On the evaluation of binary classifiers for Software Engineering

Luigi Lavazza

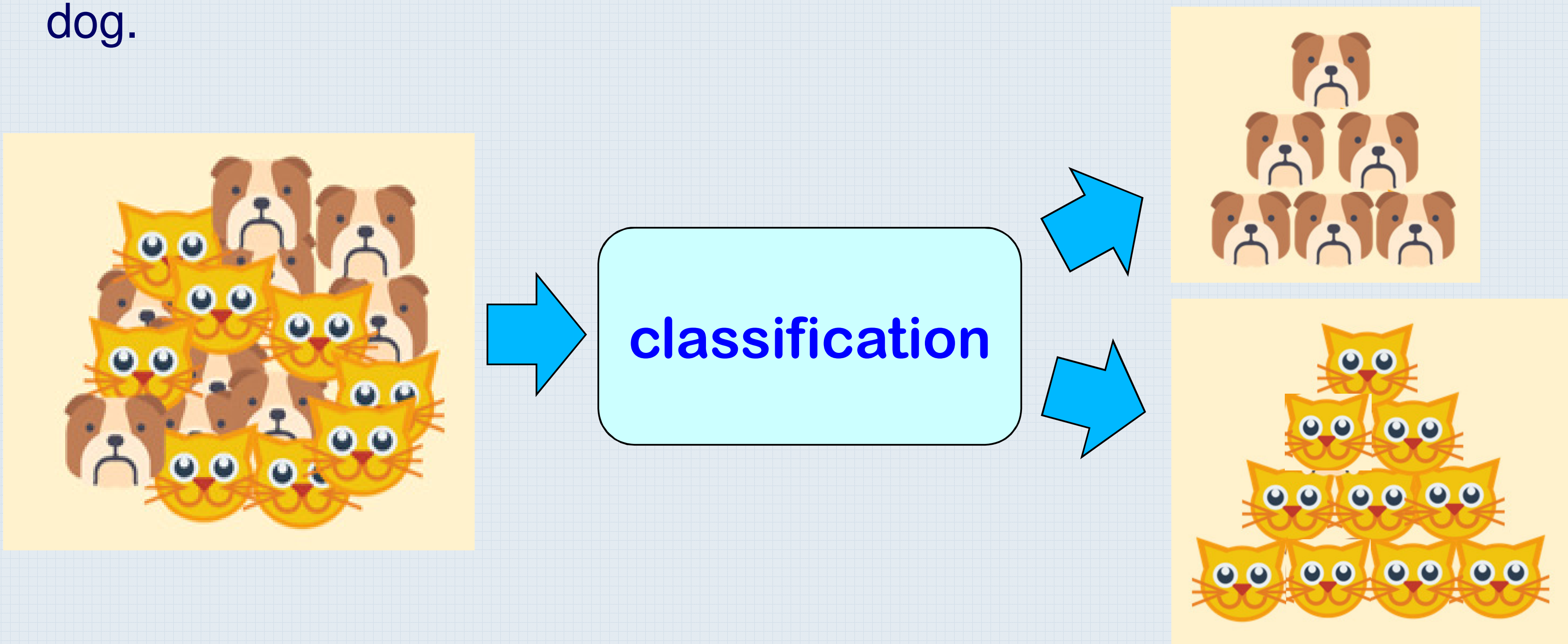
Dipartimento di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria, Varese, Italy
luigi.lavazza@uninsubria.it

Acknowledgment

This talk is based on research work carried out in cooperation with Sandro Morasca

Binary classification

- Binary classification is the task of classifying the elements of a set into two groups on the basis of a classification rule.
- For instance, given a group of animals belonging to two classes (cats and dogs), a binary classifier classifies each elements as either cat or dog.

