Dipartimento di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria

Advanced accuracy assessment for binary classifiers

Luigi Lavazza

Università degli Studi dell'Insubria, Varese, Italy

luigi.lavazza@uninsubria.it

The 15th International Conference on Software Engineering Advances
October 18 to October 22, 2020 - Porto, Portugal

Luigi Lavazza

- Luigi Lavazza is associate professor of Computer Science at the University of Insubria at Varese, Italy. Formerly he was assistant professor at Politecnico di Milano, Italy. Since 1990 he cooperates with the Software Engineering group at CEFRIEL, where he acts as a scientific consultant in digital innovation projects.
- His research interests include: Empirical software engineering, software metrics and software quality evaluation; Software project management and effort estimation; Software process modeling, measurement and improvement; Open Source Software.
- He was involved in several international research projects, and he also served as reviewer of EU funded projects.
- He is co-author of over 170 scientific articles, published in international journals, or in the proceedings of international conferences or in books.
- He has served on the PC of a number of international Software Engineering conferences; from 2013 to 2018 he was the editor in chief of the IARIA International Journal On Advances in Software.
- He is a IARIA fellow since 2011

Luigi Lavazza: research interests

- Empirical software engineering
  - Evaluation of estimation models' accuracy
- Software metrics and software quality evaluation
- Software project management and effort estimation
- Software process modeling, measurement and improvement
- Open Source Software.

Contents

- Classifiers
- Classification accuracy
- Contingency tables (aka confusion matrices)
- Accuracy indicators
- Random classification performance
- Reference values for accuracy indicators
- The ROC (receiver operating characteristics) curve and the AUC (area under the curve)
- Limits of AUC
- The relevant area of a ROC space
- Relevant areas for various indicators
- Taking Cost into Account
- Examples

Reference

- The most relevant and innovative techniques illustrated in this tutorial are described in the following paper:
  - S. Morasca and L. Lavazza , “On the Assessment of Software Defect Prediction Models via ROC Curves”, Empirical Software Engineering vol. 25, pages 3977–4019 (2020)
  - Available (open access) at
    - https://link.springer.com/article/10.1007/s10664-020-09861-4
- In this tutorial some details are not given. For details, look in the reference paper.

Binary classification

- Binary classification is the task of classifying the elements of a set into two groups on the basis of a classification rule.
- Typical binary classification problems include:
  - Medical diagnostic tests
    - to determine if a patient has a certain disease or not;
  - Quality control in industry
    - deciding whether a specification has been met;
    - In software engineering: predicting whether a given module is defective
  - In information retrieval
    - deciding whether a page should be in the result set of a search or not.
  - …

Classifiers are not perfect

- In general, the correct classification depends in a non-linear way from a huge number of factors.
- However, in practice
  - Not all factors are known
  - The relationships that links factors to outcomes is not completely and perfectly understood.
- The consequence is that some classifications are wrong
  - Almost always. Exceptions are very rare.

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
  - That is, we have a false alarm. Aka Type I error
- False negative (FN)
  - An actually positive element is wrongly classified negative
  - We have a miss: we failed to recognize a really alarming situation. Aka Type II error

Classification errors are not equally important

- Usually, false negatives are much more dangerous than false positives.
- In the medical area:
  - A false positive may lead to additional diagnostic tests or even to not needed cures
  - A false negative may lead to not curing a possibly fatal disease
- In the SE area:
  - 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 more than any superfluous QA.

Confusion matrix

- Alias contingency table

Estimated \ Actual	Negative	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

Confusion matrix

- The confusion matrix summarizes the performance of a classifier applied to a set of phenomena to be classified.
- Ideally, we want that
  - EP=AP, EN=AN,
  - FP=FN=∅
- In practice, we will never get such outcome.
- Hence, we need to represent how good the classification is.
- The confusion matrix provides the complete representation of a classifier's performance
- Quite often, more expressive and synthetic indicators are required.
  - Performance indicators
  - Alias, accuracy indicators

Performance indicators

- Dozens of performance indicators have been defined.
- Here we shall see the most popular ones.

Precision	$\frac{TP}{EP}$	proportion of estimated positives that are actual positives
Recall aka TPR (True Positive Rate)	$\frac{TP}{AP}$	proportion of actual positives that are estimated positive
FM	$\frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$	overall evaluation for positives

Aka F-measure, F-score, F1, etc.

The harmonic mean of Precision and Recall

Performance indicators

- These are the counterparts of Precision, Recall and F-measure for negatives

NPV	$\frac{TN}{EN}$	proportion of estimated negatives that are actual negatives
Specificity	$\frac{TN}{AN}$	proportion of actual negatives that are estimated negative
NM	$\frac{2}{\frac{1}{NPV}+\frac{1}{Specificity}}$	overall evaluation for negatives

Performance indicators

- Fall-out is defined as

$$1 - Specificity = 1 - \frac{TN}{AN} = \frac{AN - TN}{AN} = \frac{FP}{AN}$$

- It is also known as the False Positive Rate (FPR)

Performance indicators

- These are indicators that account for both positive and negative estimations

J	$\frac{TP}{AP} - \frac{FP}{AN}$	overall evaluation for estimated positives
Markedness	$\frac{TP}{EP} - \frac{FN}{EN}$	overall evaluation for actual positives
ϕ	$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{EN \cdot EP \cdot AN \cdot AP}}$	overall evaluation for positives and negatives

The geometric mean of J and Markedness

Performance indicators

- φ is also known as Matthews Correlation Coefficient (MCC)
- MCC is closely related to the χ^2 statistic for a 2×2 contingency table:

$$|\mathrm{MCC}| = \sqrt{\frac{\chi^2}{n}}$$

A few additional performance indicators

Accuracy	Acc	$\frac{TP+TN}{n}$
Jaccard	Jac	$\frac{TP}{n-TN}$
Ochiai-1	Oc1	$\frac{TP}{\sqrt{AP\ EP}}$
Ochiai-2	Oc2	$\frac{TP\ TN}{\sqrt{AP\ AN\ EP\ EN}}$
Tarantula	Tar	$\frac{AN\ TP}{AN\ TP+AP\ FP}$
g-mean (est.)	Ge	$\sqrt{\frac{TP}{EP}\frac{TN}{EN}}$
g-mean (act.)	Ga	$\sqrt{\frac{TP}{AP}\frac{TN}{AN}}$

How to read indicators

- Most indicators have values that range from 0 to 1, where 1 indicates perfect performance and 0 means worst possible performance
- φ ranges between -1 and 1
  - 1=perfect classification
  - 0=worst classification
  - -1=perfect inverse classification (positives are estimated negative and vice versa)
- In general, $\varphi > 0.4$ is considered acceptable.

How representative are performance indicators?

- Several indicators have been criticized.
- E.g., F-measure was conceived in the information retrieval domain. When used in other domains it appears biased,
  - It accounts for positives and their classification
  - It neglects negatives.
- You need to account for both positives and negatives:
  - Via relatively complex indicators that account for both positives and negatives
    - E.g., φ
  - Via pairs of indicators
    - E.g., recall and fall-out (the true positive ratio and the false positive ratio)

Random classifiers

- You could perform the classification by tossing a coin.
  - Like with any other classification, you get a confusion matrix
- If you perform the random classification several times, you can compute the average TP, FP, TN and FN
  - That is, you get a confusion matrix representing the average performance of random estimation
  - This confusion matrix supports the computation of various performance indicators
- For instance, if you are considering a case when AP=AN, and you toss a regular coin (i.e., the probability of positive classification is equal to the probability of negative classification), you get
  - mean(TP)=mean(FP)=mean(TN)=mean(FN)= n/4
  - Accordingly, mean precision=1/2. mean recall=1/2, mean F-measure=1/2, φ=0, etc.

Random classifiers

- Of course, we would like classifiers that perform better than random classifiers.
- So, when a new classifier is proposed, we should check if it performs better than random classification.

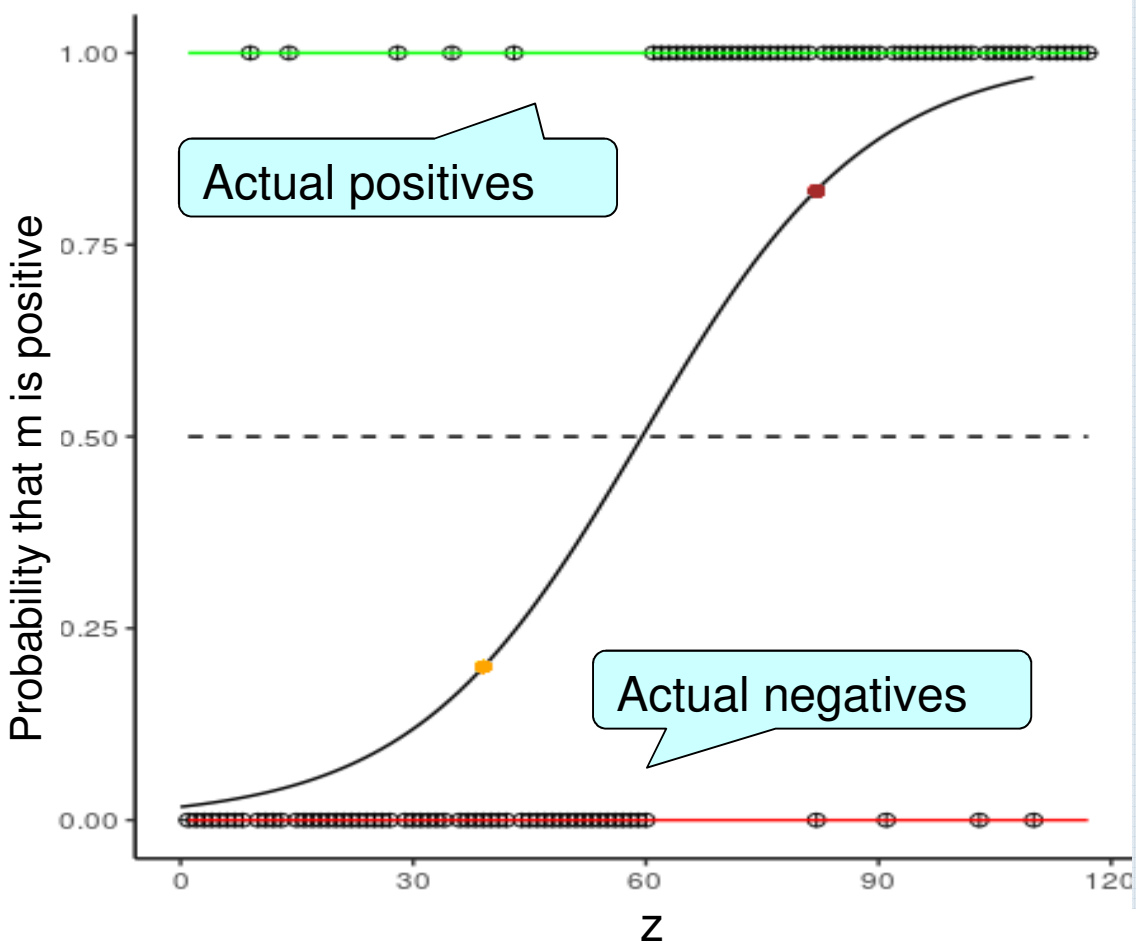
A note on random classification

- Suppose that you want to make predictions concerning a situation characterized by AP < AN.
  - For instance, you wish to predict software modules defectiveness, and you expect (e.g., based on previous projects' data) that the rate of defective modules (AP/n) is 5%.
- In these case, you should use a "coin" that has 0.05 probability of classifying a module positive.
  - E.g., a dice like this

