Applied Machine Learning

Amit Kapoor
@amitkaps

Bargava Subramanian
@bargava

Getting Started

- Download the Repo: https://github.com/amitkaps/applied-machine-learning
- Finish installation
- Run jupyter notebook in the console

Schedule

0900 - 0930: Breakfast

0930 - 1115: **Session 1** - *Conceptual*

1115 - 1130: Tea Break

1130 - 1315: **Session 2** - *Coding*

1315 - 1400: Lunch

1400 - 1530: **Session 3** - *Conceptual*

1530 - 1545: Tea Break

1545 - 1700: **Session 4** - *Coding*

Data-Driven Lens

"Data is a clue to the End Truth"

— Josh Smith

Metaphor

- A start-up providing loans to the consumer
- Running for the last few years
- Now planning to adopt a data-driven lens

What are the **type of questions** you can ask?

Type of Questions

- What is the trend of loan defaults?
- Do older customers have more loan defaults?
- Which customer is likely to have a loan default?
- Why do customers default on their loan?

Type of Questions

- Descriptive
- Inquisitive
- Predictive
- Causal

Data-driven Analytics

- **Descriptive**: Understand Pattern, Trends, Outlier
- **Inquisitive**: Conduct Hypothesis Testing
- **Predictive**: Make a prediction
- **Causal**: Establish a causal link

Prediction Challenge

It's tough to make predictions, especially about the future.

— Yogi Berra

How to make a Prediction?

- **Human Learning:** Make a *Judgement*
- **Machine Programmed:** Create explicit *Rules*
- **Machine Learning:** Learn from *Data*

Machine Learning (ML)

[Machine learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

— *Arthur Samuel*

Machine learning is the study of computer algorithm that improve automatically through experience

— *Tom Mitchell*

Machine Learning: Essense

— A pattern exists

— It cannot be pinned down mathematically

— Have data on it to learn from

"Use a set of observations (data) to uncover an underlying process"

Machine Learning

- Theory
- Paradigms
- Models
- Methods
- Process

Applied ML - Approach

— **Theory**: Understand Key Concepts (Intuition)

— **Paradigms**: Limit to One (Supervised)

— **Models**: Use Two Types (Linear, Trees)

— **Methods**: Apply Key Ones (Validation, Selection)

— **Process**: Code the Approach (Real Examples)

ML Theory: Data Types

- What are the types of data on which we are learning?
- Can you give example of say measuring temperature?

Data Types e.g. Temperature

- **Categorical**
  - *Nominal*: Burned, Not Burned
  - *Ordinal*: Hot, Warm, Cold
- **Continuous**
  - *Interval*: 30 °C, 40 °C, 80 °C
  - *Ratio*: 30 K, 40 K, 50 K

Data Types - Operations

- **Categorical**
  - *Nominal*: = , !=
  - *Ordinal*: =, !=, >, <
- **Continuous**
  - *Interval*: =, !=, >, <, -, % of diff
  - *Ratio*: =, !=, >, <, -, +, %

Case Example

Context: Loan Approval

Customer Application

- **age**: age of the applicant
- **income**: annual income of the applicant
- **year**: no. of years of employment
- **ownership**: type of house owned
- **grade**: credit grade for the applicant

Question - How much loan **amount** to approve?

Historical Data

age	income	years	ownership	grade	amount
31	12252	25.0	RENT	C	2400
24	49200	13.0	RENT	C	10000
28	75000	11.0	OWN	B	12000
27	110000	13.0	MORTGAGE	A	3600
33	24000	10.0	RENT	B	5000

Data Types

- **Categorical**
  - *Nominal*: home owner [rent, own, mortgage]
  - *Ordinal*: credit grade [A > B > C > D > E]
- **Continuous**
  - *Interval*: approval date [20/04/16, 19/11/15]
  - *Ratio*: loan amount [3000, 10000]

ML Terminology

Features: $\mathbf{x}$

- age, income, years, ownership, grade

Target: y

- amount

Training Data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n)$

- historical records

ML Paradigm: Supervised

Given a set of **feature** $\mathbf{x}$, to predict the value of **target** y

Learning Paradigm: **Supervised**

- If y is *continuous* - **Regression**
- If y is *categorical* - **Classification**

ML Theory: Formulation

- **Features** $\mathbf{x}$ *(customer application)*
- **Target** y *(loan amount)*
- **Target Function** $f : \mathcal{X} \to y$ (ideal formula)
- **Data** $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n)$ *(historical records)*
- **Final Hypothesis** $g : \mathcal{X} \to y$ (formula to use)
- **Hypothesis Set** $\mathcal{H}$ (all possible formulas)
- **Learning Algorithm** $\mathcal{A}$ (how to learn the formula)

ML Theory: Formulation

unknown target function

$f : \mathcal{X} \to \mathcal{Y}$

training data

$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \dots (\mathbf{x}_n, y_n)$

hypothesis set $\mathcal{H}$ $\rightarrow$ learning algorithm $\mathcal{A}$

final hypothesis

$g \to f$

ML Theory: Learning Model

The Learning Model is composed of the two elements

- The Hypothesis Set: $\mathcal{H} = \{h\}$ $\quad g \in \mathcal{H}$
- Learning Algorithm: $\mathcal{A}$

ML Theory: Formulation (Simplified)

unknown target function

$$y = f(\mathbf{x})$$

training data

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \dots (\mathbf{x}_n, y_n)$$

hypothesis set $\{h(\mathbf{x})\}$ $\rightarrow$ learning algorithm $\mathcal{A}$

final hypothesis

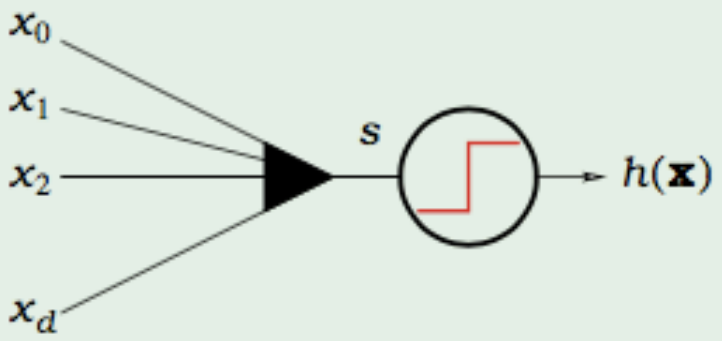
$$g(\mathbf{x}) \rightarrow f(\mathbf{x})$$

