KNN, Naïve-Bayes and Decision Tree Classifiers

Nearest Neighbor Classifier

	history	*science*	*research*	*offers*	*students*	*hall*	Class
Anthropology	0	0.537	0.477	0	0.673	0.177	A
Art	0	0	0	0.961	0.195	0.196	B
Biology	0	0.347	0.924	0	0.111	0.112	A
Chemistry	0	0.975	0	0	0.155	0.158	A
Communication	0	0	0	0.780	0.626	0	B
Computer Science	0	0.989	0	0	0.130	0.067	A
Criminal Justice	0	0	0	0	1	0	B
Economics	0	0	1	0	0	0	A
English	0	0	0	0.980	0	0.199	B
Geography	0	0.849	0	0	0.528	0	A
History	0.991	0	0	0.135	0	0	B
Mathematics	0	0.616	0.549	0.490	0.198	0.201	A
Modern Languages	0	0	0	0.928	0	0.373	B
Music	0.970	0	0	0	0.170	0.172	B
Philosophy	0.741	0	0	0.658	0	0.136	B
Physics	0	0	0.894	0	0.315	0.318	A
Political Science	0	0.933	0.348	0	0.062	0.063	A
Psychology	0	0	0.852	0.387	0.313	0.162	A
Sociology	0	0	0.639	0.570	0.459	0.237	A
Theatre	0	0	0	0	0.967	0.254	? (B)

Nearest Neighbor Classifier

Document	Class	Similarity to Theatre
Criminal Justice	B	0.967075
Anthropology	A	0.695979
Communication	B	0.605667
Geography	A	0.510589
Sociology	A	0.504672
Physics	A	0.385508
Psychology	A	0.343685
Mathematics	A	0.242155
Art	B	0.238108
Music	B	0.207746
Chemistry	A	0.189681
Computer Science	A	0.142313
Biology	A	0.136097
Modern Languages	B	0.0950206
Political Science	A	0.0762211
English	B	0.0507843
Philosophy	B	0.0345299
History	B	0
Economics	A	0

Nearest Neighbor Classifier

- 1-NN: B
- k-NN:
  - 3-NN: B
  - 5-NN: A
- Distance Weighted k-NN
  - Distance weighted 3-NN: B
  - Distance weighted 19-NN: A

Naïve Bayes Classifier [2]

	history	*science*	*research*	*offers*	*students*	*hall*	Class
Anthropology	0	1	1	0	1	1	A
Art	0	0	0	1	1	1	B
Biology	0	1	1	0	1	1	A
Chemistry	0	1	0	0	1	1	A
Communication	0	0	0	1	1	0	B
Computer Science	0	1	0	0	1	1	A
Criminal Justice	0	0	0	0	1	0	B
Economics	0	0	1	0	0	0	A
English	0	0	0	1	0	1	B
Geography	0	1	0	0	1	0	A
History	1	0	0	1	0	0	B
Mathematics	0	1	1	1	1	1	A
Modern Languages	0	0	0	1	0	1	B
Music	1	0	0	0	1	1	B
Philosophy	1	0	0	1	0	1	B
Physics	0	0	1	0	1	1	A
Political Science	0	1	1	0	1	1	A
Psychology	0	0	1	1	1	1	A
Sociology	0	0	1	1	1	1	A
Theatre	0	0	0	0	1	1	? (B)

Naïve Bayes Classifier [2]

$$P(C \mid x) = \frac{P(x \mid C)\, P(C)}{P(x)}$$

$$P(x \mid C) = P(x_1, x_2, \ldots, x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

Naïve Bayes Classifier [2]

	history	*science*	*research*	*offers*	*students*	*hall*	Class
Anthropology	0	1	1	0	1	1	A
Art	0	0	0	1	1	1	B
Biology	0	1	1	0	1	1	A
Chemistry	0	1	0	0	1	1	A
Communication	0	0	0	1	1	0	B
Computer Science	0	1	0	0	1	1	A
Criminal Justice	0	0	0	0	1	0	B
Economics	0	0	1	0	0	0	A
English	0	0	0	1	0	1	B
Geography	0	1	0	0	1	0	A
History	1	0	0	1	0	0	B
Mathematics	0	1	1	1	1	1	A
Modern Languages	0	0	0	1	0	1	B
Music	1	0	0	0	1	1	B
Philosophy	1	0	0	1	0	1	B
Physics	0	0	1	0	1	1	A
Political Science	0	1	1	0	1	1	A
Psychology	0	0	1	1	1	1	A
Sociology	0	0	1	1	1	1	A
Theatre	0	0	0	0	1	1	? (B)

$$P(\mathrm{A}) = 11/19 = 0.578947$$

$$P(\mathbf{B}) = 8/19 = 0.421053$$

$$P(C \mid x) = \frac{P(x \mid C)\, P(C)}{P(x)}$$

Naïve Bayes Classifier [2]

$$P(C \mid x) = \frac{P(x \mid C)\, P(C)}{P(x)} \qquad P(x \mid C) = P(x_1, x_2, \ldots, x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

