COMPUTATIONAL APPLICATIONS TO POLICY AND STRATEGY (**CAPS**)

Session 2 – Supervised Learning

Leo Klenner, Henry Fung, Cory Combs

Outline

1. Engaging Data
2. Case Study
3. The Branches of Supervised Learning
4. Supervised Learning Model Construction
5. Regression Models
6. Classification Models
7. Tree-based Models (CART)
8. Model Evaluation

Big-picture Goal:

To understand how machine learning decisions are made, what challenges are involved, and how technical challenges correlate to value judgments

1. Engaging Data

How do we engage data to learn?

1.1 Human Analogies to Supervised vs. Unsupervised Learning

Supervised

- An instructor provides sample problems. You try them; the instructor assesses your accuracy; and you do more exercises.
- **No new methods are employed**, only accuracy using the given methods.
- You continue until you can consistently answer new questions **of the same form**.

Unsupervised

- You are given an abstract pattern-finding exercise – for example, grouping pieces of art.
- You have no training, no context, and no feedback.
- **Reasonable individuals** may find different patterns. Some patterns may be **more valuable** than others.

1.2 Supervised vs. Unsupervised Learning Specifications

Supervised Learning

- Allows us to train machines that learn from **labeled data**.
- Includes making inferences and predictions about new, unlabeled observations based on previous observations.
- Includes regression and classification.

Unsupervised Learning

- Allows untrained machines to autonomously learn about **unlabeled data**.
- Includes pattern recognition and categorization.
- (The distinction between categorization and classification will become clear in the next session.)

1.3 Features and Observations

We can decompose all datasets into two components:

- **Features**: the variables in a dataset (the columns in a table).
- **Observations**: the number of unique records (the rows in a table).

ID [Observation Number]	Province [Feature 1]	Security Perception [Feature 2]
3032	Badghis	3
3033	Samangan	2
3034	Wardak	5

Example compilation from USAIS MISTI Survey data

1.4 Learning from Labels

Labels refer to target variables

- To say a supervised learner “trains on labeled data” means that it learns to correlate a specified target variable with selected features.
- “Making a prediction” means inferring the label of an unlabeled data point given its feature values.

We can now more clearly state the supervised vs. unsupervised distinction:

- Supervised learners train to predict or infer a target variable based on a range of features, i.e. independent variables, using examples for which the target variables are already known.
- Unsupervised learners do not train, and run various algorithms to find patterns among the features.

1.5 Using Real-world Data for Supervised Learning

You find a dataset "in the wild". How do you know whether you can use it for supervised learning?

The answer has two basic components: the questions you want to answer, and a judgment call about the features.

1. Does the dataset contain a variable you want to be able to *predict*?
2. Do you believe that variable *can be reasonably well predicted* given the available features?

The latter entails hypothesis formation and testing. One *could* dive in headlong with correlation tests – but there is a *lot of data* in the world.

Good models begin with good questions and good hypotheses.

1.6 “Better” Data: Feature Selection

- Does *more* = better? No.
- **More *relevant* = better.** We select more relevant data through **feature selection**.

- “Garbage In, Garbage Out.” Giving a model irrelevant or partially relevant features – even given accidental or confounded correlations – **teaches the model an inaccurate worldview** (as a human analogy).

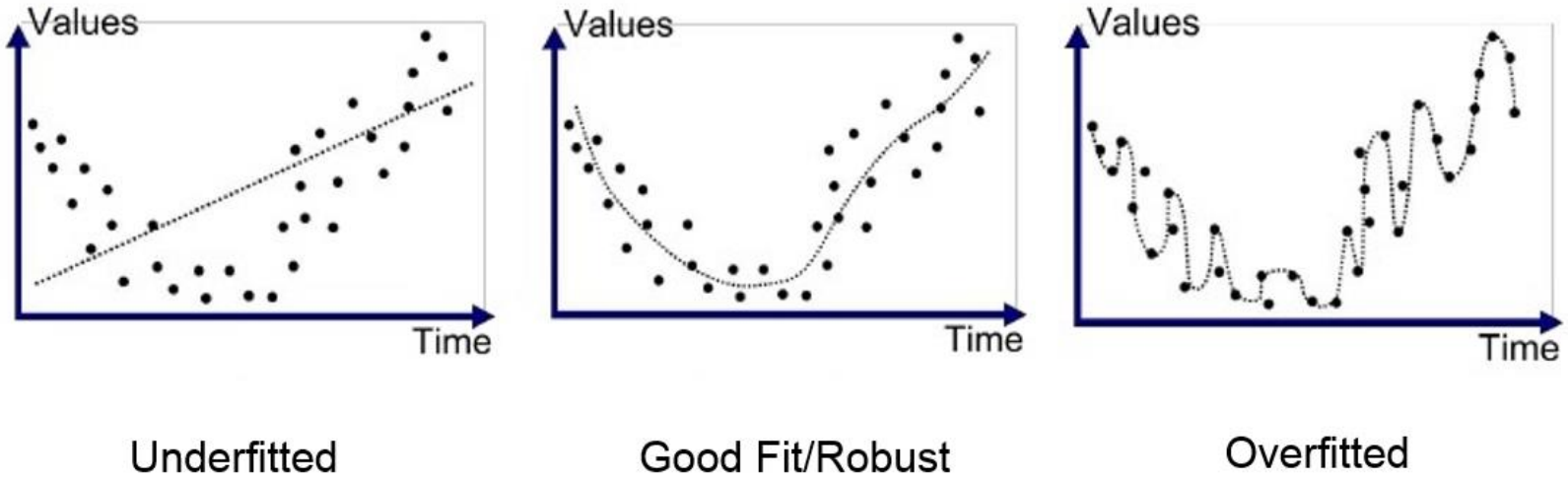
- Models also require a robust sample of *each* feature. Even large datasets can give too few of one feature.

The key take-away:

- **Appropriate feature selection** and **numerous observations for each feature** are joint necessities for successful models.

1.7 Data Risks: Overfitting and Bias

- **Overfitting**: the model has learned to map *noise*, rather than signal. It will perform very well for *training data* and very poorly for *new data*.



- **Bias**: a systematic skewing of results. (This term is laden with different meanings, from the mathematical to the ethical. We will see other forms later.)

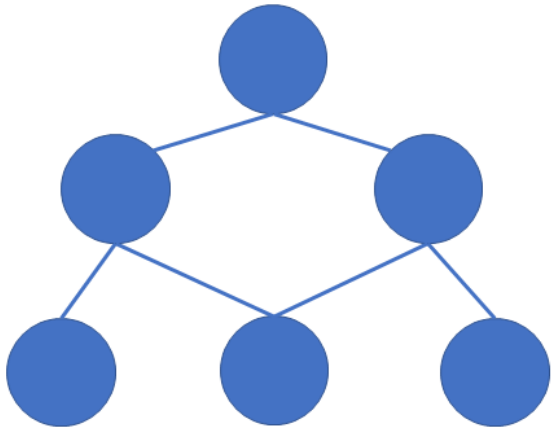
1.8 Representative Data

- Supervised learning models learn precise trends from data. They do not adapt without retraining.
- **Your model can *only* handle inputs similar to those on which it trained.** Ensure that your training and test data are representative of the real-world data you intend to model.
- Adaptation requires *online learning*, i.e. continual training, which brings its own risks and benefits.

