Information Gain

- Information gain (IG) measures how much “information” a feature gives us about the class.
  - Features that perfectly partition should give maximal information.
  - Unrelated features should give no information.
- It measures the reduction in \textit{entropy}.
  - Entropy: (im)purity in an arbitrary collection of examples.
Criterion of a Split

Suppose we want to split on the first variable \((x_1)\):

\[
\begin{array}{cccccccccc}
  x_1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  y & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{array}
\]

If we split at \(x_1 < 3.5\), we get an optimal split.
If we split at \(x_1 < 4.5\), we make a mistake (misclassification).

**Idea:** A better split should make the samples “pure” (homogeneous).
Measures for Selecting the Best Split

Impurity measures include:

Entropy:
\[
\text{Entropy} = - \sum_{i=1}^{K} p_k \log_2 p_k
\]

Gini:
\[
\text{Gini} = 1 - \sum_{i=1}^{K} p_k^2
\]

Classification error:
\[
\text{Classification error} = 1 - \max_i p_k
\]

where \( p_k \) denotes the proportion of instances belonging to class \( k \) (\( K = 1, \ldots, k \)), and \( 0 \log_2 0 = 0 \).
What is Entropy?

And how do we compute it?
Consider the following probability space with a random variable that takes four values: \( A, B, C, \) and \( D \).

<table>
<thead>
<tr>
<th></th>
<th>( A )</th>
<th>( B )</th>
<th>( C )</th>
<th>( D )</th>
</tr>
</thead>
</table>

If we select a random variable, it gives us information about its horizontal position.
Measuring Information

Let’s say the horizontal position of the point is represented by a string of zeros and ones.

Larger regions are encoded with fewer bits; smaller regions are encoded with more bits.
Expected Information

The *expected value* is the sum over all values of the product of the probability of the value and the value.

\[
\text{Expected Value} = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3
\]
Expected Information

Each time a region of color got smaller by one half:

– The number of bits of information we got when it did happen went up by one.

– The change that it did happen went down by a factor of \( \frac{1}{2} \).

That is, the information of an event \( x \) is the logarithm of one over its probability:

\[
Information(x) = \log_2 \left( \frac{1}{P(R = x)} \right)
\]
Expected Information

So in general, the expected information or “entropy” of a random variable is the same as the expected value with the Value filled in with the Information:

\[
\text{Entropy of } R = \sum_x P(R = x) \cdot \text{Information}(x)
\]

\[
= \sum_x P(R = x) \cdot \log_2 \left( \frac{1}{P(R = x)} \right)
\]

\[
= -\sum_x P(R = x) \cdot \log_2 P(R = x)
\]
Properties of Entropy

Maximized when elements are heterogeneous (impure):

If $p_k = \frac{1}{k}$, then

$$\text{Entropy} = H = -K \cdot \frac{1}{k} \log_2 \frac{1}{k} = \log_2 K$$

Minimized when elements are homogenous (pure):

If $p_i = 1$ or $p_i = 0$, then

$$\text{Entropy} = H = 0$$
Information Gain

With entropy defined as:

\[ H = - \sum_{i=1}^{K} p_k \log_2 p_k \]

Then the change in entropy, or *Information Gain*, is defined as:

\[ \Delta H = H - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R \]

where \( m \) is the total number of instances, with \( m_k \) instances belonging to class \( k \), where \( K = 1, \ldots, k \).
### Information Gain: Example

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
</tbody>
</table>
# Information Gain: Example

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
</tbody>
</table>

\[
H(Y) = - \sum_{k=1}^{K} p_k \log_2 p_k
\]

\[
= - \frac{5}{14} \log_2 \frac{5}{14} - \frac{9}{14} \log_2 \frac{9}{14}
\]

\[
= 0.94
\]
### Information Gain: Example

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
</tbody>
</table>

\[
\text{InfoGain} (\text{Humidity}) = H(Y) - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R
\]

\[
= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R
\]
### Information Gain: Example

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
</tbody>
</table>

\[
\text{InfoGain}(\text{Humidity}) = H(Y) - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R
\]

\[
0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R
\]

\[
H_L = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7}
\]
## Information Gain: Example

### Data Preprocessing

The table below shows data for different weather conditions (Outlook) and their corresponding play (Play) decision, along with other features such as Temperature, Humidity, and Windy.

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
</tbody>
</table>

### Information Gain Formula

The information gain for the Humidity feature can be calculated as follows:

$$\text{InfoGain}(\text{Humidity}) = H(Y) - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R$$

Where:

- $H(Y)$ is the entropy of the target variable (Play).
- $m_L$ and $m_R$ are the number of samples in the left and right child nodes, respectively.
- $H_L$ and $H_R$ are the entropies of the left and right child nodes, respectively.

Using the example data:

- $H(Y) = 0.94$ (from the entropy calculation of all outcomes).
- $m = 14$ (total number of examples).
- $m_L = 11$, $m_R = 3$ (splitting the examples into two groups).
- $H_L = 0.592$, $H_R = 0.985$ (entropy values for the left and right branches).

