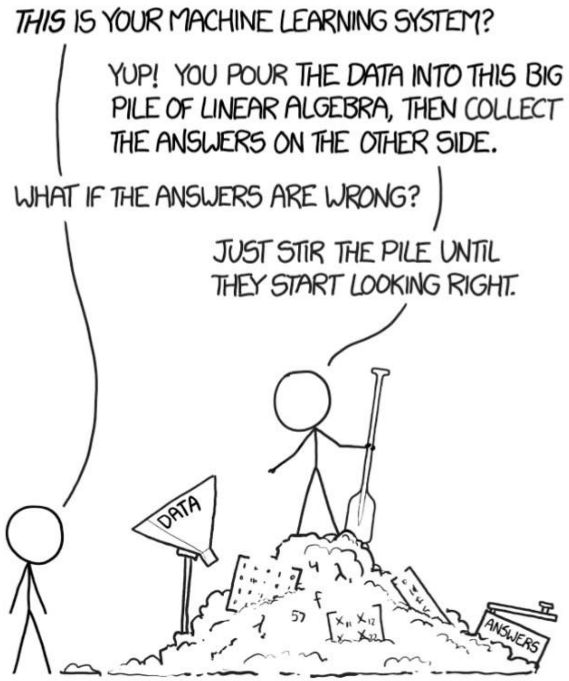
Crash Course on Machine Learning

Alex Mariakakis

University of Toronto

Department of Computer Science



THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Outline

What is covered

- High-level terms and definitions
- Conceptual walkthrough of how to approach machine learning problems
- Supervised machine learning on simple tabular data

What is not covered

- Underlying math
- Machine learning on sequences, images, and text
- Unsupervised learning
- Deep learning
- Code examples
    - Scikit-learn for Python
    - Weka for Java

What Is Machine Learning?

Machine learning takes some example data to train a model

It then learns patterns from those examples so that it can produce results for future data

This code (not the patterns) can often be re-used across problems

```
if email contains “Viagra”:
    then mark as spam;
if email contains “limited offer”:
    then mark as spam;
...
```

“Traditional” Programs

```
while accuracy low:
    classify some emails;
    check errors;
    change to reduce errors;
```

Machine Learning Programs

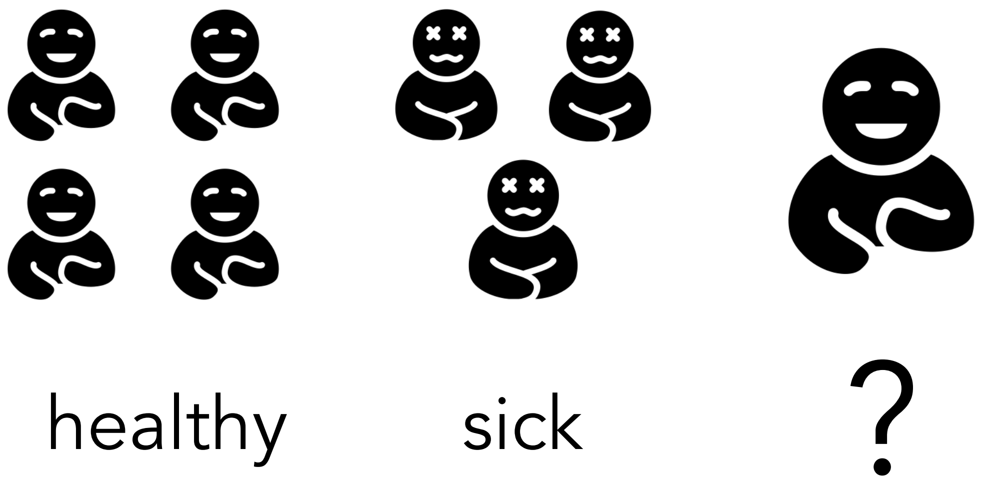
How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Types of Learning

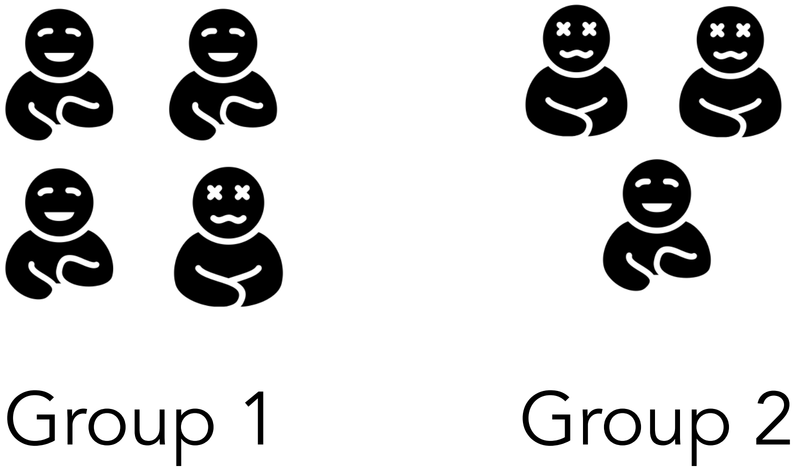
Supervised Learning

The goal is for the model to produce the desired output based on examples with labels (i.e., expected outcomes)



Unsupervised Learning

The goal is for the model to extract meaningful trends or associations in the data without having access to labels (or assuming they do not exist)

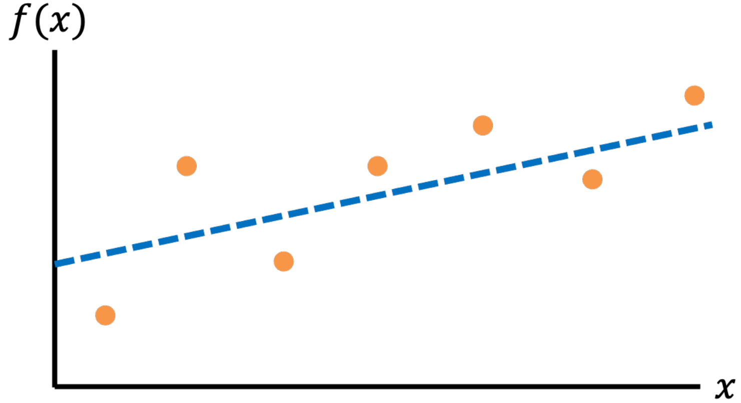


Types of Supervised Learning

Regression

Questions that can be answered with a continuous output

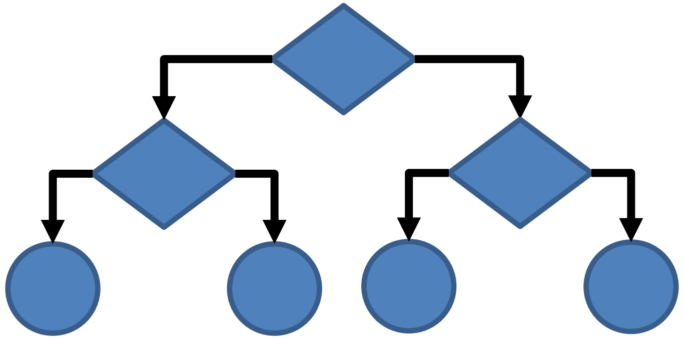
- What will the patient's heart rate be in one hour?
- For how long will the patient experience symptoms?



Classification

Questions that can be answered with a categorical answer

- Is this person sick or not?
- Is this person going to become sick in the future?

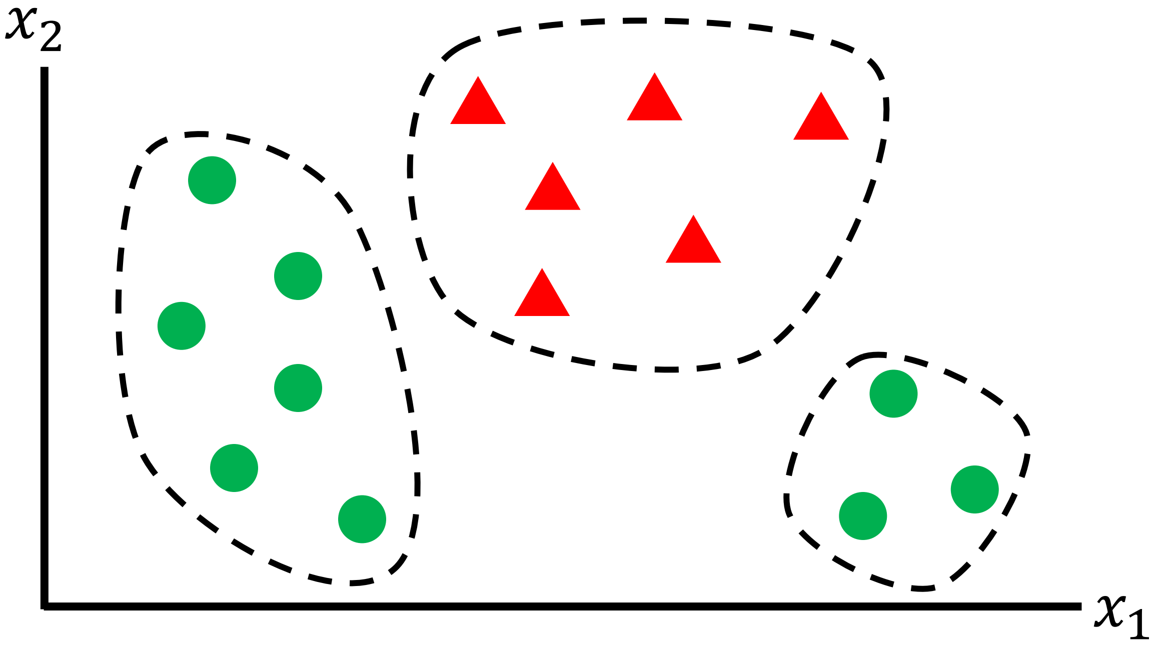


Example of Unsupervised Learning

Clustering

Segregates the data into groups such that samples in the same group are more similar to other samples in the same group and dissimilar to samples in other groups

Since the clusters do not depend on labels, the groups may or may not be related to meaningful distinctions in the data



What We Will Cover

We are going to focus on supervised learning, which means that we can assume that are data comes with labels that represent the expected outcome for each data sample

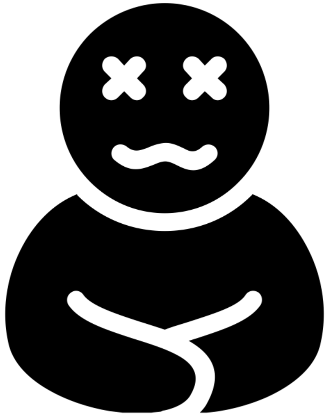
How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Feature Representations

There are millions of ways to represent data!

