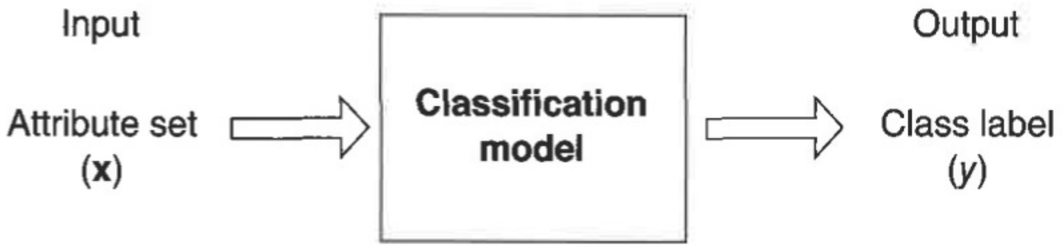
CENG3521 - Data Mining

LECTURE 3

CLASSIFICATION

Classification ,which is the task of assigning objects to one of several **predefined categories**.



Classification is the task of mapping an input attribute set x into its class label y.

Preliminaries

- The input data for a classification task is a collection of records.

- Each record, also known as an **instance** or example, is characterized by a tuple (*x*,*y*), where *x* is the attribute set and *y* a special attribute, designated as the class label (also known as category or target attribute)

Preliminaries

- Classification is the task of learning a target function that maps each attribute set *x* to one of the predefined class labels *y*.
- The target function is also known informally as a classification model.

i. Descriptive Modelling: A classification model can serve as an explanatory tool to distinguish between objects of different classes. A descriptive model summarizes the data.

ii. Predictive Modelling: A classification model can also be used to predict the class label of unknown records.

CLASSIFICATION

Preliminaries

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Preliminaries

ii. Predictive Modelling: A classification model can also be used to predict the class label of unknown records.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

Preliminaries

- Classification techniques are most suited for predicting or describing data sets with binary or nominal categories.
- They are less effective for ordinal categories(e.g. ,to classify a person as a member of high-,medium-,or low- income group) because they do not consider the implicit order among the categories.
- Other forms of relationships, such as the subclass-superclass relationships among categories are also ignored.

General Approach

- A classification technique (or classifier) is a systematic approach to building classification models from an input data set. Examples include <u>decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and naive Bayes classifier.</u>

- Each technique employs a **learning algorithm** to identify a model that **best fits the relationship** between the attribute set and class label of the input data.

General Approach

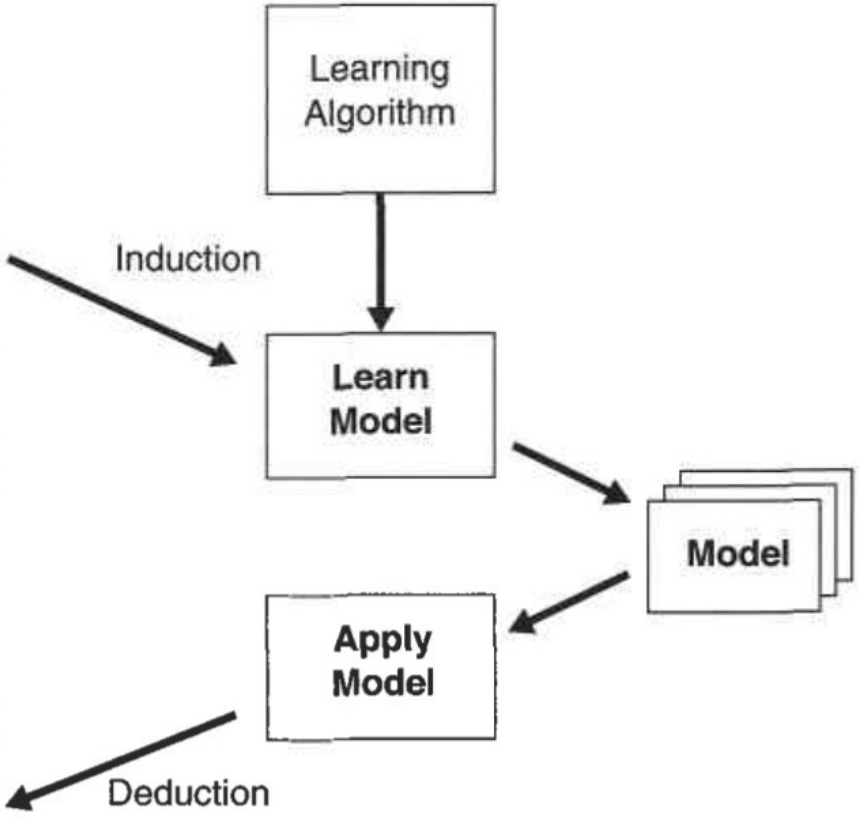
- The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before

- A key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records.

