Classifier evaluation

Binary classifier performance

Sample set: $(\mathbf{x}^{(i)}, y^{(i)})$, $i = 1, 2, \ldots, N$

Let $h(\mathbf{x})$ be a soft classifier and $\hat{y}^{(i)}$ be the predicted output

(e.g., $\hat{y}^{(i)} = \begin{cases} 1, & \text{if } h(\mathbf{x}^{(i)}) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases}$)

Several performance metrics for binary classifiers exist

They are often used to compare different classifier models

Erros em problemas de classificação binária

Classes

- Positivo
- Negativo

Quatro possíveis diagnósticos:

- Verdadeiro-positivo (TP)

  $y = 1$ and $\hat{y} = 1$

- Falso-positivo (FP)

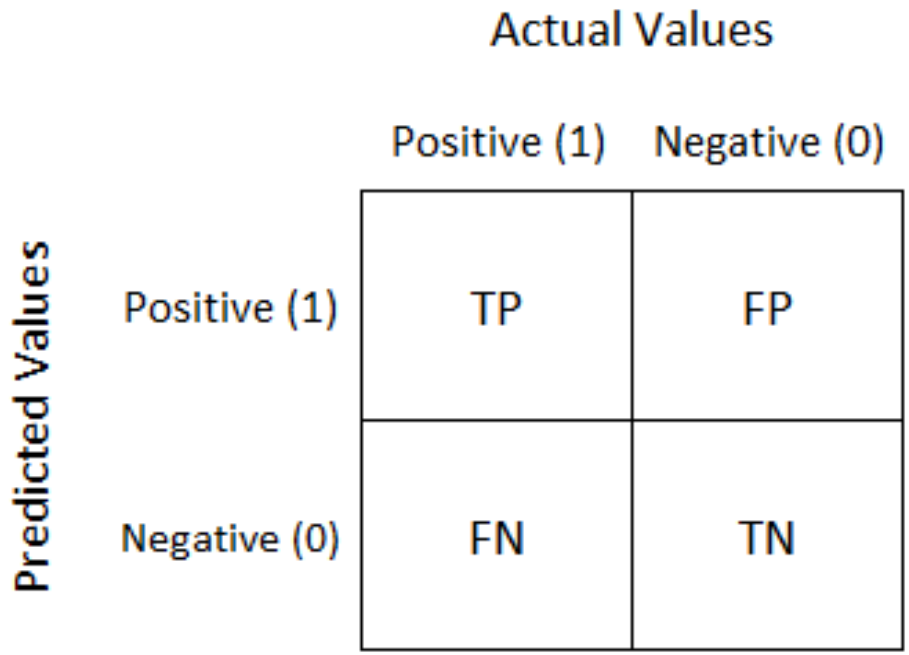
  $y = 0$ and $\hat{y} = 1$

- Falso-negativo (FN)

  $y = 1$ and $\hat{y} = 0$

- Verdadeiro-negativo (TN)

  $y = 0$ and $\hat{y} = 0$



Diversas métricas a partir de TP, FP, TN, FN

Matriz de confusão

	True condition				
Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
	True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
	False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$		

Fonte: Wikipedia

Diversas métricas a partir de TP, FP, TN, FN

	True condition				
Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
	True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
	False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$		

Fonte: Wikipedia

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

Diversas métricas a partir de TP, FP, TN, FN

	True condition			
Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
	True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
	False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Fonte: Wikipedia

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

Diversas métricas a partir de TP, FP, TN, FN

	True condition				
Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
	True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$		

Fonte: Wikipedia

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

Diversas métricas a partir de TP, FP, TN, FN

	True condition			
Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
	True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
	False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Fonte: Wikipedia

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

Diversas métricas a partir de TP, FP, TN, FN

	True condition			
Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
	True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
	False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	

F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Fonte: Wikipedia

$$\text{F1-score} = 2\frac{Precision * Recall}{Precision + Recall}$$

Métricas de desempenho no caso de múltiplas classes

Let TP_j, FP_j, TN_j, FN_j, for each j (class j against the rest)

Micro-averaging

- Compute $TP = \sum TP_j$, $FP = \sum FP_j$, $TN = \sum TN_j$, $FN = \sum FN_j$
- Compute the performance metrics from TP, FP, TN, FN

Macro-averaging

- Compute the performance metrics for each class, from TP_j, FP_j, TN_j, FN_j
- Compute the mean of each metric

Métricas de desempenho no caso de múltiplas classes

Let TP_j, FP_j, TN_j, FN_j, for each j (class j against the rest)

Micro-averaging

- Compute $TP = \sum TP_j$, $FP = \sum FP_j$, $TN = \sum TN_j$, $FN = \sum FN_j$
- Compute the performance metrics from TP, FP, TN, FN
- assigns same importance to all examples $\rightsquigarrow$ larger classes dominate

Macro-averaging

- Compute the performance metrics for each class, from TP_j, FP_j, TN_j, FN_j
- Compute the mean of each metric
- assigns same importance to all classes

(There is no consensus about which is the right one)

TP, FP, TN, FN depends on the threshold T

$$\hat{y}^{(i)} = \begin{cases} 1, & \text{if } h(\mathbf{x}^{(i)}) \geq T = 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

One can choose other values than 0.5 for the threshold T

Often we would like to maximize the true positives (TP) at the same time we would like to minimize the false positives (FP)

