ROC Analysis

This Chapter is based on ***click here***.

Bernd Bischl

Department of Statistics – LMU Munich

Evaluation of Binary Classifiers

- Consider a binary classifier $C(x)$, e.g. for cancer prediction (with true label y).
  - for $C(x) = 1 = \hat{y}$, we predict cancer
  - for $C(x) = 0 = \hat{y}$, we predict no cancer
- One possible evaluation measure is the *misclassification error* or *error rate* (i.e. the proportion of patients for which $\hat{y} \neq y$).
- Example: If 10 out of 1000 patients are misclassified, the *error rate* is 1%.
- In general, lower *error rates* are better.

Evaluation of Binary Classifiers

- Using the *error rate* for imbalanced true labels is not suggested.
- Example: Assume that only 0.5% of 1000 patients have cancer.
  - Always returning $C(x) = 0 = \hat{y}$ gives an *error rate* of 0.5%, which sounds good.
  - However, we would never predict cancer, which is bad.

$\Rightarrow$ We need also different evaluation metrics and should not only trust the *error rate*.

Confusion Matrix

The confusion matrix is a 2×2 contingency table of predictions $\hat{y}$ and true labels y. Several evaluation metrics can be derived from a confusion matrix:

Diagnostic Testing Measures

		Actual Class y		
		Positive	Negative	
$\hat{y}$ **Test outcome**	Test outcome positive	**True positive** (TP)	**False positive** (FP, Type I error)	Precision = $\frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FP}}$
	Test outcome negative	**False negative** (FN, Type II error)	**True negative** (TN)	Negative predictive value = $\frac{\#\mathrm{TN}}{\#\mathrm{FN} + \#\mathrm{TN}}$
		Sensitivity = $\frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FN}}$	Specificity = $\frac{\#\mathrm{TN}}{\#\mathrm{FP} + \#\mathrm{TN}}$	Accuracy = $\frac{\#\mathrm{TP} + \#\mathrm{TN}}{\#\mathrm{TOTAL}}$

Confusion Matrix

Terminology:

- **True positive (TP):**
  We predicted “1” and the true class is “1”.
- **True negative (TN):**
  We predicted “0” and the true class is “0”.
- **False positive (FP):**
  We predicted “1” and the true class is “0” (type I error).
- **False negative (FN):**
  We predicted “0” and the true class is “1” (type II error).
- **Positive (pos):**
  Fraction of true class labels with “1”.
- **Negative (neg):**
  Fraction of true class labels with “0”.

Confusion Matrix

The following measures can be obtained from the confusion matrix:

- True positive rate (also known as sensitivity or recall)
  - Fraction of positive observations correctly classified
  - $\text{tpr} = \dfrac{\text{TP}}{\text{TP} + \text{FN}}$
- False positive rate (also known as fall-out)
  - Fraction of negative observations incorrectly classified
  - $\text{fpr} = \dfrac{\text{FP}}{\text{FP} + \text{TN}} = 1 - \text{Specificity}$
- Accuracy
- Misclassification error (or error rate)
- Positive predictive value (or precision)
- Negative predictive value
- . . .

Accuracy and Misclassification Error

In practice, these are the most widely used metrics

- Accuracy: $\text{acc} = \dfrac{TP + TN}{N}$
  - fraction of correctly classified observations
- Error rate: $\text{error} = \dfrac{FN + FP}{N} = 1 - \text{acc}$
  - Fraction of misclassified observations

Precision (P) or Positive Predictive Value (PPV)

$$P = \frac{\#\,\text{TP}}{\#\,\text{predicted positives}} = \frac{\#\,\text{TP}}{\#\,\text{TP} + \#\,\text{FP}}$$

or

$$P = \frac{pos \cdot tpr}{pos \cdot tpr + neg \cdot fpr} = \frac{tpr}{tpr + \frac{neg}{pos} \cdot fpr}$$

- Interpretation: For all observations that we predicted $\hat{y} = 1$, what fraction actually has $y = 1$?
- Higher *precision* is better.

Recall (R)

$$R = \frac{\#\,\text{TP}}{\#\,\text{actual positives}} = \frac{\#\,\text{TP}}{\#\,\text{TP} + \#\,\text{FN}} = \text{tpr}$$

- Interpretation: For all observations that actually have $y = 1$, what fraction did we correctly detect as $\hat{y} = 1$?
- Higher *recall* is better.
- For a classifier that always returns zero (i.e. $\hat{y} = 0$), the *recall* would be zero.

