Classification and Evaluating Classifiers

Mladen Kolar (mkolar@chicagobooth.edu)

Reminder: Supervised learning

Training experience: a set of labeled examples (samples, observations) of the form $(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_n, y_n)$ where $x_i = (1, x_{i1}, \ldots, x_{ip})$ are vectors of input variables (covariates, predictors) and y is the output

What to learn: A function f which maps input X into output Y

Goal: minimize the error (loss) function

What do we use the loss function for?

We use the loss function to estimate the parameters of a model.

We use loss measures to assess our out-of-sample performance

- e.g., when doing cross-validation

How should you access if the model is any good?

Suppose that a consulting company produced a machine learning model for your churn problem.

Overfitting aside, how would you go about measuring if the produced model is any good?

Machine learners and other stakeholders must consider carefully what they would like to achieve by building a model.

What should be the loss function for classification?

Loss should be problem specific.

However, it is often useful to have generic loss functions.

Often need surrogate loss as the ultimate goal if not perfectly measurable.

We will discuss some common issues and themes in evaluation, and how to deal with them.

Misclassification rate or accuracy

For classification, an obvious loss is the **misclassification rate** or **accuracy**.

If $Y, \hat{Y} \in \{1, 2, \dots, C\}$ we can let

$$L(Y, \hat{Y}) = \begin{cases} 0, & Y = \hat{Y} \\ 1, & Y \neq \hat{Y} \end{cases}$$

If you guess right, you lose nothing, if you are wrong, you lose 1.

Then

$$MR = \frac{1}{m} \sum_{i=1}^{m} L(Y_i, \hat{Y}_i)$$

is the fraction misclassified. Accuracy is simply $1 - MR$.

Problems with misclassification rate

It is not easy to minimize — leads to a hard optimization problem.

- This is a technical point, but an important one. We will investigate alternative surrogates next.

Reduces performance of a classifier to a single number, which is a simplistic summary.

- Sometimes it is important to understand types of correct and incorrect decisions made by a classifier. For that we use the confusion matrix.

Does not work well with unbalanced classes.

- Often a classifier will put everything in a majority class (this is surprising).

The Confusion Matrix

We consider a two-class classification problem.

Confusion matrix separates out the decisions made by the classier, making it explicit how one class is being confused by another.

- Columns are labeled by actual classes.
- Rows are labeled by predicted classes.

	positive	**n**egative
Y	True positives	False positives
N	False negatives	True negatives

How can we obtain misclassification rate of a classifier?
How can we obtain accuracy of a classifier?

Example: Tabloid data

Y is 1 if a customer responds to a promotion (mailed a “tabloid’ ’) and 0 otherwise.

4 x’s from the database (selected from many other similar ones).

- nTab: number of past orders.
- moCbook: months since last order.
- iRecMer1 : 1/ months since last order in merchandise category 1.
- llDol: log of the dollar value of past purchases

10,000 in train; 5,000 in test.

Note: in the train, only 2.58% respond.

Surrogate Losses

For computational reasons, it is common to minimize a surrogate loss to misclassification rate during training.

Commonly used losses in machine learning are:

- deviance loss (can be used for any probabilistic classifier)
- hinge loss (support vector machines)
- exponential loss (AdaBoost algorithm)

Keep in mind that all these are generic losses, which are not tailored for a specific problem at hand.

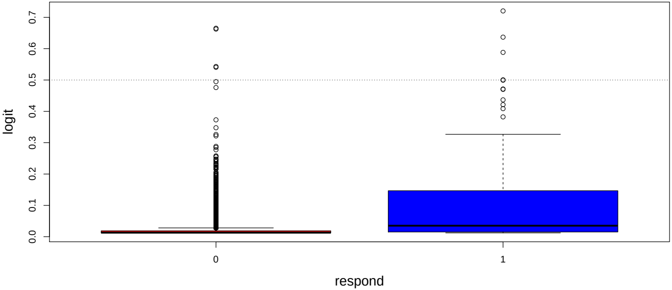
The hope is that a classifier trained to minimize these generic losses will also perform well in terms of misclassification on out-of-sample data, as well as on a problem specific metric.

Another problem with misclassification is that, in many cases, the model with best misclassification rate predicts that all instances belong to the majority class.