What differentiates sick people from healthy people?

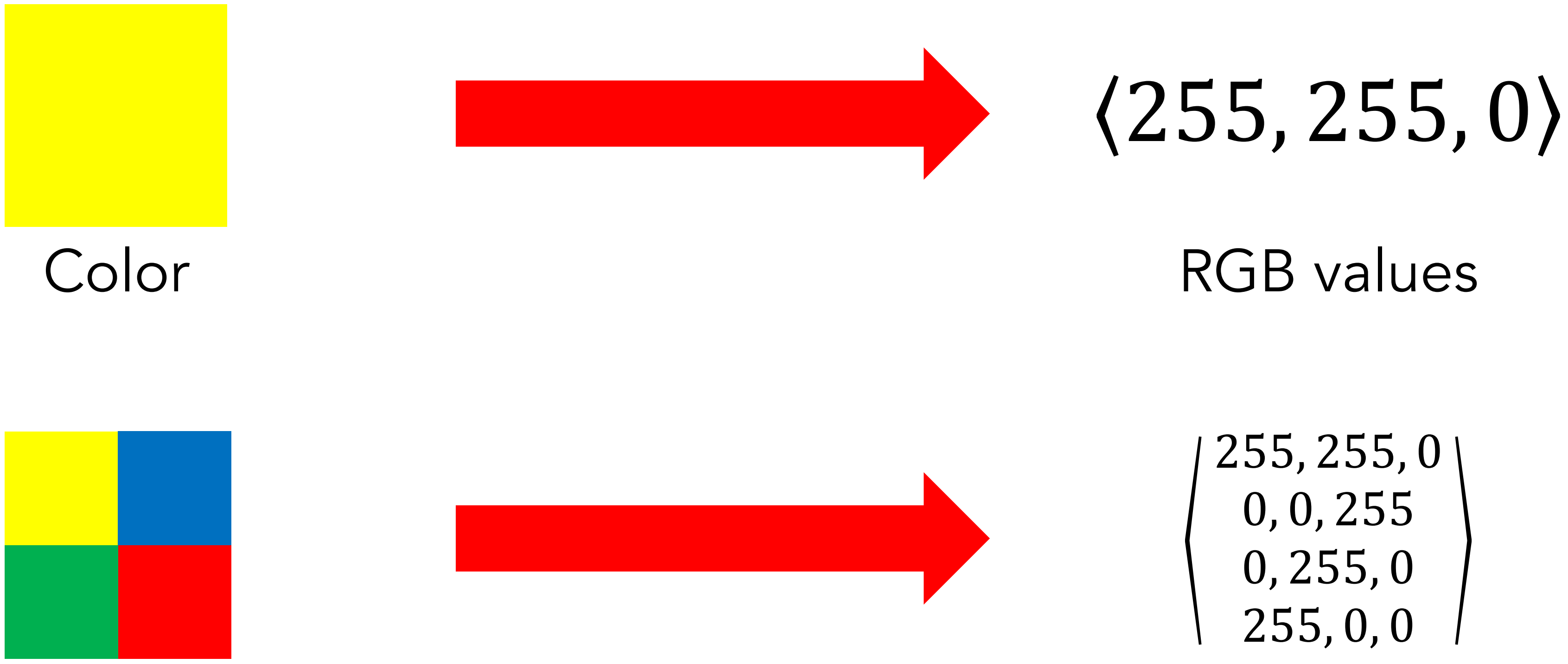


Resting heart rate = 60 bpm
SpO_2 = 97%
Cough status = normal

Resting heart rate = 80 bpm
SpO_2 = 92%
Cough status = worsening

Feature Representations: Images

There are millions of ways to represent data!



Color

$$\langle 255, 255, 0 \rangle$$

RGB values

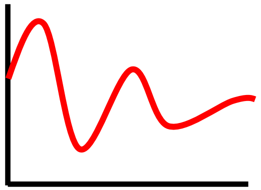
Image

$$\left\langle \begin{matrix} 255, 255, 0 \\ 0, 0, 255 \\ 0, 255, 0 \\ 255, 0, 0 \end{matrix} \right\rangle$$

Group of RGB values

Feature Representations: Sequences

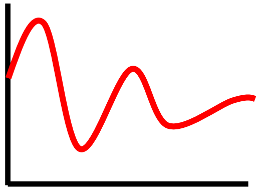
There are millions of ways to represent data!



Time-varying signal
(e.g., audio)

$\langle 10, 12, 8, 5, 3, 5, \ldots \rangle$

Series of a fixed
number of samples



Time-varying signal
(e.g., audio)

Mean = 5
Average slope = -0.1
Kurtosis = 0.1

Summary features that
describe the signal

What Does Our Data Look Like

Subject ID	Age	Heart Rate	Systolic Blood Pressure	Diastolic Blood Pressure	SpO_2	Has Sepsis?
P01	34	77	117	72	100.0	No
P02	78	87	163	79	95.9	Yes
P03	57	66	144	81	97.6	No
P04	23	93	167	94	94.0	Yes
P05	81	90	123	86	99.7	Yes

Samples

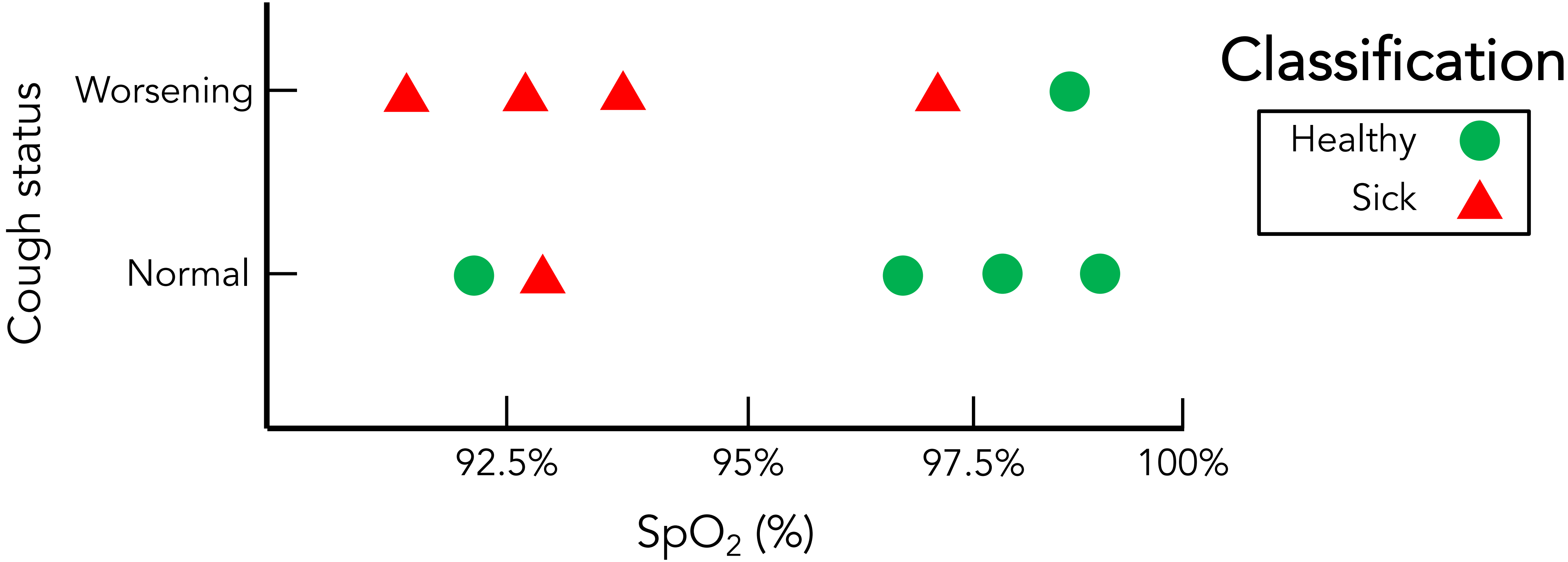
Identifying information that should not be used for learning

