Stanford MEDICINE

Stanford | Center for Health Education

FUNDAMENTALS OF MACHINE LEARNING FOR HEALTHCARE

MODULE 2 - CONCEPTS AND PRINCIPLES OF MACHINE LEARNING IN HEALTHCARE PART 1

LEARNING OBJECTIVES

- Distinguish the machine learning subfield from other areas of artificial intelligence and computer science.
- Describe the model fitting procedure in the supervised learning setting and distinguish supervised learning from unsupervised learning in healthcare applications.
- Understand the difference between structured and unstructured data, as well as some of the commonly used methods to represent unstructured data.
- Become familiar with common machine learning approaches like regression, support vector machines, and decision trees and how they might apply to clinical problems.

MACHINE LEARNING TERMS, DEFINITIONS, AND JARGON

Definitions of Machine Learning

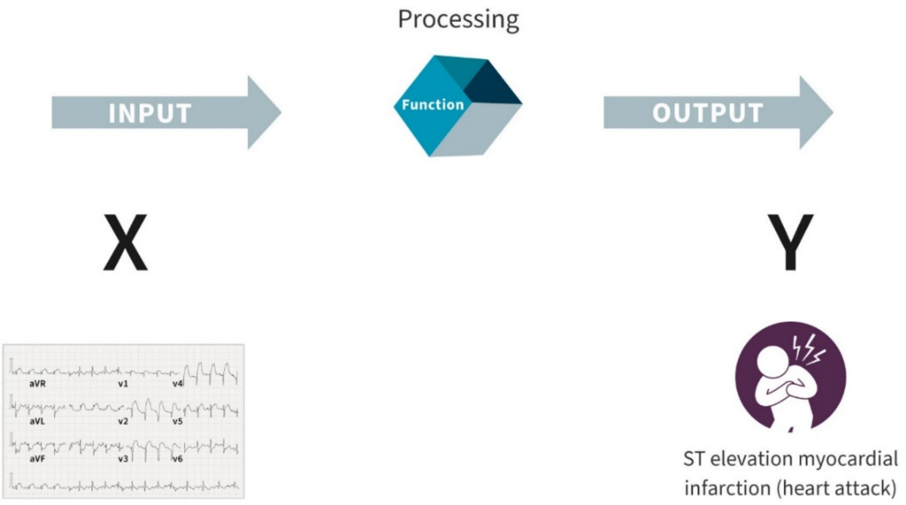
- Formal: A family of statistical and mathematical modeling techniques that uses a variety of approaches to automatically learn and improve the prediction of a target objective, without explicit programming
- Informal: Systems that improve their performance in a given task, through exposure to experience, or data

Three machine learning paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Machine learning problems fall along a spectrum of supervision between these terms.

Explaining computer programming:



- Boils down to three components: (1) the input, (2) some processing, and (3) The output
- Example:
  - Equation y = x^2. The input is x and an output y
  - Abnormality detection. The input could be an ECG and the output could be a medical diagnosis like ST elevation myocardial infarction

For both these examples, in between the input and the output, there is something that **processes the input to produce the output.**

- In example 1, it is squaring of the input x to arrive at the output y
- In example 2, there is a visual analysis being performed on the ECG leading to the output

“Processing”, we are referring to is the thing that transforms the input into the output, is called a **function**.