	history	*science*	*research*	*offers*	*students*	*hall*	Class
Anthropology	0	1	1	0	1	1	A
Art	0	0	0	1	1	1	B
Biology	0	1	1	0	1	1	A
Chemistry	0	1	0	0	1	1	A
Communication	0	0	0	1	1	0	B
Computer Science	0	1	0	0	1	1	A
Criminal Justice	0	0	0	0	1	0	B
Economics	0	0	1	0	0	0	A
English	0	0	0	1	0	1	B
Geography	0	1	0	0	1	0	A
History	1	0	0	1	0	0	B
Mathematics	0	1	1	1	1	1	A
Modern Languages	0	0	0	1	0	1	B
Music	1	0	0	0	1	1	B
Philosophy	1	0	0	1	0	1	B
Physics	0	0	1	0	1	1	A
Political Science	0	1	1	0	1	1	A
Psychology	0	0	1	1	1	1	A
Sociology	0	0	1	1	1	1	A
Theatre	0	0	0	0	1	1	? (B)

$$P(\text{A} \mid \text{Theatre}) = \frac{P(\text{Theatre} \mid \text{A})\, P(\text{A})}{P(\text{Theatre})}$$

$$P(\text{Theatre} \mid \text{A}) = P(\text{history} = 0 \mid \text{A}) \times P(\text{science} = 0 \mid \text{A}) \times P(\text{research} = 0 \mid \text{A}) \times P(\text{offers} = 0 \mid \text{A}) \times P(\text{students} = 1 \mid \text{A}) \times P(\text{hall} = 1 \mid \text{A})$$

$$P(\text{Theatre} \mid \text{A}) = \tfrac{11}{11} \times \tfrac{4}{11} \times \tfrac{3}{11} \times \tfrac{8}{11} \times \tfrac{10}{11} \times \tfrac{9}{11} = 0.0536476$$

$$P(\text{Theatre} \mid \text{B}) = \tfrac{5}{8} \times \tfrac{8}{8} \times \tfrac{8}{8} \times \tfrac{2}{8} \times \tfrac{4}{8} \times \tfrac{5}{8} = 0.0488281$$

Naïve Bayes Classifier [2]

$$P(C \mid x) = \frac{P(x \mid C)\, P(C)}{P(x)}$$

$$P(\mathrm{A}) = 11/19 = 0.578947$$

$$P(\mathbf{B}) = 8/19 = 0.421053$$

$$P(\text{Theatre} \mid \mathrm{A}) = \tfrac{11}{11} \times \tfrac{4}{11} \times \tfrac{3}{11} \times \tfrac{8}{11} \times \tfrac{10}{11} \times \tfrac{9}{11} = 0.0536476$$

$$P(\text{Theatre} \mid \mathrm{B}) = \tfrac{5}{8} \times \tfrac{8}{8} \times \tfrac{8}{8} \times \tfrac{2}{8} \times \tfrac{4}{8} \times \tfrac{5}{8} = 0.0488281$$

$$P(\mathrm{A} \mid \text{Theatre}) = \frac{(0.0536476)(0.578947)}{P(\text{Theatre})} \approx 0.0310591$$

$$P(\mathrm{B} \mid \text{Theatre}) = \frac{(0.0488281)(0.421053)}{P(\text{Theatre})} \approx 0.0205592$$

$$P(\mathrm{A} \mid \text{Theatre}) = \frac{0.0310591}{0.0310591 + 0.0205592} = 0.601707$$

$$P(\mathrm{B} \mid \text{Theatre}) = \frac{0.0205592}{0.0310591 + 0.0205592} = 0.398293$$

Laplacian Correction

Naïve Bayes Classifier – Multinomial [2]

	history	*science*	*research*	*offers*	*students*	*hall*	Class
Anthropology	0	1	1	0	4	1	A
Art	0	0	0	2	1	1	B
Biology	0	1	3	0	1	1	A
Chemistry	0	2	0	0	1	1	A
Communication	0	0	0	1	2	0	B
Computer Science	0	5	0	0	2	1	A
Criminal Justice	0	0	0	0	1	0	B
Economics	0	0	1	0	0	0	A
English	0	0	0	2	0	1	B
Geography	0	1	0	0	2	0	A
History	7	0	0	2	0	0	B
Mathematics	0	1	1	1	1	1	A
Modern Languages	0	0	0	1	0	1	B
Music	1	0	0	0	1	1	B
Philosophy	1	0	0	2	0	1	B
Physics	0	0	1	0	1	1	A
Political Science	0	5	2	0	1	1	A
Psychology	0	0	2	1	2	1	A
Sociology	0	0	1	1	2	1	A
Theatre	0	0	0	0	4	1	? (B)