Linear Algorithms

$$s = \sum_{i=1}^{d} w_i x_i$$

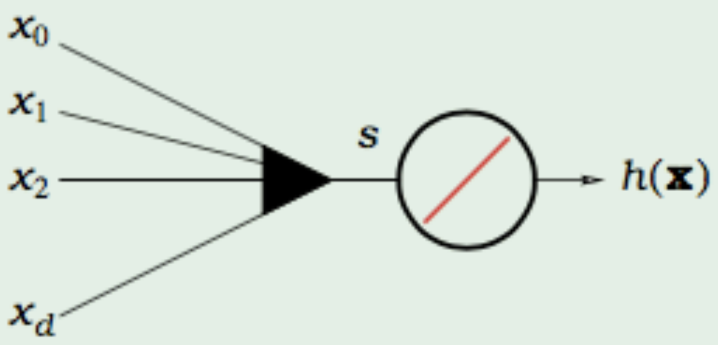
linear classification

$$h(\mathbf{x}) = \text{sign}(s)$$



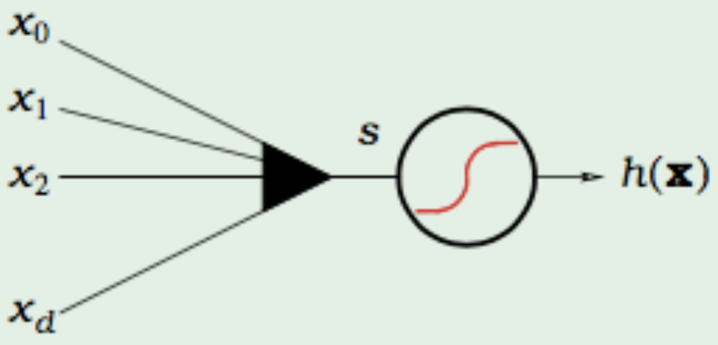
linear regression

$$h(\mathbf{x}) = s$$



logistic regression

$$h(\mathbf{x}) = \theta(s)$$



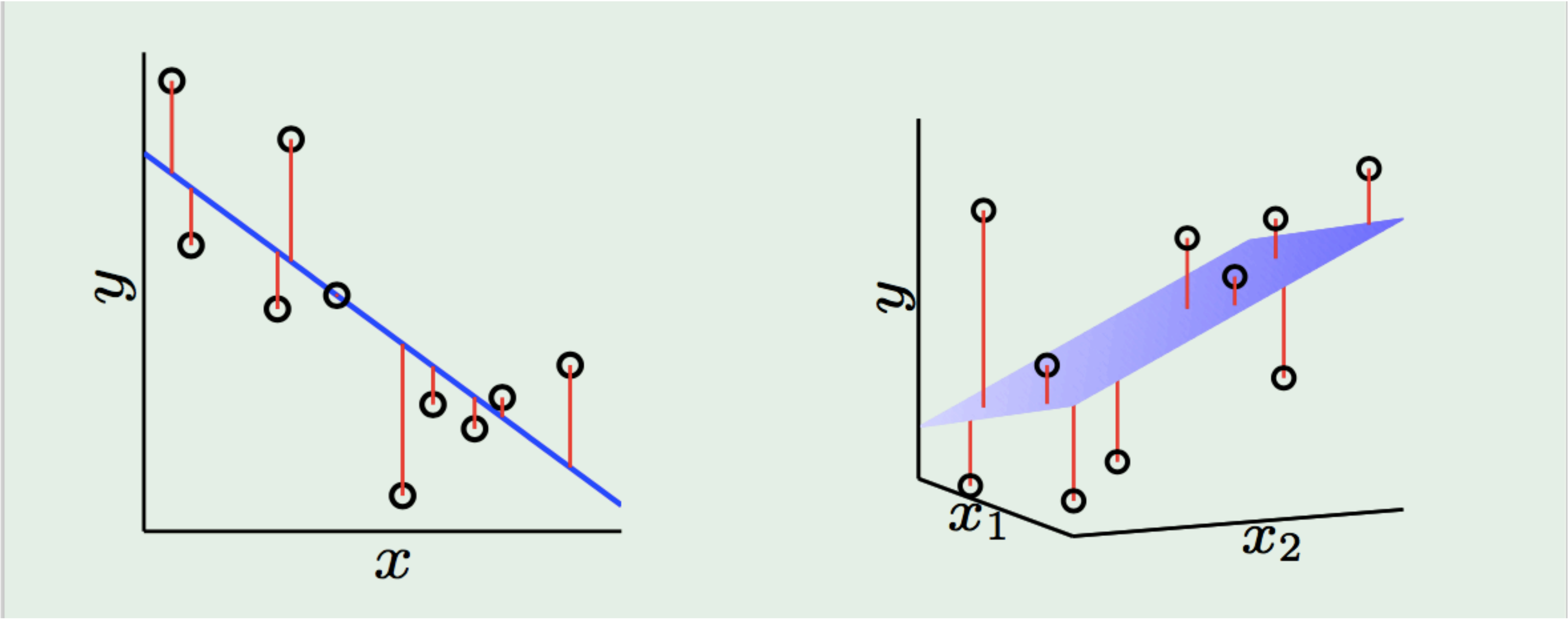
Simple Hypothesis Set: Linear Regression

For d features in training data,

$$h(\mathbf{x}) = \sum_{i=1}^{d} w_i x_i$$

How do we choose the right w_i?

Error



Error Measure - MSE

How well does $h(\mathbf{x})$ approximate to $f(\mathbf{x})$

We will use squared error $(h(\mathbf{x}) - f(\mathbf{x}))^2$

$$E_{in}(h) = \frac{1}{N}\sum_{i=i}^{N}(h(\mathbf{x}) - y_i))^2$$

Learning Algorithm - Linear Regression

- Linear Regression algorithm aims to minimise $E_{in}(h)$
- **One-Step Learning** -> Solves to give $g(\mathbf{x})$

$$g(\mathbf{x}) = \hat{y}$$

$$E_{in}(g) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

Machine Learning Process

— *Frame*: Problem definition

— *Acquire*: Data ingestion

— *Refine*: Data wrangling

— *Transform*: Feature creation

— *Explore*: Feature selection

— *Model*: Model creation & assessment

— *Insight*: Communication

Frame

Variables

- age, income, years, ownership, grade, amount, default and interest

— What are the **Features**: $\mathbf{x}$?

— What are the **Target**: y

Frame

Features: $\mathbf{x}$

- age
- income
- years,
- ownership
- grade,

Target: y

- amount * (1 - default)

Acquire

- Simple! Just read the data from csv file

Refine - Missing Value

- **REMOVE** - NAN rows
- **IMPUTATION** - Replace them with something?
  - Mean
  - Median
  - Fixed Number - Domain Relevant
  - High Number (999) - Issue with modelling
- **BINNING** - Categorical variable and "Missing becomes a category*
- **DOMAIN SPECIFIC** - Entry error, pipeline, etc.

Refine - Outlier Treatment

- What is an outlier?
- Descriptive Plots
  - Histogram
  - Box-Plot
- Measuring
  - Z-score
  - Modified Z-score > 3.5

where modified Z-score = 0.6745 * (x - x_median) / MAD

Explore

- Single Variable Exploration
- Dual Variable Exploration
- Multi Variable Exploration

Transform

Encodings

- One Hot Encoding
- Label Encoding

Feature Transformation

- Log Transform
- Sort Transform

Model - Linear Regression

Parameters

- fit_intercept
- normalization

Error Measure

- mean squared error

Real-World Challenge - Noise

- The "target function" f is not always a *function*
- Not unique target value for same input
- Need to add noise $N(0, \sigma)$

$$y = f(\mathbf{x}) + \epsilon(\mathbf{x})$$

Noise Implication

The best model we can create will have an expected error of σ^2

If Noise (σ) is large, that means feature set does not capture large enough factors in the underlying process

- Need to create **better features**
- Need to find **new features**

When are we learning?

Learning is defined as $g \approx f$, which happens when

(1) Can we make $E_{out}(g)$ is close enough to $E_{in}(g)$?

$$E_{out}(g) \approx E_{in}(g)$$

(1) Can we make $E_{in}(g)$ small enough?

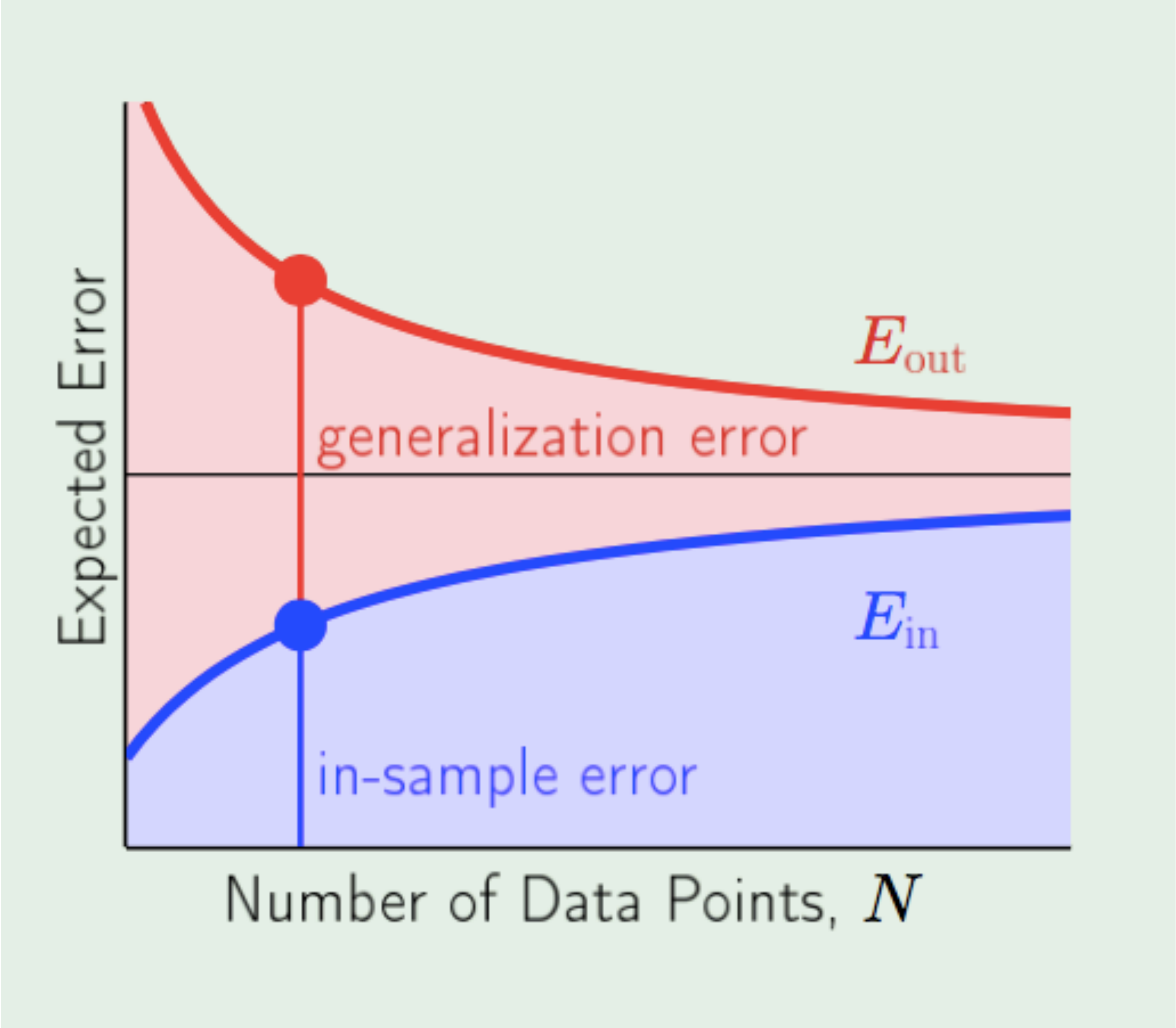
$$E_{in}(g) \approx 0$$

ML Theory: Generalisation

For Learning, $E_{out}(g) \approx E_{in}(g)$

To find the generalisation error, we need to split our data into training and test samples