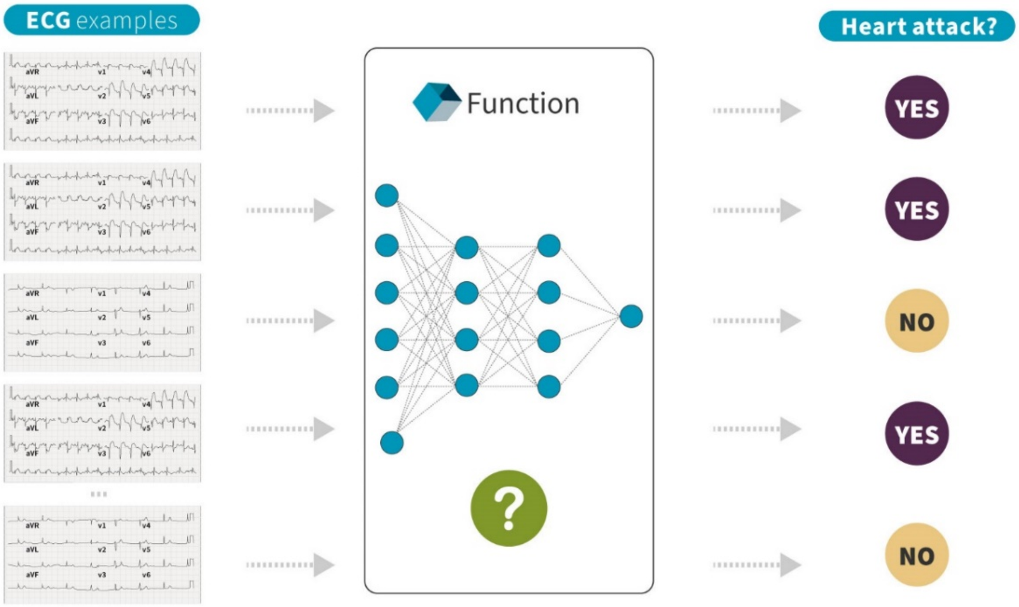
In a traditional computer programming approach, we deliberately write rules to process the inputs so that they produce the desired outputs. Traditional computer programming is also referred to as a “rules-based” approach.

In other words, computer programmers write functions with specific rules - the program written is the function, or the processing, that the computer performs to achieve an output.

Explaining Machine Learning, in particular **supervised learning**:

- The program written in this type of approach searches for (or in other terms, learns or finds) a function that can accurately map our data of inputs to outputs. We then use this function to process new inputs, and produce new outputs

If a cardiologist has already looked at the ECG (the input) and recorded a diagnosis, (the output) then we likely have two parts to the equation. All we need to do is figure out the function that solves the input-output.



Supervised learning is the process through which a program takes input-output pairs and 'learns' the function that links them together

Note that we call this 'supervised' learning because we provide input and output pairs. We are 'supervising' the model by providing it with the right answers.

- The **model**– the entity that undergoes supervised learning
  - It 'represents' or 'models' the relationships between the inputs and the outputs
  - Learning this relationship means learning a function which, in this case, means adjusting a set of numbers known as parameters
  - A model is defined entirely by its parameters and the operations between them
  - Sometimes called a model a **function approximator**– it approximates the function between the inputs and the outputs

Once the program learns a function that works well, we can use it in place of software that would have been written by traditional computer programming. We can take new inputs, put them through our learned function, and produce new outputs. This is the ultimate goal of supervised learning.

In supervised learning, as in traditional computer programming, a **program still has to be written**. However, the **purpose** of the program, to search for or learn an accurate mapping function instead of pre-specifying it, is fundamentally different.

Basic Terminology:

- Example: Single input-output pairs
- Features: Input. The part of an example that is fed into the model
- Labels: Output. The part of an example that is compared with the prediction of the model
- Dataset: A collection of examples
- Prediction: The output by a model that has learned from many examples of inputs-output examples and can now take a new input and give a new output

Dataset Terminology:

- **Training set**: A set of examples that the model is given in order to learn the function that links the inputs to the outputs
- **Validation set**: A set of examples that we hold out and do not expose the model to during training, and instead use it periodically to assess, or "validate", the generalization performance of our model, as we develop the model. We also use it to make meta-level design choices about **hyperparameters,** aspects of the program that trains the model
- **Test set**: A set of examples that we hold out until the very end of the model development process, to double-check the model's generalization performance on examples that are 'completely' unseen during any aspect of model development

Training loop: A repeated training procedure that allows the model several chances of learning good, generalizable functions from the training set

Training loop structure:

1. Start the program. The program sets up the training environment with a selection of hyperparameters, and initializes the model with a random function
2. Expose the model to examples from the training set, to learn a function from inputs to outputs
3. Evaluate how the function does on the validation set. If the model gets better performance than it ever has before, we save this version of the model
4. Repeat steps 2 and 3 until the performance on the validation set no longer goes up

Typically, we repeat for various hyperparameter settings. This is known as **hyperparameter tuning**. Different hyperparameters can produce different models.

Once we are satisfied with the model's performance on the validation set, we can run this final model on the **test set**.