Naïve Bayes Classifier – Multinomial [2]

	history	*science*	*research*	*offers*	*students*	*hall*	Class
Anthropology	0	1	1	0	4	1	A
Art	0	0	0	2	1	1	B
Biology	0	1	3	0	1	1	A
Chemistry	0	2	0	0	1	1	A
Communication	0	0	0	1	2	0	B
Computer Science	0	5	0	0	2	1	A
Criminal Justice	0	0	0	0	1	0	B
Economics	0	0	1	0	0	0	A
English	0	0	0	2	0	1	B
Geography	0	1	0	0	2	0	A
History	7	0	0	2	0	0	B
Mathematics	0	1	1	1	1	1	A
Modern Languages	0	0	0	1	0	1	B
Music	1	0	0	0	1	1	B
Philosophy	1	0	0	2	0	1	B
Physics	0	0	1	0	1	1	A
Political Science	0	5	2	0	1	1	A
Psychology	0	0	2	1	2	1	A
Sociology	0	0	1	1	2	1	A
Theatre	0	0	0	0	4	1	? (B)

$$P(t_i|C) = \frac{\sum_{j=1}^{n} n_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} n_{ij}}$$

$$P\,(history\,|\,\mathrm{A}) = (0+1)/(57+2) = 0.017$$

$$P(history\,|\,\mathrm{B}) = (9+1)/(29+2) = 0.323.$$

Naïve Bayes Classifier – Multinomial [2]

$$P(d_j|C) = \left(\sum_{i=1}^{m} n_{ij}\right)! \prod_{i=1}^{m} \frac{P(t_i|C)^{n_{ij}}}{n_{ij}!}$$

i, m – Terms
j, n - Documents

	history	*science*	*research*	*offers*	*students*	*hall*	Class
Anthropology	0	1	1	0	4	1	A
Art	0	0	0	2	1	1	B
Biology	0	1	3	0	1	1	A
Chemistry	0	2	0	0	1	1	A
Communication	0	0	0	1	2	0	B
Computer Science	0	5	0	0	2	1	A
Criminal Justice	0	0	0	0	1	0	B
Economics	0	0	1	0	0	0	A
English	0	0	0	2	0	1	B
Geography	0	1	0	0	2	0	A
History	7	0	0	2	0	0	B
Mathematics	0	1	1	1	1	1	A
Modern Languages	0	0	0	1	0	1	B
Music	1	0	0	0	1	1	B
Philosophy	1	0	0	2	0	1	B
Physics	0	0	1	0	1	1	A
Political Science	0	5	2	0	1	1	A
Psychology	0	0	2	1	2	1	A
Sociology	0	0	1	1	2	1	A
Theatre	0	0	0	0	4	1	? (B)

$$P(\text{Theatre} \mid \text{A}) = 5! \times \frac{0.017^0}{0!} \times \frac{0.288^0}{0!} \times \frac{0.22^0}{0!} \times \frac{0.068^0}{0!} \times \frac{0.305^4}{4!} \times \frac{0.017^1}{1!}$$

$$P(\text{Theatre} \mid \text{B}) = 5! \times \frac{0.323^0}{0!} \times \frac{0.0323^0}{0!} \times \frac{0.0323^0}{0!} \times \frac{0.355^0}{0!} \times \frac{0.194^4}{4!} \times \frac{0.194^1}{1!}$$

Naïve Bayes Classifier – Multinomial [2]

Thus, we obtain $P(\text{A} \mid \text{Theatre}) \approx 0.0000354208$ and $P(\text{B} \mid \text{Theatre}) \approx 0.00000476511$, and after normalization, $P(\text{A} \mid \text{Theatre}) = 0.88$ and $P(\text{B} \mid \text{Theatre}) = 0.12$. The winner is class A, with even more significant advantage over the boolean case (0.60 to 0.40).

Naïve Bayes Classifier- Gaussian [1]

If A_k is continuous-valued, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ, defined by

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{6.13}$$

so that

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}). \tag{6.14}$$

These equations may appear daunting, but hold on! We need to compute μ_{C_i} and σ_{C_i}, which are the mean (i.e., average) and standard deviation, respectively, of the values of attribute A_k for training tuples of class C_i. We then plug these two quantities into Equation (6.13), together with x_k, in order to estimate $P(x_k|C_i)$.

Naïve Bayes Classifier- Gaussian

Assume training set shown in the following table.

Temperature	Humidity	Play
85	85	No
80	90	No
65	70	No
72	95	No
71	80	No
83	78	Yes
70	96	Yes
68	80	Yes
64	65	Yes
69	79	Yes
75	80	Yes
75	70	Yes
72	90	Yes
81	75	Yes

Assume "Play" as the class attribute. Use naïve Bayes classifier to predict whether play will be possible given the Temperature = 83 and Humidity = 64. Fit Gaussian distribution to the data.