F-Measure

It is difficult to achieve a high *precision* and high *recall* simultaneously. A trade-off offers the F_1-measure, which is the harmonic mean of precision P and recall R:

$$F_1 = 2\frac{P\,R}{P + R} = \frac{2\text{tpr}}{\text{tpr} + \frac{neg}{pos} \cdot \text{fpr} + 1}$$

Example

		Patients with bowel cancer (as confirmed on endoscopy)		
		Positive	Negative	
Fecal Occult Blood Screen Test Outcome	Test Outcome Positive	True Positive (TP) = 20	False Positive (FP) = 180	Positive predictive value = TP / (TP + FP) = 20 / (20 + 180) = **10%**
	Test Outcome Negative	False Negative (FN) = 10	True Negative (TN) = 1820	Negative predictive value = TN / (FN + TN) = 1820 / (10 + 1820) ≈ **99.5%**
		Sensitivity = TP / (TP + FN) = 20 / (20 + 10) ≈ **67%**	Specificity = TN / (FP + TN) = 1820 / (180 + 1820) = **91%**	

ROC Analysis

The **R**eceiver **O**perating **C**haracteristic (ROC) curve is created by plotting the *tpr* vs. *fpr*, i.e. the

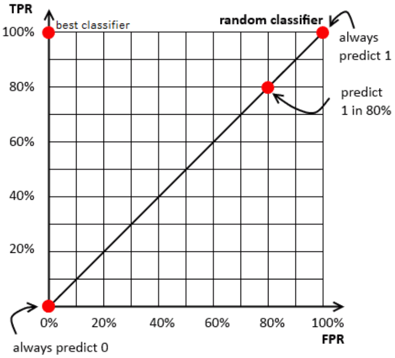
- True positive rate, tpr = $\frac{\text{TP}}{\text{TP} + \text{FN}}$
- False positive rate fpr = 1 $-$ specificity = $\frac{\text{FP}}{\text{FP} + \text{TN}}$

Properties:

- ROC curves are insensitive to class distribution.
- If the proportion of positive to negative instances changes, the ROC curve will not change.
- ROC space is 2 dimensional, i.e. X : fpr, Y : tpr.

ROC Space Baseline

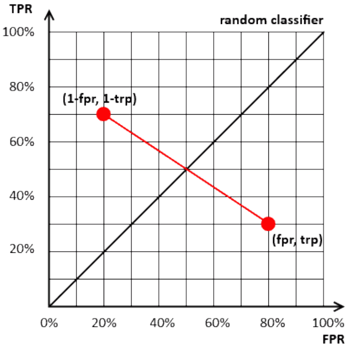
- The best classifier lies on the top-left corner.
- Example: 3 classifiers that lie on the baseline, i.e. a classifier that
  - always predicts 0 (0% change to predict 1),
  - predicts 1 in 80% cases and
  - always predict 1 (in 100% cases).



ROC Space Baseline

In practice, we can never obtain a classifier below this line. Example:

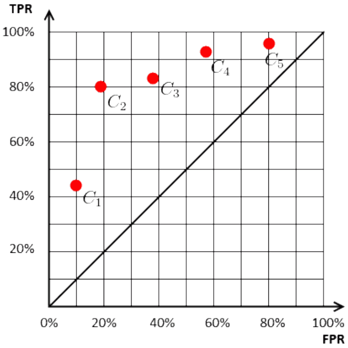
- A classifier C_1 below the line with fpr = 80%, and tpr = 30%
- We can make it better than random by inverting its prediction: $C_2(x)$: if $C_1(x) = 1$, return 0; if $C_1(x) = 0$, return 1
- Position of C_2 is then $(1 - \text{fpr}, 1 - \text{tpr}) = (20\%, 70\%)$



ROC Convex Hull

Suppose we have 5 classifiers $C_1, C_2, ..., C_5$

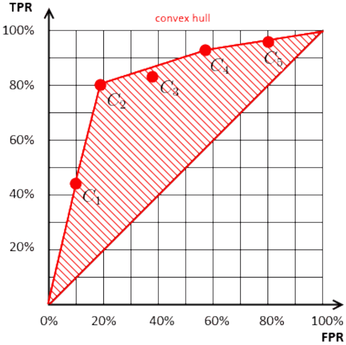
- We calculate fpr and tpr for each and plot them on one plot.
- Each classifier is a single point in the ROC space.



ROC Convex Hull

We can then try to find classifiers that achieve the best fpr and tpr.

- By the dominance principle, we have the following Pareto frontier (the "ROC convex hull").
- Classifiers below this hull are always suboptimal, e.g. C_3.



ISO Accuracy Lines

There is a simple relationship between accuracy and fpr, tpr.