Feature Types:

- **Structured data**: A patient's lab values, diagnosis codes, etc.– also commonly used with the more traditional statistical models. Structured data is commonly input into the model as a list (or vector) of numbers.
- **Unstructured data:** Images or natural language (text reports)
    - **Images** are typically represented as grids of numbers, where each number represents intensity at a given pixel location. In grayscale images, there is only one grid. In color images, there are three grids overlaid on top of each other; the Red, Green, and Blue grids.
    - **Texts** are typically represented with what are called **embeddings**. Word embeddings are geometric, numerical vector representations of words.

HOW MACHINES LEARN

Label types:

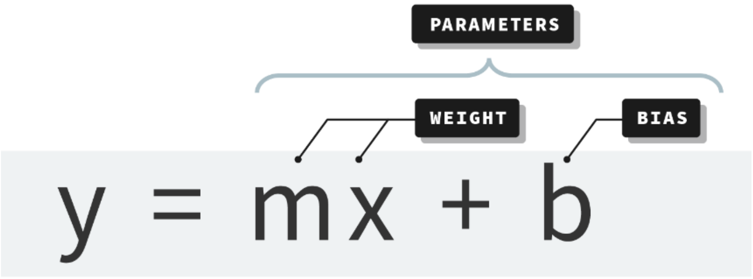
- Labels can be real numerical values. If a model predicts real numerical values, it is solving a **regression problem**
- Labels can be categories, or classes. In this case, labels are just numbers that act as category IDs. If a model predicts categories, it is solving a **classification problem**

Model training

- Mathematically, training minimizes the difference between the output of the model's function and the true label, for every sample in the training set
- **Loss:** The difference between the function output and the true label. Typically, we average or sum the loss for every data point that we have
    - If the model is poorly trained, then will have high loss
    - If it is well trained, then the difference between the true label and our function will be small on average, and thus resulting loss is small as well

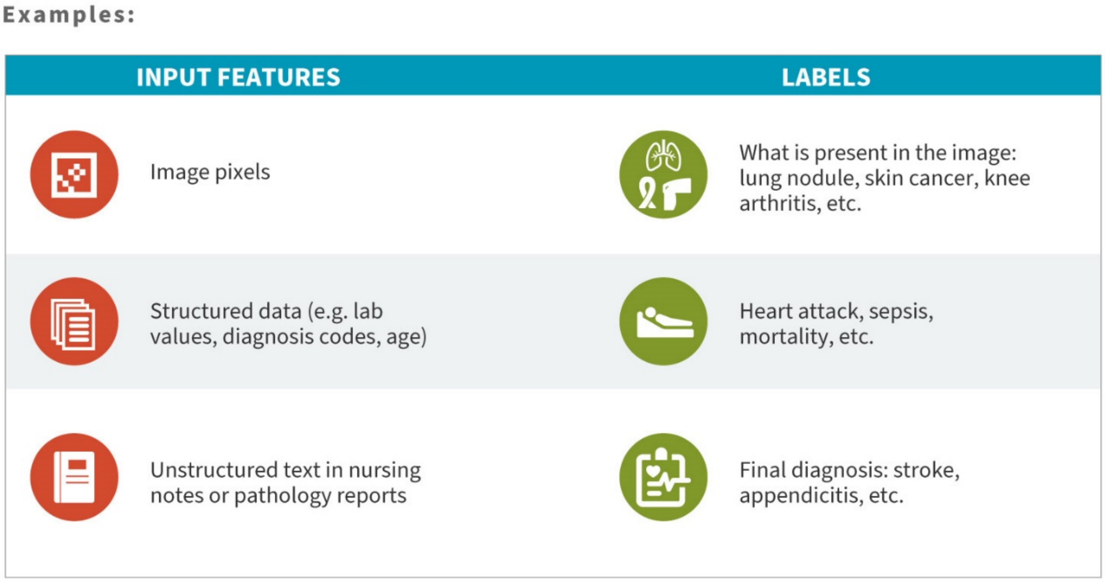
The model updates its function to map inputs to labels, as accurately as possible. This is known as **fitting** or **training** the model.

- The model updates its function by adjusting its parameters. Parameters are numerical values that are used to perform operations on the input data.



