Introduction to Data Mining

- **Data Mining** is the process of employing (computer-based) learning techniques to analyze and extract knowledge from data contained in a database
- The purpose of data mining is to identify trends and patterns in data
- The knowledge gained from data mining, can be thought of as a generalization of the data
- Data mining techniques use induction-based learning
- **Induction-based learning** is the process of forming general concept definitions by observing specific examples of concepts to be learned

Examples of induction-based learning

- **Credit card companies maintain a model of your credit card purchasing habits. Attempted transactions that don't fit the model, trigger an alert**
- **Web-sites maintain on-line purchasing profiles that associate the potential purchase of a product (e.g. a book) with the purchase of another product**
- **Text-mining software: detecting "tones" in text formats such as e-mail and web content. Identify aggressive customers, prioritize their requests**

Knowledge Discovery in Databases KDD

- a term frequently use in connection with Data Mining
- KDD is the application of the scientific method to data mining
- a typical KDD process includes:
  - ▷ methodology for extracting and preparing data
  - ▷ making decisions about what actions should be taken, once data mining has taken place
- Often a particular application involves the analysis of large volumes of data stored in several locations. The extraction/preparation step of the KDD process is the most time-consuming one

Data Mining and Learning

Data Mining is about learning, which is a very complex process. We can distinguish four levels of learning:

- **Facts**
  simple statements or truths
- **Concepts**
  sets of objects/symbols/events grouped together because they share common characteristics
- **Procedures**
  step-by-step courses of actions to achieve a goal
- **Principles**
  general truths or laws that are basic to other truths

- Computers can learn concepts

- Concepts are the output of data mining

- The form of a (learned) concept depends on the data mining tool

- Common concept structures: **trees, production rules, networks, mathematical equations**

Important Distinction:

▷ trees/rules are easy for humans to understand

▷ networks/math. equations are not easy for humans to understand

Three different perspectives for Concepts

- **classical view** Concepts have definite defining properties. These properties determine whether an individual item is an instance of a particular concept.
  Example: (a rule for good credit risk)

  `IF AnnualIncome` $\geq \$30K$ `AND # YearsWorking` ≥ 5 `AND Owns Home = TRUE`

  `THEN good credit risk = TRUE`

  All 3 conditions must be met, for the applicant to be a good credit risk

- **probabilistic view** Concepts are represented by properties that concept members probably satisfy. People store/recall concepts as generalizations created from individual observations.
  Example: (prob/stic view of a good credit risk)

  `The mean annual income of people who consistently make loan payments on time is` $\$30K$`. Most good credit risks have worked for at least` 5 `years. The majority of good credit risks own their own home.`

  The definition offers general guidelines about the characteristics of a good credit risk and cannot be directly applied to find whether a person should is a good

credit risk or not.

The definition can help with the decision-making process, by associating a probability, e.g. $(\$27K, \#YearsWorking = 4, OwnsHome) \rightarrow 0.85\%$

- **exemplar view** A given instance is determined to be an example of a concept, if it's similar enough to a set of known examples of the concept. People store/recall likely concept examples, that they used to classify new instances.

  Example: a possible list of 3 examples considered to be good credit risks

  `Ex 1:` $(\$32K, \#YearsWorking = 6, OwnsHome)$,

  `Ex 2:` $(\$52K, \#YearsWorking = 16, OwnsHome)$,

  `Ex 3:` $(\$28K, \#YearsWorking = 12, OwnsHome)$

  The definition can help with the decision-making process, by associating a probability.

Supervised Learning

- Supervised Learning is a very common concept learning method and data mining technique
- People use induction to form basic concept definitions. People see instances of concepts representing animals, plants, buildings, choose defining concept features (attributes) and form classification models. Later on, they use the models to help identify objects of similar structure.
- Supervised Learning is a two-fold process: (1) build models (2) use them to classify new instances
- How can we develop a generalized model to represent some given data?