Given large N, the expected generalisation error should be zero



ML Theory: Generalisation

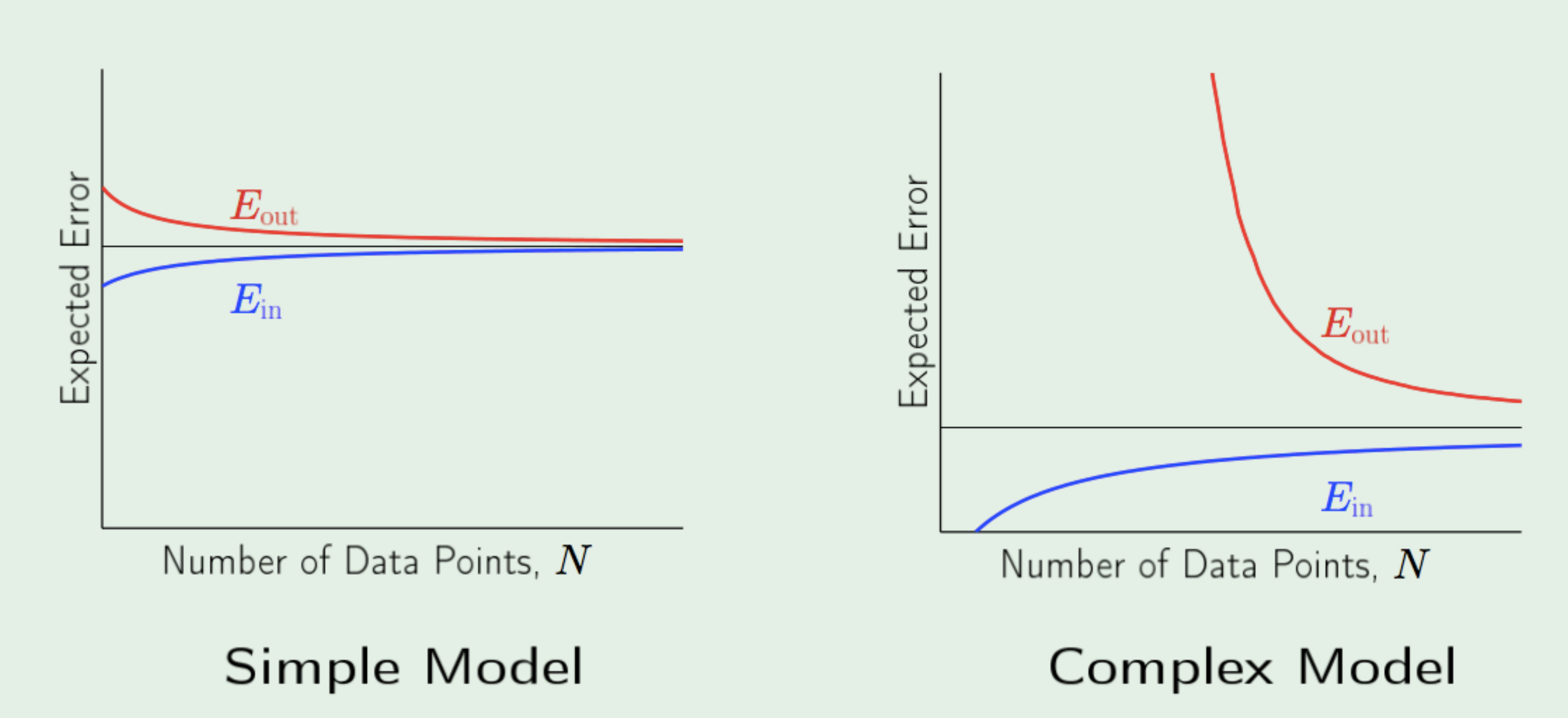
For Learning, $E_{in}(g) \approx 0$

Complex Model: Better chance of approximating f

Simple Model: Better chance of generalising E_{out}

Lets try by increasing the model complexity - More features through interaction effect