- not useful for making decisions in business applications

Recall that in our target marketing example, the probability of a response is almost always less than 0.5. If you just predict “they won’t respond” you do pretty good in terms of misclassification, but that is not helpful.

Here is the plot of $\hat{p}$ (from logit) and y=respond for the tabloid data (validation set).



While the model works, it is pretty useless to predict a response if $\hat{p} > 0.5$!

Only 112 out of 5,000 actually responded so if we just say no-one responds the MR is 2%.

Deviance loss

Deviance loss measures the quality of the model by looking at $\hat{P}(Y = y \mid X = x)$ produced by the model.

Suppose we have an (x, y) pair where x is a vector of predictors and y is the corresponding outcome. Our model gives us $\hat{P}(Y = y \mid X = x)$.

- If $\hat{P}(Y = y \mid X = x)$ is high, then the observed thing is likely under our model.
- If $\hat{P}(Y = y \mid X = x)$ is small, then the observed thing is un-likely under our model.

If the model says what happened is likely, that is good!

The **deviance** measure of how **bad** things are (for one observation (x, y)) is

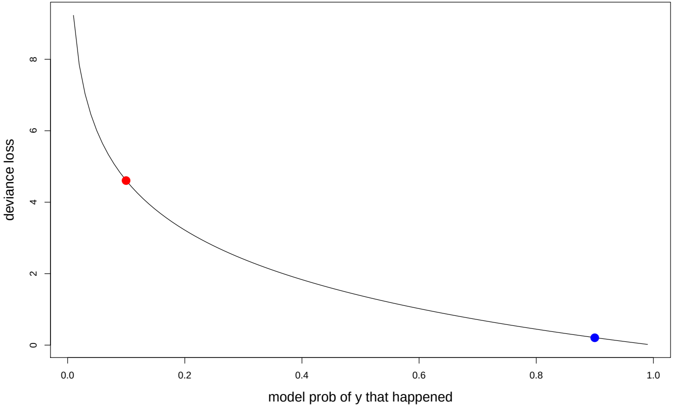
$$L(\hat{P}, x, y) = -2 \log(\hat{P}(Y = y \mid X = x))$$

The total loss for a dataset $\mathcal{D}$ (train or test) is

$$L(\hat{P}) = \sum_{i \in \mathcal{D}} L(\hat{P}, x_i, y_i) = \sum_{i \in \mathcal{D}} -2 \log(P(Y = y_i \mid x_i))$$

The deviance is not terribly interpretable, but it gets used a fair amount.

If you say a certain y outcome is likely and then it happens, your loss is small.



If you say a certain y outcome is very unlikely and then it happens, your loss is big.

Problems with Unbalanaced Classes

Suppose there are two classifiers whose confusion matrices are:

Confusion matrix of classifier A

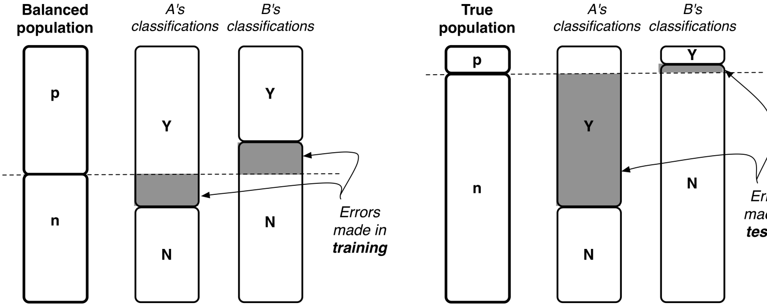
	respond	did not respond
Y	500	200
N	0	300

Confusion matrix of classifier B

	respond	did not respond
Y	300	0
N	200	500

They have the same accuracy. Which one would you prefer?

Note: It is common to train a classifier on a balanced training set.


Balanced population
A's classifications
B's classifications
p
n
Y
N
Y
N
Errors made in ***training***
True population
A's classifications
B's classifications
p
n
Y
N
Y
N
Errors made in ***testing***

Problems with Unequal Costs and Benefits

Another problem with misclassification rate is that it does not make distinction between false positive and false negative errors.

Typically very different kinds of errors with different costs because of consequences of different severity.