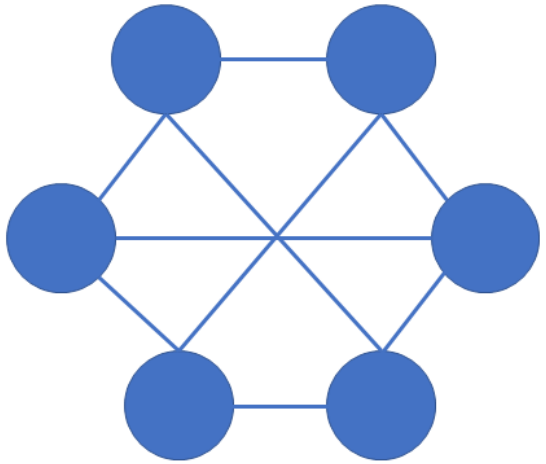
2. Case: Data Under Conditions of Uncertainty

How can we approximate organizational structure?

2.1 Centralized or Decentralized?



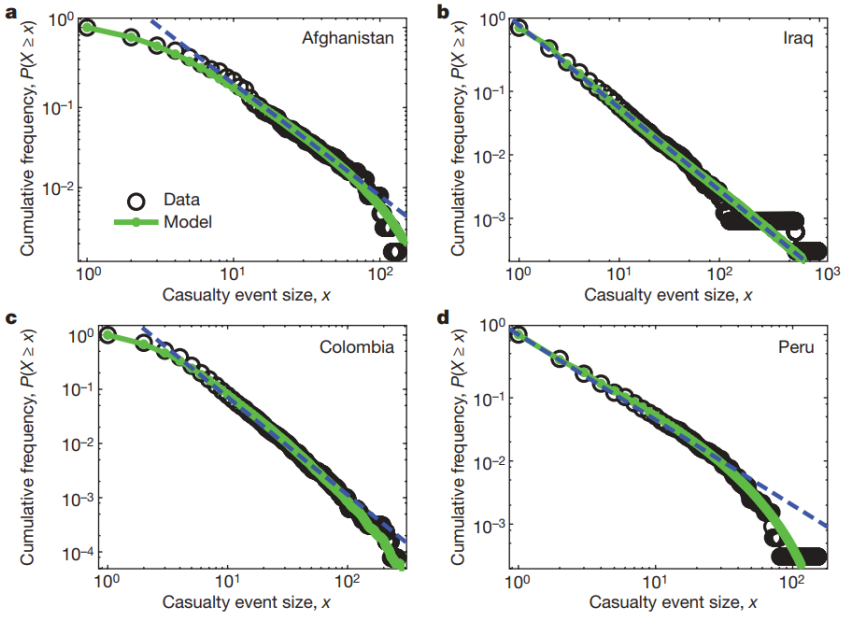
“Hierarchy”



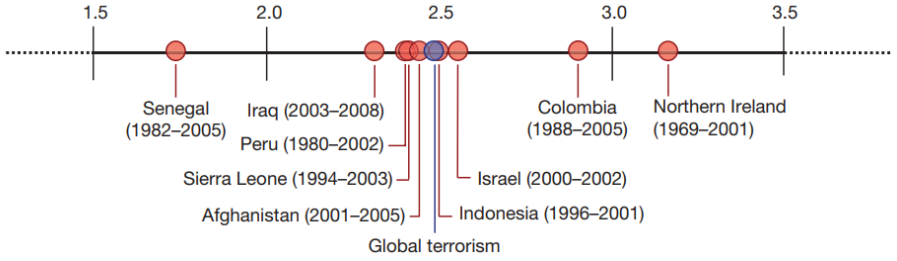
“Network”