ML Theory: Model Complexity

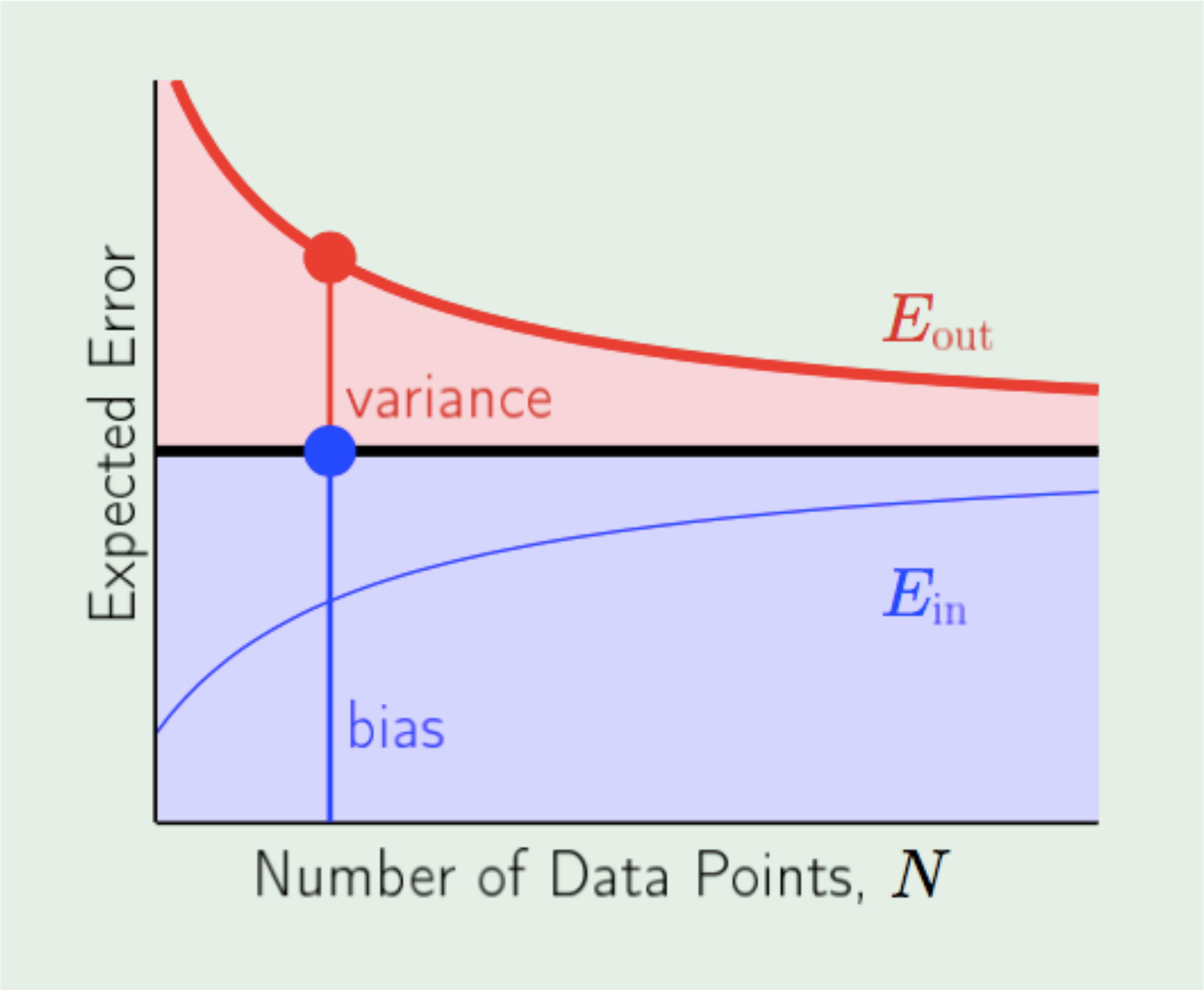


ML Theory: Bias-Variance

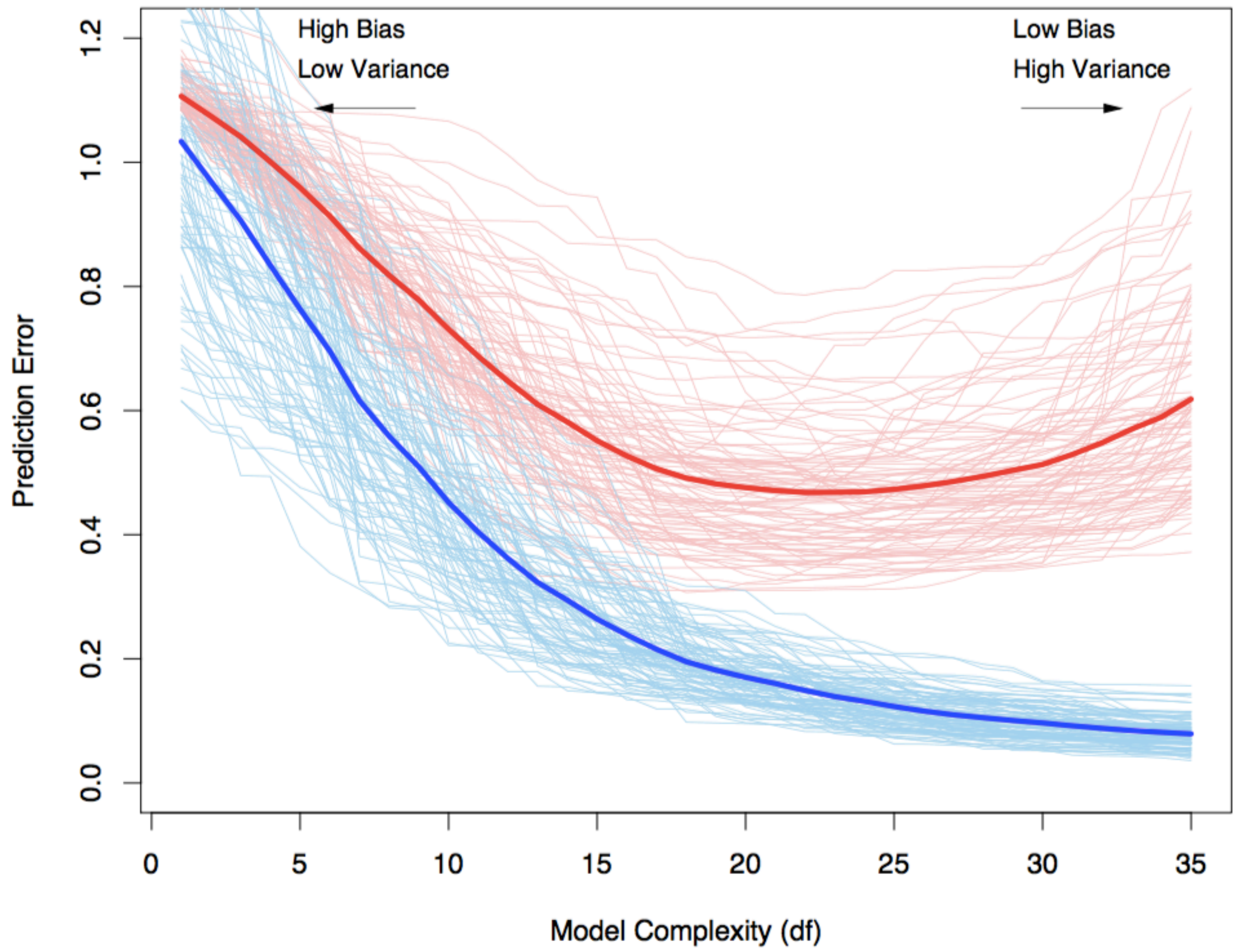
For Learning, $E_{in}(g) \approx 0$

Given large N, the expected error should be the bias

- **Bias** are the simplifying assumptions made by a model to make the target function easier to learn.
- **Variance** is the amount that the estimate of the target function will change if different training data was used.



ML Theory: Bias-Variance Tradeoff



ML Theory: Overfitting

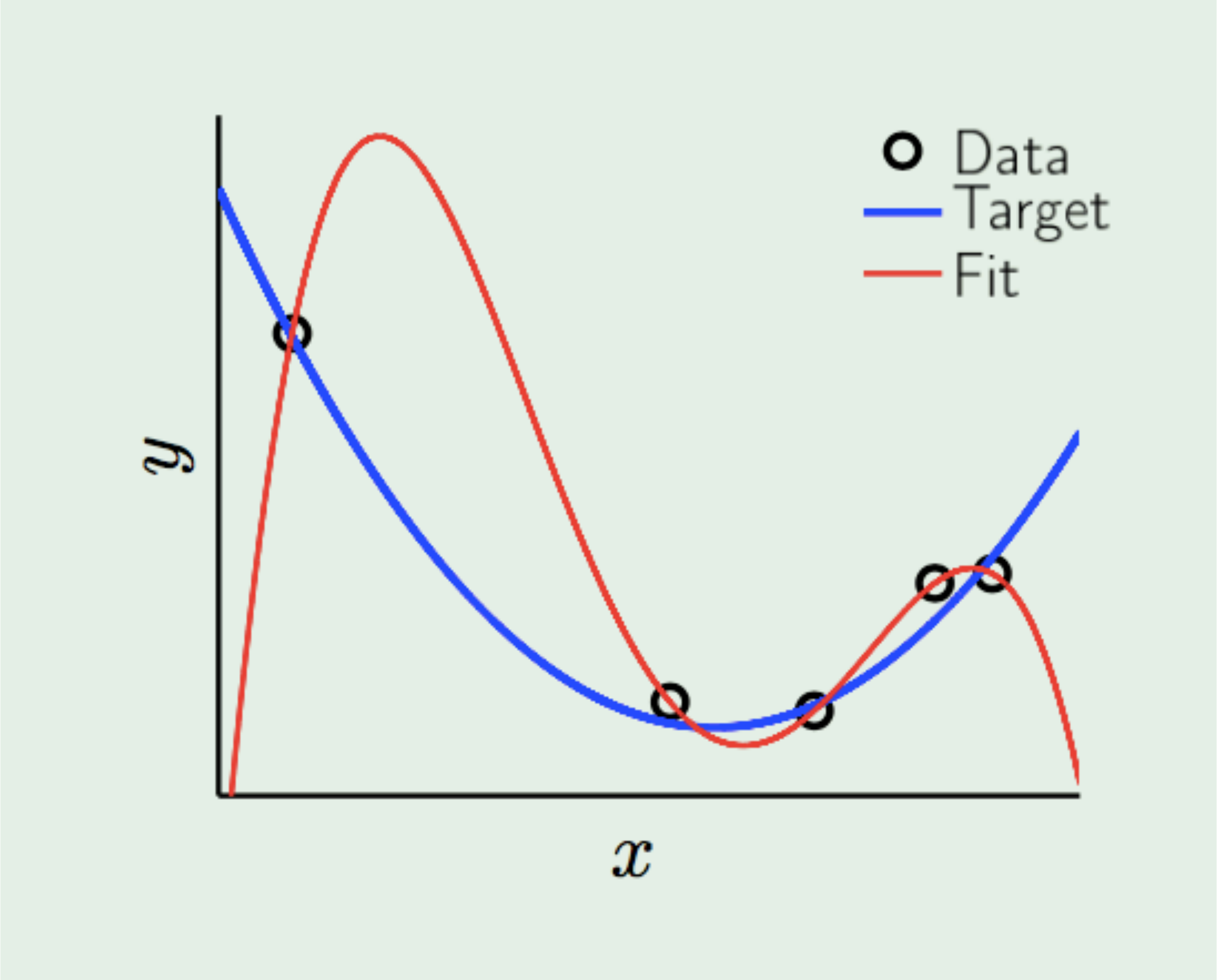
— Simple Target Function

— 5th data point - noisy

— 4th order polynomial fit

$$E_{in} = 0, \; E_{out} \text{ is large}$$

Overfitting - Fitting the data more than warranted, and hence **fitting the noise**



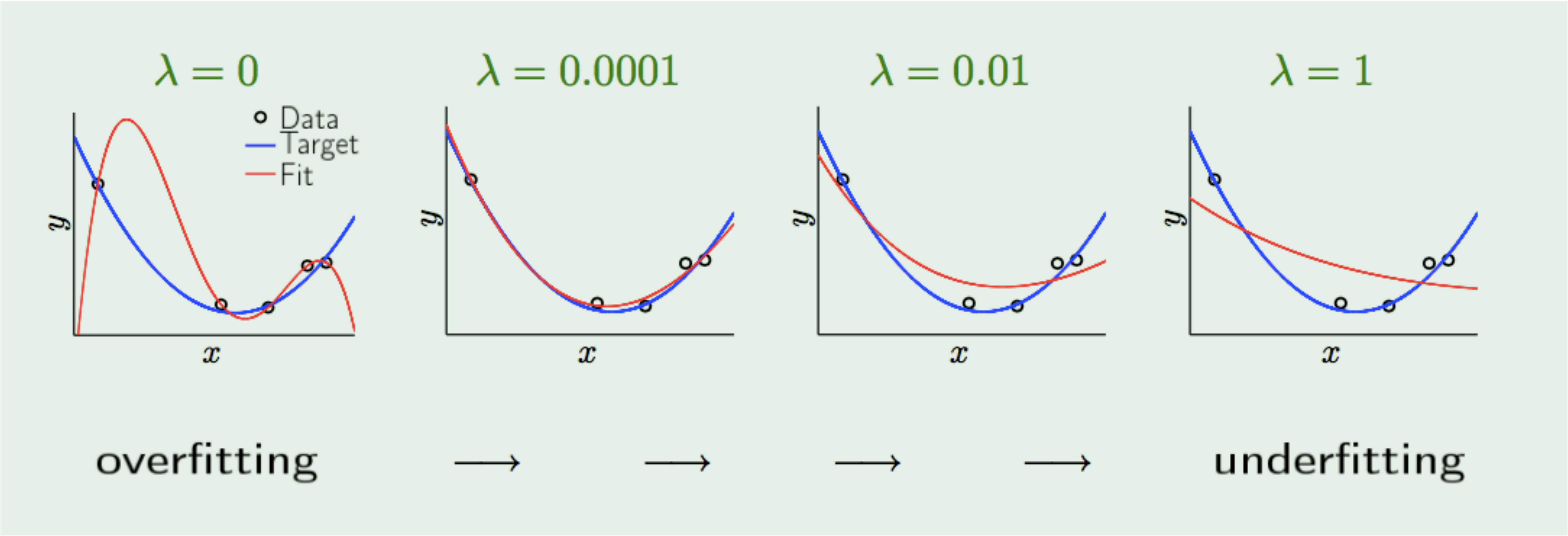
ML Theory: Addressing Overfitting

$$E_{out}(h) = E_{in}(h) + \textbf{overfit penalty}$$

- **Regularization:** Not letting the weights grow
  - Ridge: add $||w||^2$ to error minimisation
  - Lasso: add $||w||$ to error minimisation
- **Validation:** Checking when we reach bottom point

