can overfit to "noise" in the data.

For example, suppose our data looks like **Figure 1-28**. A model that exhibits overfitting might look like the red line on plot in **Figure 1-29**.

Figure 1-28. A sample dataset.

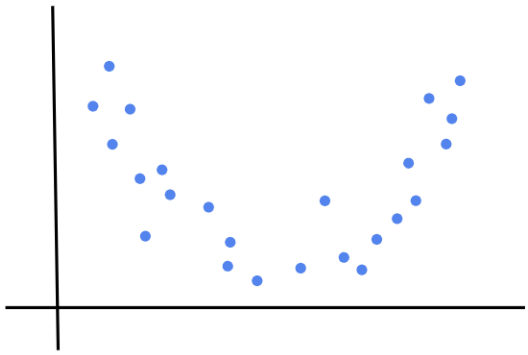
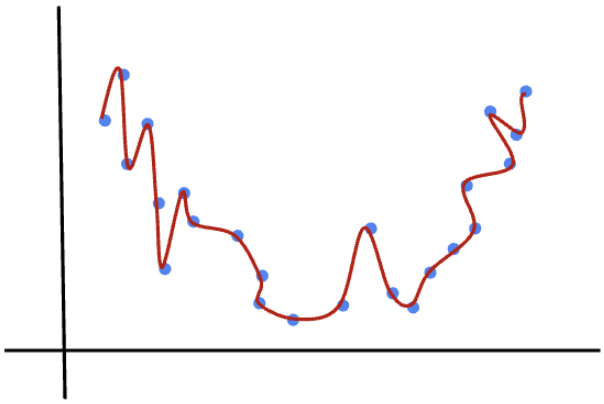
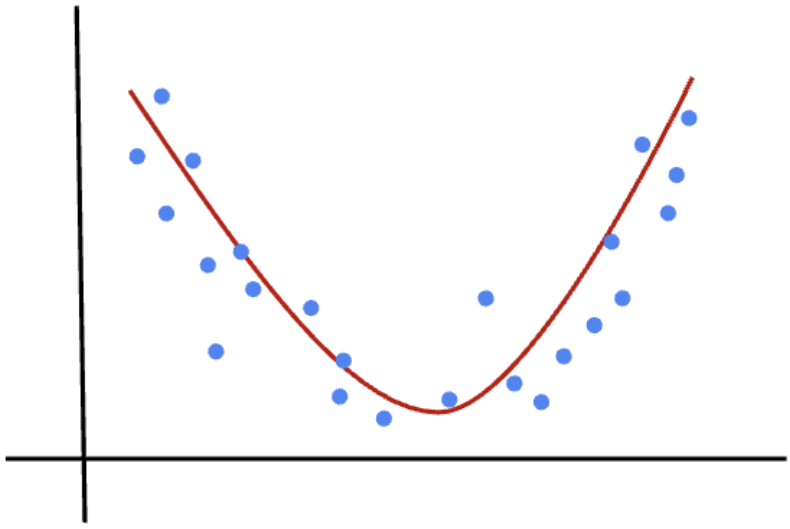


Figure 1-29. An overfitted model.



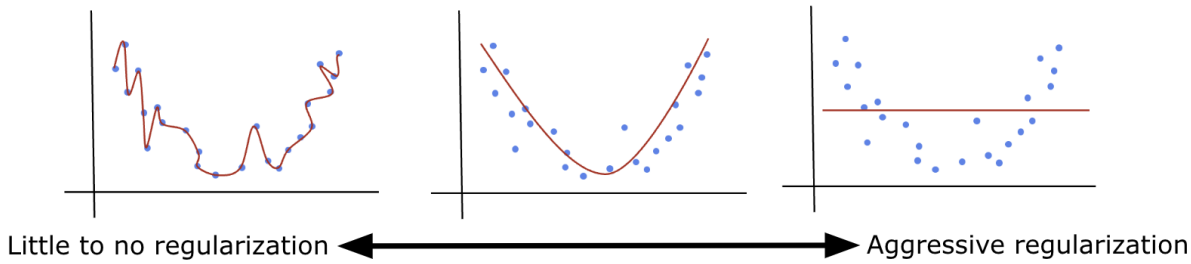
A commonly used solution to avoid overfitting is *model regularization* where we can limit the expressivity of the model to avoid overfitting (e.g. **Figure 1-30**).