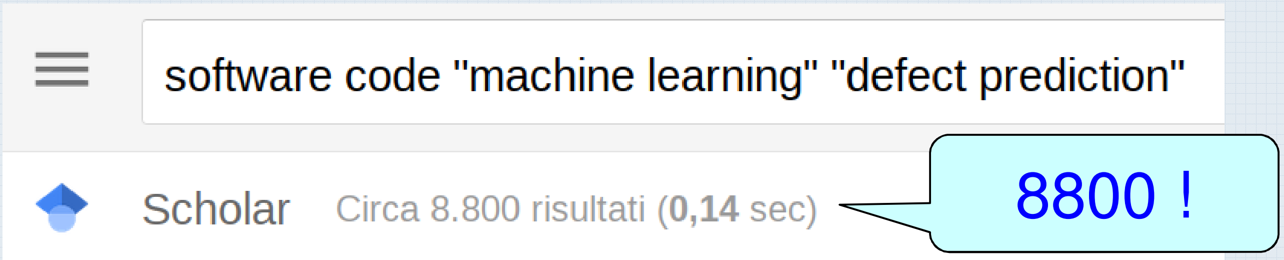


Binary classification in Software Engineering

- In SE, binary classifiers are typically used for several purposes
  - To predict defectiveness of code, to efficiently allocate resources for verification and validation.
  - To identify software modules that are most difficult to maintain, to guide refactoring.
  - To identify vulnerability of code.
  - …

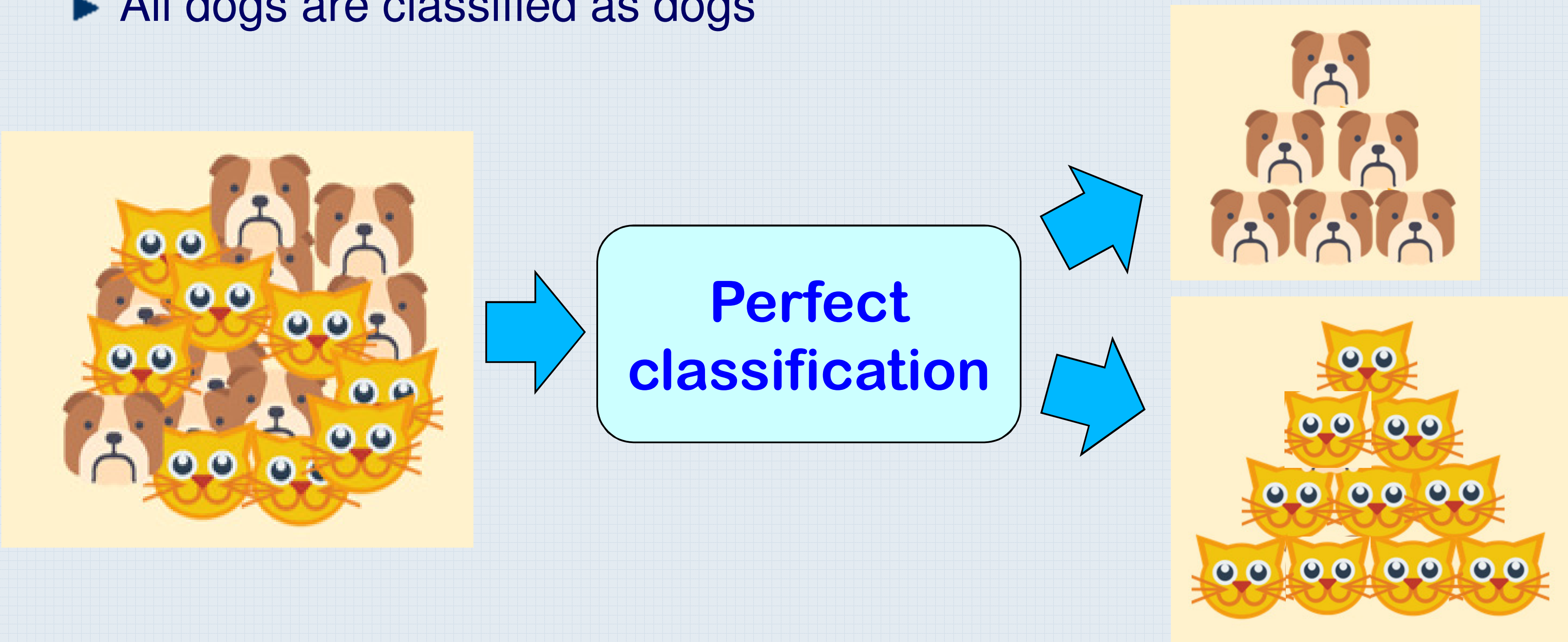
Classifiers

- Classification can be done in various ways
  - By humans
  - Automatically
    - Analogy-based
    - Statistical methods
    - AI methods
    - ...
- Currently, the availability of AI methods has made building binary classifiers a common practice