Features

Labels

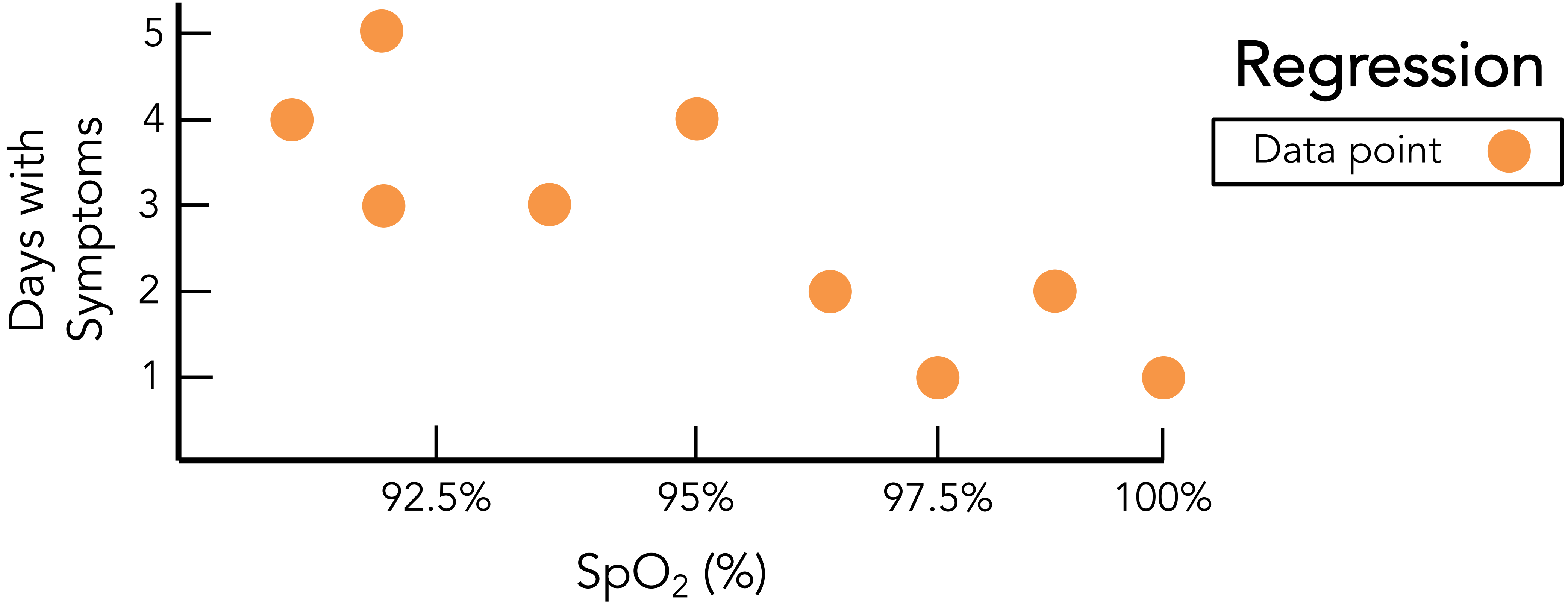
What Does Our Data Look Like

Each feature can be thought of as a dimension of a new coordinate system that represents our data



What Does Our Data Look Like

Each feature can be thought of as a dimension of a new coordinate system that represents our data



Dealing with Missing Data

It is easiest to work with data when all the entries in your dataset have the same number of features

You can fill in blanks using imputation techniques:

Technique	Note
Placeholder value (e.g., 0, -999)	Not a good idea for models that cannot distinguish between real and fake data
Mean / median	Pretty limited in the level of accuracy that can be achieved since it only cares about one feature at a time
Nearest neighbor	Requires training and tuning a special model for filling in data based on other examples

How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Dataset Splitting

When you evaluate your model, it is important that you see how well it works on data it has <u>never</u> seen before

To do this, we often partition our dataset into distinct “splits”

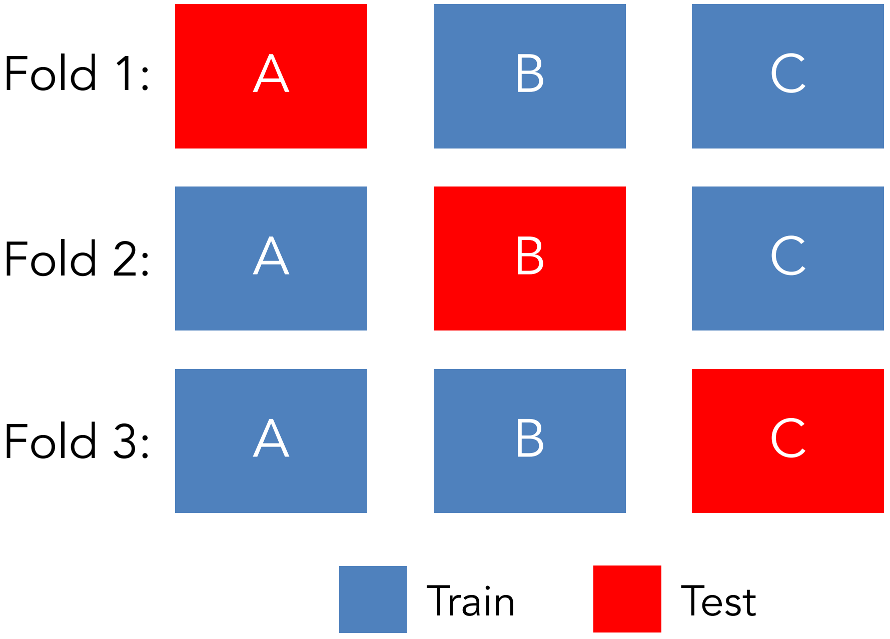
1. **Train:** Used to help the model learn how to set its internal parameters so that it can generate predictions
2. **(Optional) Validation:** Used to tune hyperparameters (more on that later)
3. **Test:** Used to evaluate the trained model’s performance

Standard Methods of Dataset Splitting

The amount of training data should be significantly higher than the amount of validation and/or testing data

These are three standard methods without a validation split (which can often be just taken from the train set)

1. **Single Split:** Generate a single partition (e.g., 80% train & 20% test)
2. **N-Fold Cross-Validation:** Split the data into N equal partitions, train one model with each partition as the test set, and then average the results
3. **Leave-One-Out Cross-Validation:** Similar to N-Fold cross-validation, but when the splits correspond to an important aspect of the dataset (e.g., N = number of users)



Fold 1:	🟥 A	🟦 B	🟦 C
Fold 2:	🟦 A	🟥 B	🟦 C
Fold 3:	🟦 A	🟦 B	🟥 C

🟦 Train 🟥 Test

Considerations for Dataset Splitting

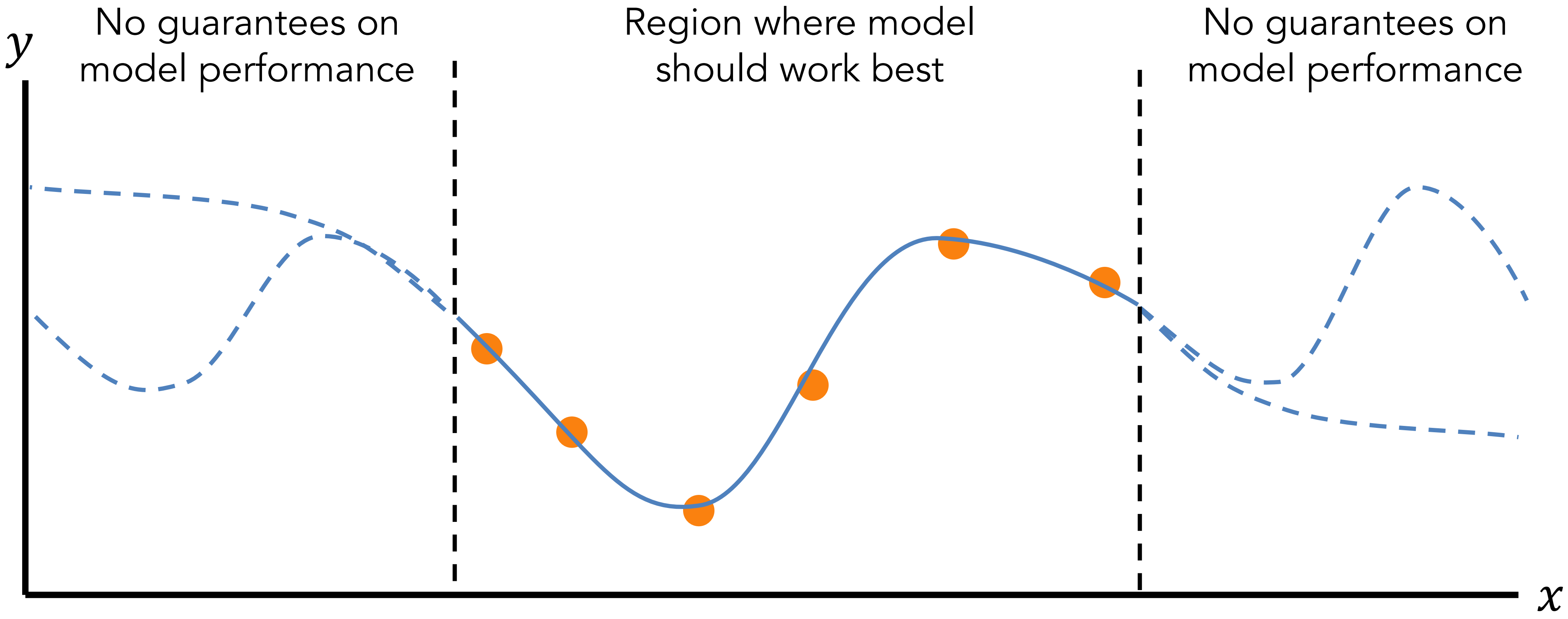
It is usually good practice to shuffle the rows of your data so that there are not any biases based on how the data was recorded or grouped

Make sure you have good representation of all possible labels in each of your data splits

- If your train split only includes healthy patients and your test split only includes sick patients, then you will get bad performance
- This problem becomes more apparent as your dataset and splits become smaller

Considerations for Dataset Splitting

If your model has not been trained on a certain kind of data before, it probably will not predict well on that data either



Considerations for Dataset Splitting

Pick the right training method to answer your question

- If you want to show that your approach works on patients that have never been seen by the model, then you should make sure that no individuals are represented in both the train and test sets
- If you think that personalized models are needed for each person, then you should split each patient’s data into train and test sets and build individual models