Figure 1-30. A regularized model.



Consider **Figure 1-31**. How do we decide on how much regularization to enforce? How much do we want to limit the expressivity of our model? A widely used type of cross validation is *K-fold cross validation*.

Figure 1-31. Effect of applying various levels of regularization strengths.

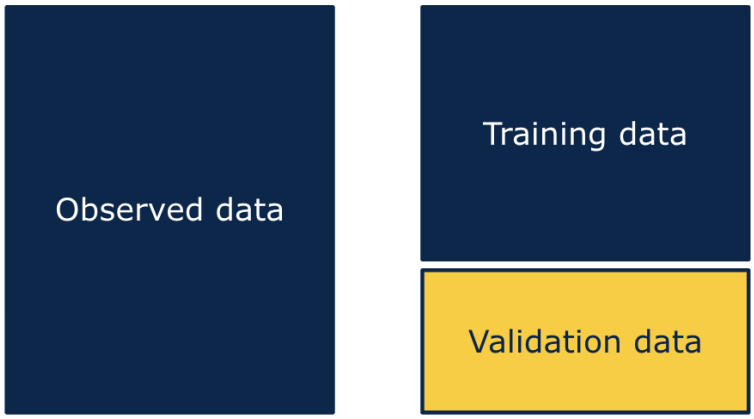


Picking the Regularization Strength

To illustrate how cross-validation works assume that we have four different levels of regularization that we're considering. Let's also say that we've decided to use K=3 which is three-fold cross-validation. In this case, we will slice the data that we are using for training into two-thirds used for model training and the remaining one-third is used for validation (**Figure 1-32**).

We train four models each with a different level of regularization using the training data only and apply each of the four models to the validation data. This means that every data point in the validation set will have four different predictions based on those four models.

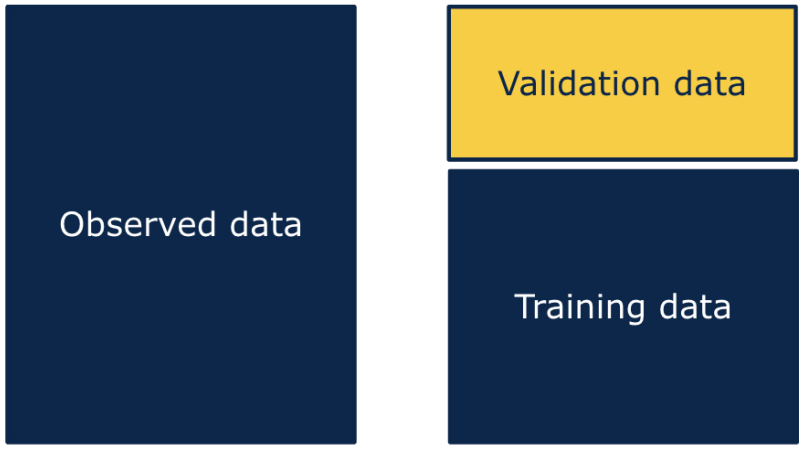
Figure 1-32.



We then evaluate the predictions on the validation data. Let's assume we've chosen the accuracy or the misclassification rate as an evaluation metric. We would measure the misclassification rate of the predictions from the four models.

Then, we shuffle the observed data and create a different split. In this split we are using two-thirds of the data for training and the remaining one-third is held out for validation. It is the same dataset that we've used but the points that were in the training data in the first step might now be in the validation data and vice versa (**Figure 1-33**). However, none of the points will be included in both the training and validation data.

Figure 1-33. An alternative split.