- In what follows we assume that random classification classifies elements positive with probability AP/n

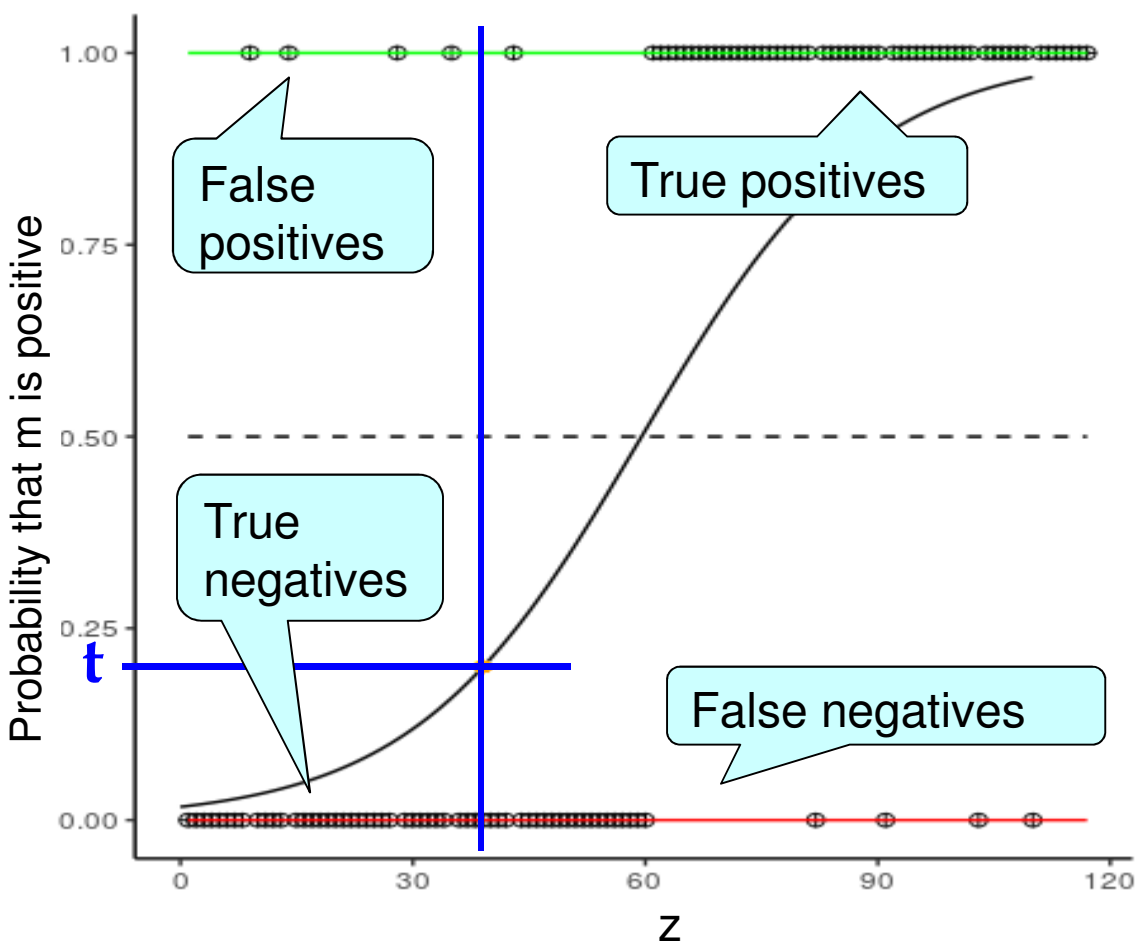
Scoring function

- Given z (which represents some characteristics of module m) function fp(z) yields the probability that m is positive.
- With fp(z) given in the picture
  - the smaller z the more likely that the considered module is negative.
  - The greater z, the more probable that it is positive.
- fp(z) takes into account that actual positives have mostly large z, while actual negatives have mostly small z.



From scoring functions to classifiers

- We can derive a classifier from a scoring function very easily.
- What we need is just a threshold t for the probability that m is positive.
- If fp(z)≥t for module m, then m is classified positive.
- If fp(z)<t for module m, then m is classified negative.
- Clearly, by varying t, we get several different classifiers.
  - Each one characterized by its own confusion matrix



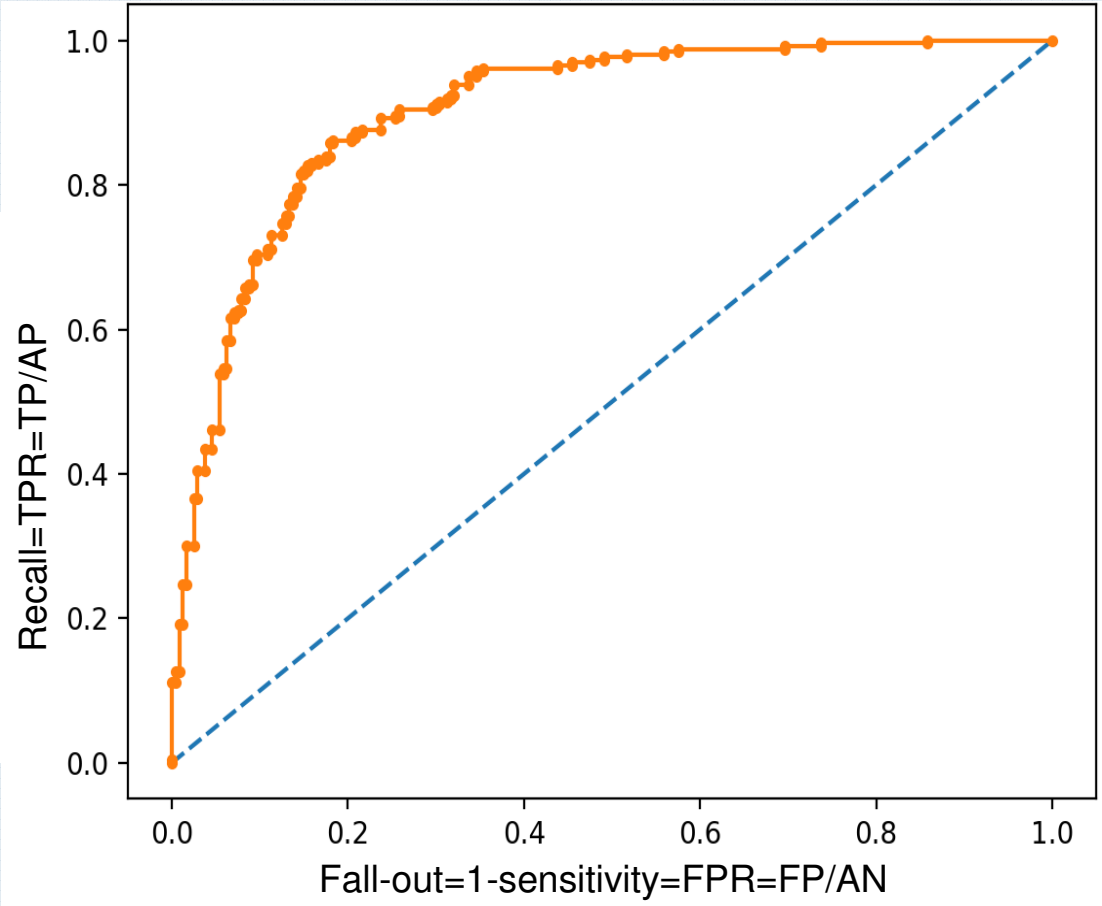
ROC curves

- A ROC curve plots the values of y = Recall against the values of x = Fall-out = 1–Specificity, computed on a test set for all the defect prediction models obtained by using all possible threshold values t.
- The [0, 1] × [0, 1] square to which a ROC curve belongs is called the ROC space.
- Given a dataset, each point (x, y) of the ROC space corresponds to a defect prediction model's confusion matrix,
  - the values of x and y allow the direct computation of TP and FP and the indirect computation of TN and FN, since AP and AN are known.
  - So, given x and y, we know the corresponding confusion matrix and we can compute all other performance indicators.

ROC curves

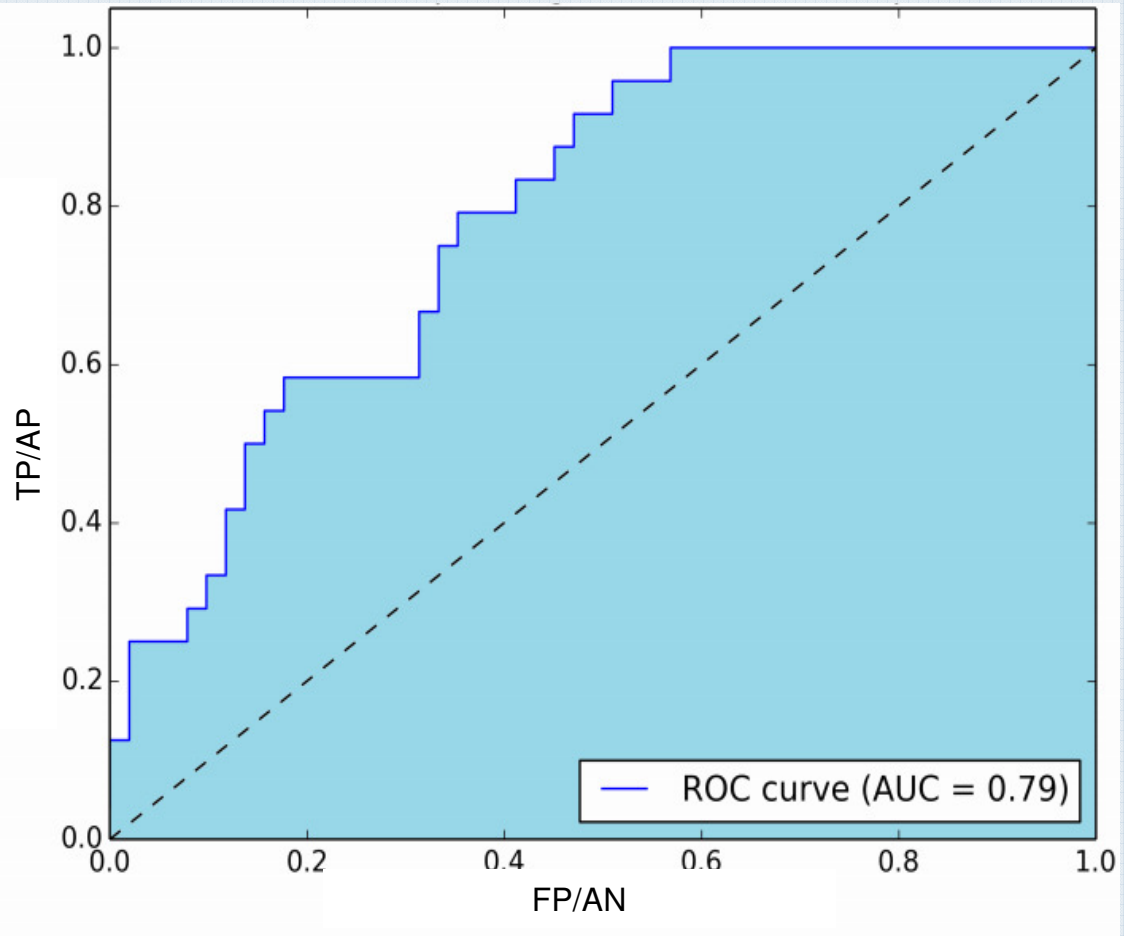
- The two variables x and y are related to t in a (non-strictly) monotonically decreasing way.
- With the increasing fp(z) function in the previous slides,
  - Setting t=1 implies that all elements are estimated negatives.
    - Hence EN=n, EP=0, TP=0, TN=AN, FP=AN-TN=0.
    - So, y=Recall=TP/AP=0, and x=FP/AN=0.
  - Setting t=0 implies that all elements are estimated positives.
    - Hence EP=n, EN=0, TP=AP, FP=EP-TP=n-TP=n-AP=AN
    - Therefore, y=Recall=TP/AP=1 and x=FP/AN=1

