Lecture 10: Bias - Variance Decomposition

Concept of BIAS and VARIANCE

- Consider an example: $z_0 \leftarrow$ true location of an object

  $z \leftarrow$ noisy GPS measurements of the location

  treated as a random variable

  - it has some mean: $\mathbb{E}[z]$ $\rightarrow \mu_z$
  - it has some variance: $\mathbb{E}[(z-\mu_z)^2]$

μ_z

- Bias describes the systematic error in the measurements z (possible offset)

$$\boxed{\text{BIAS}: \quad \mu_z - z_0}$$

- Variance describes how much the measurements vary (amount of noise in GPS measurements)

$$\boxed{\text{VARIANCE}: \quad \mathbb{E}[(z-\mu_z)^2] = \mathbb{E}[z^2] - \mu_z^2}$$

- Bias : $\mu_z - z_0$

  Variance : $\mathbb{E}[z^2] - \mu_z^2$

- Squared error between measurement and true value : $(z - z_0)^2$

- Expected squared error : $\underbrace{\mathbb{E}\left[(z - z_0)^2\right]}$
  (Averaged)

$$
\begin{aligned}
&= \mathbb{E}\left[\left((z - \mu_z) + (\mu_z - z_0)\right)^2\right] \\
&= \mathbb{E}\left[(z - \mu_z)^2\right] + \mathbb{E}\left[(\underbrace{\mu_z}_{\text{constant}} - \underbrace{z_0}_{\text{constant}})^2\right] + 2\,\mathbb{E}\left[(z - \underbrace{\mu_z}_{\text{constant}})(\underbrace{\mu_z - z_0}_{\text{constant}})\right] \\
&= \underbrace{\mathbb{E}\left[(z - \mu_z)^2\right]}_{\text{variance}} + \underbrace{(\mu_z - z_0)^2}_{\text{bias}} + 2\,(\mu_z - z_0)\underbrace{\left(\mathbb{E}[z] - \mu_z\right)}_{0}
\end{aligned}
$$

- In other words, the averaged squared error between z and z_0 is the sum of the squared bias and variance

- To obtain a small expected squared error, we have to consider both → bias, → variance

- We will now apply the bias-variance concept to a regression setting
    - z_0 will now correspond to the true relationship between inputs and outputs
    - random variable z will correspond to the model learned from training data

      since training data $\mathcal{T}$ is a random collection from $p(\underline{x}, y)$, the model z learned from it is also random as it is a function of training data $\mathcal{T}$

- Let the true relationship between input $\underline{x}$ and output y be described by some function $f_0(\underline{x})$ plus i.i.d. noise ϵ

$$y = f_0(\underline{x}) + \epsilon, \qquad \text{with} \quad \mathbb{E}[\epsilon] = 0$$
$$\mathrm{Var}[\epsilon] = \sigma^2$$

- Learned model is random variable; therefore model prediction $\hat{y}(\underline{x}; \mathcal{T})$ is r.v.!
- Define average trained model corresponding to $\bar{z}$:

$$\bar{f}(\underline{x}) = \mathbb{E}_{\mathcal{T}}[\hat{y}(\underline{x}; \mathcal{T})]$$

- Let the true relationship between input $\underline{x}$ and output y be described by some function $f_0(\underline{x})$ plus i.i.d. noise ϵ

True model

$$y = \underbrace{f_0(\underline{x})}_{\text{'}z_0\text{'}} + \epsilon, \qquad \text{with} \quad \mathbb{E}[\epsilon] = 0, \quad \mathrm{Var}[\epsilon] = \sigma^2$$

- The learned model is a r.v.; therefore model prediction $\hat{y}(\underline{x}; T)$ is r.v.!