Plugging these values into the formula:

$$\text{InfoGain}(\text{Humidity}) = 0.94 - \frac{11}{14} \times 0.592 - \frac{3}{14} \times 0.985$$

$$= 0.94 - 0.64 - 0.13$$

$$= 0.17$$

Thus, the information gain for the Humidity feature is $0.17$. This indicates that Humidity is a useful feature to split on in the decision tree.
### Information Gain: Example

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
</tbody>
</table>

\[
\text{InfoGain(Humidity)} = H(Y) - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R
\]

\[
= 0.94 - \frac{7}{14} 0.592 - \frac{7}{14} 0.985
\]

\[
= 0.94 - 0.296 - 0.4925
\]

\[
= 0.1515
\]
Information Gain: Example

- Information gain for each feature:
  - Outlook = 0.247
  - Temperature = 0.029
  - Humidity = 0.152
  - Windy = 0.048

- Initial split is on outlook, because it is the feature with the highest information gain.
Information Gain: Example

- Now we search for the best split at the next level:

Temperature = 0.571

Windy = 0.020

Humidity = 0.971
Information Gain: Example

• The final decision tree:

Note that not all leaves need to be pure; sometimes similar (even identical) instances have different classes. Splitting stops when data cannot be split any further.
The Gini index is defined as:

$$Gini = 1 - \sum_{i=1}^{K} p_k^2$$

where $p_k$ denotes the proportion of instances belonging to class $k$ ($K = 1, ..., k$).
Gini Index Properties

Maximized when elements are heterogeneous (impure):

If $p_k = \frac{1}{k}$, then

$$Gini = 1 - \sum_{k=1}^{K} \frac{1}{k^2} = 1 - \frac{1}{k}$$

Minimized when elements are homogenous (pure):

If $p_i = 1$ or $p_i = 0$, then

$$Gini = 1 - 1 - 0 = 0$$
Gini Index Example

Suppose we want to split on the first variable ($x_1$):

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
Gini = 1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2 = \frac{15}{32}
\]

If we split at $x_1 < 3.5$: \[
\Delta Gini = \frac{15}{32} - \frac{3}{8} \cdot 0 - \frac{5}{8} \cdot 0 = \frac{15}{32}
\]

If we split at $x_1 < 4.5$: \[
\Delta Gini = \frac{15}{32} - \frac{4}{8} \cdot \frac{3}{8} - \frac{4}{8} \cdot 0 = \frac{9}{32}
\]
Classification Error

The classification error is defined as:

$$\text{Classification error} = 1 - \max_i p_k$$

where $p_k$ denotes the proportion of instances belonging to class $k$ ($K = 1, \ldots, k$).
Classification Error Properties

Tends to create impure nodes:

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Splitting at $b$ has lower classification error than $a$, but results in both nodes being impure.
Splitting Based on Nominal Features

- **Multi-way split**: Use as many partitions as distinct values.

- **Binary split**: Divides values into two subsets. Need to find optimal partitioning.
Splitting Based on Continuous Features

• **Discretization**: Form an ordinal categorical feature
  – Static: discretize once at the beginning (global)
  – Dynamic: discretize ranges at different levels (local)

• **Binary decision**: Consider all possible splits and finds the best cut.
Highly Branching Features

• Features with a large number of values can be problematic
  – e.g.: ID code

• Subsets are more likely to be pure if there are a large number of values
  – Information gain is biased toward choosing features with a large number of values
  – The selection of a feature that is non-optimal for predication can result in overfitting.
# Dataset with Highly Branching Features

<table>
<thead>
<tr>
<th>ID</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>B</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>C</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>D</td>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>E</td>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>G</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>H</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>I</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>J</td>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>K</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>L</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td>M</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>N</td>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
</tbody>
</table>
IG with Highly Branching Features

The entropy of the split is 0, since each leaf node is “pure”, having only one case.
Gain Ratio

- A modification of information gain that reduces its bias on highly branching features.
- It takes into account the number and size of branches when choosing a feature.
- It does this by normalizing information gain by the "intrinsic information" of a split, which is defined as the information need to determine the branch to which an instance belongs.
Intrinsic Information

• The intrinsic information represents the potential information generated by splitting the dataset into \( v \) partitions:

\[
\text{IntrinsicInfo}(D) = - \sum_{j=1}^{v} \frac{|D_j|}{D} \cdot \log_2 \left( \frac{|D_j|}{D} \right)
\]

• High intrinsic info: partitions have more or less the same size
• Low intrinsic info: few partitions hold most of the tuples.
Gain Ratio Defined

• The gain ratio is defined as:

\[
GainRatio(F) = \frac{Gain(F)}{IntinsicInfo(F)}
\]

• The feature with the maximum gain ratio is selected as the splitting feature.
Comparing Feature Selection Measures

• Information Gain
  – Biased toward multivalued features.

• Gain Ratio
  – Tends to prefer unbalanced splits in which one partition is much smaller than the other.

• Gini Index
  – Has difficulties when the number of classes is large.
  – Favors tests that result in equal-sized partitions with purity.

• Classification Error
  – No. Just, no.