# Lect3: Classification #2
# Using obtained classifiers

Tokyo Tech.
Intro. to Comp. & Data
Lecture week3

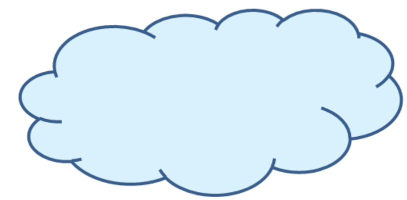
Discuss ways for making use of obtained classifiers.

1. Some basic knowledge from Prob. Theory.
2. How to test the performance of a classifier.
3. How to deal with tradeoff relations.
4. On Exercise #3.

* Some of the slide materials (in particular, green ones) are from the slides of the authors of the textbook and their group at the University of Waikato.

# 1. Basic knowledge on probability
## 1.1. Expectation and Variance

Recall that we assume a distribution $\mathcal{D}$ on a "domain" of instances



$\mathcal{D}$

**Expectation** (often denoted by $\mu$) :

discrete case

$$\mathrm{E}[\,X\,] = \sum_{x\in\mathrm{Range}(X)} x \times \Pr[\,X = x\,]$$

continuous case (omitted below)

$$\mathrm{E}[\,X\,] = \int_{x\in\mathrm{Range}(X)} x \times p(x)$$

where $p$ is a *density function* for $X$ on $\mathcal{D}$.

> **Remark**: In this course, by "mean" we mean the average on a given data set.

**Variance**:

$$\mathrm{V}[\,X\,] = \mathrm{E}[\,(X-\mu)^2] = \sum_{x\in\mathrm{Range}(X)} (x-\mu)^2 \times \Pr[\,X = x\,]$$

$$= \sum_{x\in\mathrm{Range}(X)} (x-\mathrm{E}[X])^2 \times \Pr[\,X = x\,]$$

why squared?

**Standard deviation** (denoted by $\sigma$) : $\sigma = \sqrt{\mathrm{V}[\,X\,]}$

# Important Rules (sometimes called Laws)

Consider $n$ random variables $X_1, \ldots, X_n$ .

in general

in the class

$E[\ \Sigma_i X_i\ ] = \Sigma_i E[X_i]$ ← can be derived from the def.

independent case

$E[\ X_1 \times X_2 \times \cdots\ ] = E[X_1] \times E[X_2] \times \cdots$

pair-wise independence

$E[\ X_i \times X_j\ ] = E[X_i\ ] \times E[X_j]$

$\Rightarrow V[\ X_i + X_j\ ] = V[X_i] + V[X_j] \Rightarrow V[\ \Sigma_i X_i\ ] = \Sigma_i V[X_i]$

in the class

$\Rightarrow V[\ \Sigma_i X_i\ ] = \Sigma_i E[X_i^{\ 2}] - \Sigma_i E[X_i]^{\ 2}$

all with the same exp. $\mu$ and standard deviation $\sigma$ (note that $V[X_i] = \sigma^2$)

$E[\ \Sigma_i X_i\ ] = n\mu \qquad V[\ \Sigma_i X_i\ ] = n\sigma^2 \qquad \sqrt{V[\ \Sigma_i X_i\ ]} = \sigma\sqrt{n}$

# 1.2. Law of large numbers, and ...

## Law of large numbers

Let $x_1,\ x_2, \ldots, x_m$ be the outcomes of independent experiments following the same distribution, i.e., values of some random variable $X$. Then we have

$$\text{empirical mean} := \frac{x_1 + x_2 + \cdots + x_m}{m} \to \mu = \mathrm{E}[\, X \,]$$

No good ! Confusing !

Law of large numbers

> Let $X_1,\ X_2, \ldots, X_m$ be ind. rnd. var.s with the same expectation $\mu$. Then if $m$ is sufficiently large, then we have
>
> $$\Pr\left[\, \frac{X_1 + X_2 + \cdots + X_m}{m} \fallingdotseq \mu \right] = \text{high}$$

How close?
How high?