2.2 Possible Solution – Attack Time and Casualties

From JC Bohorquez et al. 2009. Common Ecology Quantifies Human Insurgency

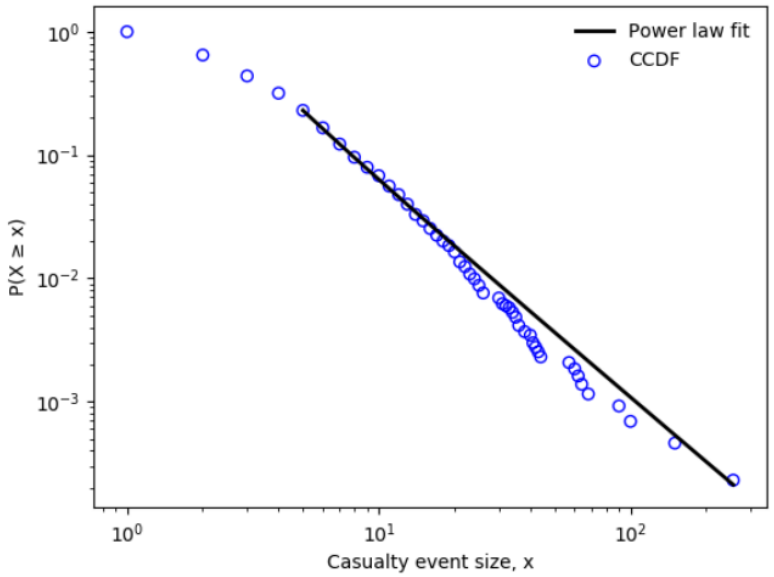


Log-log plots of power law estimtates for Afghanistan, Iran, Columbia, Peru

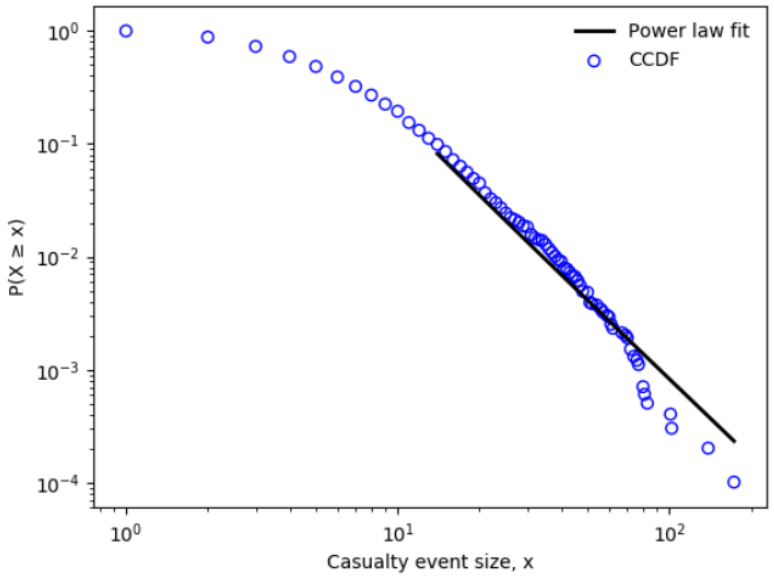


Scaling parameter α for various conflicts

2.3 Application to the Afghan Conflict

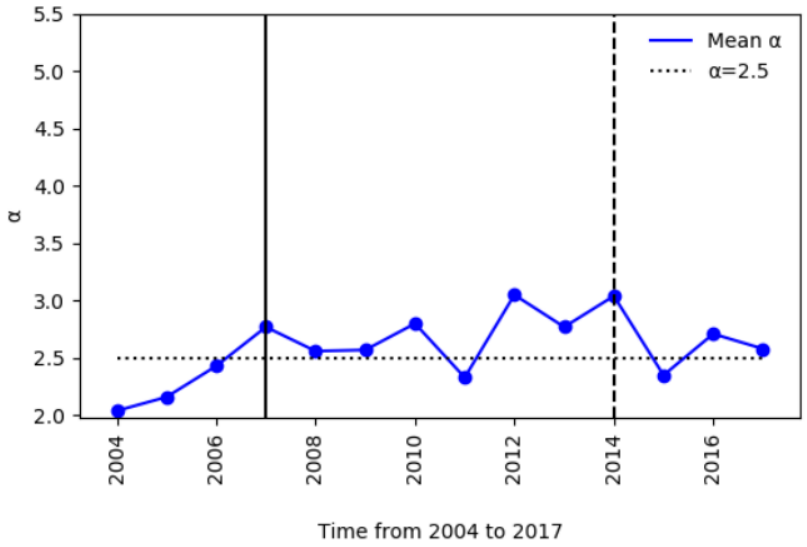


Power law estimate for total Afghan conflict 2004-17
α= 2.73
Global Terrorism Database

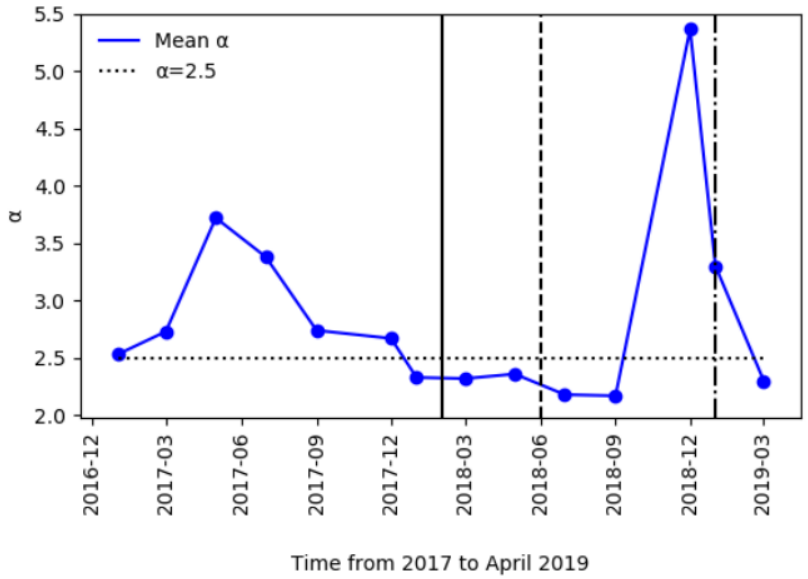


Power law estimate for Taliban only 2017-18
α= 3.03
Armed Conflict Location and Event Database

2.4 Change in the Taliban's Org Structure Over Time



Year-on-year estimates for 2004-17
Global Terrorism Database



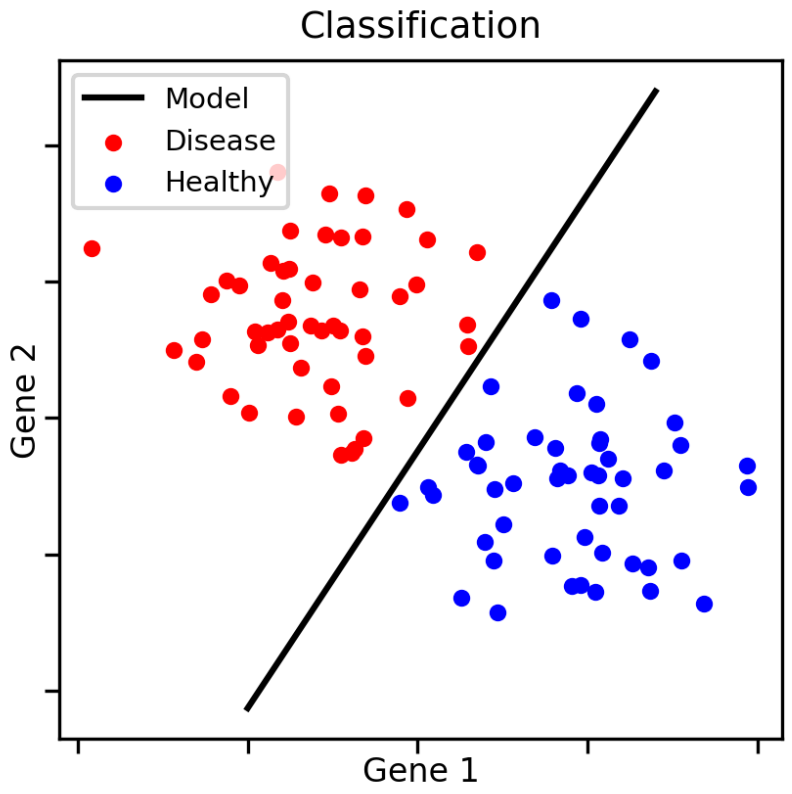
Two-month-window estimates for 2017-19 (April)
Armed Conflict Location and Event Database

3. The Branches of Supervised Learning

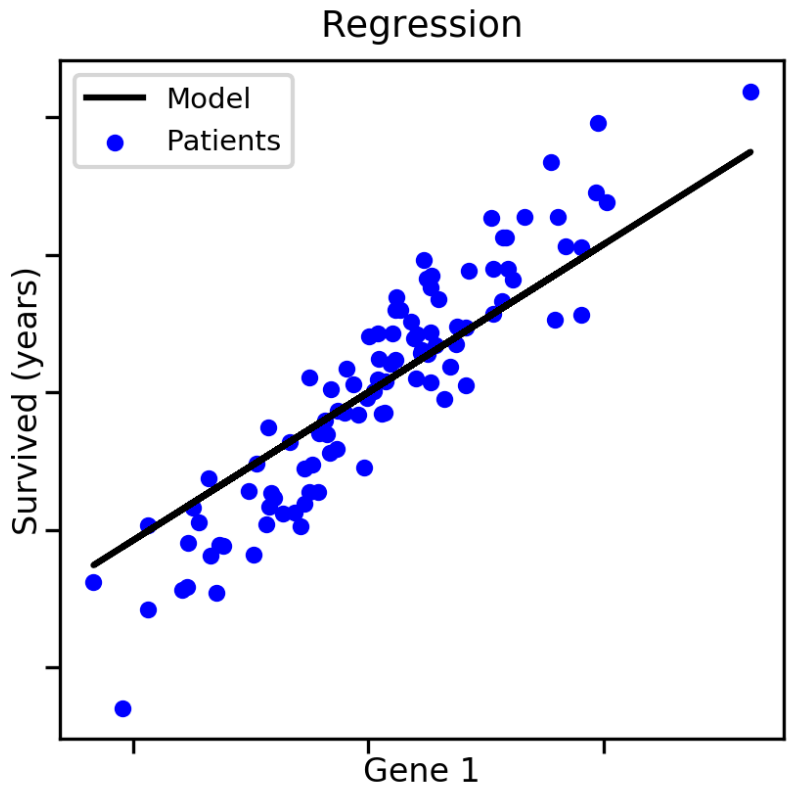
Foundations of classification and regression

3.1 Classification and Regression

Classification: the model predicts a **categorical** target variable.