Hypothetical Data Set for Disease Diagnosis

(each row, except the first, is an **instance** of data)

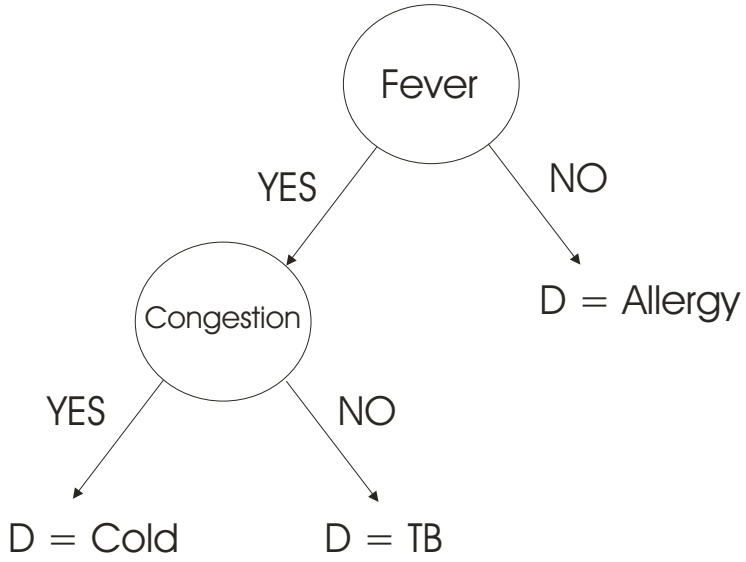
Patient ID#	Fever	Congestion	Headache	Diagnosis
1	No	Yes	Yes	Allergy
2	Yes	Yes	No	Cold
3	No	Yes	No	Allergy
4	Yes	Yes	Yes	Cold
5	No	Yes	Yes	Allergy
6	Yes	No	Yes	TB

the table is given in the **attribute-value format**, RDBMSs

- **input attributes:** Fever, Congestion, Headache (possible symptoms experienced by individuals suffering from Allergy/Cold/TB)
- **output attribute:** Diagnosis. We wish to predict its value.

Decision Trees

- A Decision Tree is a tree in which terminal nodes reflect decision outcomes and internal nodes represent tests on one or more attributes
- Decision Trees are easy to understand, can be transformed into rules and work well in practice
- There are a number of algorithms to create decision trees (C4.5, ID3)
- A Decision Tree for the Disease Diagnosis data set



ID3 (Iterative Dichotomizer), C4.5

- ID3 build a decision tree (DT) based on Information Theory
- Idea ask questions whose answers provide the most information
- Example
  (q1) Is it an animal? (divides the search space in two large domains)
  (q2) Is it my Brother? (there is practically no division at all)
- Basic Strategy
  choose splitting attributes with the highest information gain first
- The amount of information associated with an attribute value is related with the probability of occurrence:
  In (q1): the two sets have almost equal probabilities of occurrence
  In (q2): one set has an infinitesimal probability, the other set has a very large probability
- concept used to quantify information: Entropy

- Entropy is used to measure the amount of uncertainty/randomness in the data
- Entropy 0: no randomness, all data belong to the same class
- Objective of DT classification
  iteratively partition the given data set into subsets s.t. all elements in each final subset belong to the same class
- Given probabilities $p_1, \ldots, p_n$ s.t. $\sum_{i=1}^{n} p_i = 1$ the entropy is defined as:

$$H(p_1, \ldots, p_n) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i} = -\sum_{i=1}^{n} p_i \log p_i$$

- $H(p_1, \ldots, p_n)$ is maximized when $p_1 = \cdots = p_n = \frac{1}{n}$
- Suppose we have an initial set T of training samples (tuples) each of which belongs to one of the k possible classes $C_1, \ldots, C_k$.
- For a subset S of tuples from T, we denote by $p(C_i, S)$ the number of samples in S that belong to class C_i. $\quad p(C_1, S) + \cdots + p(C_k, S) = |S|$

