Machine Learning
Evaluation of Machine Learning

Gloria Inés Alvarez V.

Pontifica Universidad Javeriana Cali

Periodo 2018-2

Today Class

Subject:

- Learning evaluation
- Ensemble methods

Learning Goals:
At the end of the class you should be able to:

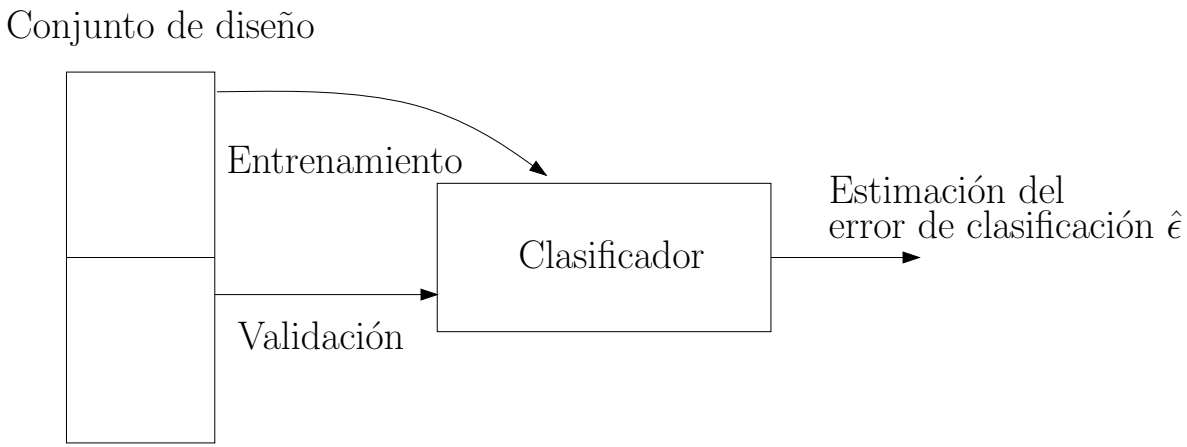
- Evaluate que quality of learning in a machine learning system
- Identify some methods to make several models work together in solving a problem

Learning Evaluation

The starting point for the evaluation:

Example	Target	Prediction
1	1	0
2	0	1
3	1	1
4	1	0
5	1	1
6	0	0
7	0	0
8	0	1
9	1	1
10	0	0
11	0	0
12	1	0
13	1	1
14	0	1
15	1	0
16	0	1
17	1	1
18	1	0
19	1	0
20	0	1

Learning Evaluation



Otro conjunto de prueba $\rightarrow$ Otro estimado del error

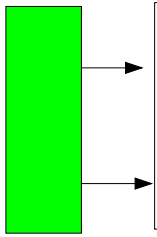
Otro conjunto de entrenamiento $\rightarrow$ Otro clasificador

$$\sigma^2_{\hat{\epsilon}} = \text{Var}(\hat{\epsilon} \mid \text{tamano conjunto de validación}) = \frac{\hat{\epsilon}(1-\hat{\epsilon})}{N}$$

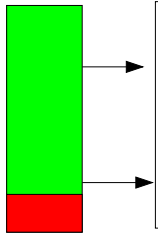
Tasa de Error de la Tarea

- Tasa de error verdadero
  1. Es la probabilidad de clasificar erroneamente un patrón seleccionado de manera aleatoria
  2. Es la tasa de error de un conjunto de tamaño infinito extraído de la misma distribución que el conjunto de entrenamiento
- Tasa de error esperado: Es el valor esperado de la tasa de error verdadero sobre conjuntos de entrenamiento de una tamaño dado
- Tasa de error de Bayes: Es el mínimo teórico de la tasa de error verdadera. Es el valor de la tasa de error verdadera si el clasificador produjo las probabilidades a posteriori verdaderas $p(\omega_i \mid x), i = 1 \ldots C$

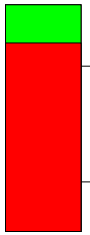
Learning Evaluation



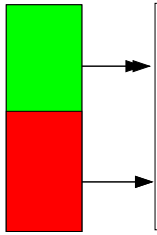
El mismo conjunto de entrenamiento y validación puede entregar un un buen clasificador pero producirá un clasificador sesgado



Un conjunto pequeño e independiente de validación produce un estimador no sesgado del error pero no confiable para un clasificador bien entrenado



Un conjunto grande e independiente de validación produce un estimado confiable (varianza pequeña) del error para un clasificador mal entrenado



La proporción 50-50 se usa con frecuencia pero se pierden datos para entrenar por eso se usa validación cruzada