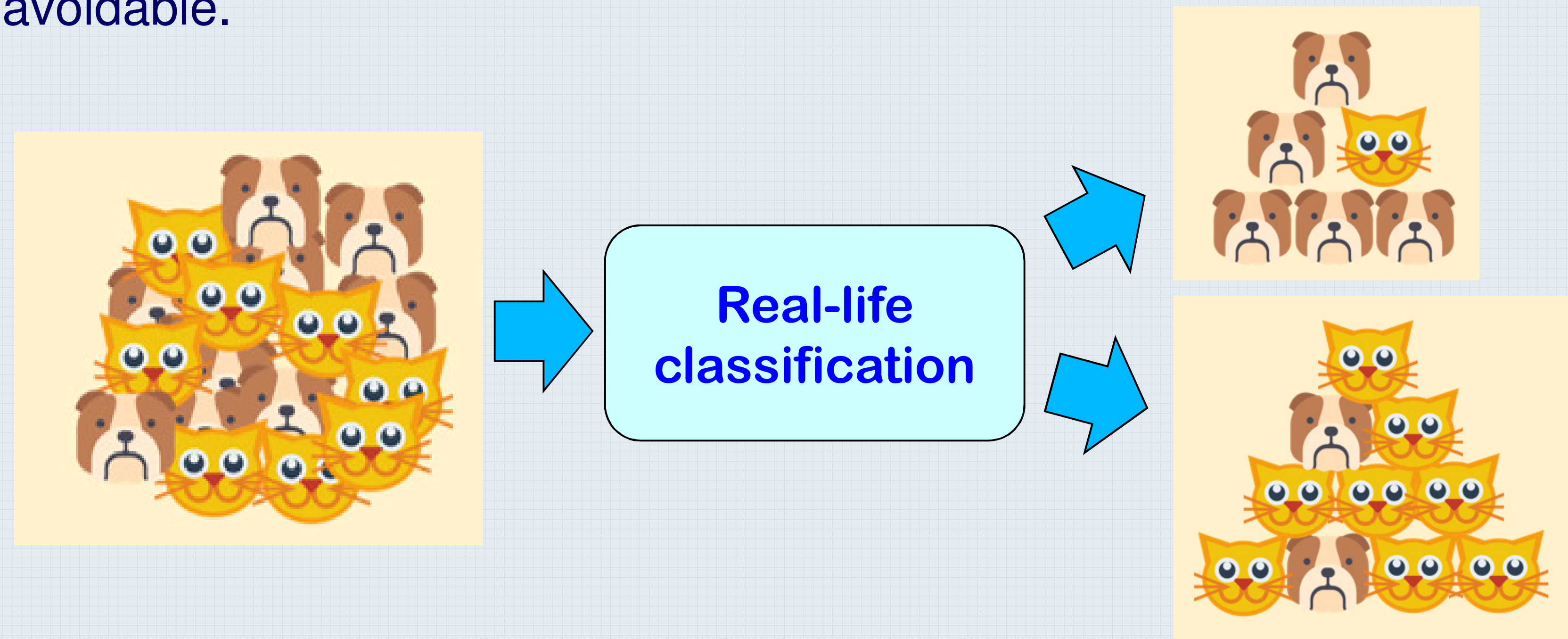
Perfect classifiers?

- Ideally, we would like that all elements are classified correctly, i.e., their class is correctly identified.
  - All cats are classified as cats
  - All dogs are classified as dogs



Classifiers are not perfect

- In general, the correct classification depends in a complex way from a huge number of factors.
  - This is why we use AI, actually.
- The consequence is that in practice, classification errors are not avoidable.