- The gain criterion is used to select which attribute to use as splitting attribute.
- The entropy of the set S is computed as: $H(S) = H(\frac{p(C_1,S)}{|S|}, \ldots, \frac{p(C_k,S)}{|S|})$
  ($p(C_i, S)$ are frequences, $\frac{p(C_i,S)}{|S|}$ are probabilities, summing to 1)
- If the test on the splitting attribute X has n outcomes, then T is partitioned in n subsets $T_1, \ldots, T_n$.
- Compute: entropies $H(T_1), \ldots, H(T_n)$, weighted sum $H_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i)$.
- The gain in information by partitioning T according to X is
  $Gain(X) = H(T) - H_X(T)$.
- The gain criterion selects a splitting attribute X s.t. $Gain(X)$ is maximized.

Example 1

Attribute1	Attribute2	Attribute3	Class
A	70	True	CLASS1
A	90	True	CLASS2
A	85	False	CLASS2
A	95	False	CLASS2
A	70	False	CLASS1
B	90	True	CLASS1
B	78	False	CLASS1
B	65	True	CLASS1
B	75	False	CLASS1
C	80	True	CLASS2
C	70	True	CLASS2
C	80	False	CLASS1
C	80	False	CLASS1
C	96	False	CLASS1

The **entropy of the starting set** T (with the Class classification) is calculated via the probabilities $p_1 = \frac{9}{14}$, $p_2 = \frac{5}{14}$ (using log base 2) (T has 14 tuples)

$$H(T) = H(p_1, p_2) = -\frac{9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14} = 0.9402859585$$

Using Attribute1 we divide the initial set T into $k = 3$ subsets T_1, T_2, T_3 corresp. to the 3 values A (5), B (4), C (5).

$H(T_1) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.9709505945$

$H(T_2) = -\frac{4}{4}\log\frac{4}{4} = 0$

$H(T_3) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} = 0.9709505945$

$H_{Attribute1}(T) = \frac{5}{14}H(T_1) + \frac{4}{14}H(T_2) + \frac{5}{14}H(T_3) = 0.6935361390$

$Gain(Attribute1) = H(T) - H_{Attribute1}(T) = 0.2467498195$

Using Attribute3 we divide the initial set T into $k = 2$ subsets T_1, T_2 corresp. to the 2 values True (6), False (8).

$H(T_1) = -\frac{3}{6}\log\frac{3}{6} - \frac{3}{6}\log\frac{3}{6} = 1$

$H(T_2) = -\frac{6}{8}\log\frac{6}{8} - \frac{2}{8}\log\frac{2}{8} = 0.8112781245$

$H_{Attribute3}(T) = \frac{6}{14}H(T_1) + \frac{8}{14}H(T_2) = 0.8921589283$

$Gain(Attribute3) = H(T) - H_{Attribute3}(T) = 0.0481270302$

Based on the gain criterion, the algorithm selects Attribute1 as a splitting attribute

- To find the optimal test, we need to analyze the test on Attribute2, whose values are numerical, "continuous" values (as opposed to the categorical values of Attribute1, Attribute3)
- How do we formulate test questions on continuous attributes?
- There is algorithm to compute an optimal threshold value Z.
- There is only a finite number of values, we can sort then as $\{v_1, \ldots, v_m\}$.
- There are $m - 1$ possible splits on the values of the attribute, all of which must be examined to find an optimal split.
- Often we choose the midpoint $\frac{v_i + v_{i+1}}{2}$ as the threshold. C4.5 chooses v_i (the smallest value) as the threshold, so that this value occurs in the database.

In our example we have the set of $m = 9$ values $\{65, 70, 75, 78, 80, 85, 90, 95, 96\}$.

We need to select one of the $m - 1 = 8$ values, with the highest information gain.

In our example this optimal value is $Z = 80$.