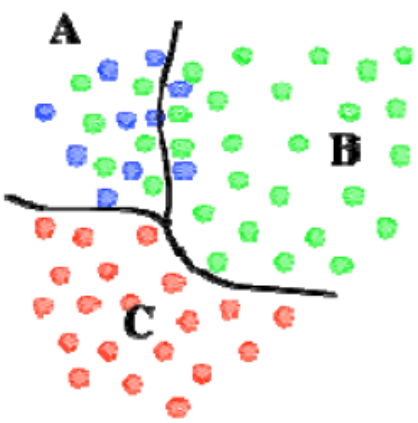
Confusion Matrix

objects from	classified to A	B	C
class A	8	2	0
class B	6	23	1
class C	4	1	15

-0.20 error in class A
-0.23 error in class B
-0.25 error in class C

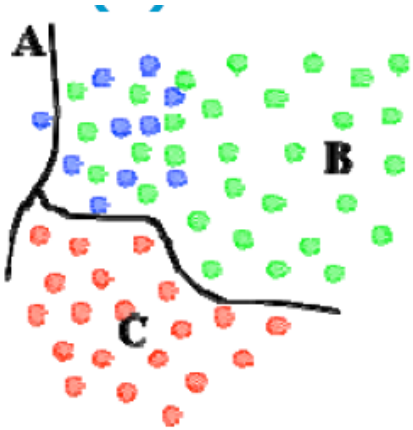
0.228 averaged error

Confusion Matrix



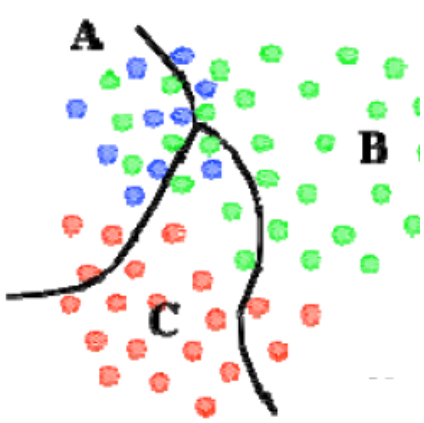
objects from \ classified to

	A	B	C
class A	8	2	0
class B	6	24	0
class C	0	0	20



classified to

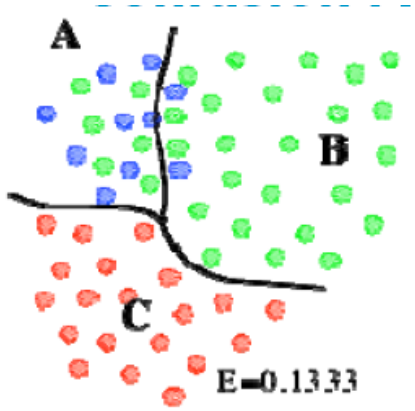
	A	B	C
class A	1	9	0
class B	0	30	0
class C	0	0	20

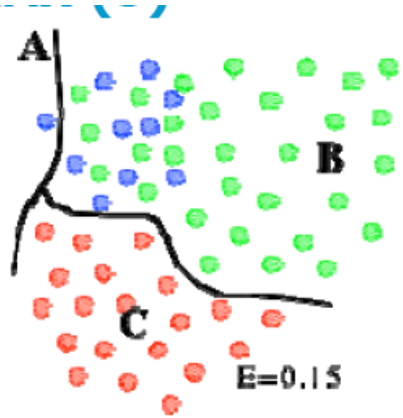


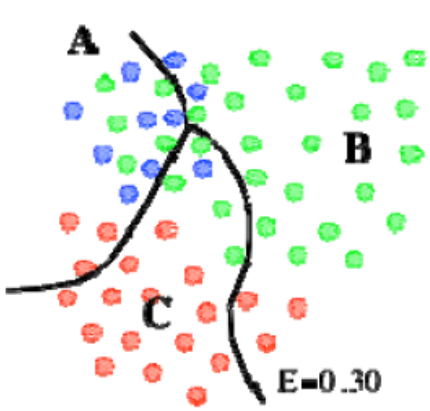
classified to

	A	B	C
class A	7	2	1
class B	5	21	4
class C	3	2	15

Confusion Matrix





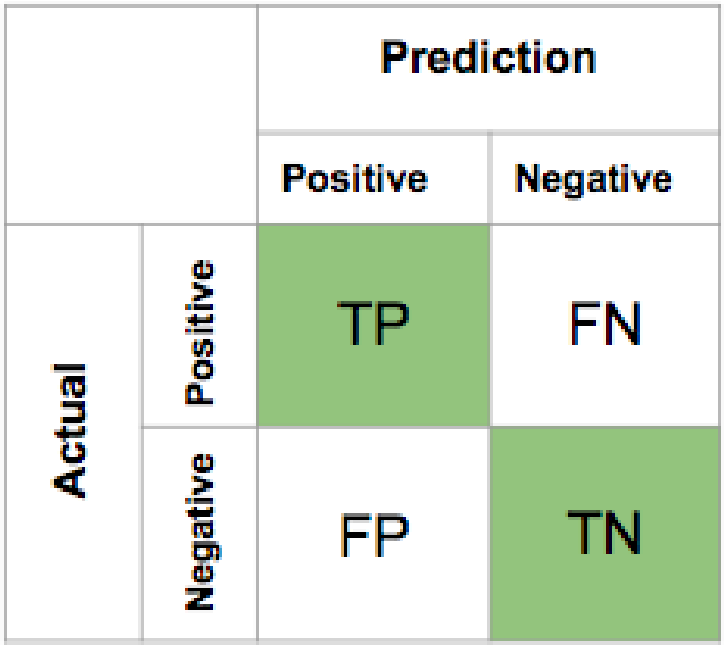


objects from	classified to A	B	C	
class A	8	2	0	0.20
class B	6	24	0	0.20
class C	0	0	20	0.00
				0.133

	classified to A	B	C	
class A	1	9	0	0.9
class B	0	30	0	0.0
class C	0	0	20	0.0
				0.30

	classified to A	B	C	
class A	7	2	1	0.30
class B	5	21	4	0.30
class C	3	2	15	0.30
				0.30

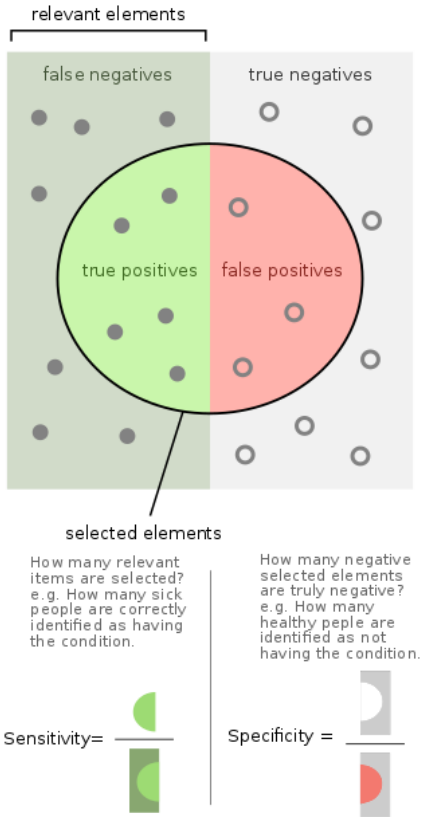
Performance Measures for Binary Tasks[1]