General Approach

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes



Test Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Confusion Matrix

- Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as a confusion matrix.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

Confusion Matrix

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

Each entry f_{ij} in this table denotes the number of records from class i predicted to be of class j. For instance; f_{01} is the number of records from class *0* incorrectly predicted as class *1*. Based on the entries in the confusion matrix, the total number of correct predictions made by the model is $(f_{11}+f_{00})$ and the total number of incorrect predictions is $(f_{10}+f_{01})$.

CLASSIFICATION

Confusion Matrix

Although a confusion matrix provides the information needed to determine how well a classification model performs, to compare the performance of different models, a performance metric such to measure **accuracy** is defined as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Confusion Matrix

Equivalently, the performance of a model can be expressed **error rate**, which is given by the following equation:

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Most classification algorithms seek models that attain the **highest accuracy**, or equivalently, the **lowest error rate** when applied to the test set.

CLASSIFICATION

Decision Trees

- How we build a decision tree is by asking a series of carefully crafted questions about the attributes of the test record.
- Each time we receive an answer a follow-up question is asked until we reach a conclusion about the class label of the record.
- The series of questions and their possible answers can be organized in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edge.

Decision Trees

The tree has three types of nodes:

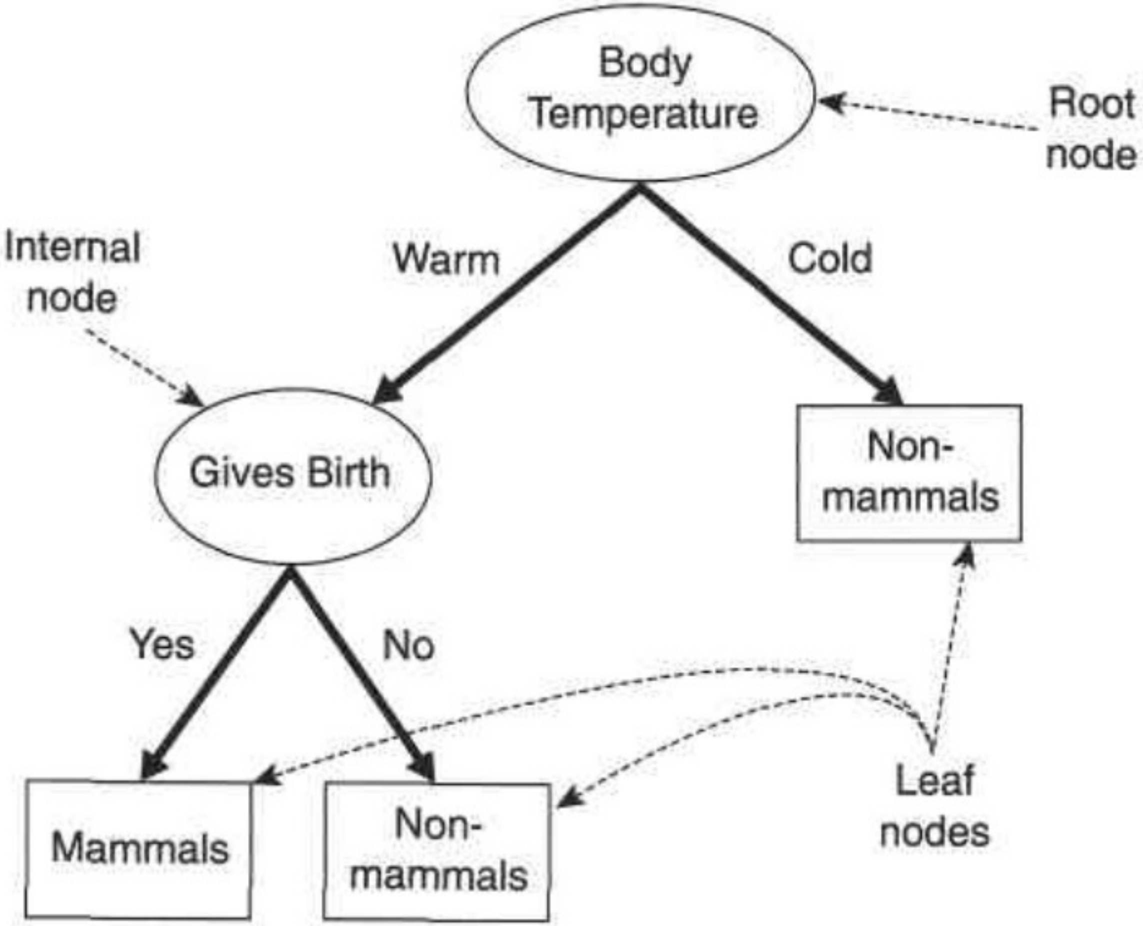
- **A root node** that has no incoming edges and zero or more outgoing edges.
- **Internal nodes**, each of which has exactly one incoming edge and two or more outgoing edges.
- **Leaf or terminal nodes**, each of which has exactly one incoming edge and no outgoing edges.