Naïve Bayes Classifier- Gaussian

(4) $$P(x_k|C_i) = \frac{1}{\sqrt{2\pi}\,\sigma_{x|C_i}} e^{-\frac{(x_k - \mu_{x|C_i})^2}{2\sigma^2_{x|C_i}}}$$

$\mu_{Temp|No} = 74.6$ $\quad$ $\sigma_{Temp|No} = 7.06$

$\mu_{Temp|Yes} = 73$ $\quad$ $\sigma_{Temp|Yes} = 5.81$

$\mu_{Humidity|No} = 84$ $\quad$ $\sigma_{Humidity|No} = 8.60$

$\mu_{Humidity|Yes} = 79.22$ $\quad$ $\sigma_{Humidity|Yes} = 8.85$

Naïve Bayes Classifier- Gaussian

$P(\text{Yes} \mid T=83, \text{Hum}=64) = P(T=83, \text{Hum}=64 \mid \text{Yes}) \times P(\text{Yes})$ (3)

$$= P(T=83 \mid \text{Yes}) \times P(\text{Hum}=64 \mid \text{Yes}) \times P(\text{Yes})$$

$$= \frac{1}{\sqrt{2\pi}\,(5.81)} e^{-\frac{(83-73)^2}{2\cdot(5.81)^2}} \times \frac{1}{\sqrt{2\pi}\,(8.85)} e^{-\frac{(64-79.2)^2}{2\cdot(8.85)^2}} \times 9/14$$

$$= \frac{1}{(2.51)(5.81)} \times e^{-100/67.51} \times \frac{1}{(2.51)(8.85)} \times e^{-231.04/78.32} \times \frac{9}{14}$$

$$= 0.069 \times 0.23 \times 0.045 \times 0.05 \times \frac{9}{14}$$

$$= 0.000023 \quad \text{— (1)}$$

Naïve Bayes Classifier- Gaussian

$$P(No \mid T=83, Hum=64) = P(T=83, Hum=64 \mid No) \times P(No)$$

$$= P(T=83 \mid No) \times P(Hum=64 \mid No) \times P(No)$$

$$= \frac{1}{\sqrt{2\pi}\,(7.06)} e^{-\frac{(83-74.6)^2}{2\cdot(7.06)^2}} \times \frac{1}{\sqrt{2\pi}\,(8.60)} e^{-\frac{(64-84)^2}{2\cdot(8.60)^2}} \times \frac{5}{14}$$

$$= \frac{1}{(2.51)(7.06)} \times e^{-70.56/99.69} \times \frac{1}{(2.51)(8.60)} \times e^{-\frac{400}{147.92}} \times \frac{5}{14}$$

$$= 0.056 \times 0.493 \times 0.046 \times 0.0669 \times 5/14$$

$$= 0.0000303 \; > \; 1$$ Result in

Prediction: No

One More Example

Class-labeled training tuples from the *AllElectronics* customer database.

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

One More Example

$$X = (age = youth, income = medium, student = yes, credit\_rating = fair)$$

We need to maximize $P(X|C_i)P(C_i)$, for $i = 1, 2$. $P(C_i)$, the prior probability of each class, can be computed based on the training tuples:

$$P(buys\_computer = yes) = 9/14 = 0.643$$
$$P(buys\_computer = no) = 5/14 = 0.357$$

To compute $PX|C_i)$, for $i = 1, 2$, we compute the following conditional probabilities:

$$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$$
$$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$$
$$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$$
$$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$$
$$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$$
$$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$$

One More Example

Using the above probabilities, we obtain

$$P(X|buys\_computer = yes) = P(age = youth \mid buys\_computer = yes) \times$$
$$P(income = medium \mid buys\_computer = yes) \times$$
$$P(student = yes \mid buys\_computer = yes) \times$$
$$P(credit\_rating = fair \mid buys\_computer = yes)$$
$$= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044.$$

Similarly,

$$P(X|buys\_computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$$

To find the class, C_i, that maximizes $P(X|C_i)P(C_i)$, we compute

$$P(X|buys\_computer = yes)P(buys\_computer = yes) = 0.044 \times 0.643 = 0.028$$
$$P(X|buys\_computer = no)P(buys\_computer = no) = 0.019 \times 0.357 = 0.007$$

Therefore, the naïve Bayesian classifier predicts $buys\_computer = yes$ for tuple X. ■

Accuracy Measures of Classifier

- Accuracy
- Error Rate/Misclassification Rate
- Re-substitution Error
- Confusion Matrix

		Predicted class	
		C_1	C_2
Actual class	C_1	true positives	false negatives
	C_2	false positives	true negatives

- Precision Positive, Precision Negative, Precision
  - If precision positive is 100%, it means that all the observations that ate predicted as positive are in fact positive but there may be some positive observations in testing set which might have been predicted as negative. If CF = [405 95;126 142]. PP = 405/531, PN = 142/237 and Precision = (500/768) * .763 + (268/768) * .599
- Recall Positive (Sensitivity), Recall Negative (Specificity), Recall
  - If recall positive is 100%, it means that all the positive observations in testing set are predicted as positive but there may be some negative observations from testing set which might have been predicted as positive.
- F-measure = (2*Precision*Recall)/(Precision + Recall)
- Generalized Case (More than two classes)

Evaluation Methodology

- Holdout Method
- Random Subsampling
- Cross Validation
    - K-fold cross-validation
    - Leave-one-out
    - Stratified cross-validation