- Define average trained model corresponding to $\bar{z}$:

$$\bar{f}(\underline{x}) = \underbrace{\mathbb{E}_T}\big[\underbrace{\hat{y}(\underline{x}; T)}_{\text{'}z\text{'}}\big]$$

average model we would achieve if we could re-train the model infinite # of times on different training datasets, each of size N

expected value over N training points drawn from $p(\underline{x}, y)$

- Recall the definition of E_{new} (for regression with squared error)

$$E_{new} = \mathbb{E}_* \left[\left(y^* - \hat{y}(\underline{x}^*; \mathcal{T}) \right)^2 \right]$$

$$\bar{E}_{new} = \mathbb{E}_{\mathcal{T}} \left[E_{new} \right] = \mathbb{E}_{\mathcal{T}} \left[\mathbb{E}_* \left[\left(y^* - \hat{y}(\underline{x}^*; \mathcal{T}) \right)^2 \right] \right]$$

- Change the order of integration

$$\bar{E}_{new} = \mathbb{E}_* \left[\mathbb{E}_{\mathcal{T}} \left[\left(y^* - \hat{y}(\underline{x}^*; \mathcal{T}) \right)^2 \right] \right]$$

replace $y^* = f_0(\underline{x}^*) + \epsilon$

$$= \mathbb{E}_* \left[\mathbb{E}_{\mathcal{T}} \left[\left(f_0(\underline{x}^*) + \epsilon - \hat{y}(\underline{x}^*; \mathcal{T}) \right)^2 \right] \right]$$

$$= \mathbb{E}_* \left[\mathbb{E}_{\mathcal{T}} \left[\left(\hat{y}(\underline{x}^*; \mathcal{T}) - f_0(\underline{x}^*) - \epsilon \right)^2 \right] \right]$$

$$= \mathbb{E}_* \left[\mathbb{E}_{\mathcal{T}} \left[\left(\hat{y}(\underline{x}^*; \mathcal{T}) - \bar{f}(\underline{x}^*) + \bar{f}(\underline{x}^*) - f_0(\underline{x}^*) - \epsilon \right)^2 \right] \right]$$

$$\bar{f}(\underline{x}) = \mathbb{E}_{\mathcal{T}} \left[\hat{y}(\underline{x}; \mathcal{T}) \right]$$

$$E_{new} = \mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[\left(\underbrace{\hat{y}(\underline{x}^*;\mathcal{T}) - \bar{f}(\underline{x}^*)}_{A_1} + \underbrace{\bar{f}(\underline{x}^*) - f_0(\underline{x}^*)}_{A_2} - \underbrace{\epsilon}_{A_3}\right)^2\right]\right]$$

$$= \mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[(A_1 + A_2 - A_3)^2\right]\right] = \mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[A_1^2 + A_2^2 + A_3^2 + 2(A_1A_2 + A_2A_3 + A_3A_1)\right]\right]$$

$$\mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}[A_1 A_2]\right] = \mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[\left(\hat{y}(\underline{x}^*;\mathcal{T}) - \bar{f}(\underline{x}^*)\right)\left(\bar{f}(\underline{x}^*) - f_0(\underline{x}^*)\right)\right]\right]$$

$$= \mathbb{E}_{*}\left[\left(\bar{f}(\underline{x}^*) - f_0(\underline{x}^*)\right)\left(\underbrace{\mathbb{E}_{\mathcal{T}}\left[\hat{y}(\underline{x}^*;\mathcal{T})\right]}_{\bar{f}(\underline{x}^*)} - \bar{f}(\underline{x}^*)\right)\right] = 0$$

$$\mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}[A_2 A_3]\right] = \mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[\left(\bar{f}(\underline{x}^*) - f_0(\underline{x}^*)\right)\epsilon\right]\right] = \mathbb{E}_{*}\left[\left(\bar{f}(\underline{x}^*) - f_0(\underline{x}^*)\right)\underbrace{\mathbb{E}_{\mathcal{T}}[\epsilon]}_{0}\right]$$

$$= 0$$

$$\mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}[A_3 A_1]\right] = \mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[\epsilon\left(\hat{y}(\underline{x}^*;\mathcal{T}) - \bar{f}(\underline{x}^*)\right)\right]\right]$$

(Noise is independent of model)

$$= \underbrace{\mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}[\epsilon]\right]}_{0} \cdot \mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[\left(\hat{y}(\underline{x}^*;\mathcal{T}) - \bar{f}(\underline{x}^*)\right)\right]\right] = 0$$

