Data Science for Non-Programmer

Classification

Denny, Ph.D.

Register – Quiz interactive

- **https://quiz.wicaksana.org/group/pusilkom-dsnp-2021-11-djpb/register**
  - Login menggunakan Google Account
  - Tidak perlu mengisi NIK yang benar

Outline

- **What is Classification and Regression?**
- **Supervised and Unsupervised Learning**
- **Decision Tree**
  - Logarithm
  - Entropy
  - Information Gain
- **Bayesian Classification**
  - Conditional Probability
  - Bayes Theorem
  - Naïve Bayes
- **Model Evaluation**
- **Ensemble**
- **Class Imbalance and Multiclass**

Machine Learning

memberikan mekanisme untuk belajar dari data

- **Ada suatu model (keteraturan/pola) dalam data yang kita miliki**
- **Kita memperkirakan parameter model berdasarkan data yang dapat diamati**
- **kemudian menggunakannya untuk membuat keputusan**

Contoh penerapan:

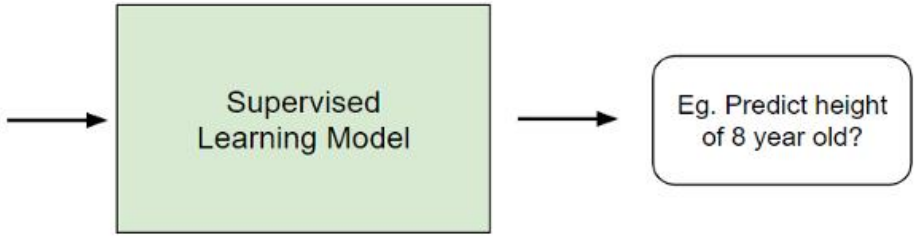
- **Classification (SPAM filtering, Handwriting Recognition)**
- **Prediction (Elections, Market analysis)**
- **Natural Language Processing**
- **...**

Model

- a system or thing used as an example to follow or imitate
- a representation of something in words or numbers that can be used to tell what is likely to happen if particular facts are considered as true

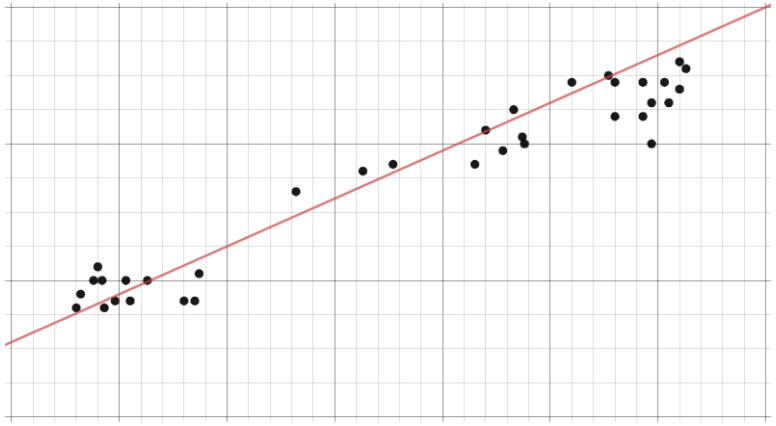
Dataset

Age	Height (inches)
5	47
7	52
11	61
18	71



$$y = mx + c$$

estimate parameter m and c from data



What is Classification and Regression?

- **Classification and regression can be used to extract models describing important data classes or to predict future data trends.**

- **Classification predicts categorical labels.**
  - Ex: categorize bank loan applications → safe or risky.
- **Regression models continuous-valued functions.**
  - Ex: predict the expenditures of potential customers on computer equipment given their income and occupation.

- **Typical Applications:**
  - Credit approval, target marketing,
  - Medical diagnosis, treatment effectiveness analysis

Microsoft Azure Machine Learning Algorithm Cheat Sheet

This cheat sheet helps you choose the best machine learning algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the goal you want to achieve with your data.

What do you want to do?

Extract information from text

Text Analytics

Derives high-quality information from text

Answers questions like: What info is in this text?

- **Extract N-Gram Features from Text** ← Creates a dictionary of n-grams from a column of free text
- **Feature Hashing** ← Converts text data to integer encoded features using the Vowpal Wabbit library
- **Preprocess Text** ← Performs cleaning operations on text, like removal of stop-words, case normalization
- **Word2Vector** ← Converts words to values for use in NLP tasks, like recommender, named entity recognition, machine translation

Predict values

Regression

Makes forecasts by estimating the relationship between values

Answers questions like: How much or how many?

- **Fast Forest Quantile Regression** ← Predicts a distribution
- **Poisson Regression** ← Predicts event counts
- **Linear Regression** ← Fast training, linear model
- **Bayesian Linear Regression** ← Linear model, small data sets
- **Decision Forest Regression** ← Accurate, fast training times
- **Neural Network Regression** ← Accurate, long training times
- **Boosted Decision Tree Regression** ← Accurate, fast training times, large memory footprint

Generate recommendations

Recommenders

Predicts what someone will be interested in

Answers the question: What will they be interested in?

- **SVD Recommender** ← Collaborative filtering, better performance with lower cost by reducing dimensionality

Discover structure

Clustering

Separates similar data points into intuitive groups

Answers questions like: How is this organized?

- **K-Means** ← Unsupervised learning

Find unusual occurrences

Anomaly Detection

Identifies and predicts rare or unusual data points

Answers the question: Is this weird?

- **One Class SVM** ← Under 100 features, aggressive boundary
- **PCA-Based Anomaly Detection** ← Fast training times

Predict between several categories

Multiclass Classification

Answers complex questions with multiple possible answers

Answers questions like: Is this A or B or C or D?

- **Multiclass Logistic Regression** ← Fast training times, linear model
- **Multiclass Neural Network** ← Accuracy, long training times
- **Multiclass Decision Forest** ← Accuracy, fast training times
- **One-vs-All Multiclass** ← Depends on the two-class classifier
- **Multiclass Boosted Decision Tree** ← Non-parametric, fast training times and scalable

Predict between two categories

Two-Class Classification

Answers simple two-choice questions, like yes or no, true or false

Answers questions like: Is this A or B?

- **Two-Class Support Vector Machine** ← Under 100 features, linear model
- **Two-Class Averaged Perceptron** ← Fast training, linear model
- **Two-Class Decision Forest** ← Accurate, fast training
- **Two-Class Logistic Regression** ← Fast training, linear model
- **Two-Class Boosted Decision Tree** ← Accurate, fast training, large memory footprint
- **Two-Class Neural Network** ← Accurate, long training times

Classify images

Image Classification

Classifies images with popular networks

Answers questions like: What does this image represent?

- **DenseNet** ← High accuracy, better efficiency

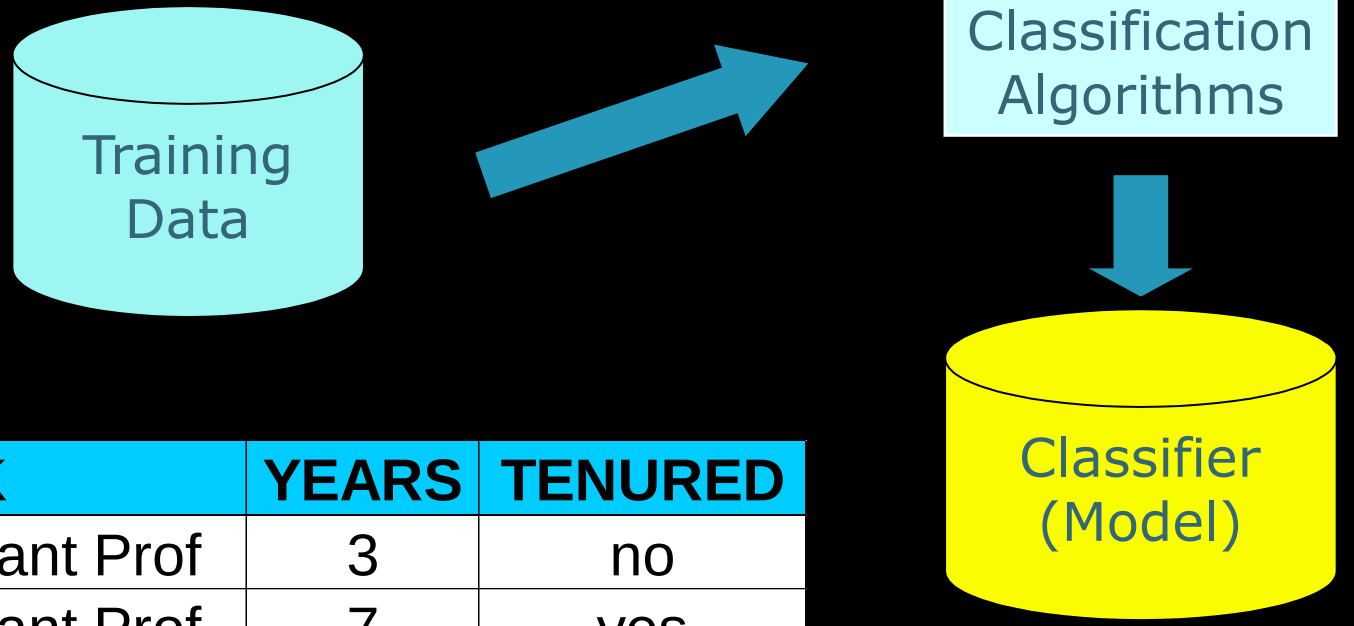
Share this poster: aka.ms/mlcheatsheet

Microsoft

What is Classification? – A two-step process

- **Model construction:**
  - Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the **class label**.
  - Data tuples are also referred to as *samples, examples,* or *objects.*
  - All tuples used for construction is called **training set**.
  - Since the class label of each training sample *is provided* → **supervised learning**. In clustering (**unsupervised learning**), the class labels of each training sample is not known, and the number or set of classes to be learned may not be known in advance.
  - The model is represented in the following forms:
    - Classification rules, (IF-THEN statements), decision tree, mathematical formulae

Classification Process (1)



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

14

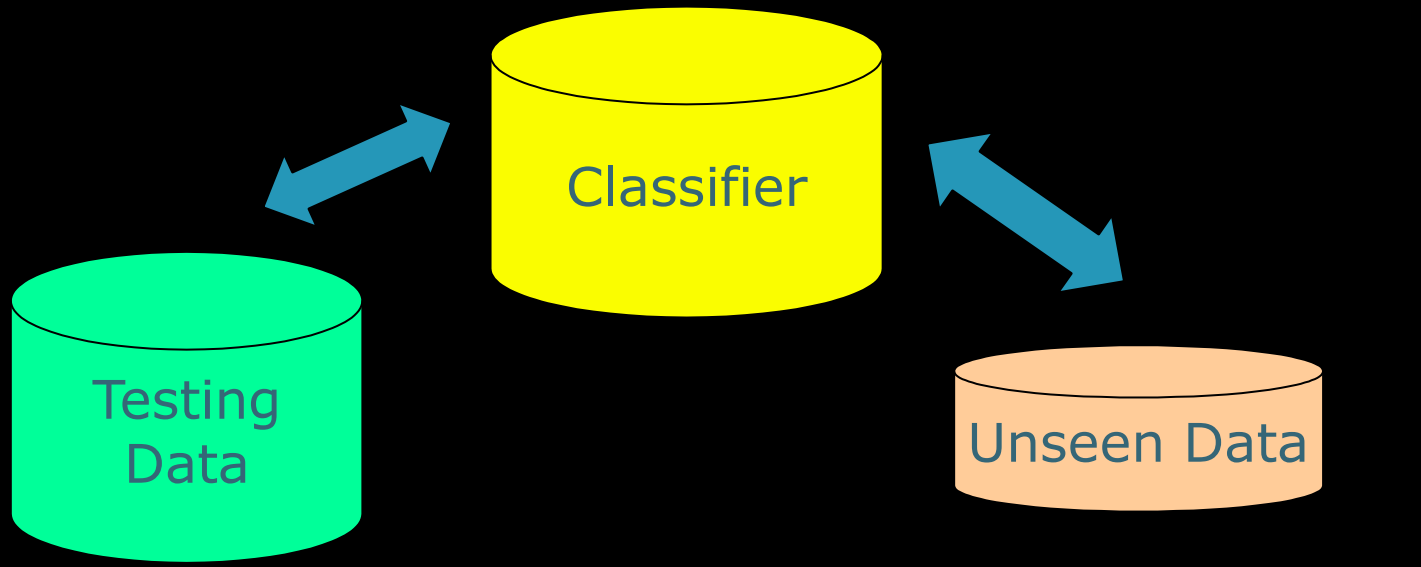
What is Classification? – A two-step process (2)

- **The model is used for classifying future or unknown objects.**
  - First, the predictive accuracy of the model is estimated
    - The known label of test sample is compared with the classified result from the model.
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model.
    - Test set is independent of training set otherwise **over-fitting** (it may have incorporated some particular anomalies of the training data that are not present in the overall sample population) will occur.

What is Classification? – A two-step process (3)

- **The model is used for classifying future or unknown objects.**
  - If the accuracy of the model is considered acceptable, the model can be used to classify future objects for which the class label is not known (unknown, previously unseen data).

Classification Process (2)



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Tenured?

Yes

What is Regression/Prediction?

- **Regression is similar to classification**
  - First, construct model.
  - Second, use model to predict future or unknown objects
    - Major method for prediction is regression:
      - Linear and multiple regression
      - Non-linear regression
- **Regression is different from classification**
  - Classification refers to predict categorical class label.
  - Regression refers to predict continuous value.

Classification vs Regression??

- **Permodelan tarif sewa ATM dan tarif sewa ruang di Indonesia.**

- **Memetakan wajib pajak berdasarkan tingkat kepatuhannya dalam memenuhi kewajiban perpajakannya dan dampak fiskal yang kemungkinan hilang dari ketidakpatuhannya tersebut, sehingga dapat direkomendasikan wajib pajak mana saja yang perlu dilakukan pengawasan secara intensif.**

- **Proses Pre-Customs Clearance yang terlalu panjang sehingga perlu penyederhanaan dan dimaksimalkan di Post-Customs Clearance saja. Hal ini penting untuk institusi karena dapat memotong dwelling time dan memperlancar arus barang dengan risiko lebih rendah.**

Logarithms

- **The logarithm of x to the base b, and is denoted $\log_b x$.**

$$y = \log_b x \quad \text{if and only if} \quad x = b^y \qquad (x > 0)$$

$\log x = \log_{10} x$ **Common logarithm**

$\log_2 x$ **Logarithm base 2 (commonly used in computer)**

Laws of Logarithms

- **If *m* and *n* are positive numbers, then**

$$\log_b mn = \log_b m + \log_b n$$

$$\log_b \frac{m}{n} = \log_b m - \log_b n$$

$$\log_b m^n = n \log_b m$$

$$\log_b 1 = 0$$

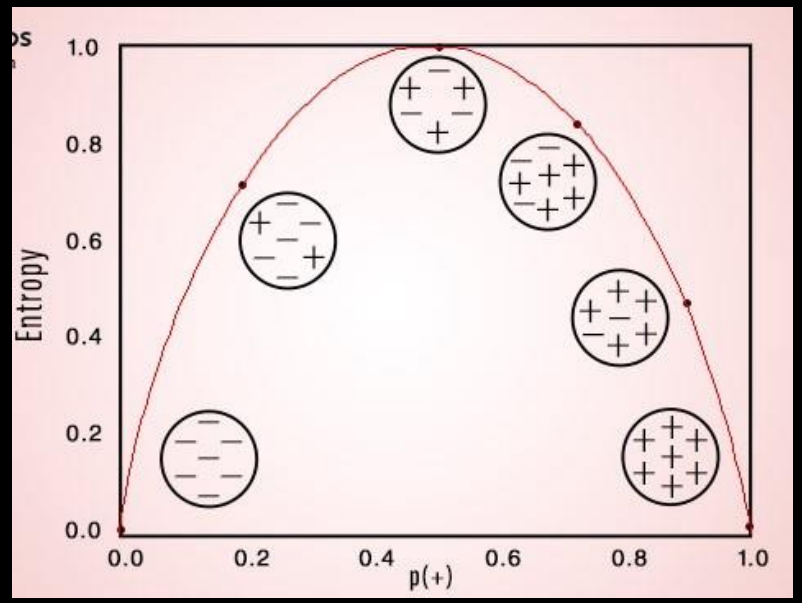
$$\log_b b = 1$$

$$\log_b a = \frac{\log a}{\log b} = \frac{\ln a}{\ln b}$$

ENTROPY

Attribute Selection Measure (review)

- **Entropy: measure of disorder, or measure of purity.**
  - The measurement of the impurity or randomness in the data points.
  - Pengukuran ketidakmurnian / kekacauan / ketidakteraturan pada data
- **Interpretation:**
  - Higher entropy ➔ higher uncertainty / impurity / disorder / high information / surprising
  - Lower entropy ➔ lower uncertainty / impurity / disorder / low information / unsurprising
- **A single toss of a fair coin has an entropy of one bit.**

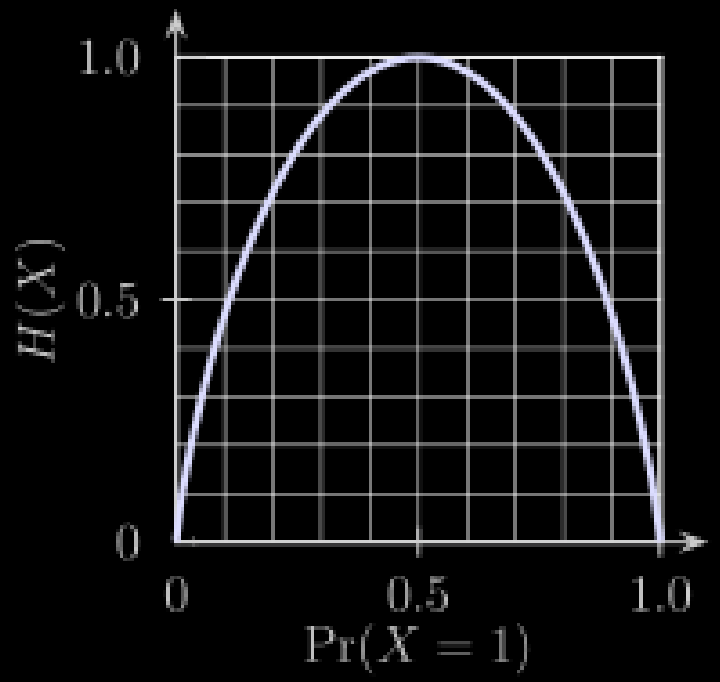


Entropy

- **For a discrete random variable *Y* taking *m* distinct values $\{y_1, y_2, \ldots, y_m\}$**
- **Untuk variabel acak diskrit *Y* yang memiliki m nilai berbeda**

$$H(X) = -\sum_{i=1}^{m} P_i \times log_2(P_i)$$

where $P_i = P(y = y_i)$



Entropy: two class, example

- **For two class (e.g. yes/no, positive/negative):**

$$H(X) = -P_1 \times log_2(P_1) - P_2 \times log_2(P_2)$$

$$H(X) = -\frac{|\boldsymbol{D_1}|}{|\boldsymbol{D}|} \times log_2\left(\frac{|\boldsymbol{D_1}|}{|\boldsymbol{D}|}\right) - \frac{|\boldsymbol{D_2}|}{|\boldsymbol{D}|} \times log_2\left(\frac{|\boldsymbol{D_2}|}{|\boldsymbol{D}|}\right)$$

- **3/7 yes, 4/7 no**
  - Entropy = $-\frac{3}{7} log_2\left(\frac{3}{7}\right) - \frac{4}{7} log_2\left(\frac{4}{7}\right) = 0.9852$

- **6/7 yes, 1/7 no**
  - Entropy = $-\frac{6}{7} log_2\left(\frac{6}{7}\right) - \frac{1}{7} log_2\left(\frac{1}{7}\right) = 0.5917$

Entropy

which one is easier to guess?