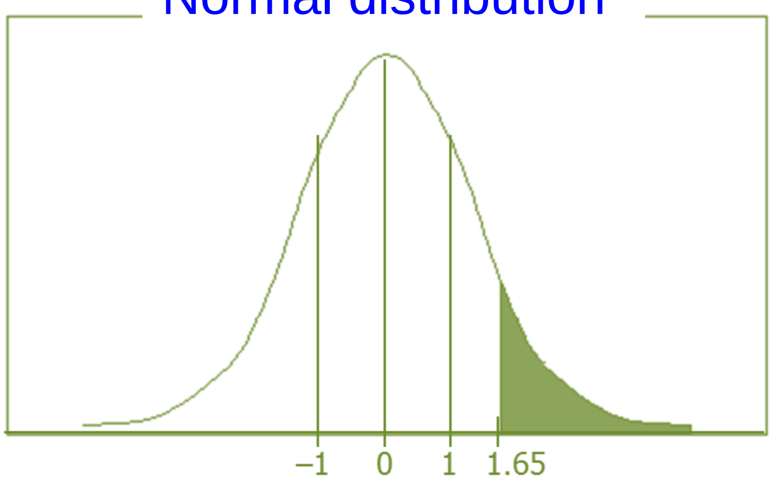
# Central Limit Theorem (Basic version)

Consider a random variable $X$ defined by $X = \sum_{i=1}^{n} X_i \,/\, n$ where $X_1, X_2, \ldots, X_n$ are independent & identical random variables with expectation $\mu$ and variance $\sigma$. Then $X$ converges to the Normal distribution $\mathrm{N}(\mu, \sigma_n)$.

Recall that

$$\mathrm{E}[X] = n\mu \,/\, n = \mu, \quad \sigma_n := \sqrt{\mathrm{V}[X]} = \sigma \,/\, \sqrt{n}$$

Normal distribution




N(0, 1)

| Pr[$X \geq z$] | $z$ |
|---|---|
| 0.1% | 3.09 |
| 0.5% | 2.58 |
| 1% | 2.33 |
| 5% | 1.65 |
| 10% | 1.28 |
| 20% | 0.84 |
| 40% | 0.25 |

For example,

$$\Pr[-1.65 \leq X \leq 1.65] = 90\%$$

# Application of the Central Limit Thm

Suppose that $X = \sum_{i=1}^{n} X_i / n$ is close to $\mathrm{N}(\mu, \sigma_n)$,
where $\mathrm{E}[X] = \mu$ and $\sigma_n := \sigma / \sqrt{n}$ (since $n$ is large enough).
Then we may assume that $(X - \mu) / \sigma_n$ follows $\mathrm{N}(0, 1)$.
Thus, e.g.,

↓ general rules

$$\boxed{\begin{array}{l} \mathrm{E}[cX] = c\mu \\ \sqrt{\mathrm{V}[cX]} = c\sigma_n \end{array}}$$

$$\Pr[(X - \mu) / \sigma_n > 2.33\,] < 0.01$$

$$\Updownarrow$$

$$\Pr[\,X > \mu + 2.33\,\sigma_n\,] < 0.01$$

$$\Updownarrow$$

$$\Pr[\,X > \mu + \underbrace{2.33\,\sigma / \sqrt{n}}_{\text{Gets smaller when } n \text{ increases.}}\,] < 0.01$$

Qualitative version
of the law of large numbers

| $\Pr[X \geq z]$ | $z$ |
|---|---|
| 0.1% | 3.09 |
| 0.5% | 2.58 |
| 1% | 2.33 |
| 5% | 1.65 |
| 10% | 1.28 |
| 20% | 0.84 |
| 40% | 0.25 |

# Application of the Central Limit Thm

Suppose that $X = \sum_{i=1}^{n} X_i \,/\, n$ is close to $\mathrm{N}(\mu, \sigma_n)$,
where $\mathrm{E}[X] = \mu$ and $\sigma_n := \sigma / \sqrt{n}$ (since $n$ is large enough).
Then we may assume that $(X - \mu) / \sigma_n$ follows $\mathrm{N}(0, 1)$.
Thus, e.g.,

How large?