The corresponding test is: `Attribute2 <= 80 or Attribute2 > 80`

Using Attribute2 we divide the initial set T into $k = 2$ subsets T_1, T_2 corresp. to the test `Attribute2 <= 80 or Attribute2 > 80`

$$H(T_1) = -\frac{7}{9}\log\frac{7}{9} - \frac{2}{9}\log\frac{2}{9} = 0.7642045067$$

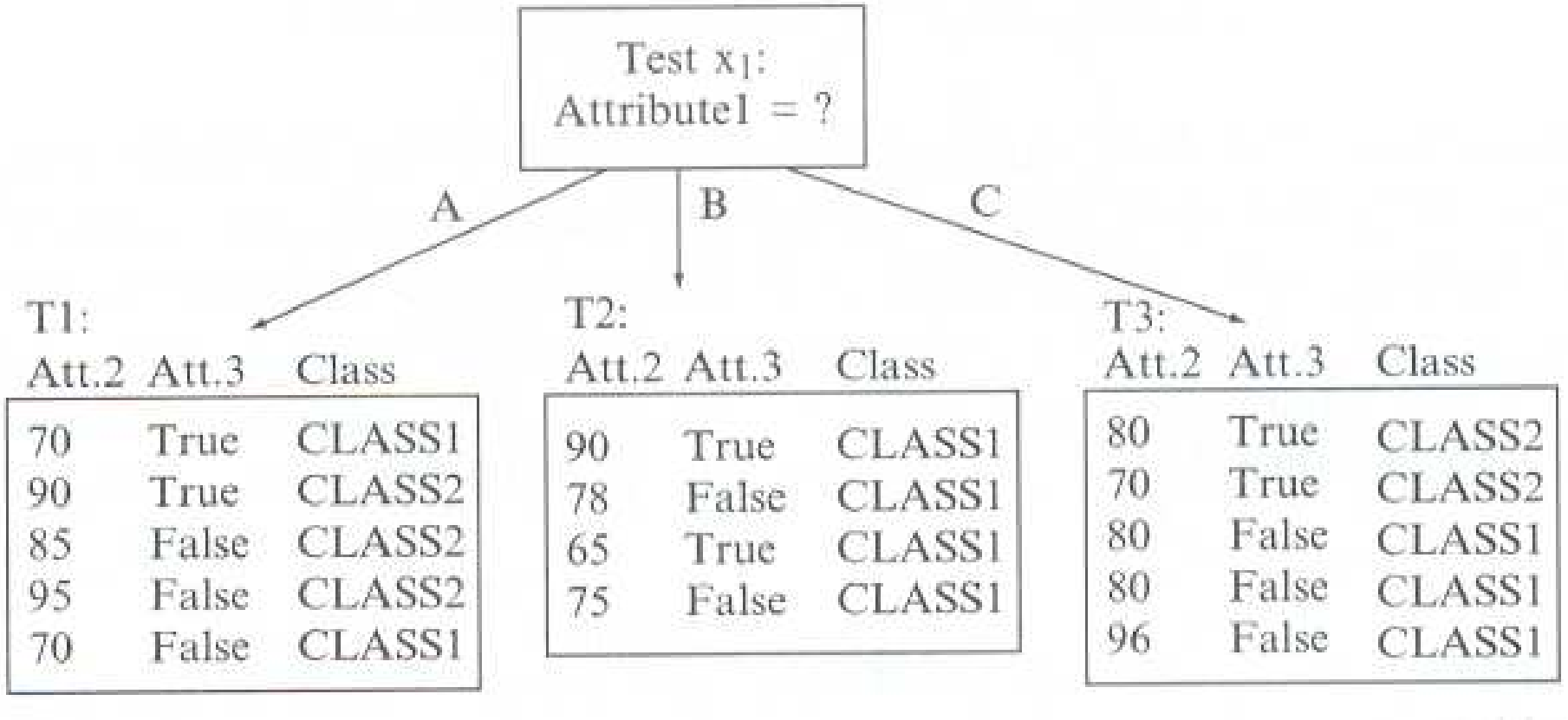
$$H(T_2) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.9709505945$$