Evaluation Methodology

- Holdout Method
- Random Subsampling
- Cross Validation
  - K-fold cross-validation
  - Leave-one-out
  - Stratified cross-validation
- Bootstrap (.632 bootstrap)
  - Assume data set of d observations.
  - The data set is sampled d times with replacement. This gives training set with d samples with probably some repetitions in it.
  - The observations that did not make it into the training set end up forming the test set.
  - If we try this out several times, on average, 63.2% of the original observations will end up in the bootstrap, and the remaining 36.8% will form the test set.
  - Where does the figure, 63.2%, come from? Each observation has probability of 1/d of being selected, so the probability of not being selected is (1 – 1/d).
  - We have to select d times, so the probability that an observation will not be selected during this whole time is $(1 - 1/d)^d$. If d is large, the probability approaches $e^{-1} = 0.368$

Evaluation Methodology

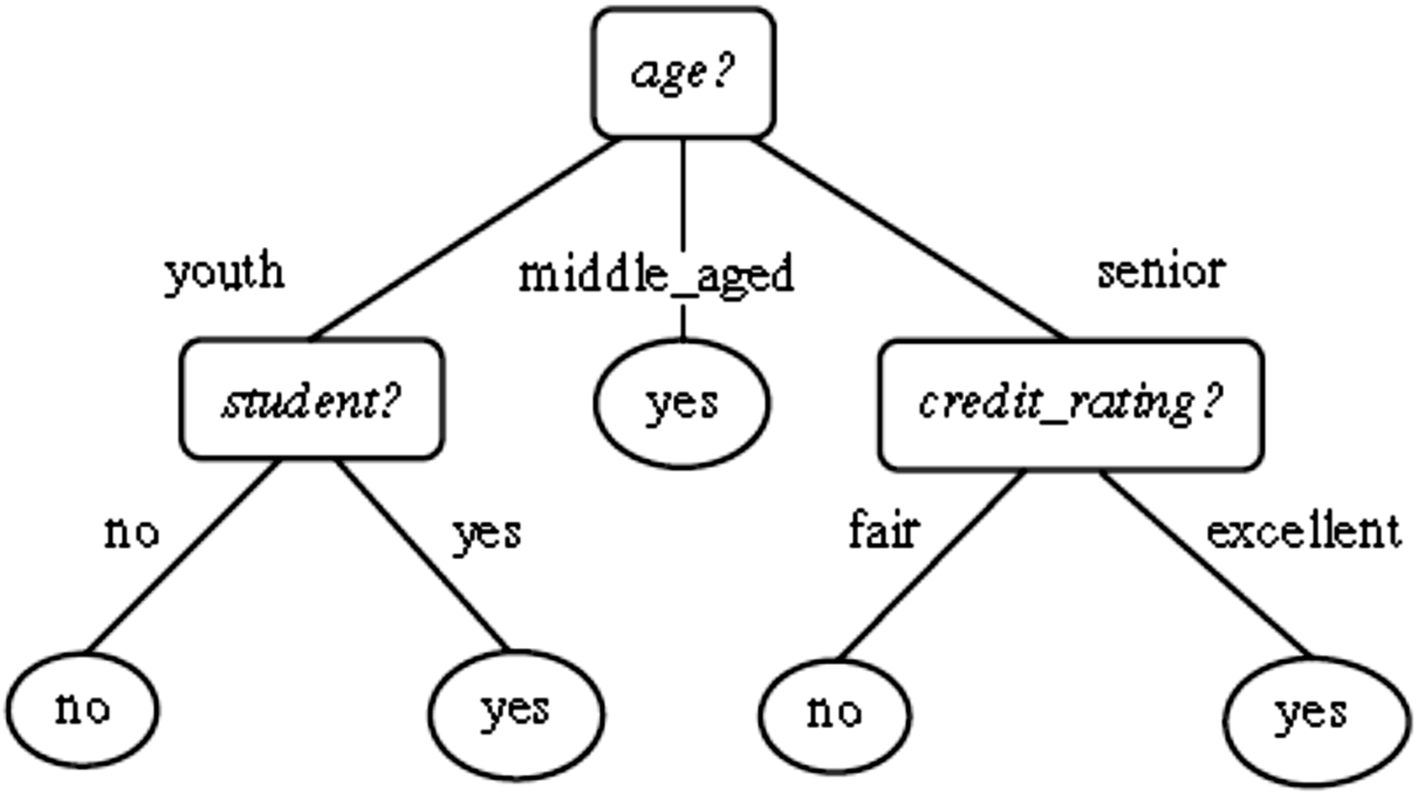
- Bootstrap (.632 bootstrap)
  - Thus, 36.8% of observations will not be selected for training and thereby end up in the test set, and the remaining 63.2% will form the training set.
  - We can repeat the sampling procedure k times, where in each iteration, we use the current test set to obtain an accuracy estimate of the model obtained from the current bootstrap sample. The overall accuracy of the model is then estimated as

$$Acc(M) = \sum_{i=1}^{k} (0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set})$$

  - where $Acc(M_i)_{test\_set}$ is the accuracy of the model obtained with bootstrap sample i when it is applied to test set i. $Acc(M_i)_{train\_set}$ is the accuracy of the model obtained with bootstrap sample i when it is applied to the original set of observations.
  - The bootstrap method works well with small data sets.