- Let N be the number of observations,
- NEG and POS the number of negative and positive observations,
- neg and pos the fraction of negative and positive observations, respectively.

$$\text{acc} = \text{tpr} \cdot \text{pos} + \text{neg} - \text{neg} \cdot \text{fpr}$$

- $\text{acc} = \frac{\text{TP} + \text{TN}}{N} = \frac{\text{TP}}{N} + \frac{\text{TN}}{N} = \frac{\text{TP}}{\text{POS}} \cdot \frac{\text{POS}}{N} + \frac{\text{NEG} - \text{FP}}{N} = \frac{\text{TP}}{\text{POS}} \cdot \frac{\text{POS}}{N} + \frac{\text{NEG}}{N} - \frac{\text{FP}}{\text{NEG}} \cdot \frac{\text{NEG}}{N} = \text{tpr} \cdot \text{pos} + \text{neg} - \text{fpr} \cdot \text{neg}$

ISO Accuracy Lines

We can rewrite this and get

$$\text{tpr} = \frac{\text{acc} - \text{neg}}{\text{pos}} + \frac{\text{neg}}{\text{pos}} \cdot \text{fpr} \;\Leftrightarrow\; y = ax + b$$

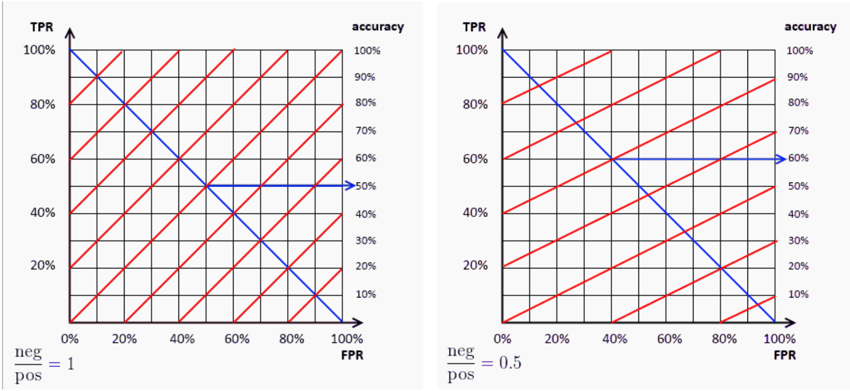
$$\text{with } y = \text{tpr},\; x = \text{fpr},\; a = \frac{\text{neg}}{\text{pos}},\; x = \frac{\text{neg}}{\text{pos}},\; b = \frac{\text{acc} - \text{neg}}{\text{pos}}$$

Properties:

- The ratio $a = \frac{\text{neg}}{\text{pos}}$ is the slope of the line (changing this ratio yields many different slopes).
- Changing the accuracy yields many parallel lines with the same slope because *acc* is included in the intercept *b*.
- “Higher” lines are better w.r.t. *acc*.

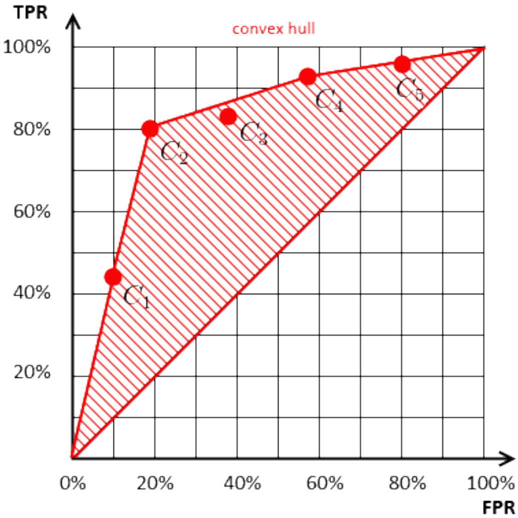
ISO Accuracy Lines

To calculate the corresponding accuracy, we have to find the intersection point of the accuracy line (red) the descending diagonal (blue).



ISO Accuracy Lines vs Convex Hull

Recall the convex hull of the ROC plot:



ISO Accuracy Lines vs Convex Hull

Each line segment of the ROC convex hull is an ISO accuracy line for a particular class distribution (slope) and accuracy. All classifiers on such a line achieve the same accuracy for this distribution:

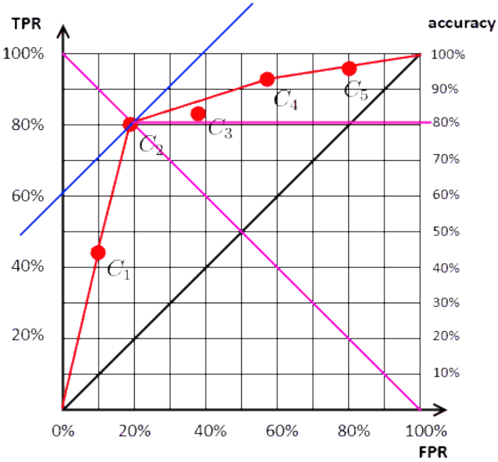
- $\text{neg}/\text{pos} > 1$
  - Distribution with more negative observations.
  - The slope is steep.
  - Classifier on the left is better.
- $\text{neg}/\text{pos} < 1$
  - Distribution with more positive observations.
  - The slope is flatter.
  - Classifier on the right is better.