- $$\mathbb{E}_*\left[\mathbb{E}_\mathcal{T}\left[A_1^2\right]\right] = \underbrace{\mathbb{E}_*\left[\mathbb{E}_\mathcal{T}\left[\Big(\underbrace{\hat{y}(\underline{x}^*;\mathcal{T})}_{`z'} \quad \underbrace{\bar{f}(\underline{x}^*)}_{`\mu_z'}\Big)^2\right]\right]}_{\text{Variance}} \qquad \left(\mathbb{E}\left[(z-\mu_z)^2\right]\right)$$

prediction

describes how much $\hat{y}(\underline{x};\mathcal{T})$ varies each time the model is trained on a different training dataset

- $$\mathbb{E}_*\left[\mathbb{E}_\mathcal{T}\left[A_2^2\right]\right] = \mathbb{E}_*\left[\mathbb{E}_\mathcal{T}\left[\left(\bar{f}(\underline{x}^*) - f_0(\underline{x}^*)\right)^2\right]\right]$$

$$= \underbrace{\mathbb{E}_*\left[\Big(\underbrace{\bar{f}(\underline{x}^*)}_{`\mu_z'} - \underbrace{f_0(\underline{x}^*)}_{`z_0'}\Big)^2\right]}_{\text{Bias}^2} \qquad \left((\mu_z-\mu_0)^2\right)$$

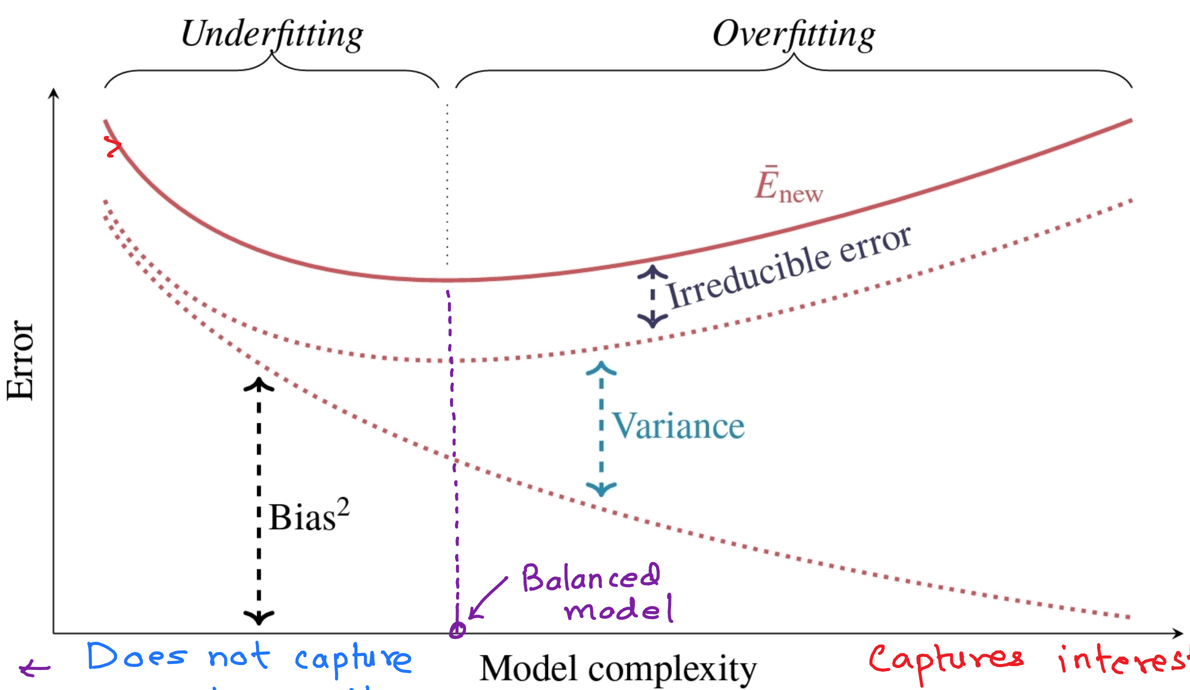
describes how much the average trained model $\bar{f}(\underline{x}^*)$ differs from the true $f_0(\underline{x}^*)$

- $$\mathbb{E}_*\left[\mathbb{E}_\mathcal{T}\left[A_3^2\right]\right] = \mathbb{E}_*\left[\mathbb{E}_\mathcal{T}\left[\epsilon^2\right]\right] = \mathbb{E}_*\left[\cancelto{\sigma^2}{Var(\epsilon)} + \cancelto{0^2}{\mu_\epsilon^2}\right] = \mathbb{E}_*\left[\sigma^2\right] = \underbrace{\sigma^2}_{\text{Irreducible error}}$$