A ROC curve


1.0
0.8
0.6
0.4
0.2
0.0
Recall=TPR=TP/AP
0.0
0.2
0.4
0.6
0.8
1.0
Fall-out=1-sensitivity=FPR=FP/AN

Area Under the Curve (AUC)

- The Area Under the Curve is the area below the ROC curve in the ROC space
- Point x=0 y=1 corresponds to perfect classification
- When the ROC curve goes through point (0,1), AUC is 1 (the maximum possible value)
- The closer the ROC curve to point (0,1), the higher the AUC
- The higher the AUC, the better the classifiers' performance



Evaluation of AUC

AUC range	Evaluation
$AUC = 0.5$	totally random, as good as tossing a coin
$0.5 < AUC < 0.7$	poor, not much better than a coin toss
$0.7 \leq AUC < 0.8$	acceptable
$0.8 \leq AUC < 0.9$	excellent
$0.9 \leq AUC$	outstanding

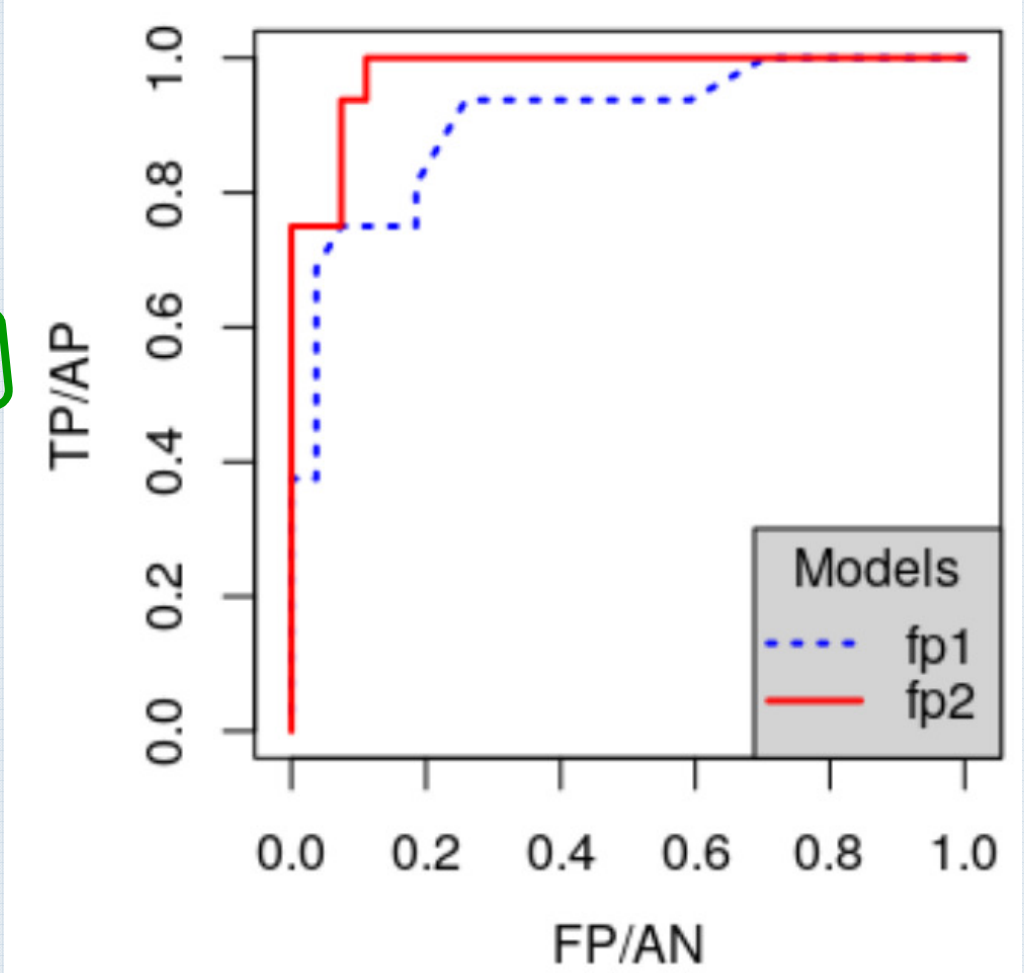
- Ref.
  - Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied logistic regression. John Wiley & Sons (2013)

Using AUC for comparison

- Traditionally, AUC is used to compare classifiers

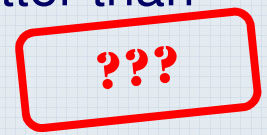
- The AUC of fp2 is greater than the AUC of fp1.
- We should conclude that fp2's performance is better than fp1's.
- In fact, for any t we have that fp2 features not worse than fp1.
  - fp2's ROC curve is never below fp1's.

OK

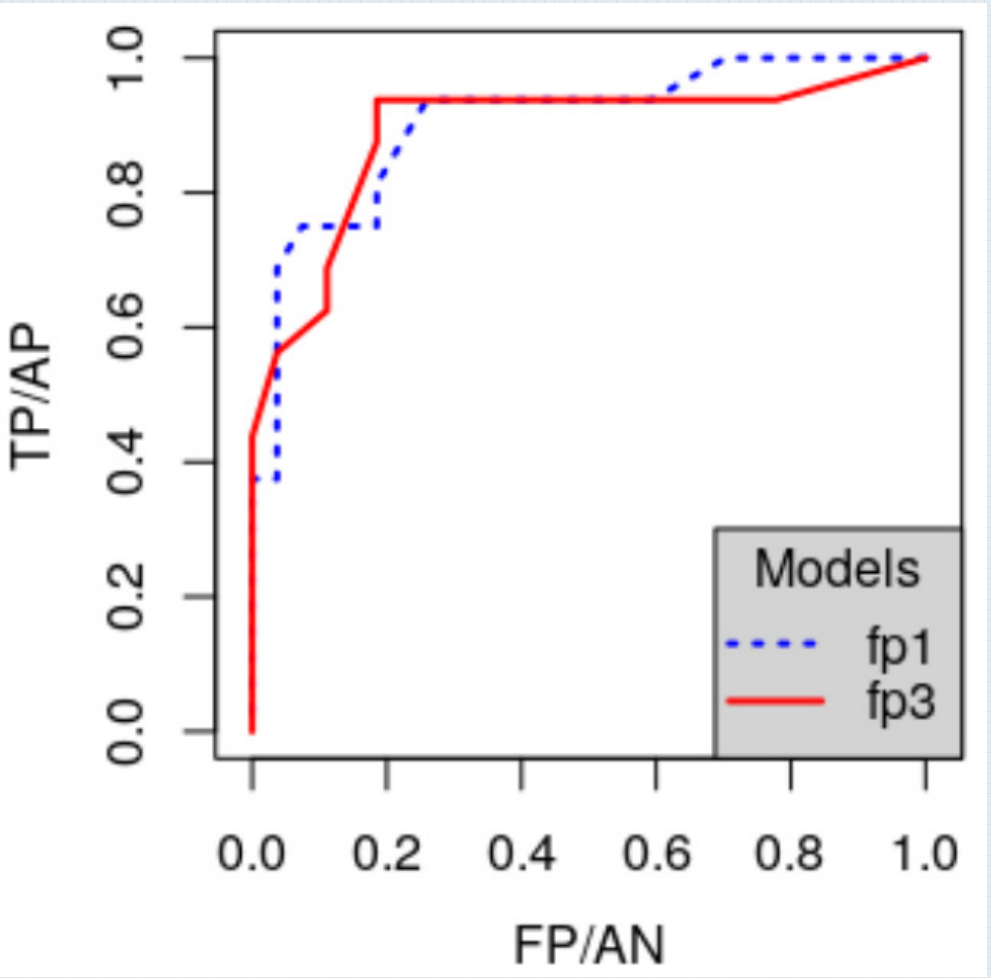


Using AUC for comparison

- A rather controversial case
- The AUC of fp1 is greater than the AUC of fp3.
- We should conclude that fp1's performance is better than fp3's.



- Yes, but
  - fp3's ROC curve gets closer to the (0,1) point than fp1's.
  - fp1's AUC is greater because fp1's ROC curve is above fp3's in the upper right region. Is this region (characterized by many false positives) really relevant?



Relevant areas

- To avoid the problem illustrated above, we consider only a relevant portion of the area under the curve.
- We make reference to random classifications to identify relevant areas, which we name Region of Interest (RoI)

Average performance of random classifications

- As already mentioned, we want classifiers that perform better than the average random classification.
- Now, classifications obtained at random with probability AP/n that an element is positive get the following confusion matrix, representing average values of TP, FP, TN and FN.