Classification

- Classification by Decision Tree Induction

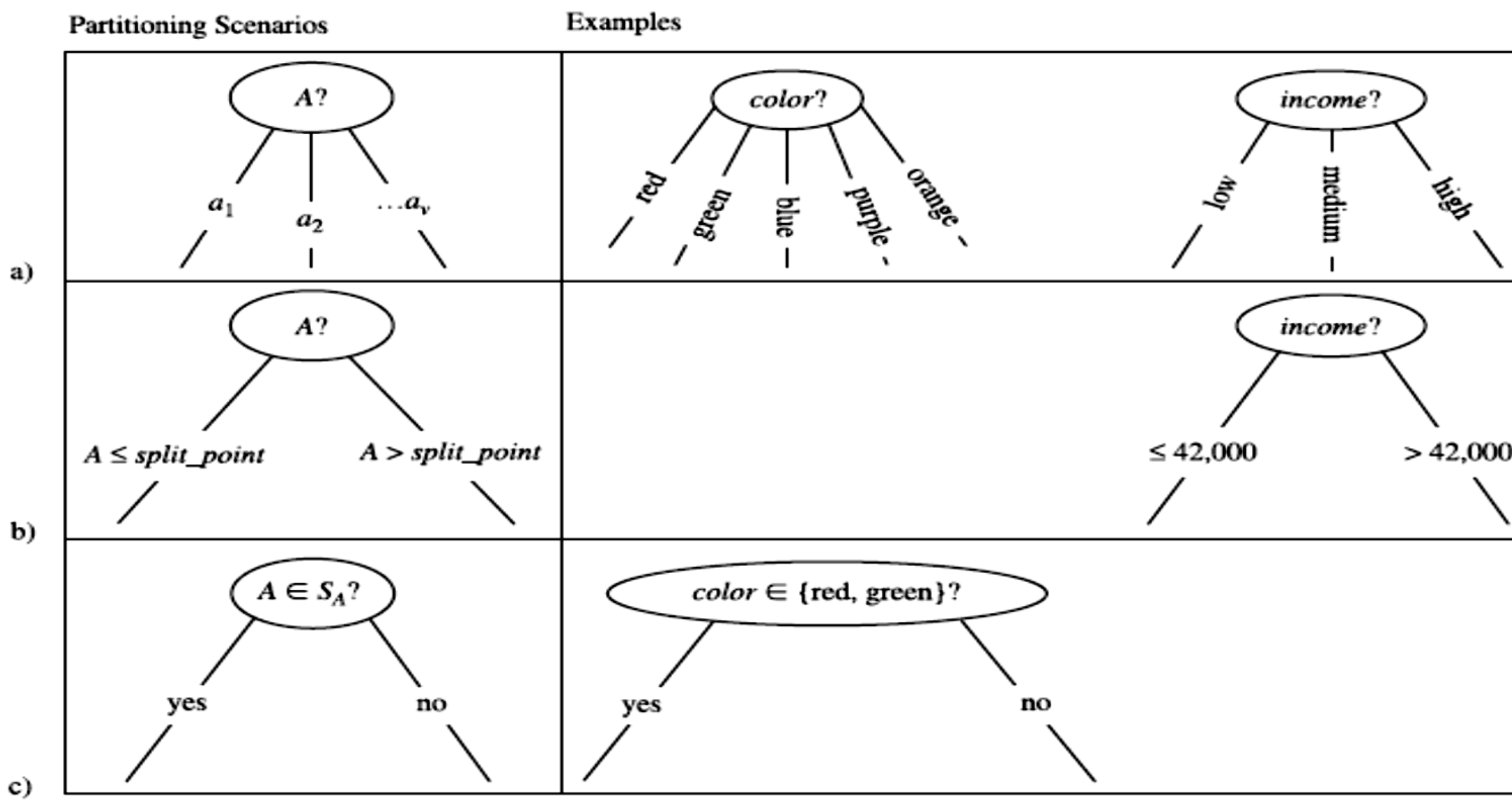


$X = (age = youth, income = medium, student = yes, credit\_rating = fair)$

Classification

- Classification by Decision Tree Induction
  - ID3 (Iterative Dichotomiser)
  - C4.5
  - CART (Classification & Regression Tree)

Classification



Three possibilities for partitioning tuples based on the splitting criterion, shown with examples. Let A be the splitting attribute. (a) If A is discrete-valued, then one branch is grown for each known value of A. (b) If A is continuous-valued, then two branches are grown, corresponding to $A \leq split\_point$ and $A > split\_point$. (c) If A is discrete-valued and a binary tree must be produced, then the test is of the form $A \in S_A$, where S_A is the splitting subset for A.

Classification

- Classification by Decision Tree Induction – ID3
  - Let node N hold the tuples of partition D.
  - The attribute with the highest information gain is chosen as the splitting attribute for node N.
  - This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions.
  - Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found.
  - The expected information needed to classify a tuple in D is given by

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

Classification

- Classification by Decision Tree Induction – ID3
  - Where p_i is the probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$.
  - A log function to the base 2 is used, because the information is encoded in bits.
  - Info(D) is just the average amount of information needed to identify the class label of a tuple in D.
  - Note that, at this point, the information we have is based solely on the proportions of tuples of each class.
  - Info(D) is also known as the entropy of D.