- Accuracy: $\frac{tp+tn}{tp+tn+fp+fn}$
- Sensitivity: $\frac{tp}{tp+fn}$
- Specificity: $\frac{tn}{tn+fp}$

[1] Image from: https://docs.wso2.com/display/ML110/Model+Evaluation+Measures

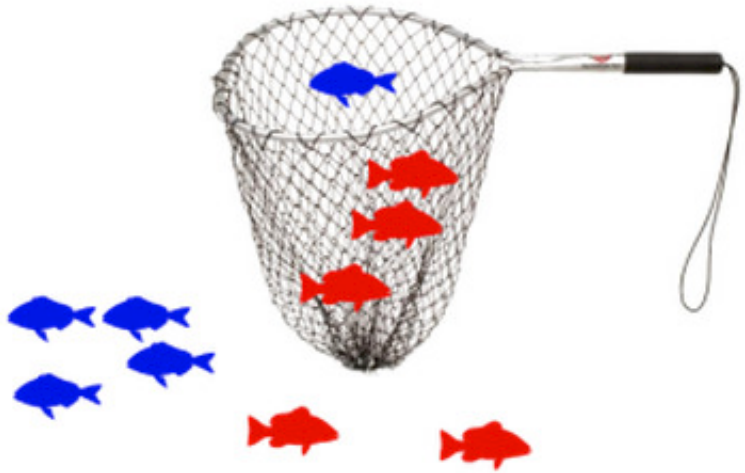
Performance Measures for Binary Tasks[2]



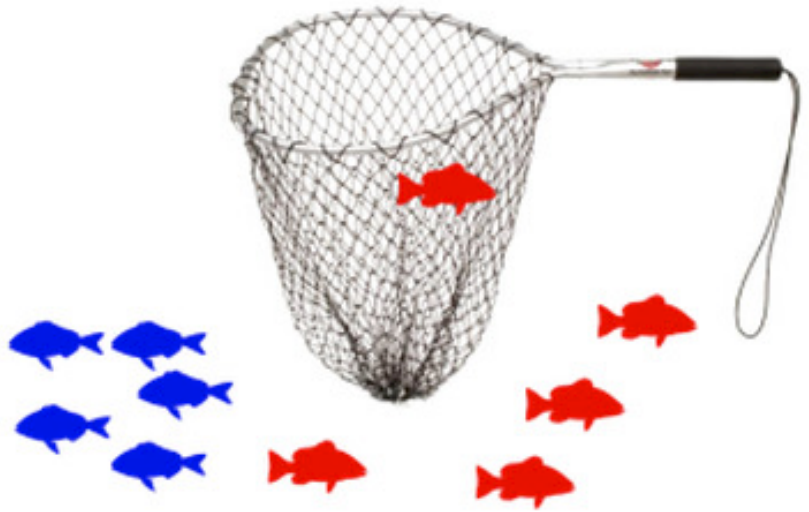
- Precision: $\frac{tp}{tp+fp}$ ability to return only relevant instances
- Recall: $\frac{tp}{tp+fn}$ ability to identify all relevant instances
- F1 score: $2 * \frac{Precision*Recall}{Precision+Recall}$

[2]Image from: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Precision and Recall



Precision and Recall



Precision and Recall

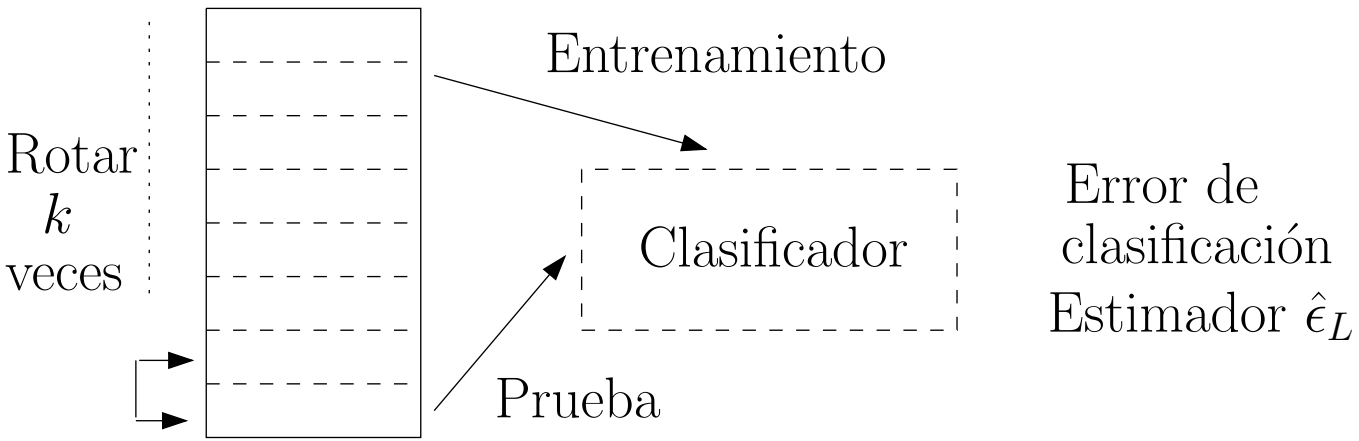
Holdout Evaluation Method

1. We divide the training set into two parts: one for training and another for test
2. The biggest disadvantage is that it reduces the size of the training set and the test set
3. Another problem is to decide how many of the N data are for training and how many are for test