Whatever the problem, it is hard to imagine a decision maker being indifferent to types of errors she makes.

- Ideally we should have estimates of costs and benefits of each decision a classifier can make.

Decision Theory and Expected Utility

Expected values are useful in organizing thinking about data-analytic problems.

It allows us to decompose data-analytic thinking into:

- the structure of the problem (possible decision outcomes)
- the elements of the analysis that can be extracted from data (probabilities of outcomes)
- the elements of the analysis that need to be extracted from other sources

Let us go back to the target marketing example.

We would like to assign each consumer a class:

- *likely responder*, or
- *not likely responder*.

We saw that a "common sense" threshold of 50% for deciding whether someone will likely respond would likely lead to targeting very few people.

Still, if, given x, the probability of a response is big, then it should make sense to target the customer.

Alternatively, if our model suggest there is a very low chance they will respond, it may be a waste of money.

How can we use expected values here?

- Let $p_R(x)$ be the estimated probability of response of a consumer described by x as an input.
- Let v_R be the benefit of a customer responding.
- Let v_{NR} be the cost of a customer not-responding.

$$\text{Expected benefit of targeting } = p_R(x) \cdot v_R + [1 - p_R(x)] \cdot v_{NR}$$

If the expected benefit of targeting is positive, you should target.

Suppose that is costs 80 cents to mail the promotion (in our tabloid example).

Suppose that (on average) a customer spends \$40, if they respond.

If they respond, your benefit is \$39.20.

If they do not respond, your cost is \$0.80.

$$p_R(x) \cdot (39.20) + [1 - p_R(x)] \cdot (-0.8) > 0 \quad \Longrightarrow \quad p_R(x) > 0.02$$

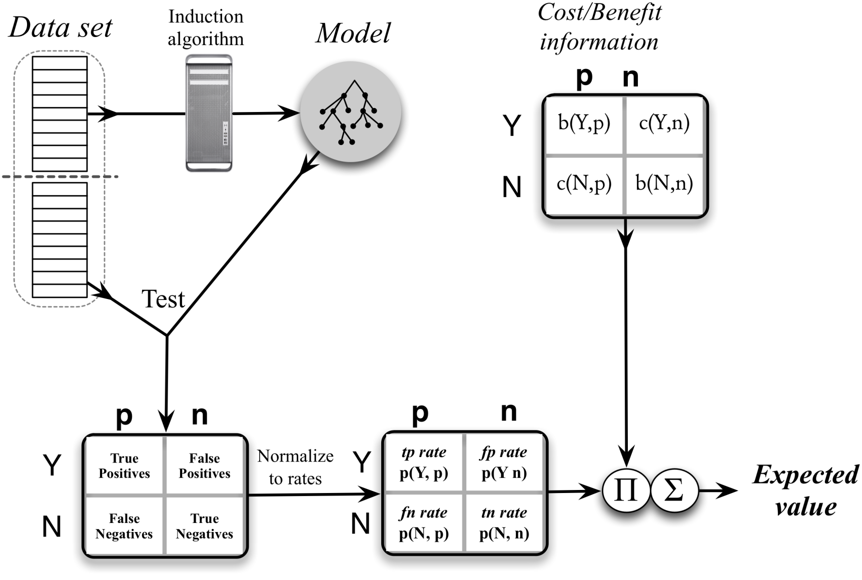
If $p_R(x) > 0.02$, we should do targeting.

Expected Value for Classifier Evaluation

We have just seen how to use expected value framework to evaluate an individual decision.

However, we will use a model to make a set of decisions by applying it to a collection of examples.

We need to look at how to evaluate and compare classifiers when applied to a particular problem.