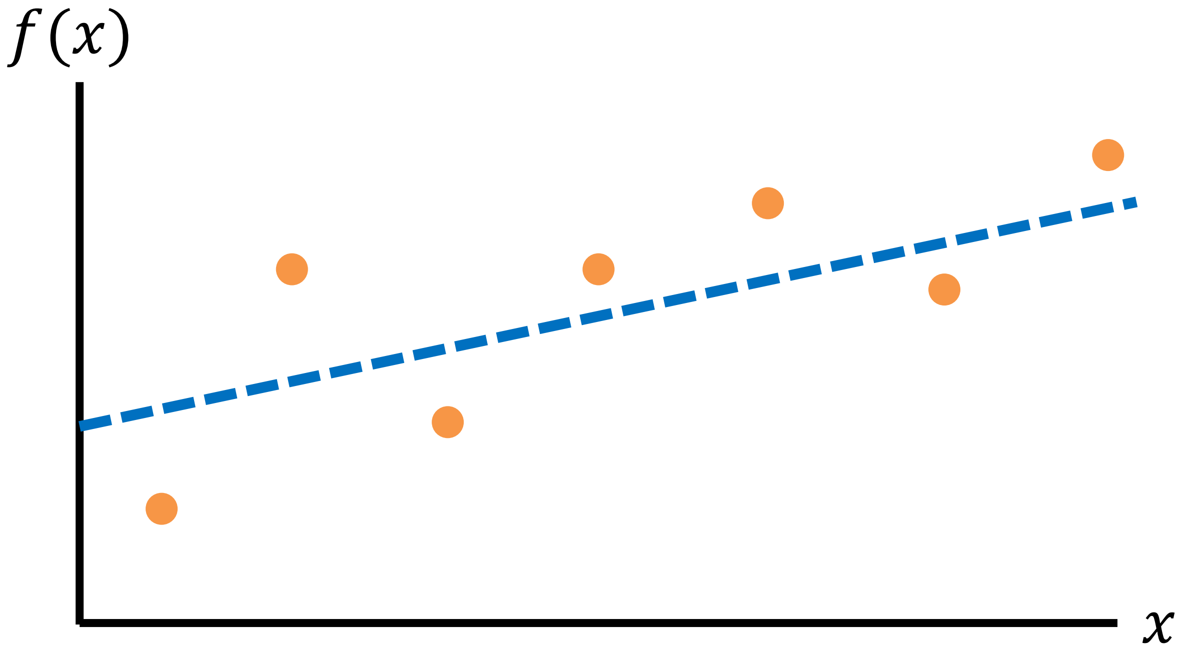
How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Regression: Linear Regression

Maps a linear combination of features to a continuous label $(-\infty, \infty)$

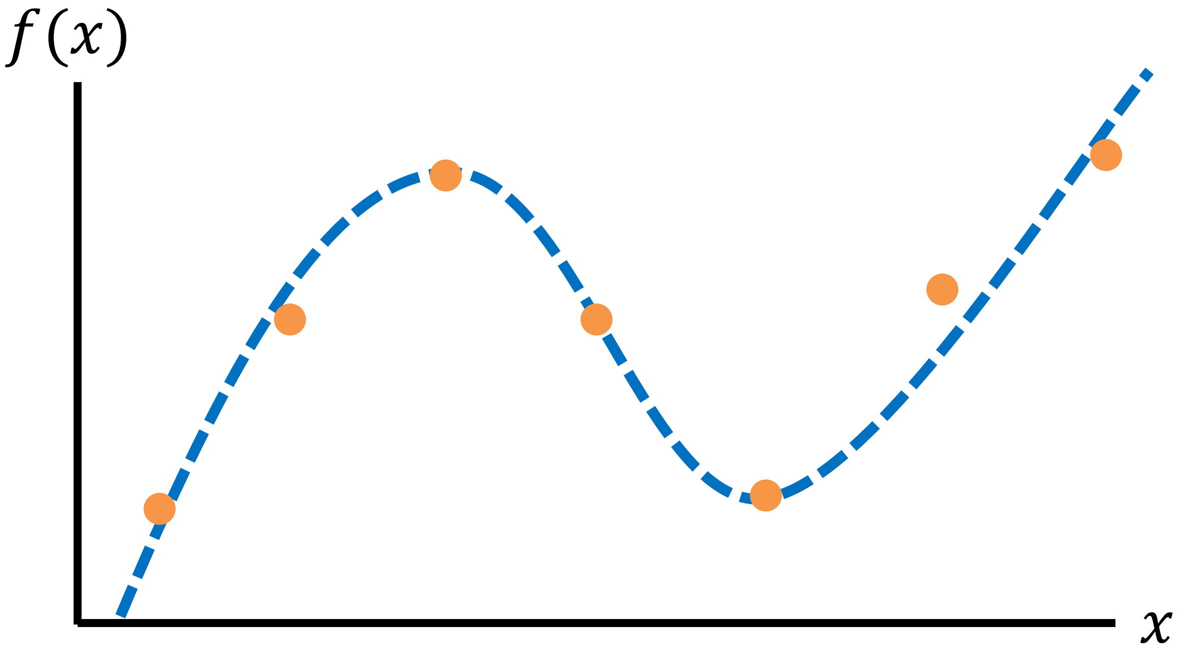
$$y = w_1 x_1 + w_2 x_2 + \cdots + b$$



Regression: Polynomial Regression

Maps a linear combination of powers of features to a continuous label $(-\infty, \infty)$

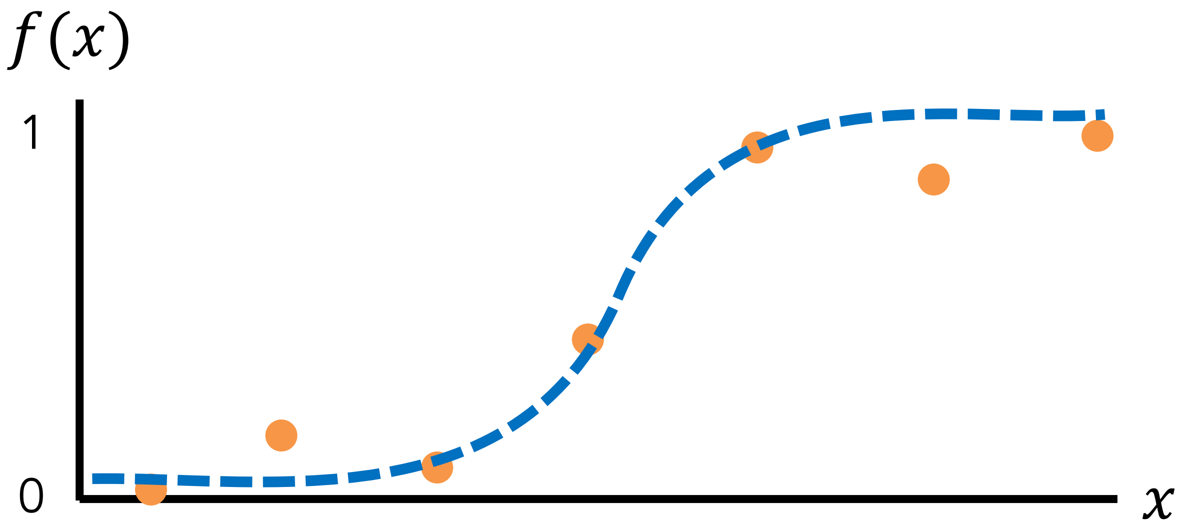
$$y = w_1 x^n + w_2 x^{n-1} + \cdots + b$$



Regression: Logistic Regression

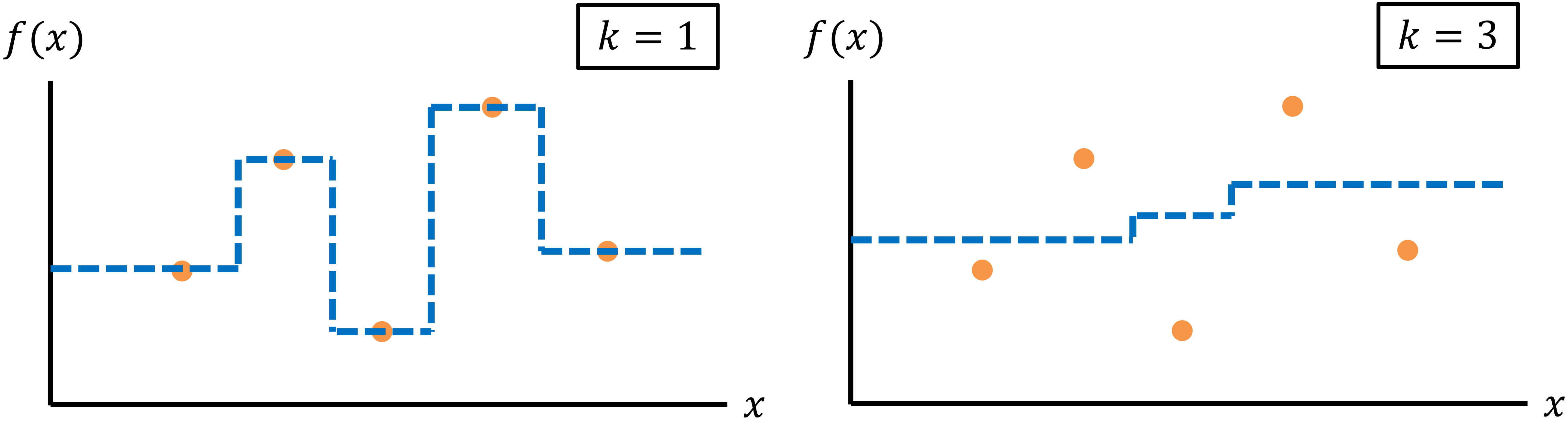
Maps a logit function on a combination of features to a continuous label $(0, 1)$