Leave-one-out Evaluation Method

1. Solves the lack of independence between the training set and the test
2. Solves the dilemma associated with the holdout method: reduced sizes of training set and validation set
3. The training is carried out with $N - 1$ samples and the validation is carried out using the excluded sample.
4. If the sample is misclassified, an error is counted
5. This procedure is repeated N times, excluding a *different* sample each time
6. The total number of errors leads to the estimation of the classification error

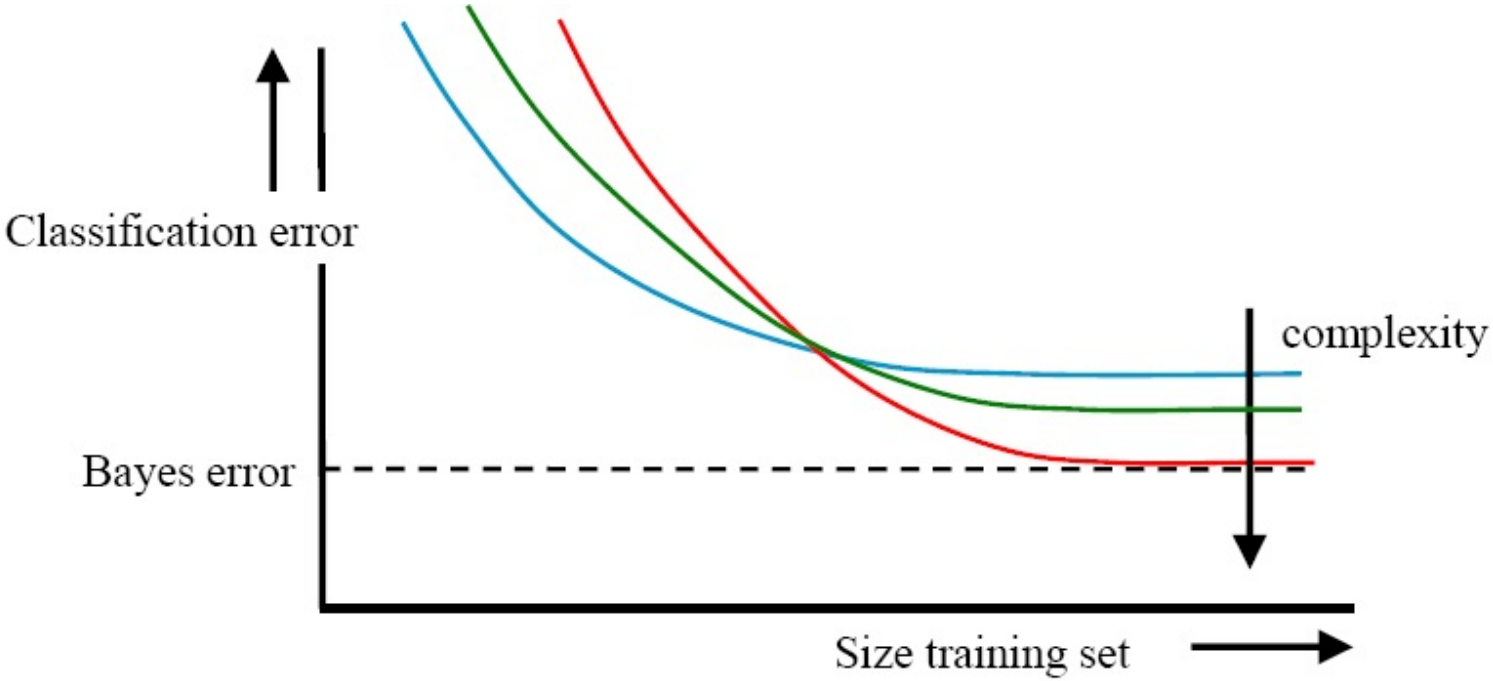
Cross Validation Evaluation Method



Cross Validation Evaluation Method

- To what fraction of the design set correspond the sizes of the test set and the training set if $k = n$?
- R/ $1/n$ y $\frac{n-1}{n}$
- Training and testing n times and errors are averaged (value usually chosen $n = 10$)
- Final classifier: is trained with all the samples *rightarrow* best possible classifier
- The estimate of the error is slightly biased in a pessimistic way