Data set

Induction algorithm

Model

Test

Cost/Benefit information

	p	**n**
Y	b(Y,p)	c(Y,n)
N	c(N,p)	b(N,n)

	p	**n**
Y	**True Positives**	**False Positives**
N	**False Negatives**	**True Negatives**

Normalize to rates

	p	**n**
Y	***tp rate*** **p(Y, p)**	***fp rate*** **p(Y n)**
N	***fn rate*** **p(N, p)**	***tn rate*** **p(N, n)**

Π Σ

Expected value

Confusion matrix

Expected rates (matrix of probabilities)

Computing aggregate performance of a classifier

$$
\begin{aligned}
\text{Expected profit} &= p(Y,p) \cdot b(Y,p) + p(N,p) \cdot b(N,p) \\
&\quad + p(N,n) \cdot b(N,n) + p(Y,n) \cdot b(Y,n)
\end{aligned}
$$

$$
\begin{aligned}
&= p(p) \cdot [p(Y \mid p) \cdot b(Y,p) + p(N \mid p) \cdot b(N,p)] \\
&\quad + p(n) \cdot [p(N \mid n) \cdot b(N,n) + p(Y \mid n) \cdot b(Y,n)]
\end{aligned}
$$

Where do the probabilities come?

Where does the cost/benefit matrix come from?

Where do the probabilities come?

We want to learn:

- when we decide to target consumers
  - what is the probability that they respond?
  - what is the probability that they do not?
- when we do not target consumers
  - would they have responded?
  - ...

We already have the counts necessary to calculate all these in the confusion matrix.

Where do the probabilities come?

A sample confusion matrix with counts:

	positive	**n**egative
Y	56	7
N	5	42

We have a total of $T = 110$ predictions.

- 61 positive examples
- 49 negative examples

$$p(Y, p) = 56/110 = 0.51 \quad p(Y, n) = 7/110 = 0.06$$
$$p(N, p) = 5/110 = 0.05 \quad p(N, n) = 42/110 = 0.38$$

$$p(p) = 0.55 \qquad p(n) = 0.45$$
$$p(Y \mid p) = \text{tp rate} = 56/61 = 0.92 \qquad p(Y \mid n) = \text{fp rate} = 7/49 = 0.14$$
$$p(N \mid p) = \text{fn rate} = 5/61 = 0.08 \qquad p(N \mid n) = \text{tn rate} = 42/49 = 0.86$$

Where does the cost/benefit matrix come from?

The costs and benefits often cannot be estimated from data.

They generally depend on external information provided via analysis of the consequences of decisions in the context of the specific business problem.

- specifying the costs and benefits will take a great deal of time and thought;
- often specified within approximate ranges.

For example, when considering churn, how valuable is it to retain a customer?

- future cell phone usage
- cost of acquiring new users

For simplicity, average estimated costs and benefits are used rather than individual-specific costs and benefits.

Cost-benefit matrix

In our targeted marketing example:

	positive	**n**egative
Y	39.2	-0.8
N	0	0

- make sure that the signs of quantities in the cost-benefit matrix are consistent
- we can focus on minimizing cost rather than maximizing profit
- mathematically, there is no difference; be consistent

Cost-benefit matrix

Common mistake in formulating cost-benefit matrices **double counting**

- a benefit in one cell;
- a negative cost for the same thing in another cell.

To prevent this, compute the benefit improvement for changing the decision on one example.

Cost-benefit matrix

For example, suppose that a fraud case costs $1,000 on average.

- the benefit of catching fraud is $+$\$1,000
- the cost of missing fraud is -$1,000

What would be the improvement in benefit for catching a case of fraud?

$$b(Y,p) - b(N,p) = \$1000 - (-\$1000) = \$2000$$

The improvement should only be about $1,000.

- this error indicates double counting

Solution

- either specify that the benefit is $1,000
- or that the cost of missing fraud is -$1,000, but not both.
- One should be zero.

Example: Best Prospects for a Charity Mailing

Fundraising organizations would like to solicit from a “good” subset of the donors

- a very large subset for an inexpensive, infrequent campaign
- a smaller subset for a focused campaign that includes an expensive incentive package

What exactly is the business problem that we would like to solve?

- Would we like to maximize the total amount of donations?
- Would we like to maximize our donation profit?

Example: Customer Churn

A wireless business has a serious problem with churn.

Marketing has designed a special retention offer.

Our task is to target the offer to some appropriate subset of our customer base.

What exactly is the business problem we would like to solve?

Example: Customer Churn

Why is churn a problem?

- Because it causes us to lose money.

Do we want to target those with the highest probability of defection?

We would not mind losing a costly customer.

- We would like to limit the amount of money we are losing—not simply to keep the most customers.

Example: Customer Churn

Can we use the same idea as in the donations example?