BIAS- VARIANCE TRADE-OFF

$$\bar{E}_{new} = \underbrace{\mathbb{E}_{*}\left[\left(\bar{f}(\underline{x}^{*}) - f_0(\underline{x}^{*})\right)^2\right]}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{*}\left[\mathbb{E}_{\mathcal{T}}\left[\left(\hat{y}(\underline{x}^{*};\mathcal{T}) - \bar{f}(\underline{x}^{*})\right)^2\right]\right]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible error}}$$

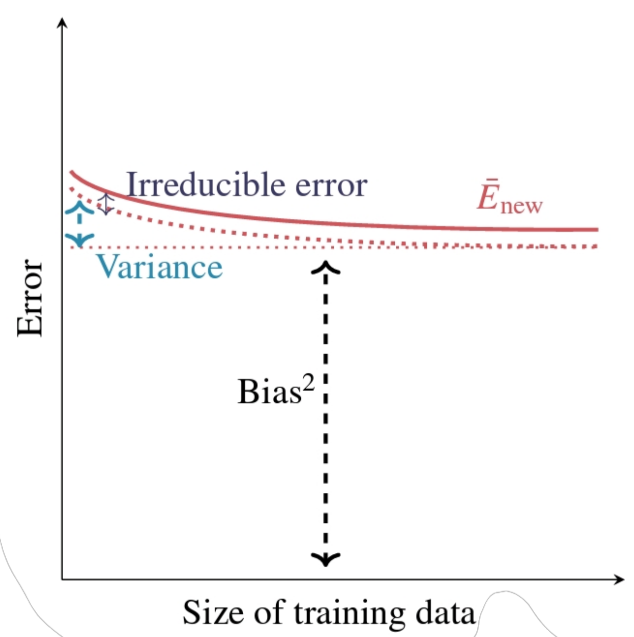
- For the bias to be small, the model has to be flexible
- For the variance to be small, the model should not be very sensitive to the data points in the training set