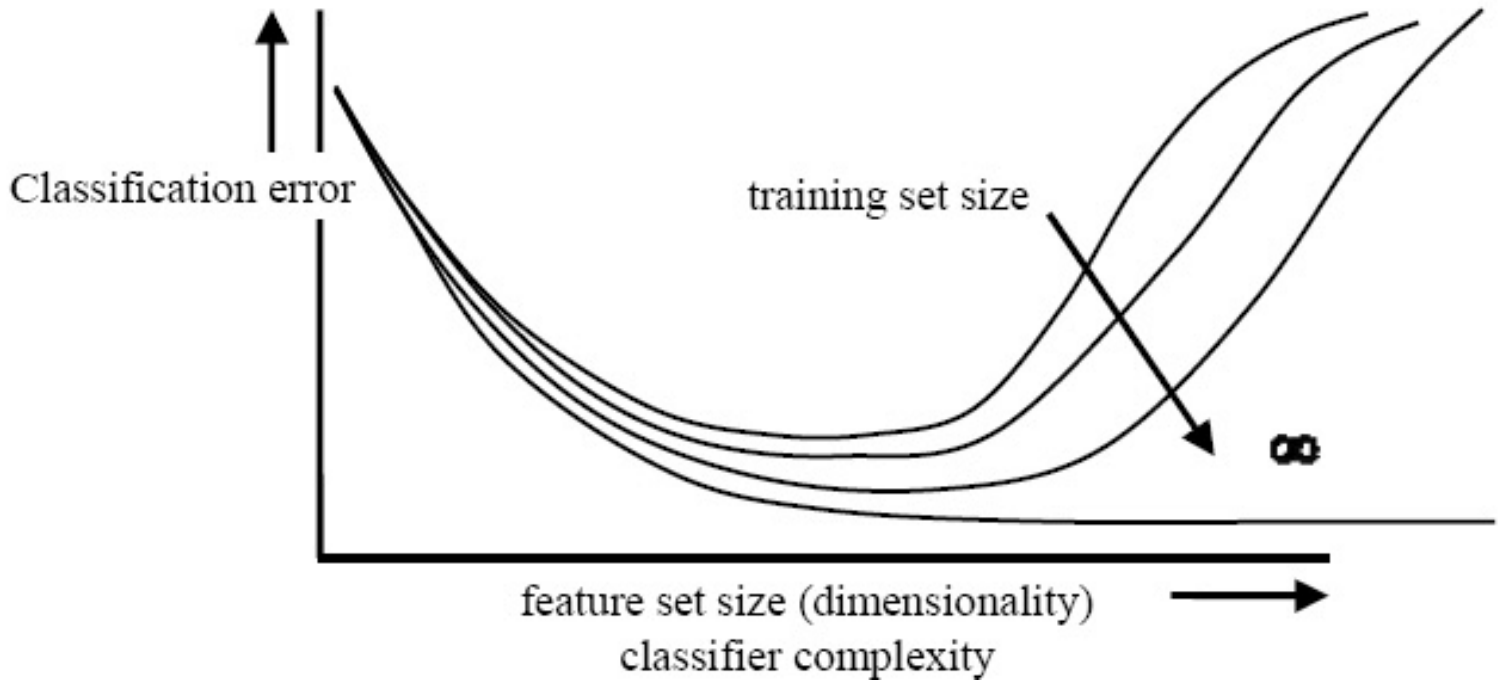
Learning Evaluation



More complex classifiers are better in case of large training sets
and worse in case of small training sets

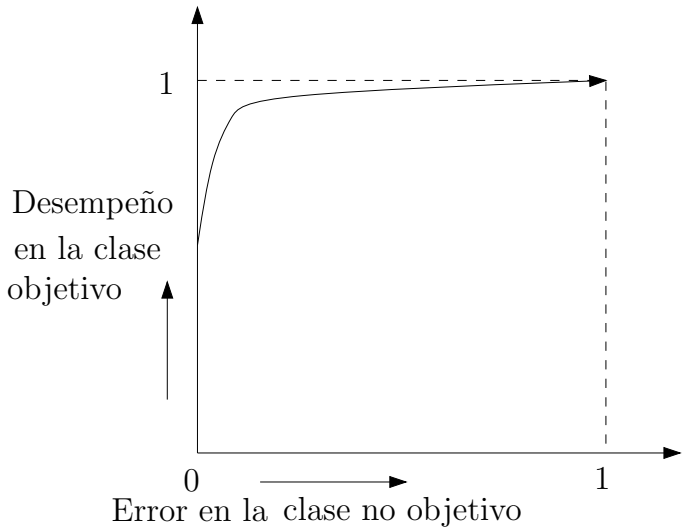
Learning Evaluation

The curse of dimensionality

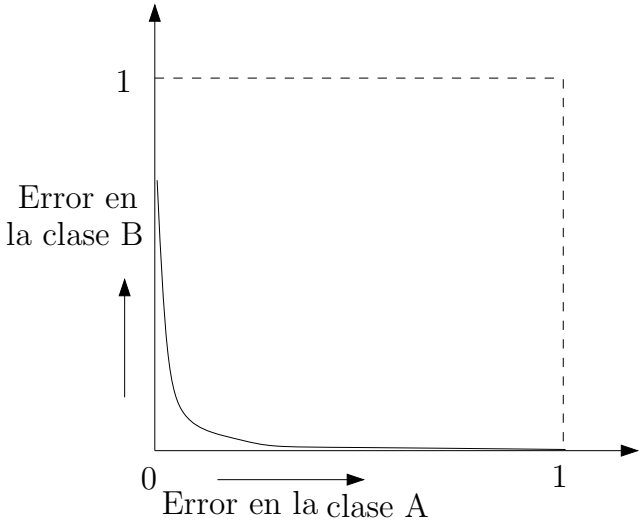


Visualization of Learning Performance

ROC (Receiver Operating Characteristic) Curves



Diagnóstico médico



Reconocimiento de patrones
en problemas de dos clases

Visualization of Learning Performance[3]

Example of ROC curves: Our task will be to diagnose 100 patients with a disease present in 50% of the general population. We will assume a black box model, where we put in information about patients and receive a score between 0 and 1. We can alter the threshold for labeling a patient as positive (has the disease) to maximize the classifier performance.