$$y = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + \cdots + b\ )}}$$



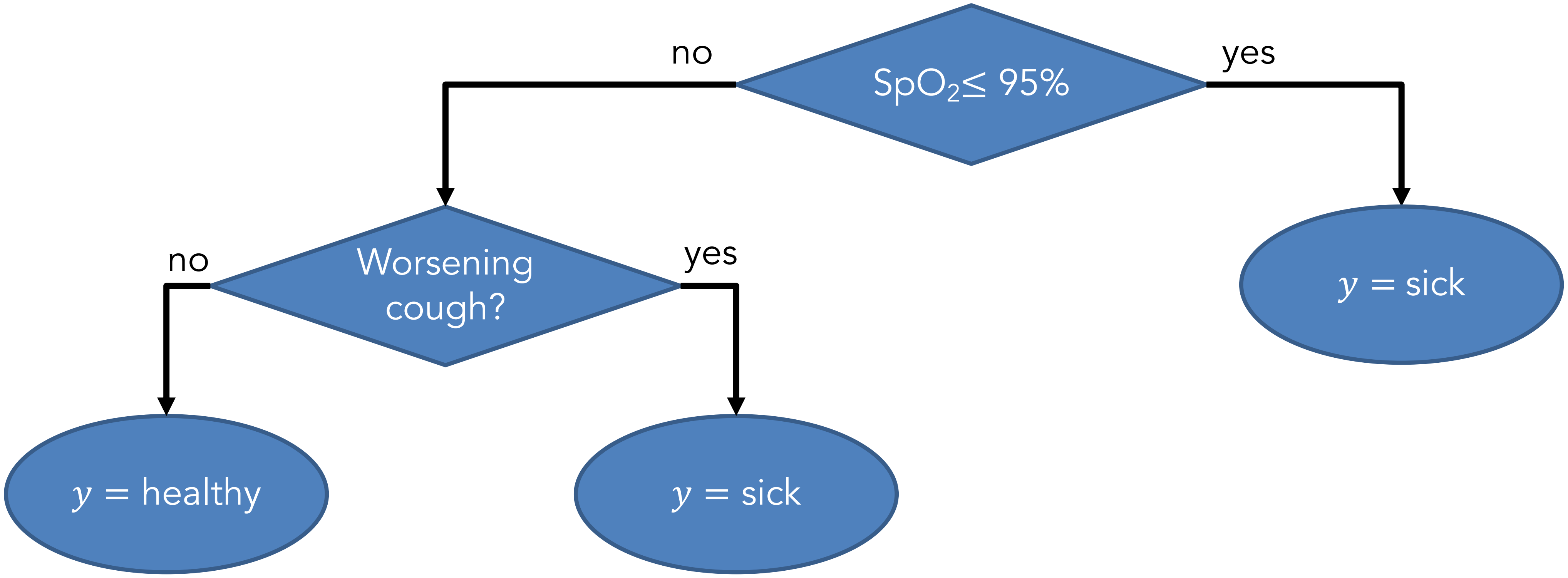
Regression: K-Nearest Neighbor

Finds the k training examples that are most similar and computes the new result based on their labels (e.g., average, median)



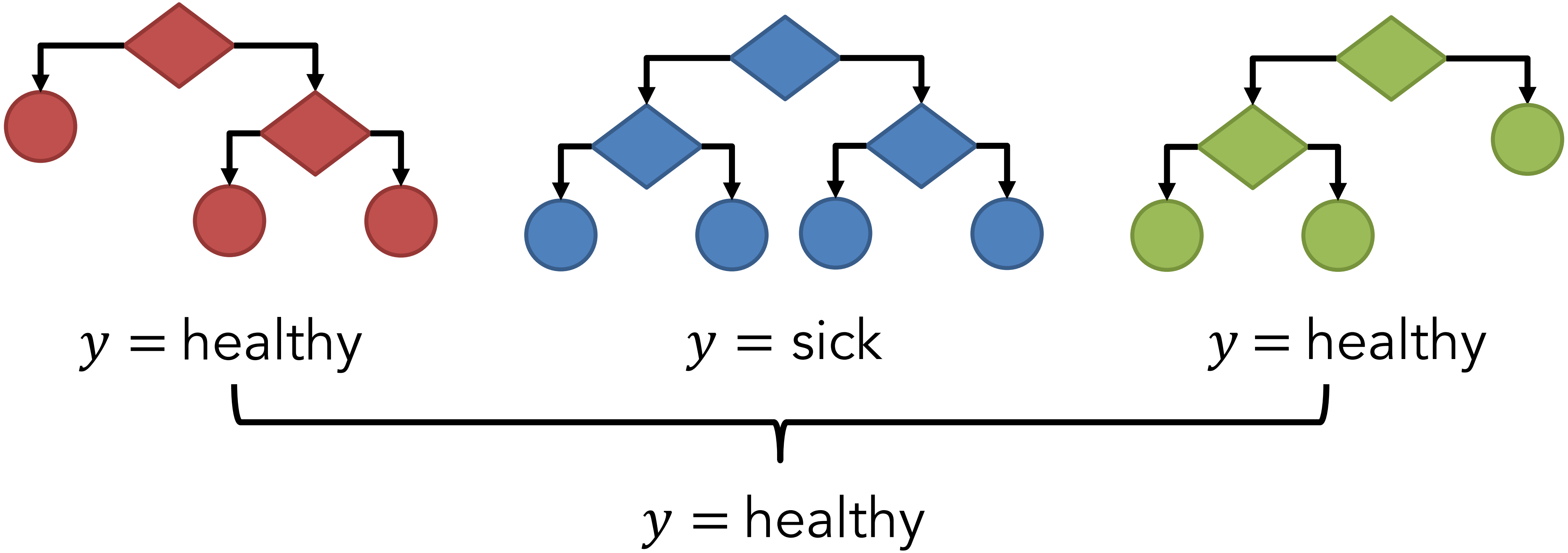
Classification: Decision Tree

Maps a combination of features to a class label based on a series of binary decisions



Classification: Random Forest

Uses a collection of decision trees (forest = multiple trees) to make a decision



$y =$ healthy

$y =$ sick

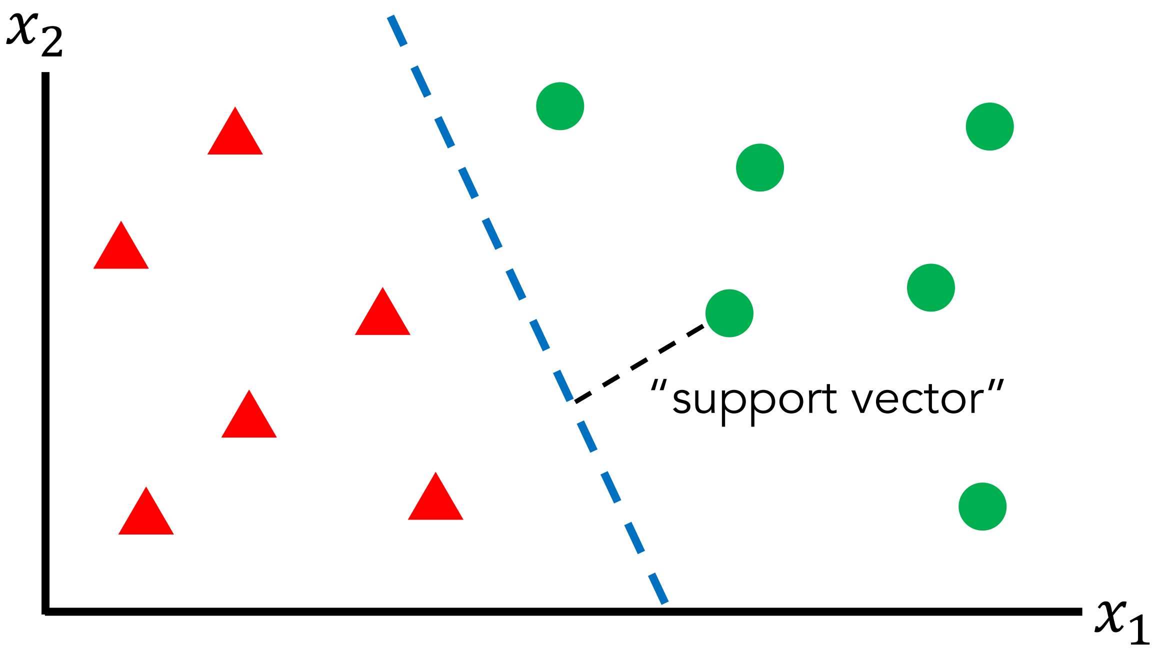
$y =$ healthy

$y =$ healthy

Classification: Support Vector Machine (SVM)

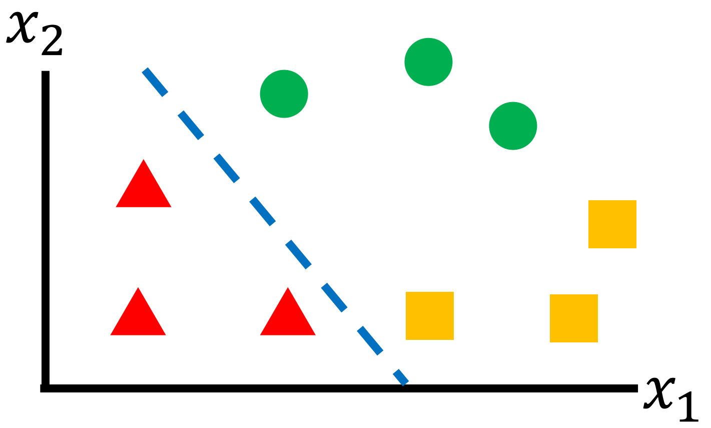
Separates two different classes with a hyperplane that maximizes the separation between the classes

$$y \geq \; w_1x_1 + w_2x_2 + \cdots + b$$



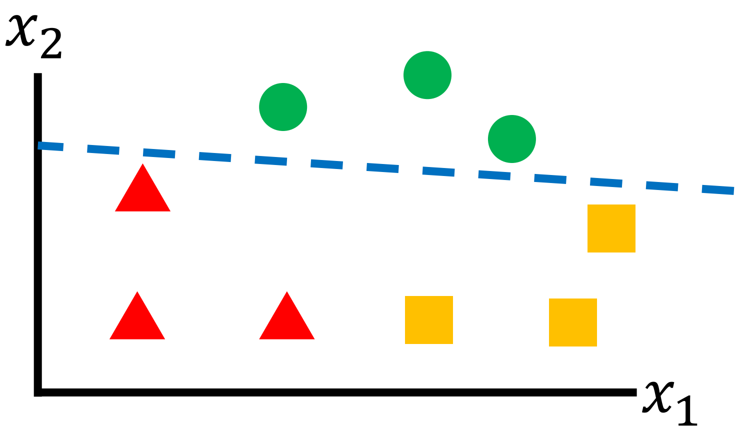
Classification: Support Vector Machine (SVM)