Regression: the model predicts a **continuous** (numerical) target variable.



3.2 Scenario Analysis

In groups, we will consider representative situations for which we want to employ supervised learning.

Information is intentionally limited, and you are encouraged to pose any and all questions you would ask your data team.

Following discussion, please prepare to share your assessments of:

1) What model type should be used?
2) What features might be promising?
3) What risks do you foresee?

3.2.1 Scenario Analysis A

Consider one of the two situations below, 1a) or 2a), given three questions:

1) What model type should be used? 2) What features might be promising? 3) What risks do you foresee?

1a) You are a bank loan officer responsible for determining whether to approve applications based on whether each applicant is likely to default.

The market you are responsible for has been historically stable and you have tens of thousands of records on hand to judge past experience of similar applicants.

2a) You are a USAID analyst evaluating the quantitative impact of renewable energy credits on household emissions outcomes.

You have data from a well-designed randomized control trial collected from tens of thousands of households representing the majority of the population.

3.2.2 Scenario Analysis B

Now consider the **variant** of the situation you considered, 1b) or 2b), given three questions:

1) What model type should be used? 2) What features might be promising? 3) What risks do you foresee?

1b) You are a loan specialist developing a national micro-lending platform for an emerging market with historically limited small-scale lending.

You have a small pool of data from trial programs in three cities.

2b) You are a State Department analyst assessing the impact of a stabilization initiative on local security in a post-conflict region using survey data.

You have just under one thousand responses collected by volunteers in their home regions.

3.2.3 Several Key Questions

For situation 1b):

1) Is there enough historical data for statistically significant results?
2) Is the market changing in such a way that the existing data will no longer be representative, i.e. the values for the same features would be different today?
3) Is the market is changing such that different features are relevant?

For situation 2b):

1) Are the survey results sufficiently complete for statistically significant results (considering NA values and missing entries)?
2) Given the subjective nature of survey data, what can we, and can we not, infer from the dataset?
3) How can and should we assess the quality and representativeness of the data?

4. Supervised Learning Model Creation

How are supervised learning models created?

4.1 Model Development Overview

- The full process of model development varies dramatically by team, problem, and environment. Indeed, the development process itself is a critical factor (see Knight Capital).
- From the pure machine learning standpoint, however, all supervised learning models follow some variation on the following theme:

1) Data preparation

   Includes collection, cleaning, manipulation

2) Model instantiation

   Selecting model type, framework, initial parameters

3) Model training

4) Model testing and evaluation

5) Model deployment

   Usually falls to server administrators, etc.

4.2 Model Construction

Basic model construction follows the following procedure:

1) **Split** the data into training and test datasets
2) **Build** the model
3) **Fit** the model to the training data
4) **Predict** values using the test data
5) **Evaluate** the model's performance

These steps are highly iterative. In particular, fitting, prediction and evaluation will be iterated continuously until suitable results are obtained. Iteration usually accounts for the majority of the construction timeline.

4.2.1 Train-test-split

Why split the data?

- The model requires labeled data to learn. However, we need to test it in a way that lets us assess its performance. To do this, we “hide” some of the known data during training.
- Once it is trained on the *train* dataset, we have it predict labels for the *test* dataset.

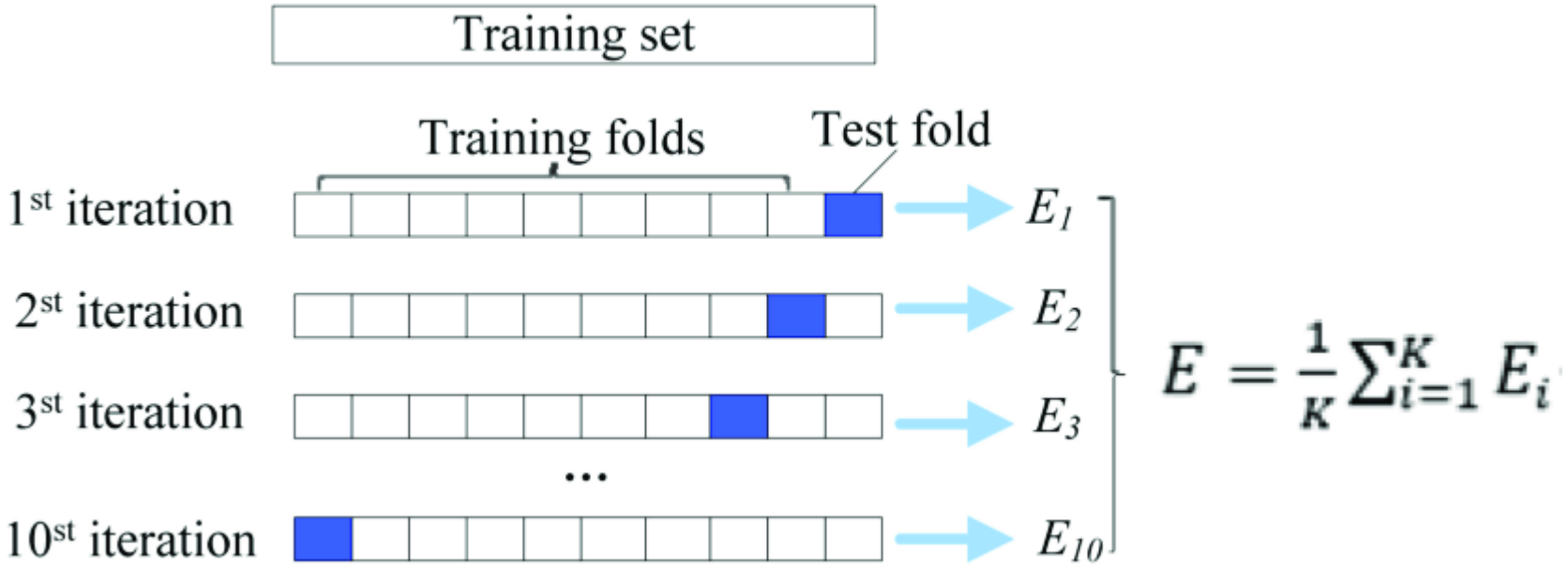
How much of the data do we put into each of *train* and *test*?

- More training data generally means better performance. More test data generally enables better insight into potential overfitting or underfitting, etc.
- The trade-off is direct and unavoidable. The solution: have more (relevant) data.

4.2.2 Cross-validation

Can different splits cause inconsistencies or introduce random noise?

- Yes. To combat this, we use **cross-validation** (CV).
- CV splits the data into *different* training and test sets, runs the model with each set, and compares the results. This effectively "smooths" potential bias from any random split.