Each classifier on the convex hull is optimal w.r.t. accuracy and for a specific distribution.

Selecting the Optimal Classifier

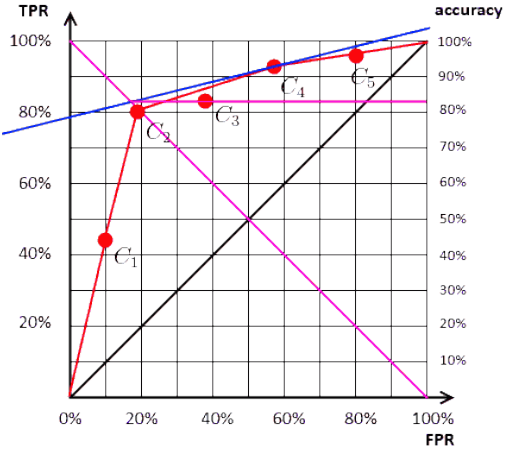
1. Compute the ratio (slope) neg/pos.
2. Find the classifier that achieves the highest accuracy for this ratio.
3. Fix the ratio and keep increasing the accuracy until the end of the hull.

Selecting the Optimal Classifier - Example



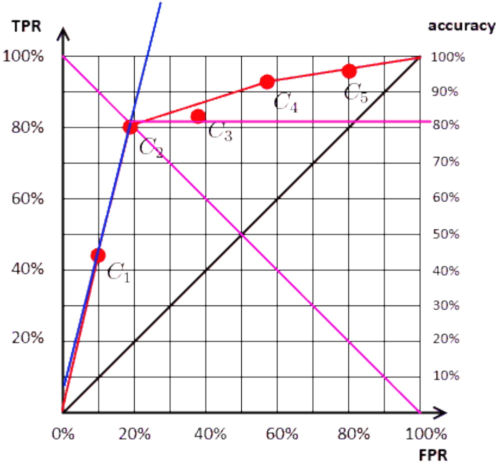
- Distribution: $neg/pos = 1/1$, best classifier: C_2, accuracy $\approx 81\%$

Selecting the Optimal Classifier - Example



- Distribution: $neg/pos = 1/4$, best classifier: C_4, accuracy $\approx 83\%$

Selecting the Optimal Classifier - Example



- Distribution: $neg/pos = 4/1$, best classifier: C_2, accuracy $\approx 81\%$

Scoring Classifiers

A scoring classifier (or ranker) is an algorithm that outputs the scores (e.g. a probabilities) for each class instead of one single label.
Do binary classification with a ranker F:

- F outputs a single number
- Set some threshold θ to transform the ranker into a classifier, e.g. as in logistic regression
  - Predict $\hat{y} = 1$ (positive class) if $F(x) > \theta$ else predict $\hat{y} = 0$
- How to set a threshold θ?
  - Use crossvalidation for finding the best value for θ.
  - Draw ROC curves, producing a point in the ROC space for **each possible threshold**.

ROC for Scoring Classifiers

Naive Method:
Given a ranker F and a dataset with N training observations:

- Consider all possible thresholds ($N - 1$ for N observations).
- For each threshold: Calculate fpr and tpr, and draw this point on the ROC space.
- Select the best threshold using the ROC analysis (for the ratio neg$/$pos).

Practical Method:

- Rank test observations on decreasing score.
- Start in $(0, 0)$, for each observation x (in the decreasing order).
  - If x is positive, move $1/$pos up
  - If x is negative, move $1/$neg right

Example Naive Method

Given:

- 20 observations

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C	N	N	N	N	N	N	N	N	N	N	N	N	P	P	P	P	P	P	P	P
Score	.18	.24	.32	.33	.4	.53	.58	.59	.6	.7	.75	.85	.52	.72	.73	.79	.82	.88	.9	.92

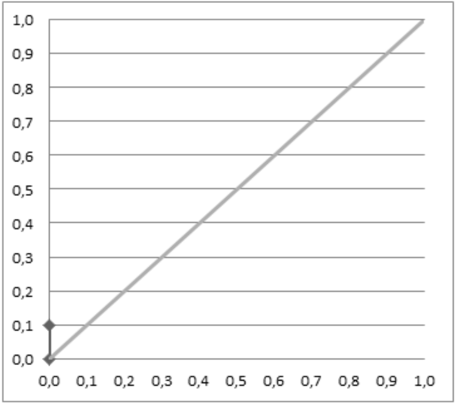
- C is the actual class of the training observations.
- $\text{neg}/\text{pos} = 1$, i.e. $1/\text{pos} = 1/\text{neg} = 0.1$

Best threshold:

- We know that the slope of the accuracy is 1.
- The best classifier for this slope is the 6th observation.