[3] From: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

Visualization of Learning Performance[4]

Example of ROC curves:

Threshold	TP	FP	TN	FN
0.0	50	50	0	0
0.1	48	47	3	2
0.2	47	40	9	4
0.3	45	31	16	8
0.4	44	23	22	11
0.5	42	16	29	13
0.6	36	12	34	18
0.7	30	11	38	21
0.8	20	4	43	33
0.9	12	3	45	40
1.0	0	0	50	50

Outcome of model at each threshold

[4] From: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

Visualization of Learning Performance[5]

Example of ROC curves:

Threshold =0.5	Actual Positives	Actual Negatives
Predicted Positives	42 (TP)	16 (FP)
Predicted Negatives	13 (FN)	29 (TN)

Confusion Matrix for Threshold of 0.5

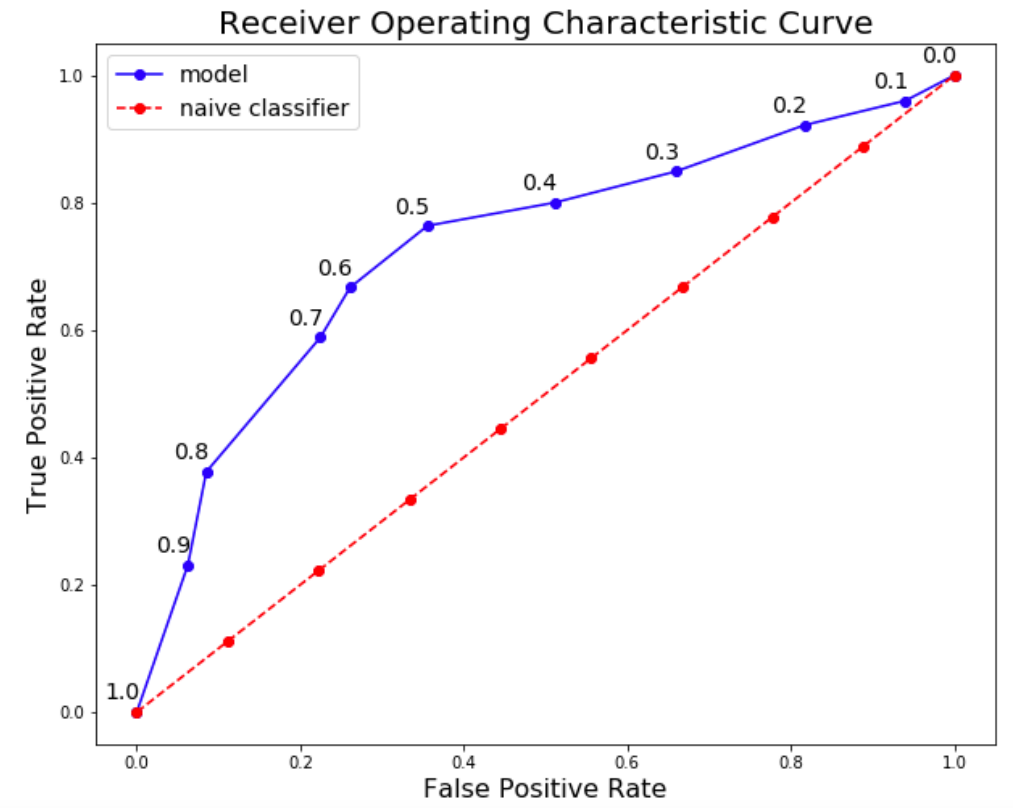
$$recall = \frac{TP}{TP+FN} = \frac{42}{42+13} = 0.76 \quad precision = \frac{TP}{TP+FP} = \frac{42}{42+16} = 0.724 \quad F1\ Score = 2 * \frac{precision*recall}{precision+recall} = 0.74$$

$$true\ positive\ rate = \frac{TP}{TP+FN} = \frac{42}{42+13} = 0.76 \quad false\ positive\ rate = \frac{FP}{FP+TN} = \frac{16}{16+29} = 0.36$$

[5]From: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

Visualization of Learning Performance

Example of ROC curves:



Visualization of Learning Performance

Example of ROC curves:

threshold	recall	precision	f1	tpr	fpr
0.0	1	0.5	0.666667	1	1
0.1	0.96	0.505263	0.662069	0.96	0.94
0.2	0.921569	0.54023	0.681159	0.921569	0.816327
0.3	0.849057	0.592105	0.697674	0.849057	0.659574
0.4	0.8	0.656716	0.721311	0.8	0.511111
0.5	0.763636	0.724138	0.743363	0.763636	0.355556
0.6	0.666667	0.75	0.705882	0.666667	0.26087
0.7	0.588235	0.731707	0.652174	0.588235	0.22449
0.8	0.377358	0.833333	0.519481	0.377358	0.0851064
0.9	0.230769	0.8	0.358209	0.230769	0.0625
1.0	0	0	0	0	0