AAAAAAAA
➔ entropy = 0

AAAABBCD
➔ entropy = 1.75

AABBCCDD
➔ entropy = 2

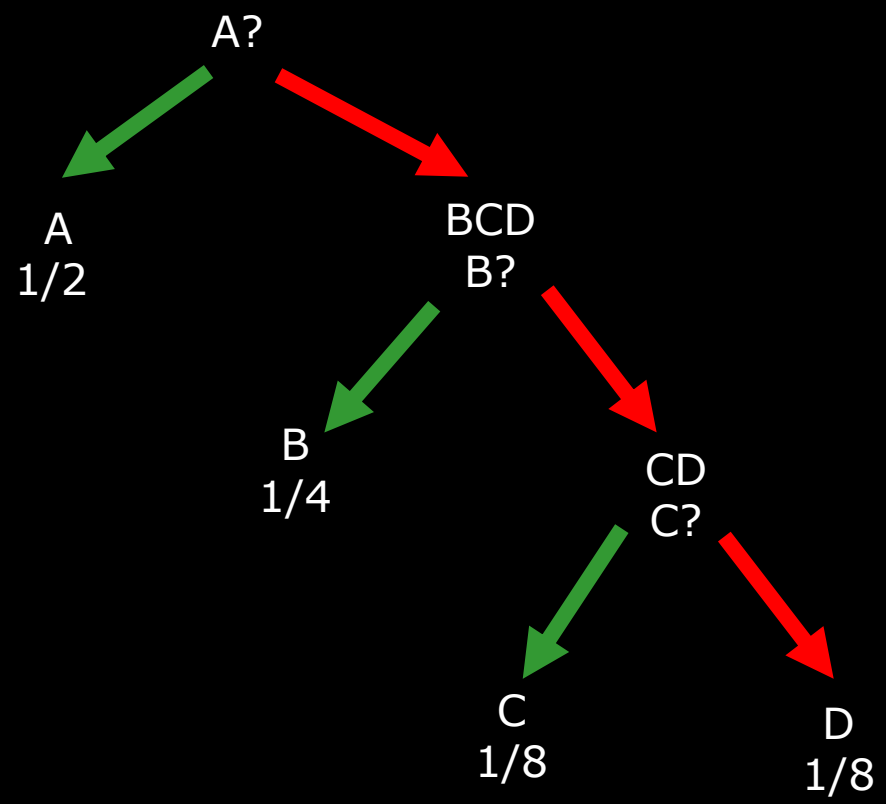
Q: average number of yes/no questions we need to ask to guess what letter we picked

rata-rata jumlah pertanyaan ya / tidak yang perlu ditanyakan untuk menebak huruf apa yang diambil dari kumpulan ini.

Entropy

- **AAAABBCD**

- **Q: average number of yes/no questions we need to ask to guess what letter we picked**



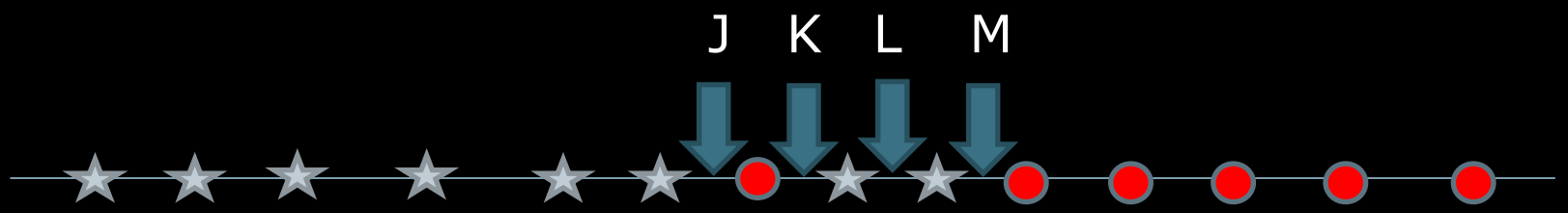
P(A) = 1/2
P(B) = 1/4
P(C) = 1/8
P(D) = 1/8

average number of yes/no questions =
½ * 1 + ¼ * 2 + 1/8 * 3 + 1/8 * 3 = 1.75

entropy = 1.75

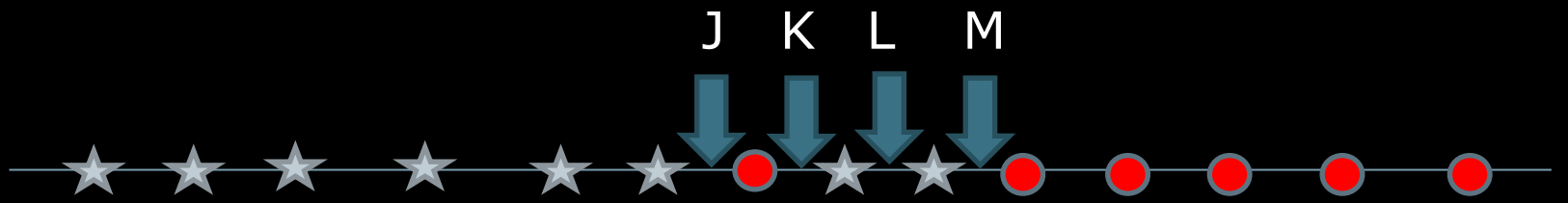
Finding the best split using Entropy

- **Let attribute A be a continuous-valued attribute**
- **Must determine the *best split point* for A**
  - Sort the value A in increasing order
  - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
  - The point with the *minimum expected information requirement* for A is selected as the split-point for A



which one is the best split?

Finding the best split using Entropy



$$Info_A(D) = \frac{|D_1|}{|D|} \times Entropy(D_1) + \frac{|D_2|}{|D|} \times Entropy(D_2)$$

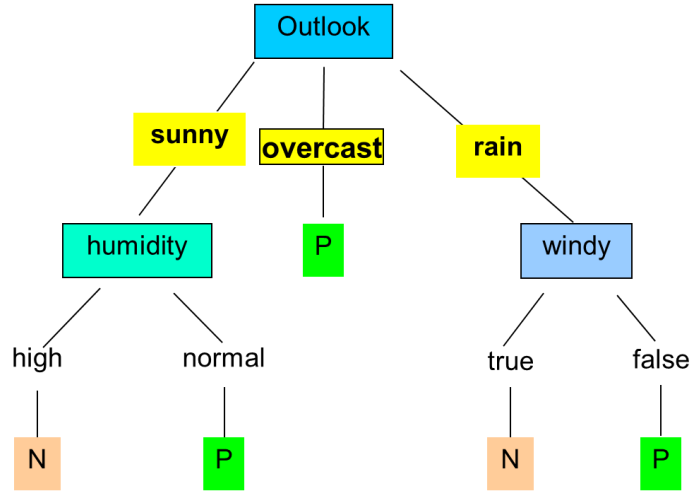
D1 is the set of tuples in D satisfying A ≤ split-point, and D2 is the set of tuples in D satisfying A > split-point

	Left Partition						Right Partition						Net
	Star	Circle	|D1|	P1	P2	Entropy	Star	Circle	|D2|	P1	P2	Entropy	Entropy
J	6	0	6	1.00	-	-	2	6	8	0.25	0.75	0.24	0.14
K	6	1	7	0.86	0.14	0.18	2	5	7	0.29	0.71	0.26	0.22
L	7	1	8	0.88	0.13	0.16	1	5	6	0.17	0.83	0.20	0.18
M	8	1	9	0.89	0.11	0.15	0	5	5	-	1.00	-	0.10

DECISION TREE

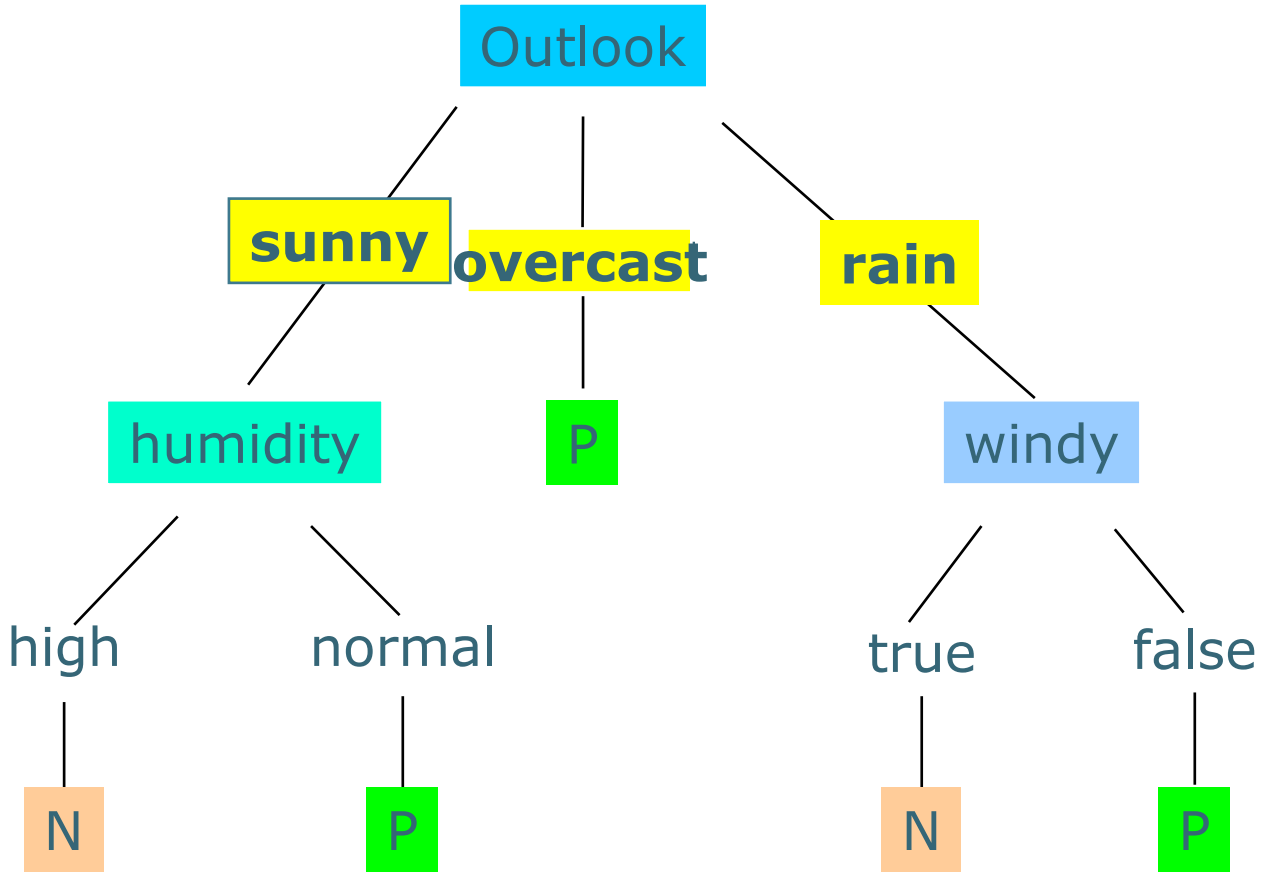
Apa itu decision tree?

- **bagan alur yang memiliki struktur seperti sebuah pohon**
  - Internal node/simpul: sebuah pengujian/tes sebuah atribut/fitur
  - Cabang/branch: hasil tes
  - Leaf node/daun: class label or distribusinya/komposisi atau hasil prediksinya.

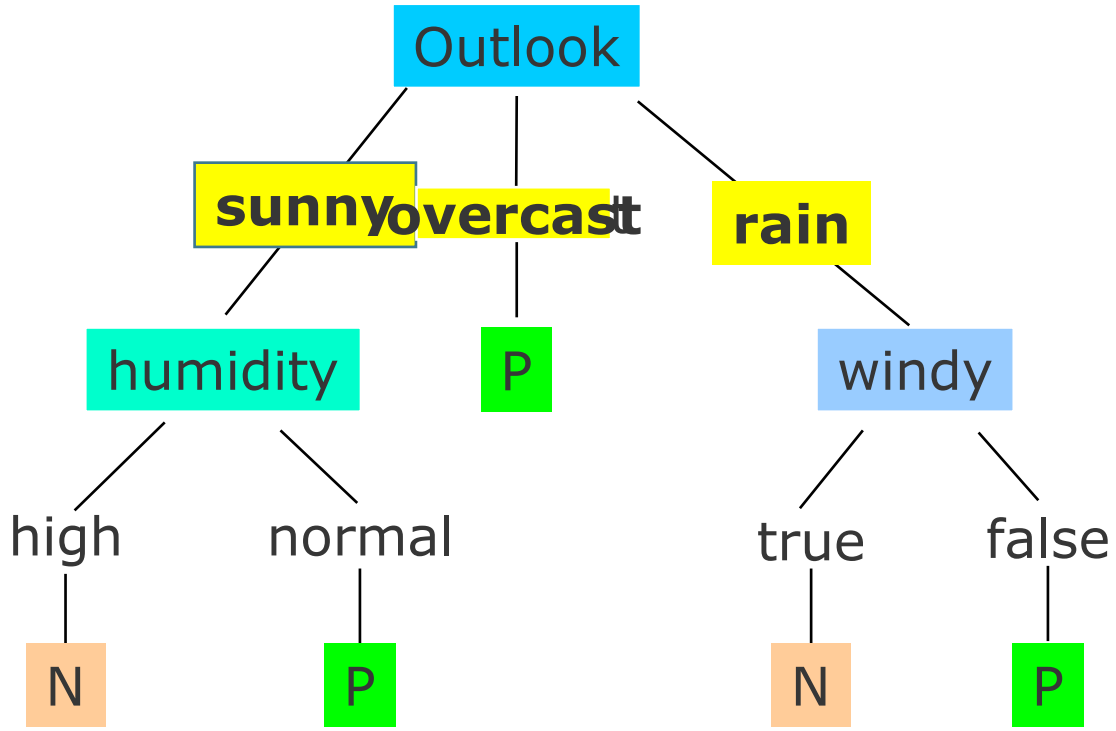


Classification using Decision Tree

Untuk mengklasifikasikan data yang tidak diketahui, nilai dari atribut diuji berdasarkan decision tree/pohon keputusan dari akar/root (di atas) sampai ke daun/leaf yang berisi hasil prediksinya untuk data tersebut.



Decision Tree → IF THEN



if Outlook = sunny and humidity = high then NEGATIVE
if Outlook = sunny and humidity = normal then POSITIVE
if Outlook = overcast then POSITIVE
...

Training Dataset

Outlook		Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

ID3 Algorithm

- **All attributes are categorical**
  - Create a node N;
  - if samples are all of the same class C, then
    - return N as a leaf node labeled with C
  - if attribute-list is empty then
    - return N as a leaf node labeled with the most common class
  - select split-attribute with highest information gain
    - label N with the split-attribute
    - for each value A_i of split-attribute, grow a branch from Node N
    - let S_i be the branch in which all tuples have the value A_i for split- attribute
    - if S_i is empty then
      - attach a leaf labeled with the most common class
      - Else recursively run the algorithm at Node S_i
  - until all branches reach leaf nodes

5 N and 9 P

Outlook	Temperature	Humidity	Windy	Class
overcast	mild	high	true	P
overcast	cool	normal	true	P
overcast	hot	high	false	P
overcast	hot	normal	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
sunny	cool	normal	false	P
rain	mild	high	true	N
rain	cool	normal	true	N
sunny	hot	high	true	N
sunny	hot	high	false	N
sunny	mild	high	false	N

Attribute Selection Measure: Information Gain (ID3/C4.5)

- **Assume all attributes to be categorical (discrete-values). Continuous-valued attributes must be discretized.**
- **Used to select the test attribute at each node in the tree.**
- **Also called *measure of the goodness of split*.**
- **The attribute with the highest information gain is chosen as the test attribute for the current node.**

Attribute Selection Measure: Information Gain (ID3/C4.5)

- **Select the attribute with the highest information gain**
- **Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by: $p_i = \frac{|C_{i,D}|}{|D|} = \frac{|D_i|}{|D|}$**
- **Expected information (entropy) needed to classify a tuple in D:**

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- **Information needed (after using A to split D into v partitions) to classify D:**

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- **Information gained by branching on attribute A**

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection Measure: Information Gain (ID3/C4.5)

- **The attribute *A* is selected such that the *information gain***

$$Gain(A) = Info(D) - Info_A(D)$$

is maximal, that is, $Info_A(D)$ is minimal since $Info(D)$ is the same to all attributes at a node.