Regularization - Ridge

$$Minimize \quad E_{in}(w) + \frac{\lambda}{N}||w||^2$$



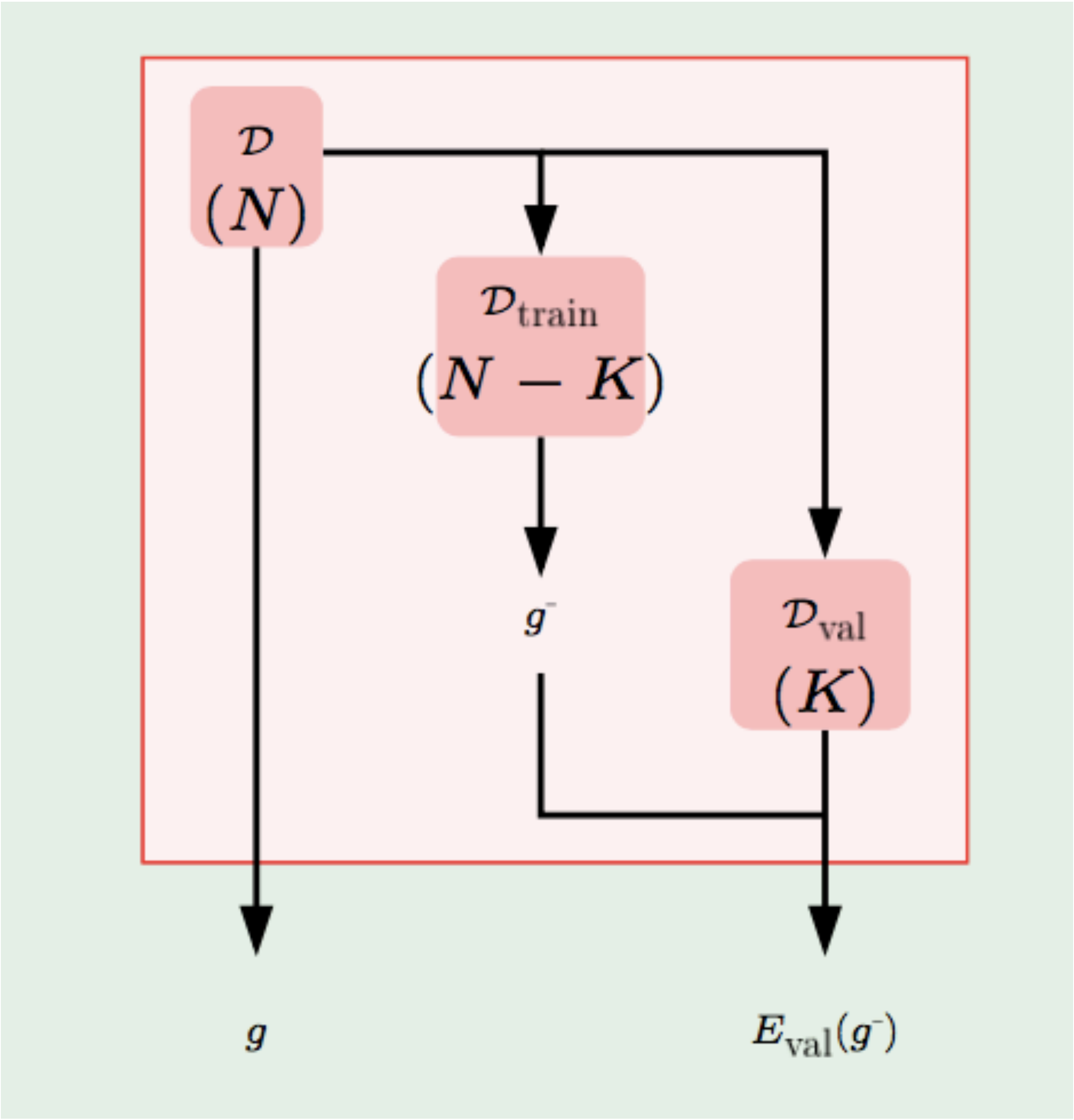
Validation

Validation set: K

Training set: $N - K$

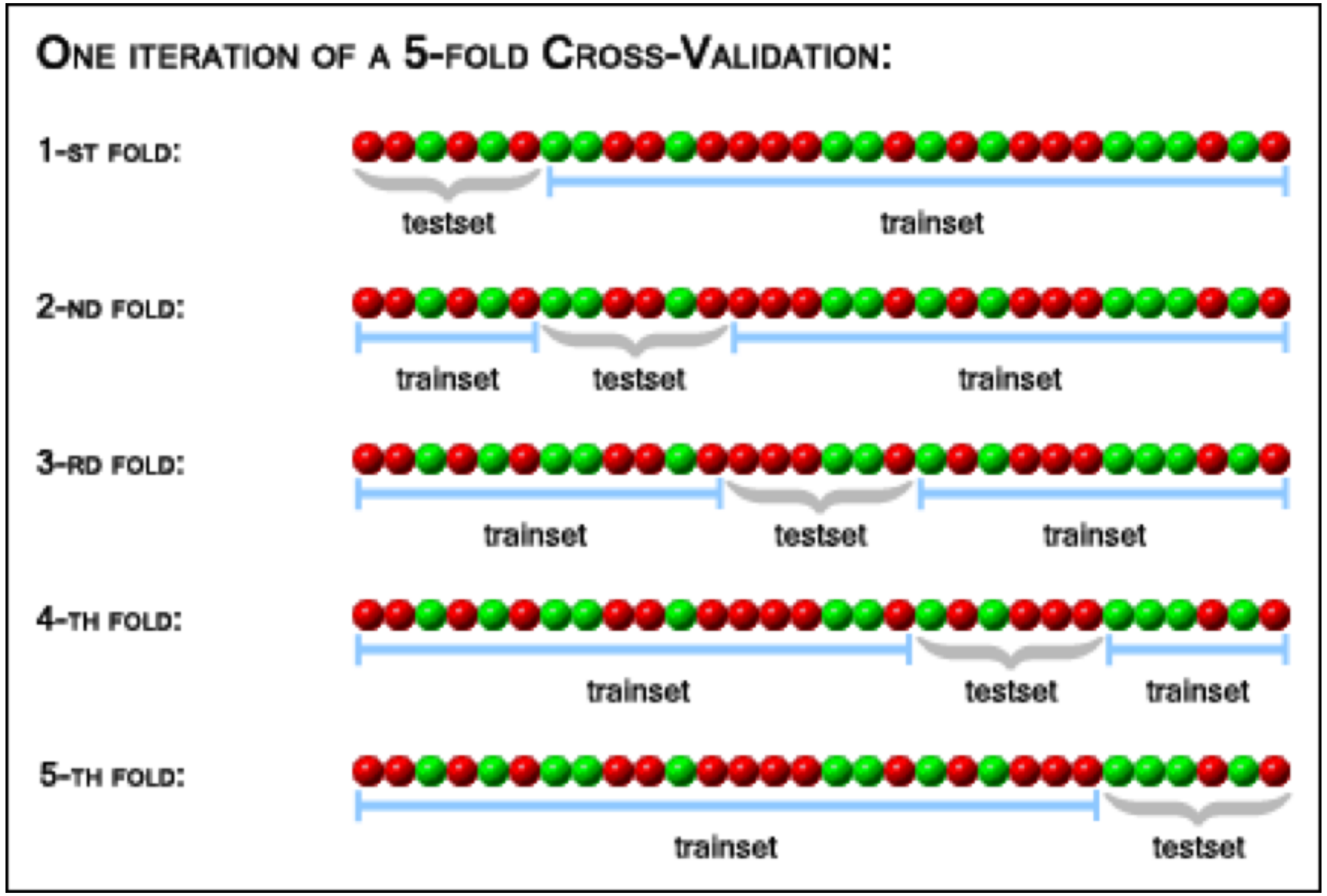
Rule of Thumb: $N = \frac{K}{5}$

Note: The validation set is used for learning



Cross Validation

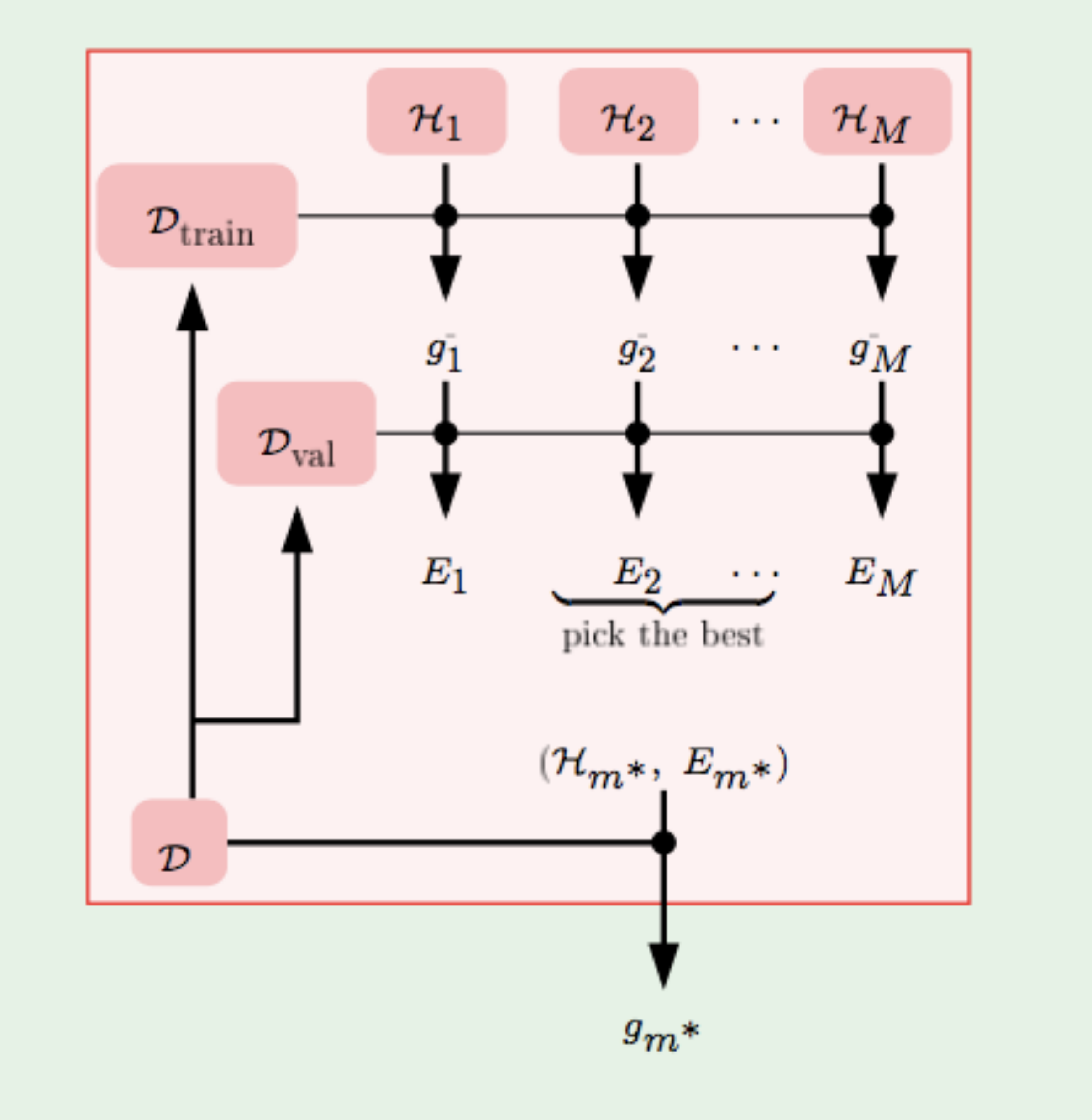
Repeats the process 5-times



Model Selection

How to choose between competing model?

Choose the function g_m with lowest cross-validation error E_m



Applied ML

- **Theory**: Formulation, Generalisation, Bias-Variance, Overfitting
- **Paradigms**: Supervised - Regression
- **Models**: Linear - OLS, Ridge, Lasso
- **Methods**: Regularisation, Validation
- **Process**: Frame, Acquire, Refine, Transform, Explore, Model

Classification Problem

Context: Loan Default