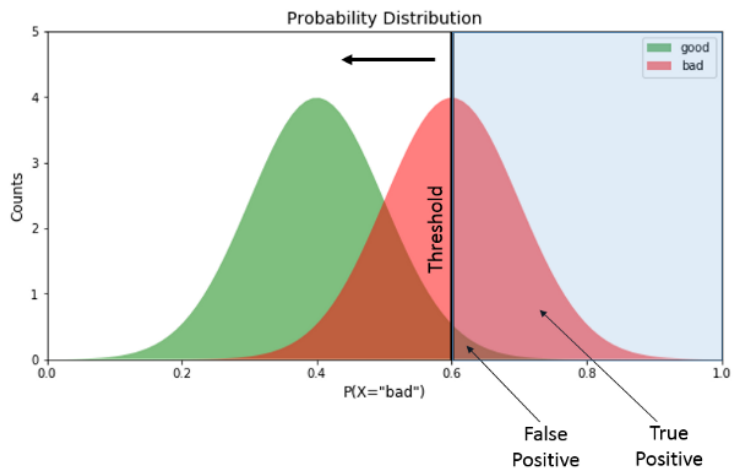
ROC curve and PR curves (shown next) are often used as tools to compare different classification approaches

ROC Curve

ROC : Receiver operating characteristic

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

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



$$TPR = \frac{True\ Positive}{Total\ Positive}$$

$$FPR = \frac{False\ Positive}{Total\ Negative}$$

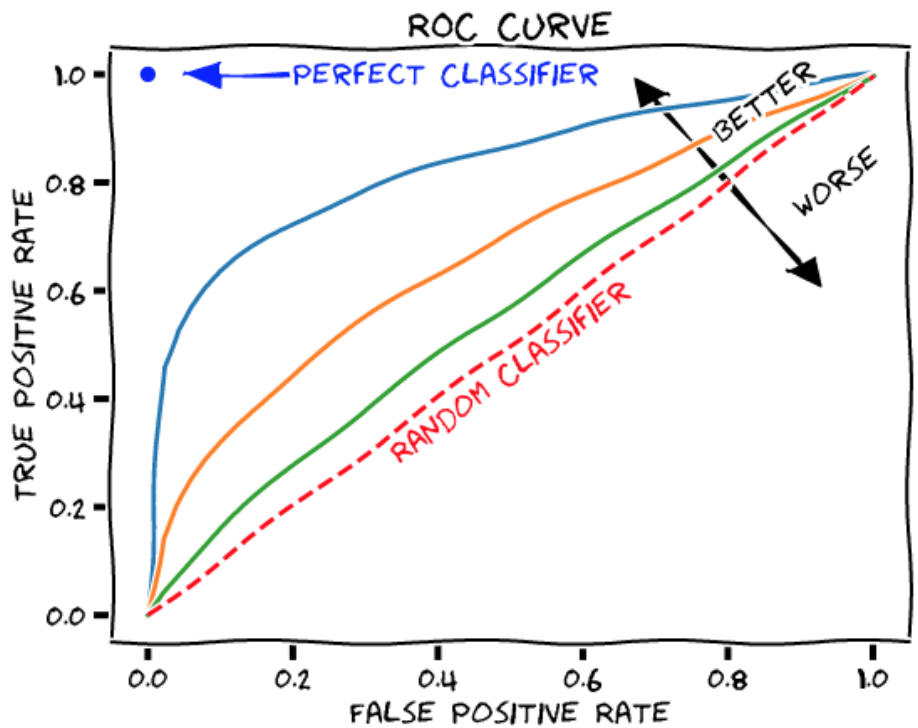
www.kdnuggets.com/2018/07/receiver-operating-characteristic-curves-demystified-python.html

$T = 1.0 \implies$ all input are classified as negative (TP=0% and FP=0%)

$T = 0.0 \implies$ all input are classified as positive (TP=100% and FP=100%)

As we vary T from 1.0 to 0.0

- **Perfectly separated classes**: TP will reach 100% while FP stays at 0%, and only after that FP will start to increase
- **General case**: TP will start to increase but so does FP too.



analyticsindiamag.com/beginners-guide-to-understanding-roc-curve-how-to-find-the-perfect-probability-threshold/