- **In the given sample data, attribute *outlook* is chosen to split at the root :**

gain(outlook) = 0.246

gain(temperature) =

gain(humidity) =

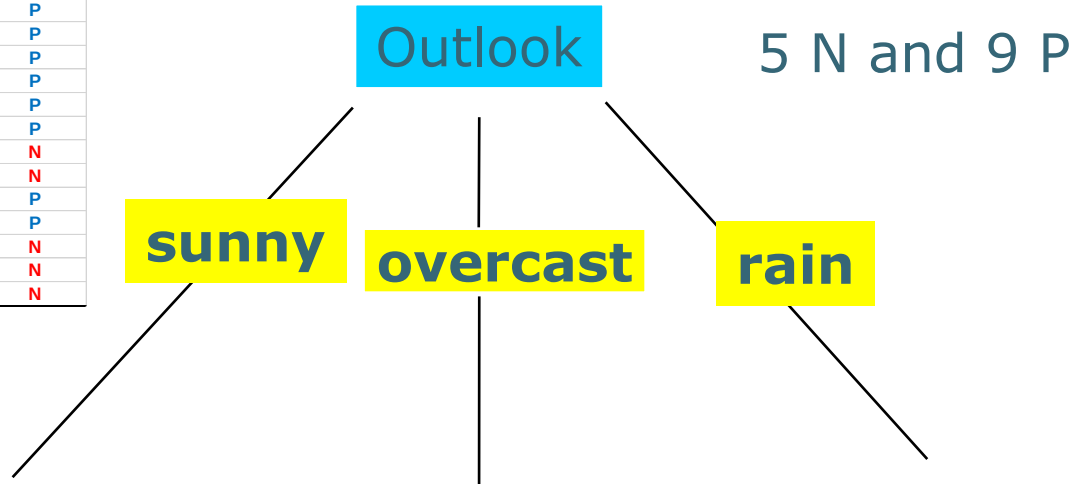
gain(windy) =

5 N and 9 P

Outlook	Temperature	Humidity	Windy	Class
overcast	mild	high	true	P
overcast	cool	normal	true	P
overcast	hot	high	false	P
overcast	hot	normal	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
sunny	cool	normal	false	P
rain	mild	high	true	N
rain	cool	normal	true	N
sunny	hot	high	true	N
sunny	hot	high	false	N
sunny	mild	high	false	N

	Entropy = - P1 * log (P1) - P2 * log (P2)								
BEFORE SPLIT	P	N	P1	log (P1)	P2	log (P2)	Entropy		
	5	9	0.357	-1.485	0.643	-0.637	0.940		
OUTLOOK	P	N	P1	log (P1)	P2	log (P2)	Entropy	**weight**	net entropy
overcast	4	0	1.000	-	-	#NUM!	-	0.285714	-
rain	3	2	0.600	-0.737	0.400	-1.322	0.971	0.357143	0.347
sunny	2	3	0.400	-1.322	0.600	-0.737	0.971	0.357143	0.347
									0.694
							Gain		**0.247**
Temperature	P	N	P1	log (P1)	P2	log (P2)	Entropy	**weight**	net entropy
hot									
cold									
mild									
Humidity	P	N	P1	log (P1)	P2	log (P2)	Entropy	**weight**	net entropy
high									
normal									
Windy	P	N	P1	log (P1)	P2	log (P2)	Entropy	**weight**	net entropy
true									
false									

Outlook	Temperature	Humidity	Windy	Class
overcast	mild	high	true	P
overcast	cool	normal	true	P
overcast	hot	high	false	P
overcast	hot	normal	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	mild	normal	false	P
rain	mild	high	true	N
rain	cool	normal	true	N
sunny	mild	normal	true	P
sunny	cool	normal	false	P
sunny	hot	high	true	N
sunny	hot	high	false	N
sunny	mild	high	false	N

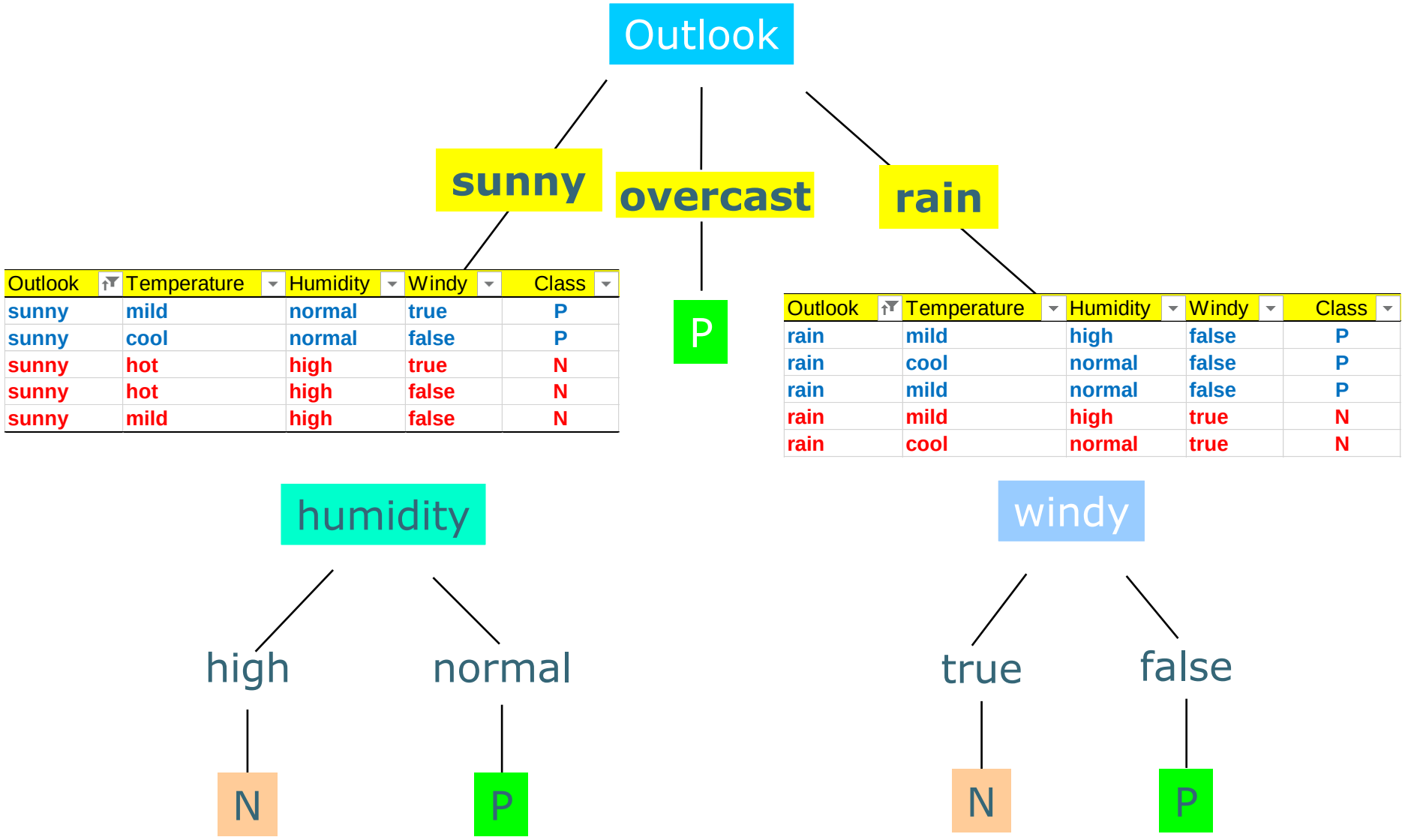


5 N and 9 P

Outlook	Temperature	Humidity	Windy	Class
sunny	mild	normal	true	P
sunny	cool	normal	false	P
sunny	hot	high	true	N
sunny	hot	high	false	N
sunny	mild	high	false	N

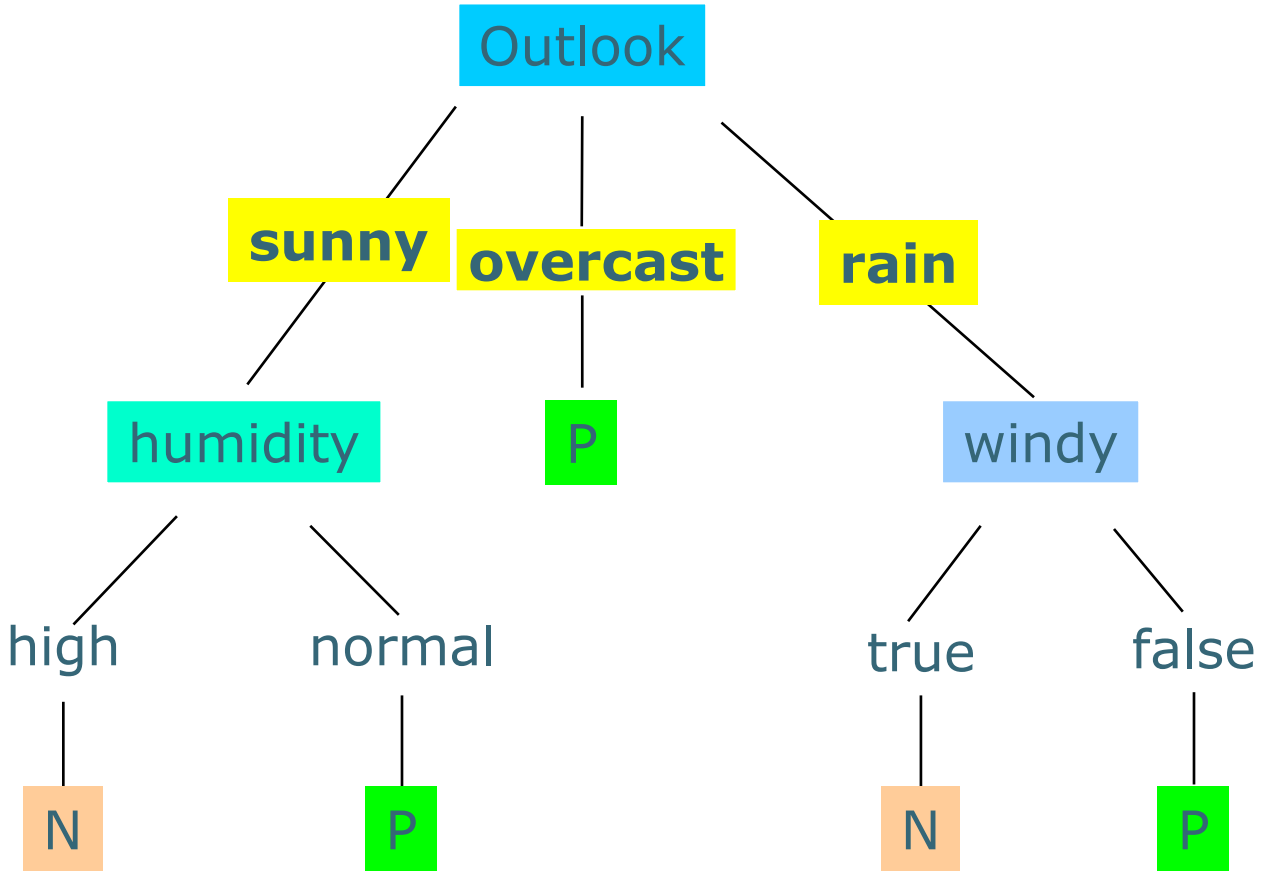
Outlook	Temperature	Humidity	Windy	Class
rain	mild	high	false	P
rain	cool	normal	false	P
rain	mild	normal	false	P
rain	mild	high	true	N
rain	cool	normal	true	N

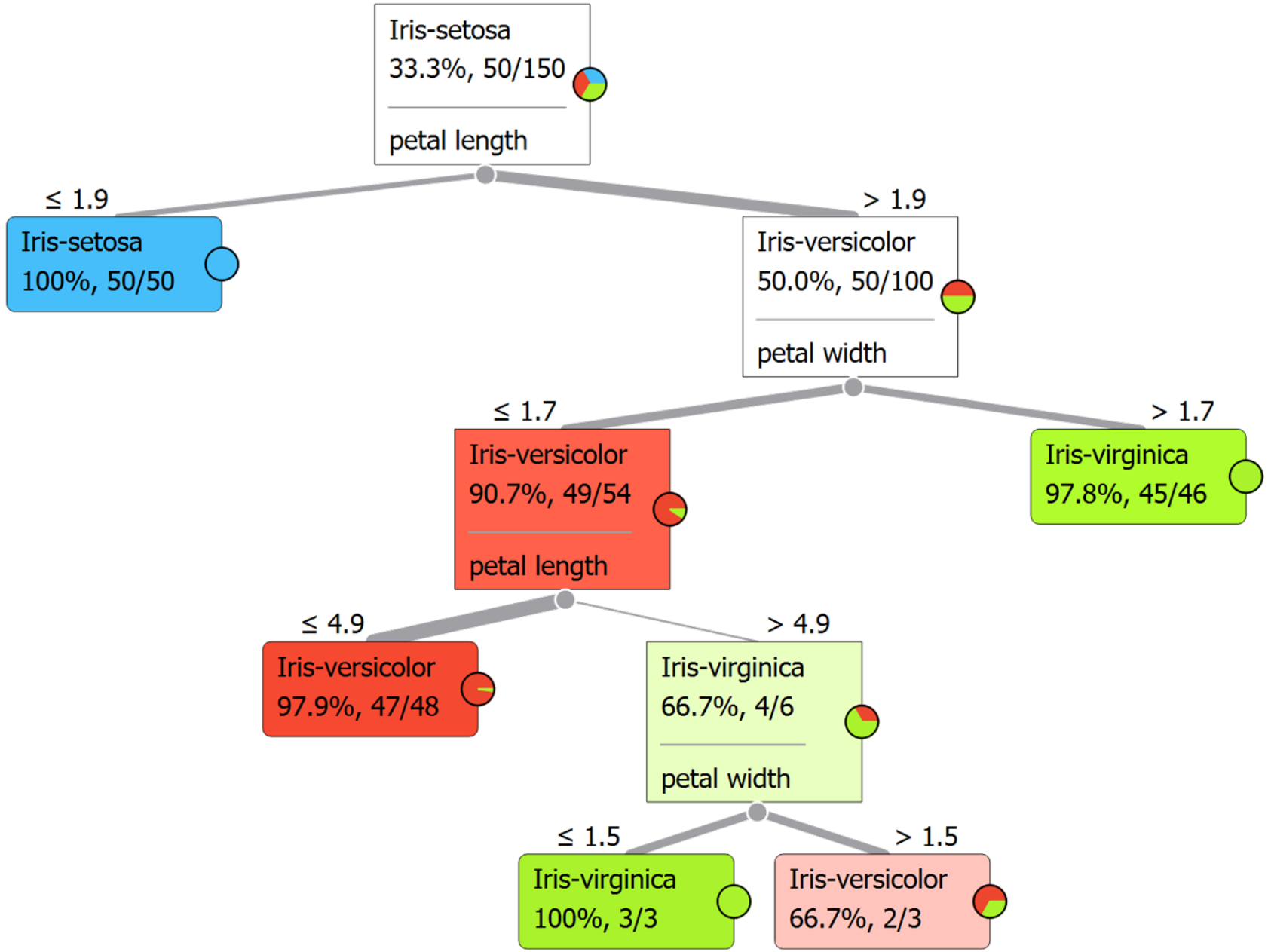
Outlook	Temperature	Humidity	Windy	Class
overcast	mild	high	true	P
overcast	cool	normal	true	P
overcast	hot	high	false	P
overcast	hot	normal	false	P



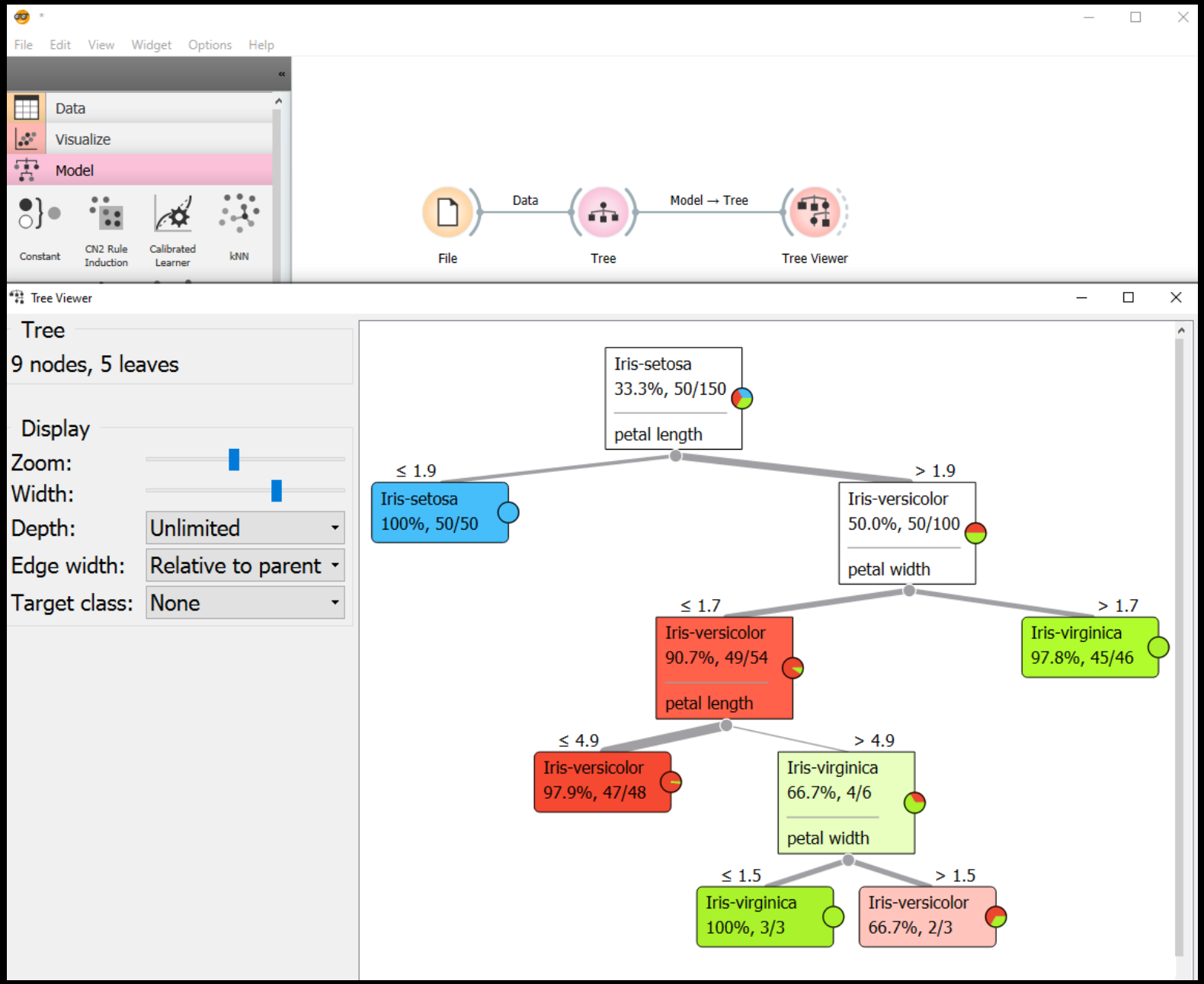
Outlook	Temperature	Humidity	Windy	Class
sunny	mild	normal	true	P
sunny	cool	normal	false	P
sunny	hot	high	true	N
sunny	hot	high	false	N
sunny	mild	high	false	N

Outlook	Temperature	Humidity	Windy	Class
rain	mild	high	false	P
rain	cool	normal	false	P
rain	mild	normal	false	P
rain	mild	high	true	N
rain	cool	normal	true	N


Outlook
sunny
overcast
rain
humidity
P
windy
high
normal
true
false
N
P
N
P


Iris-setosa
33.3%, 50/150
petal length
≤ 1.9
Iris-setosa
100%, 50/50
> 1.9
Iris-versicolor
50.0%, 50/100
petal width
≤ 1.7
Iris-versicolor
90.7%, 49/54
petal length
> 1.7
Iris-virginica
97.8%, 45/46
≤ 4.9
Iris-versicolor
97.9%, 47/48
> 4.9
Iris-virginica
66.7%, 4/6
petal width
≤ 1.5
Iris-virginica
100%, 3/3
> 1.5
Iris-versicolor
66.7%, 2/3

PRAKTEK DENGAN ORANGE


File Edit View Widget Options Help
Data
Visualize
Model
Constant
CN2 Rule Induction
Calibrated Learner
kNN
File
Data
Tree
Model → Tree
Tree Viewer
Tree Viewer
Tree
9 nodes, 5 leaves
Display
Zoom:
Width:
Depth: Unlimited
Edge width: Relative to parent
Target class: None
Iris-setosa
33.3%, 50/150
petal length
≤ 1.9
> 1.9
Iris-setosa
100%, 50/50
Iris-versicolor
50.0%, 50/100
petal width
≤ 1.7
> 1.7
Iris-versicolor
90.7%, 49/54
petal length
Iris-virginica
97.8%, 45/46
≤ 4.9
> 4.9
Iris-versicolor
97.9%, 47/48
Iris-virginica
66.7%, 4/6
petal width
≤ 1.5
> 1.5
Iris-virginica
100%, 3/3
Iris-versicolor
66.7%, 2/3

How to use a tree?