Customer Application
- **age**: age of the applicant
- **income**: annual income of the applicant
- **year**: no. of years of employment
- **ownership**: type of house owned
- **grade**: credit grade for the applicant
- **amount**: loan amount given
- **interest**: interest rate of loan

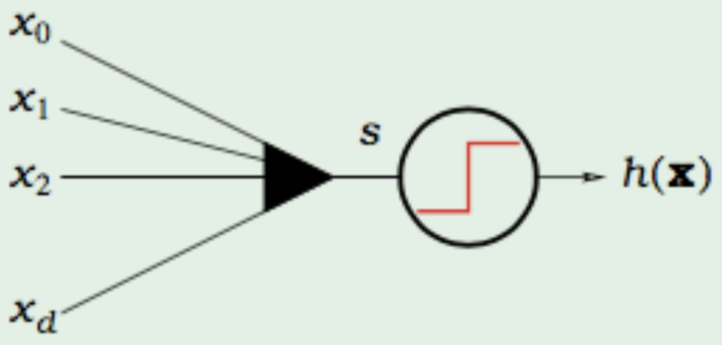
Question - Who is likely to **default**?

Linear Models

$$s = \sum_{i=1}^{d} w_i x_i$$

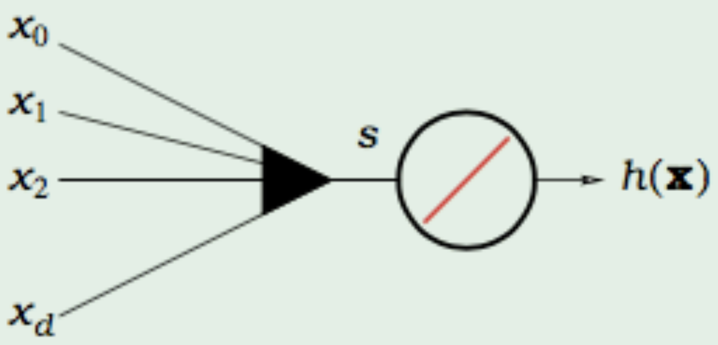
linear classification

$$h(\mathbf{x}) = \text{sign}(s)$$



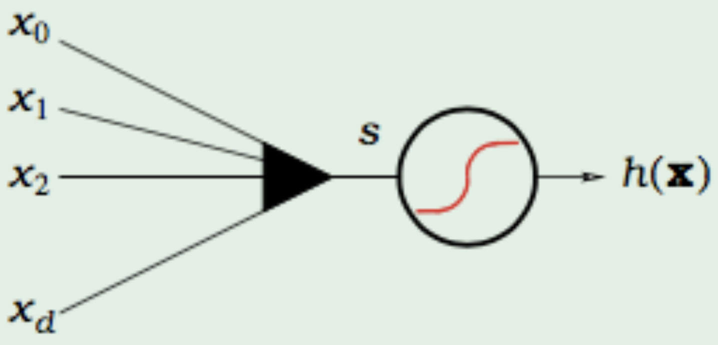
linear regression

$$h(\mathbf{x}) = s$$



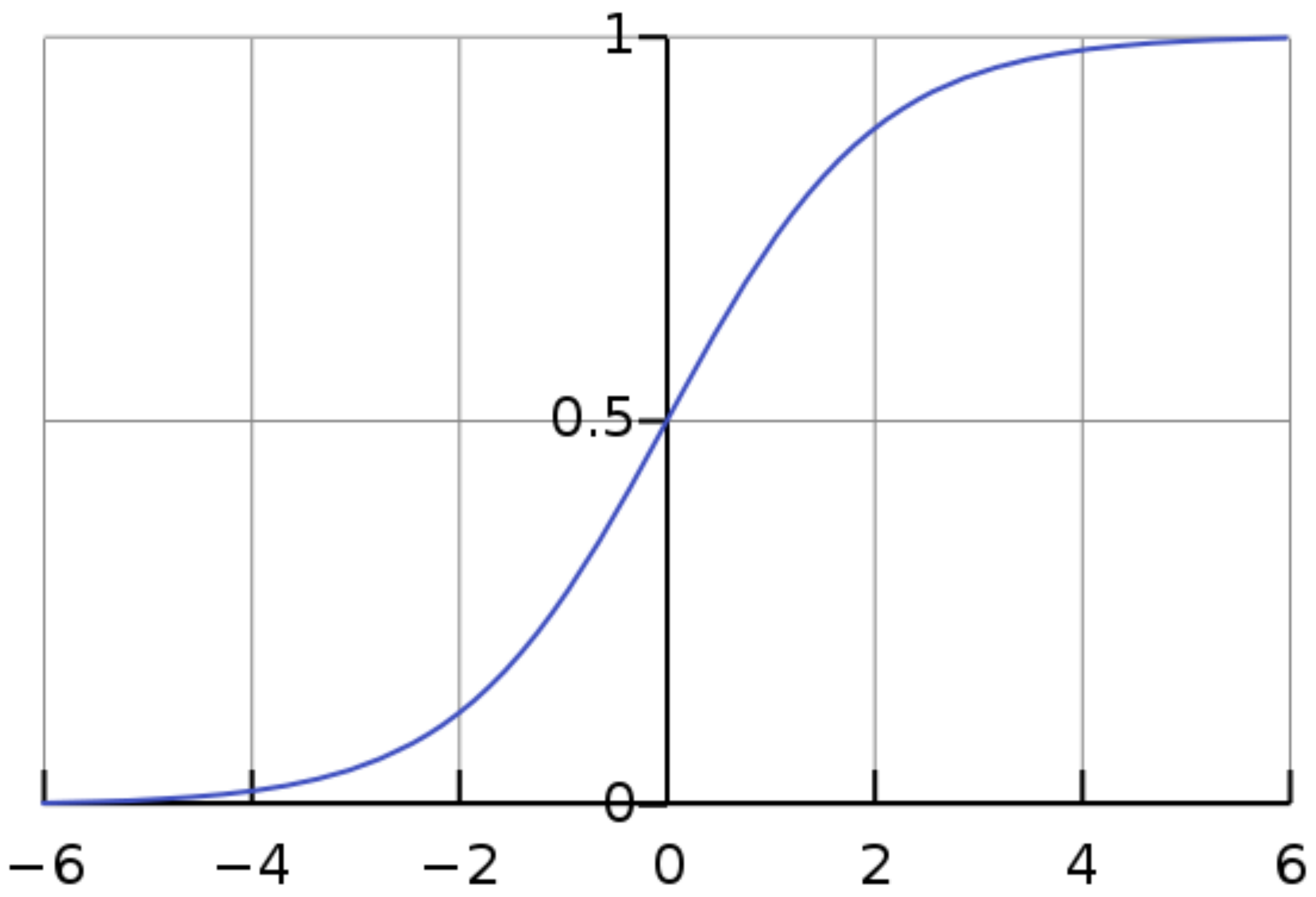
logistic regression

$$h(\mathbf{x}) = \theta(s)$$



Logit Function

$$\theta(s) = \frac{e^s}{e^s + 1} = \frac{1}{1 + e^{-s}}$$



Logistic Relationship

Find the w_i weights that best fit:

$y = 1$ if $\displaystyle\sum_{i=1}^{d} w_i x_i > 0$

$y = 0$, otherwise

Follows:

$$\theta(y_i) = \frac{1}{1 + e^{-\left(\sum_{i=1}^{d} w_i x_i\right)}}$$

Error - Likelihood / Probabilities

Where, $h(\mathbf{x}) = \sum_{i=1}^{d} w_i x_i$

Minimise the **log-likelihood** values

$$E(\mathbf{h}) = -\frac{1}{N} ln \left(\prod_{i=1}^{N} \theta(y_i h(\mathbf{x})) \right)$$

Learning Algorithm - Logistic

- Logistic Regression algorithm aims to minimise $E_{in}(h)$
- **Iterative Method** -> Solves to give $g(\mathbf{x})$

$$g(\mathbf{x}) = \hat{y}$$

$$E_{in}(g) = \frac{1}{N}\sum_{i=1}^{N} ln(1 + e^{-y_i\hat{y}_i})$$

Error Metric - Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

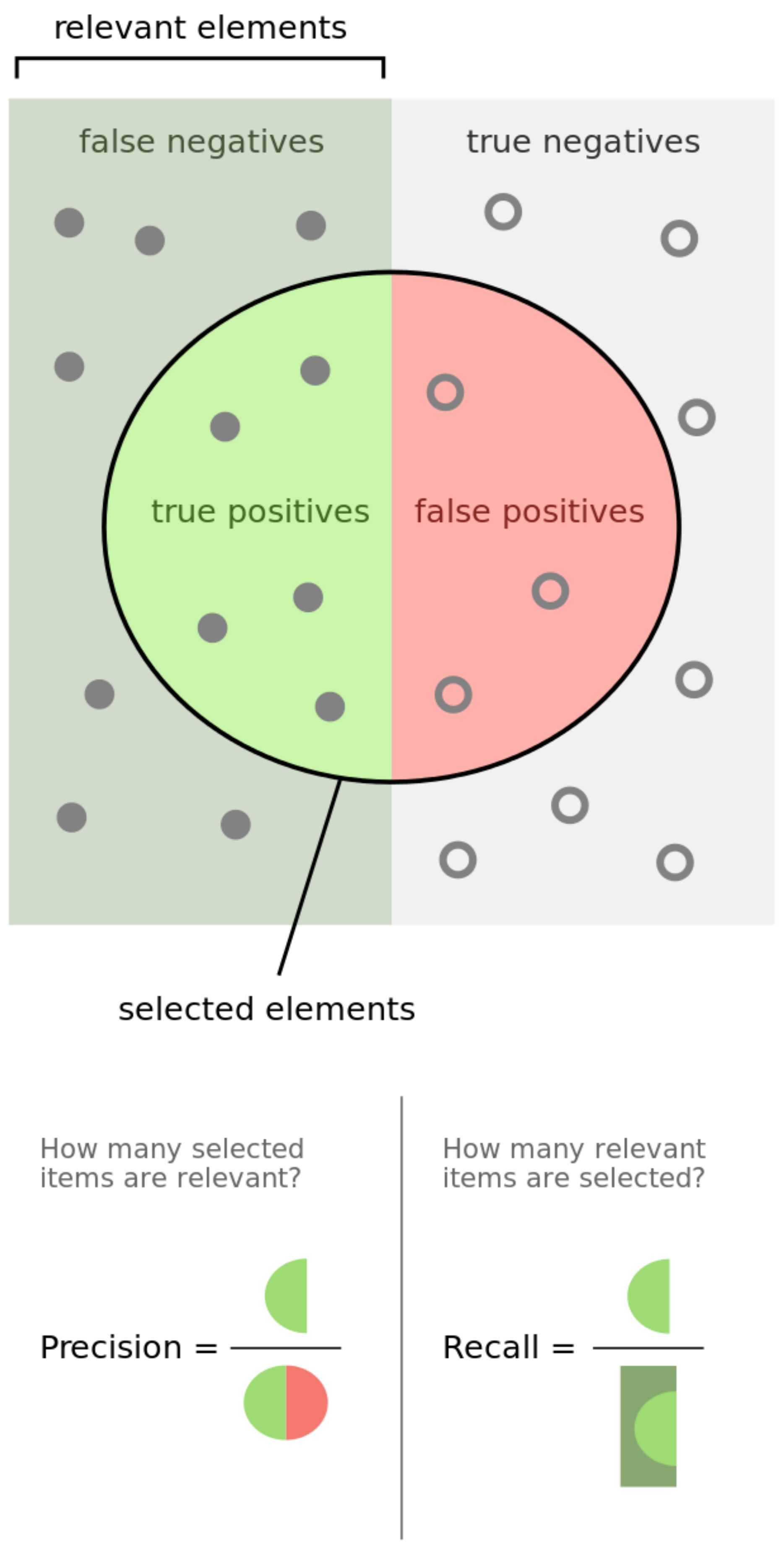
Model Evaluation

Classification Metrics

Recall (TPR) = TP / (TP + FN)

Precision = TP / (TP + FP)

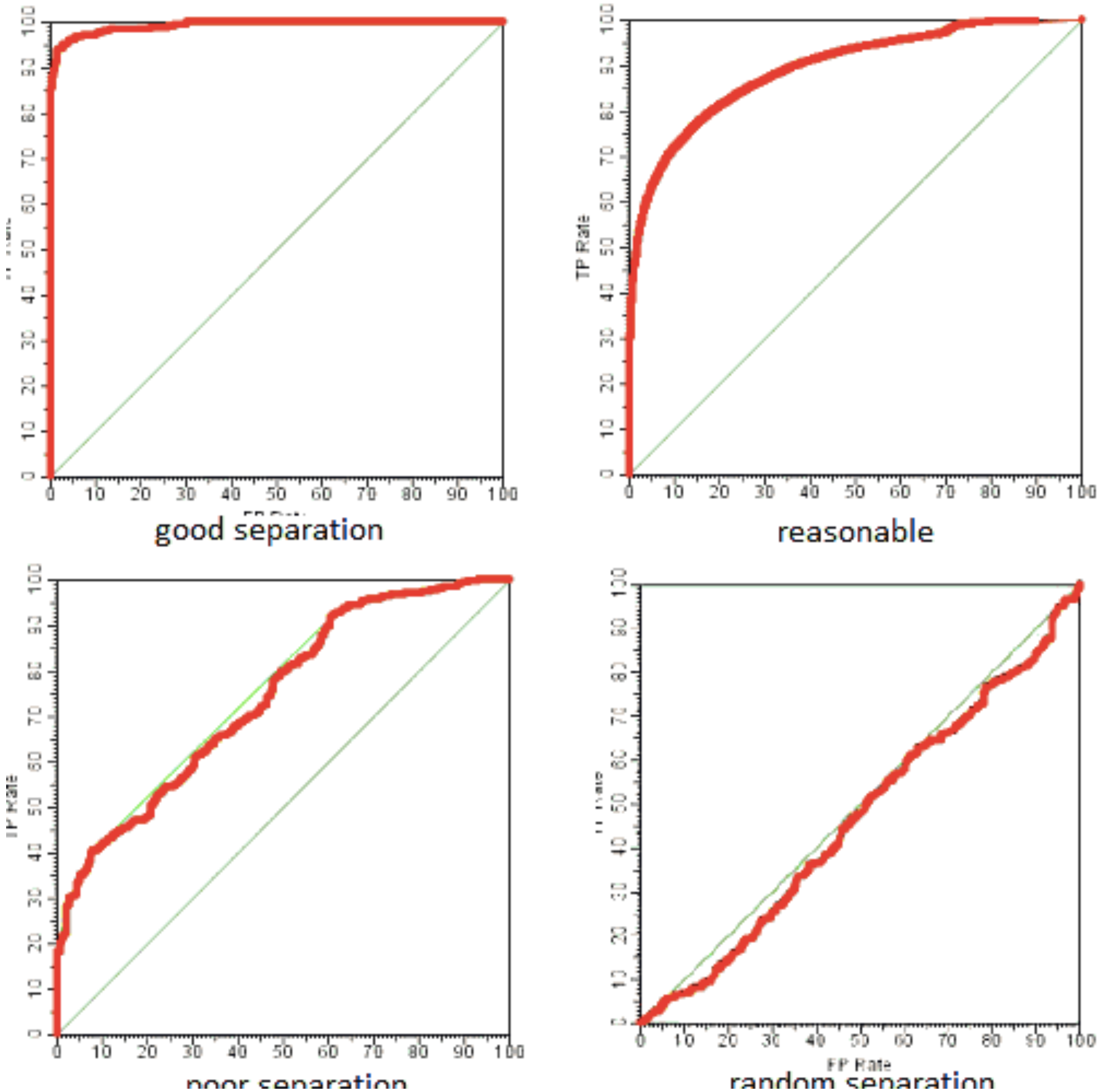
Specificity (TNR) = TN / (TN + FP)