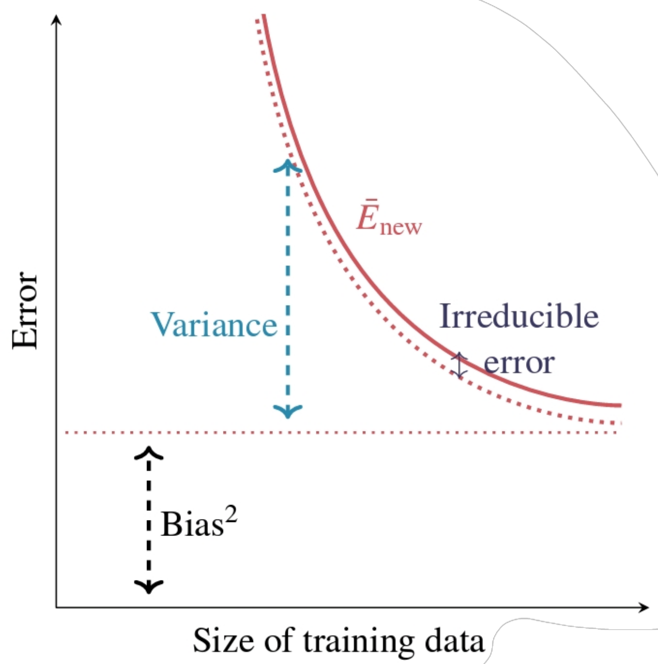


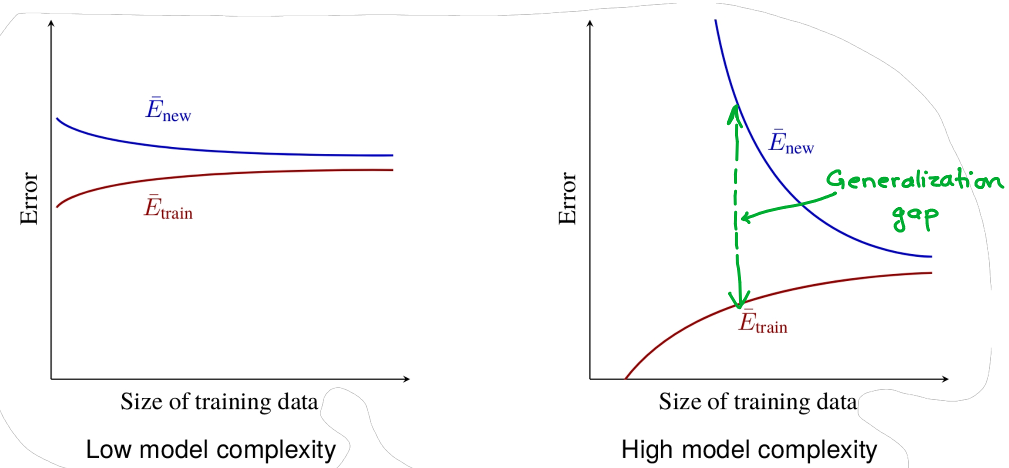
← Does not capture interesting patterns

Captures interesting patterns plus noise →

- We also know that $\bar{E}_{new}$ typically decreases with increasing training data







Intuitively, as the size of training data increases, we have more info about the parameters, hence the variance of prediction reduces!

Example of a simulated problem

Data Generation

- $N = 10$ data points
- Input $x \sim \text{UniformDist}(-5, 10)$
- $y = \min(0.1\, x^2, 3) + \epsilon$
- $\epsilon \sim \text{NormalDist}(0, 1)$

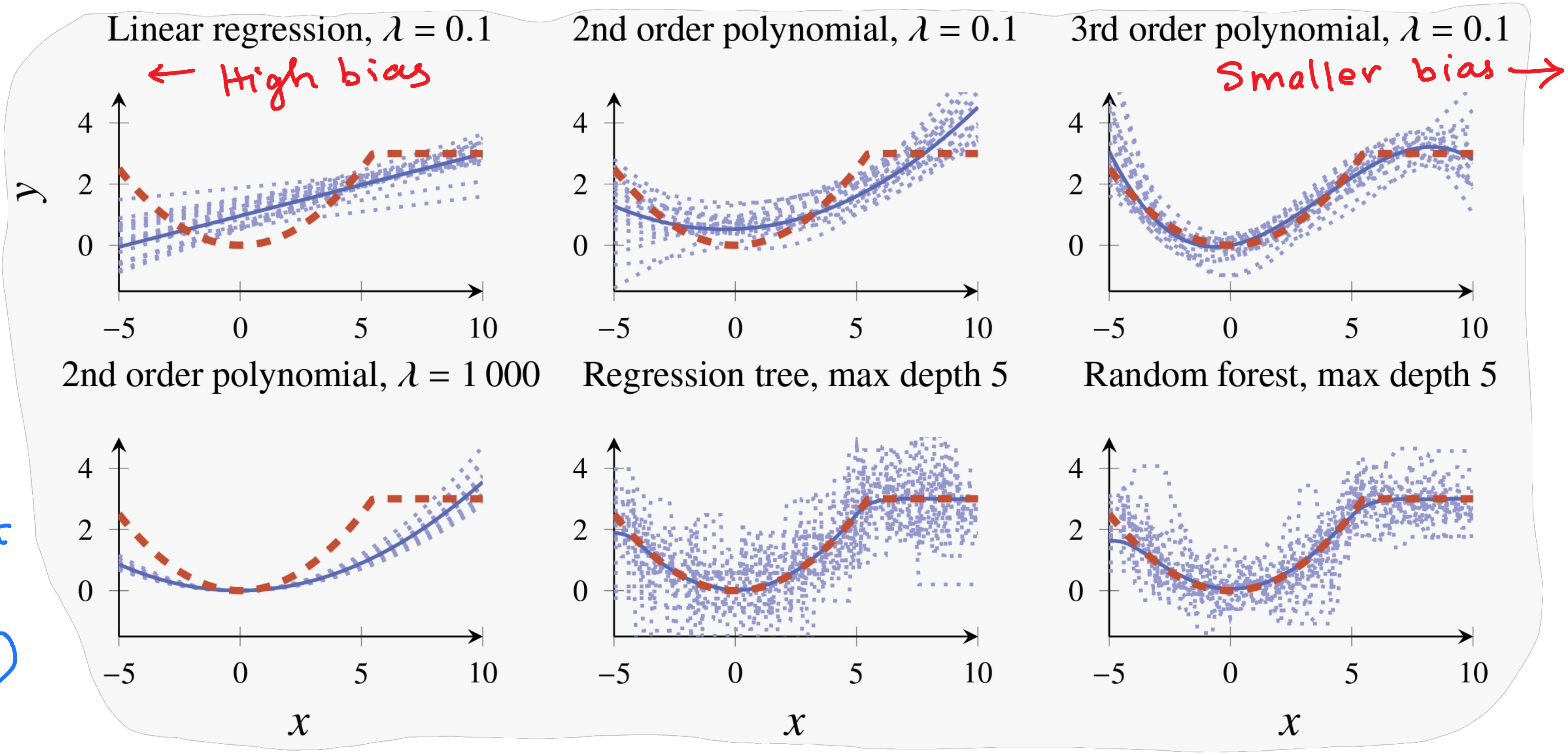
Now fit the input-output data $\{x^{(i)}, y^{(i)}\}_{i=1}^{10}$ using