	Negative	*Positive*	*Est. total*
Estimated Negative	$\frac{AN^2}{n}$	$\frac{AP\ AN}{n}$	AN
Estimated Positive	$\frac{AP\ AN}{n}$	$\frac{AP^2}{n}$	AP
Act. total	AN	AP	n

Average performance of random classifications

	Negative	*Positive*	*Est. total*
Estimated Negative	$\frac{AN^2}{n}$	$\frac{AP\ AN}{n}$	AN
Estimated Positive	$\frac{AP\ AN}{n}$	$\frac{AP^2}{n}$	AP
Act. total	AN	AP	n

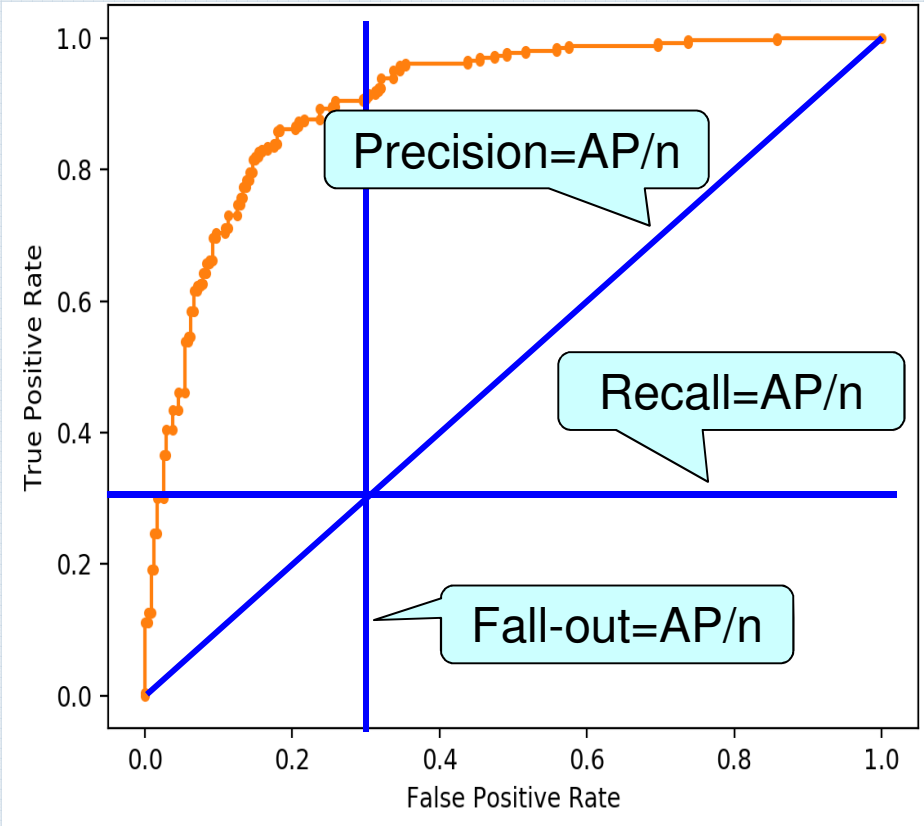
- Based on this confusion matrix, we have that the average values of performance indicators for random classifications are:
  - $Recall = \frac{TP}{AP} = \frac{AP^2}{n\ AP} = \frac{AP}{n}$
  - $Fall\_out = \frac{FP}{AN} = \frac{AP\ AN}{n\ AN} = \frac{AP}{n}$
  - $Precision = \frac{TP}{EP} = \frac{AP^2}{n\ AP} = \frac{AP}{n}$

Constraints in the ROC space

- So, we want classifiers whose performance indicators are:
  - $Recall > \frac{AP}{n}$
  - $Fall\_out < \frac{AP}{n}$
  - $Precision > \frac{AP}{n}$
- How do we represent these constraints in the ROC space?

Indicators in the ROC space

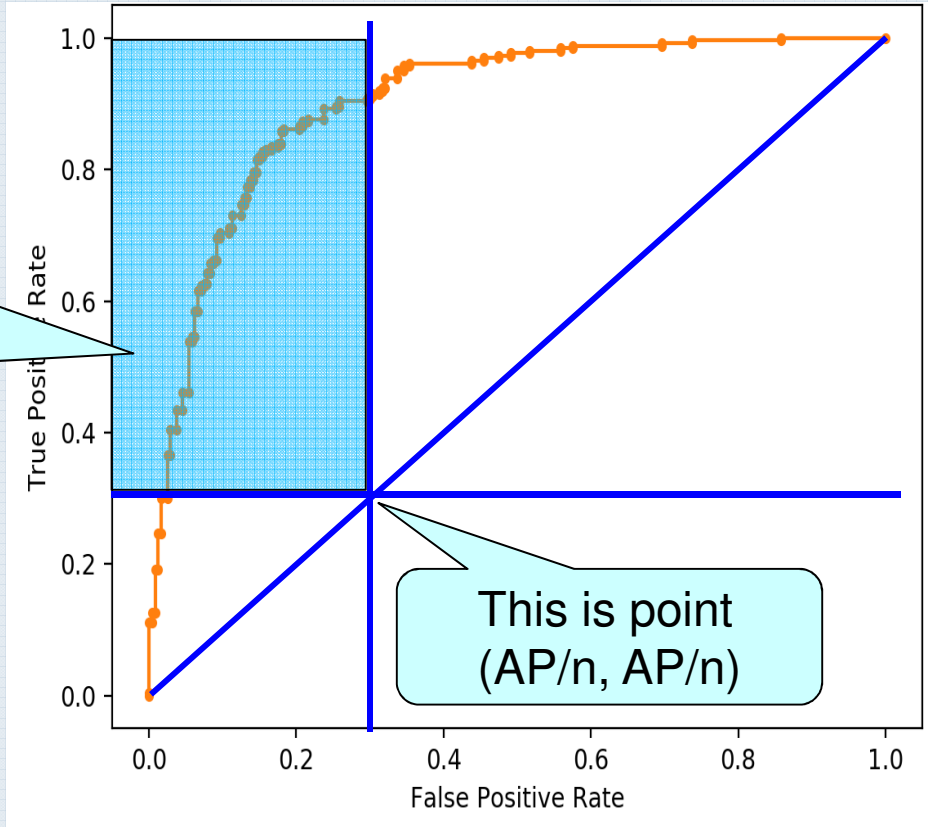
- Recall=AP/n is an horizontal line
- Fall-out=AP/n is a vertical line
- Precision=TP/EP=AP/n is line y=x



Constraints in the ROC space

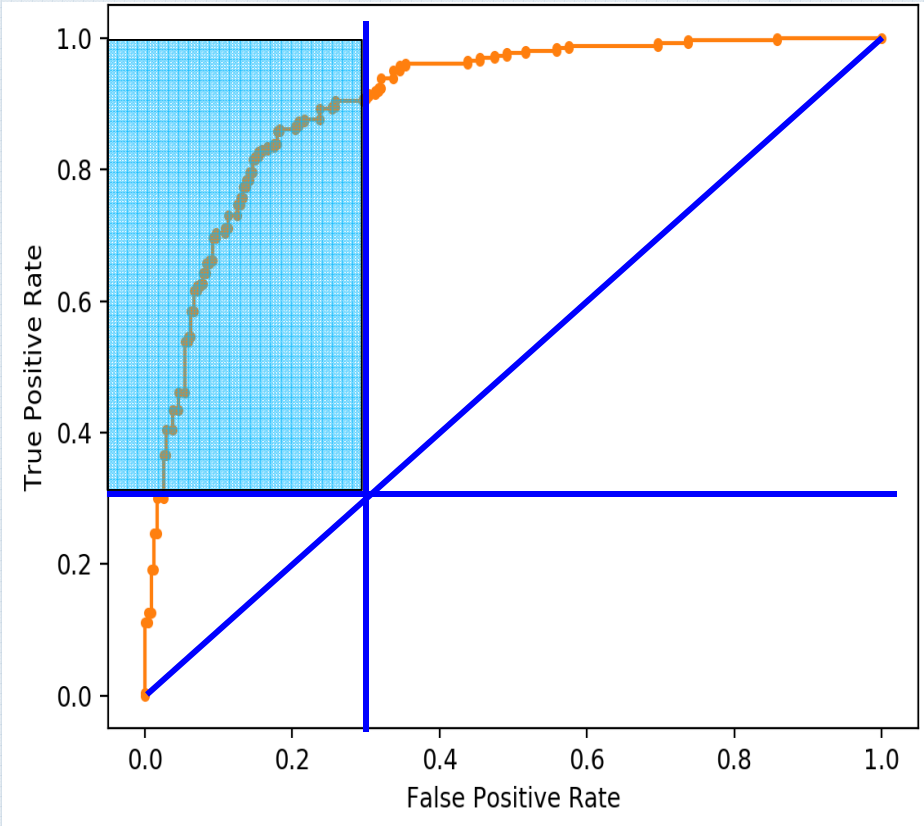
- So, we want classifiers whose performance indicators are:
  - $Recall > \frac{AP}{n}$
  - $Fall\_out < \frac{AP}{n}$
  - $Precision > \frac{AP}{n}$

This is the RoI where Recall, Precision and Fall-out are all better than the average random values



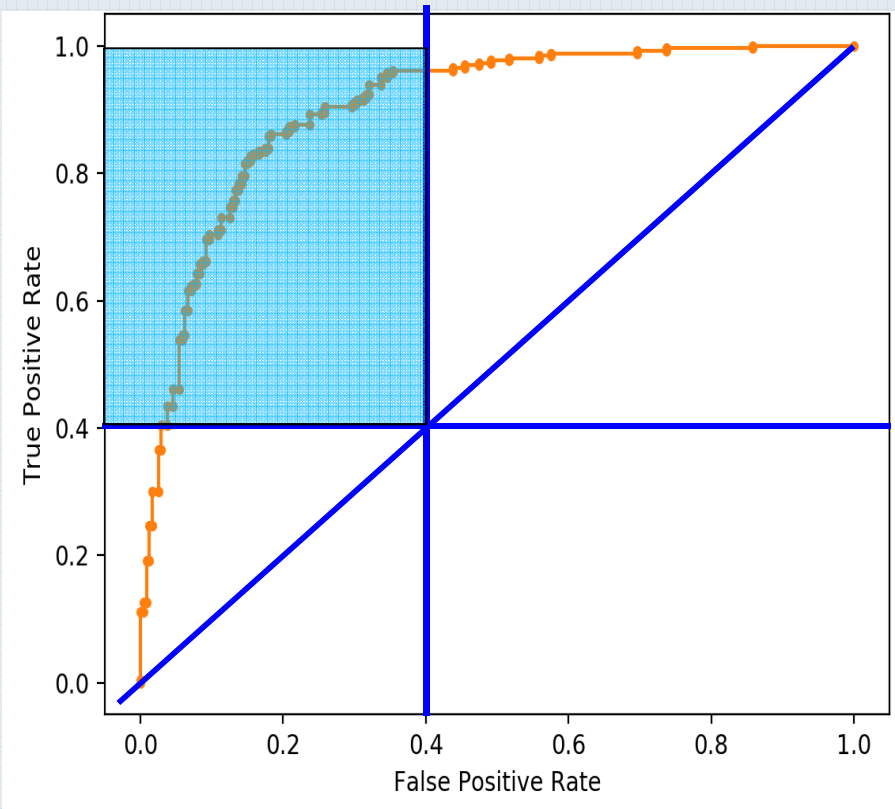
First result

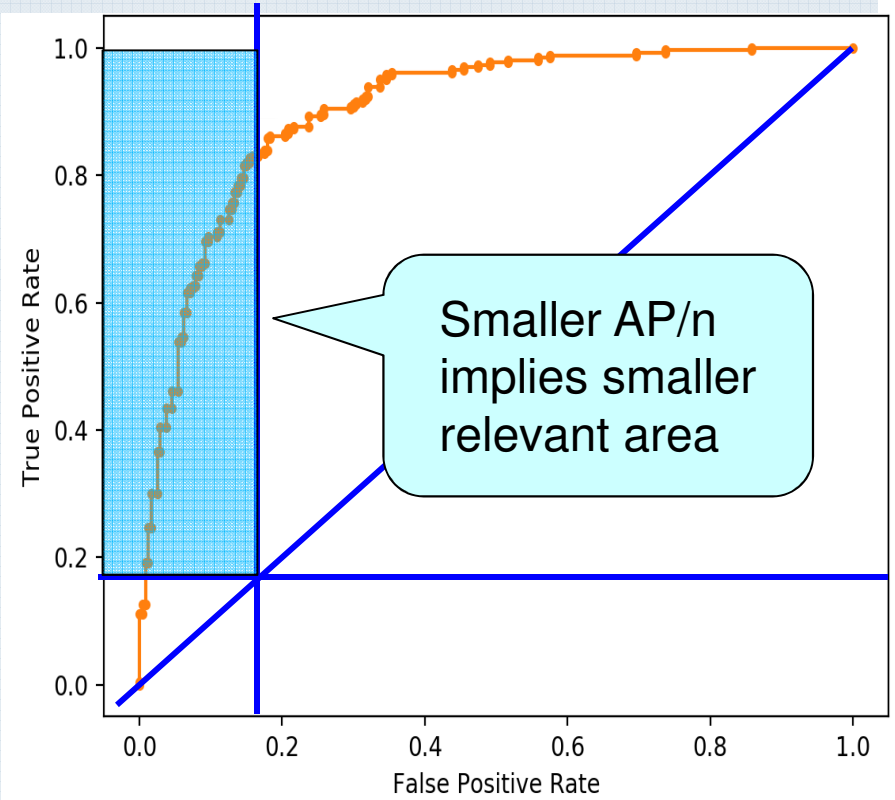
- AUC considers the area under the ROC curve in the entire ROC space
- But the only relevant part of the ROC space is where x<AP/n and y>AP/n
- Therefore, we need a new indicator, which considers only the RoI, where performances are better than the average performance of random classifications