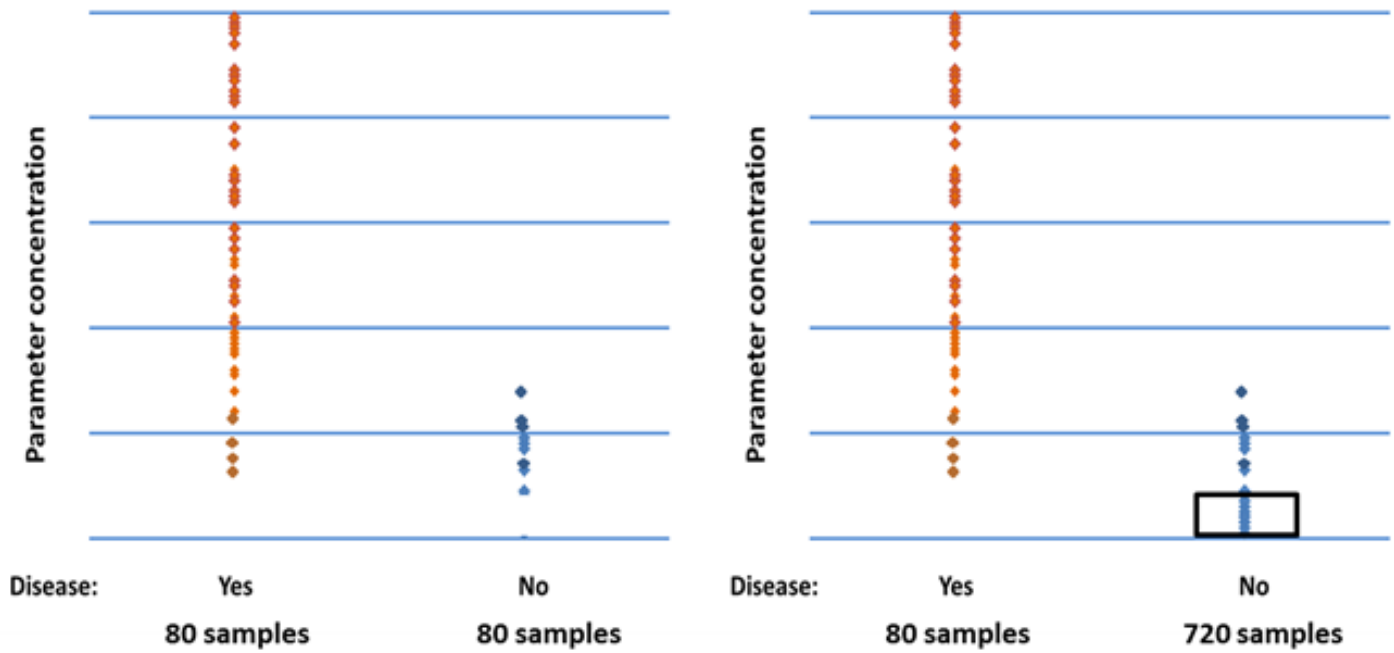
PR Curve

PR: *precision-recall*

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

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

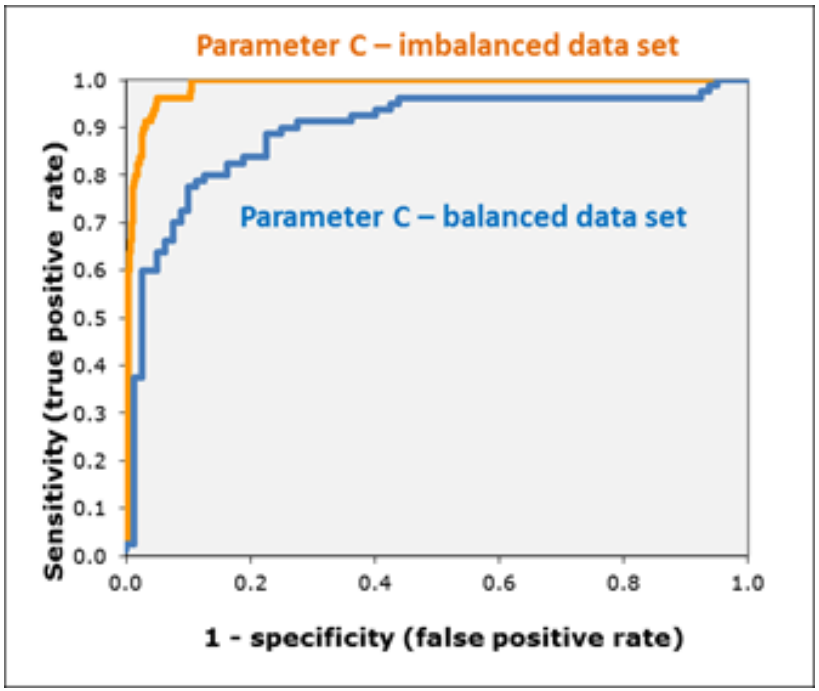
PR curves são mais apropriados quando as classes são altamente desbalanceadas (no exemplo abaixo à direita, muito mais negativos)