- **Directly**
  - test the attribute value of unknown sample against the tree.
  - A path is traced from root to a leaf which holds the label
- **Indirectly**
  - decision tree is converted to classification rules
  - one rule is created for each path from the root to a leaf
  - IF-THEN is easier for humans to understand
    - Example:

      IF age = “<=30” AND student = “no” THEN buys_computer = “no”

Classifying Large Dataset

- **Decision trees seem to be a good choice**
  - relatively faster learning speed than other classification methods
  - can be converted into simple and easy to understand classification rules
  - can be used to generate SQL queries for accessing databases
  - has comparable classification accuracy with other methods

- **Classifying data-sets with millions of examples and a few hundred even thousands attributes with reasonable speed.**

Other Classification Methods

- Bayesian Classification
- Neural Networks
- k-Nearest Neighbor Classifier
- Support Vector Machine (SVM)
- Random Forest
- Gradient Boosting
- Logistic Regression

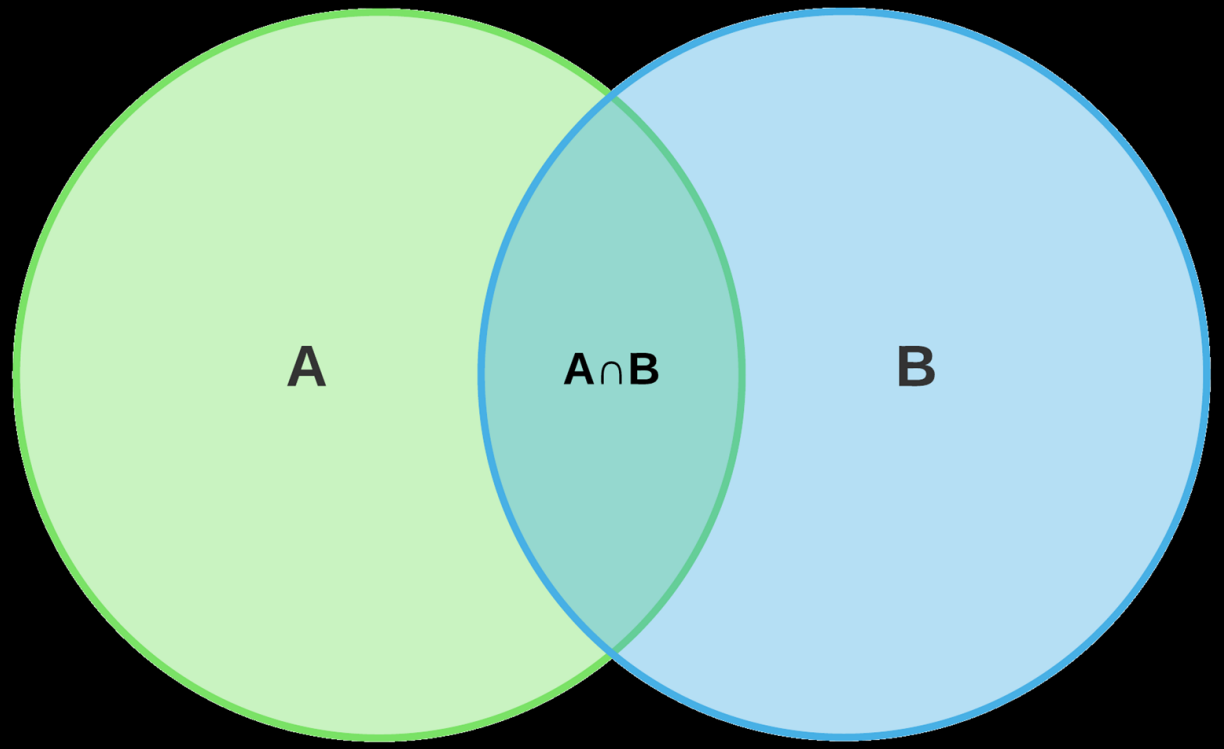
Comparing Classification Method

- **Predictive accuracy**
- **Speed and scalability**
  - time to construct the model (training time)
  - time to use the model
- **Robustness**
  - handling noise and missing values
- **Scalability**
  - efficiency in large databases (not memory resident data)
- **Interpretability:**
  - the level of understanding and insight provided by the model
- **Goodness of rules**
  - decision tree size
  - the compactness of classification rules

BAYES THEOREM

Review: Conditional Probability

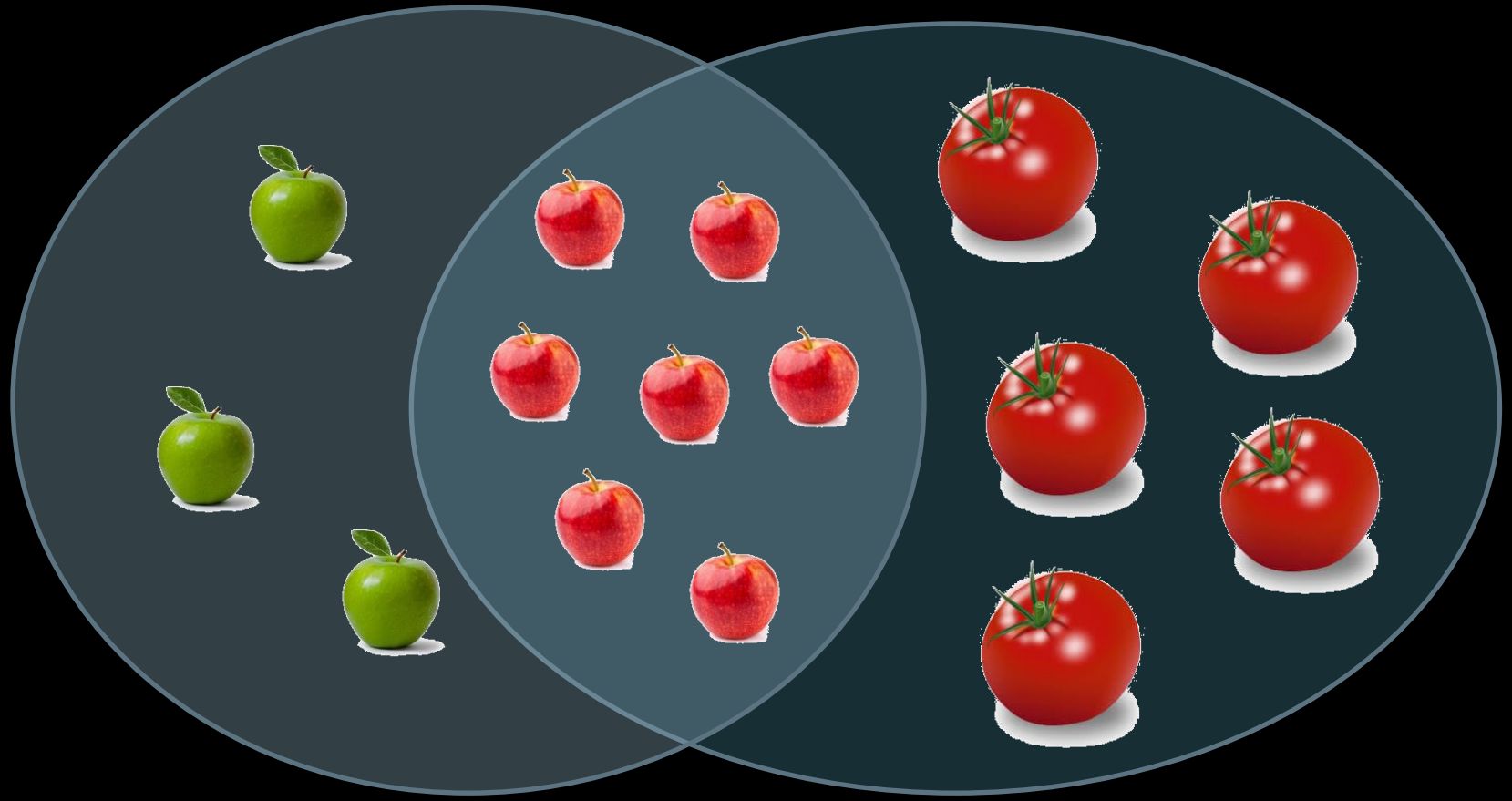
- $P(A|B) = \frac{P(A \cap B)}{P(B)}$
  - probability of A given B



Review: Conditional Probability

- $P(A|B) = \frac{P(A \cap B)}{P(B)}$

P(apple | red) = ??
P(red | apple) = ??

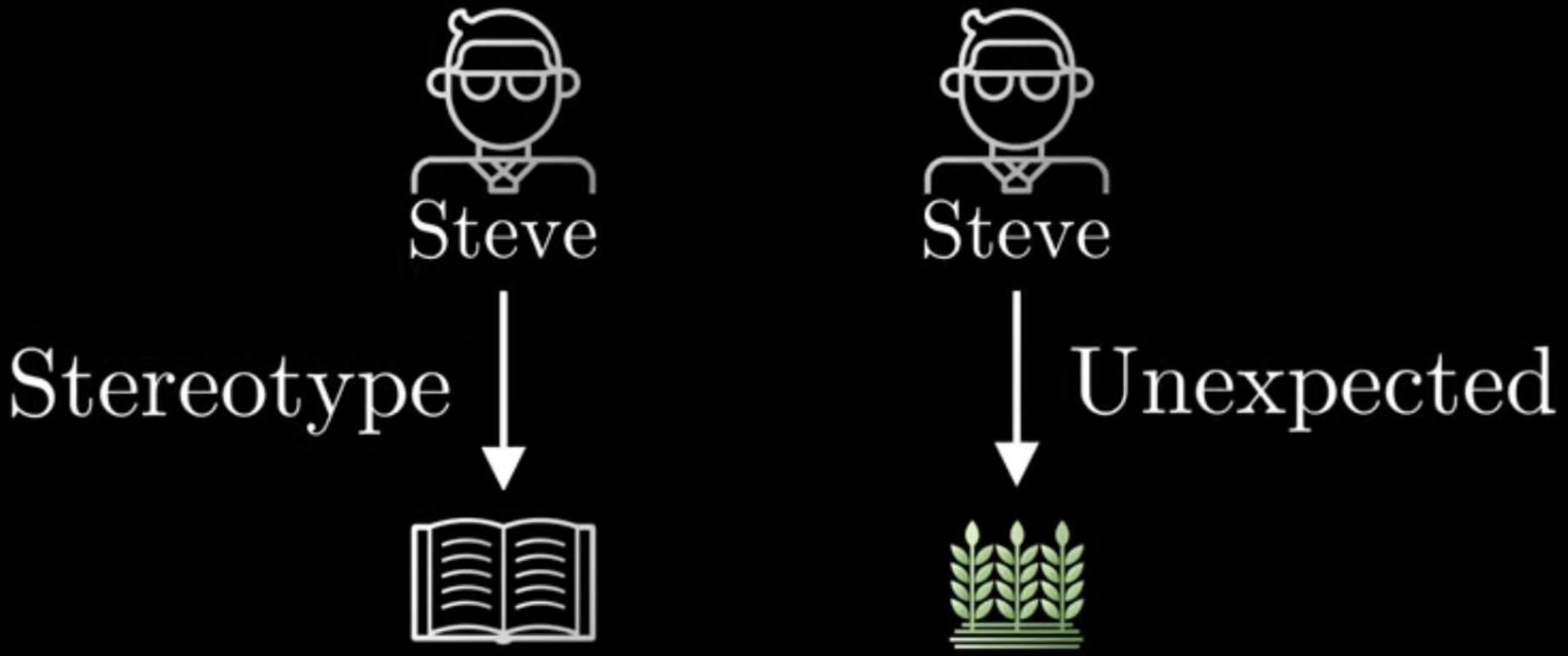


https://www.youtube.com/watch?v=HZGCoVF3YvM

Steve is very shy and withdrawn, invariably helpful but with very little interest in people or in the world of reality. A meek and tidy soul, he has a need for order and structure, and a passion for detail.

pemalu dan pendiam, lemah lembut dan rapi

Librarian or Farmer?

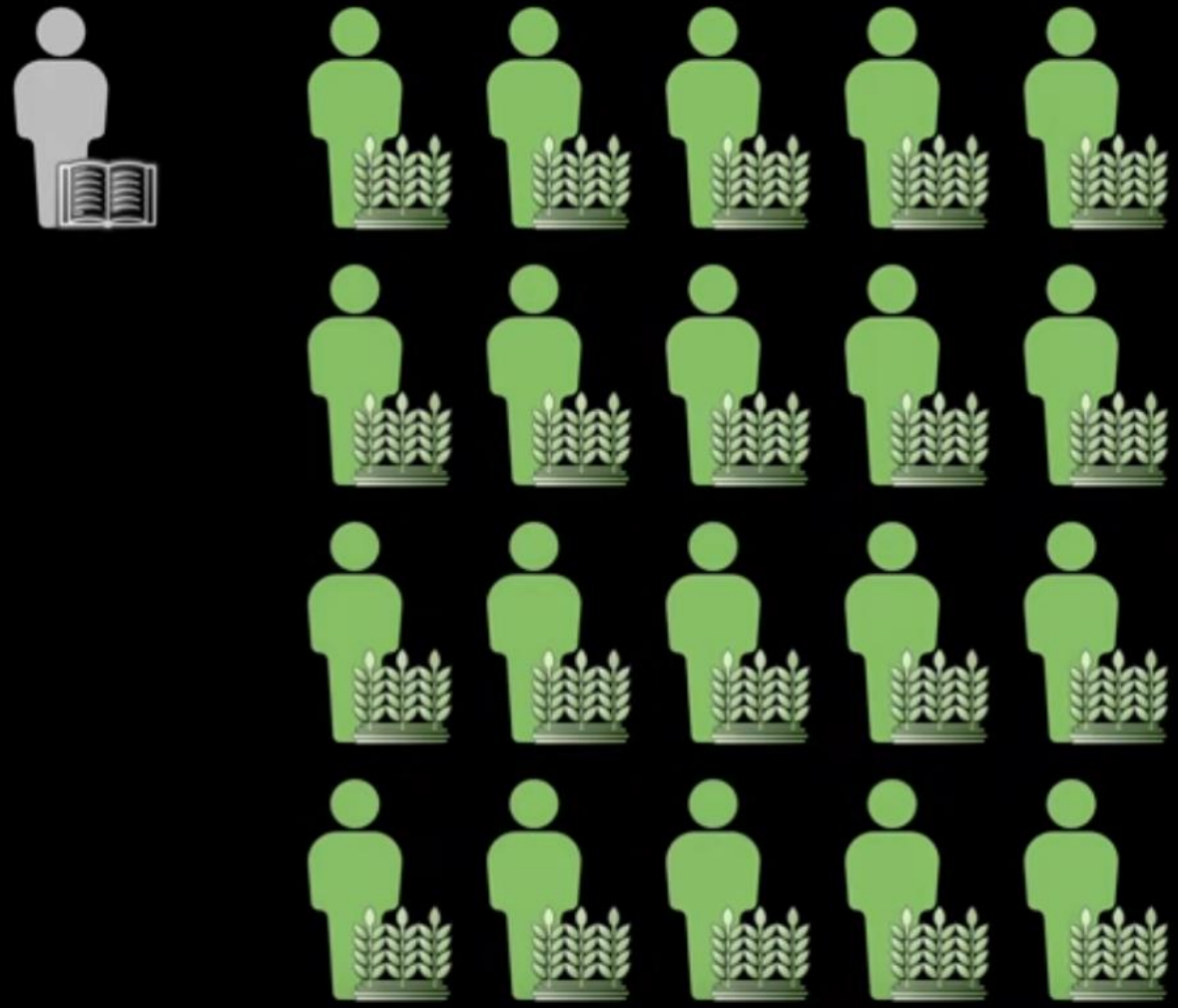


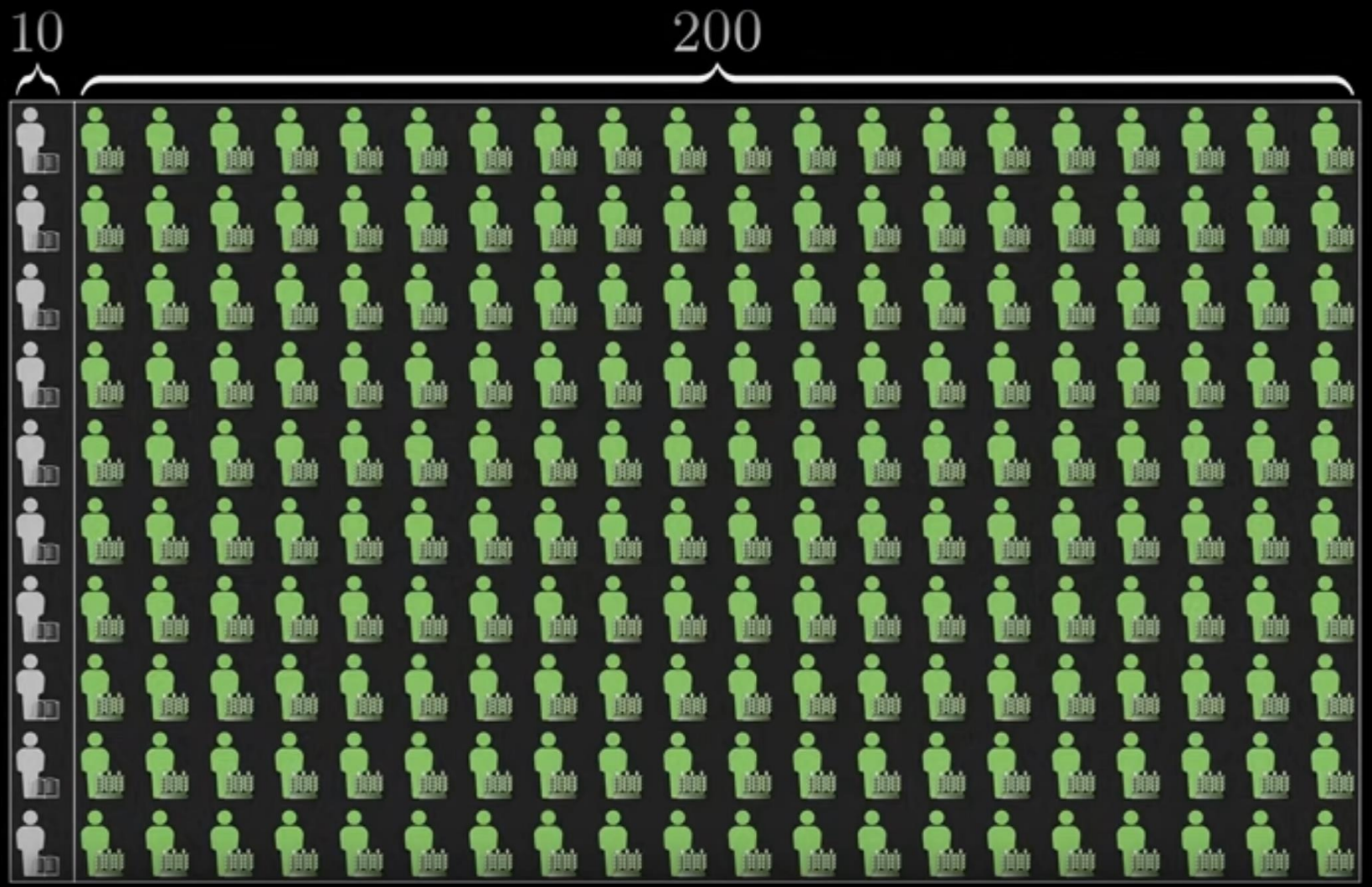
likelihood

P(description | librarian) = 40%

P(description | farmer) = 10%

Have we incorporate the ratio between librarian and farmer in our judgement?