- Example: Linear regression. The parameters are the numerical coefficients **m** and **b**, as in the equation **y=mx + b**
  - Parameters that multiply features as **weights**
  - Parameters that are added to the features as **biases**
  - It is also common practice to call all parameters– both weights and biases– “weights”
- Bias (the parameter) vs. Bias (the phenomenon)
  - Bias, the parameter: a number added to features
  - Bias, the phenomenon: a concept that relates to model performance and algorithmic fairness

Classification: predicting categorical labels



Examples:

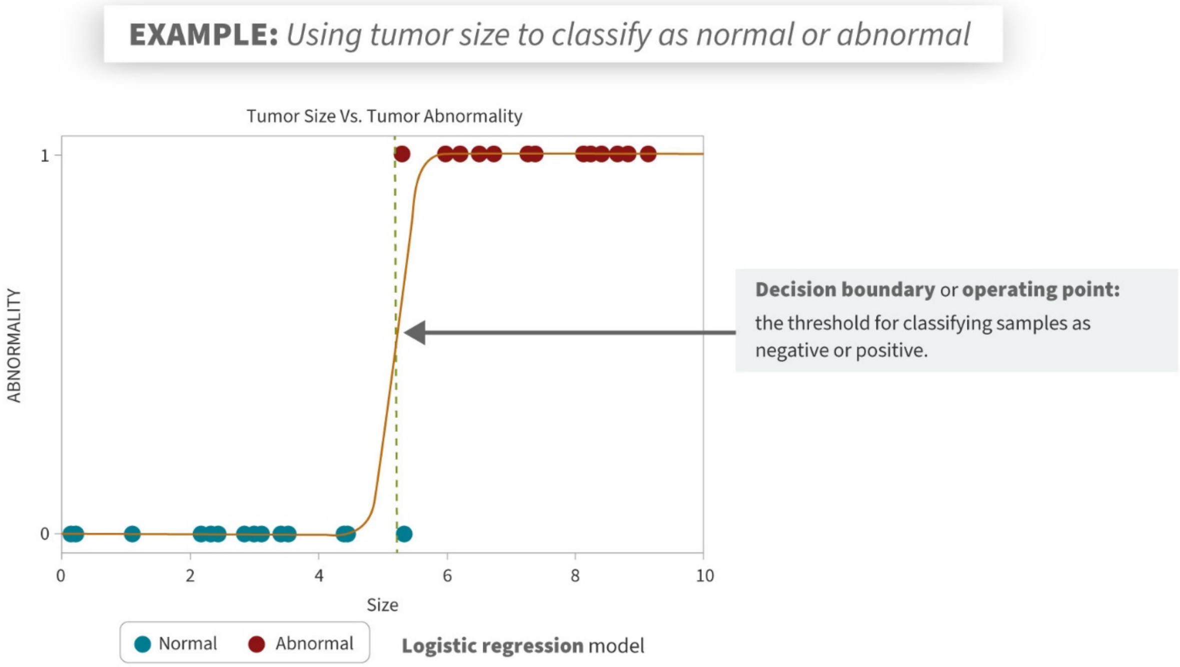
INPUT FEATURES	LABELS
Image pixels	What is present in the image: lung nodule, skin cancer, knee arthritis, etc.
Structured data (e.g. lab values, diagnosis codes, age)	Heart attack, sepsis, mortality, etc.
Unstructured text in nursing notes or pathology reports	Final diagnosis: stroke, appendicitis, etc.

Difference between classification and regression:

- Labels are categorical
- Classification models output probabilities. A model prediction in a classification task is a probability that a given set of features belongs in each category
- Probabilities are produced using the sigmoid function
- Logistic regression is a model type commonly used for classification

Decision boundary (or operating point)

- The probability number that we use as our cutoff between categories
- Commonly the decision boundary is the 50-50 mark


EXAMPLE: *Using tumor size to classify as normal or abnormal*

Decision boundary or **operating point:** the threshold for classifying samples as negative or positive.

Logistic regression model

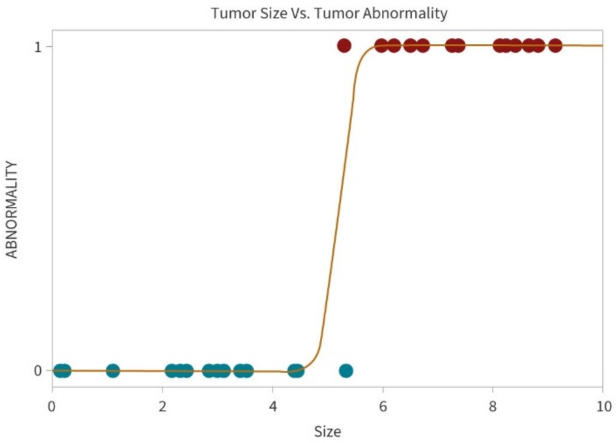
- Depending on the use case the operating point can be moved
  - Example: screening test in healthcare, perhaps false positives are acceptable and false negatives are not. The operating point can be adjusted so that more examples are classified as positive.