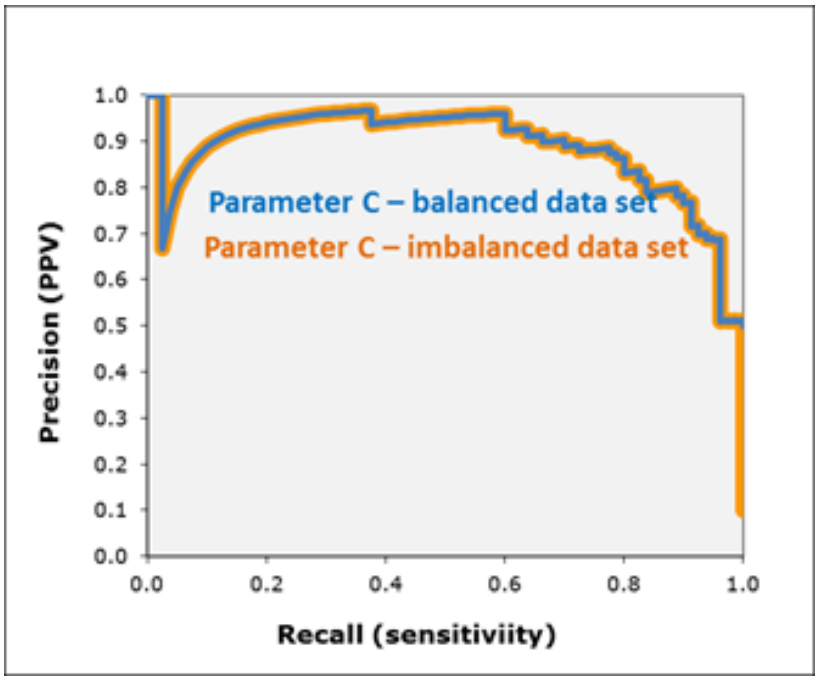


https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used

ROC x Precision-recall curve

ROC pode fornecer uma percepção incorreta quando classes estão desbalanceadas.





https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used

So far, we have discussed some performance measures

Now we will discuss how to compute this measures and how statistically significant they are

Performance measure computation

Regarding data, all we have is a dataset D

Some measures we have seen:

- E_{in} *in-sample error*
- E_{out} *out-of-sample error* (unknown, no way to compute it)
- TP, FP, TN and FN : from these, several performance metrics such as accuracy, recall, precision, F1-score, AUC, ... are computed

We have seen that E_{in} is computed over the training set

E_{in} is a (super)optimistic estimate of E_{out}

$$E_{out} = E_{in} + \text{generalization\_error}$$

Minimizing only E_{in} will lead to overfitting

How to find a more realistic estimate of E_{out} ?

Holdout method

Partition the existing dataset into two subsets:

$$D = D_{train} \cup D_{val}$$

D_{train} D_{val}

D_{train} is used for training and for computing E_{in}

D_{val} is used to compute E_{val}, an unbiased estimate of E_{out}

Drawbacks of the holdout method

Let the size of the sets be:

- $|D_{train}|$
- $|D_{val}|$
- $|D| = |D_{train}| + |D_{val}|$

Potential problems:

- large $|D_{val}| \rightsquigarrow$ small $|D_{train}|$ (small amount of training data)
- small $|D_{val}| \rightsquigarrow E_{val}$ hardly will be a good estimate of E_{out}
- when D_{train} and/or D_{val} have some bias

Cross-validation

General idea:

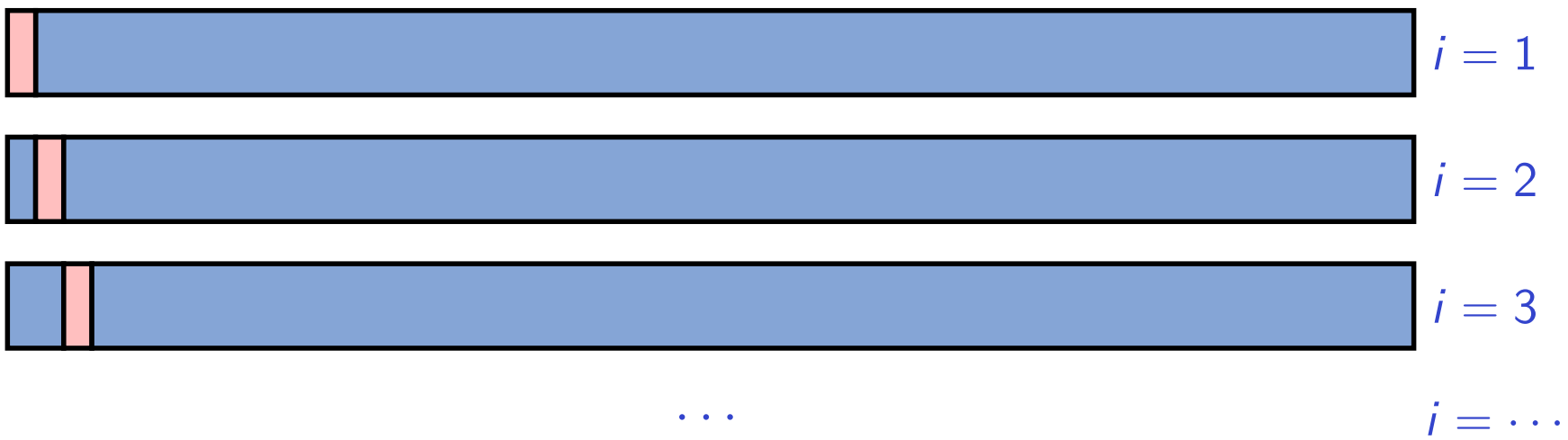
- repeat the holdout method several times, using different D_{train} and D_{val} sets sampled from $D \rightsquigarrow$ multiple values for E_{val}

- Compute the mean validation error $\overline{E}_{val}$, which should be a more robust estimator of E_{out}