4.2.3 Instantiation, fitting, prediction, evaluation

- **Building** a model usually means *instantiating* a Python class: “tell the program which model to use”.
  Most machine learning now uses high-level APIs such as Scikit-learn, PyTorch, TensorFlow and Keras.

- **Fitting** the model means training it on the training data. A model works to minimize error rates, also known as loss.

- **Predicting** means running the trained model on the test data, yielded predicted labels.

- **Evaluation** compares the predicted and actual test data labels to see how well the model performed.

5. Regression Models

What is regression in the machine learning context?

5.1 Regression

Regression is the bread and butter of continuous variable prediction. Regression models do the following:

1) find a reliable trend between correlated features;
2) use the discovered trend to predict a value of interest (the label) given feature data.

Linear models take the form where y is the label, x values are selected features, and beta values are the respective weights. :

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

“Fitting” a regression model means calculating the values of beta using the known labels from the training data.

6. Classification Models

How do machines deal with categorical data?

6.1 What a “label” really means to a machine

- In regression, the label is the *quantity* of interest. In classification, we think of a label in the more colloquial sense of a word or term.

- **For the machine, *everything* is a number**. To deal with categorical (non-numerical) data, we must first “translate” it into numbers on which algorithms can operate.

- As an example from natural language processing: each word in a dictionary may be assigned a value between 0 and N (with N words in the dictionary); then each word is translated into a vector of 0s and a single 1, which represents that unique word.

6.2 Classification

- Classification comes in two forms: binary and multiclass. We will focus on binary classification.

- Consider again the earlier loan problem: *You are a loan officer responsible for determining whether to approve applications based on whether each applicant is likely to default.*

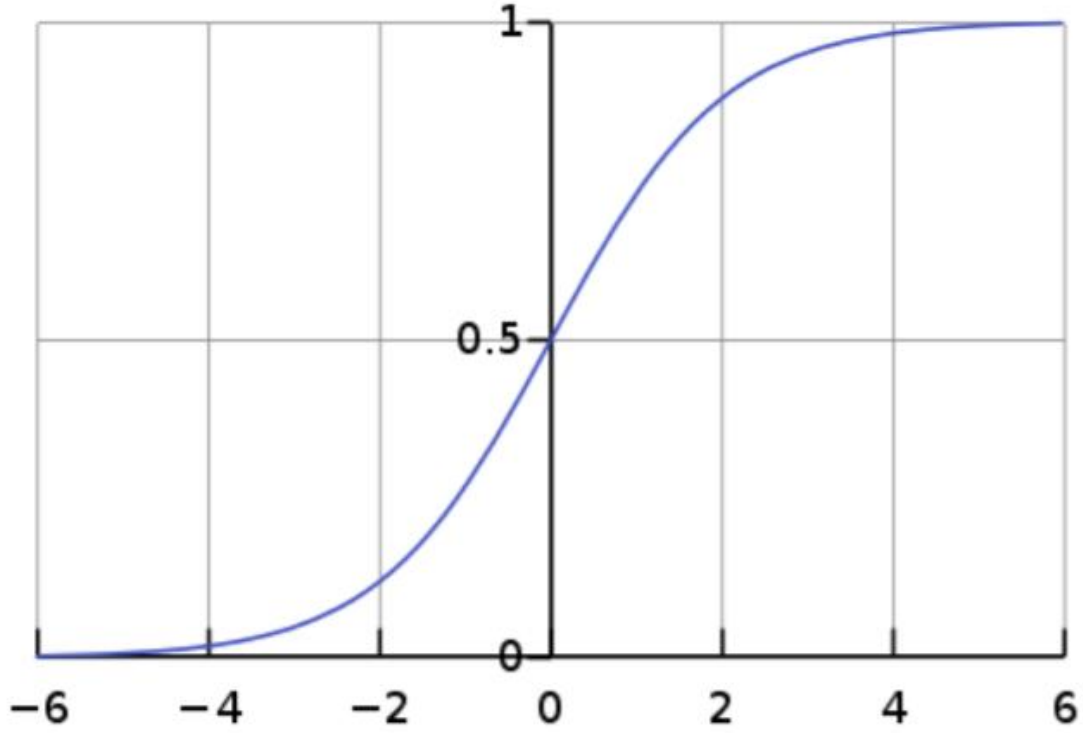
1. We construct a binary classification with labels: “default” and “no default”.
2. As a simple case, if an application is predicted to have “no default”, it will be approved.
3. We say that “default” = 1 and “no default” = 0. We can now employ a classification model with the goal: **predict either 1 or 0 for all inputs.** We want no values between 1 and 0.

6.3 Logistic Regression and the Sigmoid Function

- Logistic regression is the archetypal model for classification. (Despite its name, it is not used for regression problems.)

- How do we transform a range of possible outputs into just two outcomes? The **sigmoid function**, where p is a prediction:

$$p = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + 1}$$



6.3 Logistic Regression and the Sigmoid Function

- As we can see, the sigmoid function “squishes” values toward 0 or 1 by transforming a variable z.
- What is z? *The variable z is itself a linear regression.* It is simply a hypothesized relationship among features, reflecting their various weights in determining the outcome.

$$p = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + 1}$$

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon$$

6.4 The Vital Role of Data Imbalance

- **Data imbalance:** where some features have far less representation than others.
- While important for all supervised learning problems, classification problems are particularly famous for facing data imbalance issues.

- Why is this a problem? Say you have 100,000 cases of “no default” and 1,000 cases of “default”, and you *merely (falsely) state that no loan will default*, what is the **accuracy** of your statement?