We will repeat the process, training four different models corresponding to four different levels of regularization using the training data only and then apply those trained models to the validation data. We then measure the misclassification rate of each one of those models and repeat this process a third time.

We can now measure the average misclassification rate for each of the four models corresponding to the four levels of regularization and we will pick the level of regularization that corresponds to the lowest misclassification rate, hence the best model.

Challenges with Evaluation Criteria

Finally, there are potential challenges with the loss function. In the Methodologies Lesson of this module, we talked about how the loss function helps us pick the model that best fits our data from our model class. A challenge that we can encounter here is that our loss function can hide the true performance of our model. A challenge frequently seen in practice is that if we're developing a predictive model for a rare outcome, using a loss function that does not account for this rarity can give us a misleading view of the model performance.

In the model in **Figure 1-34**, there were two patients that had the disease in the observed data but the model predicts that none of the patients has the disease.

Figure 1-34. Observed labels and predicted labels

The model would have a misclassification rate of 0.1 which might make it look like a great model. But in reality, this model is simply predicting the same thing for all patients. In situations like this it might make more sense to look at the sensitivity and specificity of the model. We will discuss sensitivity and specificity in detail in Module 2.

Lesson 5 Knowledge Check

Refer to the Gulshan et al. (2016) Diabetic Retinopathy Study _JAMA to answer the following questions.

1. What machine learning methodology did the investigators use to develop the model in this study?
    a. Supervised Learning
    b. Unsupervised Learning
    c. Reinforcement Learning
    d. Semi-supervised learning

2. Match the datasets used in the study with the corresponding description of the dataset.

Study Dataset	Type of Dataset
Development Dataset	Test Set
EyePACS-1 Dataset	Tuning Set
Messidor-2 Dataset	Training and Validation Sets
	Validation Set

3. What are the inputs and outputs used to train this model?

Review of Key Points

- Big data, more powerful computers, and advanced computational methods have led to growth of machine learning in health care.

- Artificial intelligence and machine learning are proliferating in health care, and it has demonstrated potential to augment diagnostic decision making.

- Supervised learning, unsupervised learning, and reinforcement learning are the most commonly used methods to train health care machine learning models.

- When developing a health care machine learning model data should be split into training, validation, and test sets.

Bibliography

Catalyst, N. E. J. M. (2018). Health care big data and the promise of value-based care. NEJM Catalyst, 4(1). https://catalyst.nejm.org/doi/full/10.1056/CAT.18.0290

Doshi-Velez, F., Ge, Y., & Kohane, I. (2014). Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. Pediatrics, 133(1), e54-e63. DOI: 10.1542/peds.2013-0819

Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017 Feb 2;542(7639):115-118. DOI: 10.1038/nature21056. Epub 2017 Jan 25. Erratum in: Nature. 2017 Jun 28;546(7660):686. PMID: 28117445; PMCID: PMC8382232.

Gargeya R, Leng T. Automated Identification of Diabetic Retinopathy Using Deep Learning. Ophthalmology. 2017 Jul;124(7):962-969. DOI: 10.1016/j.ophtha.2017.02.008. Epub 2017 Mar 27. PMID: 28359545.

Gulshan V, Peng L, Coram M, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA. 2016;316(22):2402–2410. DOI:10.1001/jama.2016.17216

Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., & Faisal, A. A. (2018). The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. Nature medicine, 24(11), 1716-1720. https://doi.org/10.1038/s41591-018-0213-5

Shen L, Margolies LR, Rothstein JH, Fluder E, McBride R, Sieh W. Deep Learning to Improve Breast Cancer Detection on Screening Mammography.

Sci Rep. 2019;9(1):12495. Published 2019 Aug 29. DOI:10.1038/s41598-019-48995-4

Sjoding MW, Dickson RP, Iwashyna TJ, Gay SE, Valley TS. Racial Bias in Pulse Oximetry Measurement. The New England Journal of Medicine. 2020 Dec;383(25):2477-2478. DOI: 10.1056/nejmc2029240. PMID: 33326721; PMCID: PMC7808260.