Leave-one-out cross-validation

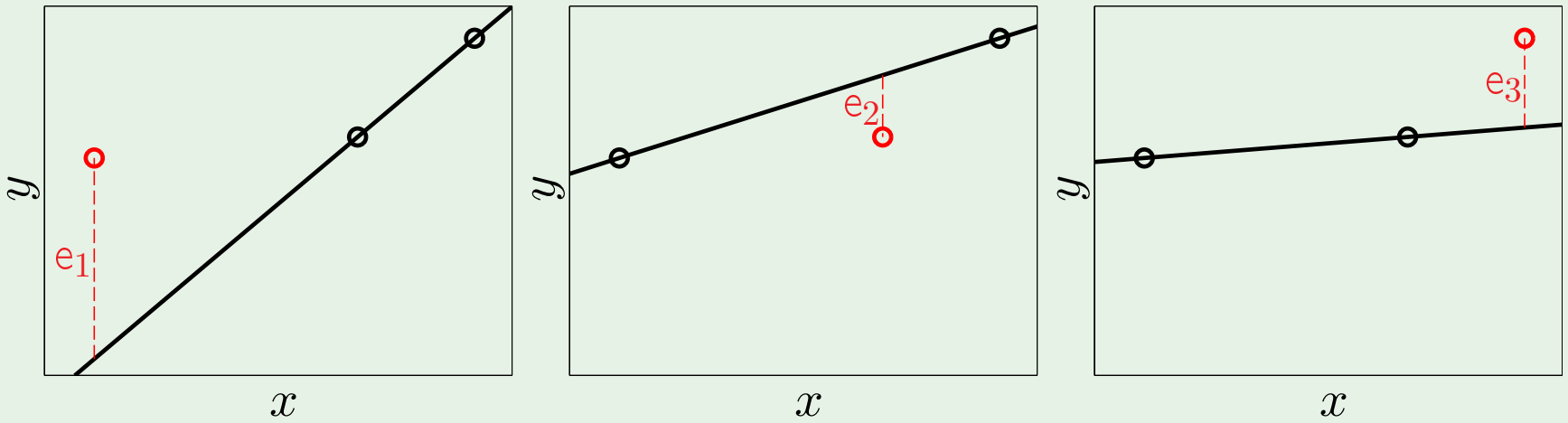
Training is repeated $|D|$ times

At training round i, $D_{train}^{(i)} = D \setminus \{\mathbf{x}^{(i)}\}$ and $D_{val}^{(i)} = \{\mathbf{x}^{(i)}\}$



Cross-validation error: $E_{cv} = \frac{1}{|D|} \sum_{i=1}^{|D|} E_{val}^{(i)}$

Illustration of cross validation

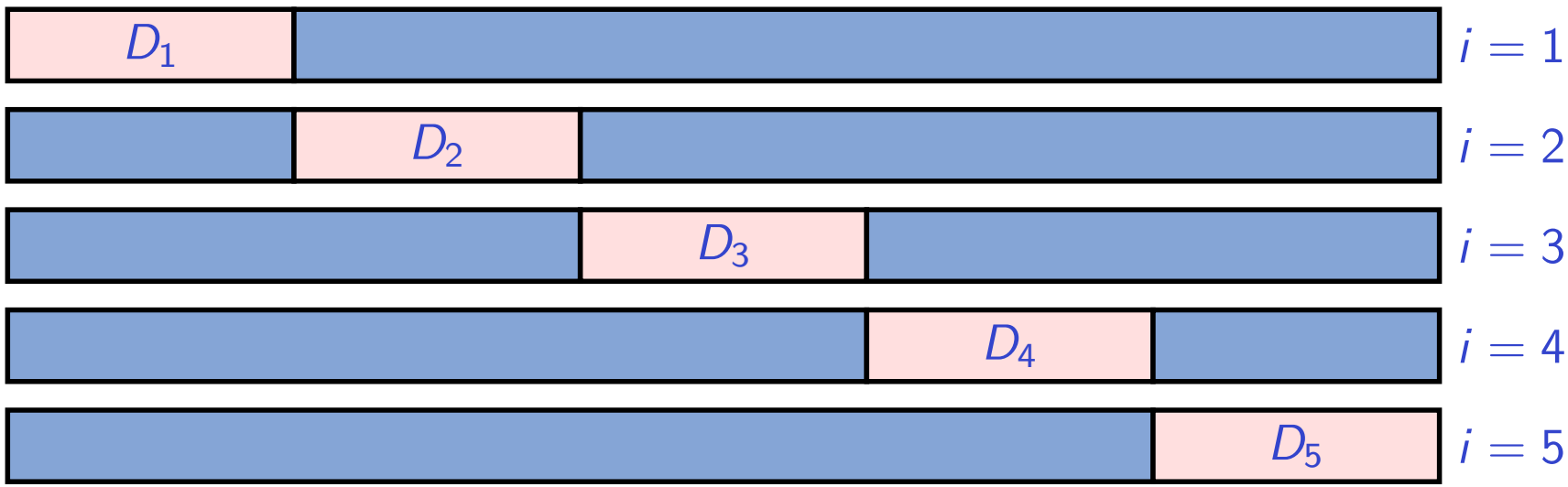


$$E_{\text{cv}} = \frac{1}{3}\left(\mathrm{e}_1 + \mathrm{e}_2 + \mathrm{e}_3 \right)$$

k-fold cross validation

Divide D into k parts $D_1, D_2, \ldots, D_k$ of approximately equal sizes
Repeat the training k times
At training round i, $D_{train}^{(i)} = D \setminus D_i$ and $D_{val}^{(i)} = D_i$

Example with $k = 5$ (five folds):



Cross-validation error: $E_{cv} = \frac{1}{k} \sum_{i=1}^{k} E_{val}^{(i)}$

- for the holdout method a common proportion is 70%$\sim$80% for training and 20%$\sim$30% for validation