$$\Pr[(X - \mu) / \sigma_n > 2.33\ ] < 0.01$$

$$\Updownarrow$$

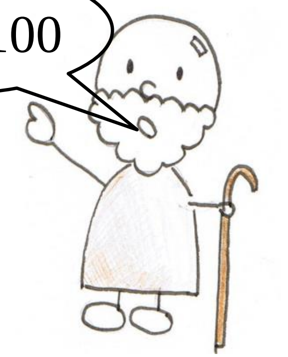
$$\Pr[\ X > \mu + 2.33\ \sigma_n\ ] < 0.01$$

$$\Updownarrow$$

$$\Pr[\ X > \mu + \underbrace{2.33\ \sigma / \sqrt{n}}\ ] < 0.01$$

Gets smaller when $n$ increases.

Well, > 100



| $\Pr[X \geq z]$ | $z$ |
|---|---|
| 0.1% | 3.09 |
| 0.5% | 2.58 |
| 1% | 2.33 |
| 5% | 1.65 |
| 10% | 1.28 |
| 20% | 0.84 |
| 40% | 0.25 |

Qualitative version
of the law of large numbers

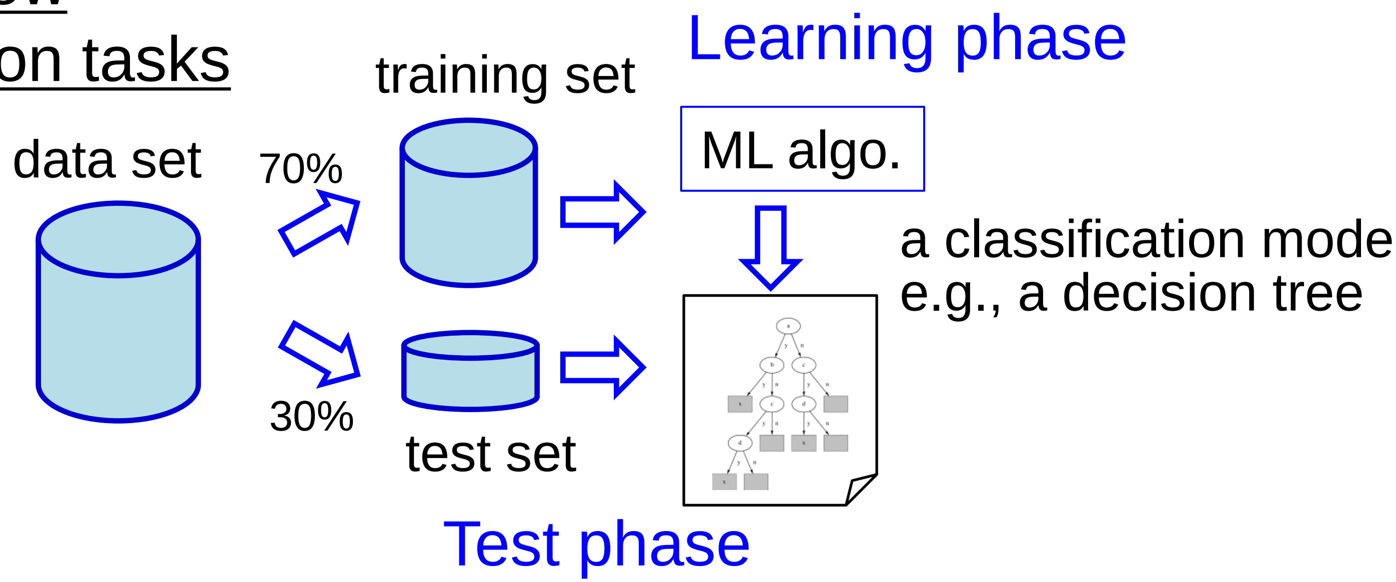
Not rigorous!!

in general

*Chernoff bound*

# 2. Testing classifiers

<u>A standard flow of classification tasks</u>




Suppose that we estimated the error prob. of the obtained decision tree T is $\hat{p}$. What does it mean?

Let $p$ be the error probability of T, and let $X_i$ denote a random variable that takes $1$ (resp., $0$) if T makes an error on the $i$ th instance of the test set. (Let $n$ denote the test set size.) Then $\hat{p}$ is nothing but a value of the random variable $X = \sum_{i=1}^{n} X_i \,/\, n$.