10
200

likelihood

P(description | librarian) = 40%

P(description | farmer) = 10%

$$P(Librarian\ given\ description) = \frac{4}{4+20} \approx 16.7\%$$

Bayes theorem

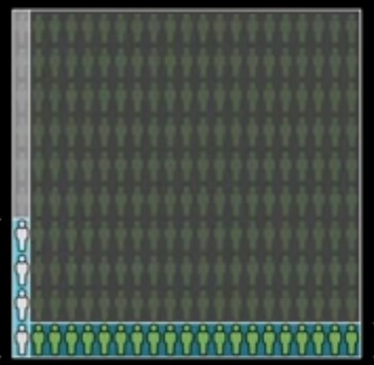
"Posterior"

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} = \frac{P(H)P(E|H)}{P(H)P(E|H) + P(\neg H)P(E|\neg H)}$$

"Prior" $\longrightarrow P(H) = 1/21$

"Likelihood"

$P(E|H) = 0.4$



$P(E|\neg H) = 0.1$

Bayes Theorem (1)

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

- **Let *X* be a data sample whose class label is unknown.**
- **Let *H* be some hypothesis, such as that the data sample *X* belongs to a specified class *C*.**
- **We want to determine *P*(*H*|X), the probability the the hypothesis *H* holds given the observed data sample *X*.**
  - Kita ingin menentukan P (H|X), probabilitas hipotesis H berlaku diberikan data yang diamati X.
- ***P*(*H*|X) is the posterior probability *or a posteriori probability*, of *H* conditioned on *X*.**

Bayes Theorem (2)

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

- **Contoh: data buah-buahan, dideskripsikan dengan warna dan bentuk.**

- **sample X = red dan bulat**
- **hipotesa H = X adalah apel**
- **P(H|X) = P(X adalah apel | X merah dan X bulat)**
  - *P*(*H*|X) merefleksikan tingkat keyakinan kita bahwa *X* adalah apel diberikan bahwa X merah dan bulat

Bayes Theorem (3)

- ***P*(*H*) is the prior probability *or a priori probability*, of *H*.**
  - The probability that any given data sample is an apple, regardless of how the data sample looks.
  - Probabilitas bahwa sampel data yang diberikan adalah apel, terlepas dari bagaimana nilai attribute pada data tersebut.

- **P(X|H): likelihood: the probability of observing the sample X, given that the hypothesis holds**
  - Probabilitas bahwa buah berwarna merah diberikan bahwa buah tersebut adalah apel.

BAYESIAN CLASSIFICATION

Bayesian Inference

Bayesian Classification

- **Bayesian classifiers are statistical classifiers.**
- **They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class.**
  - dapat memprediksi probabilitas keanggotaan kelas, seperti probabilitas bahwa sampel data tertentu milik kelas tertentu.

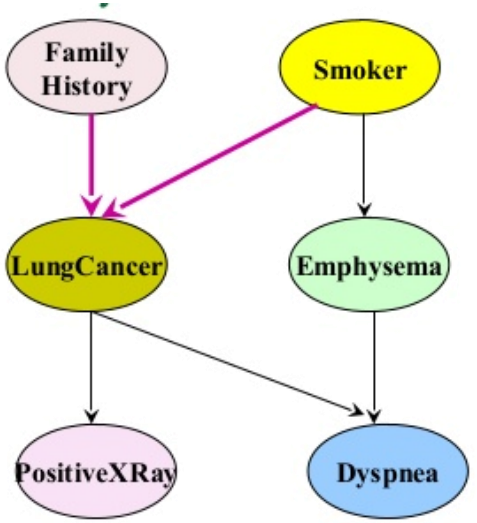
- **Bayesian classification is based on Bayes theorem.**
- **Naive Bayesian Classifier is comparable in performance with decision tree and neural network classifiers.**
  - memiliki kinerja yang sebanding dengan decision tree dan neural network classifiers.
- **have high accuracy and speed when applied to large databases.**
  - juga memiliki akurasi dan kecepatan tinggi saat diterapkan pada database besar.

Naïve Bayes Classifier

- **Example:**
  - Data have features: smoke (yes/no), lung cancer (yes/no), Dyspnea (yes/no)
  - Dyspnea: difficult or labored breathing; shortness of breath
  - Naive Bayes assume that there is no dependency between smoke and lung cancer.
  - P(Smoke and Lung Cancer | Dyspnea) =
    P(Smoke | Dyspnea) * P(Lung cancer | Dyspnea)

- **Bayesian Belief Network**

Not conditionally independent ⇒



Naïve Bayes Classifier: Training Dataset

Class:
H1: buys_computer = ‘yes’
H2: buys_computer = ‘no’

Data to be classified:
X = (age <=30,
Income = medium,
Student = yes,
Credit_rating = Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes Classifier

- **Kita ingin menghitung:**

  P(buys_computer = ‘yes’ | age <=30, Income = medium, Student = yes, Credit_rating = Fair)

  - Hipotesa H: buys_computer = ‘yes’
  - Data X: (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

  - Perlu mengetahui:
    - P(age <=30, Income = medium, Student = yes, Credit_rating = Fair| buys_computer = ‘yes’)
    - P(buys_computer = ‘yes’)
    - P(age <=30, Income = medium, Student = yes, Credit_rating = Fair)

Naive Bayes Theorem for Classification

- **We calculate:**

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

$$P(\neg H|\mathbf{X}) = \frac{P(\mathbf{X}|\neg H)P(\neg H)}{P(\mathbf{X})}$$

- **Compare which hypothesis is more probable. Which one is higher?**

$$P(H|\mathbf{X}) \quad \text{vs} \quad P(\neg H|\mathbf{X})$$

$$\frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} \quad \text{vs} \quad \frac{P(\mathbf{X}|\neg H)P(\neg H)}{P(\mathbf{X})}$$

$$P(\mathbf{X}|H)P(H) \quad \text{vs} \quad P(\mathbf{X}|\neg H)P(\neg H)$$

88

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

$$= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution
- Example:

$$P(\text{age <= 30 , income = medium, student = yes, credit\_rating = fair} \mid \text{buy\_computer=yes})$$

$$= P(\text{age <= 30}|\text{buy\_computer=yes}) \times P(\text{income = medium}|\text{buy\_computer=yes}) \times P(\text{student = yes}|\text{buy\_computer=yes}) \times P(\text{credit\_rating = fair}|\text{buy\_computer=yes})$$

Naïve Bayes Classifier

- **We need to calculate:**

$$P(\text{age <= 30 , income = medium, student = yes, credit\_rating = fair buy\_computer=yes})$$

$$=$$

$$P(\text{age <= 30} \mid \text{buy\_computer=yes}) \times P(\text{income = medium} \mid \text{buy\_computer=yes}) \times P(\text{student = yes} \mid \text{buy\_computer=yes}) \times P(\text{credit\_rating = fair} \mid \text{buy\_computer=yes})$$

How to calculate $P(x_k|C_i)$: categorical

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$P(A_k = x_k|C_i) = \frac{\#\text{ of tuples of } C_i \text{ having value } x_k \text{ for } A_k}{\#\text{ of tuples of } C_i}$$

P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222

Naïve Bayes Classifier: An Example

- **$P(C_i)$:**

  P(buys_computer = “yes”) = 9/14 = 0.643

  P(buys_computer = “no”) = 5/14= 0.357

- **Compute $P(X|C_i)$ for each class**

  P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222

  P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6

  P(income = “medium” | buys_computer = “yes”) = 4/9 = 0.444

  P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4

  P(student = “yes” | buys_computer = “yes) = 6/9 = 0.667

  P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2

  P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667

  P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes Classifier: An Example

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

$P(X|C_i)$:

$P(X|buys\_computer = \text{“yes”}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$

$P(X|buys\_computer = \text{“no”}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$

$P(X|C_i)*P(C_i)$:

$P(X|buys\_computer = \text{“yes”}) * P(buys\_computer = \text{“yes”}) = 0.028$

$P(X|buys\_computer = \text{“no”}) * P(buys\_computer = \text{“no”}) = 0.007$

Therefore, X belongs to class (“buys_computer = yes”)

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

How to calculate $P(x_k|C_i)$

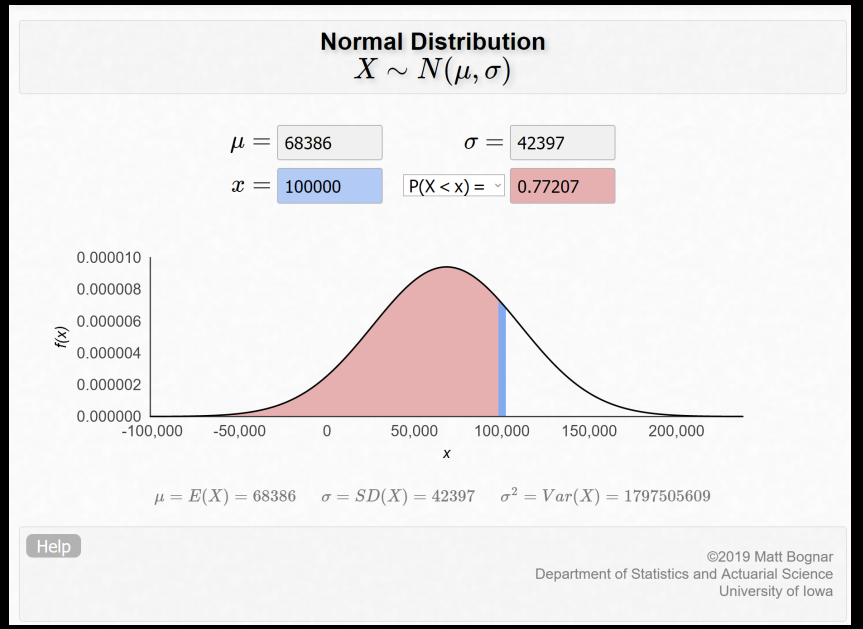
- **If A_k is continuous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian/Normal distribution $g(x, \mu, \sigma)$ with a mean μ and standard deviation σ**

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\mathrm{P}(A_k = x_k|C_i) = g(x_k, \mu_k, \sigma_{C_i})$$

- **Assumption: Normal Distribution!**

- **Alternative method:**
  - transform data into Normal distribution
  - discretization



How to calculate $P(x_k|C_i)$: numerical

City	Gender	Income	Illness
Dallas	Male	40,367	No
Dallas	Female	41,524	Yes
Dallas	Male	46,373	Yes
New York City	Male	98,096	No
New York City	Female	102,089	No
New York City	Female	100,662	No
New York City	Male	117,263	Yes
Dallas	Male	56,645	No

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(x_k|C_i) = g(x_k, \mu_k, \sigma_{C_i})$$

Formula in Excel

=NORM.DIST(x, mean, standard_dev, FALSE)

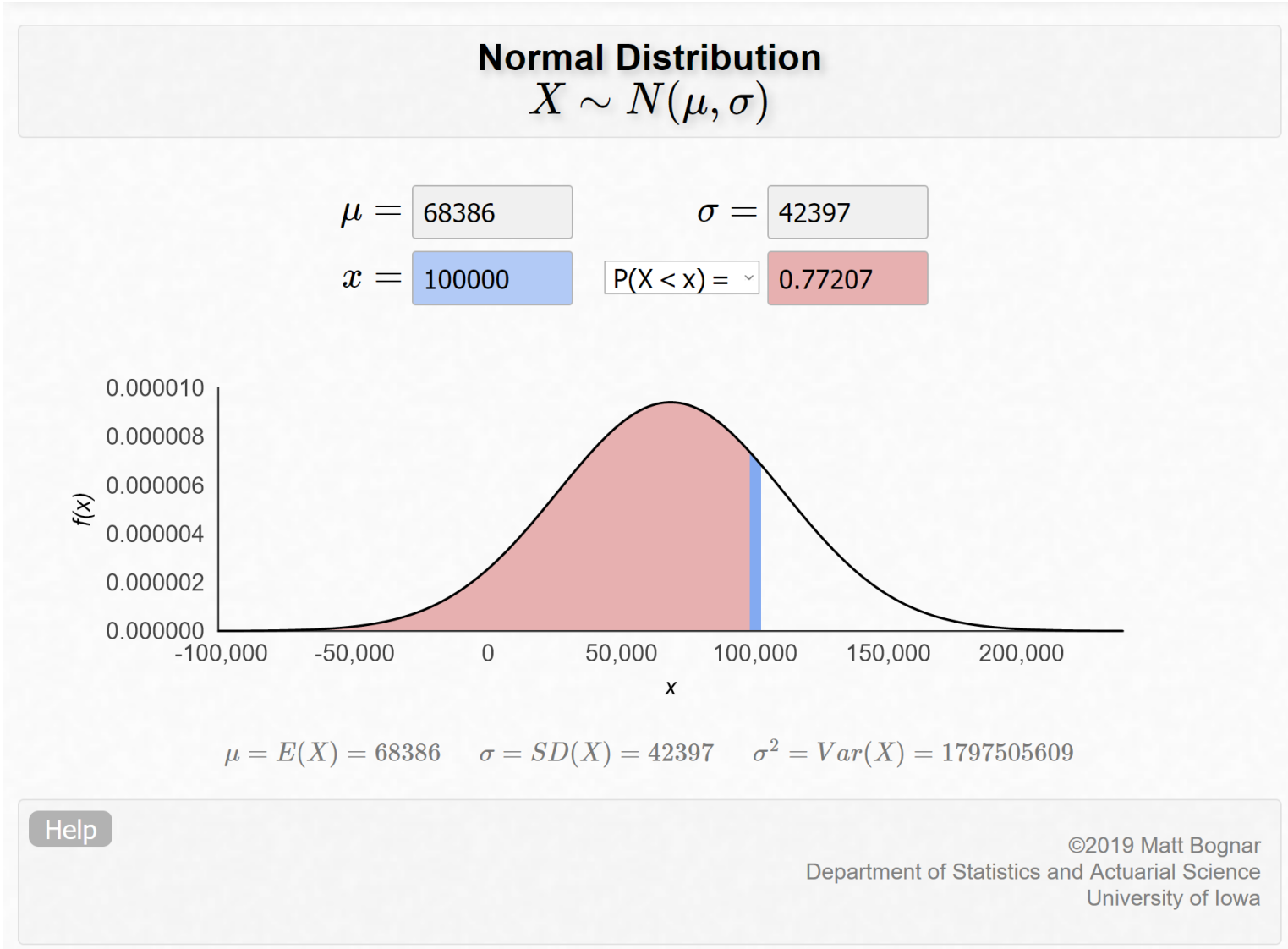
Income	Illness
41,524	Yes
46,373	Yes
117,263	Yes
40,367	No
98,096	No
102,089	No
100,662	No
56,645	No

Frequency / Likelihood Table		Income Mean	Income StDev
Illness	Yes	68,386.67	42,397.53
	No	79,571.80	28,972.49

P(income = 100000 | Illness=Yes)	0.0000071259
P(income = 100000 | Illness=No)	0.0000107391

https://medium.com/analytics-vidhya/use-naive-bayes-algorithm-for-categorical-and-numerical-data-classification-935d90ab273f