- for k-fold cross-validation, usual value of k is 5 or 10

- leave-one-out is just k-fold cross-validation, with $k = |D|$

  Requires $|D|$ training rounds $\rightsquigarrow$ computationally intense

  For small $|D|$ it could be the best option

- holdout should be sufficient if both D_{train} and D_{val} are large and representative enough of the true distribution

  (this usually is not the case in practice)

- k-fold cross-validation is largely used

Since we are considering estimators of E_{out} (such as E_{val} or E_{cv}), one of the interests is on their statistical properties

- **bias**: how much the expected value of the estimate differs from the true value
- **variance**: how spread are the estimates

Short text that helps to quickly review these concepts:

https://www.cs.utah.edu/~jeffp/teaching/cs3130/lectures/L13-Estimation.pdf

What can we say about the statistical properties of these estimators ?

Analyzing the estimate

On out-of-sample point $(\mathbf{x}, y)$, the error is $\mathbf{e}(h(\mathbf{x}), y)$

Squared error: $\big(h(\mathbf{x}) - y\big)^2$

Binary error: $[\![h(\mathbf{x}) \neq y]\!]$

$$\mathbb{E}\Big[\mathbf{e}(h(\mathbf{x}), y)\Big] = E_{\text{out}}(h)$$

$$\text{var}\Big[\mathbf{e}(h(\mathbf{x}), y)\Big] = \sigma^2$$

From a point to a set

On a validation set $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_K, y_K)$, the error is $E_{\text{val}}(h) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{e}(h(\mathbf{x}_k), y_k)$

$$\mathbb{E}\left[E_{\text{val}}(h)\right] = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[\mathbf{e}(h(\mathbf{x}_k), y_k)\right] = E_{\text{out}}(h)$$

$$\text{var}\left[E_{\text{val}}(h)\right] = \frac{1}{K^2} \sum_{k=1}^{K} \text{var}\left[\mathbf{e}(h(\mathbf{x}_k), y_k)\right] = \frac{\sigma^2}{K}$$

$$E_{\text{val}}(h) = E_{\text{out}}(h) \pm O\left(\frac{1}{\sqrt{K}}\right)$$

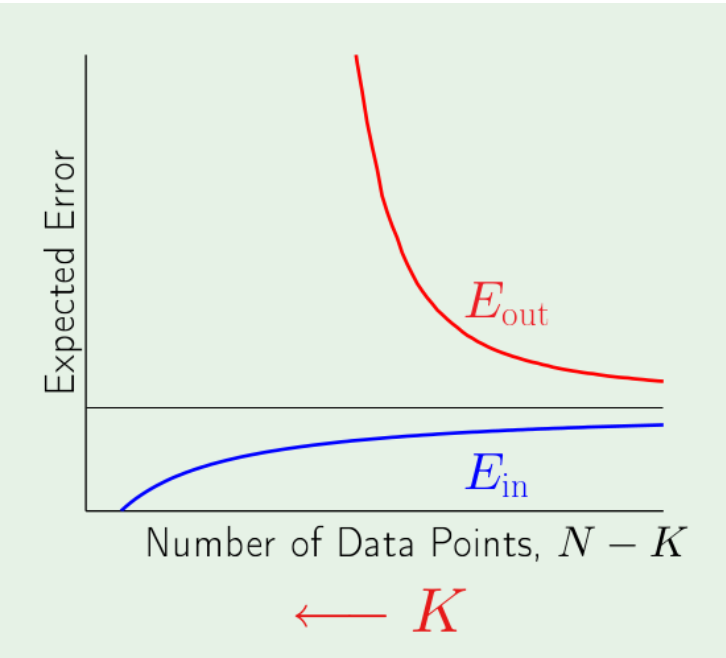
Chapter 5 of the book "Machine Learning", by Tom Mitchell shows how to compute a confidence interval for E_{out}

That is, an interval $E_{val} \pm \Delta$ that contains E_{out} with high probability ($\Delta = O(\frac{1}{\sqrt{K}})$)

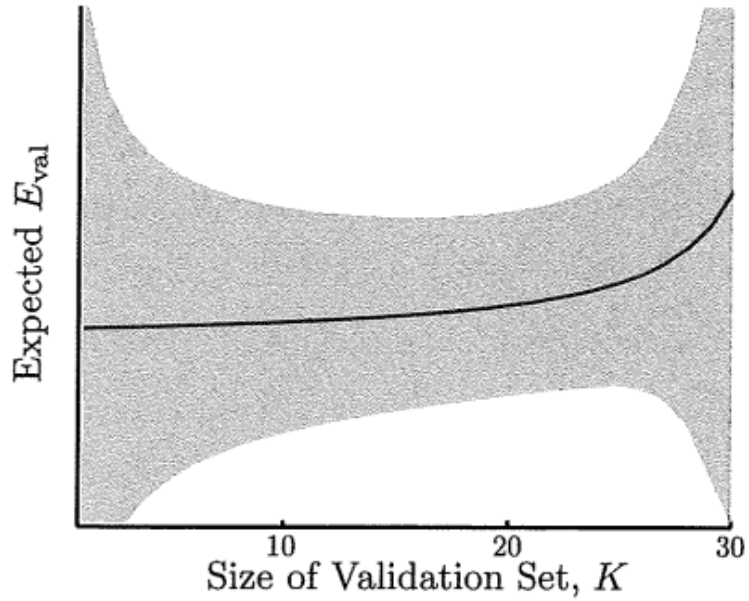
Large $K = |D_{val}|$ yields a good estimate of E_{out} (small variance) but at the same time, with less training data, E_{out} tend to be larger than when using the whole dataset