Evaluating classifiers' accuracy

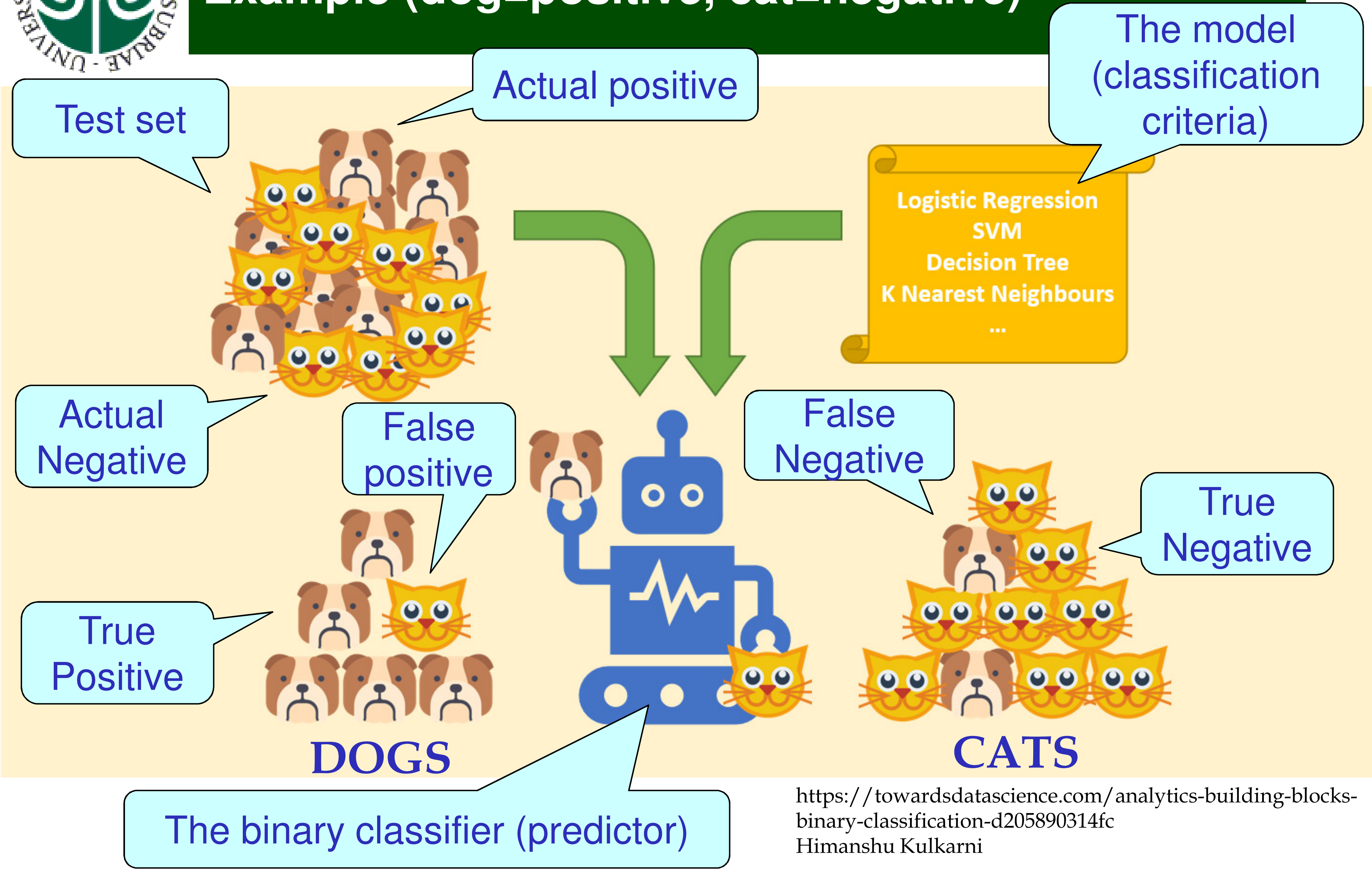
- Practical usage of classifiers requires that we know "how good" a classifier is at correctly guessing the class of the given elements.
- Typical questions:
  - *Is this classifier sufficiently accurate for the intended usage?*
  - *Which of a set of available classifiers is the most accurate?*

- **Accuracy** is the property of correctly guessing the elements' classes.