Can separate between multiple classes by combining the results of multiple one-versus-rest classifiers



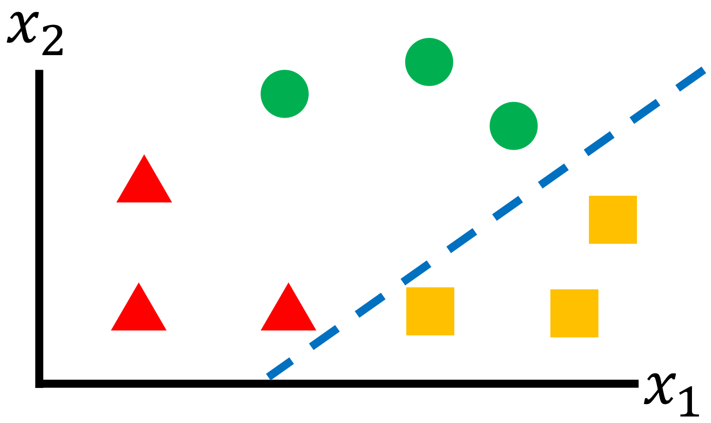
Hypertensive vs. non-hypertensive

$y =$ non-hypertensive



healthy vs. non-healthy

$y =$ healthy



Hypotensive vs. non-hypotensive

$y =$ non-hypotensive

$y =$ healthy

Regression ⟷ Classification

Models for classification often have an analogue for doing regression

Classification	Regression
<u>Support Vector Machine (SVM)</u> Classifies based on which side of the decision boundary the input is on	<u>Support Vector Regression (SVR)</u> Regresses based on how far away the input is from the decision boundary
<u>Decision Tree</u> Leaf nodes correspond to classes	<u>Decision Tree Regression</u> Leaf nodes correspond to continuous values

Regression ↔ Classification

Regression can be changed to classification with thresholds

- Assuming logistic regression is trained to predict the probability that a person is sick, you may label the patient as sick if the regression output is ≥ 0.5
- The patient should be considered sick and stay in hospital if the predicted number of future days with symptoms is ≥ 3

Regression ↔ Classification

Regression gives more information

- Imagine if one patient is predicted to have 4 future days of symptoms and another patient is predicted to have 8 future days of symptoms
- Both would be considered “sick” and may be asked to stay in the hospital, but I may want to prioritize treating the latter

Classification may be “easier” in some cases since there is less room for making mistakes

- Telling the difference between 1…2…3… future days of symptoms may be harder than telling the difference between sick and healthy patients

How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Hyperparameters

Hyperparameters are tunable parameters that are set before you train your model to control how it learns

We can learn the best settings for these values by doing a separate round of training with a validation split

Model	Examples of Hyperparameters
K-Nearest Neighbors	Number of neighbors: how many neighbors to use for calculation Radius of neighbors: the radius within which neighbors can be considered
Decision Tree	Max depth: how deep is the tree allowed to be Min samples for split: the minimum of samples required to justify a new decision

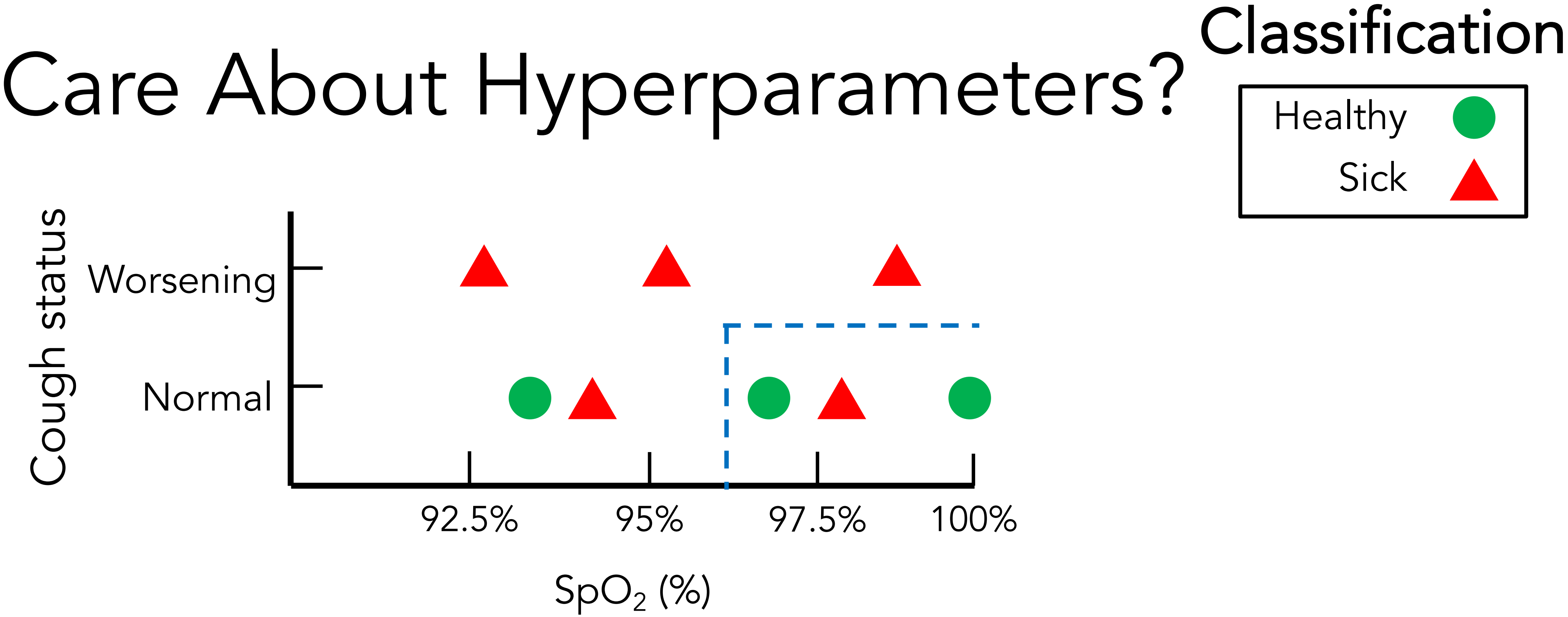
Why Care About Hyperparameters?

Machine learning is great because it can learn complex relationships that we can not readily identify ourselves

However, if we give the model too much flexibility, the model may learn complicated rules that neither make sense nor generalize well to unseen data

This is known as “overfitting”

Why Care About Hyperparameters?



Simple Decision Tree

"If $SpO_2 \geq 96\%$ AND normal cough, the patient is healthy"

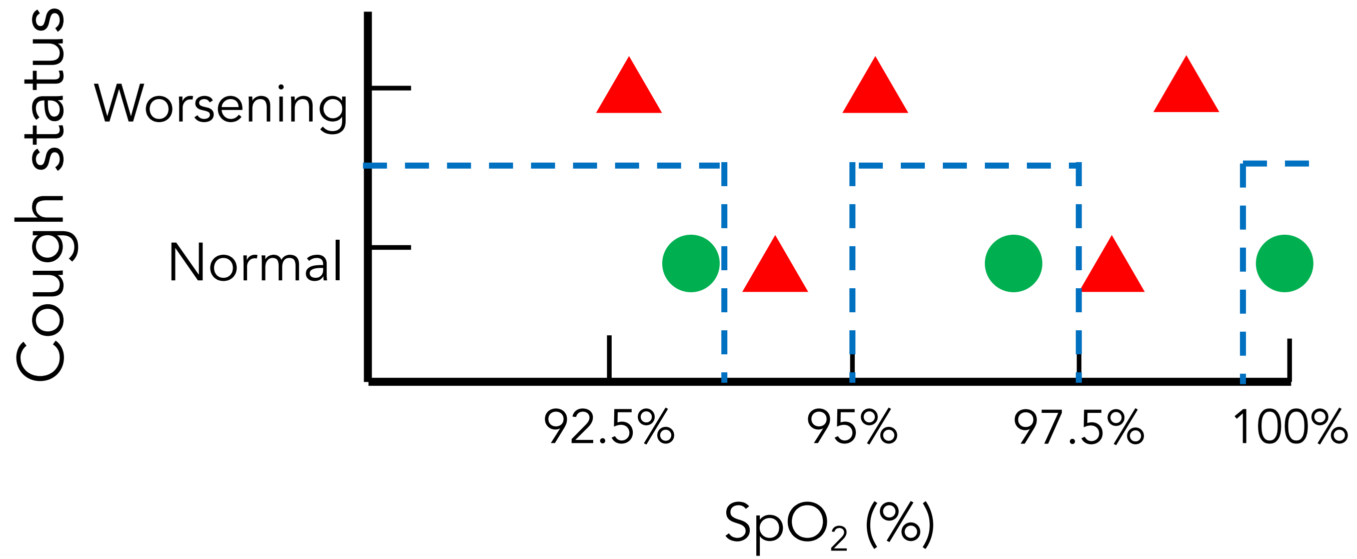
The model has okay accuracy (75%), but the rule seems clinically sensible

Why Care About Hyperparameters?

Classification

Healthy ●

Sick ▲



Complex Decision Tree

"If normal cough AND NOT (93.5% $\leq SpO_2 \leq$ 95% OR 97.5% $\leq SpO_2 \leq$ 99%), the patient is healthy"

The model has perfect accuracy (100%), but the rule does not seem to be clinically sensible

How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Why Use Feature Selection?