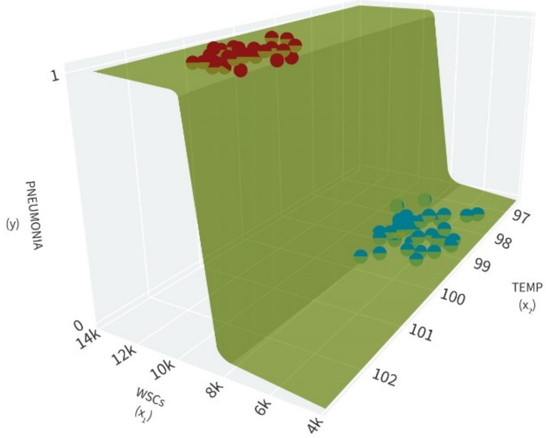
The multiple-feature setting is similar to the one-feature setting.

Consider the classification setting:

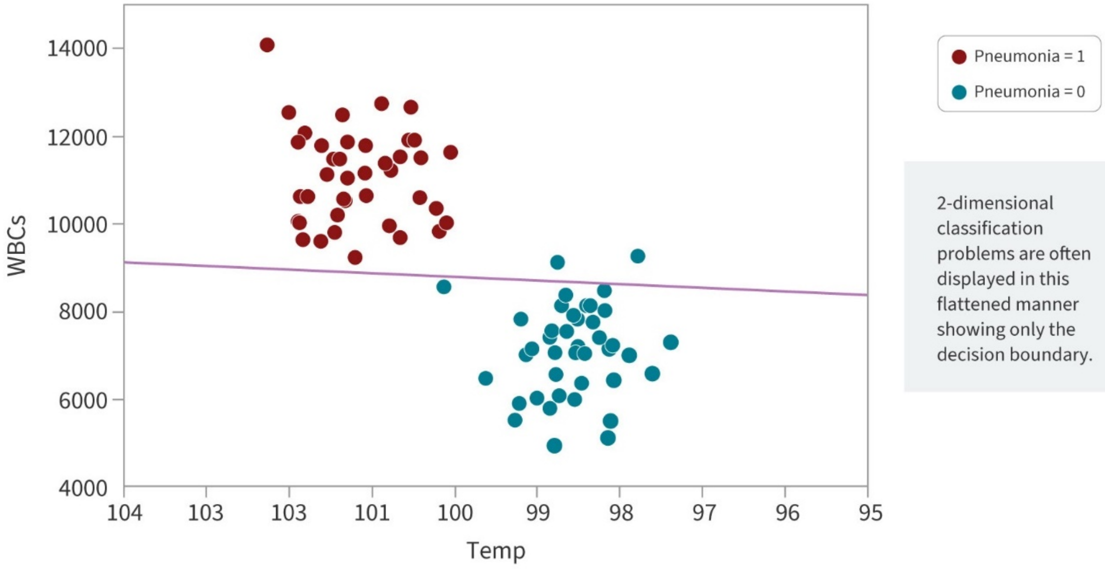
- In both the one-feature and two-feature cases, we can visualize the label as a separate dimension whose values separate the categories

- The model still has to learn a function with a decision boundary that separates data points with different labels (also called “y’s”)
- However, in the two-feature case, we need to adjust parameter values corresponding to both features, x1 and x2, to find a good model fit. Recall that the parameter values are the model weights, which are multiplied with the features





- Drawing the decision boundary is similar as well. Recall that a common decision boundary is where the function outputs 0.5, as in there is a 50-50 chance that the output for a sample sitting on the decision boundary is 1 vs 0.
  - Since y in this case can only take two values, we can flatten this entire figure to two dimensions and mark the y's only through color, in other words a point whose y is 1 is red, and a point whose y is 0 is blue. We can also demarcate the decision boundary by drawing a line everywhere our function equals 0.5.