(a) Linear regression with L_2-regularization
(b) Linear regression with a quadratic polynomial and L_2-regularization
(c) Linear regression with a cubic polynomial and L_2-regularization
(d) Regression Tree, (e) A random forest with 10 regression trees

--- True model $f_0(x)$

— Mean model $\bar{f}(x)$

--- Different model $\hat{y}(x^*; \mathcal{T})$ learned from different training datasets



Tools for evaluating binary classifiers

Confusion Matrix

- Create a training set and hold-out validation set
- Train a binary classifier (say logistic regression)
- Separate the validation data into 4 groups depending upon actual output y and model prediction $\hat{y}(\underline{x})$
- Create confusion matrix (gives overview of a classifier)

	$y = -1$	$y = 1$	Total
$\hat{y}(\underline{x}) = -1$	TN	FN	nt^* (pred)
$\hat{y}(\underline{x}) = 1$	FP	TP	pt^* (pred)
Total	nt (true)	pt (true)	N

$nt, pt \leftarrow$ negative/positive total

TN $\leftarrow$ True negative

TP $\leftarrow$ True positive

FP $\leftarrow$ False positive

FN $\leftarrow$ False negative

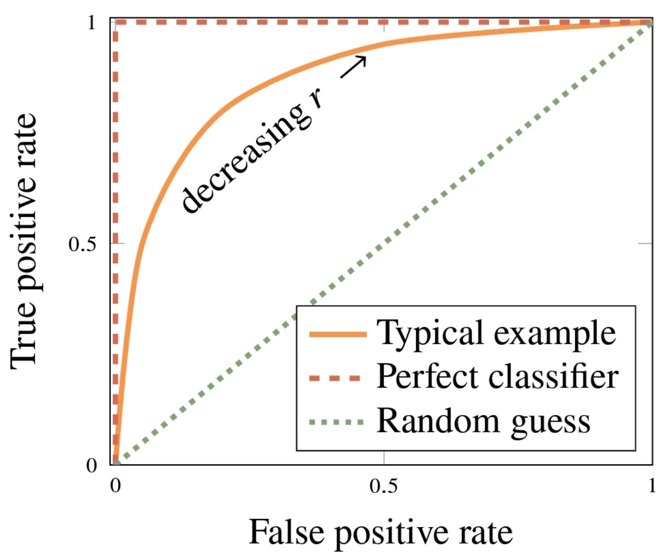
$$\text{Misclassification rate} = \frac{(FP + FN)}{N}$$

ROC (Reciever Operating Characteristics)

- Many classifiers use a threshold for classification (e.g. logistic regression)

- If we want to compare different classifiers for a certain problem without specifying the decision threshold 'r', the ROC curve is useful

- For different values of $r \in [0, 1]$
  - plot $\left(\frac{TP}{pt}\right)$ vs $\left(\frac{FP}{nt}\right)$



- A perfect classifier always predicts the correct class for all $r \in (0, 1)$

- Hence ROC curve for perfect classifier touches upper left corner

- A poor classifier giving out random guesses will give a straight diagonal line