# 2. Testing classifiers

Suppose that we estimated the error prob. of the obtained decision tree T is $\hat{p}$. What does it mean?

Let $p$ be the error probability of T, and let $X_i$ denote a random variable that takes 1 (resp., 0) if T makes an error on the $i$ th instance of the test set. (Let $n$ denote the test set size.) Then $\hat{p}$ is nothing but a value of the random var. $X = \sum_{i=1}^{n} X_i / n$.

Note that

$$E[X_i] = p \qquad E[X] = E[\textstyle\sum_{i=1}^{n} X_i / n] = p$$

$$V[X_i] = p(1-p)$$

$$V[X] = V[\textstyle\sum_{i=1}^{n} X_i / n] = np(1-p) / n^2 = p(1-p) / n$$

$\Rightarrow \sigma_n$ (for $X$) $= \sqrt{p(1-p) / n}$

in the class

For example, let us examine the case $p = 0.2$.

Thus, we have

$$\Pr[\,|(X - p) / \sigma_n| > 2.33\,] < 0.02$$

$$\Leftrightarrow \Pr[\,|\hat{p} - p| > 2.33\,\sigma_n\,] < 0.02$$

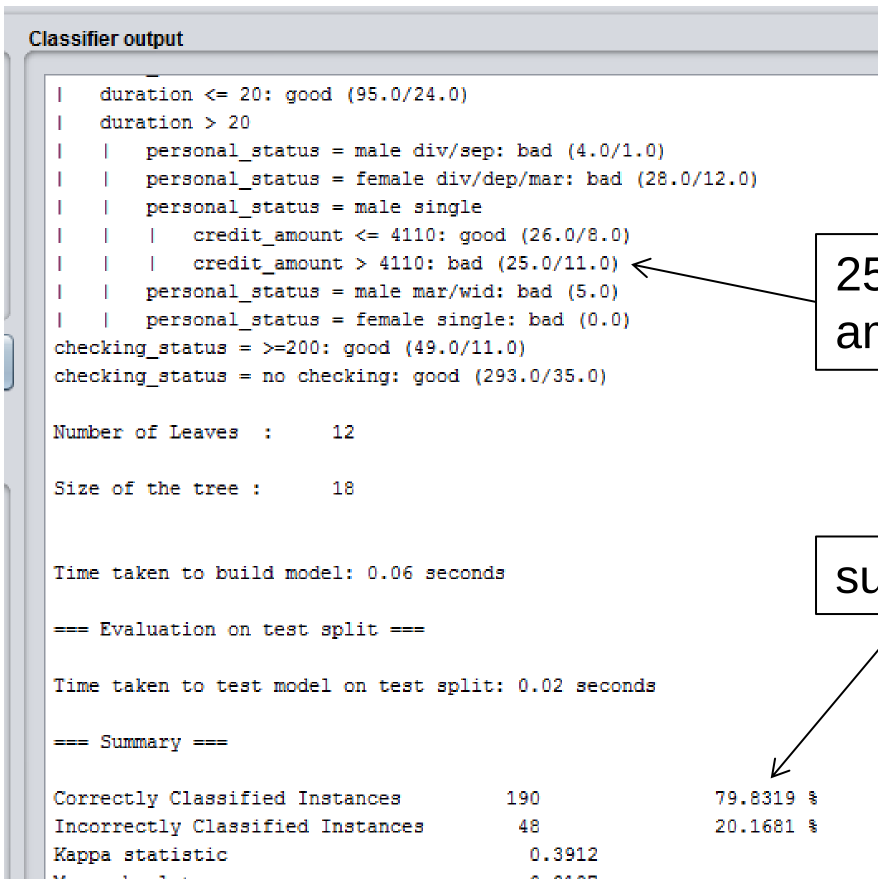
← We may conclude this.

# Testing the quality of the decision at each leaf

Similarly we can estimate the error probability on the decision made at each leaf node of the tree.

Weka

The result of executing "Percentage and split" with default 66%.