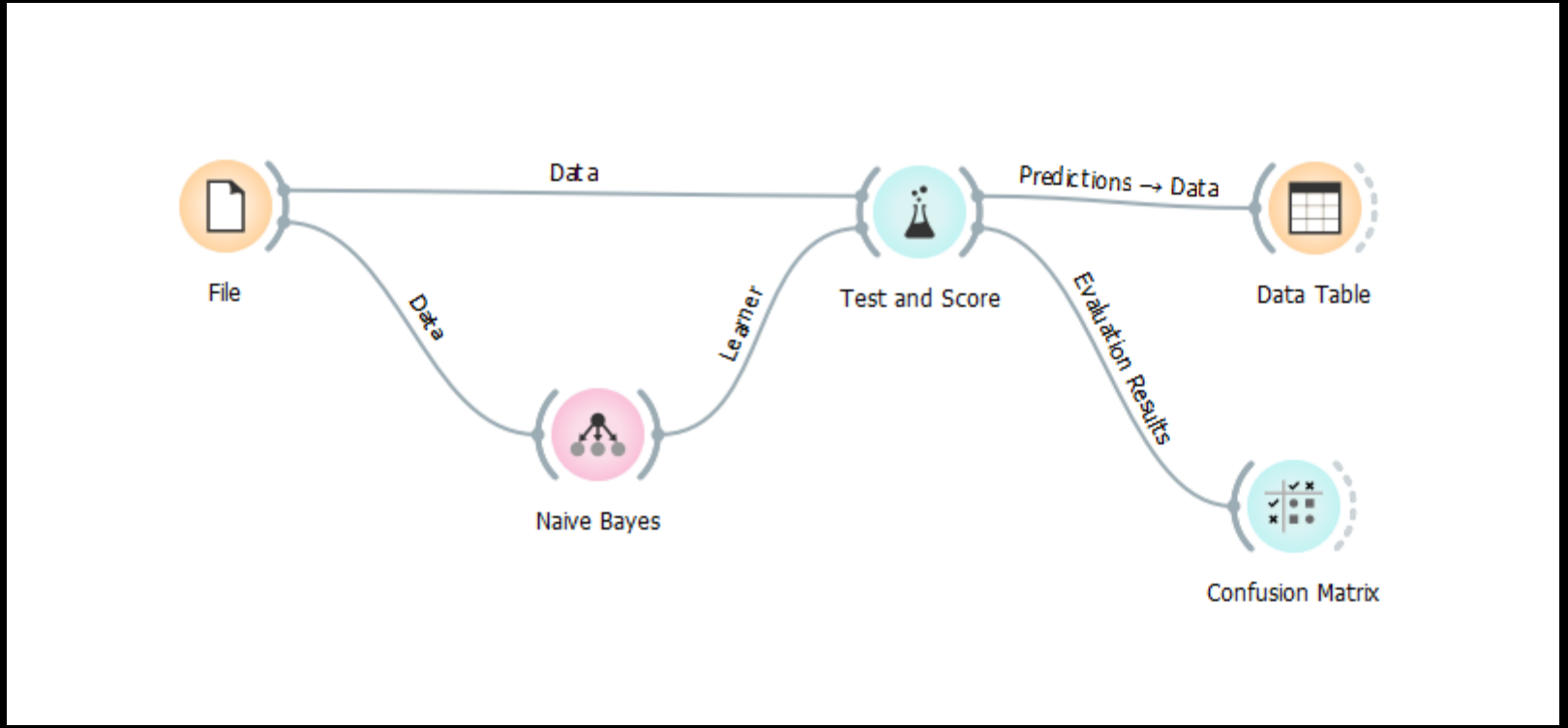
Normal Distribution



https://homepage.divms.uiowa.edu/~mbognar/applets/normal.html

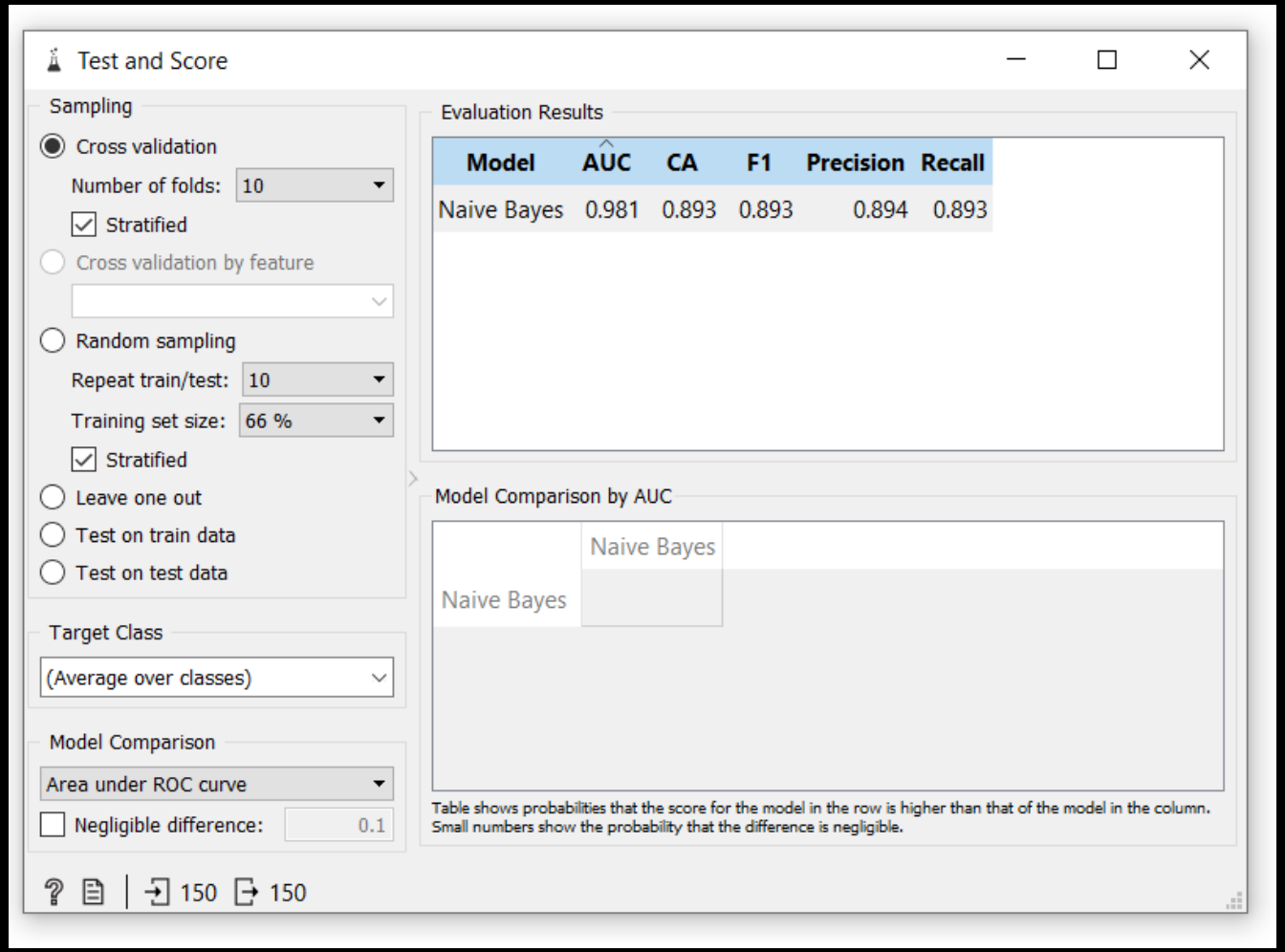
Modeling dengan Orange – Naïve Bayes

- **Gunakan data Iris**



Modeling dengan Orange – Naïve Bayes

- **Gunakan data Iris**



Modeling dengan Orange – Naïve Bayes

- **Gunakan data Iris**

Data Table

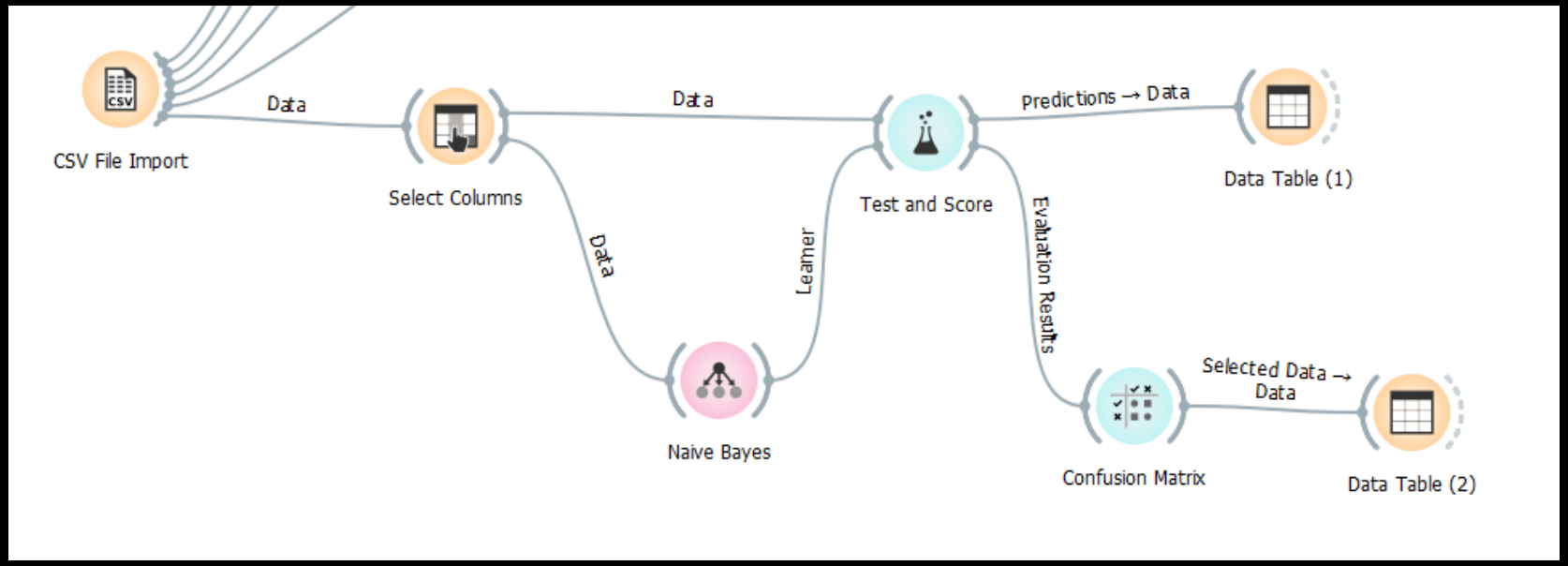
Variables
- [x] Show variable labels (if present)
- [ ] Visualize numeric values
- [x] Color by instance classes

Selection
- [x] Select full rows

	iris	Naive Bayes	Naive Bayes (Iris-setosa)	Naive Bayes (Iris-versicolor)	Naive Bayes (Iris-virginica)	Fold	sepal length	sepal width	petal length	petal width
1	Iris-setosa	Iris-setosa	0.999404	0.000567456	2.83728e-05	1	4.8	3.4	1.6	0.2
2	Iris-setosa	Iris-setosa	0.996428	0.000934323	0.00263809	1	5.8	4.0	1.2	0.2
3	Iris-setosa	Iris-setosa	0.976329	0.0232829	0.000388048	1	5.0	3.0	1.6	0.2
4	Iris-setosa	Iris-setosa	0.99989	7.21207e-05	3.81815e-05	1	5.5	4.2	1.4	0.2
5	Iris-setosa	Iris-setosa	0.999905	5.67741e-05	3.78494e-05	1	5.0	3.2	1.2	0.2
6	Iris-versicolor	Iris-virginica	0.000168511	0.370724	0.629107	1	6.9	3.1	4.9	1.5
7	Iris-versicolor	Iris-versicolor	0.00190142	0.994588	0.00351031	1	5.2	2.7	3.9	1.4
8	Iris-versicolor	Iris-virginica	0.000168511	0.370724	0.629107	1	6.7	3.1	4.4	1.4
9	Iris-versicolor	Iris-versicolor	9.80873e-05	0.990486	0.00941638	1	6.3	2.3	4.4	1.3
10	Iris-versicolor	Iris-versicolor	0.00361272	0.994942	0.00144509	1	6.2	2.9	4.3	1.3
11	Iris-virginica	Iris-virginica	6.53111e-05	0.00111029	0.998824	1	6.3	3.3	6.0	2.5
12	Iris-virginica	Iris-virginica	1.96352e-05	0.0235622	0.976418	1	7.3	2.9	6.3	1.8
13	Iris-virginica	Iris-virginica	3.73492e-05	0.000373492	0.999589	1	6.7	3.3	5.7	2.1
14	Iris-virginica	Iris-virginica	1.03725e-05	0.000622349	0.999367	1	7.4	2.8	6.1	1.9
15	Iris-virginica	Iris-virginica	3.73492e-05	0.000373492	0.999589	1	6.7	3.3	5.7	2.5
16	Iris-setosa	Iris-setosa	0.999388	0.000437407	0.000174963	2	4.3	3.0	1.1	0.1
17	Iris-setosa	Iris-setosa	0.99749	0.0023586	0.000151192	2	5.7	4.4	1.5	0.4
18	Iris-setosa	Iris-setosa	0.996942	0.002851	0.000206594	2	5.4	3.4	1.7	0.2
19	Iris-setosa	Iris-setosa	0.994011	0.00583697	0.000152269	2	4.7	3.2	1.6	0.2
20	Iris-setosa	Iris-setosa	0.975943	0.0236924	0.000364498	2	4.8	3.0	1.4	0.3
21	Iris-versicolor	Iris-versicolor	2.94803e-05	0.759412	0.240559	2	6.0	2.9	4.5	1.5
22	Iris-versicolor	Iris-virginica	0.00145022	0.386458	0.612092	2	6.0	3.4	4.5	1.6
23	Iris-versicolor	Iris-versicolor	0.0212696	0.978403	0.000327225	2	5.8	2.6	4.0	1.2
24	Iris-versicolor	Iris-versicolor	0.0212696	0.978403	0.000327225	2	5.6	2.7	4.2	1.3

Self Study

- **Buat model Naïve Bayes untuk data credit card**



Model Evaluation

Model Evaluation and Selection

- **Evaluation metrics: How can we measure accuracy? Other metrics to consider?**
- **Use testing set of class-labeled tuples instead of training set when assessing accuracy**

- **Methods for estimating a classifier's accuracy:**
  - Holdout method
  - Cross-validation
  - Leave one out

103

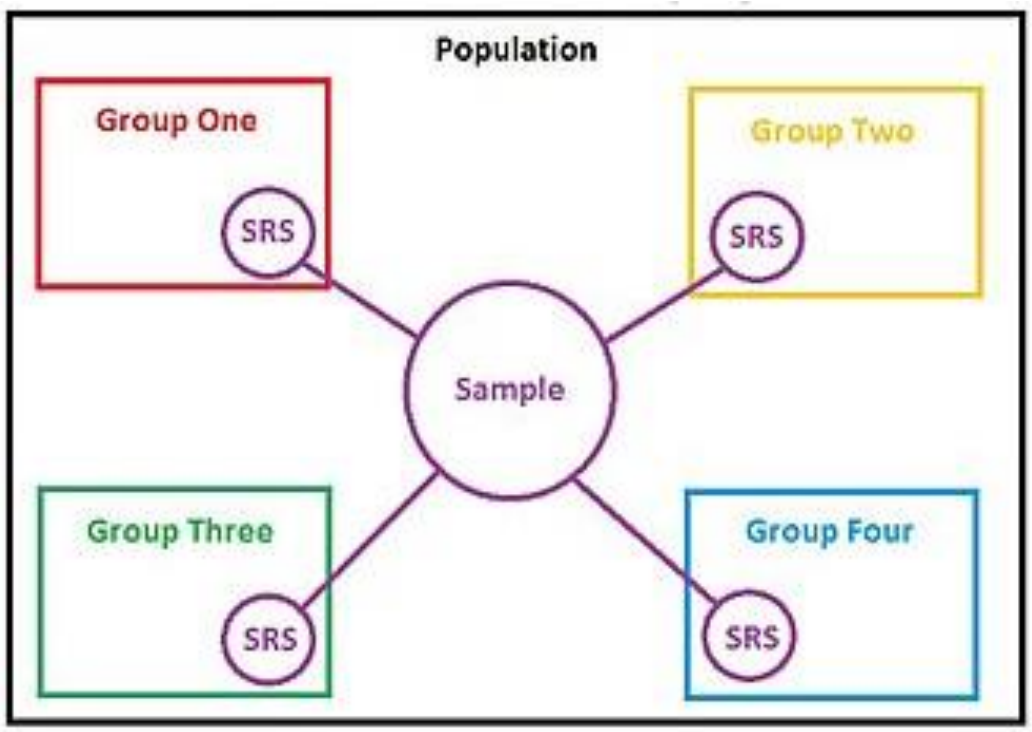
Validation Strategies

- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling vs Stratified Random Sampling
- **Repeat holdout k times, accuracy = avg. of the accuracies obtained**

Sampling Methods

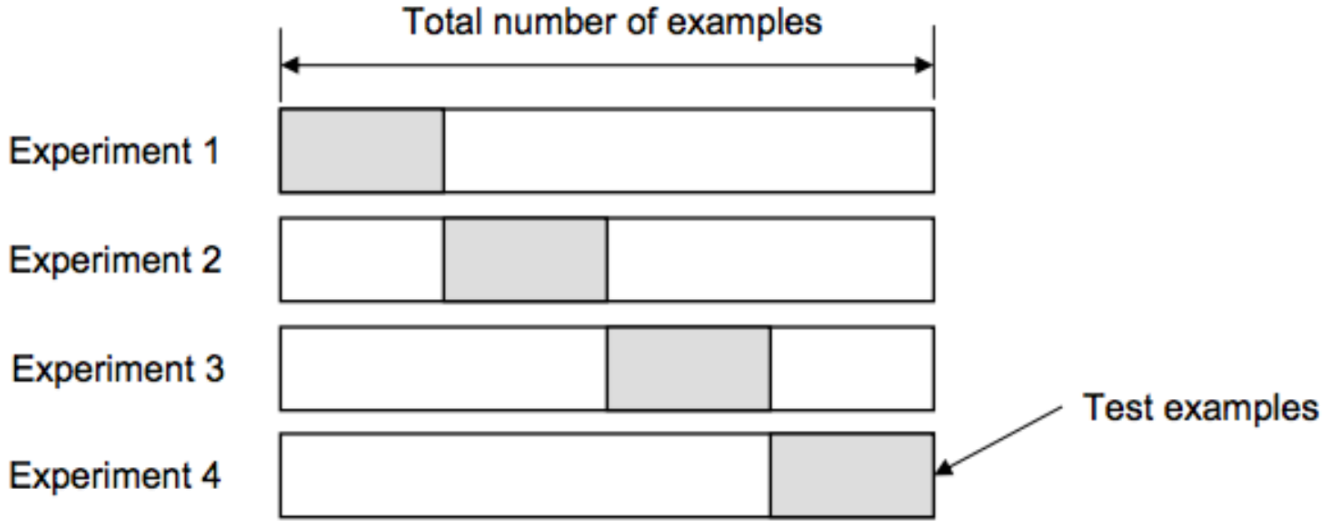
- **Simple Random Sampling**

- **Stratified Random Sampling**



Validation Strategies

- **Cross-validation (*k*-fold, where k = 10 is most popular)**
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - **Stratified cross-validation**: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data



Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

Classifier Evaluation Metrics: Accuracy, Error Rate

- **Classifier Accuracy, or recognition rate: percentage of test set tuples that are correctly classified**
  - Accuracy = (TP + TN)/All

- **Error rate: 1 – accuracy, or**
  - Error rate = (FP + FN)/All

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

Classifier Evaluation Metrics: Class Imbalance

- **Class Imbalance Problem:**
  - One class may be rare, e.g., fraud, or HIV-positive
  - Significant majority of the negative class and minority of the positive class
- **Sensitivity a.k.a. Recall: True Positive recognition rate**
  - Sensitivity = TP/P
- **Specificity: True Negative recognition rate**
  - Specificity = TN/N

A\P	C	¬C	
C	**0**	**1,000**	**1,000**
¬C	**0**	**99,000**	**99,000**
	0	**100,000**	**100,000**

Classifier Evaluation Metrics: Asymmetrical cost of False Prediction

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P’	N’	All

A\P	C	¬C	
C	0	1,000	1,000
¬C	0	99,000	99,000
	0	100,000	100,000

- **Are the cost of FN equal to FP?**
- **Between FN and FP, which one is more costly?**

- **Kesalahan apa dalam deteksi Covid yang menyebabkan biaya lebih tinggi? Kesalahan karena False Positive (FP) atau False Negative (FN)? Jelaskan dengan singkat.**

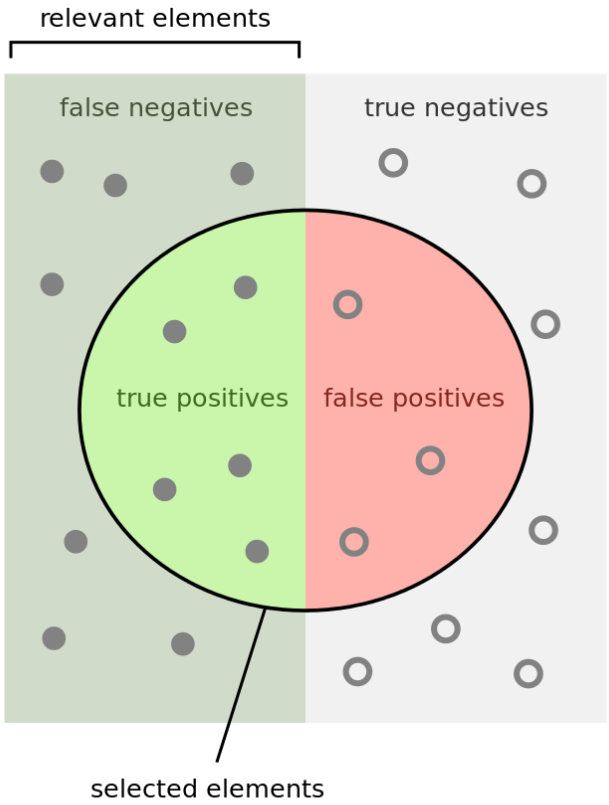
Classifier Evaluation Metrics: Precision vs Recall