Terminology

- True positive (TP)
  - An actually positive element is correctly classified positive
- True negative (TN)
  - An actually negative element is correctly classified negative
- False positive (FP)
  - An actually negative element is wrongly classified positive
- False negative (FN)
  - An actually positive element is wrongly classified negative

Example (dog=positive, cat=negative)

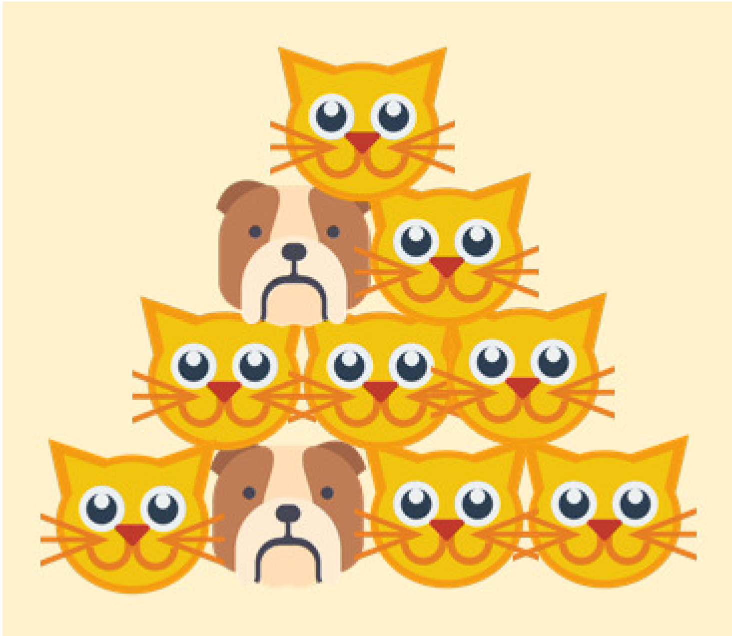


https://towardsdatascience.com/analytics-building-blocks-binary-classification-d205890314fc
Himanshu Kulkarni

Accuracy evaluation problem

Classified as dogs

Classified as cats



How accurate is this classification?

The confusion matrix (CM)

		Actual		
		Negative	Positive	
Estimated	Negative	TN (True Negatives)	FN (False Negatives)	EN = TN + FN (Estimated Negatives)
	Positive	FP (False Positives)	TP (True Positives)	EP = FP + TP (Estimated Positives)
		AN = TN + FP (Actual Negatives)	AP = FN + TP (Actual Positives)	n = AN + AP = EN + EP

The confusion matrix (CM)

Estimated		Actual: Negative	Actual: Positive	
	Negative	TN (True Negatives)	FN (False Negatives)	EN = TN + FN (Estimated Negatives)
	Positive	FP (False Positives)	TP (True Positives)	EP = FP + TP (Estimated Positives)
		AN = TN + FP (Actual Negatives)	AP = FN + TP (Actual Positives)	n = AN + AP = EN + EP

- Note that
  - AP an AN (hence n) are properties of the test set.
  - TP and TN depend on the classifier.
  - TP+FN=AP and TN+FP=AN, hence, given a value per column, the rest of the matrix is determined

Prevalence

- The rate of actual positives is named prevalence, and indicated as $\rho = AP/n = AP/(AP+AN)$
- Prevalence is a property of the test set
- Prevalence is important, because performance metrics depend on it.