Classifier output

```
|   duration <= 20: good (95.0/24.0)
|   duration > 20
|   |   personal_status = male div/sep: bad (4.0/1.0)
|   |   personal_status = female div/dep/mar: bad (28.0/12.0)
|   |   personal_status = male single
|   |   |   credit_amount <= 4110: good (26.0/8.0)
|   |   |   credit_amount > 4110: bad (25.0/11.0)
|   |   personal_status = male mar/wid: bad (5.0)
|   |   personal_status = female single: bad (0.0)
checking_status = >=200: good (49.0/11.0)
checking_status = no checking: good (293.0/35.0)

Number of Leaves  :     12

Size of the tree :      18


Time taken to build model: 0.06 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.02 seconds

=== Summary ===

Correctly Classified Instances         190               79.8319 %
Incorrectly Classified Instances        48               20.1681 %
Kappa statistic                          0.3912
```

the stat. results of the obtained decision tree on the whole (i.e., 700) instances

25 instances reach this node, among which 11 instances are misclassified

success rate (on c238left.txt)

# 2. Testing classifiers
# 2.1. Two well-known techniques

It would be nice if we have enough number of instances for training and testing. In practice, we are given only limited number of instances. We show two techniques for dealing with such situations.

## Cross validation

- *Cross-validation* avoids overlapping test sets
  - First step: split data into *k* subsets of equal size
  - Second step: use each subset in turn for testing, the remainder for training
- Called *k-fold cross-validation*
- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

often used

10-fold cross validation

# Bootstrap

Warning: There are many Bootstrap methods. The following method (from the textbook) is the simplest one.

- The *bootstrap* uses sampling *with replacement* to form the training set
  - Sample a dataset of *n* instances *n* times *with replacement* to form a new dataset of *n* instances
  - Use this data as the training set
  - Use the instances from the original dataset that don't occur in the new training set for testing
- Also called the *0.632 bootstrap*
  - A particular instance has a probability of $1-1/n$ of *not* being picked
  - Thus its probability of ending up in the test data is:

$$\left(1-\frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

  - This means the training data will contain approximately 63.2% of the instances

# 3. Tradeoff relations



- In practice, different types of classification errors often incur different costs
- Examples:
  - Terrorist profiling
    - "Not a terrorist" correct 99.99% of the time
  - Loan decisions
  - Oil-slick detection
  - Fault diagnosis
  - Promotional mailing

Two issues:

- unbalanced ratio
  ↑ by F-value
- unbalanced cost
  ↑ by tradeoff analysis

- The *confusion matrix*:

| | | Predicted class | |
|---|---|---|---|
| | | Yes | No |
| **Actual class** | Yes | True positive | False negative |
| | No | False positive | True negative |

# 3. Tradeoff relations
## 3.1. F-value

❖ The *confusion matrix*:

| | | Predicted class | |
|---|---|---|---|
| | | Yes | No |
| **Actual class** | Yes | True positive | False negative |
| | No | False positive | True negative |

When the positive instance ratio is small, the precision may not be a good measure for the performance of the obtained model.

$$\text{precision (i.e., correct prob.)} = \frac{TP}{\text{Actual Yes}} = \frac{TP}{TP + FN}$$

$$\text{recall} = \frac{TP}{\text{Predicted Yes}} = \frac{TP}{TP + FP}$$

we want both large

$$\text{F-value} = \frac{2}{\text{cor. prob}^{-1} + \text{recall}^{-1}} = \frac{2TP}{2TP + FN + FP}$$