2-dimensional classification problems are often displayed in this flattened manner showing only the decision boundary.

The two-feature setting is a straightforward extension of the 1-feature setting. While it is more difficult to visualize, the same idea holds for **any number of features**. For binary classification, the geometric intuition is the same: **find a function whose decision boundary sits between the two sets of samples**.

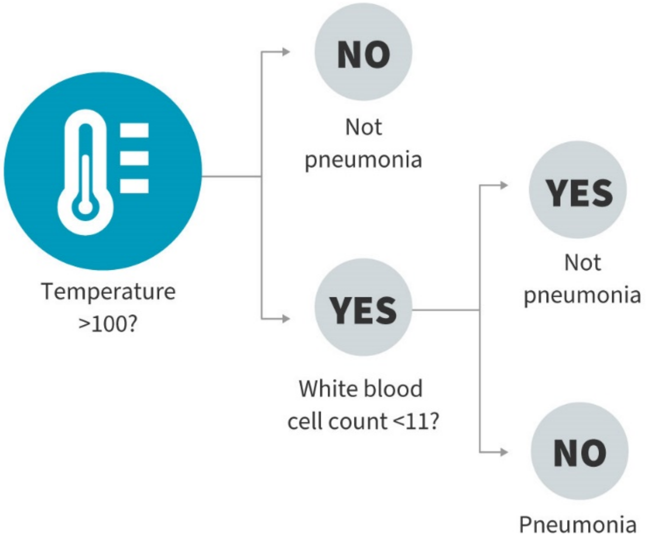
SUPERVISED MACHINE LEARNING

The "No Free Lunch" theorem: No one Machine Learning algorithm is best for all problems

- Regression variant: Polynomial Regression
  - Useful for handling non-linearly separable data where the best fit line is not a straight line
  - It fits a curve instead of a line
- Other common regression variants
  - Examples: Lasso Regression, Ridge Regression, ElasticNet Regression
  - At their core, they are all functions that can act as a classifier and can be adjusted to better fit the relationships between the features to predict the correct label.

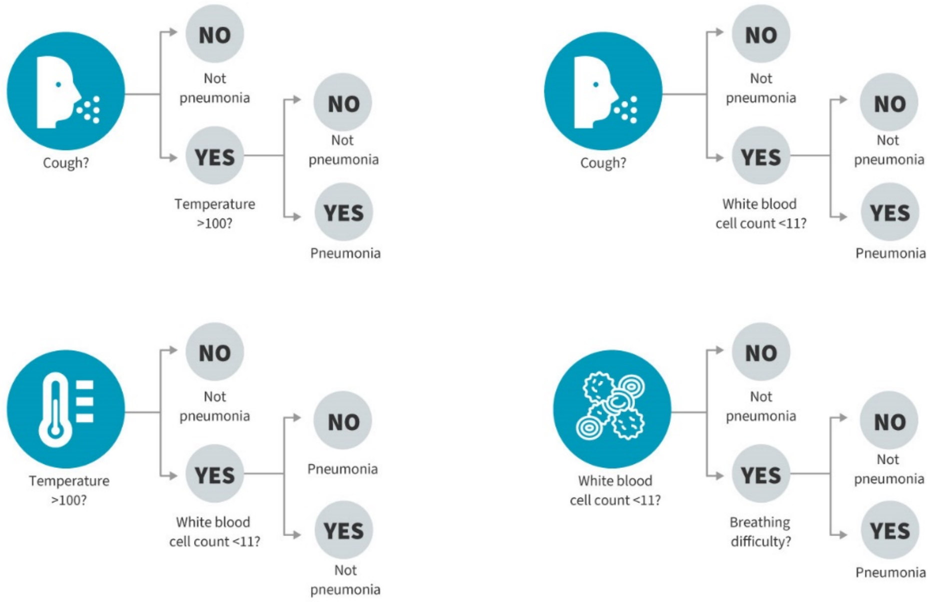
TRADITIONAL MACHINE LEARNING

Decision tree algorithms, specifically classification tree algorithms, generate a tree structure where the branching points are decision points that are based on the relationships between features found in the training dataset.