$$H_{Attribute2}(T) = \frac{9}{14}H(T_1) + \frac{5}{14}H(T_2) = 0.8380423952$$

$$Gain(Attribute2) = H(T) - H_{Attribute2}(T) = 0.1022435633$$

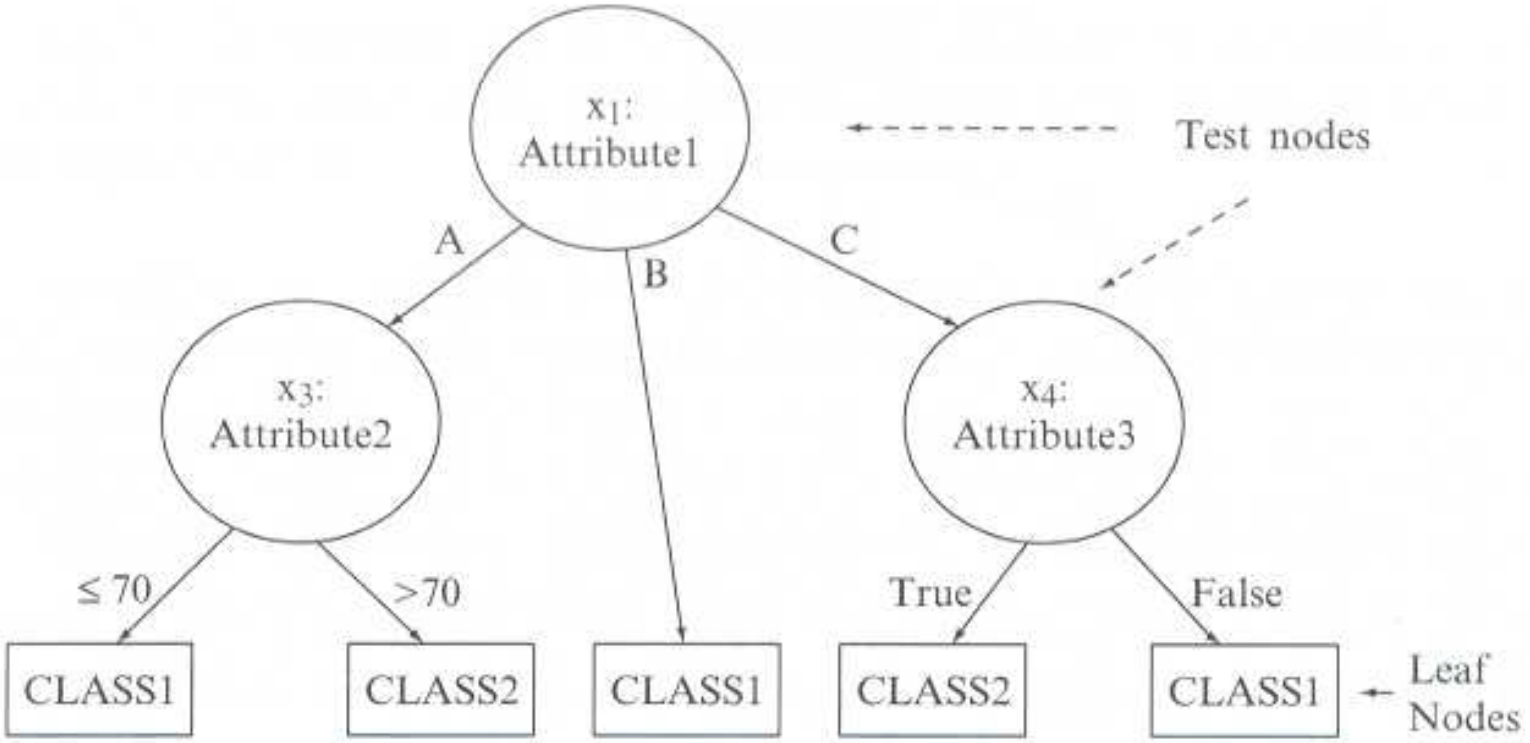
Based on the gain criterion, the algorithm still selects Attribute1 as a splitting attribute

Decision Tree after Attribute1 has been identified as the split. attrib. (highest gain)



- Repeat the process for all (3) children nodes, created from the Attribute1 splitting.
- Remark that the node in the middle will be a leaf node.