$K = |D_{val}|$

E_{in}, E_{out} $\times$ training samples



Mean and variance of E_{val}



$$E_{out} \leq E_{val} + O(\frac{1}{\sqrt{K}})$$

Is E_{cv} a good estimator of E_{out} ?

k models $\rightsquigarrow$ k values for E_{out} $\rightsquigarrow$ average $\overline{E}_{out}$

It can be demonstrated that E_{cv} is an unbiased estimator of $\overline{E}_{out}$

The variance of E_{cv} can not be easily computed

Empirically, it has been observed that E_{cv} is a good estimator of $\overline{E}_{out}$

Further reading:

- Dietterich, Thomas G., Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput., 10(7), p.1895-1923, 1998
- Chapter 7 of "The Elements of Statistical learning", by Hastie *et al.*

Can we do better ?

There are a lot of discussions in literature about how to get unbiased estimates of E_{out} with small variance, etc

Further reading:

- Dietterich, Thomas G., Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput., 10(7), p.1895-1923, 1998
- Chapter 7 of "The Elements of Statistical learning", by Hastie *et al.*

- we have seen some performance metrics
- we have seen some techniques for estimating these metrics
- we have seen some algorithms, and we should be able to obtain multiple models by training these algorithms under distinct training setups

What's next ? **How do we choose a model ?**

Model selection

Here we call as **model** any specific hypothesis h in the hypothesis space $\mathcal{H}$ that resulted after training

For example, after doing logistic regression we have a weight vector $\mathbf{w}$ which characterizes the learned classifier (the model)

As we have seen, we can compute $E_{val}(h)$ over a validation set

Model selection

Suppose you have two models, h_1 and h_2, as well as $E_{val}(h_1)$ and $E_{val}(h_2)$

If $E_{val}(h_1) < E_{val}(h_2)$, would you choose h_1 without hesitation ?

What if $E_{val}(h_1) = E_{val}(h_2)$?

Model selection in practice

Based on validation or cross-validation error

Usually the one with smallest validation error is chosen

Statistical tests can be applied to test whether $E_{val}(h_1) = E_{val}(h_2)$ or not

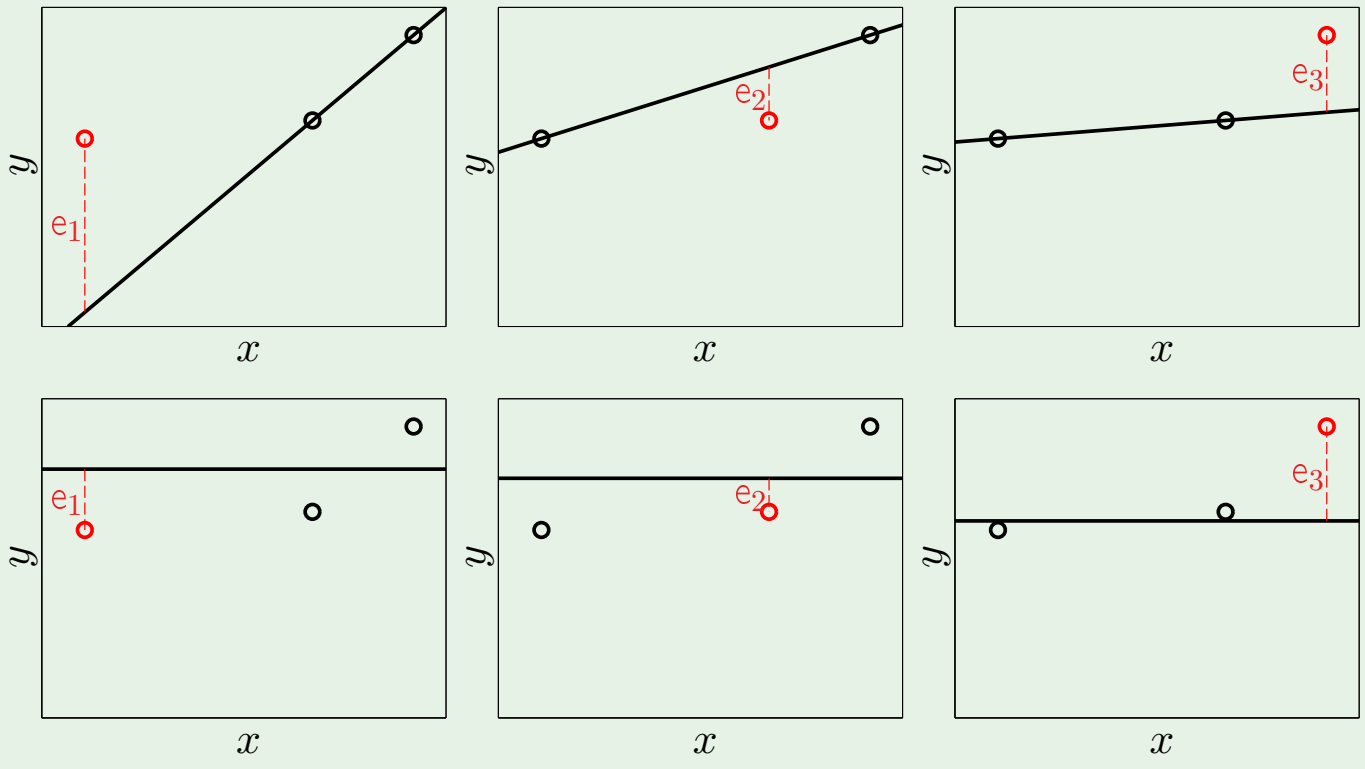
Holdout error: Hypothesis test (see for instance Chapter 5 of the book "Machine Learning", by Tom Mitchell)