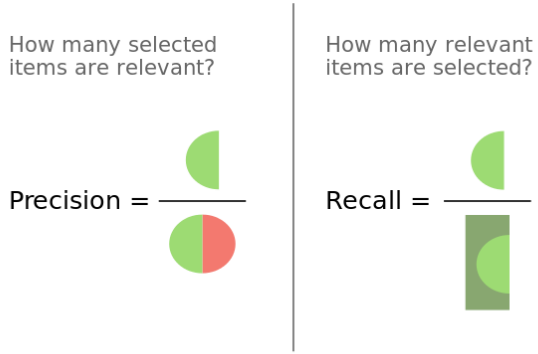
- **Precision: exactness – what % of tuples that the classifier labeled as positive are actually positive**

$$precision = \frac{TP}{TP + FP}$$

- **Recall: completeness – what % of positive tuples did the classifier label as positive? a.k.a. Sensitivity**
  - Perfect score is 1.0

$$recall = \frac{TP}{TP + FN}$$





Classifier Evaluation Metrics: F1 score

- Precision helps when the costs of false positives are high.
- Recall helps when the cost of false negatives is high

- F1 is an overall measure of a model's accuracy that combines precision and recall
- F measure (F1 or F-score): harmonic mean of precision and recall
  - $F = \frac{2 \times precision \times recall}{precision+recall}$

Classifier Evaluation Metrics: Example

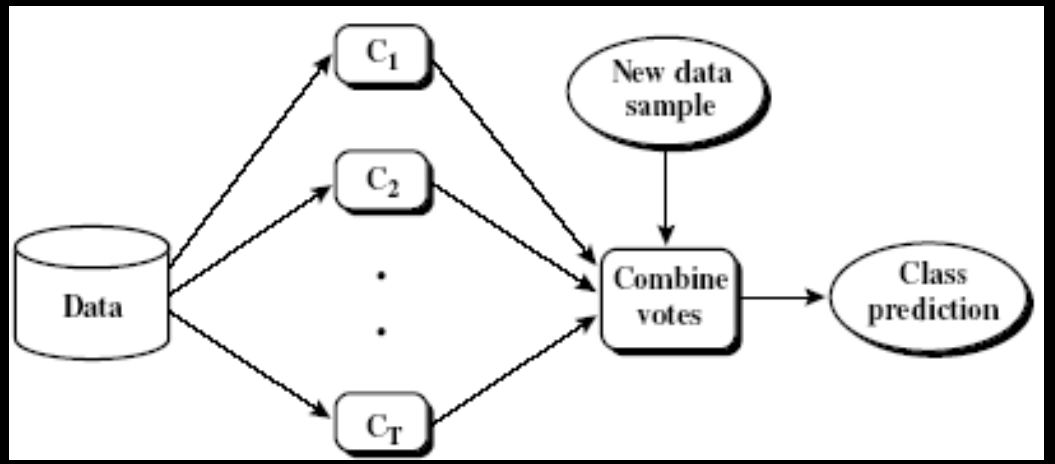
Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*)
cancer = no	**140**	**9560**	9700	98.56 (*specificity*)
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%
- F1 score = ??

Ensemble Methods

Ensemble Methods: Increasing the Accuracy

- **Ensemble methods**
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, …, M_k, with the aim of creating an improved model M*
- **Popular ensemble methods**
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers
  - Ensemble: combining a set of heterogeneous classifiers



119

Bagging: Boostrap Aggregation

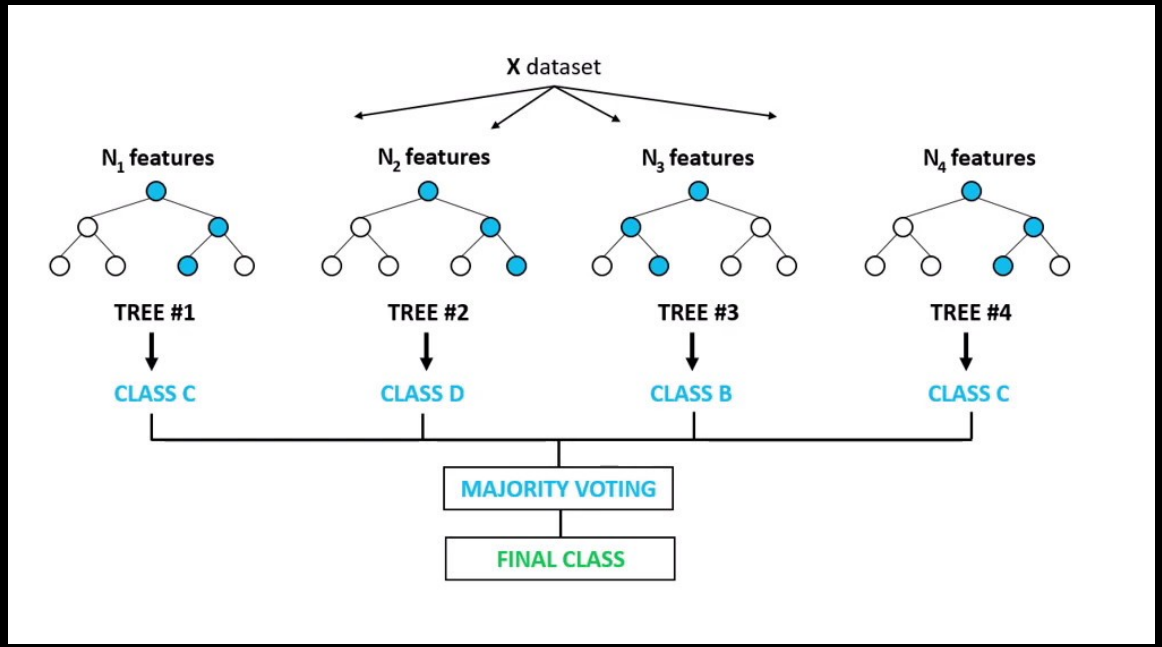
- **Analogy: Diagnosis based on multiple doctors' majority vote**
- **Training**
  - Given a set D of *d* tuples, at each iteration *i*, a training set D_i of *d* tuples is sampled with replacement from D (i.e., bootstrap)
  - A classifier model M_i is learned for each training set D_i
- **Classification: classify an unknown sample X**
  - Each classifier M_i returns its class prediction
  - The bagged classifier M* counts the votes and assigns the class with the most votes to **X**
- **Prediction: can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple**
- **Accuracy**
  - Often significantly better than a single classifier derived from D
  - For noise data: not considerably worse, more robust
  - Proved improved accuracy in prediction

Boosting

- **Analogy: Consult several doctors, based on a combination of weighted diagnose weight assigned based on the previous diagnosis accuracy**
- **How boosting works?**
  - **Weights** are assigned to each training tuple
  - A series of k classifiers is iteratively learned
  - After a classifier M_i is learned, the weights are updated to allow the subsequent classifier, M_{i+1}, to **pay more attention to the training tuples that were misclassified** by M_i
  - The final **M* combines the votes** of each individual classifier, where the weight of each classifier's vote is a function of its accuracy
- **Boosting algorithm can be extended for numeric prediction**
- **Comparing with bagging: Boosting tends to have greater accuracy, but it also risks overfitting the model to misclassified data**

Random Forest (Breiman 2001)

- **Random Forest:**
  - Each classifier in the ensemble is a *decision tree* classifier and is generated using a random selection of attributes at each node to determine the split
  - During classification, each tree votes and the most popular class is returned
- **Insensitive to the number of attributes selected for consideration at each split, and faster than bagging or boosting**



Class Imbalance and Multiclass

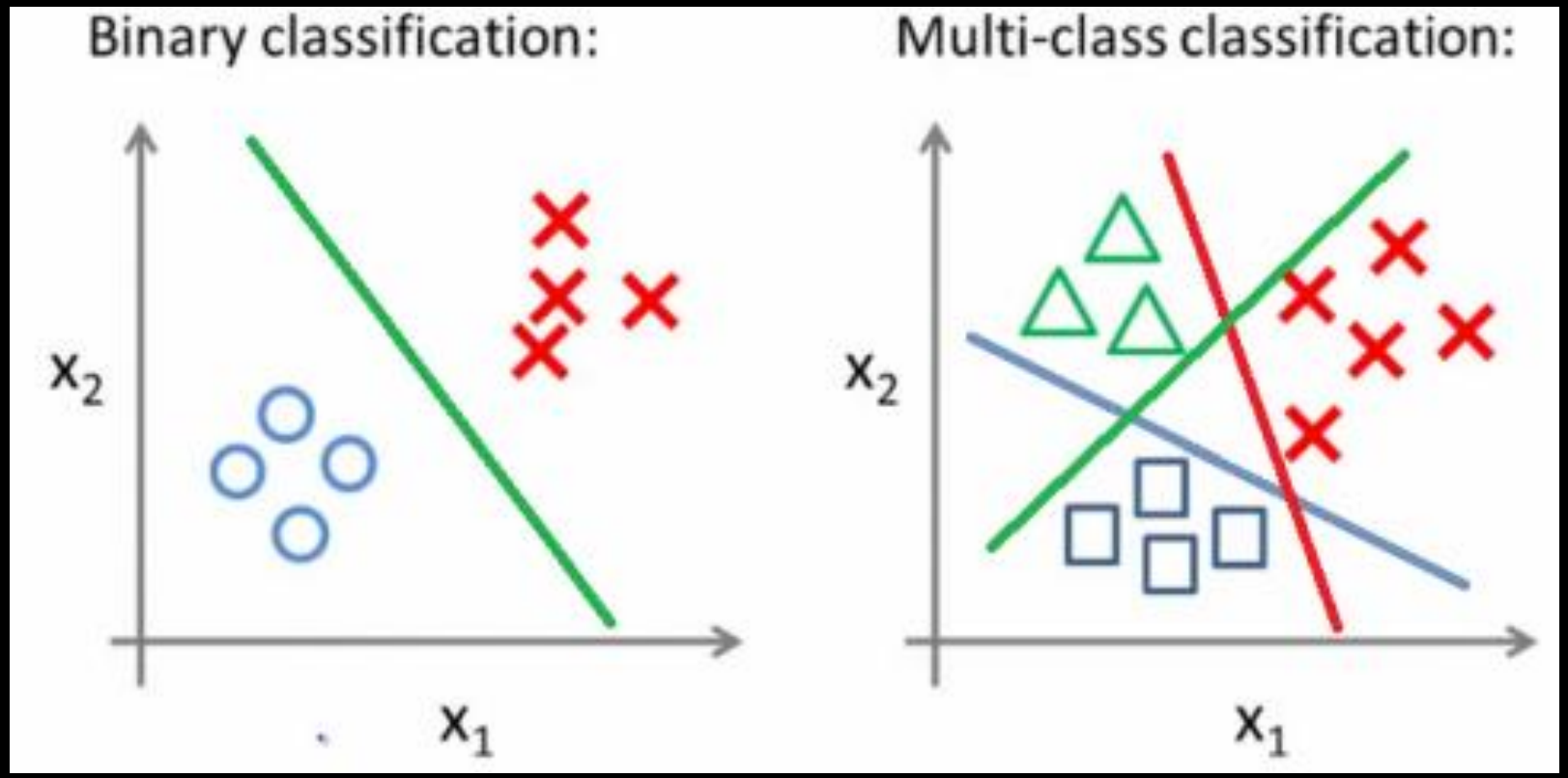
Classification of Class-Imbalanced Data Sets

- **Class-imbalance problem: Rare positive example but numerous negative ones, e.g., medical diagnosis, fraud, oil-spill, fault, etc.**
- **Traditional methods assume a balanced distribution of classes and equal error costs: not suitable for class-imbalanced data**
- **Typical methods for imbalance data in 2-class classification:**
  - **Oversampling**: re-sampling of data from positive class
    - SMOTE: Synthetic Minority Oversampling Technique
  - **Under-sampling**: randomly eliminate tuples from negative class

- **Still difficult for class imbalance problem on multiclass tasks**

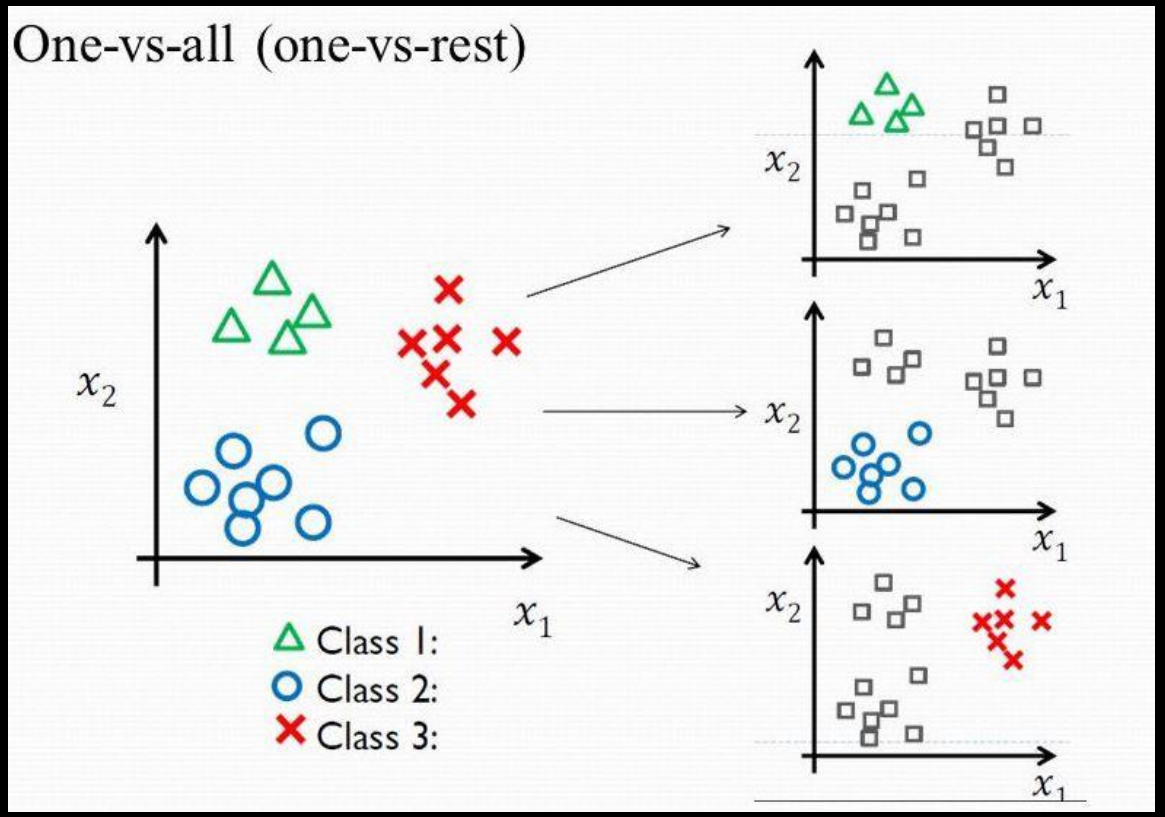
Multiclass Classification

- **Classification involving more than two classes (i.e., > 2 Classes)**



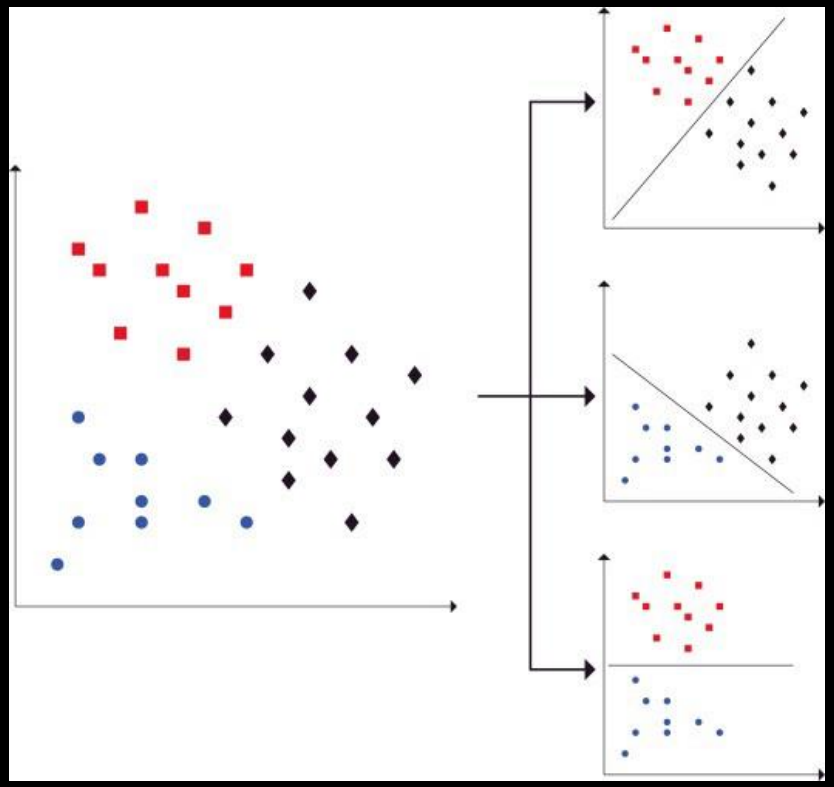
Multiclass Classification

- **One-vs.-all (OVA): Learn a classifier one at a time**
  - Given m classes, train m classifiers: one for each class
  - Classifier j: treat tuples in class j as *positive* & all others as *negative*
  - To classify a tuple **X**, the set of classifiers vote as an ensemble