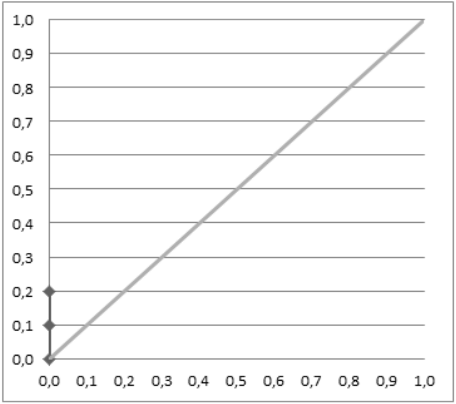
Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



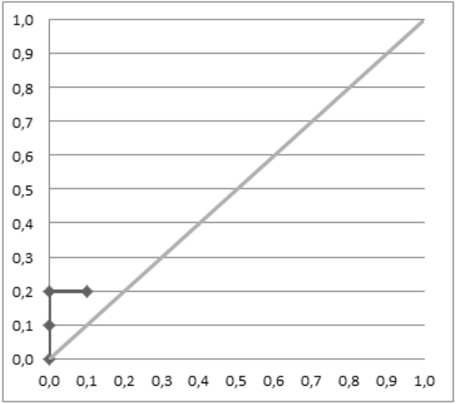
Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



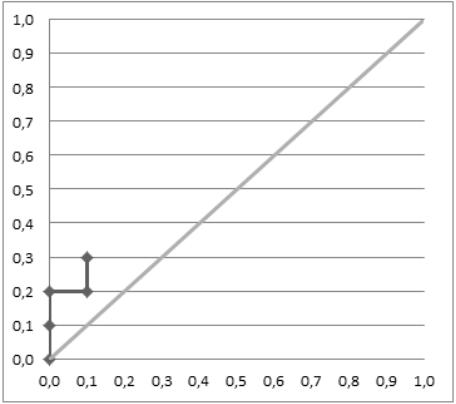
Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



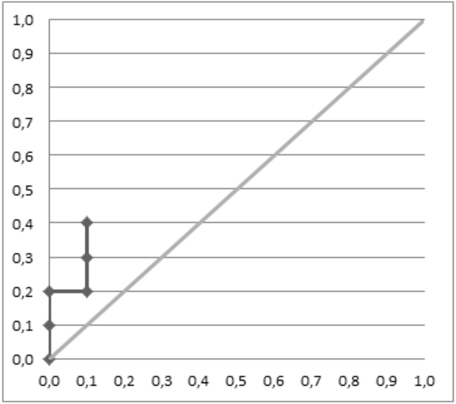
Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



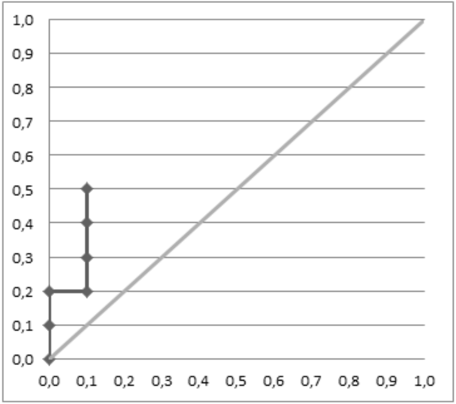
Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



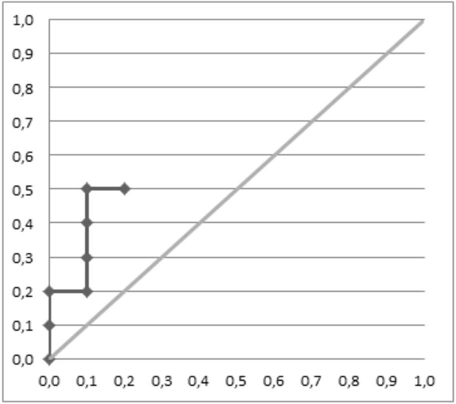
Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