Decision Trees

- In a decision tree, each leaf node is assigned a class label.
- The non- terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics.

CLASSIFICATION

Decision Trees



CLASSIFICATION

Hunt’s Algorithm

In Hunt's algorithm, a decision tree is grown in a recursive fashion by partitioning the training records into successively purer subsets.

Hunt's Algorithm

Let D_t be the set of training records that are associated with node t and y: $\{y_1, y_2, \ldots, y_c\}$ be the class labels

The following is a recursive definition of Hunt's algorithm.

- Step 1: If all the records in D_t belong to the same class y_t, then t is a leaf node labeled as y.
- Step 2: If D_t ; contains records that belong to more than one class, **an attribute test condition** is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition and the records in D_t are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node

CLASSIFICATION

Hunt's Algorithm

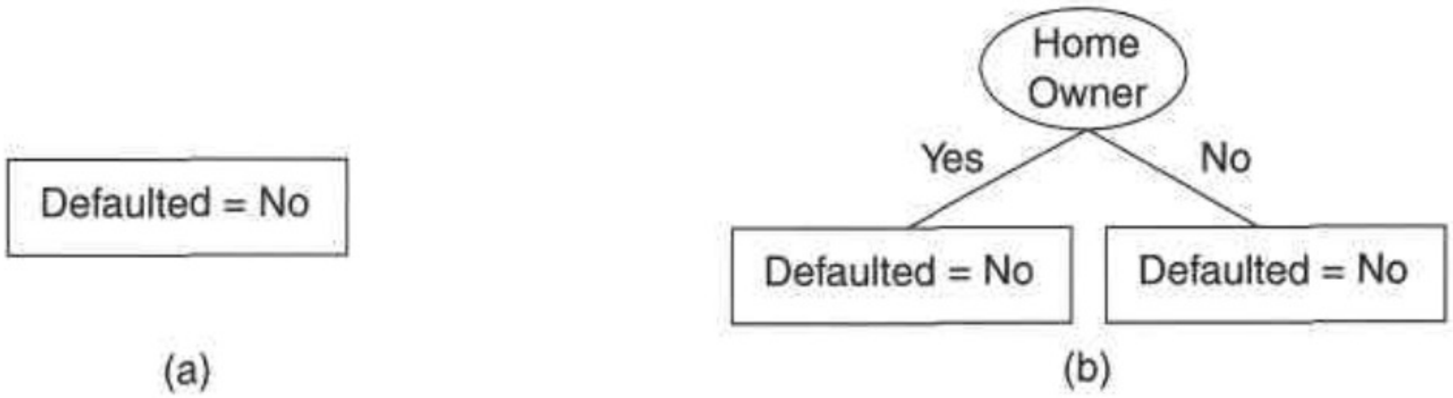
Example:

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

CLASSIFICATION

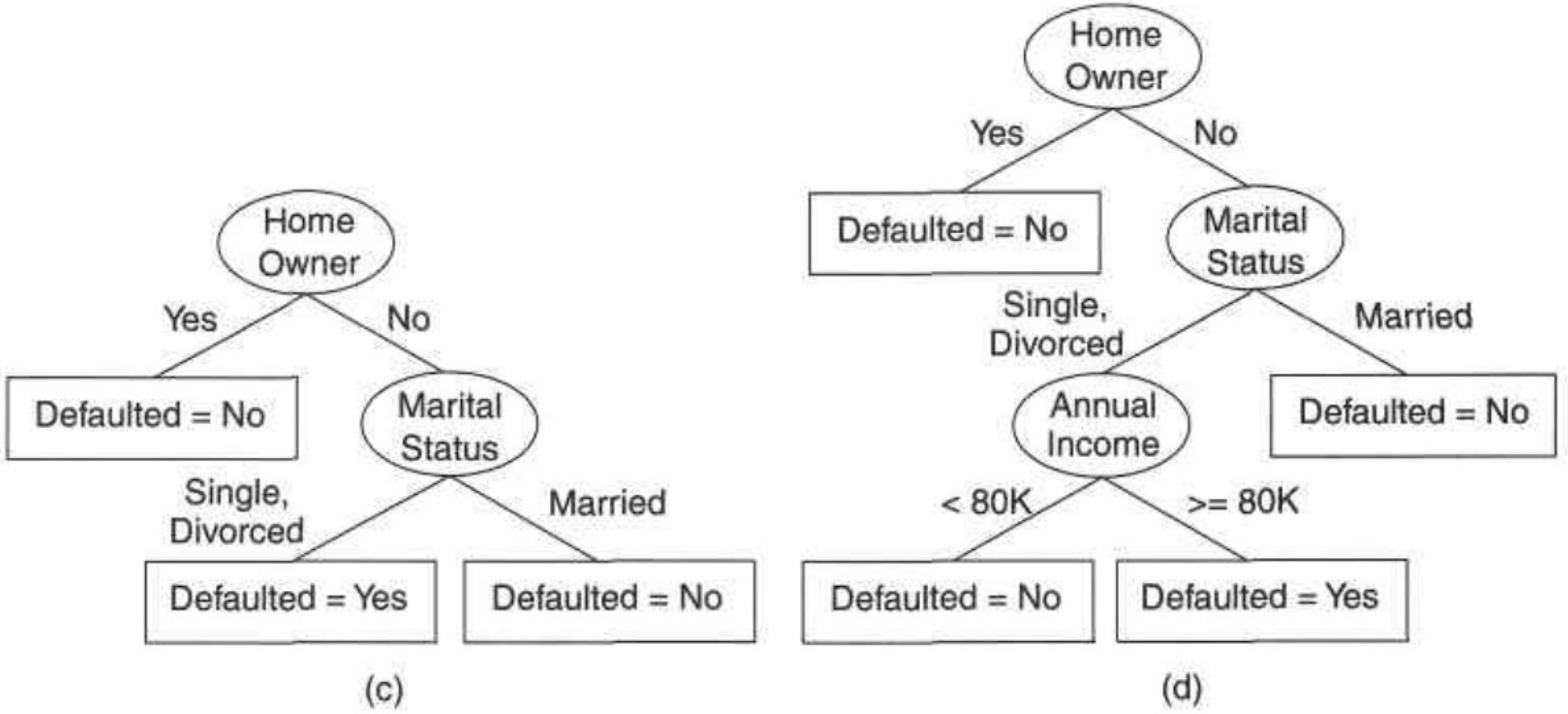
Hunt’s Algorithm

Example:



Hunt's Algorithm

Example:



Attribute Selection Measures

An **attribute selection measure** is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D*, of class-labeled training tuples into individual classes.

Attribute selection measures are also known as **splitting rules** because they determine how the tuples at a given node are to be split.

The attribute selection measure provides **a ranking** for each attribute describing the given training tuples. The attribute having the best score for the measure is chosen as the *splitting attribute* for the given tuples.

Attribute Selection Measures

- The tree node created for partition *D* is labeled with the splitting criterion, branches are grown for each out- come of the criterion, and the tuples are partitioned accordingly.

Three popular attribute selection measures:

1. *information gain*
2. *gain ratio*, and
3. *Gini index*.

1. Information Gain

This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages.

- Let node *N* represent or hold the tuples of partition *D*.
- The attribute with the highest information gain is chosen as the splitting attribute for node *N*.

The expected information needed to classify a tuple in *D* is given by:

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

where *pi* is the nonzero probability that an arbitrary tuple in *D* belongs to class *Ci*

1. Information Gain

$$Gain(D, A) = H(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i)$$

The information gain $G(D, A)$ through the use of the attribute A is determined by the difference of the average information content of the dataset $D = D_1 \cup D_2 \cup \cdots \cup D_n$ divided by the n-value attribute A and the information content $I(D)$ of the undivided dataset.

1. Information Gain

Example:

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

1. Information Gain

Example:

age?

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

CLASSIFICATION

2. Gini Index

Gini index measures the impurity of *D*, a data partition or set of training tuples, as :

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

where;

p_i = probability of an object being classified into a particular class

2. Gini Index

Example 1:

(Continuous variables)

Index	A	B	C	D	E
1	4.8	3.4	1.9	0.2	positive
2	5	3	1.6	1.2	positive
3	5	3.4	1.6	0.2	positive
4	5.2	3.5	1.5	0.2	positive
5	5.2	3.4	1.4	0.2	positive
6	4.7	3.2	1.6	0.2	Positive
7	4.8	3.1	1.6	0.2	positive
8	5.4	3.4	1.5	0.4	positive
9	7	3.2	4.7	1.4	negative
10	6.4	3.2	4.7	1.5	negative
11	6.9	3.1	4.9	1.5	negative
12	5.5	2.3	4	1.3	negative
13	6.5	2.8	4.6	1.5	negative
14	5.7	2.8	4.5	1.3	negative
15	6.3	3.3	4.7	1.6	negative
16	4.9	2.4	3.3	1	negative

2. Gini Index

Example 2: (Categorical variables)

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