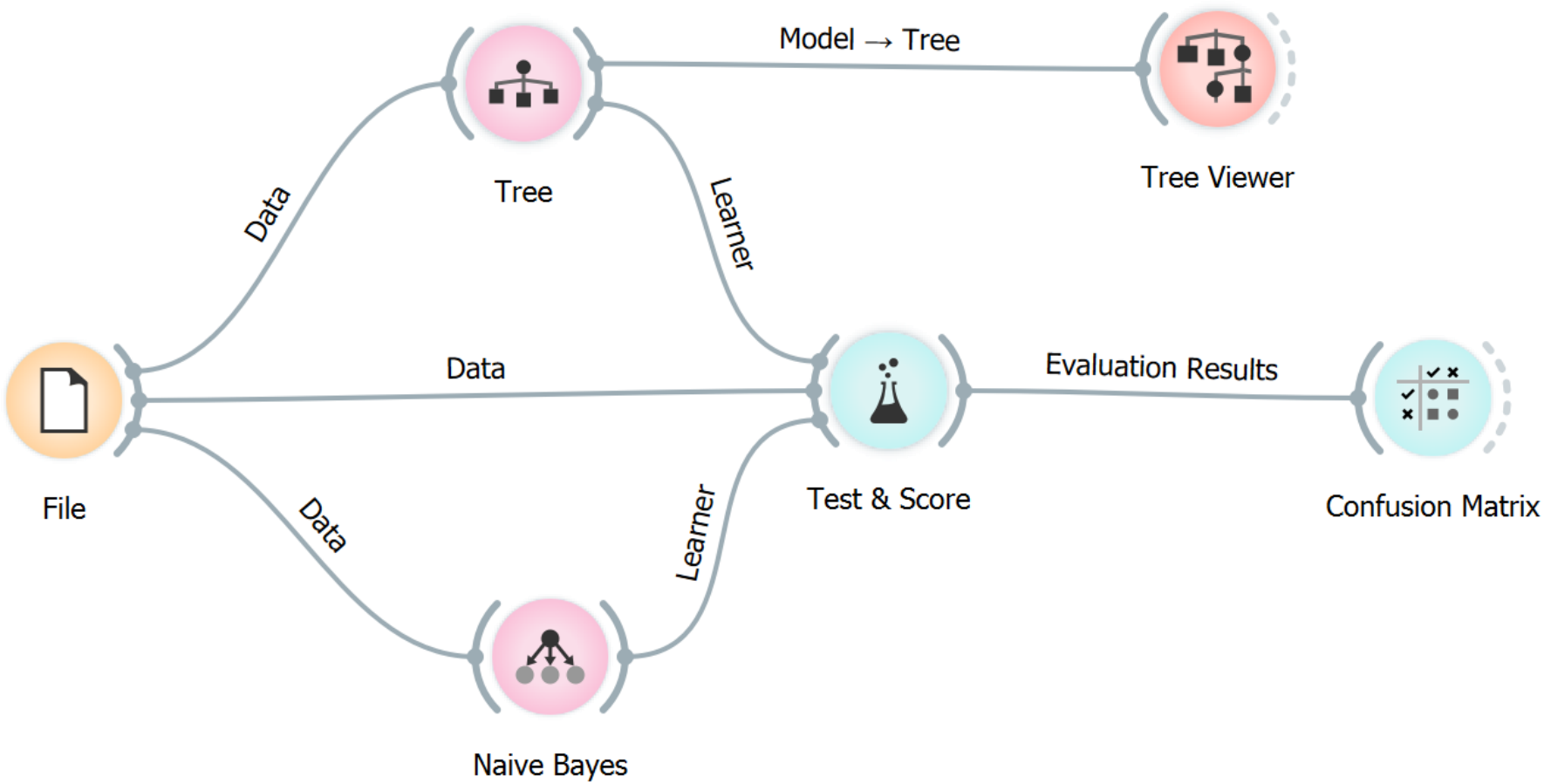


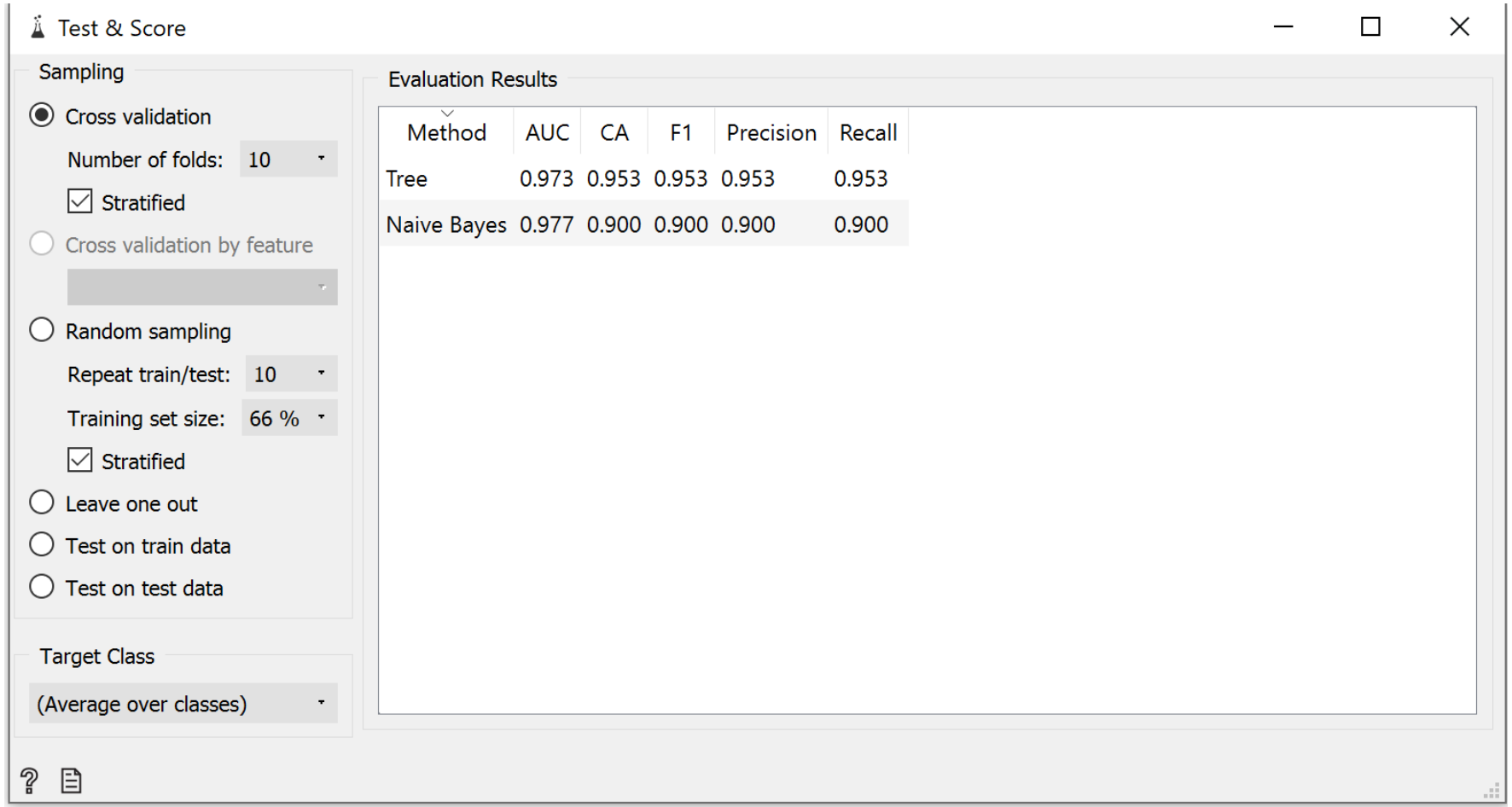
Multiclass Classification

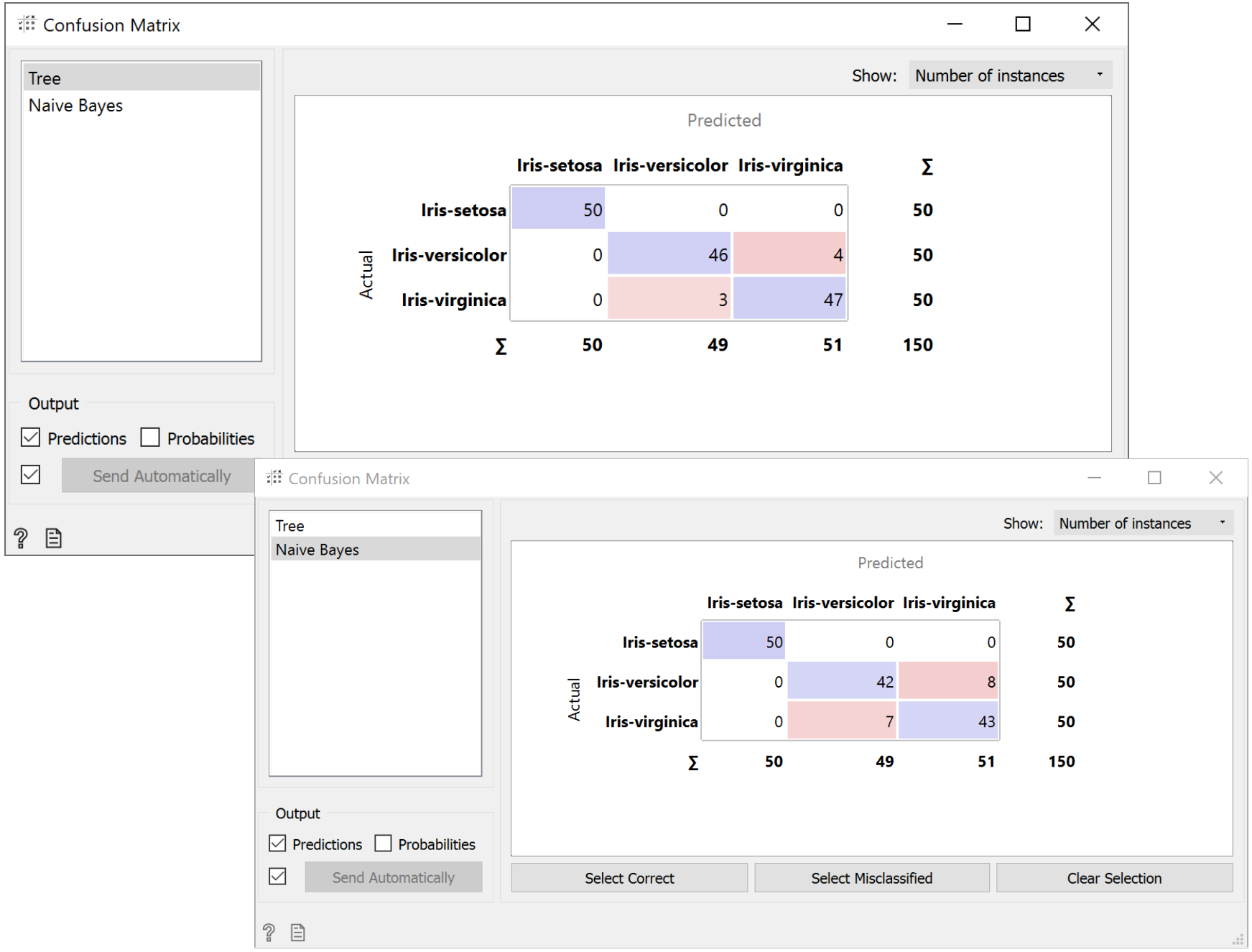
- **One vs One: Learn a classifier for each pair of classes**
  - Given m classes, construct m(m-1)/2 binary classifiers
  - A classifier is trained using tuples of the two classes
  - To classify a tuple X, each classifier votes. X is assigned to the class with maximal vote
- **Comparison**
  - All-vs.-all tends to be superior to one-vs.-all
  - Problem: Binary classifier is sensitive to errors, and errors affect vote count



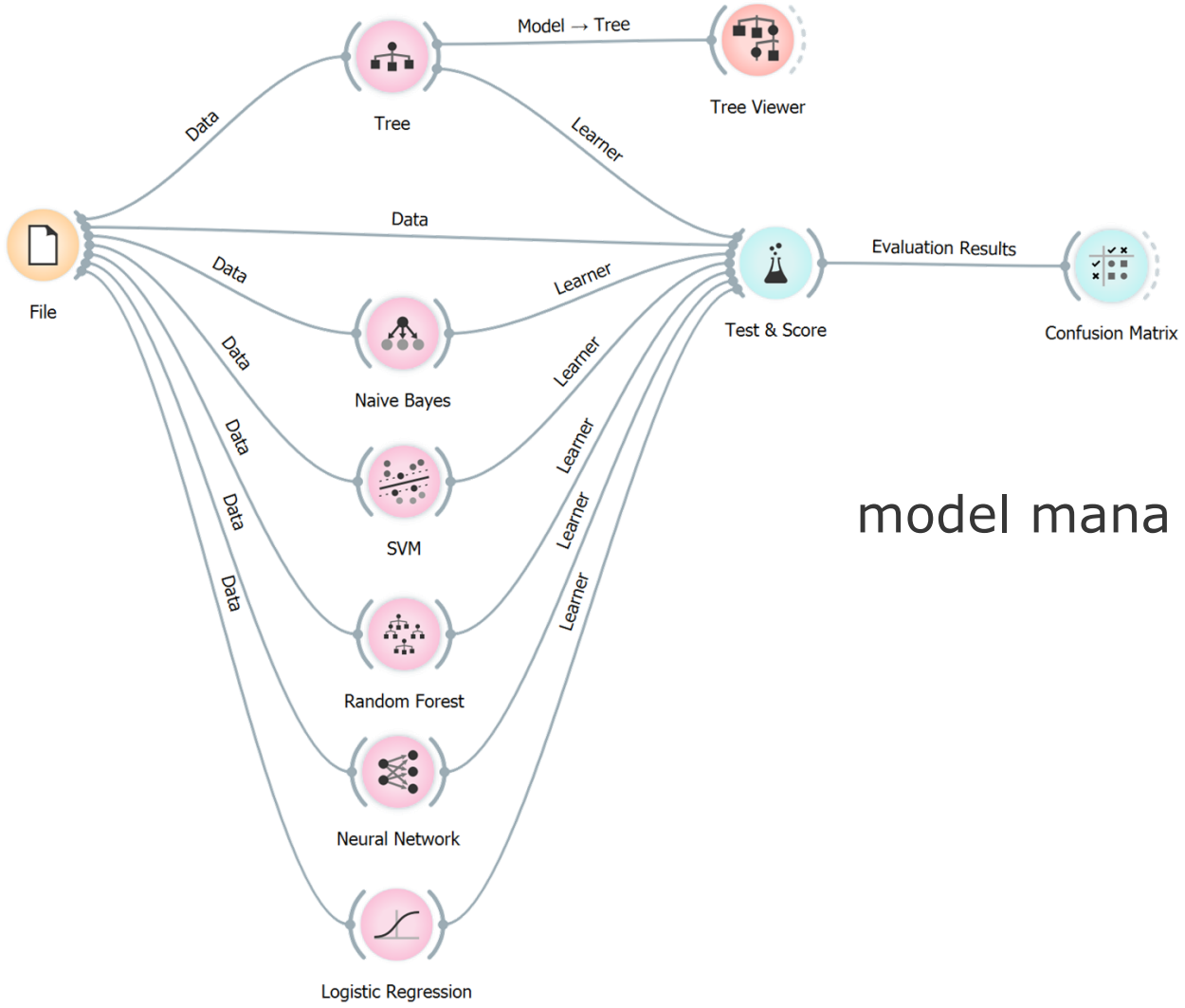
PRAKTEK DENGAN ORANGE


Model → Tree
Tree
Tree Viewer
Data
Learner
Data
Evaluation Results
File
Test & Score
Confusion Matrix
Data
Learner
Naive Bayes


Test & Score
Sampling
Cross validation
Number of folds: 10
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 66 %
Stratified
Leave one out
Test on train data
Test on test data
Target Class
(Average over classes)
Evaluation Results
Method AUC CA F1 Precision Recall
Tree 0.973 0.953 0.953 0.953 0.953
Naive Bayes 0.977 0.900 0.900 0.900 0.900

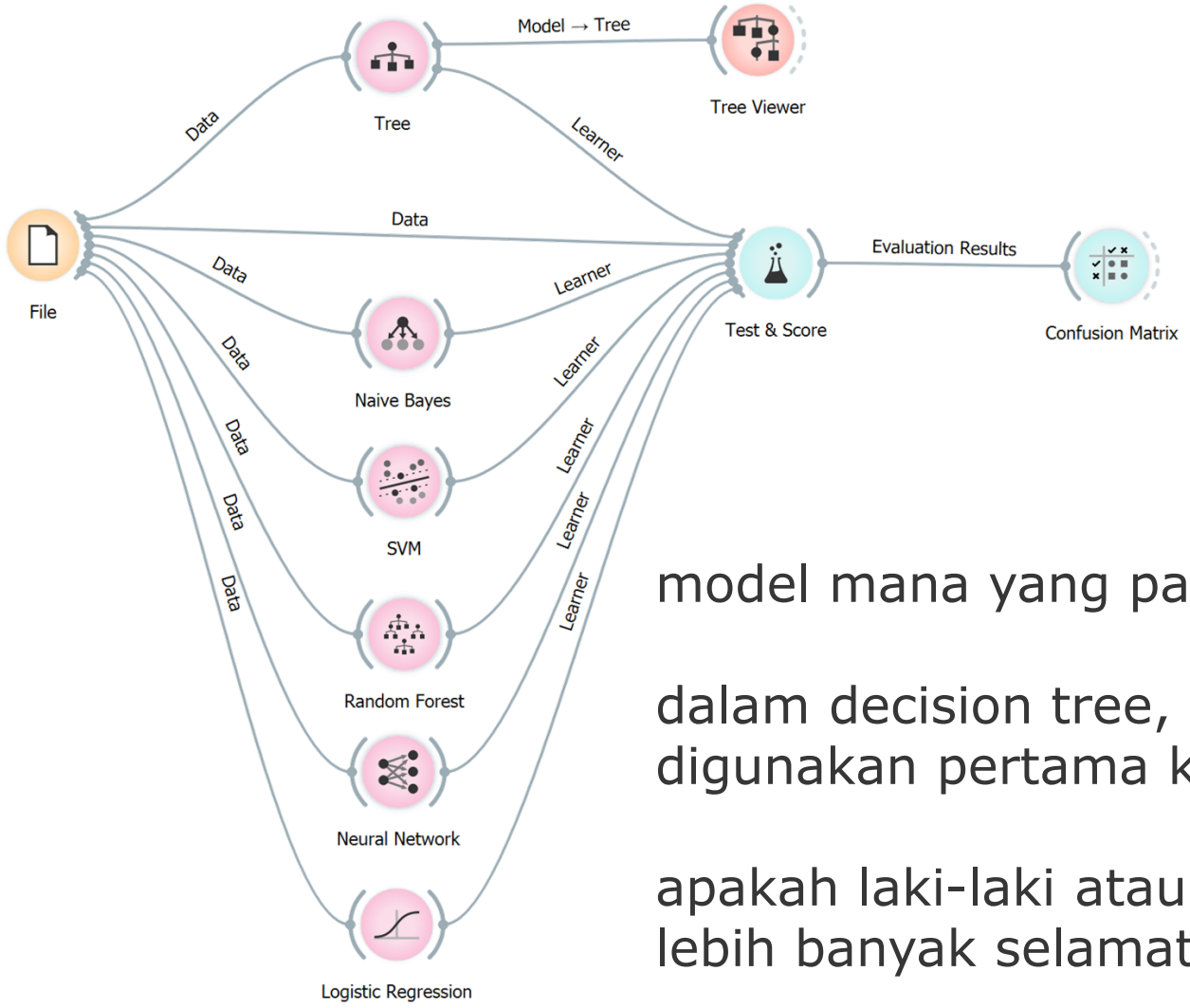

Confusion Matrix
Tree
Naive Bayes
Show: Number of instances
Predicted
Iris-setosa Iris-versicolor Iris-virginica Σ
Actual
Iris-setosa 50 0 0 50
Iris-versicolor 0 46 4 50
Iris-virginica 0 3 47 50
Σ 50 49 51 150
Output
Predictions
Probabilities
Send Automatically
Confusion Matrix
Tree
Naive Bayes
Show: Number of instances
Predicted
Iris-setosa Iris-versicolor Iris-virginica Σ
Actual
Iris-setosa 50 0 0 50
Iris-versicolor 0 42 8 50
Iris-virginica 0 7 43 50
Σ 50 49 51 150
Output
Predictions
Probabilities
Send Automatically
Select Correct
Select Misclassified
Clear Selection

Mencoba dengan Algoritma lain



model mana yang paling baik?

Mencoba dengan Data “Titanic”



model mana yang paling baik?

dalam decision tree, atribut apa yang digunakan pertama kali?

apakah laki-laki atau perempuan yang lebih banyak selamat?

TANYA JAWAB