The confusion matrix (CM)

- The confusion matrix provides the complete representation of a classifier’s performance
  - For a given dataset
    - note that the dataset is always known, otherwise we would not have a classification to evaluate

Performance metrics

- Performance metrics (alias, accuracy indicators) were introduced
  - Because there is no absolute ordering among CMs
    - A CM may have less FP and more FN than another CM:

CM1

AN=100	AP=100	
TN=80	FN=20	EN=100
FP=20	TP=80	EP=100

CM2

AN=100	AP=100	
TN=75	FN=15	EN=90
FP=25	TP=85	EP=110

  - To get a synthetic, one-number indicator

- Performance metrics try to "condense" the confusion matrix into a single number
- <u>All performance metrics are computed based on the confusion matrix</u>

How many performance metrics?

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = 1-\mathrm{FNR}$$

specificity, selectivity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} = 1-\mathrm{FPR}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} = 1-\mathrm{FDR}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}} = 1-\mathrm{FOR}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{P}} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}} = 1-\mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} = 1-\mathrm{TNR}$$

false discovery rate (FDR)

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TP}} = 1-\mathrm{PPV}$$

false omission rate (FOR)

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TN}} = 1-\mathrm{NPV}$$

Positive likelihood ratio (LR+)

$$\mathrm{LR+} = \frac{\mathrm{TPR}}{\mathrm{FPR}}$$

Negative likelihood ratio (LR-)

$$\mathrm{LR-} = \frac{\mathrm{FNR}}{\mathrm{TNR}}$$

prevalence threshold (PT)

$$\mathrm{PT} = \frac{\sqrt{\mathrm{TPR}(-\mathrm{TNR}+1)}+\mathrm{TNR}-1}{(\mathrm{TPR}+\mathrm{TNR}-1)} = \frac{\sqrt{\mathrm{FPR}}}{\sqrt{\mathrm{TPR}}+\sqrt{\mathrm{FPR}}}$$

threat score (TS) or critical success index (CSI)

$$\mathrm{TS} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}+\mathrm{FP}}$$

Prevalence

$$\frac{\mathrm{P}}{\mathrm{P}+\mathrm{N}}$$

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{P}+\mathrm{N}} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$

balanced accuracy (BA)

$$\mathrm{BA} = \frac{TPR+TNR}{2}$$

F1 score

is the harmonic mean of precision and sensitivity:

$$\mathrm{F}_1 = 2\times\frac{\mathrm{PPV}\times\mathrm{TPR}}{\mathrm{PPV}+\mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$$

phi coefficient (φ or r_φ) or Matthews correlation coefficient (MCC)

$$\mathrm{MCC} = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

Fowlkes–Mallows index (FM)

$$\mathrm{FM} = \sqrt{\frac{TP}{TP+FP}\times\frac{TP}{TP+FN}} = \sqrt{PPV\times TPR}$$

informedness or bookmaker informedness (BM)

$$\mathrm{BM} = \mathrm{TPR}+\mathrm{TNR}-1$$

markedness (MK) or deltaP (Δp)

$$\mathrm{MK} = \mathrm{PPV}+\mathrm{NPV}-1$$

Diagnostic odds ratio (DOR)

$$\mathrm{DOR} = \frac{\mathrm{LR+}}{\mathrm{LR-}}$$

This is just a sample!

source: Wikipedia

Problems

- There are many problems with performance metrics
  or, better, with how they are used.
- Now we will have a quick look at the most frequent problems that can be found in research papers (also published in prestigious venues)

Possible conflicts

PROBLEM

- Given the confusion matrices CM1 and CM2 generated by different classifiers applied to the same dataset, different performance metrics provide conflicting indications.
- An example involving Precision=TP/EP and Recall=TP/AP

CM1

AN=100	AP=100	
TN=80	FN=20	EN=100
FP=20	TP=80	EP=100

CM2

AN=100	AP=100	
TN=70	FN=12	EN=82
FP=30	TP=88	EP=118

- $\text{Precision}_1=0.8 > \text{Precision}_2=0.75$
- $\text{Recall}_1=0.8 < \text{Recall}_2=0.88$

- So, should we trust Precision or Recall?