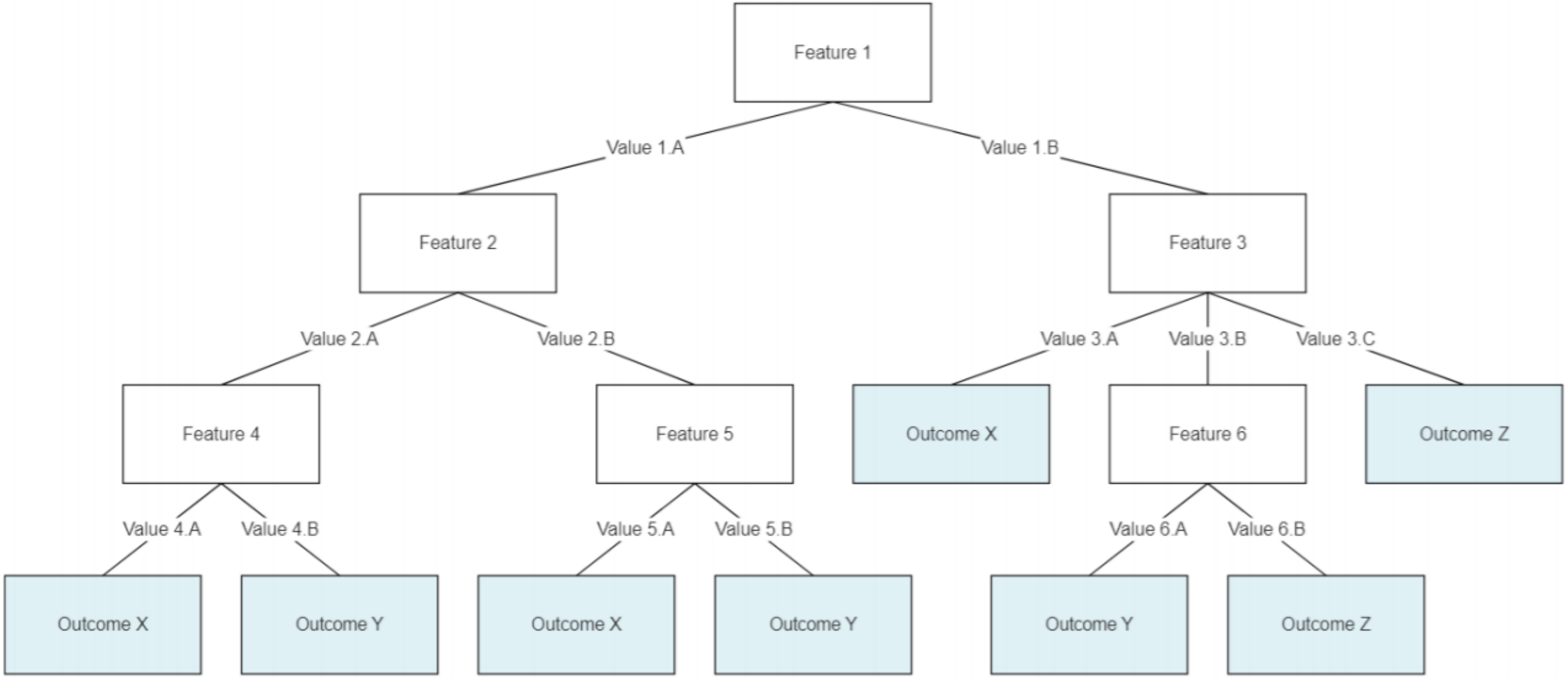
- Consider early cancer detection systems. Given the low incidence of cancer in most large populations, should you base your system on a measure of accuracy?

- One solution is to gain more data. This is not always possible. However, we can replace accuracy with **other metrics** to deal with this issue.

7. Tree-based Models (CART)

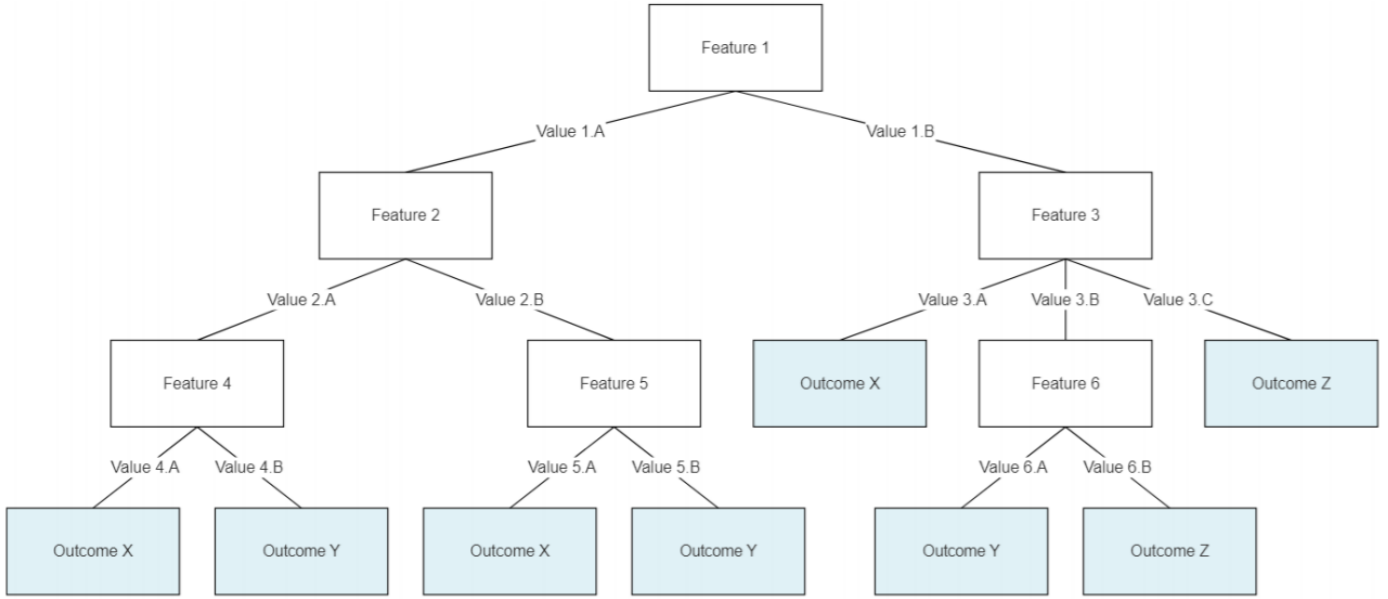
Explainable alternatives for supervised learning

7.1 Classification and Regression Trees (CART)



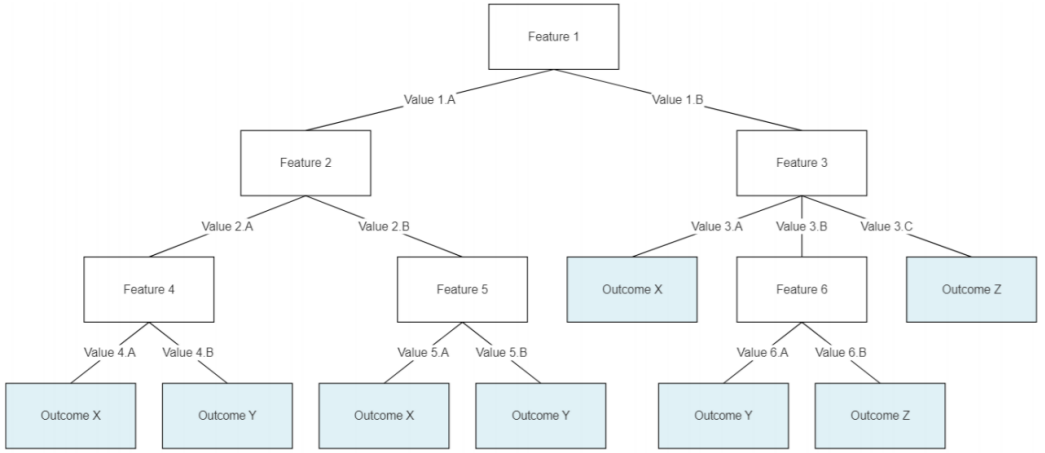
7.2 Explainability as a trade-off

- While high-performing, the previous models are not highly intuitive.
- Tree-based models, meanwhile, allow much easier graphical representative and an intuitive flow, similar to the game “twenty questions”.
- Unlike the game, trees can be used to predict any number of labels, including continuous variables, and are not bound to binary splits.
- Tree-based models do, however, tend to overfit. We can mitigate overfitting, but it adds complexity to model development; hence they have not yet taken over modeling.