The more features we use, the higher the likelihood that we can successfully solve our problem

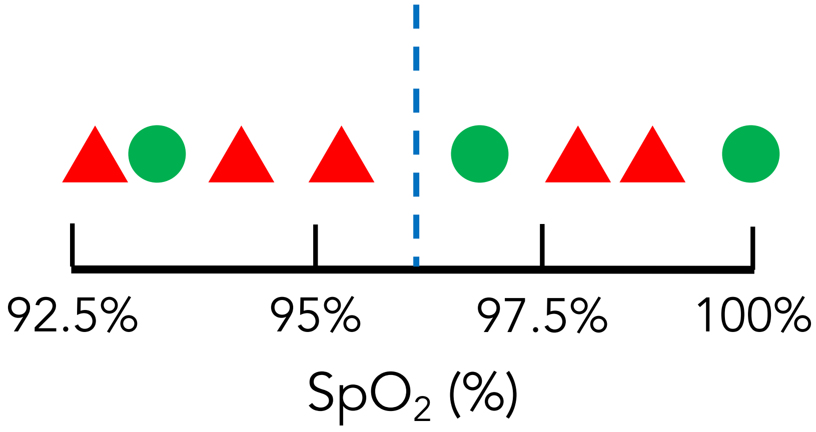
However, if we use too many features, the model may learn complicated rules that neither make sense nor generalize well to unseen data

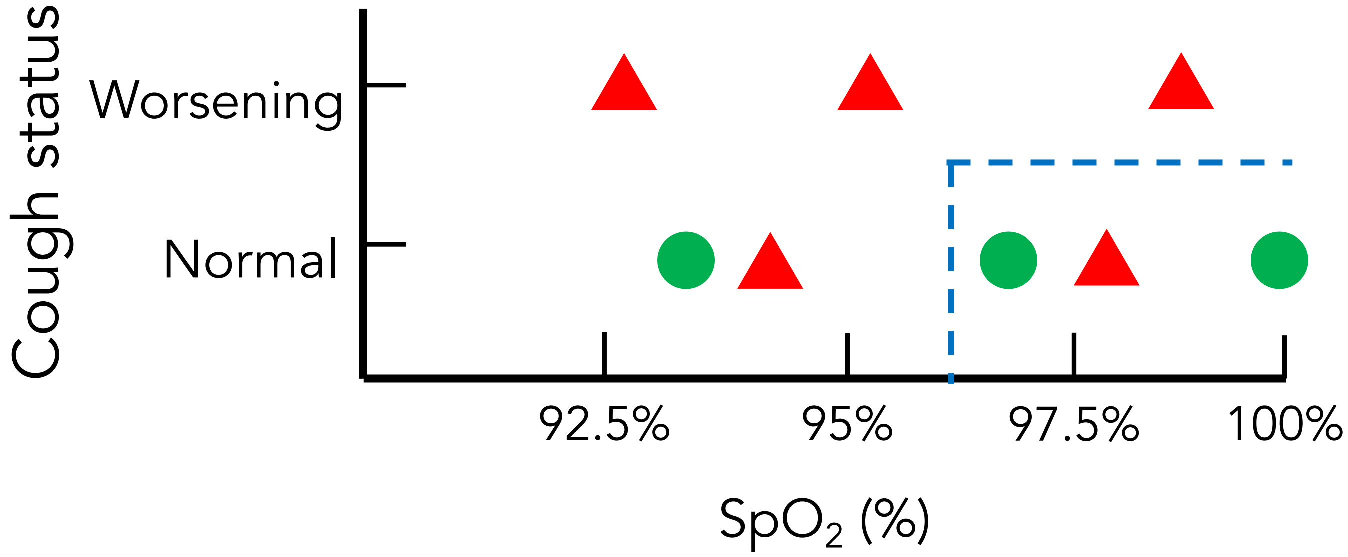
This tradeoff is known as the “curse of dimensionality” and is closely tied to overfitting

Why Use Feature Selection?

Classification

Healthy	●
Sick	▲





One Feature

"If $SpO_2 \geq 96\%$, the patient is healthy"

The model has okay accuracy (62.5%), and the rule seems clinically sensible

Two Features

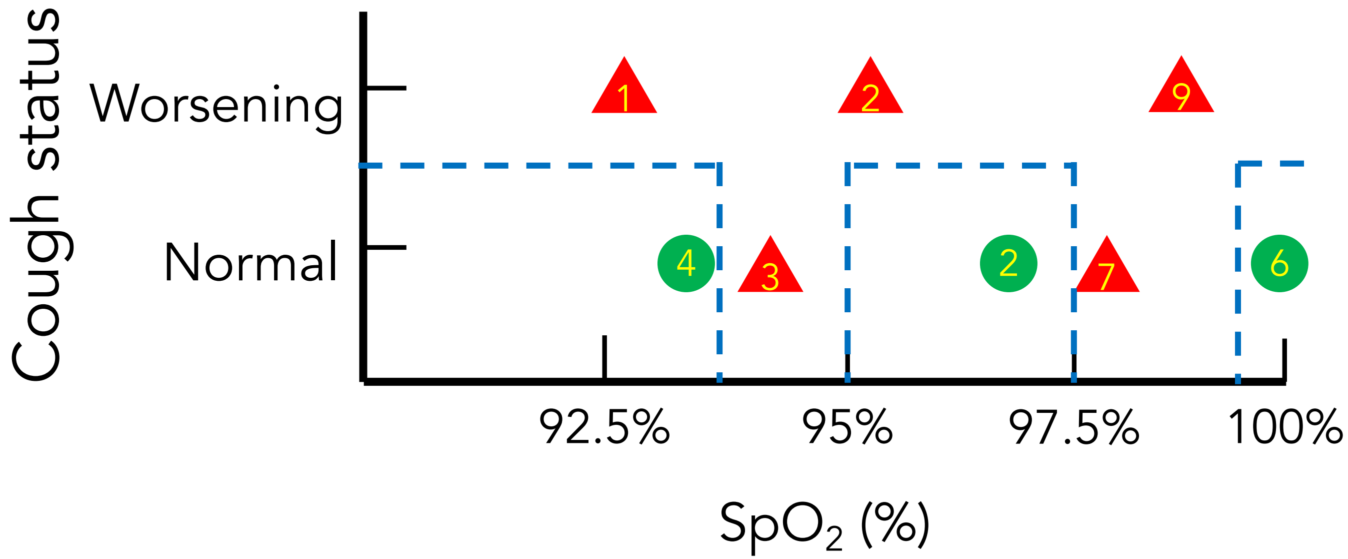
"If $SpO_2 \geq 96\%$ and normal cough, the patient is healthy"

The model has better accuracy (75%), and the rule still seems clinically sensible

Why Use Feature Selection?

Classification

Healthy	●
Sick	▲



Number in yellow: last digit of SIN

Three Features

“If normal cough AND last digit of SIN is even, the patient is healthy”

The model has perfect accuracy (100%), but it took advantage of a random feature to make a rule that does not seem to be clinically sensible

How to Select the Best Features

You can use domain expertise to ignore features that do not make sense

There are also automated techniques you can use to automatically pick the features that are likely to be the most interesting mathematically

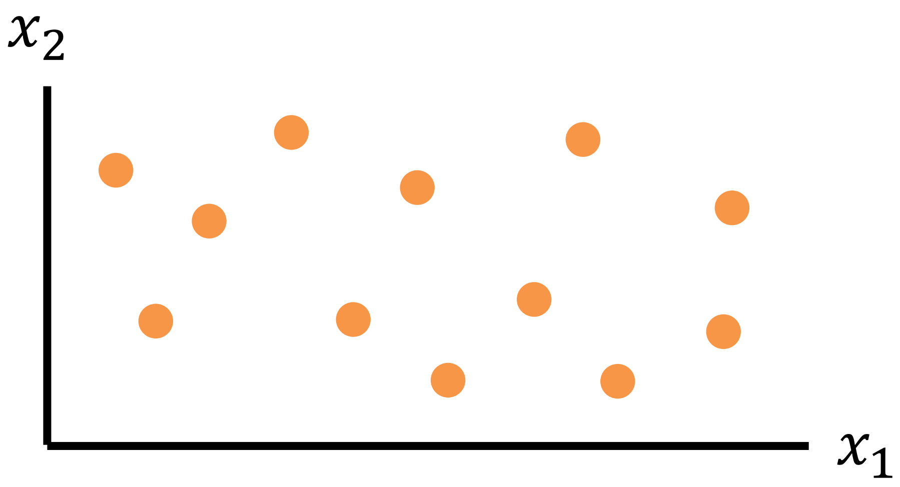
Some techniques care about the labels, while others just care about properties of the data itself

Example of Label-Agnostic Feature Selection: Variance Threshold

Only keep features that are spread apart (high variance) to keep the data as “unique” as possible