Classification

- Classification by Decision Tree Induction – ID3
  - Now, suppose we were to partition the tuples in D on some attribute *A* having v distinct values, $\{a_1, a_2, \ldots, a_v\}$, as observed from the training data.
  - If *A* is discrete-valued, these values correspond directly to the v outcomes of a test on *A*.
  - Attribute *A* can be used to split D into v partitions or subsets, $\{D_1, D_2, \ldots, D_v\}$, where D_j contains those tuples in D that have outcome a_j of *A*.
  - These partitions would correspond to the branches grown from node N. Ideally, we would like this partitioning to produce an exact classification of the tuples.

Classification

- Classification by Decision Tree Induction – ID3
  - That is, we would like for each partition to be pure.
  - However, it is quite likely that the partitions will be impure (e.g., where a partition may contain a collection of tuples from different classes rather than from a single class).
  - How much more information would we still need (after the partitioning) in order to arrive at an exact classification?
  - This amount is measured by $Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$.
  - The term $|D_j|/|D|$ acts as the weight of the j^{th} partition. $Info_A(D)$ is the expected information required to classify a tuple from D based on the partitioning by A.
  - The smaller the expected information (still) required, the greater the purity of the partitions.

Classification

- Classification by Decision Tree Induction – ID3
  - Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on A).
  - That is, $Gain(A) = Info(D) - Info_A(D)$
  - In other words, $Gain(A)$ tells us how much would be gained by branching on A.
  - It is the expected reduction in the information requirement caused by knowing the value of A.
  - The attribute A with the highest information gain, ($Gain(A)$), is chosen as the splitting attribute at node N.
  - This is equivalent to saying that we want to partition on the attribute A that would do the "best classification," so that the amount of information still required to finish classifying the tuples is minimal (i.e., minimum $Info_A(D)$).

Classification

- Classification by Decision Tree Induction – ID3

Table 6.1 Class-labeled training tuples from the *AllElectronics* customer database.

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Classification

- Classification by Decision Tree Induction – ID3

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

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

$$\begin{aligned} Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) \\ &+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) \\ &+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

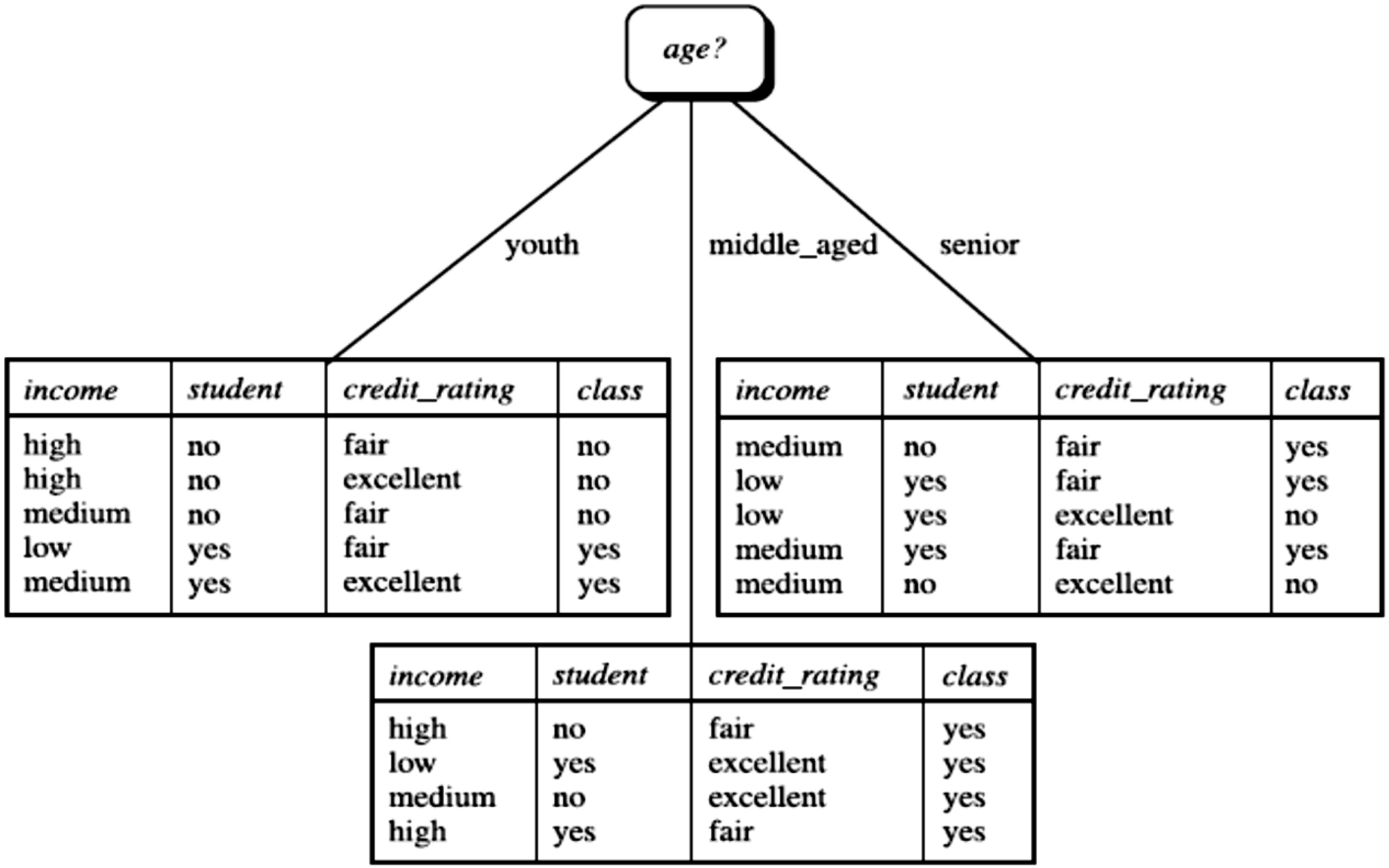
$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j).$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