# 3. Tradeoff relations
# 3.2. Lift chart

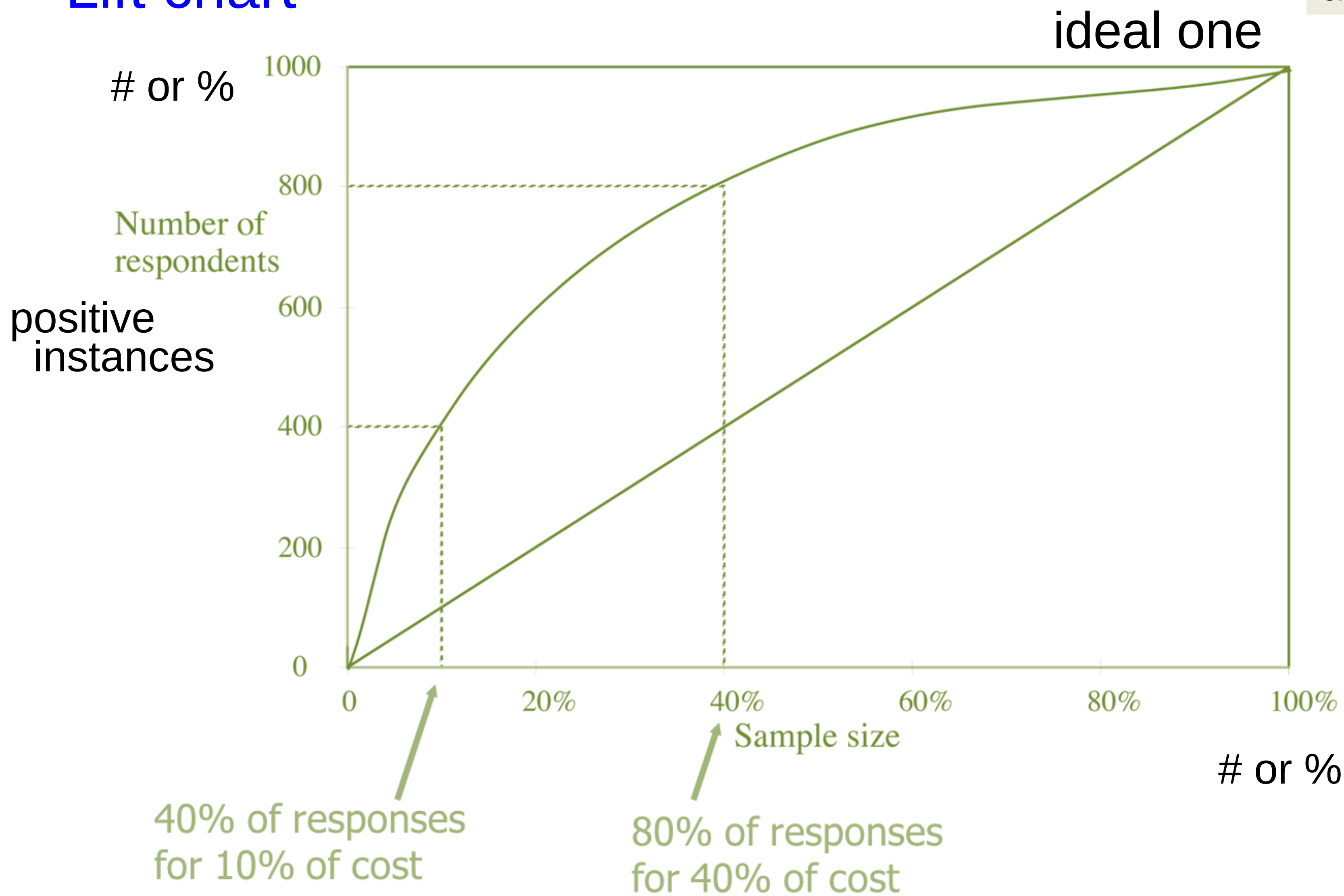
Consider the Naive Bayes method.

- Sort $\underbrace{\text{instances}}_{\text{of test set}}$ according to predicted probability of being positive:

| | Predicted probability | Actual class |
|---|---|---|
| 1 | 0.95 | Yes |
| 2 | 0.93 | Yes |
| 3 | 0.93 | No |
| 4 | 0.88 | Yes |
| ... | ... | ... |