The new indicator

- Just taking the area under the curve in the RoI would not be representative.
- Consider what happens with different values of AP/n





The new indicator

- Ratio of Relevant Areas (RRA)

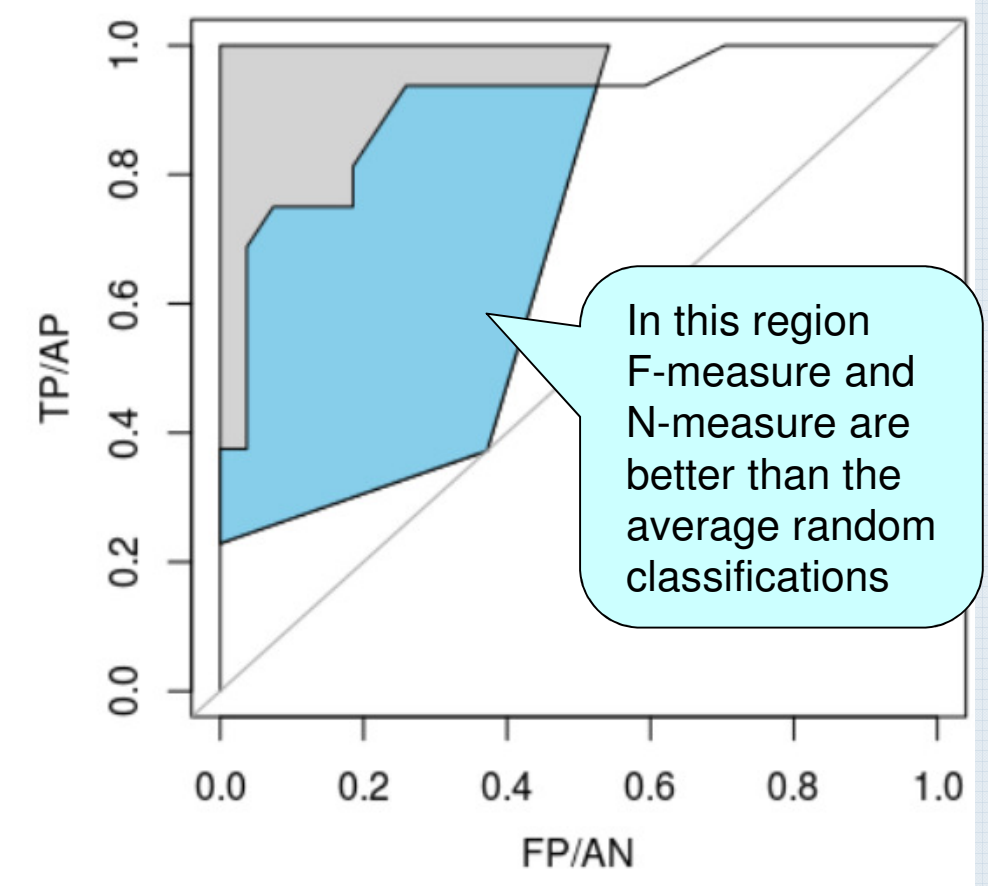
$$\text{RRA} = \frac{\text{area under the ROC curve that belongs to the RoI}}{\text{area of the RoI}}$$

- Since we consider the ratio, it is not that important how large the RoI is.

Defining the RoI

- The RoI can be determined in several different ways

- For instance, one could define the RoI based on F-measure and N-measure instead of Recall and Fall-out.
- Accordingly, we have RRA(Recall, Fallout), RRA(F-measure, N-measure), etc.

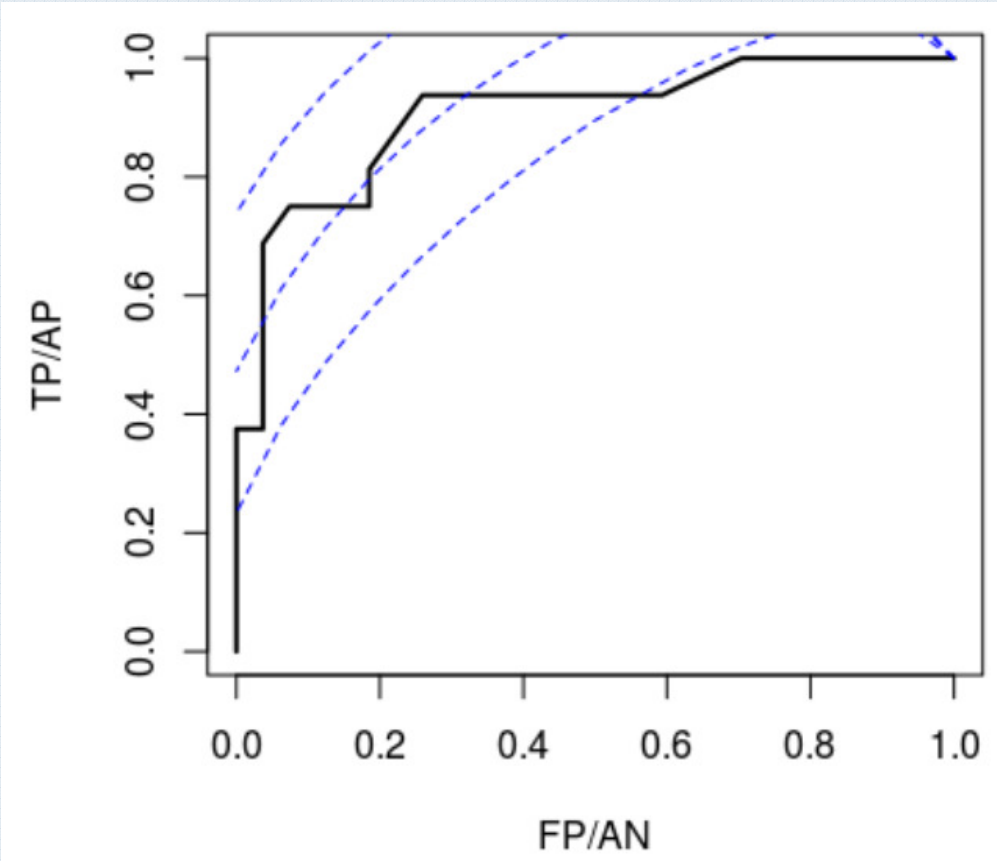


Using non random baselines

- Let us consider the evaluation of performance via φ.
- Random classifications have average $\varphi = 0$.
- So, practically any classifier achieves an accuracy level that is better than the average random classifications' φ.
- In conclusion, random classifiers are not a meaningful baseline when φ is used as the performance indicator.
- On the other hand, it is generally believed that φ should be greater or equal to 0.4 to be acceptable.
- Therefore, we could look for a RoI where $\varphi \geq 0.4$

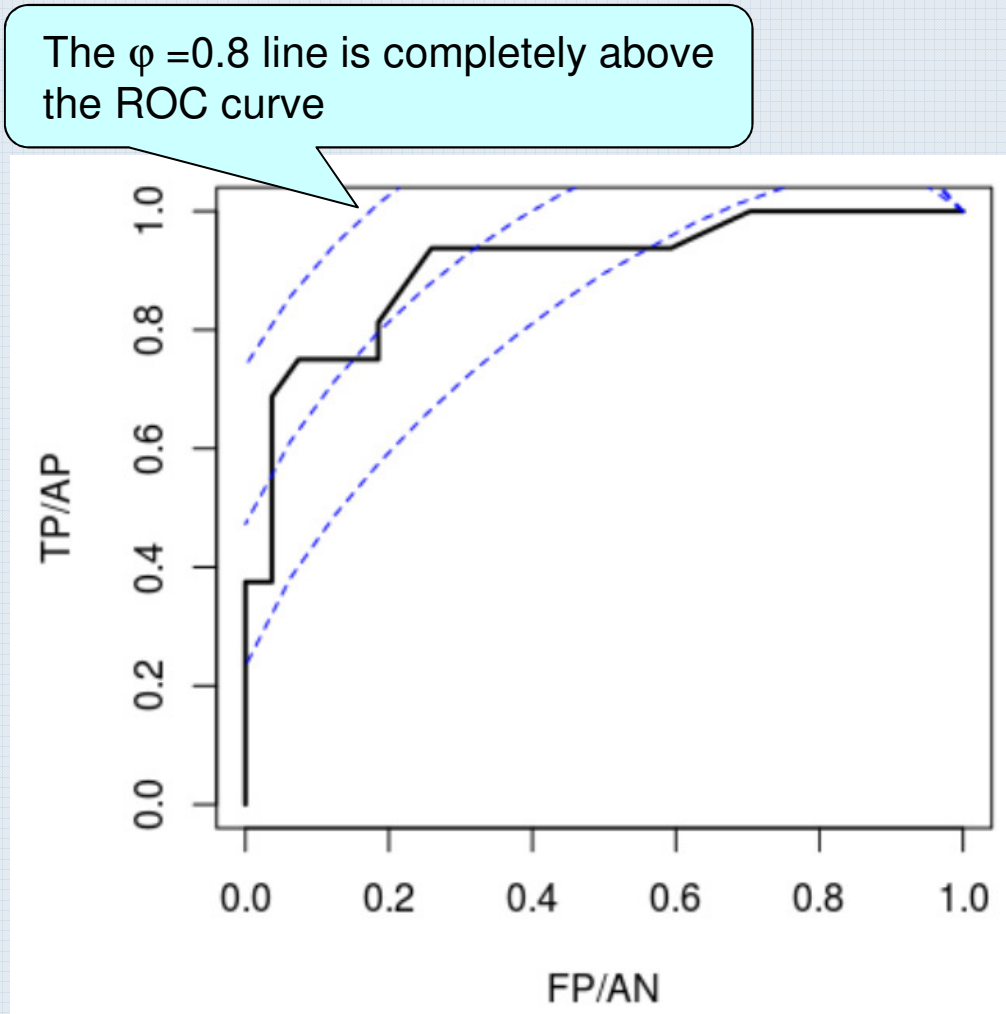
RoI based on φ

- The curve φ = 0.4 happens top be an ellipse in the ROC space
  - All curves φ = constant are ellipses: see details in the reference paper
- The picture shows the lines φ=0.4, φ=0.6 and φ =0.8
  - The greater φ, the higher the curve
  - This is consistent with better performance being close to point (0,1)



RoI based on φ

- Note that the ROC curve shown in the picture has null RRA(φ=0.8), since no part of the ROC curve is above the φ=0.8 line.