7.3 Tree mechanics

- **Nodes:** the assessed features (transparent boxes)
- **Branches**: also, splits; possible values (lines)
- **Leaves**: possible outcomes (blue boxes)



To train a tree:

1. Test which feature best splits the labeled data
2. Set the identified feature as the first node
3. Test which feature best splits each now-separated set of labeled data
4. Establish the identified features as the next respective nodes

And so forth, until no further information is gained, or a specified cutoff is reached.

For continuous variables, the model also determines threshold values that best split the data.

7.4 Tree Algorithms

Tree-based models can use various algorithms to decide how to split the data. Key algorithms include:

- **Information gain**, which minimizes *entropy*, a function of the probability of two outcomes.
- **The Gini index**, which measures information "purity" as the probability that a random classification will be incorrect.
- **Chi-square**, which is a form of statistical significance of the splits comparing parent and child nodes.

Note that these algorithms all serve to conduct **feature selection**.

7.4.1 Information Gain

- **Information gain**, which minimizes *entropy*, a function of the probability of two outcomes.

$$Entropy = -\sum_{j=1}^{n} -p_j log_2 p_j$$

where $E(S)$ is entropy, p is the probability of outcome 1 and q is the probability of outcome 2.

Information gain is equal to $1 - E(S)$. The algorithms identifies the split that minimizes entropy, which indicates maximal information gained.

7.4.2 The Gini Index

- **The Gini index**, which measures information “purity” as the probability that a random classification will be incorrect.

$$Gini = 1 - \sum_{i=1}^{n} p_i^2$$

where n is the number of categories and p_i is the probability of a data point's classification in the given category.

7.4.3 Chi-square

- **Chi-square**, which is a form of statistical significance of the splits comparing parent and child nodes.

The chi-square formula is, for a given observation O and a given expected value E:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

7.5 Tree Algorithm Governance

There are two basic ways to handle overfitting in a single tree:

1. **Allow** the tree to overfit, then **prune** it: identify and remove features with low impact
2. **Specify** a halting threshold: stop the tree from having "too many" branches

In practice, more advanced algorithms can also handle overfitting through other means. Notably:

- **Random forest** runs myriad trees and smooths the results, resulting in an average that minimizes any particular bias, and retains (in principle) the real trends in the data.

Two key take-aways:

1. **Modeling is a trial and error process**. It requires time and exploration.
2. **No one has a crystal ball**. While the outcome of choosing one model or another is hard to predict, multiple models can be tested and compared. It is resource intensive but often helpful.

8. Model Evaluation

How do we achieve and verify the desired model performance?

8.1 Regression

Regression models are evaluated by variations on a theme: how *close* are the predicted results to the true values?

The two most common assessments are **mean absolute error (MAE) and root mean square error (RMSE)**:

$$MAE = \frac{1}{n}\sum_{i}^{n}|y - \hat{y}| \qquad RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(y - \hat{y})^2}$$

8.2.1 Classification: the Confusion Matrix

Classification models are a little more complex and interesting – and important to know in detail.

The key metrics for classification are based on the **confusion matrix**:

	Actual Values Positives: 1 Negatives: 0	
Predicted Values Positives: 1 Negatives: 0	**True Positives (TP)** *predicted: 1 actual: 1*	**False Positives (FP)** *predicted: 1 actual: 0*
	False Negatives (FN) *predicted: 0 actual: 1*	**True Negatives (TN)** *predicted: 0 actual: 0*

A simple 2x2 confusion matrix. Confusion matrices can be arbitrarily complex, reflecting all labels.

8.2.2 Classification: Fundamental Metrics

The confusion matrix display **four fundamental metrics**. For context, consider a classification system designed to detect cancer. Assume a relatively low incidence in the given population - say, 0.5%.

- **Accuracy**: measures percentage of total correct predictions: (TP * TN)/(TP + FP + FN + TN)
  - Common and useful for balanced datasets; potentially dangerous for imbalanced datasets.
- **Precision**: measures percentage of predicted positives that were correct: (TP)/(TP + FP)
  - For example, of the people the model predicted to have cancer, how many actually had cancer?
- **Sensitivity**: measures percentage of actual positives that were correctly predicted: (TP)/(TP + FN)
  - For example, of the people who actually had cancer, how many did we detect?
- **Specificity**: measures percentage of actual negatives that were correctly predicted: (TN)/(TN + FP)
  - For example, of the people who did not actually have cancer, how many did we predict did not have cancer?

8.2.3 Classification: the Ethics of Metrics

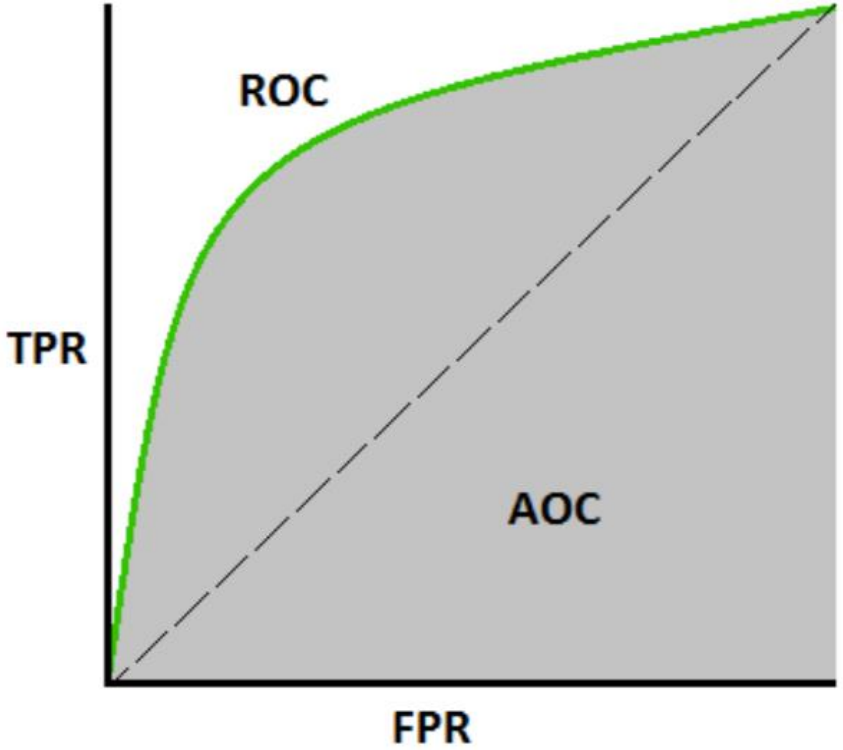
- For many problems, **it matters in which direction we are wrong**.
- No algorithm is perfect; and no algorithm *could* be perfect: the world is not fully deterministic.

Consider the following for the cancer threat detection system:

- Would you rather miss an actual threat or incorrectly suggest a threat, and find none?
- While the simple answer is obvious, what would *complete* risk aversion suggest? (Hint: a useless system.)
- So, what *threshold* is acceptable? What *balance* of true positives and false positives (etc.)?