The expected benefit of sending the special offer as

$$\text{Expected benefit of offer} = p(S \mid x) \cdot v_S(x) + (1 - p(S \mid x)) \cdot v_{NS}(x)$$

Further simplify by assuming that the value if the customer does not stay is zero.

Example: Customer Churn

Let us look separately at the expected benefit if we send offer or if we do not send.

$$\mathbb{E}[B_T(x)] = p(S \mid x, T) \cdot (u_S(x) - c) + (1 - p(S \mid x, T)) \cdot (u_{NS}(x) - c)$$

$$\mathbb{E}[B_{NT}(x)] = p(S \mid x, \mathrm{not}\, T) \cdot u_S(x) + (1 - p(S \mid x, \mathrm{not}\, T)) \cdot u_{NS}(x)$$

We would like to target those customers for whom we would see the greatest expected benefit from targeting them.

Example: Customer Churn

Target those customers where

$$\mathrm{VT}(x) = \mathbb{E}[B_T(x)] - \mathbb{E}[B_{NT}(x)]$$

is the largest

Suppose $u_{NS}(x) = 0$

$$\begin{aligned} \mathrm{VT}(x) &= p(S \mid x, T) \cdot u_S(x) - p(S \mid x, \mathrm{not}\, T) \cdot u_S(x) - c \\ &= [p(S \mid x, T) - p(S \mid x, \mathrm{not}\, T)] \cdot u_S(x) - c \end{aligned}$$

- What models do we need to build?

Regression evaluation

When applying machine learning to an actual problem

- What is important in the application?
- What is the goal?
- Are we assessing the results appropriately given the actual goal?

Example: Personalized recommendations

A customer rates a movie by giving it one to five stars.

The recommendation model predicts how many stars a user will give an unseen movie.

An analyst might describe each model by reporting the mean-squared-error (or whatever metric)

- why is this metric appropriate?
- is it meaningful?
- is there a better metric?

Hopefully the analyst thought about the choice of the metric carefully and is not simply reporting some measure he learned about in a class.

Developing a predictive model is an iterative process

It is important to set up an evaluation metric early on.

We need to consider carefully what would be a reasonable baseline against which to compare model performance.

- data science team can understand whether they are improving performance;
- demonstrating to stakeholders that machine learning has added value.

Simple baselines

- limited data sources
- limited features
- domain knowledge

Other evaluation metrics

You will encounter a number of evaluation metrics in data science.

- All of them fundamentally summarize the confusion matrix.

	positive	**n**egative
Y	True positives	False positive
N	False negative	True negatives

True positive rate = TP / (TP + FN) — frequency of being correct
False negative rate = FN / (TP + FN) — frequency of being incorrect

True negative rate = TN / (TN + FP)
False positive rate = FP / (TN + FP)

Other evaluation metrics

You will encounter a number of evaluation metrics in data science.

- All of them fundamentally summarize the confusion matrix.

	positive	**n**egative
Y	True positives	False positive
N	False negative	True negatives

Precision = TP / (TP + FP)
Recall = TP / (TP + FN) = True positive rate
F-measure $= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

Specificity = TN / (TN + FP) = *True negative rate*
Sensitivity = TP / (TP + FN) = *True positive rate*

See: `https://en.wikipedia.org/wiki/Sensitivity_and_specificity`

Visualizing Model Performance

The expected profit framework takes a specific set of conditions and generates a single number that represents a profit.

There are a lot of assumptions and details that go into this calculation.

- knowledge of costs and benefits
- accurate estimates of probabilities

Stakeholders outside data science team may have little patience for detail, so we want to provide a higher-level, more intuitive view of model performance.