Final Decision Tree



- **Serious deficiency** of ID3: favors tests with many outcomes.
- An attribute that has a unique value for each tuple, would be the best splitting attribute, because it would result in a partition with only one tuple in each of the n subsets $T_1, \ldots, T_n$.
- To fix this problem, in C4.5 we perform a **normalization**, to take into account the cardinality of each partition. This typically leads to more compact DTs.
- Instead of: $Gain(X) = H(T) - H_X(T)$
- Now we use: $GainRatio(X) = \dfrac{Gain(X)}{SplitInfo(X)}$, $SplitInfo(X) = H\left(\dfrac{|T_1|}{|T|}, \cdots, \dfrac{|T_n|}{|T|}\right)$
- $SplitInfo(X)$ represents the potential information by dividing the initial set T into n subsets $T_1, \ldots, T_n$.
- The $GainRatio(X)$ measure expresses the proportion of information generated by the splitting, that is useful for classification.
- The $GainRatio$ test selects the attribute X that maximizes the $GainRatio(X)$.
- $SplitInfo(Att1) = -\frac{5}{14}\log\frac{5}{14} - \frac{4}{14}\log\frac{4}{14} - \frac{5}{14}\log\frac{5}{14} = 1.577406282$
- $GainRatio(Att1) = Gain(Att1)/SplitInfo(Att1) = \frac{0.2467498195}{1.577406282} = 0.1564275623$

Example 2

Name	Gender	Height	Output1	Output2
Kristina	F	1.6 m	Short	Medium
Jim	M	2 m	Tall	Medium
Maggie	F	1.9 m	Medium	Tall
Martha	F	1.88 m	Medium	Tall
Stephanie	F	1.7 m	Short	Medium
Bob	M	1.85 m	Medium	Medium
Kathy	F	1.6 m	Short	Medium
Dave	M	1.7 m	Short	Medium
Worth	M	2.2 m	Tall	Tall
Steven	M	2.1 m	Tall	Tall
Debbie	F	1.8 m	Medium	Medium
Todd	M	1.95 m	Medium	Medium
Kim	F	1.9 m	Medium	Tall
Amy	F	1.8 m	Medium	Medium
Wynette	F	1.75 m	Medium	Medium

The **entropy of the starting set** (with the Output1 classification) is calculated via the probabilities $p_1 = \frac{4}{15}$ short, $p_2 = \frac{8}{15}$ medium, $p_3 = \frac{3}{15}$ tall (using log base 10)

$$H(T) = H(p_1, p_2, p_3) = \frac{4}{15}\log\frac{15}{4} + \frac{8}{15}\log\frac{15}{8} + \frac{3}{15}\log\frac{15}{3} = 0.4384696839$$

▷ Choosing the **gender** as a **splitting attribute**: 9 F tuples and 6 M tuples

Entropy of subset of F tuples: $H_F = \frac{3}{9}\log\frac{9}{3} + \frac{6}{9}\log\frac{9}{6} = 0.2764345910$

Entropy of subset of M tuples: $H_M = \frac{1}{6}\log\frac{6}{1} + \frac{2}{6}\log\frac{6}{2} + \frac{3}{6}\log\frac{6}{3} = 0.4392472912$

The information **gain** by using the gender attribute as a splitting attribute is:

$$Gain(Gender) = H(p_1, p_2, p_3) - \left(\frac{9}{15}H_F + \frac{6}{15}H_M\right) = 0.0969100128$$

▷ Choosing the **height** as a **splitting attribute**, we have 11 different values.

We can divide them in the 6 ranges:

$$(0, 1.6], (1.6, 1.7], (1.7, 1.8], (1.8, 1.9], (1.9, 2.0], (2.0, \infty)$$

- Optimized DT: (w.r.t. Output1 classification)

  Height (≤ 1.7 Short) ($> 1.7, \leq 1.95$ Medium) (> 1.95 Tall)
- Experiment with the Output2 classification