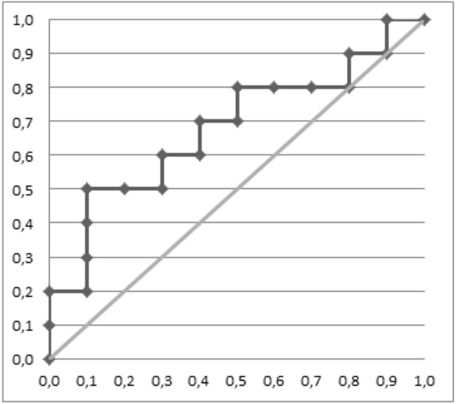
Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1

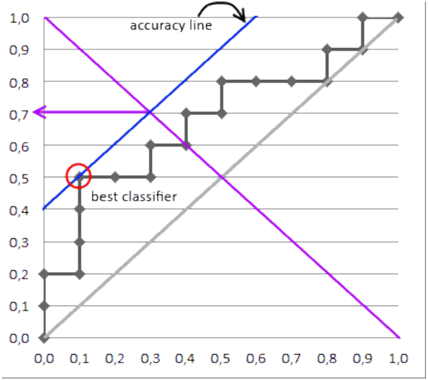


Example Naive Method

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



Example Naive Method



- Score of the best (6th) classifier is used as the threshold θ.
- Predict positive class for $\theta \geqslant 0.54$ ($\Rightarrow$ accuracy = 0.7).

Example Practical Method

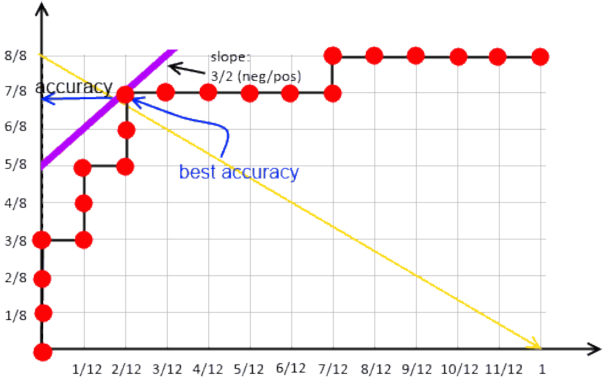
Given: 20 training observations, 12 negative and 8 positive

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C	N	N	N	N	N	N	N	N	N	N	N	N	P	P	P	P	P	P	P	P
Score	.18	.24	.32	.33	.4	.53	.58	.59	.6	.7	.75	.85	.52	.72	.73	.79	.82	.88	.9	.92

$\Rightarrow$ sort by score and draw the curves:

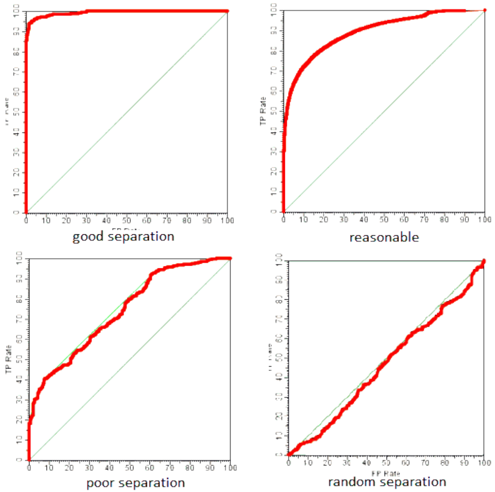
#	20	19	18	12	17	16	11	15	14	10	9	8	7	6	13	5	4	3	2	1
C	P	P	P	N	P	P	N	P	P	N	N	N	N	N	P	N	N	N	N	N
Score	.92	.9	.88	.85	.82	.79	.75	.73	.72	.7	.6	.59	.58	.53	.52	.4	.33	.32	.24	.18

Example Practical Method



- Best accuracy achieved with observation # 18.
- Setting $\theta = 0.88 \Rightarrow$ accuracy of $15/20 \hat{=} 75\%$.