Custom definition of the RoI

- You may define the RoI as you consider most appropriate.
- For instance you could consider the region where
  - Recall is better than the average random classifiers'
    and
  - Fallout is better than the average random classifiers'
    and
  - Phi is greater than 0.4

Taking Cost into Account

- Let us assume that
  - each false negative has cost c_{FN}
  - each false positive has cost c_{FP}
- Total cost TC is TC = c_{FN} FN + c_{FP} FP
- By setting $\lambda = \frac{c_{FN}}{c_{FN}+c_{FP}}$ and considering the Normalized Cost $NC = \frac{TC}{n(c_{FN}+c_{FP})}$
- We get $NC = \lambda \frac{1}{1+k}(1-y) + (1-\lambda)\frac{k}{1+k}x$ where $k = \frac{AN}{AP}$
- The average normalized cost of random classifications is $\frac{AP\ AN}{n}$.
- Hence, we want that $\mathrm{NC} < \frac{AP\ AN}{n}$.
  - This inequality defines a RoI

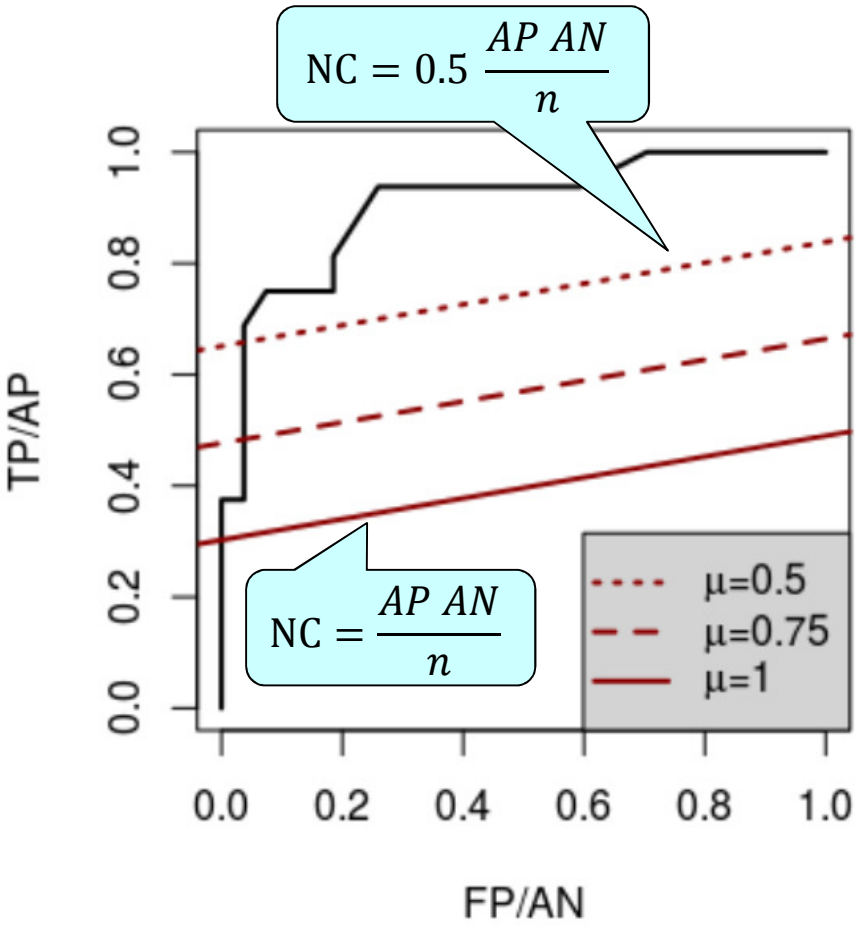
RoI defined by normalized cost

- The equation of $\mathrm{NC} = \frac{AP\ AN}{n}$ In the ROC space is

$$y = \frac{1-\lambda}{l}kx + 1 - \frac{k}{\lambda(1+k)}$$

- In case we want $\mathrm{NC} < \mu\frac{AP\ AN}{n}$ (that is, we want to achieve at least a μ reduction of the normalized cost), we have to consider equation

$$y = \frac{1-\lambda}{l}kx + 1 - \mu\frac{k}{\lambda(1+k)}$$



RoI defined by normalized cost

- We can select as the RoI the region above the NC line characterized by λ and μ.
- Accordingly, we can compute RRA(λ, μ)

RoI defined by normalized cost

- Interestingly, there is a well-defined relationship between the values of λ and performance indicators.

value of λ	Performance indicator
0	Fallout
$\frac{AN}{2n}$	N-measure
$\frac{AN}{n}$	Precision
$\frac{1}{2}+\frac{AN}{2n}$	F-measure
1	Recall

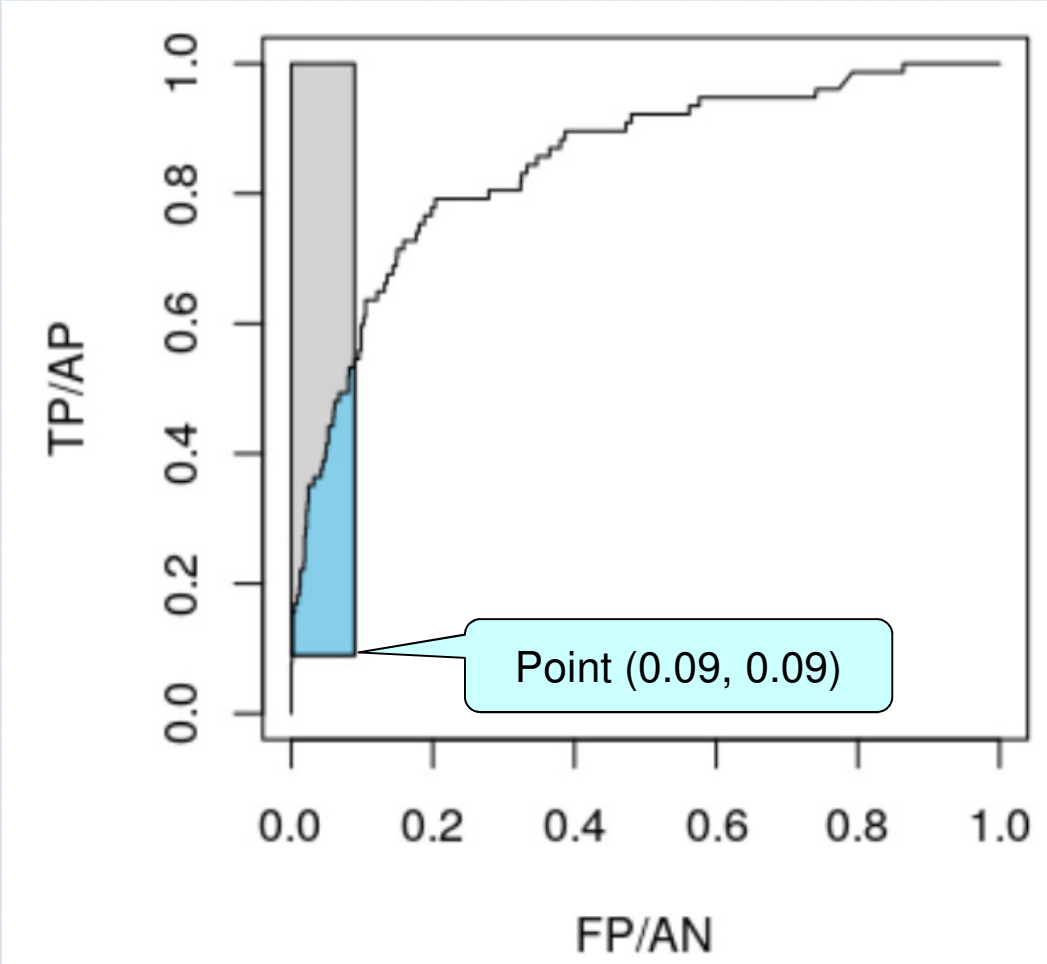
Setting the relative cost of false positives and false negatives equates to choosing a performance indicator!

Example 1: Java class defectiveness

- Data collected by Jureczko and Madeyski and available from the SEACRAFT repository
  - https://zenodo.org/search?page=1&size=20&q=Jureckzo&sort=title
- Population:
  - Java classes of open-source projects
- Data:
  - Defectiveness (yes/no)
  - Static code measures (RFC, LOC, CBO, etc.)

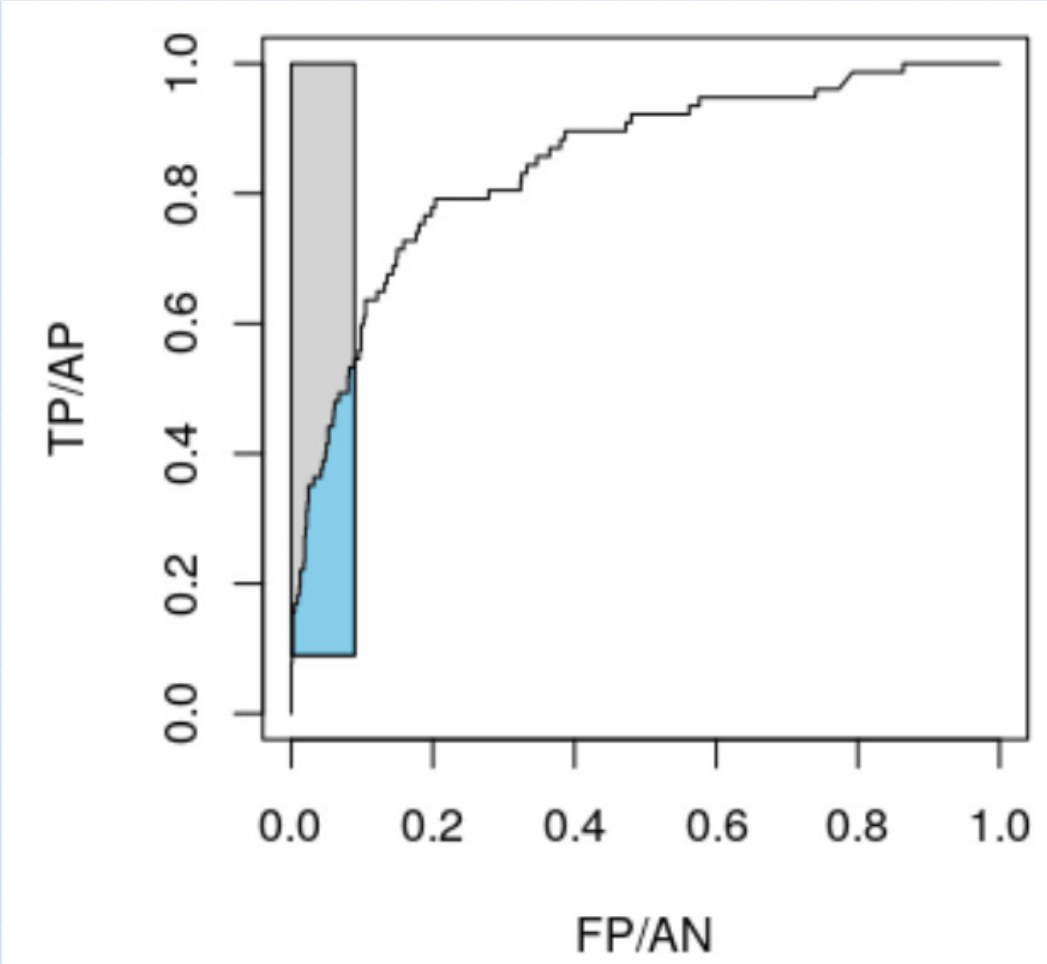
Example

- A classifier of class defectiveness for the Tomcat project
- The code of the Tomcat project is of good quality.
  - The Defectiveness ratio AP/n is 0.09 (i.e., only 9% of the classes were found defective) in the considered release.
- While AUC considers ALL the ROC space, only a small region is actually relevant, when both Recall and Fallout are required to be better than the average random classifiers.