Possible conflicts (cont'd)

PROBLEM

- The situation does not change if you use "more sophisticated" performance metrics.
- Example involving F-measure (the harmonic mean of Precision and Recall) and ϕ (which is computed based on precision, recall and other metrics):

CM1

AN=100	AP=100	
TN=80	FN=20	EN=100
FP=20	TP=80	EP=100

CM2

AN=100	AP=100	
TN=70	FN=12	EN=82
FP=30	TP=88	EP=118

- $FM_1=0.8 < FM_2=0.81$
- $\phi_1=0.6 > \phi_2=0.59$
- So, should we trust FM or ϕ?

What about random classification?

	AN	AP
EN	(1-ρ) AN	ρ AN
EP	(1-ρ) AP	ρ AP

TPR_{rnd}	ρ
TNR_{rnd}	$(1-\rho)$
FPR_{rnd}	ρ
FNR_{rnd}	$(1-\rho)$
PPV_{rnd}	ρ
NPV_{rnd}	$(1-\rho)$
FOR_{rnd}	ρ
Acc_{rnd}	$\rho^2 + (1-\rho)^2$
BA_{rnd}	$1/2$
$Gmean_{rnd}$	$\sqrt{\rho(1-\rho)}$
GM_{rnd}	$2\ \rho(1-\rho)$
FM_{rnd}	ρ
J_{rnd}	$2\ \rho - 1$
MK_{rnd}	0
ϕ_{rnd}	0

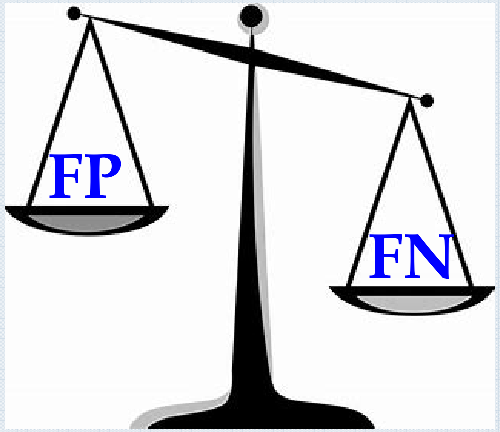
A minimum acceptability threshold

PROBLEM

- If we build a classifier, as a minimum we would like it to be a better predictor than a random classifier.
- Think of module defectiveness prediction based on code measures: why should I measure the code if I am better off throwing dices?
- In most published papers
  - The obtained performance metrics are not compared with the average value that would be obtained by random estimation
    - In some cases, the published performance metrics are actually worse than random
  - The mean of performance metrics obtained from datasets having different prevalence is computed.
    - It does not make sense. You put together indications having different thresholds. A value that “lowers the average” could actually be better then others.

What about costs?

- We are not interested in classification accuracy per se: accuracy is interesting because it affects <u>costs</u>



- Usually, false negatives are much more expensive than false positives.
  - A false positive may lead to additional verifications, testing, inspections or not needed refactoring of already correct code
  - A false negative may lead to releasing a defective module. Usually this costs much more than any superfluous QA.

Performance metrics ignore costs

PROBLEM

- Most performance metrics do not take into account that false positives and false negatives may have (very) different costs.
- Hence, these metrics can be misleading.
  - You may choose a classifier that appears better, but in practice causes greater costs!

Considering cost

- We can use cost as the figure of merit, instead of some abstract metric
- There are many cost models
  - Misclassification cost:

$$MC = FP\ C_{FP} + FN\ C_{FN}$$

    where C_{FN} is the cost of false negatives and C_{FP} is the cost of false positives
  - More sophisticated cost models, that consider the cost of treating true positives, the existence of a budget, etc.

Conclusions

- Research involving new ways of building binary classifiers is very active
  - And we can expect that it will be even more active in the near future.
- Unfortunately, the awareness of the characteristics and limits of performance metrics is very limited.
- We need better ways of representing classification accuracy than traditional performance metrics.
- To this end, the usage of “cost” indicators, directly linked to the usage of classifiers as predictors are promising.
  - What cost model should we use?
  - Are performance metric useful to minimize costs?

Thanks for your attention!

QUESTIONS?