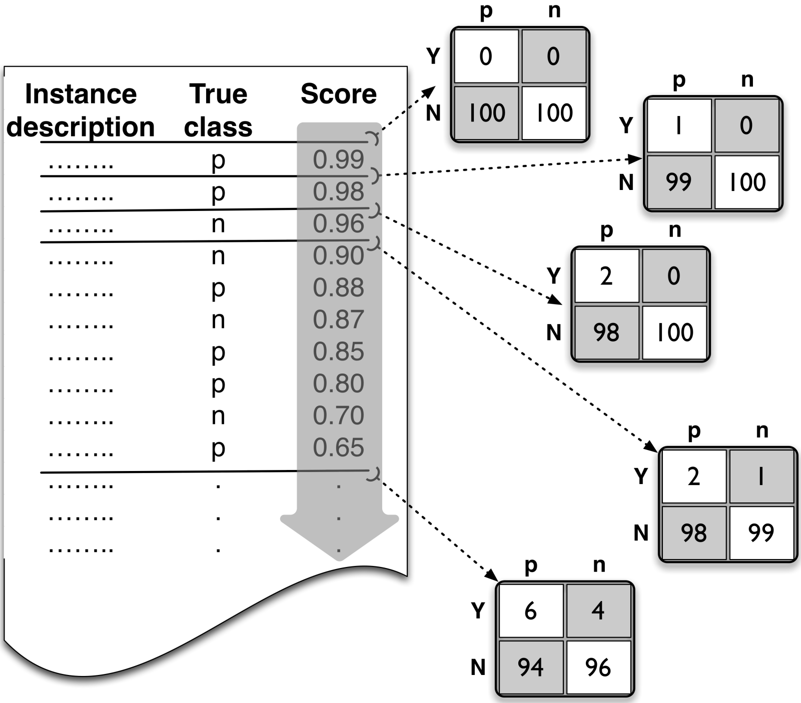
We have seen how the score assigned by a model can be used to compute a decision for each individual case based on its expected value.

A different strategy for making decisions is to rank a set of cases by these scores, and then take actions on the cases at the top of the ranked list.

- we may decide to take the top *n* cases;
- or, equivalently, all cases that score above a given threshold.

This is a good idea/useful when:

- we use classifiers that do not provide true probabilities,
- estimated probabilities are not accurate,
- problems where you have a budget.



Instance description	True class	Score
....	p	0.99
....	p	0.98
....	n	0.96
....	n	0.90
....	p	0.88
....	n	0.87
....	p	0.85
....	p	0.80
....	n	0.70
....	p	0.65
....	.	.
....	.	.
....	.	.

	p	n
Y	0	0
N	100	100

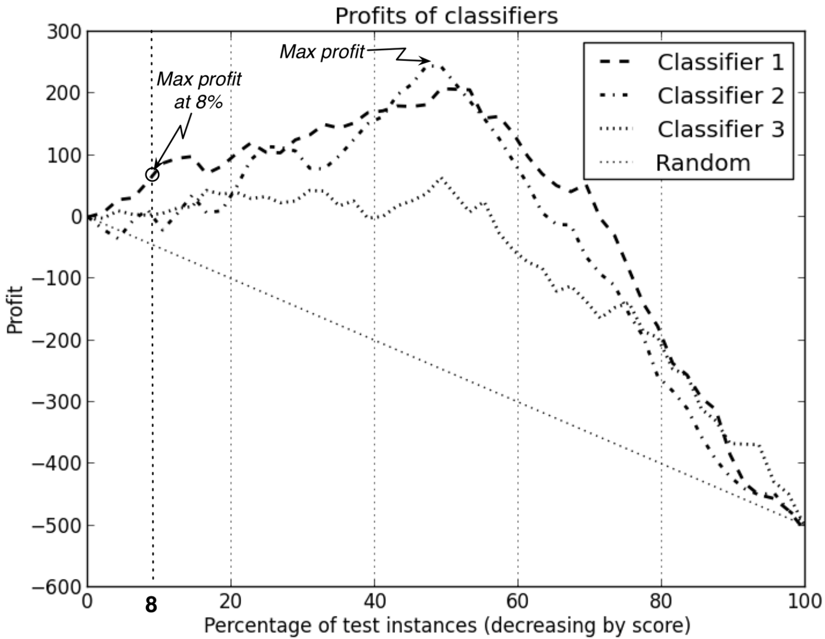
	p	n
Y	1	0
N	99	100

	p	n
Y	2	0
N	98	100

	p	n
Y	2	1
N	98	99

	p	n
Y	6	4
N	94	96

Profit curves



The ROC Curve

Stands for Receiver Operating Characteristic.