Other ROC Curve Examples

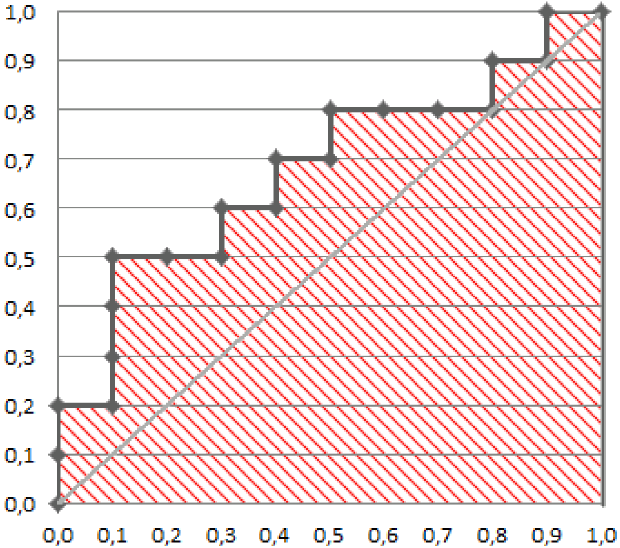


AUC: Area Under ROC Curve

The area under the ROC curve (AUC $\in [0, 1]$) is

- a measure for evaluating the performance of a classifier:
  - AUC = 1: Perfect classifier, for which all positives are ranked higher than all negatives
  - AUC = 0.5: Randomly ordered
  - AUC = 0: All negatives are ranked higher than all positives
  - Interpretation of AUC: Probability that a classifier C ranks a randomly drawn positive observation "+" higher than a randomly drawn negative observation "$-$".
- related to the Mann-Whitney-U test, which
  - estimates the probability that randomly chosen positives are ranked higher than randomly chosen negatives.
  - uses test statistic $U = \dfrac{AUC}{POS \cdot NEG}$.
- related to the Gini coefficient = $2 \cdot AUC - 1$ (area above diag.)

AUC: Area Under ROC Curve



Multiclass AUC

- Consider multiclass classification, where a classifier predicts the probability p_k of belonging to class k for each class.
- Hand and Till (2001) proposed to average the AUC of pairwise comparisons (1 vs. 1) of a multiclass classifier.
  - estimate $AUC(i, j)$ for each pair of class i and j
  - $AUC(i, j)$ is the probability that a randomly drawn member of class i has a lower probability of belonging to class j than a randomly drawn member of class j.
  - for K classes, we have $\binom{K}{2} = \frac{K(K-1)}{2}$ values of $AUC(i, j)$ that are then averaged to compute the Multiclass AUC.

Calibration and Discrimination

We consider data with a binary outcome y.

- **Calibration:** When the predicted probabilities closely agree with the observed outcome (for any reasonable grouping).
  - **Calibration in the large** is a property of the *full sample*. It compares the observed probability in the full sample (e.g. proportion of observations for which $y = 1$) with the average predicted probability in the full sample.
  - **Calibration in the small** is a property of *subsets* of the sample. It compares the observed probability in each subset with the average predicted probability in that subset.
- **Discrimination:** Ability to perfectly separate the population into $y = 0$ and $y = 1$. Measures of discrimination are, for example, AUC, sensitivity, specificity.

Calibration and Discrimination

- Calibration is desirable, but not sufficient. Example:

Obs. Nr.	truth	Pred Rule 1	Pred Rule 2
1	1	1	0
2	1	1	0
3	0	0	0
4	0	0	0
5	0	0	1
6	0	0	1
Avg Prob	30%	30%	30%

- Both prediction rules have identical calibration in the large (30%), however, rule 1 is better than rule 2.

Calibration and Discrimination

A well discriminating classifier can have a bad calibration, e.g.

```r
y = c(1,1,0,0,0,0)
pred = c(0.95, 0.95, 0.5, 0.5, 0.5, 0.5)
# perfect discrimination w.r.t. AUC
mlr::measureAUC(pred, y, negative = 0, positive = 1)
```

```
## [1] 1
```

```r
# bad calibration w.r.t. calibration-in-the-large
c(mean(y), mean(pred))
```

```
## [1] 0.3333333 0.6500000
```

ROC Analysis in R

- `generateThreshVsPerfData` calculates one or several performance measures for a sequence of decision thresholds from 0 to 1.
- It provides S3 methods for objects of class `Prediction`, `ResampleResult` and `BenchmarkResult` (resulting from `predict.WrappedModel`, `resample` or `benchmark`).
- `plotROCCurves` plots the result of `generateThreshVsPerfData` using `ggplot2`.
- More infos `http://mlr-org.github.io/mlr-tutorial/release/html/roc_analysis/index.html`

Example 1: Single predictions

```r
library(mlr)
set.seed(1)
# get train and test idices
n = getTaskSize(sonar.task)
train.set = sample(n, size = round(2/3 * n))
test.set = setdiff(seq_len(n), train.set)

# fit and predict
lrn = makeLearner("classif.lda", predict.type = "prob")
mod = train(lrn, sonar.task, subset = train.set)
pred = predict(mod, task = sonar.task, subset = test.set)
```