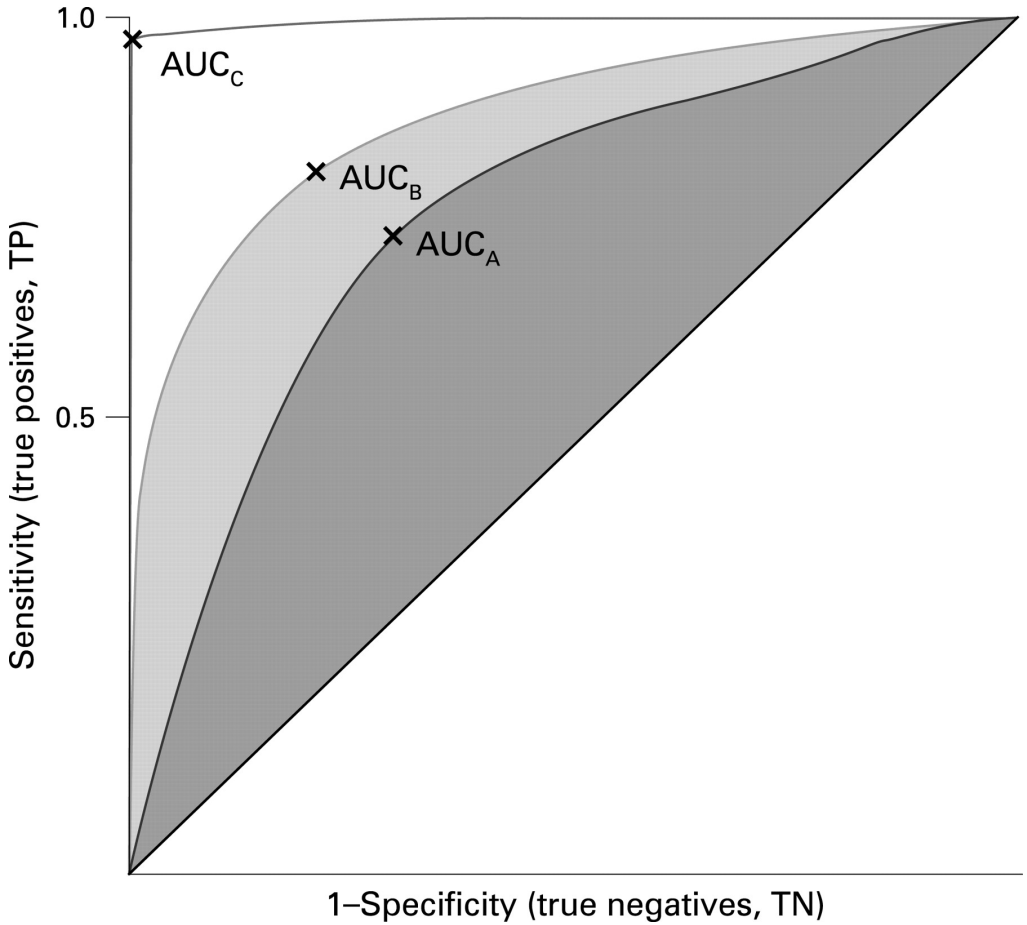
ROC Curves

Inputs: L, the set of test instances; $f(i)$, the probabilistic classifier's estimate that instance i is positive; *min* and *max*, the smallest and largest values returned by f; *increment*, the smallest difference between any two f values.

```
1:  for t = min to max by increment do
2:     FP ← 0
3:     TP ← 0
4:     for i ∈ L do
5:        if f(i) ≥ t then                    /* This example is over threshold */
6:           if i is a positive example then
7:              TP ← TP + 1
8:           else        /* i is a negative example, so this is a false positive */
9:              FP ← FP + 1
10:          end if
11:       end if
12:    end for
13:    Add point (FP/N, TP/P) to ROC curve
14: end for
15: end
```

ROC Curves

Area Under the Curve (AUC)



Ensemble Methods

Condorcet's jury theorem:
Suppose there are 25 classifiers:

- Each one with an error rate of $\epsilon = 0,35$
- All of them are independent
- Probability that ensemble classifier makes a wrong prediction is: $\sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1-\epsilon)^{25-i} = 0,06$

A group of not very precise models can significantly improve their performance working in group. In fact, the probability tends to cero as the number of models increases.

Types of Ensemble Methods

- By manipulating the training set (bagging, boosting)
- By manipulating the input features (random forest)
- By manipulating the class labels (binary relabeling and voting)
- By manipulating the learning algorithm (variants of the same method)

Ensemble Methods

Bayes Optimal Classifier:

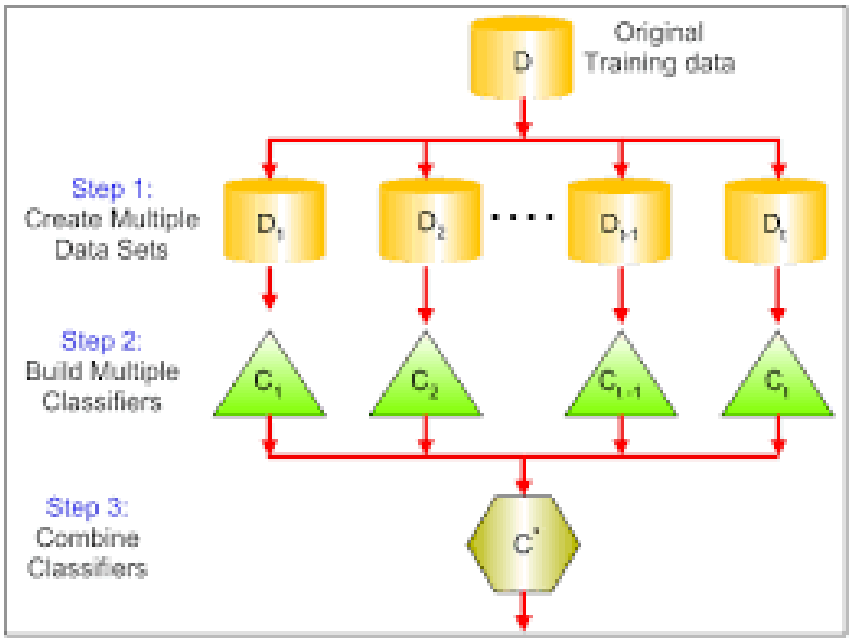
$$y = \underset{c_j \in C}{\operatorname{argmax}} \sum_{h_i \in H} P(c_j|h_i)P(T|h_i)P(h_i)$$

where: y es the predicted class. C is the set of classes, H is the hypothesis space, T is the training data.