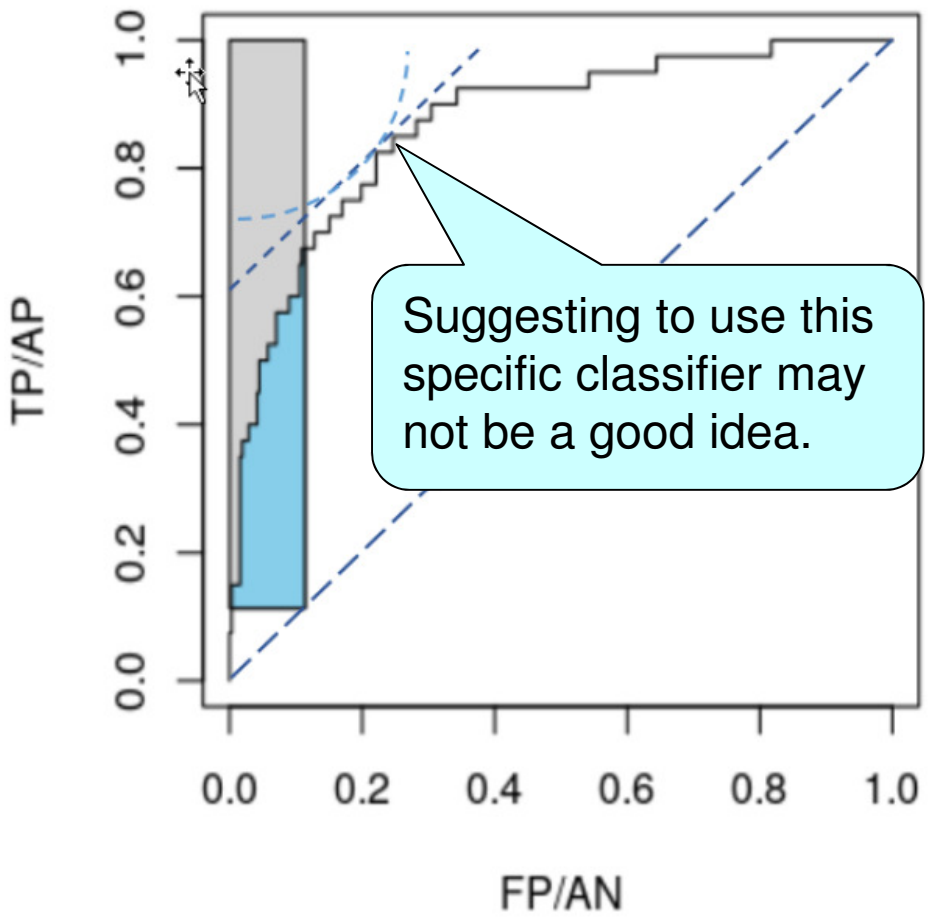
Example

- A classifier of defectiveness for the classes of the Tomcat project
- AUC=0.69, indicating that the considered classifier is close to acceptable.
- However, the indication by AUC appears to be unduly affected by the part of the ROC curve that is not relevant.
- RRA(Recall, Fall-out)=0.23, indicating that the classifier is not very good.
- RRA(φ=0.4) is just above zero, indicating that the classifier is NOT good.



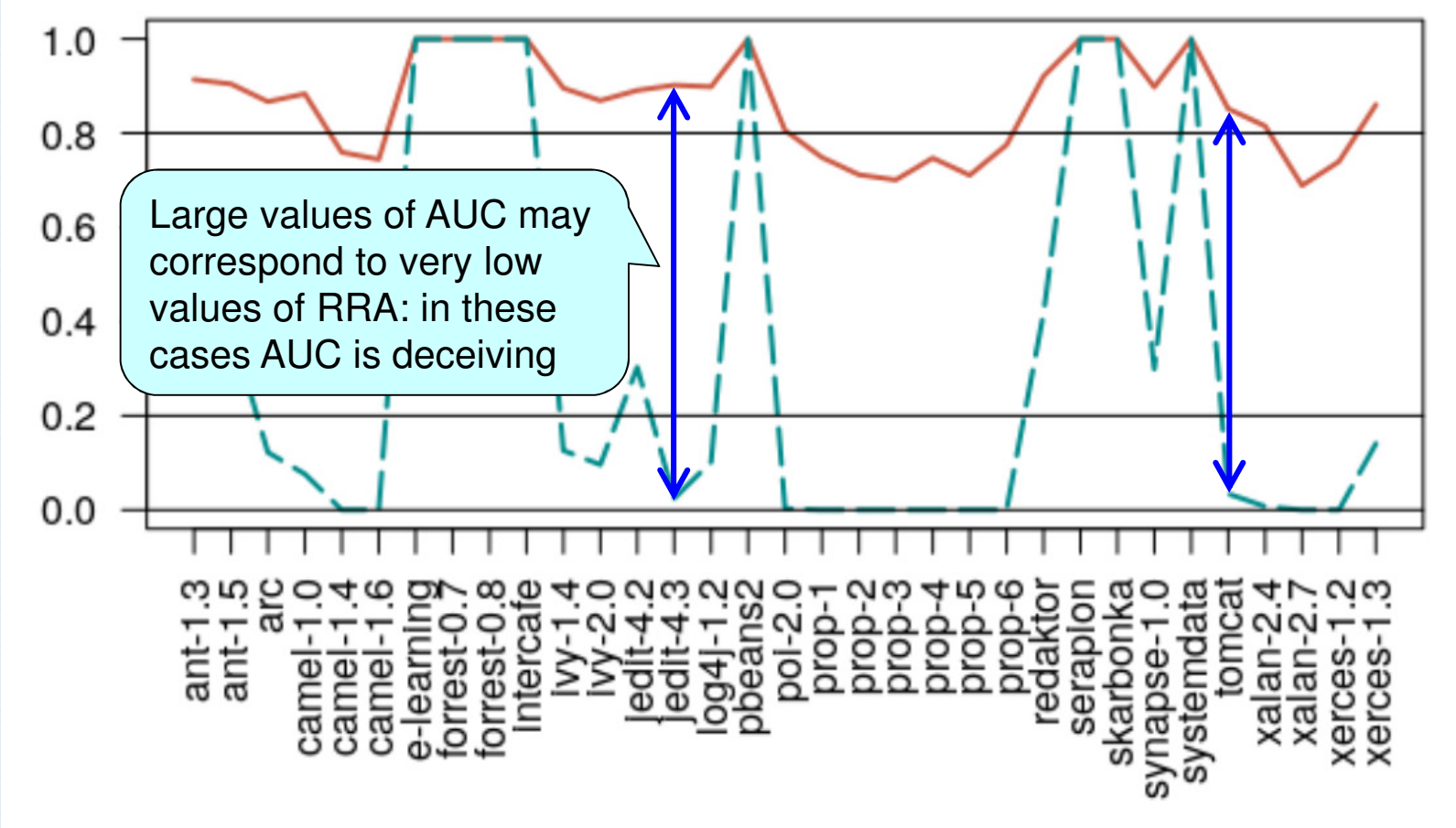
Example

- A classifier of defectiveness for the classes of the Ivy 2.0 project
- In the literature it is often reported that the best classifier is the one that is closest to point (0,1), or the one identified by the highest point on the tangent that is parallel to the diagonal.
- This example shows that following these indications you get a classifier whose fallout is worse than the average random classifications' fallout.



Example

- AUC (red continuous line) and RRA(φ = 0.4) (blue dashed line) for projects with very low (AP/n ≤ 0.2) or very high (AP/n ≥ 0.8) defectiveness rate.

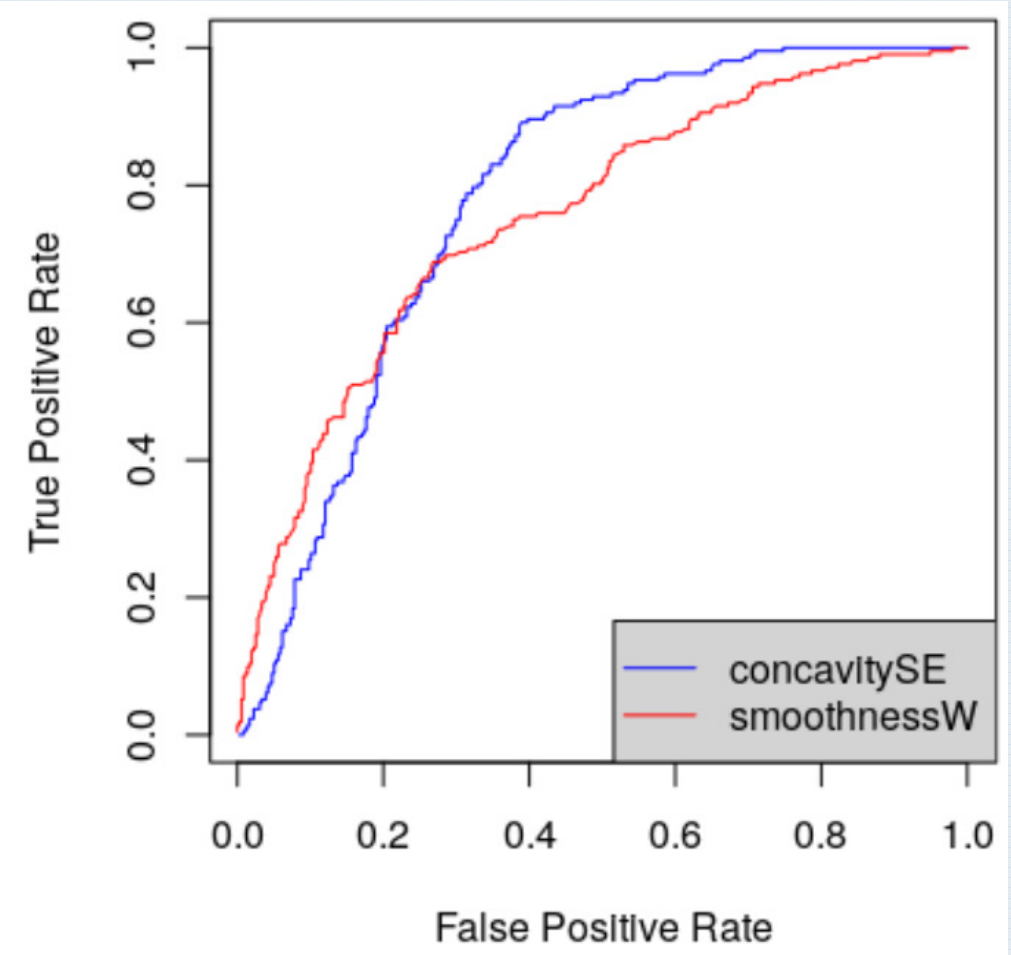


Example 2: breast cancer classification

- Wisconsin Breast Cancer Data
- Data
  - Malignant tumour (yes/no)
  - Size and shape measure (worst value, mean, …)