Example 1: Single predictions

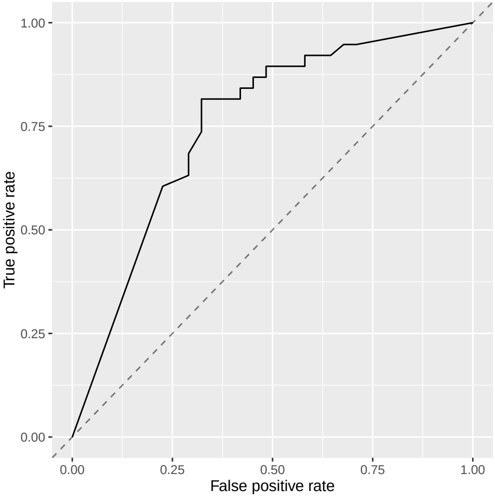
We calculate fpr, tpr and compute error rates:

```
df = generateThreshVsPerfData(pred, measures = list(fpr, tpr, mmce))
```

- `generateThreshVsPerfData` returns an object of class `ThreshVsPerfData`, which contains the performance values in the `$data` slot.
- By default, `plotROCCurves` plots the performance values of the first two measures passed to `generateThreshVsPerfData`.
- The first is shown on the x-axis, the second on the y-axis.

Example 1: Single predictions

```
df = generateThreshVsPerfData(pred, measures = list(fpr, tpr, mmce))
plotROCCurves(df)
```



Example 1: Single predictions

The corresponding area under curve auc can be calculated by

```r
performance(pred, auc)
```

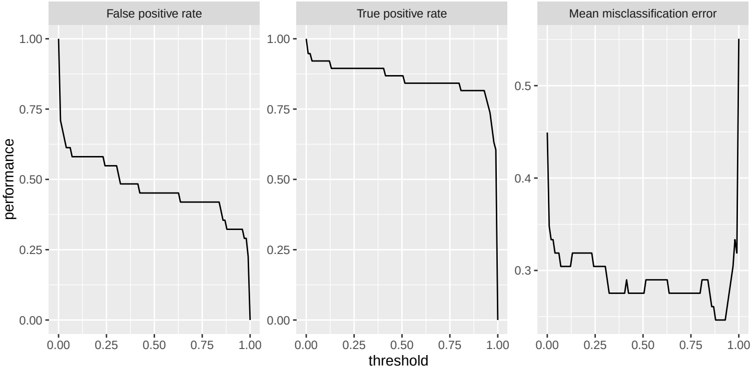
```
##       auc
## 0.7860781
```

`plotROCCurves` always requires a pair of performance measures that are plotted against each other.

Example 1: Single predictions

If you want to plot individual measures vs. the decision threshold, use

```
plotThreshVsPerf(df)
```



Example 2: Benchmark Experiment

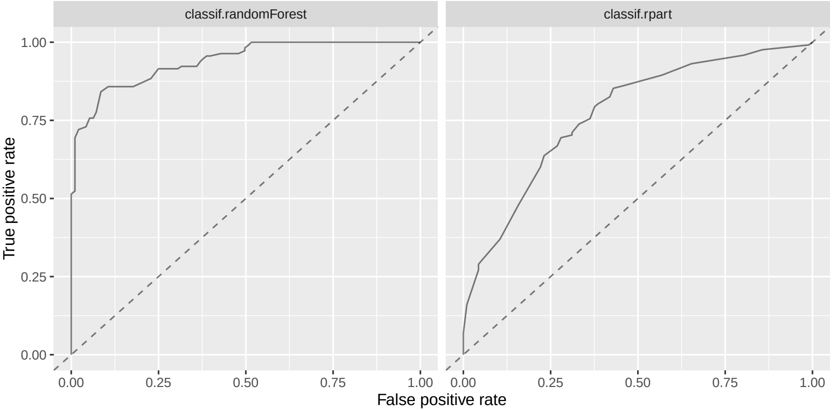
```r
lrn1 = makeLearner("classif.randomForest", predict.type = "prob")
lrn2 = makeLearner("classif.rpart", predict.type = "prob")

cv5 = makeResampleDesc("CV", iters = 5)

bmr = benchmark(learners = list(lrn1, lrn2), tasks = sonar.task,
  resampling = cv5, measures = list(auc, mmce), show.info = FALSE)
bmr
```

```
##         task.id           learner.id auc.test.mean mmce.test.mean
## 1 Sonar-example classif.randomForest     0.9390542      0.1786295
## 2 Sonar-example        classif.rpart     0.7589779      0.2932636
```

Calling `generateThreshVsPerfData` and `plotROCCurves` on the `BenchmarkResult` produces a plot with ROC curves for all learners in the experiment.

Example 2: Benchmark Experiment

```
df = generateThreshVsPerfData(bmr, measures = list(fpr, tpr, mmce))
plotROCCurves(df)
```



Example 2: Benchmark Experiment

```
plotThreshVsPerf(df)
```