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

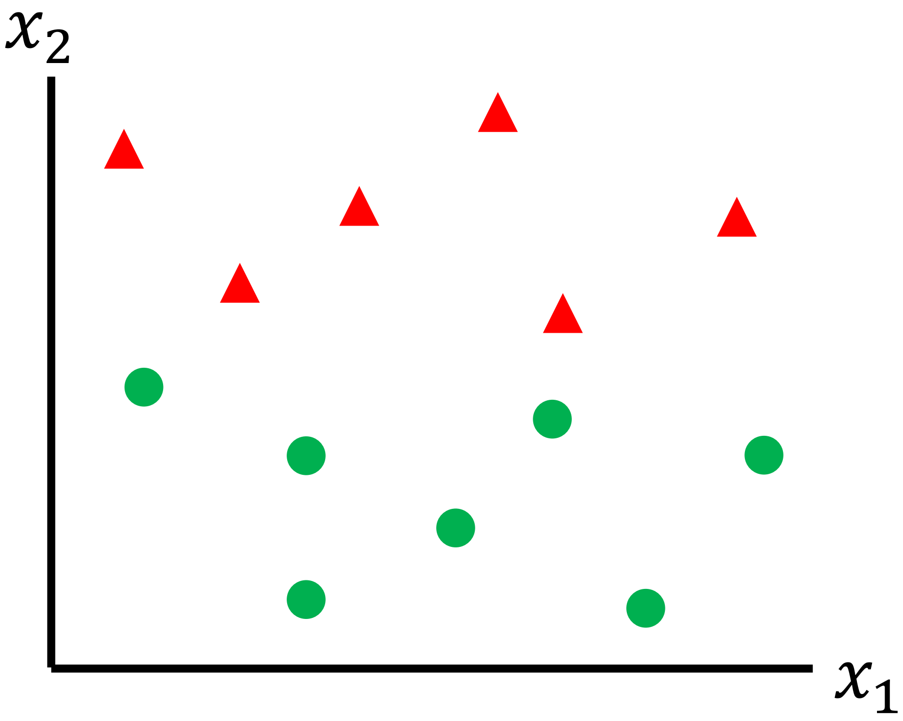
σ^2 = variance
$\bar{x}$ = mean value
N = number of points

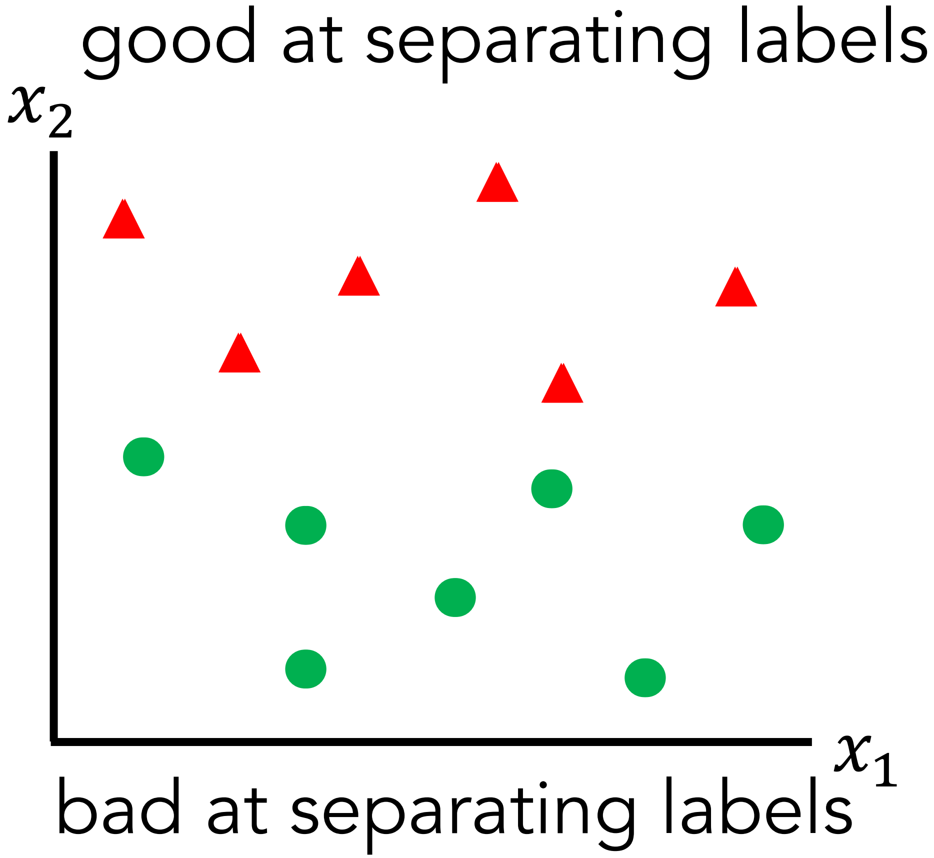


x_1 has a wider spread than x_2, so x_1 will likely make it easier to find separate instances with different labels

Example of Label-Aware Feature Selection: Univariate Feature Selection

Performs statistical tests using one feature at a time and picks the ones that work best





good at separating labels

bad at separating labels

How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Regression Results

Actual (days)	Predicted (days)
4	3
5	5
5	10
3	2
4	6

Average error: $\frac{1}{N}\sum_i \text{Predicted}_i - \text{Actual}_i$

Average percent error: $\frac{1}{N}\sum_i \frac{\text{Predicted}_i - \text{Actual}_i}{\text{Actual}_i}$

Correlation coefficient: A number between -1 and 1 that tells you the strength and direction of a relationship actual and predicted values

- 1 = As the actual value changes, the predicted value always changes in the same direction
- 0 = No relationship
- -1 = As the actual value changes, the predicted value always changes in the opposite direction

Binary Classification Results: Overall Performance

Actual (diagnosis)	Predicted (diagnosis)
healthy	healthy
sick	healthy
sick	sick
sick	sick
healthy	sick

Confusion Matrix

		Actual Label	
		Positive	Negative
Predicted Label	Positive (sick)	True positive (TP)	False positive (FP)
	Negative (healthy)	False negative (FN)	True negative (TN)

Accuracy $= \dfrac{TP+TN}{TP+FP+FN+TN}$ F1-score $= \dfrac{2*TP}{2*TP+FP+FN}$

F1-score is better suited for imbalanced datasets
(i.e., significantly different numbers of positives and negatives)

Binary Classification Results: Information Retrieval

Actual (diagnosis)	Predicted (diagnosis)
healthy	healthy
sick	healthy
sick	sick
sick	sick
healthy	sick

Confusion Matrix

		Actual Label	
		Positive	Negative
Predicted Label	Positive (sick)	True positive (TP)	False positive (FP)
	Negative (healthy)	False negative (FN)	True negative (TN)

$$\text{Precision} = \frac{TP}{TP+FP}$$

What fraction of items that were returned were relevant to my query?

$$\text{Recall} = \frac{TP}{TP+FN}$$

What fraction of items that were relevant to my query were returned?

Binary Classification Results: Diagnostics

Actual (diagnosis)	Predicted (diagnosis)
healthy	healthy
sick	healthy
sick	sick
sick	sick
healthy	sick

Confusion Matrix

		Actual Label	
		Positive	Negative
Predicted Label	Positive (sick)	True positive (TP)	False positive (FP)
	Negative (healthy)	False negative (FN)	True negative (TN)

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

What fraction of people with the given health condition were given a positive test result?

$$\text{Specificity} = \frac{TN}{TN+FP}$$

What fraction of people without the given health condition were given a negative test result?

Multi-Class Classification Results: Overall Performance

		Actual Label		
		Healthy	Hypotensive	Hypertensive
Predicted Label	Healthy	AA	AB	AC
	Hypotensive	BA	BB	BC
	Hypertensive	CA	CB	CC

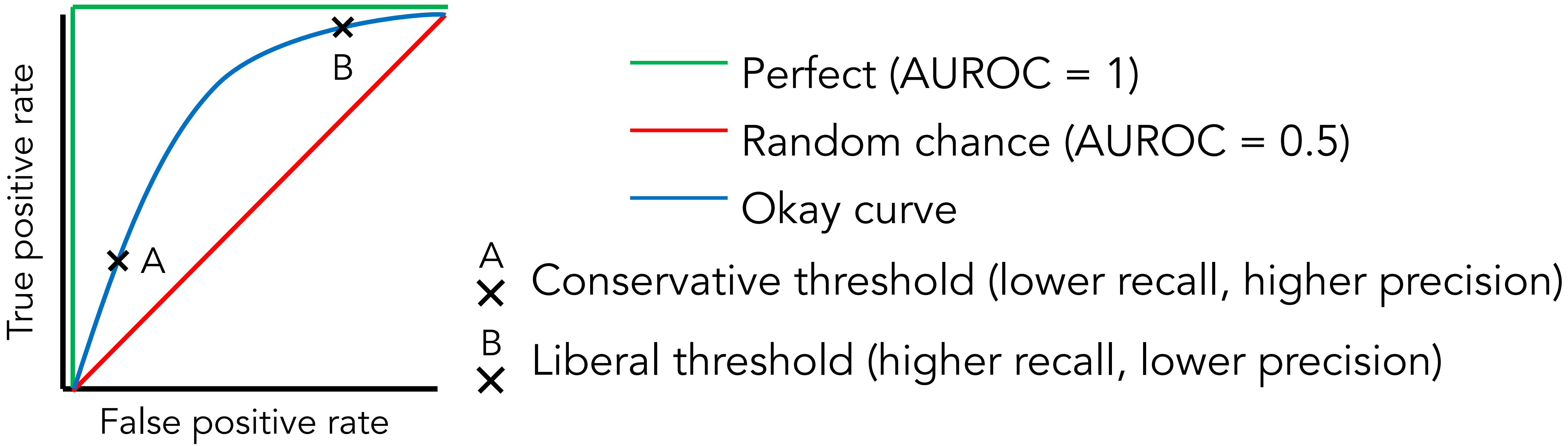
Overall Accuracy =

$$\frac{AA + BB + CC}{AA + AB + AC + BA + BB + BC + CA + CB + CC}$$

Receiver Operating Characteristic (ROC) Curve

Shows how a classifier’s performance changes depending on its internal decision threshold

Area under the ROC Curve (AUROC) is an aggregate measure between 0 and 1 that reports summarizes model performance across all thresholds



How to Do Machine Learning in 5–7 Easy Steps

1. Define the problem we are trying to solve
2. Create your features and labels
3. Decide how the data should be split for training and testing
4. Select an appropriate model
5. (Optional) Select your hyperparameters
6. (Optional) Add feature selection
7. Use an appropriate method for interpreting results

Resources

Overviews of Machine Learning	
“Glossary of Machine Learning Terms”	Google
Coding Examples	
“Machine Learning Mastery”	Jason Brownlee