- Advantages:
  - They can be very fast to train with high dimensional datasets
  - They are simple to understand and interpret– every branch in the tree is a decision point based on some relationship between the features
- Disadvantages:
  - They can sometimes be inaccurate. This effect can be mitigated by using decision tree variants such as Random Forests

Random forests improve decision trees by ensembling, or combining, the predictions of many, many decision trees. Typically, decision trees are trained on all features using all samples in the training dataset. Each of the decision trees in a random forest algorithm only (1) sees a subset of the features made available for each sample and (2) sees a subset of the samples in the training dataset.

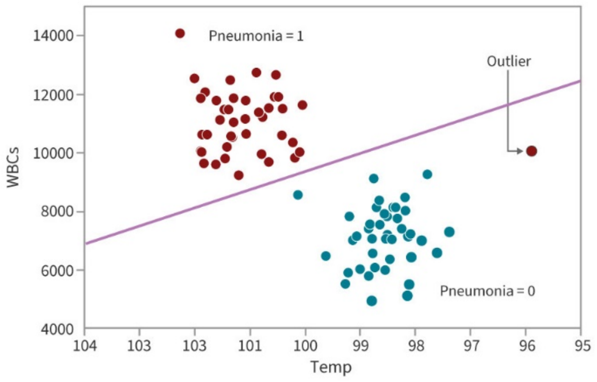


- Advantages:
  - The diversification of decision trees may produce some bad classifiers, but many other trees will be right. So, as a group, the trees are able to classify things more correctly than any single decision tree.
- Disadvantages:
  - Slower to train than decision trees.

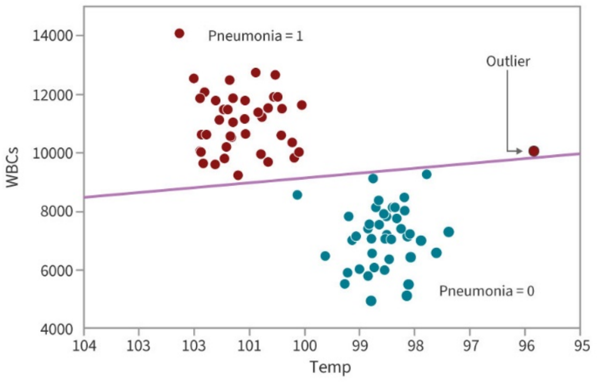
Support Vector Machine (SVM) is another supervised learning machine learning algorithm used for classification problems similar to Logistic Regression (LR).

- Logistic Regression considers all samples equally

- SVMs consider samples that are near the decision boundary more strongly than Logistic Regression, which in turn makes SVMs more robust to outliers



SVM



Logistic Regression

A large dataset with no labels at all and no feasible way to label them on your own

- **Unsupervised learning** seeks to examine a collection of unlabeled examples and group them by some notion of shared commonality.
- **Clustering** is one of the common unsupervised learning tasks. We can use clustering to define the label or category

In unsupervised learning the difficulty lies not in obtaining the grouping, but in evaluating it or determining whether the grouping that is found is actually meaningful.

The challenge, then, is whether the presence of the groups (i.e., clusters) or learning that a new patient is deemed a member of a certain group is actionable in the form of offering different treatment options or making a clinical decision.

CITATIONS AND ADDITIONAL READINGS

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning: data mining, inference, and prediction.* New York: Springer.

Matheny, M., Israni, S. T., Ahmed, M., & Whicher, D. (2020). Artificial intelligence in health care: The hope, the hype, the promise, the peril. *Natl Acad Med*, 94-97. https://nam.edu/artificial-intelligence-special-publication/

Shah, S. J., Katz, D. H., Selvaraj, S., Burke, M. A., Yancy, C. W., Gheorghiade, M., Bonow, R. O., Huang, C. C., & Deo, R. C. (2015). Phenomapping for novel classification of heart failure with preserved ejection fraction. *Circulation*, *131*(3), 269–279. https://www.ncbi.nlm.nih.gov/pubmed/25398313

Shah, S. J., Katz, D. H., Selvaraj, S., Burke, M. A., Yancy, C. W., Gheorghiade, M., Bonow, R. O., Huang, C. & Deo, R. C. (2015). Phenomapping for novel classification of heart failure with preserved ejection fraction. *Circulation*, *131*(3), 269-279. https://www.ahajournals.org/doi/10.1161/circulationaha.114.010637