Ensemble Methods[6]

Bootstrap aggregating (Bagging):

- Each model in the ensemble vote with equal weight
- Each model in the ensemble is trained using a randomly drawn subset of the training set
- Drawing with replacement! (63% of data are unique)



[6]Image from: https://slideplayer.com/slide/10747814/

Ensemble Methods

Boosting:

- Designed to improve performance of weak learners ($acc > 0,5$) example: Adaboost
- Examples have weights
- At each iteration a new model is produced and the examples are reweighed to prioritize the misclassified examples
- During testing, each model get a weighted vote proporcional to their accuracy on training data

Ensemble Methods

Stacking:

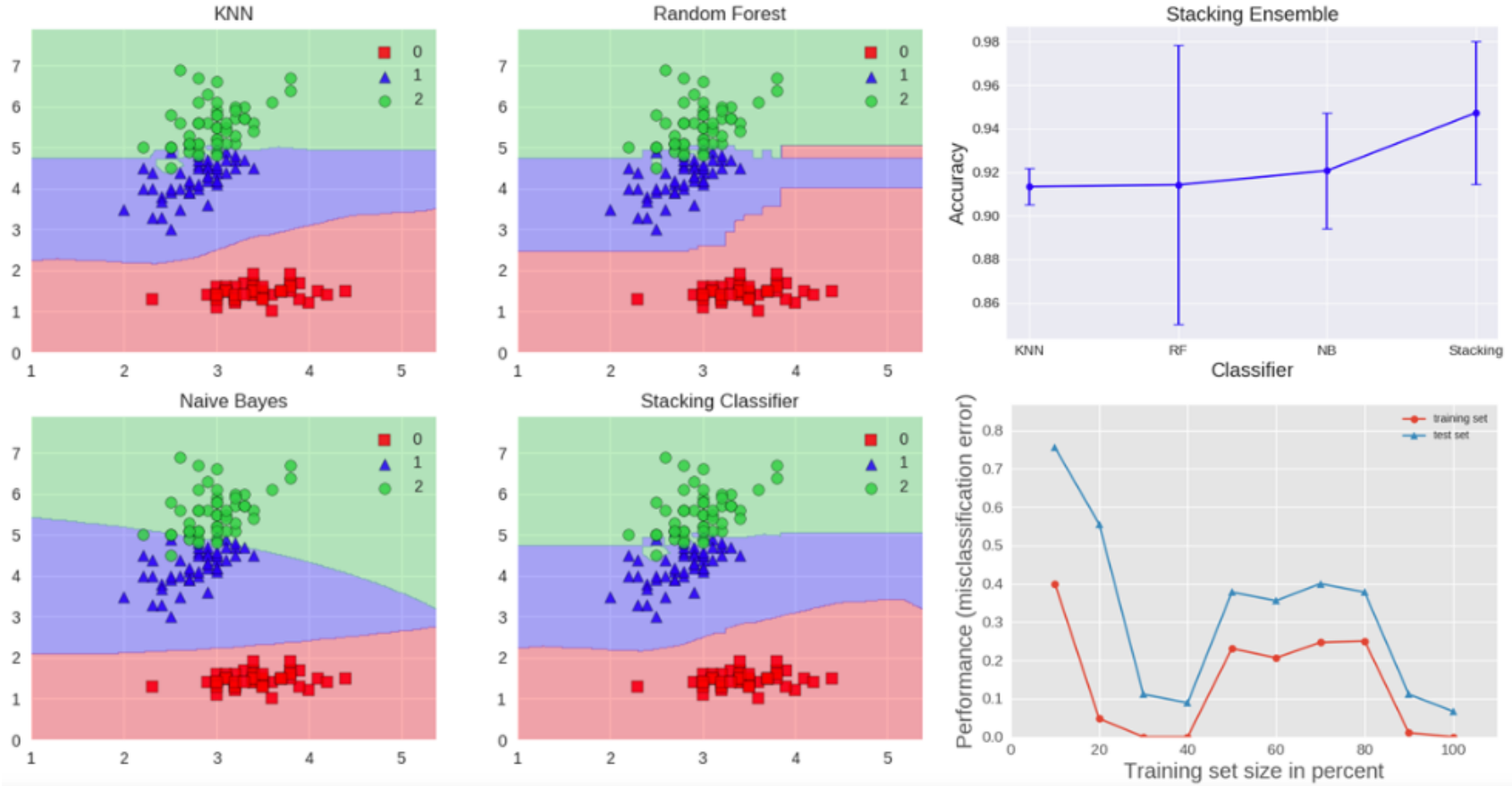
- Combines multiple classification or regression models via a meta-classifier or a meta-regressor
- The base level models are trained based on a complete training set
- The meta-model is trained on the outputs of the base level model as features
- During testing, each model get a weighted vote proporcional to their accuracy on training data

Algorithm Stacking

1: Input: training data $D = \{x_i, y_i\}_{i=1}^{m}$
2: Ouput: ensemble classifier H
3: *Step 1: learn base-level classifiers*
4: **for** $t = 1$ to T **do**
5: learn h_t based on D
6: **end for**
7: *Step 2: construct new data set of predictions*
8: **for** $i = 1$ to m **do**
9: $D_h = \{x'_i, y_i\}$, where $x'_i = \{h_1(x_i), ..., h_T(x_i)\}$
10: **end for**
11: *Step 3: learn a meta-classifier*
12: learn H based on D_h
13: return H

Ensemble Methods[7]

Stacking example



[7]Image from:
https://blog.statsbot.co/ensemble-learning-d1dcd548e936

Exercise

Using data in slide 3 calculate the following:

- Confusion matrix
- Sensitivity, specificity, accuracy
- Precision, recall, F1 score
- What additional information would you need to be able to draw an ROC curve for this classifier?
- How do you evaluate this classifier from the obtained measurements?
- Which method of ensemble would be useful to improve the performance of this classifier? You can make some assumptions.

Today Class

Subject:

- Learning evaluation
- Ensemble methods

Learning Goals:
At the end of the class you should be able to:

- Evaluate que quality of learning in a machine learning system
- Identify some methods to make several models work together in solving a problem

Concepts to Remember

- Confusion matrix
- Precision - recall - F1 score
- ROC curves
- AUC
- Bagging
- Boosting
- Stacking

For the next class: read Torgo chapter 2