Similarly, we can compute $Gain(income) = 0.029$ bits, $Gain(student) = 0.151$ bits, and $Gain(credit\_rating) = 0.048$ bits. Because *age* has the highest information gain among the attributes, it is selected as the splitting attribute. Node N is labeled with *age*, and branches are grown for each of the attribute's values. The tuples are then partitioned accordingly, as shown in Figure 6.5.

Classification

- Classification by Decision Tree Induction – ID3



youth:

income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

middle_aged:

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

senior:

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

Figure 6.5 The attribute *age* has the highest information gain and therefore becomes the splitting attribute at the root node of the decision tree. Branches are grown for each outcome of *age*. The tuples are shown partitioned accordingly.

Classification

- Classification by Decision Tree Induction – C4.5
  - Gain Ratio
    - The information gain measure is biased toward tests with many outcomes.
    - That is, it prefers to select attributes having a large number of values.
    - For example, consider an attribute that acts as a unique identifier, such as product ID.
    - A split on product ID would result in a large number of partitions (as many as there are values), each one containing just one tuple.
    - Because each partition is pure, the information required to classify data set D based on this partitioning would be $Info_{product\_ID}(D) = 0$.
    - Therefore, the information gained by partitioning on this attribute is maximal.
    - Clearly, such a partitioning is useless for classification. C4.5, a successor of ID3, uses an extension to information gain known as gain ratio, which attempts to overcome this bias.

Classification

- Classification by Decision Tree Induction – C4.5
  - Gain Ratio
    - It applies a kind of normalization to information gain using a "split information" value defined analogously with Info(D) as

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right).$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}.$$

    - The attribute with the maximum gain ratio is selected as the splitting attribute.

Classification

- ## Classification by Decision Tree Induction – C4.5
  - ### Gain Ratio

Example 6.2 **Computation of gain ratio for the attribute *income*.** A test on *income* splits the data of Table 6.1 into three partitions, namely *low*, *medium*, and *high*, containing four, six, and four tuples, respectively. To compute the gain ratio of *income*, we first use Equation (6.5) to obtain

$$SplitInfo_A(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right).$$

$$= 0.926.$$

From Example 6.1, we have $Gain(income) = 0.029$. Therefore, $GainRatio(income) = 0.029/0.926 = 0.031$. ■

Classification

- Classification by Decision Tree Induction – CART
  - Gini Index
    - The Gini index measures the impurity of D, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

    - The Gini index considers a binary split for each attribute.
    - For discrete-valued attribute, all possible combinations of its value except empty set and power set are considered for binary split.

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2).$$

    - For a discrete-valued attribute, the subset that gives the minimum gini index for that attribute is selected as its splitting subset.
    - For continuous-valued attributes, each possible split point must be considered.

Classification

- Classification by Decision Tree Induction – CART
  - Gini Index
    - The reduction in impurity that would be incurred by a binary split on a discrete- or continuous-valued attribute *A* is

$$\Delta Gini(A) = Gini(D) - Gini_A(D).$$

    - The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute.

Classification

- Classification by Decision Tree Induction – CART
  - Gini Index

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459.$$

$$\begin{aligned}
&Gini_{income \in \{low, medium\}}(D) \\
&= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2) \\
&= \frac{10}{14}\left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right) \\
&= 0.450 \\
&= Gini_{income \in \{high\}}(D).
\end{aligned}$$

Classification

- Classification by Decision Tree Induction
  - Information gain, as we saw, is biased toward multivalued attributes.
  - Although the gain ratio adjusts for this bias, it tends to prefer unbalanced splits in which one partition is much smaller than the others.
  - The Gini index is biased toward multivalued attributes and has difficulty when the number of classes is large.
  - It also tends to favour tests that result in equal-sized partitions and purity in both partitions.
  - Although biased, these measures give reasonably good results in practice.

Disclaimer

- Content of this presentation is not original and it has been prepared from various sources for teaching purpose.