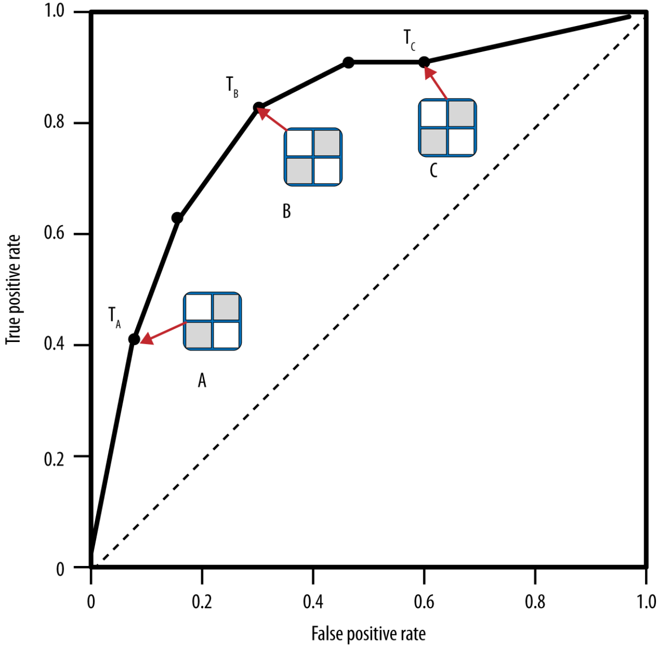
- This used widely in signal processing.

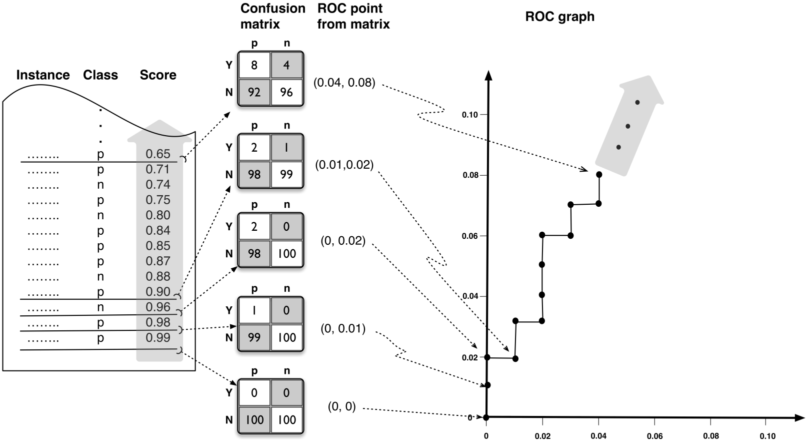
There are two conditions underlying the profit calculation:

- the proportion of positive and negative instances in the target population is known,
- the costs and benefits are known.

If the above are known and are expected to be stable, profit curves may be a good choice for visualization.

The ROC curve shows the entire space of performance possibilities by depicting relative trade-offs that a classifier makes between *benefits* (true positives) and *costs* (false positives).


1.0
0.8
0.6
0.4
0.2
0
True positive rate
0
0.2
0.4
0.6
0.8
1.0
False positive rate
T_A
T_B
T_C
A
B
C



Instance	Class	Score
	.	
	.	
	.	
....	p	0.65
....	p	0.71
....	n	0.74
....	p	0.75
....	n	0.80
....	p	0.84
....	p	0.85
....	p	0.87
....	n	0.88
....	p	0.90
....	n	0.96
....	p	0.98
....	p	0.99

Confusion matrix

ROC point from matrix

	p	n
Y	8	4
N	92	96

(0.04, 0.08)

	p	n
Y	2	1
N	98	99

(0.01,0.02)

	p	n
Y	2	0
N	98	100

(0, 0.02)

	p	n
Y	1	0
N	99	100

(0, 0.01)

	p	n
Y	0	0
N	100	100

(0, 0)