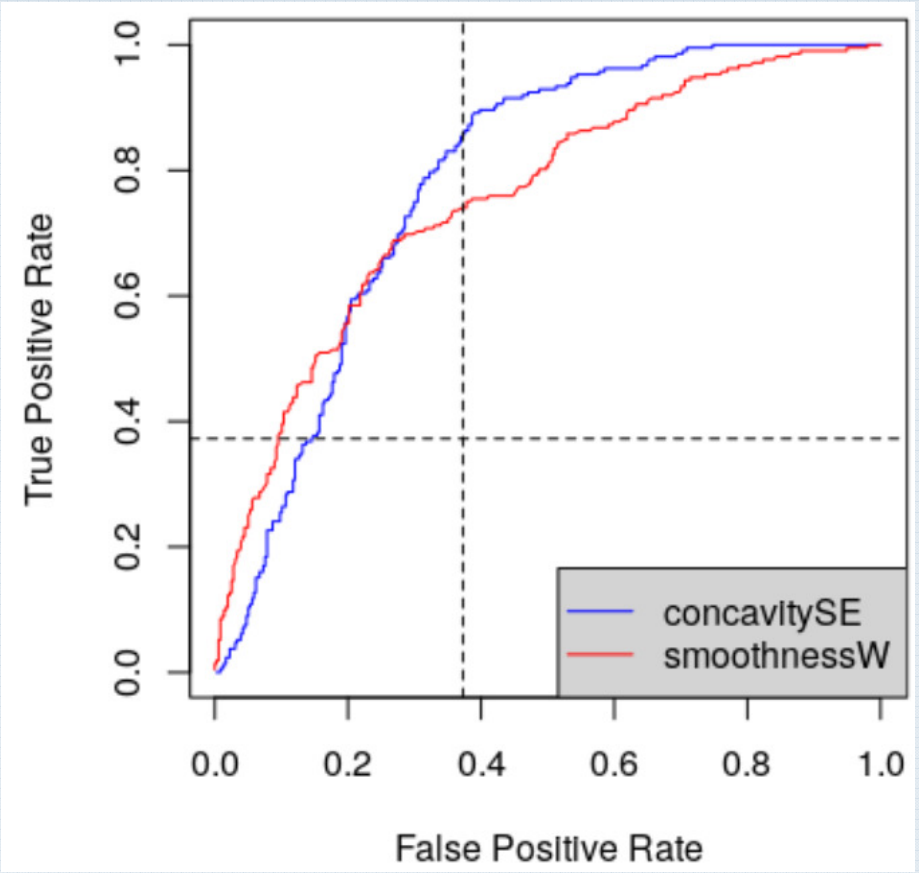
Example

- Is the classifier based on ConcavitySE better than the classifier based on SmoothnessW?



Example

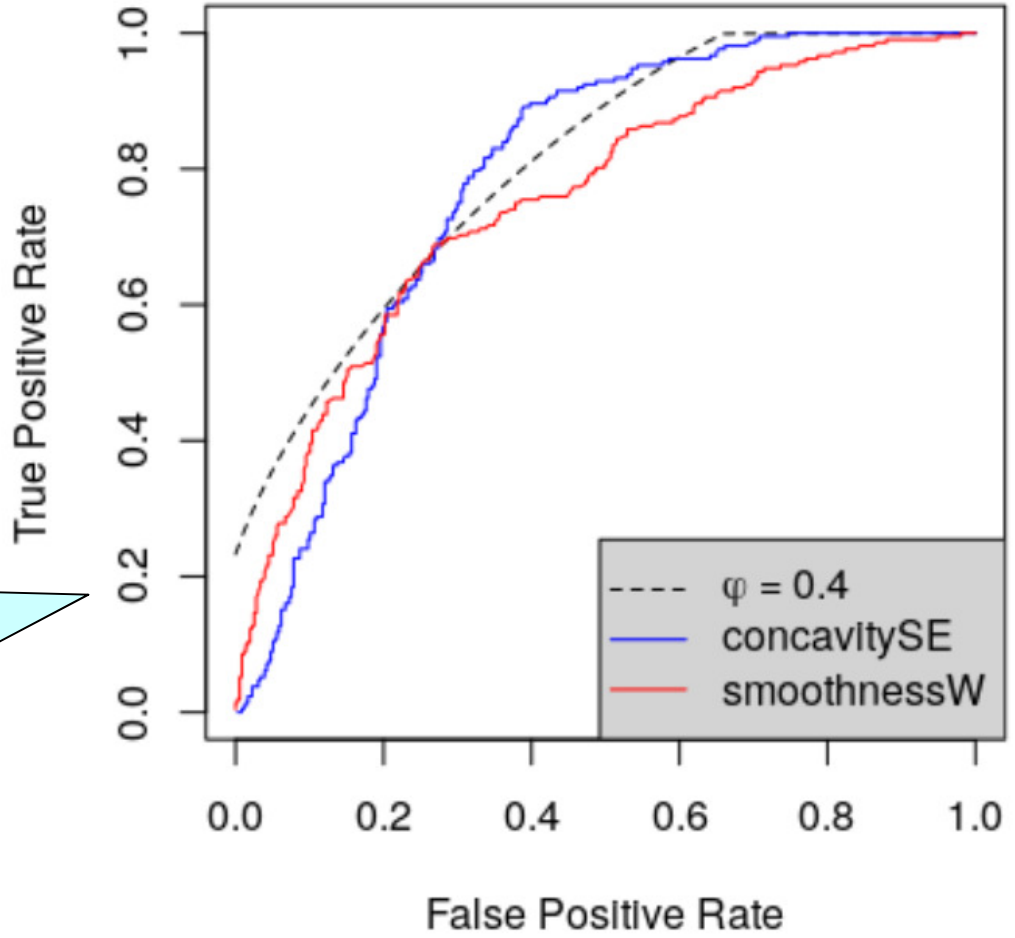
- ConcavitySE has RRA(TPR, FPR) = 0.2719
- SmoothnessW having RRA(TPR, FPR) = 0.272
- So, do the two models provide practically equivalent performance?



Example

- Let us compute RRA(φ=0.4):
  - For ConcavitySE RRA(φ=0.4)=0.26
  - For SmoothnessW RRA(φ=0.4)=0.008.

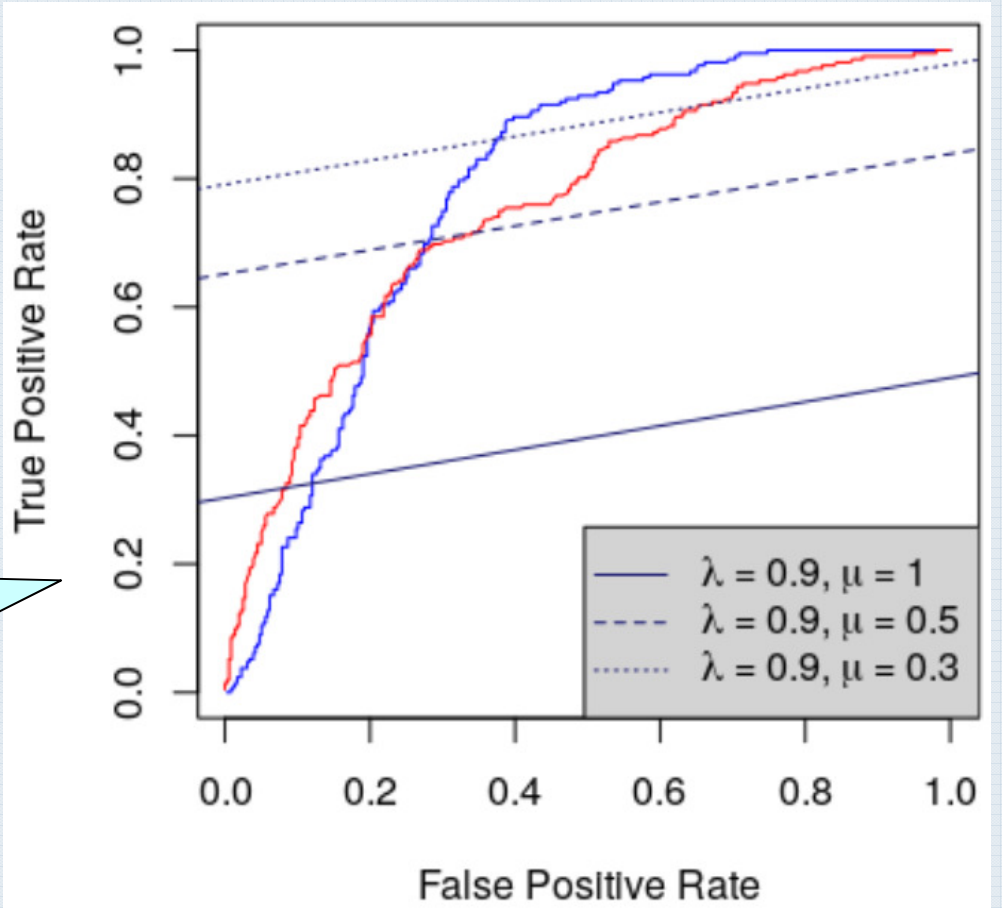
The classifier based on ConcavitySE appears definitely preferable.



Example

- Let us evaluate the performance based on cost.
- False negatives are much more critical than false positives, hence λ=0.9
  - For ConcavitySE
    RRA(λ=0.9, μ=0.3)=0.26
  - For SmoothnessW
    RRA(λ=0.9, μ=0.3)=0.07.

The classifier based on ConcavitySE appears definitely preferable.



Conclusions

- The area under a ROC curve (AUC) is often used to evaluate classifiers' performance.
- However, AUC takes into account large regions of the ROC space where classifications are worse than the average random classification, when performance is evaluated via commonly used indicators (like recall, F-measure, φ, etc.)
- In this tutorial we saw that sounder performance evaluation can be achieved by applying two fundamental concepts:
  - only models that outperform reference ones should be considered
    - Better than random estimation
    - Acceptable φ
    - etc.
  - any combination of performance metrics (including cost) can be used

Thanks for your attention!

QUESTIONS?