- *x* axis is sample size
  *y* axis is number of true positives

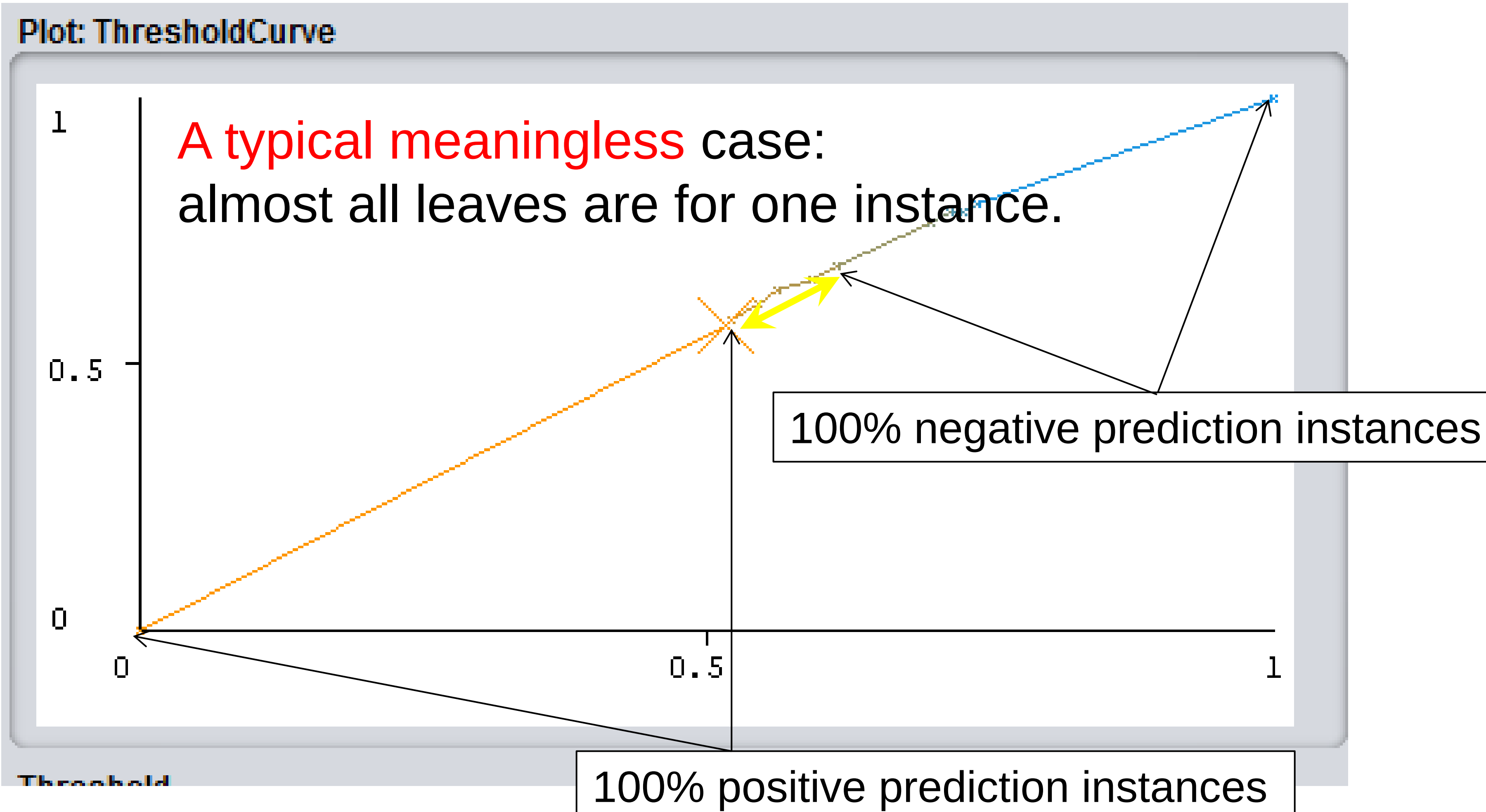
# Lift chart




Similar ones: *ROC curve, Recall-precision*

Can we draw a lift chart for decision trees?

Yes ! By evaluating leaves.




positive instance ratio

# 4. On Exercise #3

Classification rule discovery project:
- How to evaluate and use obtained models.

## Task #1: Understand statistical values

(a) Use credit-g.arff (given as a sample data in Weka) to study the meaning of stat. data on a obtained decision tree for credit-g.arff.

## Task #2: Create better and/or useful rules

data set breast-cancer.arff

no-recurrence-event

(b) Try to make a rule with relatively small false-positive rate by giving more weight to negative instances.

(c) Derive "rules" with the true negative rate > 70%.