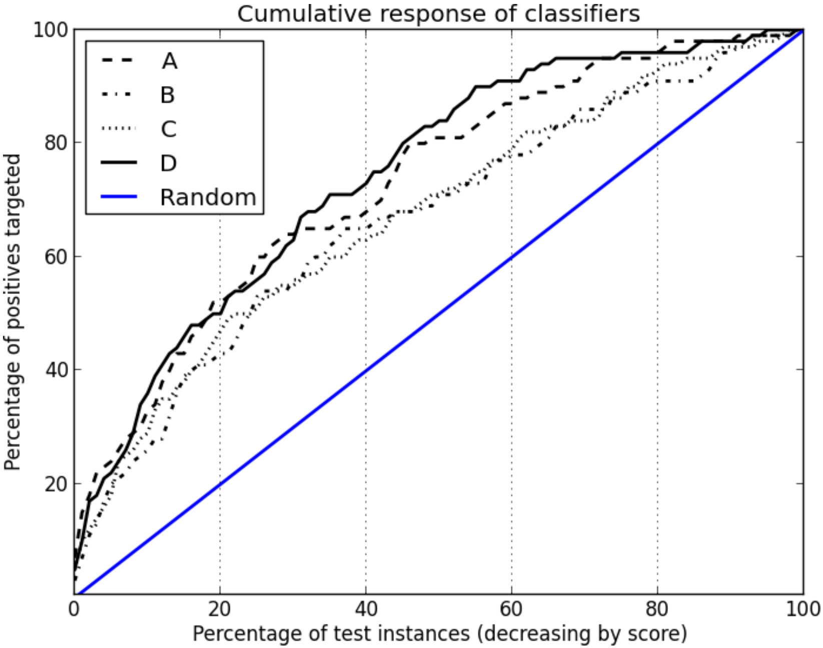
ROC graph

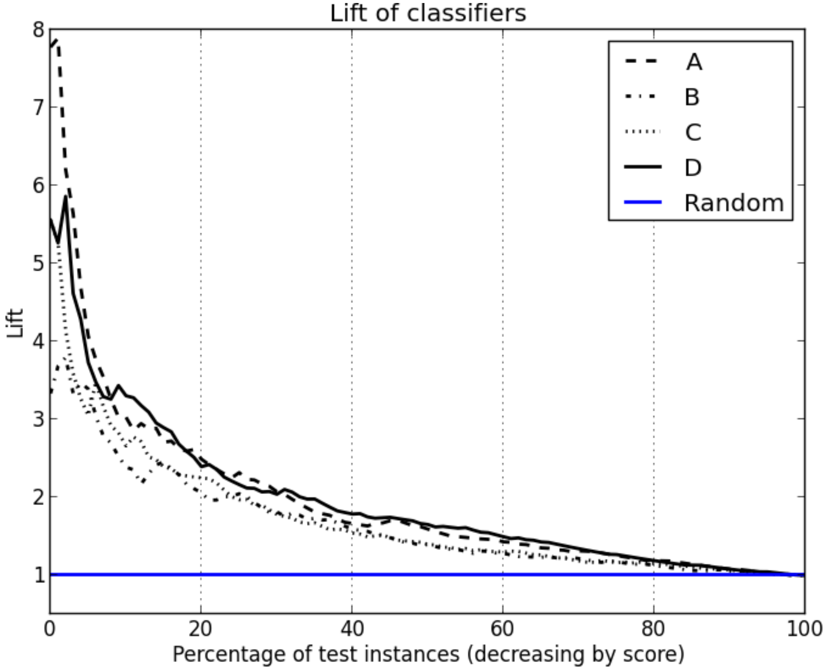
An advantage of ROC graphs is that they decouple classifier performance from the conditions under which the classifiers will be used.

- They are independent of the class proportions as well as the costs and benefits.

An important summary statistic is the *area under the ROC curve* (AUC).

- ROC curve provides more information than its area,
- the AUC is useful when a single number is needed to summarize performance, or
- when nothing is known about the operating conditions.


Cumulative response of classifiers
A
B
C
D
Random
Percentage of positives targeted
Percentage of test instances (decreasing by score)
0
20
40
60
80
100
20
40
60
80
100


Lift of classifiers
Lift
Percentage of test instances (decreasing by score)
0
20
40
60
80
100
1
2
3
4
5
6
7
8
A
B
C
D
Random

Summary

Evaluation of classifiers is a critical part of analysis.

Conveying information about evaluation to stakeholders is equally important.

We have seen how different losses are used during training and testing phase.

Visualization techniques can be used to summarize performance of different procedures.