Model Evaluation

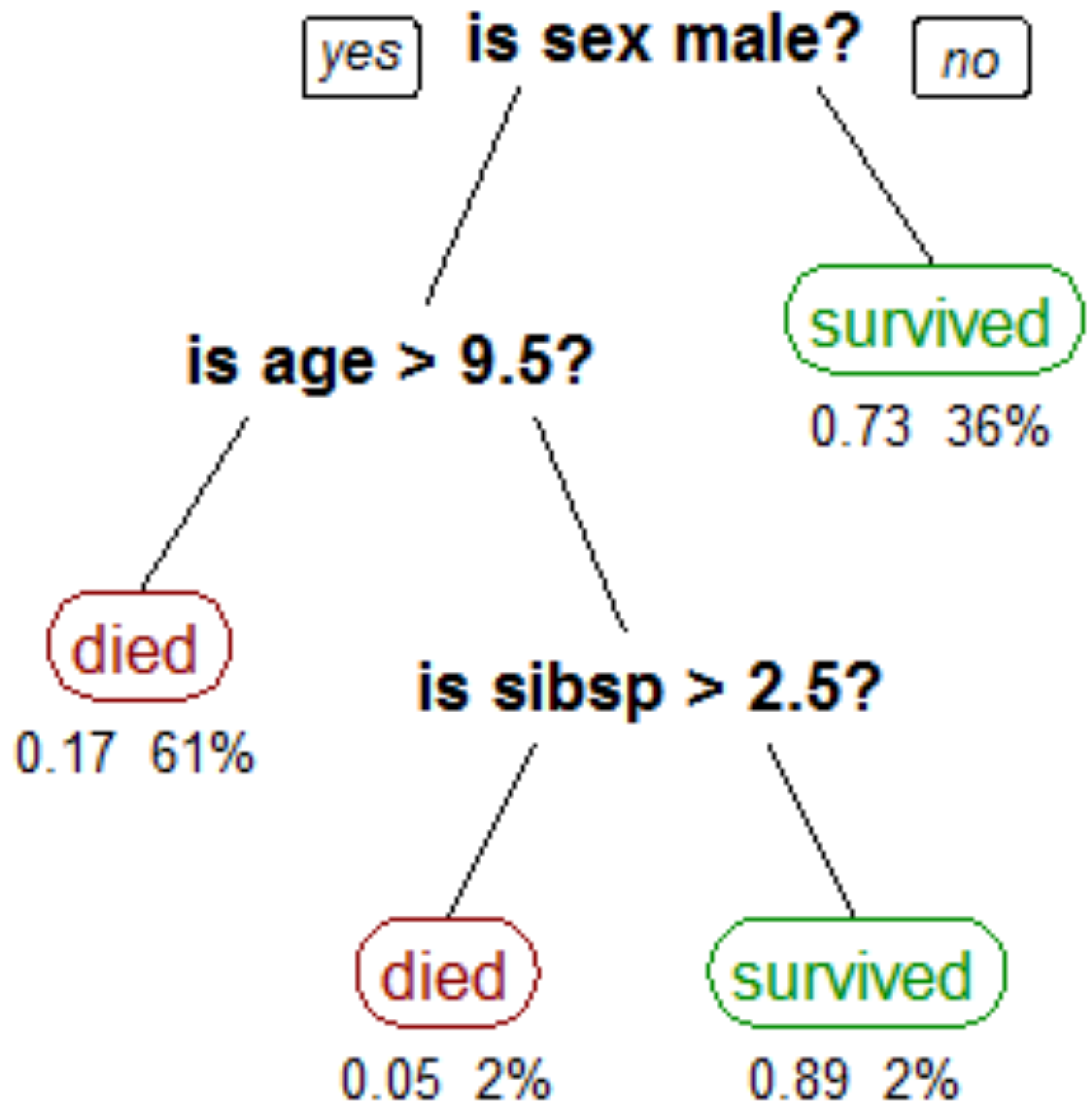
Receiver Operating Characteristic Curve

Plot of TPR vs FPR at different discrimination threshold



Decision Tree

Example: Survivor on Titanic



Decision Tree

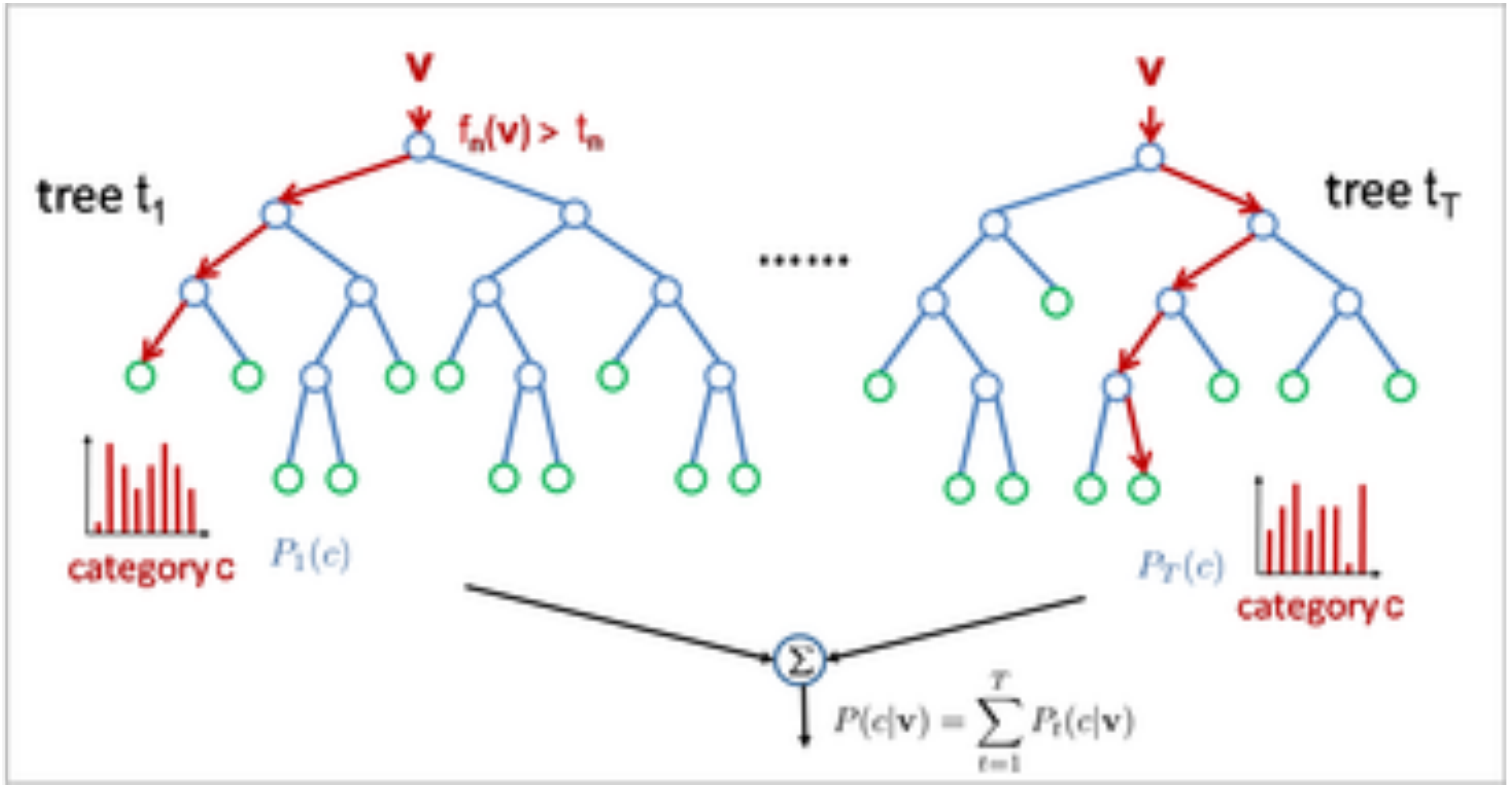
- Easy to interpret
- Little data preparation
- Scales well with data
- White-box model
- Instability – changing variables, altering sequence
- Overfitting

Bagging

- Also called bootstrap aggregation, reduces variance
- Uses decision trees and uses a model averaging approach

Random Forest

- Combines bagging idea and random selection of features.
- Similar to decision trees are constructed – but at each split, a random subset of features is used.



If you torture the data enough, it will confess.

— Ronald Case

Challenges

- Data Snooping
- Selection Bias
- Survivor Bias
- Omitted Variable Bias
- Black-box model Vs White-Box model
- Adherence to regulations

Day 1 Coverage

Day 1: Reflections