Cross-validation error: paired t-test (see Dietterich, Thomas G., Approximate statistical tests for comparing supervised classification learning algorithms)

Model selection using CV

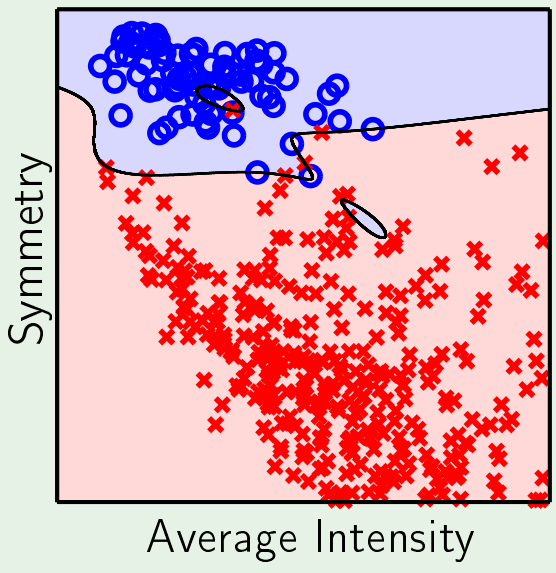
Linear:

Constant:



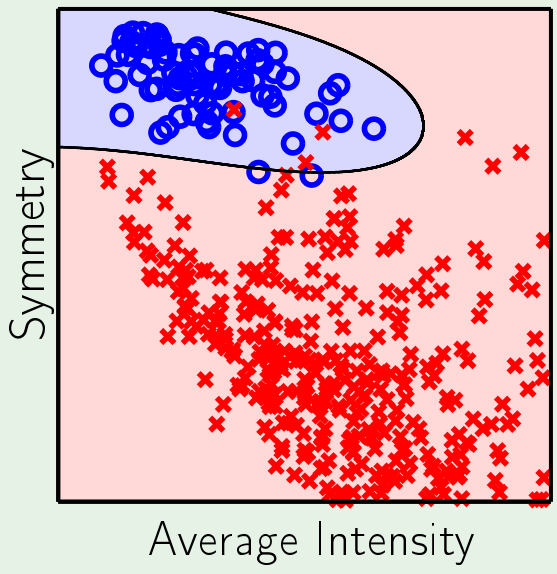
The result

without validation



$E_{\text{in}} = 0\%$ $E_{\text{out}} = 2.5\%$

with validation



$E_{\text{in}} = 0.8\%$ $E_{\text{out}} = 1.5\%$

The process of model selection and performance evaluation

1. Divide the dataset D into $D_{train+val}$ and D_{test}

2. Isolate D_{test} (put it under quarantine ...)

3. Use $D_{train+val}$ for training and choosing a model
   Depending on the selection technique different partitions of $D_{train+val}$ will be used for training and for error estimation

4. the chosen model can be retrained using the whole dataset $D_{train+val}$
   (advantage is that we have more training data)

5. Having the final model, compute E_{test} over D_{test}
   E_{test} would be a less biased estimator of E_{out} than E_{val} and E_{cv} (since these last two would be an optimistic estimate)

In many situations, we just want to choose the best model

We do not need to have an estimate of E_{out}

In such situation, it is common to not consider D_{test}
(the whole set D is used for training and model choice only)

Obviously, the validation error of the chosen model is biased
(because we chose the model with minimum E_{val} value)

The same observation holds with respect to any of the metrics computed on D_{val}, after a model is chosen based on its E_{val} value

Summary of what we have discussed

- Performance metrics: E_{in}, E_{out}, accuracy, precision, recall, etc
- Estimation of E_{out}: E_{val}, E_{cv}
- E_{val}, E_{cv} are used for model selection
  Thus they are no longer an unbiased estimator of E_{out}
- validation set: it is used in the learning/model selection process
- test set: it is totally independent of the learning/model selection process; used to obtain an unbiased estimate of E_{out}

References

Section 4.3 of "Learning from data" by Mostafa *et al.*

There are references at:

https://stats.stackexchange.com/questions/18348/differences-between-cross-validation-and-bootstrapping-to-estimate-the-predictio

Chapter 7 of "The Elements of Statistical learning", by Hastie *et al.*

Chapter 5 of "Machine Learning" , by Tom Mitchell

Steven L. Salzberg, On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach, Data Min. Knowl. Discov., 3, pp.317-328, 1993

Dietterich, Thomas G., Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput., 10(7), p.1895-1923, 1998.