8.2.4 Classification: Area Under the Curve – Receiver Operating Characteristics Curve

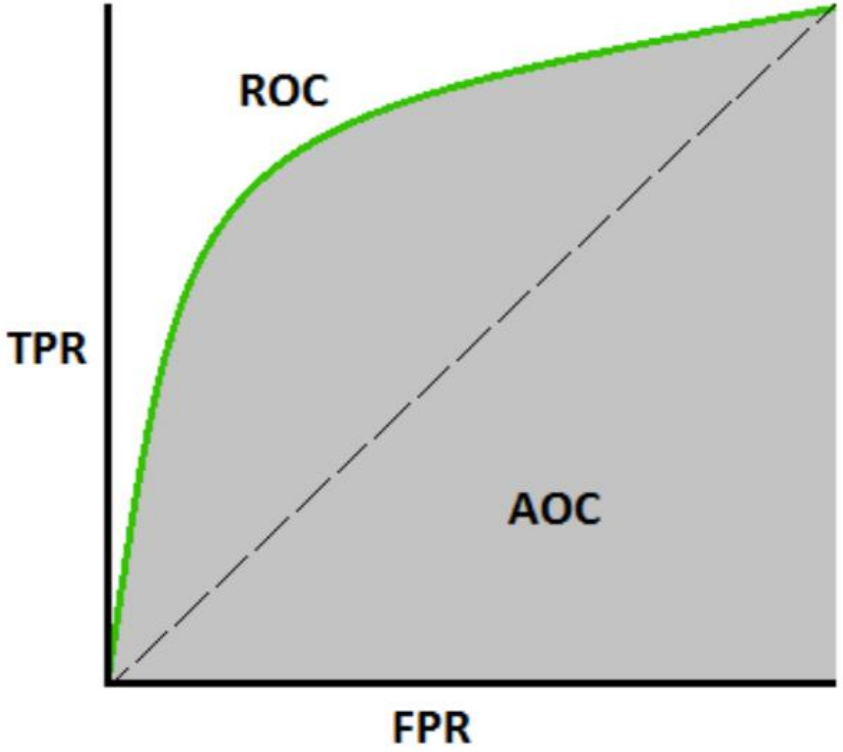
- Real-world interests often require *balancing* interests. A key tool for this is the **AUC-ROC** curve:



Model AUC-ROC curve. Credit Sarang Narkhede, Towards Data Science.

8.2.4 Classification: Area Under the Curve – Receiver Operating Characteristics Curve

- The **AUC-ROC** curve allows us to identify an optimal point between the true positive rate, which is just sensitivity, and the false positive rate, which equals 1 – specificity:



Model AUC-ROC curve. Credit Sarang Narkhede, Towards Data Science.

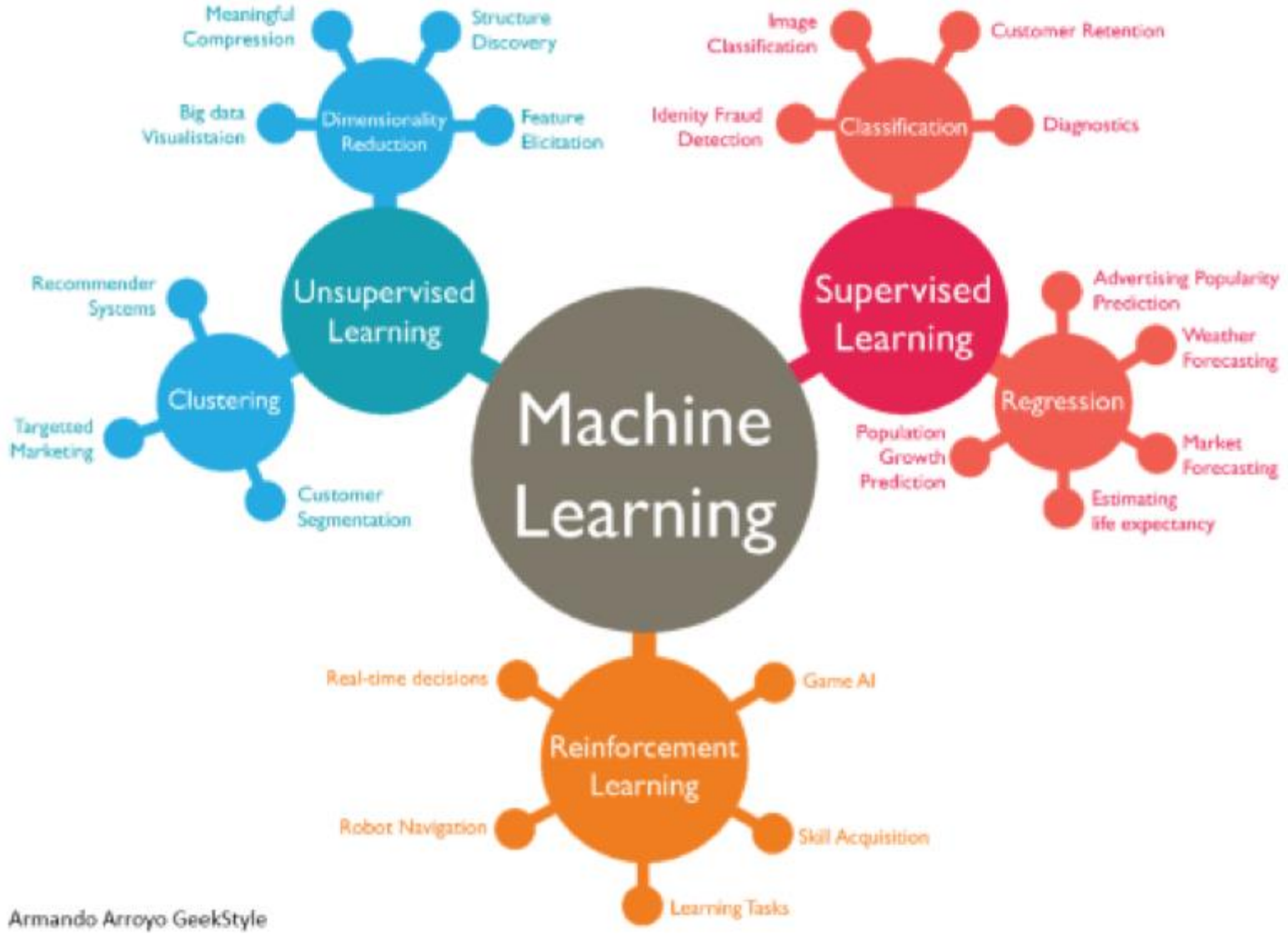
8.3 Metrics

Metrics can be complex. The problems they elucidate, however, are essential. The question is not simply *whether* your model works, but *in what ways* – with what trade-offs, with what intended ethics, and with what actual outcomes.

With these starting points, you have a several key sets of questions in your toolkit for bridging the gap between policy makers and engineers:

- Does the model need to be easily understandable to stakeholders, in terms of 1) the features it uses and 2) the way it calculates the outcomes?
- Does a model fit test data as well as it fits the training data? If not, is your dataset balanced? Finally, do the test data wholly reflect the real data of interest?
- Does the evaluation method align with the intended goals and ethical considerations? In which direction are incorrect results biased: false positives or false negatives?
- … what questions would you suggest?

Looking ahead



Next steps for further exploration?